

Integrating SAS® and Microsoft Excel: Exploring the Many Options Available to You

Vincent DelGobbo, SAS Institute Inc., Cary, NC

ABSTRACT

This paper explains different techniques available to you when importing and exporting SAS® and Microsoft Excel data. You learn how to import Excel data into SAS using the IMPORT procedure, the SAS DATA step, SAS® Enterprise Guide®, and other methods. Exporting data and analytical results from SAS to Excel is performed using the EXPORT procedure, the SAS DATA step, SAS Enterprise Guide, the SAS Output Delivery System (ODS), and other tools. The material is appropriate for all skill levels, and the techniques work with various versions of SAS software running on the Windows, UNIX (including Linux), and z/OS operating systems. Some techniques require only Base SAS® and others require the SAS/ACCESS® Interface to PC Files.

INTRODUCTION

Because Microsoft Excel is a popular way to store and display data, you're often asked to import Excel data into SAS or to export SAS data and analytical results to Excel. This paper discusses several techniques that SAS provides for importing and exporting Excel data. Some of the techniques require only Base SAS and others require that you license the SAS/ACCESS Interface to PC Files (SAS Institute Inc. [2018k](#)).

The Excel "LabResults.xlsx" file that we will import into SAS is shown in Figure 1.

	A	B	C	D	E	F	G	H	I	J	K
1	Unique Subject Identifier	LAB Test or Examination Name	Age in AGEU at RFSTDC	Age Units	Sex	Baseline Result in Std. Units	Baseline Std. Units	Visit 5 Result in Std. Units	Visit 5 Std. Units	Baseline Collection Date/Time	Visit 5 Collection Date/Time
2	01-701-1034	CALCIUM	77	YEARS	F	2.54490	mmol/L	2.44510	mmol/L	6/24/2014 15:30	7/29/2014 8:40
3	01-701-1133	CALCIUM	81	YEARS	F	2.32035	mmol/L	2.29540	mmol/L	10/23/2012 10:57	11/26/2012 11:52
4	01-701-1146	CALCIUM	75	YEARS	F	2.32035	mmol/L	2.32035	mmol/L	5/7/2013 13:52	6/16/2013 12:45
5	01-701-1287	CALCIUM	56	YEARS	F	2.27045	mmol/L	2.09580	mmol/L	1/17/2014 12:16	2/20/2014 11:09
6	01-701-1383	CALCIUM	72	YEARS	F	2.22055	mmol/L	2.27045	mmol/L	1/22/2013 11:43	3/12/2013 11:50
7	01-703-1258	CALCIUM	78	YEARS	F	2.17065	mmol/L	2.29540	mmol/L	7/10/2012 16:30	8/17/2012 11:15
8	01-703-1295	CALCIUM	88	YEARS	F	2.32035	mmol/L	2.32035	mmol/L	10/24/2013 12:10	12/18/2013 12:45
9	01-703-1335	CALCIUM	67	YEARS	F	2.34530	mmol/L	2.32035	mmol/L	12/28/2013 16:10	4/13/2014 10:00

Figure 1. The "LabResults" Excel Workbook

Some of the column headings and worksheet names contain spaces or special characters such as a period (.) or a forward slash (/). These characters can be problematic when used in SAS data set and variable names, but these problems are easily corrected using the appropriate SAS options, discussed later. Techniques to import the Excel datetime values in the "Baseline Collection Date/Time" and "Visit 5 Collection Date/Time" columns are also discussed.

Table 1 shows some of the attributes of the "LabResults" SAS data set that we want to export to Excel.

Variable	Type	Length	Format	Label
AGE	Numeric	8		Age in AGEU at RFSTDTC
AGEU	Character	6		Age Units
COUNTRY	Character	3		Country
SEX	Character	1		Sex
arm	Character	20		Description of Planned Arm
baseline_lbdtn	Numeric	8	E8601DT19.	Baseline Collection Date/Time
baseline_lbstresn	Numeric	8		Baseline Result in Std. Units
baseline_lbstresu	Character	8		Baseline Std. Units
baseline_visitnum	Numeric	8		Baseline Visit Number
lbtest	Character	38		LAB Test or Examination Name
race	Character	78		Race
studyid	Character	12		Study Identifier
usubjid	Character	11		Unique Subject Identifier
visit5_lbdtn	Numeric	8	E8601DT19.	Visit 5 Collection Date/Time
visit5_lbstresn	Numeric	8		Visit 5 Result in Std. Units
visit5_lbstresu	Character	8		Visit 5 Std. Units
visit5_visitnum	Numeric	8		Visit 5 Visit Number
...

Table 1. Attributes of the Variables of the "LabResults" SAS Data Set

The data values are similar to the values in the "LabResults.xlsx" file in [Figure 1](#). The SAS data set contains two SAS datetime variables ("baseline_lbdtn" and "visit5_lbdtn"), labels, and more variables than we want to export to Excel. The order of the variables in the SAS data set is different than the order we want in our Excel workbook.

The code in this paper was tested using SAS 9.4M5, SAS 9.4M6, SAS Enterprise Guide 7.15 HF7 (7.100.5.6177), and Microsoft Excel 2016 software. The sample data and code are available for download here:

<https://github.com/sascommunities/sas-global-forum-2019/blob/master/2991-2019-DelGobbo>

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

REQUIREMENTS

SAS recommends that you use the .xlsx file format because of its enhancements and the expanding support that it's receiving (SAS Institute Inc. [2018e](#)). The newer SAS/ACCESS

XLSX engine is recommended to read and write .xlsx files instead of the older EXCEL engine. The SAS/ACCESS techniques in this paper use the XLSX engine.

The requirements for using the Base SAS techniques are shown in Table 2.

Technique	SAS Version	SAS Operating System Version	Excel Version
<u>PROC IMPORT</u>	All	Windows, UNIX, Linux	All
<u>DATA Step Import</u>	All	All	All
<u>Enterprise Guide Import</u>	All	All	All
<u>PROC EXPORT</u>	All	Windows, UNIX, Linux	All
<u>DATA Step Export</u>	All	All	All
<u>Enterprise Guide Export</u>	All	All	All
<u>ODS Excel Export</u>	9.4M3 or later	All	2007 or later

Table 2. Requirements for the Base SAS Techniques

The SAS/ACCESS techniques can be used on the Windows, UNIX, and Linux operating systems and require Excel 2007 or later. Table 3 shows the SAS requirements.

Technique	SAS Version
<u>PROC IMPORT</u>	9.3 and later
<u>DATA Step Import</u>	9.4M2 and later
<u>PROC DATASETS Import</u>	9.4M2 and later
<u>PROC EXPORT</u>	9.3 and later
<u>DATA Step Export</u>	9.4M2 and later
<u>PROC DATSETS Export</u>	9.4M2 and later

Table 3. SAS Requirements for the SAS/ACCESS Techniques

IMPORTING EXCEL DATA INTO SAS

In this section we discuss techniques for importing the Excel data shown in [Figure 1](#).

IMPORTING EXCEL DATA USING BASE SAS

Although SAS Enterprise Guide can import *native* Excel files using Base SAS, you can only import *delimited* files using the IMPORT procedure or the DATA step.

Creating Delimited Files Using Excel

You can create delimited files in Excel by opening a workbook and then saving the desired worksheet in a delimited format, such as a comma-separated value (CSV) file.

The steps to create the CSV file used in our sample code follow:

1. In Excel, open the workbook shown in [Figure 1](#).
2. Navigate to the "CALCIUM" worksheet.
3. Select **File > Save As** and then specify the values shown in Figure 2.



Figure 2. Excel Option Values for Creating a CSV File

4. Click **Save** and then exit Excel.

Importing Delimited Files Using the IMPORT Procedure

Use the IMPORT procedure (SAS Institute Inc. [2018f](#)) to import the "Calcium.csv" file.

```
proc import out=work.calcium
  file='Calcium.csv'
  dbms=csv
  replace;
  guessingrows=max;
run; quit;

proc contents data=work.calcium varnum; run; quit;
```

The "calcium" SAS data set is created and replaced if it already existed. The GUESSINGROWS statement directs PROC IMPORT to scan all the rows in each column to determine the data types.

PROC IMPORT uses the Excel column headings for SAS variable names. However some of the resulting variable names, such as "Baseline Collection Date/Time", do not conform to the rules for SAS variable names because they contain spaces or non-alphanumeric characters (SAS Institute Inc. [2018j](#)). The IMPORT procedure alters these names to make them valid SAS names by replacing spaces and non-alphanumeric characters with underscores (_). Compare the variable names in the CONTENTS procedure output in [Figure 3](#) with the column headings in the input data ([Figure 1](#)) to see the changes. All the underscores in [Figure 3](#) are the result of substitutions.

Variables in Creation Order					
#	Variable	Type	Len	Format	Informat
1	Unique_Subject_Identifier	Char	11	\$11.	\$11.
2	LAB_Test_or_Examination_Name	Char	7	\$7.	\$7.
3	Age_in_AGEU_at_RFSTDTC	Num	8	BEST12.	BEST32.
4	Age_Units	Char	5	\$5.	\$5.
5	Sex	Char	1	\$1.	\$1.
6	Baseline_Result_in_Std_Units	Num	8	BEST12.	BEST32.
7	Baseline_Std_Units	Char	6	\$6.	\$6.
8	Visit_5_Result_in_Std_Units	Num	8	BEST12.	BEST32.
9	Visit_5_Std_Units	Char	6	\$6.	\$6.
10	Baseline_Collection_Date_Time	Num	8	DATETIME.	ANYDTDTM40.
11	Visit_5_Collection_Date_Time	Num	8	DATETIME.	ANYDTDTM40.

Figure 3. Attributes and Initial Variable Names for the "calcium" SAS Data Set

The substitutions occur because the VALIDVARNAME option is set to **V7** by default (SAS Institute Inc. [2018n](#)), as you can see in the following code:

```
1  proc options option=validvarname; run;
```

SAS (r) Proprietary Software Release 9.4 TS1M5

VALIDVARNAME=V7 Specifies the rules for valid SAS variable names that can be created and processed during a SAS session.

To use the Excel column headings for the SAS variable names without any substitutions, specify **ANY** for the VALIDVARNAME option before executing the IMPORT procedure.

```
options validvarname=any;
```

```
proc import out=work.calcium
  file='Calcium.csv'
  dbms=csv
  replace;
  guessingrows=max;
run; quit;

proc contents data=work.calcium varnum; run; quit;
```

The SAS variable names shown in the PROC CONTENTS output in [Figure 4](#) match the column headings in the input data ([Figure 1](#)).

Variables in Creation Order					
#	Variable	Type	Len	Format	Informat
1	Unique Subject Identifier	Char	11	\$11.	\$11.
2	LAB Test or Examination Name	Char	7	\$7.	\$7.
3	Age in AGEU at RFSTDTC	Num	8	BEST12.	BEST32.
4	Age Units	Char	5	\$5.	\$5.
5	Sex	Char	1	\$1.	\$1.
6	Baseline Result in Std. Units	Num	8	BEST12.	BEST32.
7	Baseline Std. Units	Char	6	\$6.	\$6.
8	Visit 5 Result in Std. Units	Num	8	BEST12.	BEST32.
9	Visit 5 Std. Units	Char	6	\$6.	\$6.
10	Baseline Collection Date/Time	Num	8	DATETIME.	ANYDTDTM40.
11	Visit 5 Collection Date/Time	Num	8	DATETIME.	ANYDTDTM40.

Figure 4. Attributes and Final Variable Names for the "calcium" SAS Data Set

Reference the variables using SAS name literal syntax (SAS Institute Inc. [2018](#)).

```
options validvarname=any;

proc print data=work.calcium;
  var sex
      'unique subject identifier'n
      'baseline result in std. units'n
      'baseline collection date/time'n;
run; quit;
```

A portion of the PROC PRINT output showing the final variable names is in [Figure 5](#).

Obs	Sex	Unique Subject Identifier	Baseline Result in Std. Units	Baseline Collection Date/Time
1	F	01-701-1034	2.5449	24JUN14:15:30:00
2	F	01-701-1133	2.32035	23OCT12:10:57:00
3	F	01-701-1146	2.32035	07MAY13:13:52:00
4	F	01-701-1287	2.27045	17JAN14:12:16:00
5	F	01-701-1383	2.22055	22JAN13:11:43:00
6	F	01-703-1258	2.17065	10JUL12:16:30:00

Figure 5. Partial PRINT Procedure Results with the Final Variable Names

You might want to use an ISO date format, such as the E8601DT format, for the datetime columns (SAS Institute Inc. [2018b](#)). PROC IMPORT does not provide an option to specify formats when importing delimited files, but you can use the DATASETS procedure to change the formats after the Excel data is imported (SAS Institute Inc. [2018h](#) and [2018d](#)).

```
options validvarname=any;

proc datasets lib=work nolist;
  modify calcium;
  format 'Baseline Collection Date/Time'n
        'Visit 5 Collection Date/Time'n e8601dt19.;
run; quit;
```

An alternate method for formatting the datetime variables is provided in the [Importing Delimited Files Using the DATA Step](#) section.

Importing Delimited Files Using the DATA Step

The IMPORT procedure generates and submits DATA step code to import data from delimited files. When using the SAS windowing environment (SAS Institute Inc. [2018o](#)), you can recall the generated code and then make modifications to change the way that the data is imported. For example, you can specify a format for the datetime columns.

In SAS, select **Run > Recall Last Submit** to recall the previously submitted code. The editor widow contains the DATA step code used to perform the import, *not* the PROC IMPORT code that you submitted. Locate the FORMAT statements for the baseline and visit 5 collection datetime columns in the generated code.

```
format "Baseline Collection Date/Time"n datetime.;
format "Visit 5 Collection Date/Time"n datetime.;
```

Add the desired format.

```
format "Baseline Collection Date/Time"n e8601dt19.;
format "Visit 5 Collection Date/Time"n e8601dt19.;
```

Submit the code and then use the PRINT procedure to verify that the datetime values are now formatted using the E8601DT19 format ([Figure 6](#)).

Obs	Sex	Unique Subject Identifier	Baseline Result in Std. Units	Baseline Collection Date/Time
1	F	01-701-1034	2.5449	2014-06-24T15:30:00
2	F	01-701-1133	2.32035	2012-10-23T10:57:00
3	F	01-701-1146	2.32035	2013-05-07T13:52:00
4	F	01-701-1287	2.27045	2014-01-17T12:16:00
5	F	01-701-1383	2.22055	2013-01-22T11:43:00
6	F	01-703-1258	2.17065	2012-07-10T16:30:00

Figure 6. Partial PRINT Procedure Results Using the E8601DT Format

Because you have full control of the generated import code, you can perform other tasks, such as creating a new variable by adding the following to the DATA step code:

```
length 'Age Group' $7;

if ('Age in AGEU at RFSTDTC' < 65)
  then 'Age Group' = '< 65';
else if ('Age in AGEU at RFSTDTC' <= 75)
  then 'Age Group' = '65 - 75';
else 'Age Group' = '> 75';
```

Importing Excel Files Using SAS Enterprise Guide

If Excel is installed on the same machine as SAS Enterprise Guide, then you can use Base SAS through SAS Enterprise Guide to import *native* Excel files without first saving a worksheet as a delimited file.

You can import an entire worksheet, a range of cells within the specified worksheet, or a predefined named range within the workbook ([Figure 7](#)).

In SAS Enterprise Guide, select **File > Import Data** to start the Import Data task. Navigate to "LabResults.xlsx", and then follow the prompts. Select the **CALCIUM** worksheet and then click **Next**.

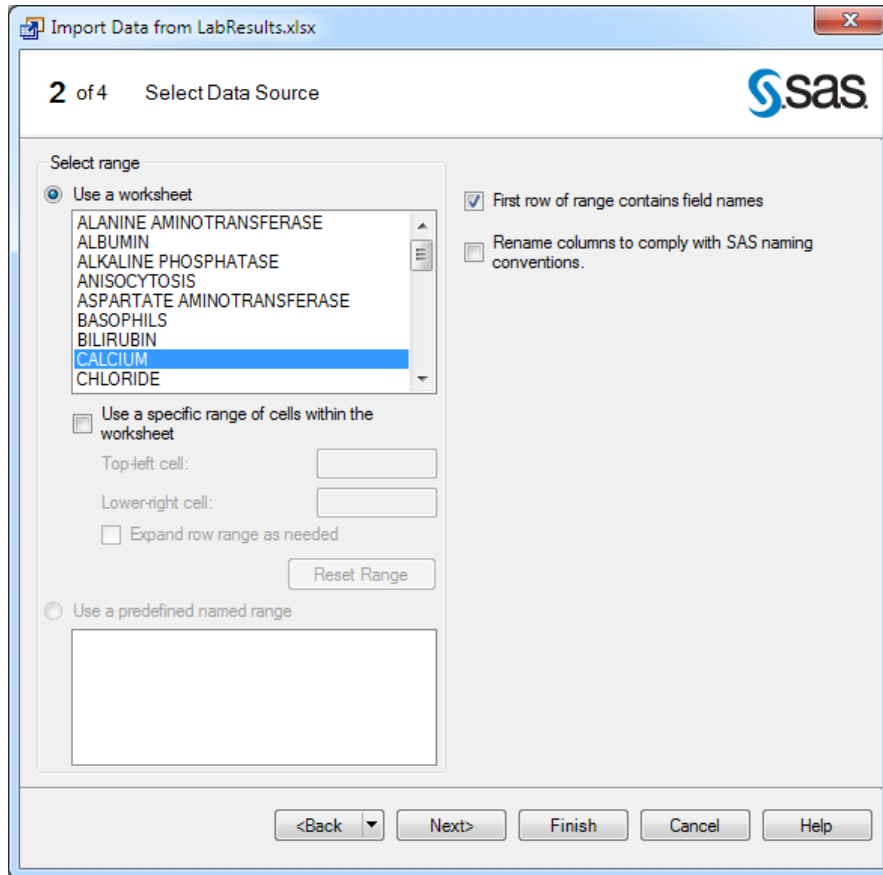


Figure 7. SAS Enterprise Guide "Select Data Source" Window

The Excel column headings are used verbatim for the SAS variable names. By default, SAS Enterprise Guide uses **DATETIME18.** for the output format of the datetime columns ([Figure 8](#)). To change the output format, select the **Baseline Collection Date/Time** and **Visit 5 Collection Date/Time** rows as shown in [Figure 8](#), and then click **Modify**.

Select columns and define attributes:

Inc	Source Name	Name	Label	Type	Source Informat	Len.	Output Format	Output Informat
<input checked="" type="checkbox"/>	Unique Subject ...	Unique Subject Identifier	Unique Subject Identif...	String	\$CHAR11.	11	\$CHAR11.	\$CHAR11.
<input checked="" type="checkbox"/>	LAB Test or Ex...	LAB Test or Examination Name	LAB Test or Examinati...	String	\$CHAR7.	7	\$CHAR7.	\$CHAR7.
<input checked="" type="checkbox"/>	Age in AGEU at...	Age in AGEU at RFSTDTC	Age in AGEU at RFST...	Number	BEST12.	8	BEST12.	BEST12.
<input checked="" type="checkbox"/>	Age Units	Age Units	Age Units	String	\$CHAR5.	5	\$CHAR5.	\$CHAR5.
<input checked="" type="checkbox"/>	Sex	Sex	Sex	String	\$CHAR1.	1	\$CHAR1.	\$CHAR1.
<input checked="" type="checkbox"/>	Baseline Result ...	Baseline Result in Std. Units	Baseline Result in Std....	Number	BEST12.	8	BEST12.	BEST12.
<input checked="" type="checkbox"/>	Baseline Std. U...	Baseline Std. Units	Baseline Std. Units	String	\$CHAR6.	6	\$CHAR6.	\$CHAR6.
<input checked="" type="checkbox"/>	Visit 5 Result in ...	Visit 5 Result in Std. Units	Visit 5 Result in Std. U...	Number	BEST12.	8	BEST12.	BEST12.
<input checked="" type="checkbox"/>	Visit 5 Std. Units	Visit 5 Std. Units	Visit 5 Std. Units	String	\$CHAR6.	6	\$CHAR6.	\$CHAR6.
<input checked="" type="checkbox"/>	Baseline Collect...	Baseline Collection Date/Time	Baseline Collection Da...	Date/Time	DATETIME18.	8	DATETIME18.	DATETIME18.
<input checked="" type="checkbox"/>	Visit 5 Collectio...	Visit 5 Collection Date/Time	Visit 5 Collection Date...	Date/Time	DATETIME18.	8	DATETIME18.	DATETIME18.

Select All Clear All Modify...

Figure 8. SAS Variable Attributes for the Imported Excel Data

Click the button next to the output format, and then specify the **E8601DTw.d** format with an **Overall width** of **19**, as shown in Figure 9.

The image shows two overlapping dialog boxes. The left dialog, titled "Field Attributes for Multiple Selections", has the "Type" set to "Date/Time". Under "Output attributes", the "Length" is set to 8 and the "Output format" is set to DATETIME18. A blue circle highlights the ellipsis button next to the output format. The right dialog, titled "Output Data Set Format", shows the "Date/Time" category selected in the "Categories" list and the "E8601DTw.d" format selected in the "Formats" list. Below, the "Attributes" section shows "Overall width" set to 19 and "Decimal places" set to 0. The "Description" is "Extended ISO 8601 datetime from yyyyymmddThhmmssnnnnn" and the "Example" shows the value "1236688496 (10Mar1999 12:34:56)" resulting in the output "1999-03-10T12:34:56".

Figure 9. Specifying a SAS Format for the Datetime Variables

Click **OK** to close the open dialogs, and then proceed through the remaining prompts. The datetime values are formatted using the E8601DT format.

If you encounter an error indicating that you do not have the correct Microsoft Office components installed on your machine, then you might need to update your version of SAS Enterprise Guide. Refer to the **Hotfix** tab in SAS Problem Note 61224 for details (SAS Institute Inc. [2017](#)).

IMPORTING EXCEL DATA USING THE SAS/ACCESS INTERFACE TO PC FILES

The SAS/ACCESS Interface to PC Files enables you to import Excel data directly from an XLSX file, without first saving the data in a delimited file. The techniques in this section import the Excel XLSX file shown in [Figure 1](#).

Importing Excel Files Using the IMPORT Procedure

To read data from the "LabResults.xlsx" file ([Figure 1](#)), use PROC IMPORT and the XLSX engine. The first worksheet is imported by default. Use the SHEET statement to import a specific sheet.

```
options validvarname=any;

proc import out=work.calcium
  file='LabResults.xlsx'
  dbms=xlsx
  replace;
  sheet='calcium';
run; quit;

proc contents data=work.calcium varnum; run; quit;
```

The GUESSINGROWS statement that we used when [importing a delimited file](#) is not supported and, in this situation, not needed, because the XLSX engine scans all the rows in each column to determine the data types.

The CONTENTS procedure output in [Figure 10](#) shows that the SAS variable names match the column headings in the input data ([Figure 1](#)).

Variables in Creation Order						
#	Variable	Type	Len	Format	Informat	Label
1	Unique Subject Identifier	Char	11	\$11.	\$11.	Unique Subject Identifier
2	LAB Test or Examination Name	Char	7	\$7.	\$7.	LAB Test or Examination Name
3	Age in AGEU at RFSTDTC	Num	8	BEST.		Age in AGEU at RFSTDTC
4	Age Units	Char	5	\$5.	\$5.	Age Units
5	Sex	Char	1	\$1.	\$1.	Sex
6	Baseline Result in Std. Units	Num	8	BEST.		Baseline Result in Std. Units
7	Baseline Std. Units	Char	6	\$6.	\$6.	Baseline Std. Units
8	Visit 5 Result in Std. Units	Num	8	BEST.		Visit 5 Result in Std. Units
9	Visit 5 Std. Units	Char	6	\$6.	\$6.	Visit 5 Std. Units
10	Baseline Collection Date/Time	Num	8	DATETIME16.		Baseline Collection Date/Time
11	Visit 5 Collection Date/Time	Num	8	DATETIME16.		Visit 5 Collection Date/Time

Figure 10. Attributes of the Variables for the "calcium" SAS Data Set

To apply a format to the datetime columns, use a FORMAT statement.

```
proc import out=work.calcium
  file='LabResults.xlsx'
  dbms=xlsx
  replace;
  sheet='calcium';
  format 'visit 5 collection date/time'n
         'baseline collection date/time'n e8601dt19.;
run; quit;
```

Importing Excel Files Using the XLSX LIBNAME Engine and the DATA Step

When you use the XLSX LIBNAME engine, each worksheet is interpreted as a SAS data set in the library (SAS Institute Inc. 2018g). This means that you can access worksheet data like a SAS data set.

Individual worksheets can be imported using the DATA step. The following code imports the "CALCIUM" worksheet and specifies a format for the SAS datetime values:

```
options validvarname=any;

libname xl xlsx 'LabResults.xlsx';

data work.calcium;
set xl.calcium;
attrib 'baseline collection date/time'n
       'visit 5 collection date/time'n format=e8601dt19.;
run;
```

The SAS variable names match the column headings in the input data ([Figure 1](#)).

Importing Excel Files Using the XLSX LIBNAME Engine and the DATASETS Procedure

It's cumbersome to import all the worksheets in a workbook to SAS data sets using PROC IMPORT or the DATA step because the process needs to be repeated for each worksheet. Instead, use the DATASETS procedure to copy all the worksheets to individual data sets in a SAS library. The Excel column headings and worksheet names are used verbatim for the SAS variable names and data set names.

```
options validvarname=any;

libname xl xlsx 'LabResults.xlsx';

proc datasets nolist;
  copy in=xl out=work;
run; quit;
```

When worksheet names don't conform to the rules for SAS data set names (SAS Institute Inc. 2018j), you'll get an error. For example, the space character in the worksheet named "ALANINE AMINOTRANSFERASE" produces this error:

```
ERROR: The value 'ALANINE AMINOTRANSFERASE'n is not a valid SAS name.
ERROR: File WORK.'ALANINE AMINOTRANSFERASE'n.DATA has not been saved because
       copy could not be completed.
```

The errors occur because the VALIDMEMNAME option is set to **COMPAT** by default (SAS Institute Inc. 2018m), as you can see in the following code:

```
1  proc options option=validmemname; run;

SAS (r) Proprietary Software Release 9.4 TS1M5

VALIDMEMNAME=COMPAT Specifies the rules for naming SAS data sets, SAS data
views, and item stores.
```

Earlier, we specified the VALIDVARNAME option to support variable names with embedded spaces and special characters. Here, we want to enable embedded spaces and special characters in data set names, so we use the VALIDMEMNAME option.

```
options validvarname=any validmemname=extend;

libname xl xlsx 'LabResults.xlsx';

proc datasets nolist;
  copy in=xl out=work;
run; quit;
```

All the worksheets are successfully imported to SAS data sets in the WORK library. Reference the data sets using SAS name literal syntax.

```
proc print data=work.'alanine aminotransferase'n;
  ... ;
run; quit;
```

EXPORTING SAS DATA TO EXCEL

In this section we discuss techniques for exporting SAS data. See [Table 1](#) for information about the "LabResults" SAS data set.

EXPORTING SAS DATA USING BASE SAS

Although SAS Enterprise Guide and the ODS EXCEL destination can create *native* Excel files with only Base SAS, you can only create *delimited* files using the EXPORT procedure or the DATA step. Then you can use Excel to open the delimited files.

Exporting SAS Data to Delimited Files Using the EXPORT Procedure

You can export an entire SAS data set to a CSV file using PROC EXPORT (SAS Institute Inc. 2018c).

```
proc export data=sample.LabResults
  file='LabResults.csv'
  dbms=csv
  replace;
run; quit;
```

Although PROC EXPORT lets you specify data set options such as WHERE, DROP, and KEEP for the input data set, you cannot choose the order in which variables are written to the file. To create a customized output file, create a view with the SQL procedure.

The following code selects variables in a specific order from the "LabResults" data set, but only includes data for the calcium lab test. The resulting data is exported to a CSV file.

```
proc sql;
  create view work.LabResults_Subset as
  select usubjid, lbtest, age, ageu, sex, baseline_lbstresn,
         baseline_lbstresu, visit5_lbstresn, visit5_lbstresu,
         baseline_lbdtn, visit5_lbdtn
  from sample.LabResults
  where (lbtest eq 'CALCIUM');
quit;

proc export data=work.LabResults_Subset
  file='Calcium.csv'
  dbms=csv
  replace;
run; quit;
```

Excel interprets the SAS datetime values as text if you double-click the CSV file to open it (Figure 11).

	A	B	C	D	E	F	G	H	I	J	K
1	usubjid	lbtest	AGE	AGEU	SEX	baseline_lbstresn	baseline_lbstresu	visit5_lbstresn	visit5_lbstresu	baseline_lbdtn	visit5_lbdtn
2	01-701-1034	CALCIUM	77	YEARS	F	2.5449	mmol/L	2.4451	mmol/L	2014-06-24T15:30:00	2014-07-29T08:40:00
3	01-701-1133	CALCIUM	81	YEARS	F	2.32035	mmol/L	2.2954	mmol/L	2012-10-23T10:57:00	2012-11-26T11:52:00
4	01-701-1146	CALCIUM	75	YEARS	F	2.32035	mmol/L	2.32035	mmol/L	2013-05-07T13:52:00	2013-06-16T12:45:00
5	01-701-1287	CALCIUM	56	YEARS	F	2.27045	mmol/L	2.0958	mmol/L	2014-01-17T12:16:00	2014-02-20T11:09:00
6	01-701-1383	CALCIUM	72	YEARS	F	2.22055	mmol/L	2.27045	mmol/L	2013-01-22T11:43:00	2013-03-12T11:50:00
7	01-703-1258	CALCIUM	78	YEARS	F	2.17065	mmol/L	2.2954	mmol/L	2012-07-10T16:30:00	2012-08-17T11:15:00
8	01-703-1295	CALCIUM	88	YEARS	F	2.32035	mmol/L	2.32035	mmol/L	2013-10-24T12:10:00	2013-12-18T12:45:00
9	01-703-1335	CALCIUM	67	YEARS	F	2.3453	mmol/L	2.32035	mmol/L	2013-12-28T16:10:00	2014-04-13T10:00:00

Figure 11. "Calcium.csv" File Displaying Datetime Values as Text

For Excel to properly interpret the SAS datetime values, from Excel, you need to select the **Data** tab of the ribbon, and then select **From Text/CSV** (Figure 12).

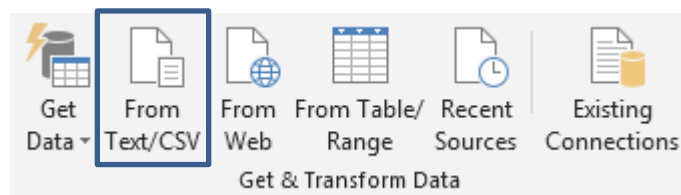


Figure 12. "Get & Transform Data" Group of the Microsoft Excel "Data" Tab

The SAS datetime values are now interpreted as Excel datetime values (Figure 13).

	A	B	C	D	E	F	G	H	I	J	K
1	usubjid	lbttest	AGE	AGEU	SEX	baseline_lbstresn	baseline_lbstresu	visit5_lbstresn	visit5_lbstresu	baseline_lbdtn	visit5_lbdtn
2	01-701-1034	CALCIUM	77	YEARS	F	2.5449	mmol/L	2.4451	mmol/L	6/24/2014 15:30	7/29/2014 8:40
3	01-701-1133	CALCIUM	81	YEARS	F	2.32035	mmol/L	2.2954	mmol/L	10/23/2012 10:57	11/26/2012 11:52
4	01-701-1146	CALCIUM	75	YEARS	F	2.32035	mmol/L	2.32035	mmol/L	5/7/2013 13:52	6/16/2013 12:45
5	01-701-1287	CALCIUM	56	YEARS	F	2.27045	mmol/L	2.0958	mmol/L	1/17/2014 12:16	2/20/2014 11:09
6	01-701-1383	CALCIUM	72	YEARS	F	2.22055	mmol/L	2.27045	mmol/L	1/22/2013 11:43	3/12/2013 11:50
7	01-703-1258	CALCIUM	78	YEARS	F	2.17065	mmol/L	2.2954	mmol/L	7/10/2012 16:30	8/17/2012 11:15
8	01-703-1295	CALCIUM	88	YEARS	F	2.32035	mmol/L	2.32035	mmol/L	10/24/2013 12:10	12/18/2013 12:45
9	01-703-1335	CALCIUM	67	YEARS	F	2.3453	mmol/L	2.32035	mmol/L	12/28/2013 16:10	4/13/2014 10:00

Figure 13. "Calcium.csv" File Correctly Displaying Datetime Values

If you want to use SAS variable labels for the Excel column headings, use the LABEL option (Figure 14).

```
proc export data=work.LabResults_Subset
  file='Calcium.csv'
  dbms=csv
  replace
  label;
run; quit;
```

	A	B	C	D	E	F	G	H	I	J	K
1	Unique Subject Identifier	LAB Test or Examination Name	Age in AGEU at RFSTDT	Age Units	Sex	Baseline Result in Std. Units	Baseline Std. Units	Visit 5 Result in Std. Units	Visit 5 Std. Units	Baseline Collection Date/Time	Visit 5 Collection Date/Time
2	01-701-1034	CALCIUM	77	YEARS	F	2.5449	mmol/L	2.4451	mmol/L	6/24/2014 15:30	7/29/2014 8:40
3	01-701-1133	CALCIUM	81	YEARS	F	2.32035	mmol/L	2.2954	mmol/L	10/23/2012 10:57	11/26/2012 11:52
4	01-701-1146	CALCIUM	75	YEARS	F	2.32035	mmol/L	2.32035	mmol/L	5/7/2013 13:52	6/16/2013 12:45
5	01-701-1287	CALCIUM	56	YEARS	F	2.27045	mmol/L	2.0958	mmol/L	1/17/2014 12:16	2/20/2014 11:09
6	01-701-1383	CALCIUM	72	YEARS	F	2.22055	mmol/L	2.27045	mmol/L	1/22/2013 11:43	3/12/2013 11:50
7	01-703-1258	CALCIUM	78	YEARS	F	2.17065	mmol/L	2.2954	mmol/L	7/10/2012 16:30	8/17/2012 11:15
8	01-703-1295	CALCIUM	88	YEARS	F	2.32035	mmol/L	2.32035	mmol/L	10/24/2013 12:10	12/18/2013 12:45
9	01-703-1335	CALCIUM	67	YEARS	F	2.3453	mmol/L	2.32035	mmol/L	12/28/2013 16:10	4/13/2014 10:00

Figure 14. "Calcium.csv" File Displaying SAS Variable Labels Used for Column Headings

Exporting SAS Data to Delimited Files Using the DATA Step

Like PROC IMPORT, the EXPORT procedure generates and submits DATA step code when creating delimited files. When using the SAS windowing environment, you can recall the generated code and then make changes if PROC EXPORT doesn't meet your needs.

Select **Run > Recall Last Submit** to recall the previously submitted code. The editor window contains the DATA step code used to perform the export, *not* the PROC EXPORT code that you submitted.

Exporting SAS Data to Excel Files Using SAS Enterprise Guide

If Excel is installed on the same machine as SAS Enterprise Guide, then you can create *native* Excel files using only Base SAS.

In SAS Enterprise Guide, follow these steps to export the "LabResults" SAS data set to an Excel XLSX file:

1. Add the "LabResults" data set to the project.
2. Right-click the SAS data set node and then select **Export > Export LABRESULTS**.
3. In the **Files of type** drop-down list, select **Microsoft Excel Workbooks (*.xlsx)**.
4. Choose a location, enter a file name, and then click **Save**.

All the columns and rows are exported to a single worksheet named "LABRESULTS", and Excel interprets the SAS datetime values as Excel datetime values. SAS variable names, not labels, are used for the Excel column headings.

If you encounter an error indicating that you do not have the correct Microsoft Office components installed on your machine, then you might need to update your version of SAS Enterprise Guide. Refer to the **Hotfix** tab in SAS Problem Note 61224 for details (SAS Institute Inc. [2017](#)).

Exporting SAS Data to Excel Files Using the ODS EXCEL Destination

Any procedure output can be exported to Excel using the ODS EXCEL destination. An advantage of this technique is that you can create attractive, *native* multi-sheet workbooks with little effort.

The following code uses the PRINT procedure to create an XLSX file named "LabResults.xlsx" with one sheet for each lab test.

```
ods _all_ close;

ods Excel file='LabResults.xlsx'
  options(sheet_name='#byval(lbtest)' suppress_bylines='yes');

proc print data=sample.LabResults noobs;
  by lbtest;
  var usubjid lbtest age ageu sex baseline_lbstresn baseline_lbstresu
      visit5_lbstresn visit5_lbstresu baseline_lbdtn visit5_lbdtn;
run; quit;

ods Excel close;
```

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

The second ODS statement uses the EXCEL destination to generate the XLSX output and then stores the output in a file. The BY statement in the PRINT procedure ensures that a separate worksheet is created for each distinct value of the "lbtest" BY variable. We use **#byval** (SAS Institute Inc. [2018a](#)) to automatically name the worksheets based on the

value of the "lbtest" variable. The **suppress_bylines** option specifies not to include BY line text in the output.

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

SAS variable names, not labels, are used for the Excel column headings (Figure 15).

	A	B	C	D	E	F	G	H	I	J	K
1	usubjid	lbtest	AGE	AGEU	SEX	baseline_lbstresn	baseline_lbstresu	visit5_lbstresn	visit5_lbstresu	baseline_lbdtn	visit5_lbdtn
2	01-701-1034	CALCIUM	77	YEARS	F	2.54490 mmol/L		2.44510 mmol/L		2014-06-24T15:30:00	2014-07-29T08:40:00
3	01-701-1133	CALCIUM	81	YEARS	F	2.32035 mmol/L		2.29540 mmol/L		2012-10-23T10:57:00	2012-11-26T11:52:00
4	01-701-1146	CALCIUM	75	YEARS	F	2.32035 mmol/L		2.32035 mmol/L		2013-05-07T13:52:00	2013-06-16T12:45:00
5	01-701-1287	CALCIUM	56	YEARS	F	2.27045 mmol/L		2.09580 mmol/L		2014-01-17T12:16:00	2014-02-20T11:09:00
6	01-701-1383	CALCIUM	72	YEARS	F	2.22055 mmol/L		2.27045 mmol/L		2013-01-22T11:43:00	2013-03-12T11:50:00
7	01-703-1258	CALCIUM	78	YEARS	F	2.17065 mmol/L		2.29540 mmol/L		2012-07-10T16:30:00	2012-08-17T11:15:00
8	01-703-1295	CALCIUM	88	YEARS	F	2.32035 mmol/L		2.32035 mmol/L		2013-10-24T12:10:00	2013-12-18T12:45:00
9	01-703-1335	CALCIUM	67	YEARS	F	2.34530 mmol/L		2.32035 mmol/L		2013-12-28T16:10:00	2014-04-13T10:00:00

Figure 15. Initial Excel Workbook Created Using the ODS EXCEL Destination

Important note: In SAS 9.4 M5 and earlier, the SAS datetime column values are interpreted by Excel as text strings (Figure 15). This issue has been corrected in SAS 9.4 M6. If you are using SAS 9.4 M5 or earlier, use the DATETIME format instead of E8610DT to overcome this problem.

```
ods _all_ close;

ods Excel file='LabResults.xlsx'
  options(sheet_name='#byval(lbtest)' suppress_bylines='yes');

proc print data=sample.LabResults noobs;
  by lbtest;
  var usubjid lbtest age ageu sex baseline_lbstresn baseline_lbstresu
      visit5_lbstresn visit5_lbstresu baseline_lbdtn visit5_lbdtn;
  format baseline_lbdtn visit5_lbdtn datetime18.;
run; quit;

ods Excel close;
```

Specify the LABEL option to use SAS variable labels for the Excel column headings, and use a style override to center the column headings.

```
ods _all_ close;

ods Excel file='LabResults.xlsx'
  options(sheet_name='#byval(lbtest)' suppress_bylines='yes');

proc print data=sample.LabResults noobs label style(header)=[just=center];
  by lbtest;
  var usubjid lbtest age ageu sex baseline_lbstresn baseline_lbstresu
      visit5_lbstresn visit5_lbstresu baseline_lbdtn visit5_lbdtn;
  format baseline_lbdtn visit5_lbdtn datetime18.;
run; quit;

ods Excel close;
```

To specify the *Excel* format applied to a column, use the TAGATTR attribute in a style override. Isolate the variable in a separate VAR statement, and then apply the style override. Here we specify the Excel **m/d/yyyy h:mm** format for the two datetime variables.

```
ods _all_ close;

ods Excel file='LabResults.xlsx'
  options(sheet_name='#byval(lbtest)' suppress_bylines='yes');

proc print data=sample.LabResults noobs label style(header)=[just=center];
  by lbtest;
  var usubjid lbtest age ageu sex baseline_lbstresn baseline_lbstresu
      visit5_lbstresn visit5_lbstresu;
  var baseline_lbdtn
      visit5_lbdtn / style(column)=[tagattr='format:m/d/yyyy h:mm'];
  format baseline_lbdtn visit5_lbdtn datetime18.;
run; quit;

ods Excel close;
```

The final "LabResults.xlsx" file is shown in Figure 16.

	A	B	C	D	E	F	G	H	I	J	K
1	Unique Subject Identifier	LAB Test or Examination Name	Age in AGEU at RFSTDT	Age Units	Sex	Baseline Result in Std. Units	Baseline Std. Units	Visit 5 Result in Std. Units	Visit 5 Std. Units	Baseline Collection Date/Time	Visit 5 Collection Date/Time
2	01-701-1034	CALCIUM	77	YEARS	F	2.54490	mmol/L	2.44510	mmol/L	6/24/2014 15:30	7/29/2014 8:40
3	01-701-1133	CALCIUM	81	YEARS	F	2.32035	mmol/L	2.29540	mmol/L	10/23/2012 10:57	11/26/2012 11:52
4	01-701-1146	CALCIUM	75	YEARS	F	2.32035	mmol/L	2.32035	mmol/L	5/7/2013 13:52	6/16/2013 12:45
5	01-701-1287	CALCIUM	56	YEARS	F	2.27045	mmol/L	2.09580	mmol/L	1/17/2014 12:16	2/20/2014 11:09
6	01-701-1383	CALCIUM	72	YEARS	F	2.22055	mmol/L	2.27045	mmol/L	1/22/2013 11:43	3/12/2013 11:50
7	01-703-1258	CALCIUM	78	YEARS	F	2.17065	mmol/L	2.29540	mmol/L	7/10/2012 16:30	8/17/2012 11:15
8	01-703-1295	CALCIUM	88	YEARS	F	2.32035	mmol/L	2.32035	mmol/L	10/24/2013 12:10	12/18/2013 12:45
9	01-703-1335	CALCIUM	67	YEARS	F	2.34530	mmol/L	2.32035	mmol/L	12/28/2013 16:10	4/13/2014 10:00

Figure 16. Final Excel Workbook Created Using the ODS EXCEL Destination

For extensive information about using ODS to create Excel files, refer to earlier papers by this author (DelGobbo 2018).

EXPORTING SAS DATA USING THE SAS/ACCESS INTERFACE TO PC FILES

The SAS/ACCESS Interface to PC Files enables you to create *native* Excel files. The techniques in this section export SAS data (Table 1) to Excel XLSX files.

Exporting SAS Data to Excel Files Using the EXPORT Procedure

The following code first creates a "LabResults_Subset" SAS view, selecting variables in a specific order. Then a workbook named "LabResults.xlsx" with data for all values of the "lbtest" variable is created, and the SAS variable labels are used for the Excel column

headings.

```
proc sql;
  create view work.LabResults_Subset as
  select usubjid, lbtest, age, ageu, sex, baseline_lbstresn,
         baseline_lbstresu, visit5_lbstresn, visit5_lbstresu,
         baseline_lbdtn, visit5_lbdtn
  from sample.LabResults;
quit;

proc export data=work.LabResults_Subset
  file='LabResults.xlsx'
  dbms=xlsx
  replace
  label;
run; quit;
```

The "LabResults.xlsx" file is not replaced if it already exists. A "LabResults" worksheet is replaced if it exists, otherwise a new worksheet is added to the workbook. Delete the existing file before running PROC EXPORT if you want to create a new workbook.

To create a multi-sheet workbook with worksheets for specific values of "lbtest", use the WHERE data set option and the EXPORT procedure SHEET statement.

```
proc export data=work.LabResults_Subset (where=(lbtest eq 'CALCIUM'))
  file='LabResults.xlsx'
  dbms=xlsx
  replace
  label;
  sheet='CALCIUM';
run; quit;

proc export data=work.LabResults_Subset (where=(lbtest eq 'EOSINOPHILS'))
  file='LabResults.xlsx'
  dbms=xlsx
  replace
  label;
  sheet='EOSINOPHILS';
run; quit;
```

The SAS datetime values are interpreted as Excel datetime values (Figure 17).

	A	B	C	D	E	F	G	H	I	J	K
1	Unique Subject Identifier	LAB Test or Examination Name	Age in AGEU at RFSTDC	Age Units	Sex	Baseline Result in Std. Units	Baseline Std. Units	Visit 5 Result in Std. Units	Visit 5 Std. Units	Baseline Collection Date/Time	Visit 5 Collection Date/Time
2	01-701-1034	CALCIUM	77 YEARS	F		2.5449	mmol/L	2.4451	mmol/L	6/24/2014 15:30	7/29/2014 8:40
3	01-701-1133	CALCIUM	81 YEARS	F		2.32035	mmol/L	2.2954	mmol/L	10/23/2012 10:57	11/26/2012 11:52
4	01-701-1146	CALCIUM	75 YEARS	F		2.32035	mmol/L	2.32035	mmol/L	5/7/2013 13:52	6/16/2013 12:45
5	01-701-1287	CALCIUM	56 YEARS	F		2.27045	mmol/L	2.0958	mmol/L	1/17/2014 12:16	2/20/2014 11:09
6	01-701-1383	CALCIUM	72 YEARS	F		2.22055	mmol/L	2.27045	mmol/L	1/22/2013 11:43	3/12/2013 11:50
7	01-703-1258	CALCIUM	78 YEARS	F		2.17065	mmol/L	2.2954	mmol/L	7/10/2012 16:30	8/17/2012 11:15
8	01-703-1295	CALCIUM	88 YEARS	F		2.32035	mmol/L	2.32035	mmol/L	10/24/2013 12:10	12/18/2013 12:45
9	01-703-1335	CALCIUM	67 YEARS	F		2.3453	mmol/L	2.32035	mmol/L	12/28/2013 16:10	4/13/2014 10:00

Figure 17. "LabResults.xlsx" File Created Using the EXPORT Procedure

Exporting SAS Data to Excel Files Using the XLSX LIBNAME Engine and the DATA Step

To create a multi-sheet Excel workbook, use the XLSX LIBNAME engine, the "LabResults_Subset" view [discussed earlier](#), and the DATA step.

```
options validvarname=any validmemname=extend;

libname xl xlsx 'LabResults.xlsx';

data xl.'Alanine Aminotransferase'n;
set work.LabResults_Subset;
where (lbtest eq 'ALANINE AMINOTRANSFERASE');
run;

data xl.Albumin;
set work.LabResults_Subset;
where (lbtest eq 'ALBUMIN');
run;
```

The resulting workbook contains two worksheets, "Alanine Transferase" and "Albumin". In these worksheets, the SAS datetime values are interpreted as Excel datetime values, and the SAS variable names, not the labels, are used for the Excel column headings.

Exporting SAS Data to Excel Files Using the XLSX LIBNAME Engine and the DATASETS Procedure

To export all or some of the SAS data sets in a library to an Excel workbook, use PROC DATASETS. In the exported file, the data for each data set is in its own worksheet, and the SAS data set name is used for the Excel worksheet name.

To specify which data sets are copied to the Excel workbook, use the DATASETS procedure with a SELECT statement.

```
libname xl xlsx 'SASHelp.xlsx';

proc datasets nolist;
  copy in=sashelp out=xl;
  select shoes class retail / memtype=data;
run; quit;
```

The resulting "SASHelp.xlsx" file has three worksheets: "SHOES", "CLASS", and

"RETAIL". SAS variable names, not labels, are used for the Excel column headings (Figure 18).

	A	B	C	D	E	F	G
1	Region	Product	Subsidiary	Stores	Sales	Inventory	Returns
2	Africa	Boot	Addis Ababa	12	29761	191821	769
3	Africa	Men's Casual	Addis Ababa	4	67242	118036	2284
4	Africa	Men's Dress	Addis Ababa	7	76793	136273	2433
5	Africa	Sandal	Addis Ababa	10	62819	204284	1861
6	Africa	Slipper	Addis Ababa	14	68641	279795	1771
7	Africa	Sport Shoe	Addis Ababa	4	1690	16634	79
8	Africa	Women's Casual	Addis Ababa	2	51541	98641	940
9	Africa	Women's Dress	Addis Ababa	12	108942	311017	3233
10	Africa	Boot	Algiers	21	21297	73737	710
11	Africa	Men's Casual	Algiers	4	63206	100982	2221
12	Africa	Men's Dress	Algiers	13	123743	428575	3621

Figure 18. "SASHelp.xlsx" File Created Using the DATASETS Procedure

CONCLUSION

SAS provides many techniques for working with Excel data. Except for SAS Enterprise Guide, all the Base SAS techniques for importing Excel data require that you first save the Excel data in a delimited text file. Only SAS Enterprise Guide and the ODS EXCEL destination can create XLSX files with Base SAS. The EXCEL destination exports any SAS procedure output to Excel, including graphic images. You can specify rich formatting attributes such as fonts, colors, and Excel number formats.

The SAS/ACCESS Interface to PC Files provides direct access to XLSX files without having to work with intermediary delimited files.

The VALIDVARNAME and VALIDMEMNAME options are often needed if you want to use Excel column headings and worksheet names as-is for SAS variable and data set names, respectively.

REFERENCES

DelGobbo, Vincent. 2018. "Related Papers". Available

<http://support.sas.com/rnd/papers/intro-multisheet-excel-with-sas/index.html>

SAS Institute Inc. 2017. Problem Note 61224. "In SAS® Enterprise Guide®, the Import Data task might fail to open Microsoft Excel 2016 (0365) files". Available

<http://support.sas.com/kb/61/224.html>

SAS Institute Inc. 2018a. "#BYVALn | #BYVAL(variable-name)". In *SAS® 9.4 Global Statements: Reference*. Cary, NC: SAS Institute Inc. Available

https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=lestmtsglobal&docsetTarget=p10gcmmf83iaxn1ilrx4pra969n.htm#n17bs500gdppsxn1rpsu0nlxr18e

SAS Institute Inc. 2018b. "E8601DTw.d Format". In *SAS® 9.4 Formats and Informats: Reference*. Cary, NC: SAS Institute Inc. Available

https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=leforinforref&docsetTarget=n03vr57gdozvean1klfaaa14c9c0.htm

SAS Institute Inc. 2018c. "EXPORT Procedure". In *Base SAS® 9.4 Procedures Guide, Seventh Edition*. Cary, NC: SAS Institute Inc. Available
https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=proc&docsetTarget=n045uxf7ll2p5on1ly4at3vpd47e.htm

SAS Institute Inc. 2018d. "FORMAT Statement". In *Base SAS® 9.4 Procedures Guide, Seventh Edition*. Cary, NC: SAS Institute Inc. Available
https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=proc&docsetTarget=n09nl7f8smat49n1rz1rav5i6a8a.htm#n09nl7f8smat49n1rz1rav5i6a8a

SAS Institute Inc. 2018e. "General Enhancements". In *SAS/ACCESS® 9.4 Interface to PC Files: Reference, Fourth Edition*. Cary, NC: SAS Institute Inc. Available
https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=acpcref&docsetTarget=acpcrefwhatsnew94.htm#p14uv4s8tynm1gn1417r5jdivocu

SAS Institute Inc. 2018f. "IMPORT Procedure". In *Base SAS® 9.4 Procedures Guide, Seventh Edition*. Cary, NC: SAS Institute Inc. Available
https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=proc&docsetTarget=n1qn5sclnu2l9dn1w61ifw8wqhts.htm

SAS Institute Inc. 2018g. "LIBNAME Statement Syntax for the XLSX Engine". In *SAS/ACCESS® 9.4 Interface to PC Files: Reference, Fourth Edition*. Cary, NC: SAS Institute Inc. Available
https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=acpcref&docsetTarget=n0oj9f6i838mymn148890ckla700.htm

SAS Institute Inc. 2018h. "MODIFY Statement". In *Base SAS® 9.4 Procedures Guide, Seventh Edition*. Cary, NC: SAS Institute Inc. Available
https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=proc&docsetTarget=n0ahh0eqtadmp3n1uwv55i2gyxiz.htm#n0ahh0eqtadmp3n1uwv55i2gyxiz

SAS Institute Inc. 2018i. "Rules for SAS Data Set Names, View Names, and Item Store Names". In *SAS® 9.4 Language Reference: Concepts, Sixth Edition*. Cary, NC: SAS Institute Inc. Available
<https://go.documentation.sas.com/?docsetId=lrcon&docsetTarget=p18cdcs4v5wd2dn1q0x296d3qek6.htm&docsetVersion=9.4#n0nfuxt1u6y9jin120qszztjh00o>

SAS Institute Inc. 2018j. "Rules for SAS Variable Names". In *SAS® 9.4 Language Reference: Concepts, Sixth Edition*. Cary, NC: SAS Institute Inc. Available
<https://go.documentation.sas.com/?docsetId=lrcon&docsetTarget=p18cdcs4v5wd2dn1q0x296d3qek6.htm&docsetVersion=9.4#p0w2dq3equv4b3n1m81ydfv0j7v0>

SAS Institute Inc. 2018k. *SAS/ACCESS® 9.4 Interface to PC Files: Reference, Fourth Edition*. Cary, NC: SAS Institute Inc. Available
https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=acpcref&docsetTarget=titlepage.htm

SAS Institute Inc. 2018l. "SAS Name Literals". In *SAS® 9.4 Language Reference: Concepts, Sixth Edition*. Cary, NC: SAS Institute Inc. Available
<https://go.documentation.sas.com/?docsetId=lrcon&docsetTarget=p18cdcs4v5wd2dn1q0x296d3qek6.htm&docsetVersion=9.4#n0qc68k2iyhs5kn1t2bvt41myhn1>

SAS Institute Inc. 2018m. "VALIDMEMNAME= System Option". In *SAS® 9.4 System Options: Reference, Fifth Edition*. Cary, NC: SAS Institute Inc. Available https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=lesysoptsref&docsetTarget=n10nwm6blrcrtmn0zdcwyxlwx fjh.htm

SAS Institute Inc. 2018n. "VALIDVARNAME= System Option". In *SAS® 9.4 System Options: Reference, Fifth Edition*. Cary, NC: SAS Institute Inc. Available https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=lesysoptsref&docsetTarget=p124dqd k8zoqu3n1r4nsfqu5vx52.htm

SAS Institute Inc. 2018o. "What Is the SAS Windowing Environment?". In *SAS® 9.4 Language Reference: Concepts, Sixth Edition*. Cary, NC: SAS Institute Inc. Available https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=lrcon&docsetTarget=p1g1z9eg910jban14ka4q6piemof.htm

ACKNOWLEDGMENTS

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

RECOMMENDED READING

SAS Institute Inc. 2018. "DATASETS Procedure". *Base SAS® 9.4 Procedures Guide, Seventh Edition*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=proc&docsetTarget=n1hmips60w5w3yn1hj9kl na7aplw.htm

SAS Institute Inc. 2018. "SQL Procedure". *SAS® 9.4 SQL Procedure User's Guide, Fourth Edition*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=sqlproc&docsetTarget=n1oihmdy7om5rmn1aorxui3kxizl.htm

CONTACT INFORMATION

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

Vincent DelGobbo
SAS Institute Inc.
sasvcd@sas.com

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

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

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

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.