Working with Excel spreadsheets using the SAS/ACCESS® LIBNAME statement
Gwen Babcock, New York State Department of Health, Albany, NY

ABSTRACT
SAS® programmers often receive data in a Microsoft Excel spreadsheet, and need to convert it into a data set. They are often requested to provide output in a Excel spreadsheet as well. This paper describes the basics of using the SAS/ACCESS LIBNAME statement to convert data from an Excel spreadsheet to a SAS data set, as well as to convert a SAS data set to an Excel spreadsheet. Examples are given using SAS 9.3 under Windows XP; very little prior knowledge of SAS is assumed. In addition, tips for solving several common problems are given.

INTRODUCTION
There are many methods that can be used in SAS to work with data in Excel spreadsheets: DDE, PROC IMPORT, PROC EXPORT, ODS, and others. However, using the SAS/ACCESS LIBNAME statement is often the easiest and most straightforward method to work with Excel data in SAS. Others have published papers showing how to make use of this method in advanced and detailed ways (Benjamin 2011; Choate and Martell, 2006). This paper will present only the basics of this method; it is intended for beginning SAS users. It gives examples of importing and exporting SAS data sets to Excel using SAS 9.3 running under Windows XP. To run this code, you must have SAS/ACCESS in addition to base SAS.

SO, SOMEONE SENT YOU A SPREADSHEET
You receive a spreadsheet. It might look like this:

But, you want to get the data into SAS. How does SAS see the spreadsheet? You can use SAS to import data by starting with a simple LIBNAME statement:

```
libname xdata "E:\temp\DataSentToYou.xlsx";
```

A typical LIBNAME statement refers to a directory, and two-level names are used to refer to individual data sets. For a spreadsheet, the LIBNAME refers to a single spreadsheet file, and two-level names are used to refer to individual worksheets or named ranges. Knowing that each worksheet (or named range) should be imported as a separate SAS data set, you might try something like this:

```
data new;
set xdata.hello;
run;
```

But the SAS log will say:

```
ERROR: File XDATA.hello.DATA does not exist.
```

NOTE: The SAS System stopped processing this step because of errors.

So how can you know what name to use to get data from a spreadsheet? You can use PROC CONTENTS:

```
proc contents data=xdata._all_;run;
```

Instead of naming a specific data set, we use the _all_ keyword. Then, in the results pane, you can see something like this:
This is a listing of all the worksheets and named ranges in the spreadsheet. Now we know what name to use to get the data we want.

```
data new;
set xdata.'Hello$'
run;
proc print data=new;run;
```

### YOU WANT TO PUT YOUR SAS DATA INTO A SPREADSHEET

Suppose you have a SAS data set called 'mydata' that you want to put into a spreadsheet.

```
data mydata;
input these are my data $;
datalines;
10 20 30 40
50 60 70 B
;
run;
```

You can assign a new library name:

```
libname xout "E:\temp\DataToGive.xlsx"
```

```
NOTE: Libref XOUT was successfully assigned as follows:
  Engine:         EXCEL
  Physical Name:  E:\temp\DataToGive.xlsx
```

```
data xout.mydata;
set mydata;
run;
```
There are two limitations to this technique. First, SAS will not create a new directory for you. If the directory that you want to put the spreadsheet in does not exist, you will get an error message. Second, SAS will not overwrite an existing worksheet within the spreadsheet.

OPEN YOUR SPREADSHEET: HOW TO MAKE SAS LET GO

If you are working with a spreadsheet in SAS, and then try to immediately open it in Excel, you may see a message like the following:

```
Microsoft Excel

10 libname xout clear;
```

This is because spreadsheets cannot be opened by more than application at a time, and the LIBNAME statement opens the spreadsheet in SAS. To avoid this conflict, use the CLEAR option with the LIBNAME statement:

```
libname xout clear;
in the log you will see:
10 libname xout clear;
NOTE: Libref XOUT has been deassigned.
```

Then you will be able to open the file in Excel.

EXCEL COLUMN HEADERS AND SAS VARIABLE NAMES

In the previous example, SAS used the Excel column headers as the SAS variable names. By default, SAS variable names must meet certain requirements, such as starting with a letter or underscore (SAS Institute Inc, 2013b), but Excel column headers can contain virtually any string. When the column headers are not valid SAS variable names, SAS will convert the names for you. Consider the following example:
As before, we can use a data step to import the data and look at it using PROC PRINT.

```sas
data newer;
set xdata.'World$'n;
run;
proc print data=newer;run;
```

<table>
<thead>
<tr>
<th>Obs</th>
<th>___times0</th>
<th>___times</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22</td>
<td>23</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>68</td>
<td>10</td>
</tr>
</tbody>
</table>

SAS converts the invalid characters to underscores, and appends a number to distinguish between variables with identical names. It will even assign a variable name to Excel columns without headers. However, the variable names SAS comes up with may now appear quite cryptic, unlike the original column headers. In order to distinguish between the two “___times” variables, you can run PROC CONTENTS and look at the list of variables and attributes:

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>___times</td>
<td>Num</td>
<td>8</td>
<td>:-) times</td>
</tr>
<tr>
<td>2</td>
<td>___times0</td>
<td>Num</td>
<td>8</td>
<td>:- ( times</td>
</tr>
</tbody>
</table>

You can see that the original Excel column headers are preserved in the variable labels. We can use the LABEL option with PROC PRINT to print out the data with the labels as column headers, and the LABEL statement to add a label to the "F3" variable:

```sas
proc print data=newer label; label F3="other times";run;
```

<table>
<thead>
<tr>
<th>Obs</th>
<th>:-) times</th>
<th>:- ( times</th>
<th>Other times</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22</td>
<td>23</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>68</td>
<td>10</td>
</tr>
</tbody>
</table>

Variable labels can also be used if you wish to export SAS data to Excel with column headers that are not SAS variable names. As an example, let us add a LABEL statement to the data step we previously used:
data mydata_labeled;
input this is my data $;
label this="Happy times :(-)" is="Sad times :-(";
datalines;
10 20 30 40
50 60 70 B
;
run;

Then, as before, we can use a LIBNAME statement with a DATA step to export the data. The only difference is that we are now placing the code "(DBLABEL=YES)" after the new data set name.

libname xout "E:\temp\DataToGive.xlsx";

data xout.mydata_labeled (dblabel=YES);
set mydata_labeled;
run;

libname xout clear;

This is what the spreadsheet looks like:

Since we used the DBLABEL option, the SAS labels are used as Excel column headers.

CONCLUSIONS

It can be simple to work with Microsoft Excel spreadsheets using the SAS/ACCESS LIBNAME statement. Once the LIBNAME statement has been run, you can use simple data steps to read and write data from the spreadsheet. PROC CONTENTS with _all_ can be used to get a listing of all the worksheets and named ranges in the spreadsheet. PROC CONTENTS can also be used to get the original Excel column headers, which is useful when SAS alters them to create valid SAS variable names. Conversely, the DBLABEL data set option can be used to export SAS data with column headers which are not valid SAS variable names. The CLEAR option can be used with the LIBNAME statement to release the spreadsheet so it is available to be opened in Microsoft Excel.

REFERENCES


ACKNOWLEDGMENTS

Thanks to Steve Forand, Thomas Talbot, and the New York State Department of Health, Center for Environmental Health for supporting this work.
CONTACT INFORMATION

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

Gwen D LaSelva
New York State Department of Health
Bureau of Environmental and Occupational Epidemiology
Empire State Plaza-Corning Tower, Room 1203
Albany, NY 12237
Work Phone: 518-402-7785
Fax: 518-402-7959
Email: gdb02@health.state.ny.us

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.