Paper FP_82

It's not the Yellow Brick Road but the SAS PC FILES SERVER® will take you Down the LIBNAME PATH= to Using the 64-Bit Excel Workbooks.

William E Benjamin Jr, Owl Computer Consultancy, LLC, Phoenix AZ.

ABSTRACT

SAS users who also use Excel or produce Excel workbooks will eventually find that the rapid pace of hardware and software changes occurring today will soon meet them head on. The need for faster computers and bigger workbooks are accelerating. This issue is not new, 8-Bit computers were replaced with 16-Bit CPU's, and they lost the battle to 32-Bit computers. Today new computer hardware routinely comes with up to eight 64-Bit CPUs in a single 2X2 piece of hardware for an affordable price. The fictional path down the yellow brick road to OZ was fraught with many challenges. SAS Institute has stepped up to the challenge of the way their users mix and match hardware and software. Since SAS Institute could not control the users and their hardware, the institute expanded the way programmers can use the LIBNAME statement to create a PATH= to open the way to create and read new Excel formats. Some new interfaces were built to facilitate transfers, one interface called the PCFILES server allows passing data to and from SAS and Excel across the boundary between 32 and 64-bit computers and software. The examples presented here will clear the fog and open the doorway past the curtain to view the processes available to open a PATH= to become your company's SAS to Excel wizard.

INTRODUCTION

Since the advent of personal computers using 8 bit hardware over 25 years ago many advances have occurred. Central Processing Units (CPU) have gotten more and more powerful, while the chips have remained about the same size and the number of computer circuits in each CPU has increased. Today the basic computer instruction uses 64 bits of information while the first Personal Computers only used 8. Additionally, the same 2 by 2 CPU chip now may hold 4 or 8 CPU's. We are currently transitioning from the 32 Bit computers and software to the new 64 bit computers operating around the world that transition may take some time for the software to catch up with the new hardware. The problem is that the software makers need to be able to provide their product without regard to what hardware the customers have. Or they will not have any customers. SAS Institute and Microsoft both must meet the needs of their customers.

THE PROBLEM

When your boss asks you for the latest analyses of your newest data file – in a spreadsheet, how do you get it there? Over the weekend the IT department just upgraded your computer to a new 64-bit laptop, with 64-bit SAS software and the old 32-bit Microsoft Excel package. (or the other way around) Now a five minute job seems to have no solution.

Even though Microsoft does not seem to care about how other software packages can access their files, SAS has put a lot of time and effort into helping their customers use SAS Software to access other file structures, in this case the much requested Excel spreadsheet. SAS specifically addresses the transfer of data from 32-bit to 64-bit files. An interface program called the SAS PC FILES SERVER® has been designed and implemented to serve this purpose. When SAS software is installed with the SAS PC FILES SERVER active and the Microsoft ACE (Access Database Engine) software installed this problem can be solved.

THE SOLUTION

As with just about anything relating to SAS software there are multiple was to accomplish a specific task. Here we will focus upon one solution. Note that the "Windows" operating system is not as important as the SAS software version, the Excel software, the SAS PC FILES SERVER version or the Microsoft ACE software. Here we will focus on what I have, and since I do not have unlimited resources The O/S may seem a little bit older than the most current versions but the other software working. I am using Windows Vista Ultimate for 64-bit hardware, with Base SAS version 9.3 TS1M0 for a Windows 64-bit system, and Microsoft Excel 2010 (Version 14 – 32-bit). Even though the title says 64-bit Excel The key to making the processing work is to make sure that all of the products are reading and writing to the expected "Bit Alignment" a SAS PC FILES SERVER that runs in 64-bit mode won't run on a 32-bit computer. Contact SAS Technical Support to resolve any issues.

64-bit compatible components

My Hardware My Operating System My SAS Software My SAS PC FILES SERVER Microsoft ACE software <u>32-bit components</u> Excel 2010

Now the SAS LIBNAME statement PATH= option comes to my rescue.

First let us look at the different syntax options and parameters available for the LIBNAME statement on various operating systems as they relate to accessing Microsoft Excel files. These options are available for versions 9.0 and forward unless otherwise noted. Some general LIBNAME options are discussed, but the main point of the following tables are to point out LIBNAME options specifically designed for use with Microsoft Excel files.

For SAS/Access Interface to PC Files

```
LIBNAME libref <engine> <physical-file-name>
<SAS/ACCESS-engine-connection-options>
<SAS/ACCESS-libname-options>;
```

	LIBNAME Syntax Descriptions
Argument/Alias	Description
Libref	A user assigned valid SAS name one to eight characters long
ENGINE	An optional name of a SAS method of accessing a data file. See the documentation for your SAS version to determine if you can access the newer versions of Excel. Microsoft Excel engines include 5, 95, 97, 2000, 2002, 2003, 2007, 2010. There is no engine for Excel4. SAS Version 9.2 TS Level 2M3 and above allows access to the Microsoft Office 2007 and 2010 products Excel 2007 and Excel 2010.
physical-file-name	This is the path and file name of a Microsoft Excel file.
SAS/ACCESS-engine- connection-options	See additional resources listed below.
SAS/ACCESS-libname- options	See additional resources listed below.

Figure 1. LIBNAME Syntax Descriptions.

	ENGINE-CONNECTION-OPTIONS					
Argument/Alias	Description					
HEADER= / GETNAMES= or HDR=	Choose to read the first input row from Excel as either variable names or data.					
INIT= / INIT_STRING=	Provide an access provider to connect to a data source, see the SAS documentation for further details.					
MIXED= / MIXED_DATA= or MIXED_DATATYPE	Choose to allow Excel to determine if data columns contain numeric or character data or cause all data to be read as character data.					
PATH= / DATASRC= and DS=.	Identifies the full path and file name of an external Excel file					
PROMPT=	Allows for an interactive assignment of a data source.					
VERSION= / VER=	This option is used when writing Excel files. The default is to write Excel version 97 (Excel 97 to Excel 2003 format					

Figure 2. LIBNAME Engine Connection Option Descriptions.

PATH OPTION

This option allows for the entry of a file name and the full path to access the file. While the same information is entered by using a "<physical-file-name>", this option requires the engine name to be included on LIBNAME command.

LIBNAME myxls PCFILES SERVER=xxxxx PORT=8621 PATH="c:\demo.xls"; * default port V9.2;

LIBNAME myxls PCFILES SERVER=xxxxx PORT=9621 PATH="c:\demo.xls"; * default port V9.3+;

Note that the "PORT=" option is optional when the SAS PC FILES SERVER is installed using the default port. And the "SERVER=" option is not required if the SAS and Excel software are running on the same computer.

EXAMPLES

The rest of the paper is showing examples of methods available to write and read data to and from Excel file. The examples will all use the PATH= option. Some will use a SAS LIBNAME statement, others will use PROC SQL and show you how to access Excel files (read and write) using PROC SQL.

EXAMPLE 1 - WRITING DATA TO A 32-BIT EXCEL FILE FROM 64-BIT SAS

The images on the next page Show a LIBNAME Statement in Figure 1 along with the SAS Code to write a simple dataset to the output Excel file. This code creates an Excel workbook, and expects the SHEET name to ne new. This means that you can write a very large number of new sheets. Be careful because the Microsoft ACE engine may have limitations.

ſ	😽 SAS - [Editor - Untitled	1*]	1	Aure .	The Province of the	
	📃 File Edit View T	ools Run Soluti	ons Window Help			- 8 ×
	~		- 🗅 🛩 🖬 🎒	🛕 🐰 🖻 🛍 🖍	🐌 🔍 🛪 X 🛈 🛷	
	Explorer 🛛	LIBNAME >	ls_data PCFILES p	path='C:\My_excel_i	files\my_excel_shoes_file	.xls'; 🔺
1	Contents of 'SAS Environm	DATA set s	data.'sheet1': ashelp.shoes;	n;		
	Libraries	run;				
	File Shortcuts					
1	e					-
L	Favorite					• •
	Resu Explorer	🖪 Output - (Ur	ititled) 📄 Log - (Untitled)	tor - Untitled1 *	
	C:\Users\owner				L	n 1, Col 65

Figure 3. SAS Code to write an Excel file using the SAS PC FILES SERVER.

You may get error messages, but most of the time these are related to the path not existing or the file/sheet existing. Neither of these conditions work well.



Figure 4. SAS LOG output to write an Excel file using the SAS PC FILES SERVER.

EXAMPLE 2 - READING DATA FROM A 32-BIT EXCEL FILE INTO 64-BIT SAS

The next example shows reading the Excel file just created in example 1.

🐺 SAS - [Editor - Untitle	ed1*]	
File Edit View	Tools Run Solutions Window Help	_ 8 ×
	🔻 🗋 🖆 🖬 🏻 🎒 🔃 🐇 🛍 🛍 🗠 👘 🔃 * 🗙 🕐 🥔	
Explorer 🛛	LIBNAME xls_data PCFILES path='C:\My_excel_files\my_excel_shoes_file	e.xls'; 🔺
Contents of 'SAS Environm	DATA my_shoes; set xls data.'sheet1'n;	
Libraries	run;	
File Shortcuts		
Favorite	∭ < ™	•
Resu Q Explorer	🔛 Output - (Untitled)	
C:\Users\owner		Ln 4, Col 28

Figure 5. SAS Code to read an Excel file using the SAS PC FILES SERVER.

ſ	😽 SAS - [Log - (Untitled)	
	🔳 File Edit View	Tools Solutions Window Help
	✓	🗸 🗋 🚔 🖬 🖨 🐧 🐰 🛍 🛍 🗠 🛛 🐌 🔀 🍳 🖈 🛈 🛷
	Explorer 🛛	23 LIBNAME xls_data PCFILES path='C:\My_excel_files\my_excel_shoes_file.xls'; NOTE: Libref XLS_DATA was successfully assigned as follows:
ł		Engine: PCFILES Physical Name: C:\My_excel_files\my_excel_shoes_file.xls 24
1	Libraries	25 DATA my_shoes; 26 set xls_data.'sheet1'n; 27 run;
-	File Shortcuts	NOTE: There were 395 observations read from the data set XLS_DATA.sheet1. NOTE: The data set WORK.MY_SHOES has 395 observations and 7 variables. NOTE: DATA statement used (Total process time): real time 0.42 seconds cpu time 0.00 seconds
		-
ľ	Favorite	۲
	Folders 👻	Output - (Untitled)
	C:\Users\owner	

Figure 6. SAS Log output for reading an Excel file using the SAS PC FILES SERVER.

😽 SAS - [VIEWTABLE: Wo	ork.My_sho	oes]	regular a land	-	range	roaned"m	L'anga'ind		
🔳 File Edit View	Tools Da	ta Solution	ns Window He	lp					- 8 X
~		•	🗋 🖻 🔳 (🗐 🗋 🖹 h	ĮΩ κγ)	× ↓ª ↓ẩ [)= 13 13 13	🗵 🤣	
Explorer 🛛		Region	Product	Subsidiary	Stores	Sales	Inventory	Returns	*
Contents of 'Work'	1	Africa	Boot	Addis Ababa	12	\$29,761.00	\$191,821.00	\$769.00	
	2	Africa	Men's Casual	Addis Ababa	4	\$67,242.00	\$118,036.00	\$2,284.00	
	3	Africa	Men's Dress	Addis Ababa	7	\$76,793.00	\$136,273.00	\$2,433.00	
My shoes	4	Africa	Sandal	Addis Ababa	10	\$62,819.00	\$204,284.00	\$1,861.00	
my_snocs	5	Africa	Slipper	Addis Ababa	14	\$68,641.00	\$279,795.00	\$1,771.00	
	6	Africa	Sport Shoe	Addis Ababa	4	\$1,690.00	\$16,634.00	\$79.00	
	7	Africa	Women's Casual	Addis Ababa	2	\$51,541.00	\$98,641.00	\$940.00	
	8	Africa	Women's Dress	Addis Ababa	12	\$108,942.00	\$311,017.00	\$3,233.00	
	9	Africa	Boot	Algiers	21	\$21,297.00	\$73,737.00	\$710.00	
	10	Africa	Men's Casual	Algiers	4	\$63,206.00	\$100,982.00	\$2,221.00	
	11	Africa	Men's Dress	Algiers	13	\$123,743.00	\$428,575.00	\$3,621.00	
	12	Africa	Sandal	Algiers	25	\$29,198.00	\$84,447.00	\$1,530.00	-
									•
Resu 🔍 Explorer	🔡 Ou	tput - (Untit	led)	g - (Untitled)		Editor - Untitled1	* VIEV	VTABLE: Work.	Му
C:\Users\owner				_	_				÷.

Figure 7. The contents of the SAS dataset created by reading the Excel worksheet.

EXAMPLE 3 – READING DATA TO A 32-BIT EXCEL BINARY WORKBOOK (XLSB)

😪 SAS - [Editor - Untitled1 *]	
🗊 File Edit View Tools Run Solutions Window Help	_ 8 ×
 ・ ・	
Results LIBNAME out_data PCFILES PATH="C:\My_Excel_Files\Excel_Shoes.xlsb";	*
Results DATA out_data."Shoes"n; SET SASHELP.Shoes; RUN;	
Besults QL Expl. Dutput - (Untitled) E Log - (Untitled) Editor - Untitled1 *	*
C:\Users\owner	Ln 6, Col 1

Figure 8. SAS Code to write an Excel binary file using the SAS PC FILES SERVER.



Figure 9. SAS Log output for writing an Excel binary file using the SAS PC FILES SERVER. (XLSB)

Notice that when we create a copy of the workbook and then rename the file with the extension .ZIP we can open the "ZIP" file and view the contents. Most of these files are xml text files but some are in fact binary files. These files are stored in a compressed zip file with the label .xlsb.

With a second secon	~			A Recent for		_ D X
File Edit View Tools Help Organize Image: Second state of the second state	$\bigcirc \bigcirc$	MY_Excel_Files Excel_Shoes - Co	opy.zip ▶ xl ▶	✓ Search		م
Organize Iteract all files Favorite Name Type Compressed size Password Size Favorite Name Type Compressed size Password Size Favorite Name File Folder File Folder File Folder File Folder D worksheets File Folder Item File Folder Item File Folder D styles.bin BIN File Item Item Item Item	File Edi	it View Tools Help				
Favorite Name Type Compressed size Password Size Image: Password Image: Passwo	🎍 Organ	nize 🔻 🏢 Views 👻 🚺 Extract all files				0
Image: Second state of the second s	Favorite I	Name	Туре	Compressed size	Password	Size
Image: Draw of the me File Folder Image: Draw of the me File Folder Image: Provide the me File Folder Image: Provide the me BIN File Image: Draw of the me BIN File	🗱 D	🍌 _rels	File Folder			
P Distyles.bin BIN File 1 KB No 1 kB Max Distyles.bin BIN File 1 KB No 1 kB	D	b theme	File Folder			
I I I I I I I I I I I I I I I I I I I		📕 worksheets	File Folder			
M. Warthook hin BIN File 1 KB No 1 k	F	styles.bin	BIN File	1 KB	No	1 K
	🚺 M	workbook.bin	BIN File	1 KB	No	1 K
M	M.,					
Folders (Folders	•	III			Þ

Figure 10. Windows directory listing of the contents of a xlsb file.

EXAMPLE 4 – USING PROC SQL TO READ DATA FROM EXCEL

This example uses PROC SQL to read data from an Excel file. This example just prints the data onto the output window, but with a SQL CREATE command it could just as easily write the data to a SAS dataset.

😽 SAS - [Editor - Untitled	11*] Han han han han han han		23	J
🔳 File Edit View T	Fools Run Solutions Window Help		8 ×	
~	- 🔽 🕞 🖬 🎒 🔃 🐰 🖿 🛍 🗠 👘 🕲 🖈 🗙 🕐 🥔			
Results	<pre>PROC SQL; CONNECT TO pcfiles AS db (PATH="C:\My_Excel_Files\my_excel_shoe SELECT * FROM CONNECTION TO db (SELECT * FROM "SHEET1"n); DISCONNECT FROM db; QUIT;</pre>	s_file.xls");	
	< III		-	
Results Expl	Dutput - (Untitled)			
C:\Users\owner		Ln 7, Col 1	d	4

Figure 11. SAS Code to read Excel data using PROC SQL.

¥g SAS - [Log - (Untitled)]
I File Edit View Tools Solutions Window Help
✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ Ø
Results 34 PROC SQL; 35 CONNECT TO pcfiles AS db (PATH="C:\My_Excel_Files\my_excel_shoes_file.xls") 36 SELECT * FROM CONNECTION TO db 37 (SELECT * FROM "SHEETI"n); 38 DISCONNECT FROM db; 39 QUIT; NOTE: PROCEDURE SQL used (Total process time): real time 0.15 seconds cpu time 0.03 seconds
Results Q Expl Dutput - (Untitled)
C:\Users\owner

Figure 12. SAS LOG showing the result of reading Excel data using PROC SQL.

😽 SAS - [Output - (Untit	tled)]			the list time it	and Made	the first how of details hard to be	
🔳 File Edit View	Tools Solution	s Window Help					_ 8 ×
~		- 🖌 🖆 🖬 🖨 🛛) 🐰 🖿 💼 🗠 🗠	📄 🔁 🚉 🖈 🔇) 🛷		
Results 🛛				The	SAS System	21:25 Monday, Septemb	er 23, 2013 101 🔺
Results	Region	Product	Subsidiary	Stores	Sales	Inventory	Returns
🗄 🚠 SQL: The SAS	Africa	Boot	Addis Ababa	12	\$29,761.00	\$191,821.00	\$769.00
	Africa	Men's Casual	Addis Ababa	4	\$67,242.00	\$118,036.00	\$2,284.00
	Africa	Men's Dress	Addis Ababa		\$76,793.00	\$136,273.00	\$2,433.00
	Africa	Sanda I	Addis Ababa	10	\$62,819.00 \$60,641,60	\$204,284.00 \$070,705,00	\$1,861.00
	Africa	Slipper	HOOIS HDADA	14	468,641.00 \$1,000.00	#279,795.00 \$10 C94 AA	al,771.00
	Africa	Homop's Court	Addio Ababa	4	\$E1 E41 00	\$90 641 00	\$940.00
	Africa	Momen's Dress	Addis Ababa	12	\$108 942 00	\$311 017 00	\$3 233 00
	Africa	Boot	Aloiers	21	\$21 297 00	\$73 737 00	\$710.00
	Africa	Men's Casual	Aloiers	4	\$63,206,00	\$100,982,00	\$2 221 00
	Africa	Men's Dress	Aloiers	13	\$123,743.00	\$428.575.00	\$3.621.00
	Africa	Sanda l	Aloiers	25	\$29,198.00	\$84,447,00	\$1,530.00
	Africa	Glinner	Aloiere	17	\$64 991 00	\$949 199 00	\$1 923 00
	•			III			+
I → III							
Results 🔍 Expl	🕒 Output -	(Untitled) 🗄 Log - (Unt	itled)	ditor - Untitled1 *			
C:\Users\owner							

Figure 13. The Output Listing generated reading Excel data using PROC SQL.

EXAMPLE 5 – READING DATA FROM AN EXCEL NAMED RANGE INTO SAS

This example shows how to read data from the middle of an Excel Workbook Spreadsheet. When SAS uses a LIBNAME statement to access an Excel workbook it assigns two names to each worksheet. One name is assigned as a "Range" and the other as a "Sheet\$". A range name does not have a "\$" at the end of the name, while the sheet name does have a "\$". When SAS references the data in the Excel workbook a sheet name needs to be referenced as a SAS "Named-Constant" as follows: SET libname.'SHEET1\$'n; But, a range name can be referenced as SET libname.range; (without the ' 'n notation).

The screen shot below is of an example of a Named-Range in the middle of an Excel spreadsheet. Many people would first try using DDE to capture data in the middle of a spreadsheet, but if the data can be assigned a named range by Excel than SAS can read the data directly.

	J 9 - 1	(≈ - -	E	xcel_with_na	med_range	xlsx - Micro	osoft Excel				×
Fi	le Hor	me Insert	Page Layou	t Formulas	Data F	leview Vie	w Develop	er JMP	sas 🗠 🕜	- 6	23
J Ins Fund	$\begin{array}{c} x \\ x \\ \end{array} \begin{array}{c} \Sigma \\ \hline B \\ rent \\ ction \end{array} \begin{array}{c} \Sigma \\ B \\ \hline B \\ Fi \end{array}$	utoSum 👻 ecently Used inancial 👻 Function	 Eogica Text Text Date 8 Library 	l ▼	Name Manager	[™] Define <i>f</i> [™] Use in f [™] Create f Defined Na	Name 👻 Formula 👻 from Selection imes	Formula Auditing	Calculation	n	
M	y_named	_range 🕞	- (-	∫∞ O							~
	А	в	С	D	E	F	G	н	1	J	
1											
2											
3											
4											_
5											_
6									1		
-					viy data 1	Ny data 2	iviy data 3	iviy data 4	iviy data 5		-
<u> </u>					2	2	3	4	5		
10						4	-+	5	7		
11					4	5	6	7	8		
12					5	6	7	8	9		
13					6	7	8	9	0		
14									Ē		
15											
16											-
14 4	► ► Sh	eet1 / She	eet2 / Shee	et3 <u>2</u>		1					<u> </u>
Rea	dy 🔛		Average:	5.166666667	Count: 35	Sum: 155		100% (-	,	(±	1

Figure 14. An example of an Excel named range.

The SAS code in Figure 15 shows how to read the Excel named range.

Image: SAS - [01_27_2013_code.sas *]	ſ
🗈 File Edit View Tools Run Solutions Window Help	1
✓ ✓ ✓ ✓ ✓ ✓ ✓ Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø	
Results 🛛	L
Results	L
<pre> data test; SET xlRange.my_named_range; run; </pre>	
Results Log - (Untitled)	
C:\Users\owner Ln 69, Col 1	

Figure 15. SAS code used to read a named range from Excel, NOTE there are no quotes on the SET statement.

🥰 SAS - [Log - (Untitled)]
E File Edit View Tools Solutions Window Help
✓ ▼ ■ × □ ≡ ≒ ₩ Q Ø
Results 42 43 LIBNAME x1Range pcfiles path='C:\My_excel_files\Excel_with_named_range.x1sx'; NOTE: Libref XLRANGE was successfully assigned as follows: Engine: PCFILES Physical Name: C:\My_excel_files\Excel_with_named_range.x1sx'; 44 45 data test; 46 SET x1Range.my_named_range; 47 run; NOTE: There were 6 observations read from the data set XLRANGE.my_named_range. NOTE: DATA statement used (Total process time): real time 0.07 seconds cpu time 0.01 seconds
Results 🖳 Expl 🖾 Output - (Untitled) 🖺 Log - (Untitled) 🖉 JOB_01_Write_data_to_E 🔣 01_27_2013_code.sas *
C:\Users\owner

Figure 16. SAS Log showing the results of reading a named range from Excel.

😽 SAS - [VIEWTABLE: W	ork.Test]						x
E File Edit View	Tools Data	Solutions	Window	Help		-	Ξ×
~	_	-	🗋 🖻	i 🕹 🗋		⊠ ⊳ X	↓a ↓
Explorer 🛛 🖾		My data 1	My data 2	My data 3	My data 4	My data 5	
Contents of 'Work'	1	1	2	3	4	5	
	2	2	3	4	5	6	
	3	3	4	5	6	7	=
Test	4	4	5	6	7	8	
i cor	5	5	6	7	8	9	
	6	6	7	8	9	0	
							-
	•						- F
		nt 🗐 Le	a-()		2 01 27 2		NT I
Resu 🖳 Explorer			a.(1.00_01			
C:\Users\owner							

Figure 17. SAS View showing the results of reading a named range from Excel.

Figure 17 shows the results of reading the named range from Excel, the variable labels are shown as evidenced by the spaces visible as "My data 1", "My data 2" etc. The actual variable names have underscores instead of spaces as "My_data_1" etc.

EXAMPLE 6 – SHOWING OUTPUT FILE STRUCTURES

This example is a little different from the other examples. It does the same work, that of writing data to an Excel Workbook. But, the object of this example is to shoe that the same SAS DATA STEP will write out Different Excel WORKBOOK formats. We want to write out a Version 97-2003 Excel workbook, a Version 2007/10 formatted Excel (xlsx) Workbook, and a Version 2007/10 formatted Excel Workbook.

😽 SAS - [Editor - Untitled	it *) the defails read the defails read the	
🔳 File Edit View T	ools Run Solutions Window Help	_ 8 ×
✓		
Results Results	LIBNAME x1s_V97 PCFILES PATH='C:\My_excel_files\my_excel_file1_V97.x1s'; LIBNAME x1s_V07x PCFILES PATH='C:\My_excel_files\my_excel_file1_V07.x1sx'; LIBNAME x1s_V07b PCFILES PATH='C:\My_excel_files\my_excel_file1_V07.x1sb'; DATA x1s_V97.'sheet1'n	~
🗗 Results 🔍 Expl	Image: A constraint of the second	•
C:\Users\owner	L	n 3, Col 64

Figure 18. SAS Code to write three Excel files of different formats.

😨 SAS - [Log - (Untitled)]	23
I File Edit View Tools Solutions Window Help	Ξ×
Results Image: Classing and the second of the second o	
	-
Results 🖳 Expl Output - (Untitled)	
C:\Users\owner	

Figure 19. SAS LOG showing the successful completion of the job step.

Microsoft Excel	X
Excel cannot open the and that the file exten	ile 'my_excel_file1_V07.xlsx' because the file format or file extension is not valid. Verify that the file has not been corrupted ion matches the format of the file.
	OK

Figure 20. Windows error message displayed when trying to open the xlsx file.

We need to look at this example a little closer, since Excel does not thing this file is acceptable. First let's create a new file directly with Excel 2010. (file EXCEL_2012_Created_Directl.... Below).



Figure 21. Directly listing of the three files created with SAS and the new Workbook.

One thing nice about Windows icons is that Microsoft goes to great pains to (by default at lease) show different file formats with different icons. In Figure 19 I have displayed the file "Tiles" for each of the four file we just created. Notice that the V97.xls file and the one we created directly are different. The V97 file bears the old Excel icon while the manually created file has the new 2007/2010 icon. Also note that the V07 files both have the same icon. Since we can open the xlsb file (after issuing a LIBNAME statement to clear the LIBNAME references) using Excel that might be a hint about how to fix the problem. If we rename the *.xlsx file to a name that ends in *.xlsb the file should open (if the LIBNAME has been cleared)

CONCLUSION

The PATH= option when accessing an Excel Workbook using the SAS PC FILES SERVER is a convenient way to open EXCEL Workbooks for either input or output. Unfortunately you need to have the SAS product SAS/Access Interface to PC Files[®]. But, as with many other products maybe someday you will not need a separate product.

ADDITIONAL RESOURCES

SAS/ACCESS® 9.4 Interface to PC Files Reference

http://support.sas.com/documentation/cdl/en/acpcref/64792/PDF/default/acpcref.pdf

SAS/ACCESS[®] 9.3 Interface to PC Files Reference

http://support.sas.com/documentation/cdl/en/acpcref/63181/PDF/default/acpcref.pdf

SAS/ACCESS [®] 9.2 Interface to PC Files Reference Second Edition

http://support.sas.com/documentation/cdl/en/acpcref/63184/PDF/default/acpcref.pdf

SAS/ACCESS [®] 9.1 Interface to PC Files Reference

http://support.sas.com/documentation/onlinedoc/91pdf/index.html

CONTACT INFORMATION

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

Name: William E Benjamin Jr Enterprise: Owl Computer Consultancy, LLC Address: P.O.Box 42434 City, State ZIP: Phoenix AZ, 85080 Work Phone: 623-337-0269 E-mail: William@owlcomputerconsultancy.com

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.