Paper 10863-2016

Migrating from SAS® 9.3 to SAS® 9.4: SAS communicating with MS Office Products

Hong Zhang, Mathematica Policy Research

ABSTRACT

Microsoft Office products play an important role in most enterprises. SAS is combined with Microsoft Office to assist in decision making in everyday life. Most SAS users have moved from SAS 9.3, or 32-bit SAS, to SAS 9.4 for its exciting new features. This paper describes a few things that do not work quite the same in SAS 9.4 as they did in the 32-bit version. It discusses the reasons for the differences and the new approaches that SAS 9.4 provides. The paper focuses on how SAS 9.4 works with Excel and Access. Furthermore, this presentation summarizes methods by LIBNAME engines and by import or export procedures. It also demonstrates how to interactively import or export PC files with SAS 9.4. The issue of incompatible format catalogs is addressed as well.

INTRODUCTION

We use MS Office files often in our daily lives. When SAS moves to major release 9.4, errors could occur with the legacy codes connecting SAS to Excel or Access. One of the fundamental differences between SAS 9.4 and previous releases is the bitness. SAS 9.4 is 64-bit whereas prior ones were 32-bit. Therefore, the way to connect 32-bit MS Office products is changed in SAS 9.4 by new engines: the PCFILES LIBNAME engine with SAS PC Files Server and the XLSX LIBNAME engine. This paper describes the new approaches in how SAS communicates with Excel and Access when SAS and MS Office are not bit compatible. It summarizes all of the methods of connection by libname statement and import/export procedures with the corresponding options. This paper also addresses incompatible format catalogs between SAS 9.4 and 32-bit SAS. Two methods can be used to solve this issue.

CONNECTING SAS AND EXCEL/ACCESS

Before moving to SAS 9.4, the EXCEL and ACEESS libname engines could have been used to communicate with Excel and Access respectively, as shown here:

```
SAS EXCEL engine for Excel: LIBNAME xls ‘mydrive:\demo\demo.xlsx’;
SAS ACCESS engine for Access: LIBNAME adb 'mydrive:\demo\demo.accdb';
```

The above codes work when SAS and MS Office bitness match; this includes scenarios when both SAS and MS Office are 32-bit or 64-bit.

When SAS is upgraded to 9.4 but MS Office stays in 32-bit, errors will occur by submitting the same codes:

ERROR: Connect: Class not registered

ERROR: Error in the LIBNAME statement

THE CAUSE OF THE ERRORS

The way SAS talks to MS Office in the above codes is through MS ACE Driver. MS ACE Driver comes with MS Office products and always has the same bitness as MS Office. MS Office 2007 and prior versions are all 32-bit. Starting with Office 2010, the products have 32-bit and 64-bit options.

If SAS and MS Office ACE Driver do not match in bitness, the above codes will no longer work.

THE NEW APPROACHES IN HOW 9.4 COMMUNICATES WITH MS EXCEL AND ACCESS

To work with 32-bit Excel/Access, two new approaches are available in SAS 9.4.

SAS PC FILES SERVER

SAS PC Files Server is an application that receives client requests to access data files from MS Office such as Excel and Access. It communicates with the SAS libname engine, PCFILES, which resides in SAS/ACCESS interface to PC files. SAS PC Files Server can run as either 32-bit or 64-bit application. It must match the bit architecture of Microsoft Office and the Microsoft Access Database Engine Driver. The three are “bit-compatible.” SAS PC Files Server is the bridge of bitness gap between SAS and MS Office.

Here are the examples of codes using SAS PC File Server with PCFILES libname engine:

For communicating with Excel:

LIBNAME statement:

```
      LIBNAME xls PCFILES path=‘mydrive:\proc_demo\demo.xlsx’;
```

Or

PROC IMPORT / EXPORT:

```
      PROC IMPORT DATAFILE="mydrive:\proc_demo\demo.xlsx"

      DBMS=EXCELCS OUT=demo REPLACE;

      SHEET=‘state';

      RUN;
```

When importing data, the file extension varies, depending on the type of Excel file (XLS, XLSX, XLSB, or XLSM).

```
      PROC EXPORT DATA=SASINVOICES

      OUTFILE='mydrive:\PROC_demo\demo2.xlsb'

      DBMS=EXCELCS REPLACE;

      SHEET='YourInvoices';

      RUN;
```

When you create a new Microsoft Excel file with Proc Export and DBMS = EXCELCS, use an .xlsb or .xls extension. If you name the new file with an .xlsx extension, no errors occur. However, the file is still in the .xlsb format. You cannot open the file using the Excel application until you rename it with an .xlsb extension.

For communicating with Access:

LIBNAME statement:

```
      LIBNAME abc PCFILES path='mydrive:\demo\demo.accdb';
```

or

Proc IMPORT / EXPORT:

```
      PROC IMPORT DBMS=ACCESSCS DATATABLE='Customers'
      OUT=WORK.SASCUSTOMERS REPLACE;
      DATABASE='mydrive:\yourAccessdirectory\AnnualFiles.accdb';
      RUN;
      PROC EXPORT DATA=SASCUSTOMERS
      OUTTABLE='Customers_by_Zip'
      DBMS=ACCESSCS REPLACE;
      DATABASE='mydrive:\demo\AnnualFiles.accdb';
      RUN;
```

SAS XLSX engine for Excel

Starting with the second maintenance release, SAS/ACCESS 9.4 Interface to PC Files has added a new SAS LIBNAME engine called XLSX. It enables you to directly read and write data between Microsoft Excel XLSX files and SAS. It is the application that uses File Format Technology to communicate with Microsoft Excel 2007, 2010, and later files.

LIBNAME statement:

```
     LIBNAME abc xlsx ' mydrive:\demo\demo.xlsx';
```

Proc IMPORT/EXPORT:

```
     PROC IMPORT DATAFILE="mydrive:\demo\demo.xlsx"
     DBMS=XLSX  OUT=WORK.SASINVOICES REPLACE;
     SHEET='Invoice';
     RUN;

     PROC EXPORT DATA=SASINVOICES
     OUTFILE='mydrive:\PROC_demo\demo2.xlsx'
     DBMS=XLSX REPLACE;
     SHEET='YourInvoices';
     RUN;
```

The SAS XLSX engine has an advantage in that it is not limited to 255 columns in Excel, so it can export more than 255 columns to the 2007 Excel release or later and can import Excel files with more than 255 columns.

SUMMARY OF POSSIBLE APPROACHES

Because there are a variety of ways for SAS to communicate with Excel/Access, it is necessary to summarize the methods by proc import/export procedure and by libname engines.

SAS connects MS Office files by proc import/export:

SAS to Excel:

DBMS Specifications For Excel	Uses	Requires	Operating Platform
EXCEL	SAS Excel LIBNAME engine	Microsoft ACE or Jet Provider	Microsoft Windows
XLS	File formats technology		Microsoft Windows, UNIX, Linux
XLSX	File formats technology		Microsoft Windows, UNIX, Linux
EXCELCS	PCFiles LIBNAME engine	SAS PC Files Server Excel Driver on Windows	Microsoft Windows

Display 1: List of DBMS specifications for Excel in proc import/export

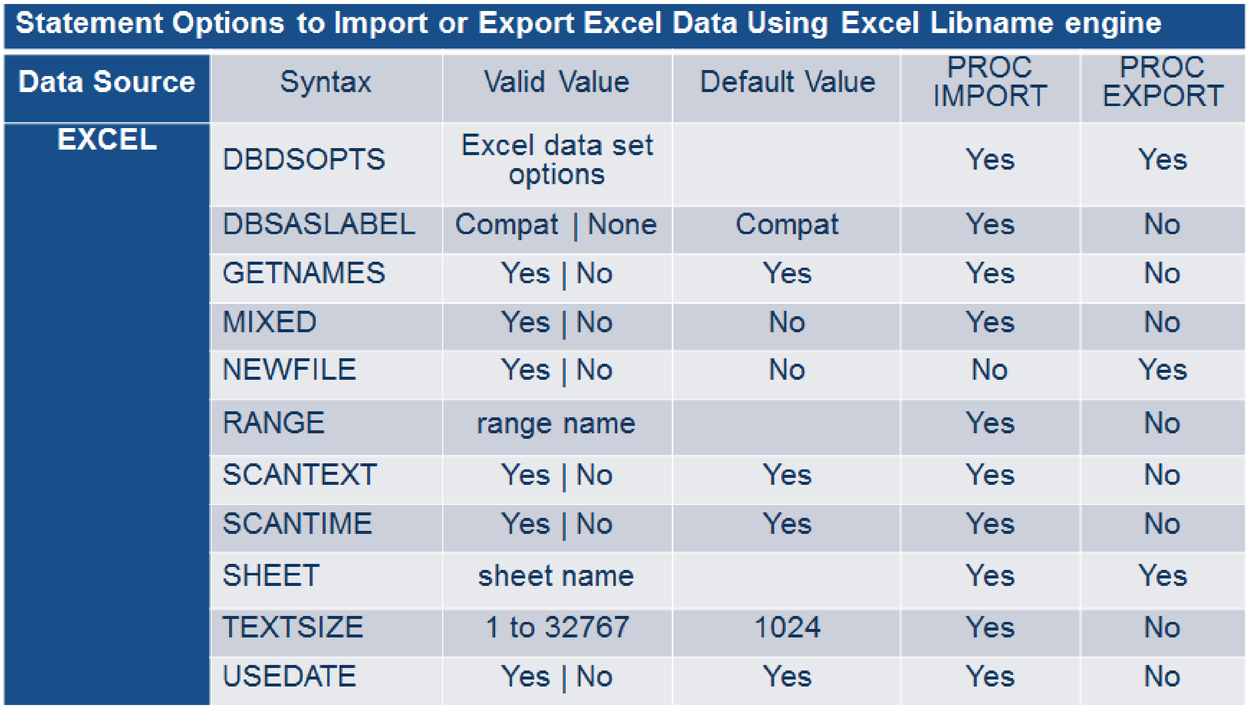
SAS to Access:

DBMS Specifications For Access	Uses	Requires	Operating Platform
ACCESS	SAS Access LIBNAME engine	Microsoft ACE or Jet Provider	Microsoft Windows
ACCESSCS	PCFiles LIBNAME engine	SAS PC Files Server Access Driver on Windows	Microsoft Windows

Display 2: List of DBMS specifications for Access in proc import/export

Please note that the statement options in proc import/export procedures are not unique. They are different in each engine or technology used.

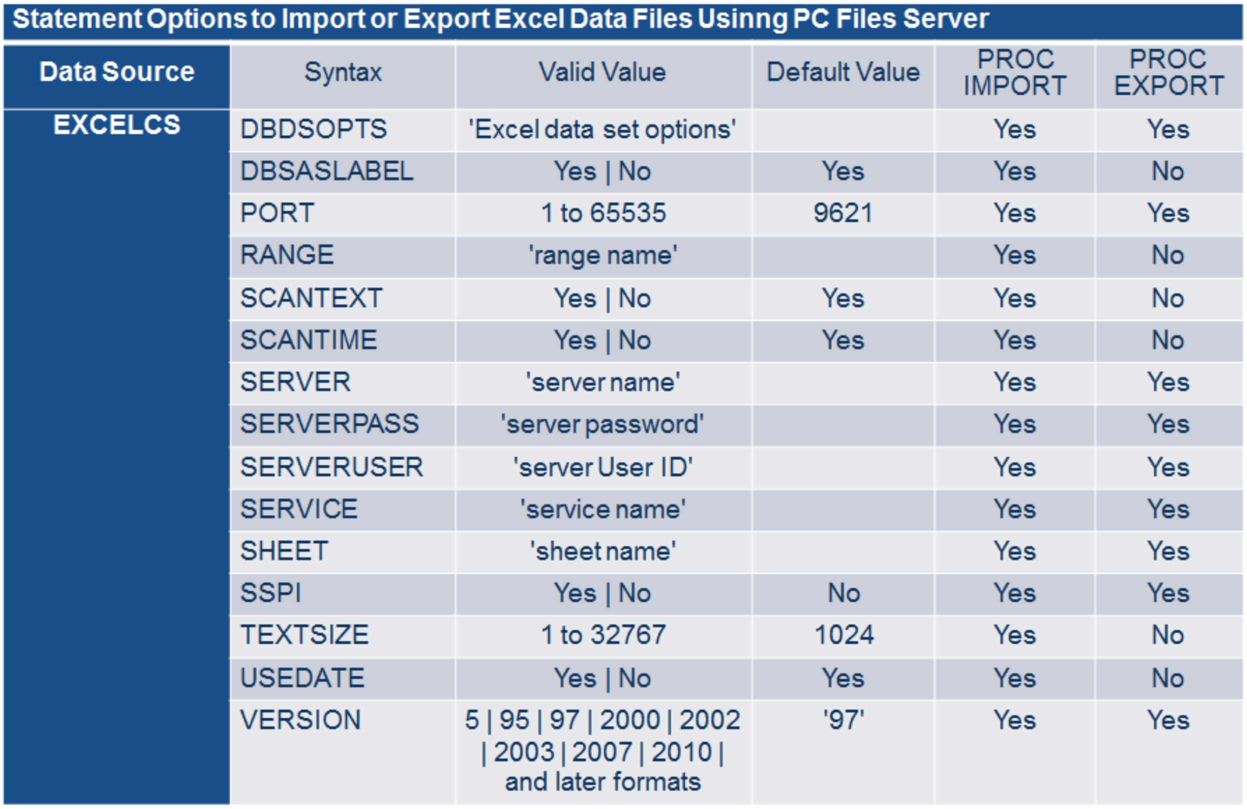
The following tables summarize the statement options of different technologies.



Statement Options to Import or Export Excel Data Using Excel Libname engine

Data Source	Syntax	Valid Value	Default Value	PROC IMPORT	PROC EXPORT
EXCEL	DBDSOPTS	Excel data set options		Yes	Yes
	DBSASLABEL	Compat | None	Compat	Yes	No
	GETNAMES	Yes | No	Yes	Yes	No
	MIXED	Yes | No	No	Yes	No
	NEWFILE	Yes | No	No	No	Yes
	RANGE	range name		Yes	No
	SCANTEXT	Yes | No	Yes	Yes	No
	SCANTIME	Yes | No	Yes	Yes	No
	SHEET	sheet name		Yes	Yes
	TEXTSIZE	1 to 32767	1024	Yes	No
	USEDATE	Yes | No	Yes	Yes	No

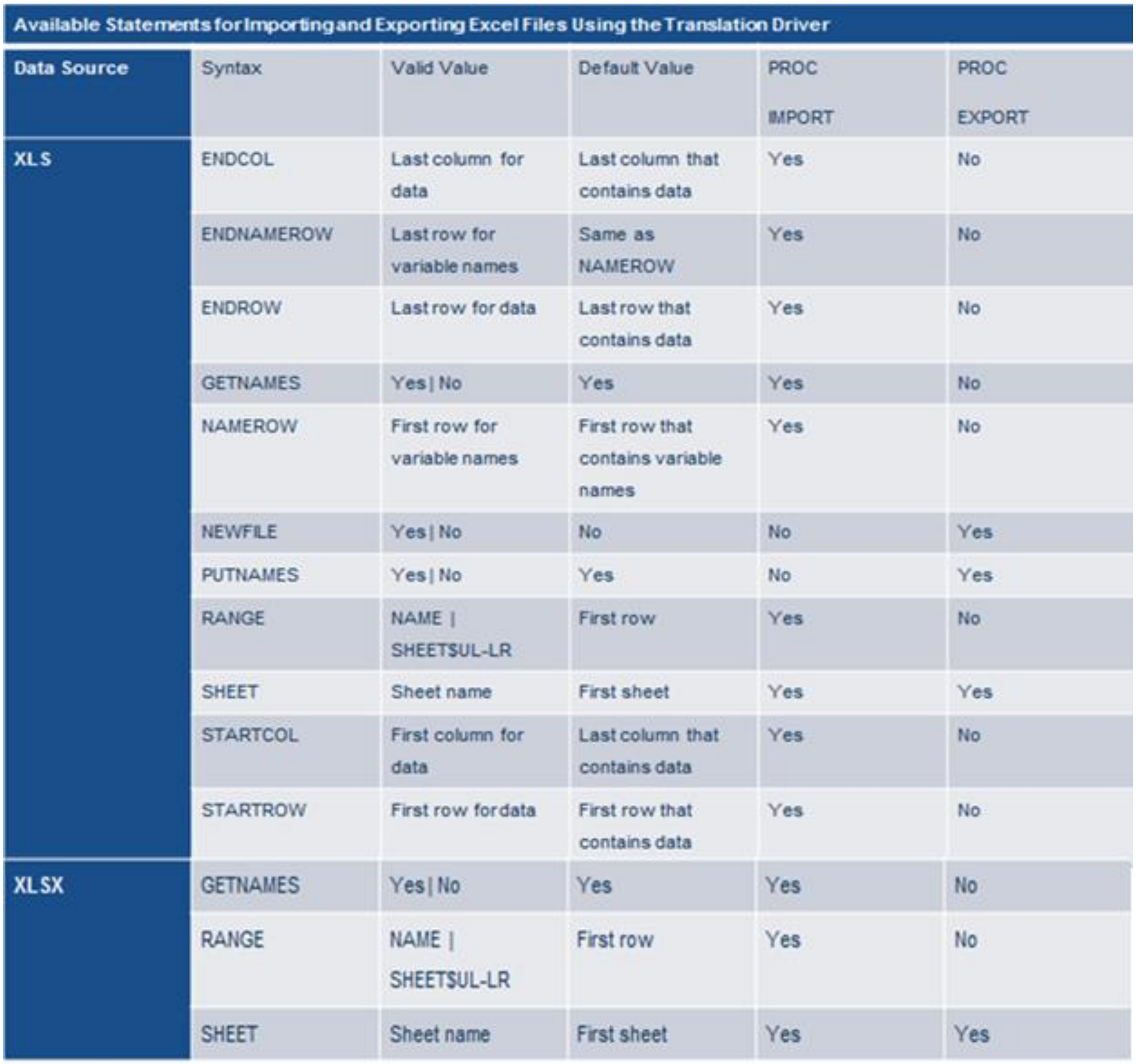
Display 3: Statement options for DBMS = EXCEL in proc import/export



Statement Options to Import or Export Excel Data Files Usinng PC Files Server

Data Source	Syntax	Valid Value	Default Value	PROC IMPORT	PROC EXPORT
EXCELCS	DBDSOPTS	'Excel data set options'		Yes	Yes
	DBSASLABEL	Yes | No	Yes	Yes	No
	PORT	1 to 65535	9621	Yes	Yes
	RANGE	'range name'		Yes	No
	SCANTEXT	Yes | No	Yes	Yes	No
	SCANTIME	Yes | No	Yes	Yes	No
	SERVER	'server name'		Yes	Yes
	SERVERPASS	'server password'		Yes	Yes
	SERVERUSER	'server User ID'		Yes	Yes
	SERVICE	'service name'		Yes	Yes
	SHEET	'sheet name'		Yes	Yes
	SSPI	Yes | No	No	Yes	Yes
	TEXTSIZE	1 to 32767	1024	Yes	No
	USEDATE	Yes | No	Yes	Yes	No
	VERSION	5 | 95 | 97 | 2000 | 2002 | 2003 | 2007 | 2010 | and later formats	'97'	Yes	Yes

Display 4: Statement options for DBMS = EXCELCS in proc import/export



Available Statements for Importing and Exporting Excel Files Using the Translation Driver

Data Source	Syntax	Valid Value	Default Value	PROC IMPORT	PROC EXPORT
XLS	ENDCOL	Last column for data	Last column that contains data	Yes	No
	ENDNAMEROW	Last row for variable names	Same as NAMEROW	Yes	No
	ENDROW	Last row for data	Last row that contains data	Yes	No
	GETNAMES	Yes | No	Yes	Yes	No
	NAMEROW	First row for variable names	First row that contains variable names	Yes	No
	NEWFILE	Yes | No	No	No	Yes
	PUTNAMES	Yes | No	Yes	No	Yes
	RANGE	NAME | SHEET$UL-LR	First row	Yes	No
	SHEET	Sheet name	First sheet	Yes	Yes
	STARTCOL	First column for data	Last column that contains data	Yes	No
	STARTROW	First row for data	First row that contains data	Yes	No
XLSX	GETNAMES	Yes | No	Yes	Yes	No
	RANGE	NAME | SHEET$UL-LR	First row	Yes	No
	SHEET	Sheet name	First sheet	Yes	Yes

Display 5: Statement options for DBMS = XLS, XLSX in proc import/export

SAS connects MS Office files by SAS LIBNAME Statements:

Through MS ACE Driver

- Excel LIBNAME statement
- Access LIBNAME statement

By File Format Technology

- XLSX LIBNAME statement

By PC Files Server

- PCFILES LIBNAME statement

TRANSFERRING SAS DATA INTERACTIVELY

Import and Export Wizards are the convenient tools for transferring data from SAS to external data files. When SAS is upgraded to 9.4, the Wizards will have some new features.

Use Import Wizard to read data from Excel and Access

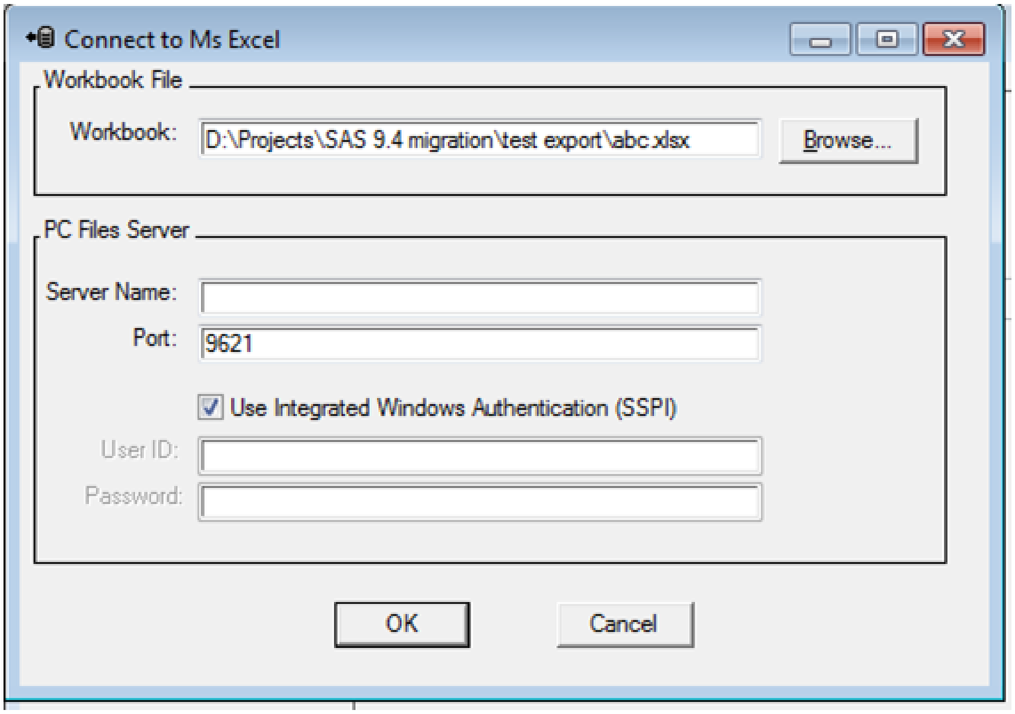
When SAS and MS Office are bit compatible, to read an Excel file, the data sources you could specify are "Microsoft Excel Workbook(*.xls, *.xlsb, *,xlsm and .xlsx)." Through the data source selected, SAS determines the engine to use to import. For the above data sources, the SAS EXCEL LIBNAME engine is used. To read from Access databases, the data source could be "Microsoft Access Database(.mdb, *.accdb)," which is through the SAS Access LIBNAME engine.

The same data sources, however, could not be selected for the scenario of SAS and MS Office bit incompatible. You would get errors by specifying those data sources. Instead, to read an Excel file, the data source "Microsoft Excel Workbook on PC Files Server" should be specified. If you specify Microsoft Excel Workbook on PC Files Server, SAS assigns DBMS = EXCELCS, which indicates the Excel Driver of PC File Server is used. The Excel file can be in any of these formats*: XLS, XLSX*, *XLSB, or XLSM.*

To read from Access databases, "Microsoft Access Database on PC Files Server" should be used. By specifying that source, SAS assigns DBMS = ACCESSCS, which indicates the Access Driver of PC File Server is used.

Use Export Wizard to write the SAS data to Excel and Access files

When SAS and MS Office files are bit incompatible, select the same data sources as in Import Wizard to export the files. In several places, using Excel Driver of PC File Server does not work exactly the same as using the SAS EXCEL LIBNAME engine when writing to Excel files. Users need to pay attention to the file location of the Excel file. Your entry for the Workbook location depends on whether the Excel file already exists.



Workbook: D:\Projects\SAS 9.4 migration\test export\abc.xlsx

Port: 9621

☑ Use Integrated Windows Authentication (SSPI)

Display 6: The window of specifying the output file in proc export

If the Excel file exists, you can navigate to its location using the Browse dialog box. The Excel file can be in any of these formats: XLS, XLSX, or XLSB. If the Excel file does not exist, enter the full directory path, including the Excel filename that you want to create. Do not use the Browse button. However, you can copy the directory path using Windows Explorer.

Any new Excel file must have an .XLSB or .XLS extension on it. It cannot have any other extension type. If you need a different extension, open Excel and save a default XLSX file that SAS can use.

INCOMPATIBLE SAS FORMAT CATALOG

The SAS catalog file is another area in which SAS 9.4 works differently than in 32-bit SAS. If a SAS catalog file created by 32-bit SAS is opened in a 64-bit SAS environment, you will get errors. Look at the following codes:

```
libname dsn "&path.\sasdata";

options fmtsearch=(dsn.par, dsn.stu);/* par and stu are 32-bit format
catalogs*/

data parent_with_fmt;

set dsn.parent;

Format p_mode fmode_.;

run;
```

Please note that “par” and “stu” in the above program are 32-bit format catalog files. When the above codes are submitted in a 64-bit SAS system, the following errors occur:

ERROR: File DSN.PAR.CATALOG was created for a different operating system.

ERROR: File DSN.STU.CATALOG was created for a different operating system.

ERROR 48-59: The format FMODE_ was not found or could not be loaded.

THE CAUSE OF THE ERRORS

SAS catalog file from the 32-bit SAS system and the 64-bit SAS system have different internal file structures, even though they are in the same operating environment. SAS considers them to be files from different platforms. Unlike the SAS data sets, SAS catalogs are not portable across platforms, because Cross-Environment Data Access (CEDA) does not support catalog files.

SOLUTIONS

Convert catalog file to SAS data set and then convert back to catalog

Because SAS data files can be moved and accessed transparently across platforms by CEDA, you can use PROC FORMAT to convert a catalog file into a data set in 32-bit SAS, then use the same procedure to change the data set back to a catalog file in 64-bit SAS. Please look at the following examples:

In 32-bit SAS, submit the following code:

```
      libname fmt 'D:\test_catalog';

      libname out 'D:\test_catalog\datasets';

      proc format library=fmt.airline cntlout=out.airline;

      Run;
```

This program will create a data set called *airline*.sas7bdat in the directory “datasets,” containing all the information about the formats in the catalog file.

64-bit SAS can then create the new catalog file by using the following code:

```
libname fmt 'D:\test_catalog';

libname in 'D:\test_catalog\datasets';

proc format library=fmt.airline_64 cntlin=in.airline;

Run;
```

This will read the data set airline.sas7bdat and write the formats in a catalog file airline_64.sas7bcat again.

Use PROC CPORT/PROC CIMPORT

Another alternative for moving catalog files is to use Proc CPORT and CIMPORT. The advantage of this approach is to convert a lot of catalog files. PROC CPORT and CIMPORT can also convert multiple data sets. The catalogs and data sets can be put in a single directory. In 32-bit SAS, you will create a single .cpt file containing all the catalogs and data sets in the directory, then have 64-bit SAS convert that .cpt file back into 64-bit versions of the original catalogs and data sets.

Run the following code by 32-bit SAS:

```
libname my32 ‘d:\my32files'; /*32 bit files source folder*/

Filename all ‘d:\mydemo\trans.cpt’;

proc cport library=my32 file=all memtype=catalog; run;
```

In 64-bit SAS, submit the following code:

```
libname my64 'c:\my64files'; /* 64-bit target folder*/

Filename all ‘d:\mydemo\trans.cpt’;

proc cimport infile=all library=my64; run;
```

My64 will contain all the 64-bit catalogs and data sets.

CONCLUSION

This paper is a comprehensive document for the areas in which 64-bit SAS system works differently than in 32-bit SAS in terms of communicating with MS Office files—specifically, MS Excel and Access. When SAS and MS Office are not bit compatible, the new solutions are provided in 64-bit SAS system to bridge the gap of bitness. In addition, this paper summarizes all of the approaches of connction by libname statements and by proc import/export, respectively. It discusses several points in which Import and Export Wizard users must pay attention. Finally, the paper addresses the issue of SAS catalog files being incompatible between 32-bit and 64-bit SAS systems.

RECOMMENDED READING

- *SAS/Access® 9.4 Interface to PC Files: Reference, Third Edition*
- *SAS® 9.4 PC Files Server Installation and Configuration*
- *System Files Requirement for SAS 9.4 Foundation for Microsoft Windows for x64*
- *Moving and Accessing SAS® 9.4 Files*

CONTACT INFORMATION

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

Hong Zhang
Mathematica Policy Research
(609) 799-3535
HZhang@mathematica-mpr.com
http://www.mathematica-mpr.com