PharmaSUG China 2014 - Paper PT09

A simple way to access the data in EXCEL through SAS v9/ ACCESS® libname and Excel engine

Lianbo Zhang, FMD, Nanjing, JS

ABSTRACT

Aiming to summarize libname access and Excel engines which is useful to read excel file into SAS dataset but not familiar to many people who prefer to "PROC IMPORT" or "infile input". This paper will discuss the principle and useful techniques about data transform between SAS and Excel, including named range, spreadsheet, libname, Excel engines. And also solving some problems we may meet. We use SAS® 9.3 64-bit and Excel 2013 64-bit in win7 x64. SAS/ACCESS software for PC files must be available when we use the Excel engine.

INTRODUCTION

It is very common for a SAS programmer to import data from MS Excel, SAS offers more ways to get information from the world's best analysis software (SAS) into the world's most ubiquitous analysis software (MS Excel). Most people will choose the method:

- ① "PROC IMPORT DBMS=EXCEL", it is very familiar to us, but we all know, "PROC IMPORT" may meet many problems when the data is not very clean. And its ability to treat different formats or some Symbols in dataset is limited.
- Save.xls, .xlsx file as .csv, then use "infile input". Anyway, "infile input" is far more powerful than "PROC IMPORT", but the convert progress of different files may cause to information lost.

So we need a moderate method. Since SAS version 9, the EXCEL libname engine has been available to installations that license the SAS Access to PC Files software. This engine allows programs to interact with Excel workbooks almost as if they are native SAS Data sets. Excel libnames can be used with virtually any SAS procedure and with the data step.

This article will introduce the SAS/ACCESS Libname, excel engine, with common use options, and summarize the issues occurring and corresponding solutions. May it can provide readers a good way to treat dataset in their work.

INSTALL CONFIGURATION

Use excel engine requires the "bitness" of Access Database Engine (ACE driver) and PC Files Server in SAS 9.3 be same. Only one version of the ACE driver and PC Files Server can be installed on x64 system.

It means if we use 64-bit SAS 9.3 but a 32-bit MS office we should change our Access Database Engine version to 64-bit, or we will see the message below:

ERROR: Connect: Class not registered ERROR: Error in the LIBNAME statement. Connection Failed. See log for details.

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

EXCEL LIBNAME STATEMENT

LIBNAME statement to reference not only SAS files but also files that were created with other software products, such as database management systems.

SAS can read or write these files by using the appropriate engine for that file type. The basic Excel libname syntax should be familiar to most SAS users:

```
LIBNAME libref<engine-name><physical-file-name><libname-options>; <SAS Code>
LIBNAME libref CLEAR;
```



where

- •libref is 1 to 8 characters long, begins with a letter or underscore, and contains only letters, numbers, or underscores.
- •Engine is the name of a library engine that is supported in operating environment. SAS can read or write these files by using the appropriate engine for that file type. For some file types, it is necessary to tell SAS which engine to use.
- •SAS-data-library is the name of a SAS library in which SAS data files are stored. The specification of the physical name of the library differs by operating environment.
- ●The LIBNAME statement is global, which will exist and exclusive lock on an existing Excel file, unless we modify them, cancel them, or end SAS session.

EXPORT SAS DATASET TO EXCEL FILE

On purpose to show more details of importing progress, we will first export some demo data sets into an excel file, create an example.

We have many ways to achieve that: data step, proc datasets, proc sql, etc.

```
libname workbook 'd:\workbook.xlsx';
* data step;
data workbook.cars;
 set sashelp.cars;
run;
data workbook.class;
 set sashelp.class;
run;
* proc datasets ;
proc datasets nolist;
copy out=workbook in = sashelp;
select cars class;
quit;
* proc sql ;
proc sql;
 create table workbook.cars as
 select * from sashelp.cars;
 create table workbook.class as
```

```
select * from sashelp.class;
quit;
* we must clear the libname first, if we want to open the excel data;
libname workbook clear;
```

Several problems we may meet:

1. When using "proc datasets" to copy sorted data, there will be a warning as below:

```
WARNING: Engine EXCEL does not support SORTEDBY operations. SORTEDBY information cannot be copied.
```

It is due to SORTEDBY flag in a dataset header cannot be copied into the excel files. To remove these information, we need first modify the data set though <statement Modify &option SORTEDBY= _null_ > in "proc datasets", which avoids rewriting the entire data set.

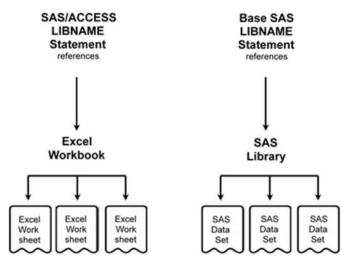
SAS excel engine can't support the REPLACE option, so it cannot overwrite the same excel file which already created, and will generate the error as below:

```
ERROR: The MS Excel table class has been opened for OUTPUT. This table already exists, or there is a name conflict with an existing object. This table will not be replaced. This engine does not support the REPLACE option.
```

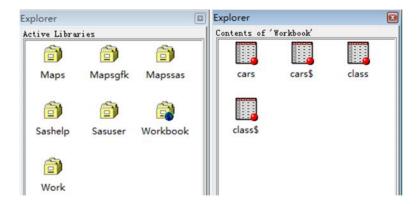
- 3. Unlike SAS dataset libname, the excel libname must be clear, otherwise the excel file will be locked, occupied by SAS, thus cannot be opened by MS Excel.
- 4. In version 2007, the created workbook.xlsx file, will be auto backup, generate "Backup of workbook.xlk".it seems an Excel issue in 2007 not SAS, which disappeared in 2013. To disable the function, we can operate in 2007:
 - ① Click the "File" tab and then "Save As."
 - 2 Click "Tools" and choose "General Options"
 - ③ Un-check the "Always create backup" box to stop Excel from automatically backing up our files.

DIFFERENCE BETWEEN SAS/ACCESS LIBNAME AND BASE SAS LIBNAME

Through excel engine, a whole Excel file will be viewed as a SAS library, the members inside (spread sheet or named range) will be viewed as data sets.



in SAS Explorer, we see its icon is different from other SAS libraries, which means it is not a normal library of SAS data sets.



We can also submit: libname_all_list; to see detail message in log.

```
9
NOTE:
      libname _all_ list;
Libref= WORKBOOK
       Libref=
       Scope=
                    DMS Process
       Engine=
                    EXCEL
        Physical Name= d:\workbook.xlsx
        User= Admin
                     SASUSER
NOTE: Libref=
       Scope=
Engine=
                     Kernel
V9
        Physical Name= D:\User\Documents\My SAS Files\9.3
Filename= D:\User\Documents\My SAS Files\9.3
Libref= WORK
       Filename=
Libref=
                    Kerne 1
V9
        Scope=
        Engine=
                     TEMP
        Access=
        Physical Name= C:\Users\user\AppData\Local\Temp\SAS Temporary File:
       Filename= C:\Users\user\AppData\Local\Temp\SAS Temporary Files\_TD
```

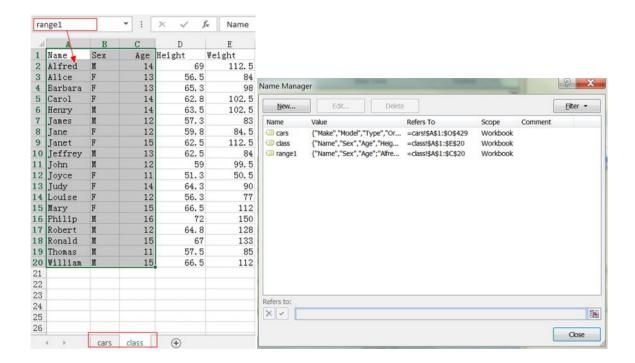
We will also find many difference between the data store in SAS and Excel. All of them are attributed to that SAS data is stored in database, while excel data is stored in table. One prefers to store the value and the attribution, and the other one prefers to show the appearance.

SPREADSHEET AND NAMED RANGE

Pay attention to the name literals of data sets in Workbook.xlsx, each of them have another copy which name have a '\$' in the tail. What is it means?

In this example, the name with a '\$' (cars\$, class\$) stands for the spreadsheet. In an Excel document, the sheet's name is always with a '\$', we just can't see it in Excel, but it exists. As for SAS data set, names special characters are not be allowed.

The name without a'\$' (cars, class) stands for the named range in spreadsheet. Named Range is a range of cells within a worksheet that you define in Excel and assign a name to. Just choose the cells we want, type a name into name box on the right top of table, and then press **ENTER.**



We can also press CTRL + F3 to manage the named range.

Named range is very useful in Excel. More detail information are on MS Office support website.

IMPORT EXCEL FILE INTO SAS DATA SETS THROUGH EXCEL ENGINE

Now we begin our major mission: import excel file into SAS data sets. SAS recognize both spreadsheet and named range in Excel document as data sets. We use the table 'Workbook.xlsx', import its contents into our SAS data sets.

IMPORTDATA IN SPREADSHEET

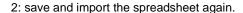
```
libname workbk1 'd:\workbook.xlsx';
data cars;
set workbk1.'cars$'n;
run;
data class;
set workbk1.'class$'n;
run;
libname workbk1 clear;
```

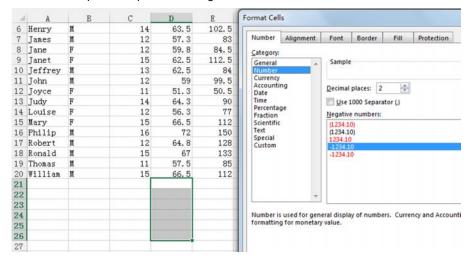
Name Literals, 'cars\$'n and "cars\$"n are both OK, n represents name, \$ represents spreadsheet.

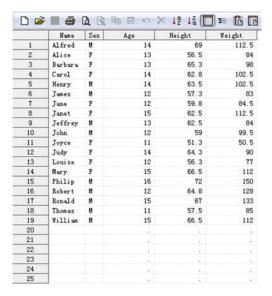
We generate two data sets in work library, they are from the two spreadsheets (cars, class) in 'workbook.xlsx', SAS will recognize and read the whole valid data in spreadsheet, even if the data are not start from 1st row or 1st column. But sometimes the cells which outside our Valid data's zone are activated but without any visible value, SAS will still read them into dataset, generate the new variable name 'F1 F2 F3 etc.', assign missing value to them.

Let's show an example to explain "cells activated but without any visible value"

1: change some blank outside cells' format, but do not assign any value in them.







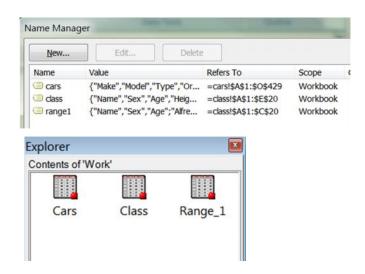
Then the imported data set adds some blank records. We may need a more accurate method to specify the range.

SOLUTION: Add absolute range

```
libname workbk1 'd:\workbook.xlsx';
data class;
set workbk1.'class$A1:E20'n;
run;
libname workbk1 clear;
```

IMPORT DATA IN NAMED RANGE

```
libname workbk2 'd:\workbook.xlsx';
data cars;
set workbk2.cars;
run;
data class;
set workbk2.class;
run;
data range_1;
set workbk2.range1;
run;
libname workbk2 clear;
```



As we mentioned before, once named range has been established, we just need reference its name (no dollar sign), then we can access the data within the name range. SAS correctly read the data in named range (cars, class and range1 which we build manually). We can define many more named ranges in different spreadsheets.

FAQ AND USEFUL OPTIONS SUPPLIED

1. How to define data's type and format when we import them into SAS?

dbSasType=

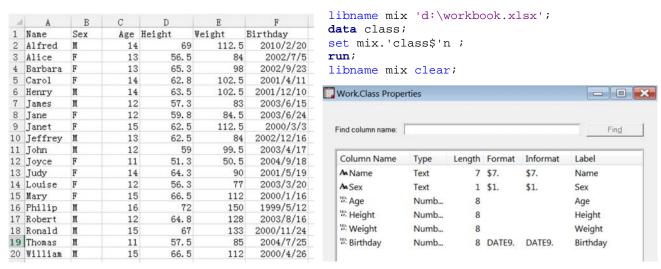
Syntax

dbSasType=(<column-name-1=<'>SAS-data-type<'>><...<column-name-n=<'SAS-data-type<'>>>)

column-name specifies a DBMS column name, which means the name in Excel not the variable name in SAS. It is recommended using standard variable name before we import it into SAS.

SAS-data-type includes: CHAR (n), NUMERIC, DATE, TIME, and DATETIME.

By default, SAS recognize the data's type according to the original data in excel, as for length and data value, we will discuss later.



We can custom define the type of different variables, using dbsastype = , or format and input / put in date step.

	Name	Sex	Age	Height	Weight	Birthday	
1	Alfred	М	14	69	112.5	2/20/2001	
2	Alice	F	13	56.5	84	7/5/2002	
3	Barbara	F	13	65.4	98	9/23/2002	
4	Carol	F	14	62.7	102.5	4/11/2001	
5	Henry	M	14	63.5	102.5 83	12/10/2001	
6	James	M	12	57.2		6/15/2003	
7	Jane	F	12	59.7	84.5	6/24/2003	
8	Janet	F	15	62.5	112.5	3/3/2000	
9	Jeffrey	M	13	62.5	84	12/16/2002	
10	John	M	12	59	99.5	4/17/2003	
11	Joyce	F	11	51.2	50.5	9/18/2004	
12	Judy	F	14	64.2	90	5/19/2001	
13	Louise	F	12	56.2	77	3/20/2003	
14	Mary	F	15	66.5	112	1/26/2000	
15	Philip	M	16	72	150	5/12/1999	
16	Robert	M	12	64.7	128	8/6/2003	
17	Ronald	M	15	67	133	11/24/2000	
18	Thomas	M	11	57.5	85	7/25/2004	
19	William	M	15	66.5	112	4/26/2000	

```
libname mix 'd:\workbook.xlsx';
data class;
set mix.'class$'n
  ((dbsastype=(age=char2 height=char4 weight=char5
birthday=char10))
  ;
run;
libname mix clear;
```

Column Name	Type	Length	Format	Informat	Label
A Name	Text	7	\$7.	\$7.	Name
A Sex	Text	1	\$1.	\$1.	Sex
A Age	Text	2	\$2.	\$2.	Age
A Height	Text	4	\$4.	\$4.	Height
A Weight	Text	5	\$5.	\$5.	Weight
A Birthday	Text	10	\$10.	\$10.	Birthday

2. How to convert Date value between SAS and Excel?

SCANTIME=YES

dbSAStype= (var_datatime=datatime)

Date value as treated as numeric both in SAS and excel.

As we know, SAS date use '01Jan1960'd as zero, after that day, each day SAS date add 1; before that day, for each day SAS date minus 1.

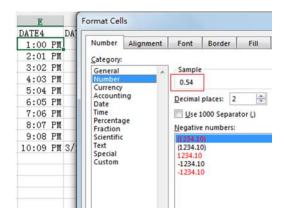
While Excel date use '01Jan1900' as one, after that day, each day Excel add 1; before that day, by default, it cannot be shown correctly, in the other words, only positive integers have their corresponding dates.

4	A	В
1	6	1900/1/6
2	5	1900/1/5
3	4	1900/1/4
4	3	1900/1/3
5	2	1900/1/2
6	1	1900/1/1
7	0	1900/1/0
8	-1	#######
9	-2	#######
10	-3	#######
11	-4	#######
12		

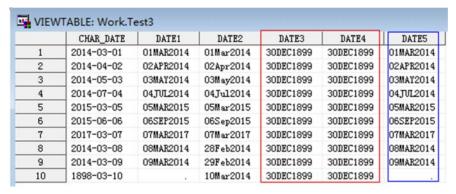
- 2	A	В	C	D	E		F
1	CHAR_DATE	DATE1	DATE2	DATES	DATE4	DATE5	
2	2014-03-01	2014/3/1	01Mar2014	8:00	1:00 PM	T .	2014/3/1 8:00
3	2014-04-02	2014/4/2	02Apr2014	9:00	2:01 PM	1	2014/4/2 9:00
4	2014-05-03	2014/5/3	03May2014	10:00	3:02 PM	1	2014/5/3 10:00
5	2014-07-04	2014/7/4	04Jul2014	11:00	4:03 PM	1	2014/7/4 11:00
6	2015-03-05	2015/3/5	05Mar2015	12:00	5:04 PM	[2015/3/5 12:00
7	2015-06-06	2015/9/6	06Sep2015	13:00	6:05 PM	[2015/9/6 13:00
8	2017-03-07	2017/3/7	07Mar2017	14:00	7:06 PM	[2017/3/7 14:00
9	2014-03-08	2014/3/8	28Feb2014	15:00	8:07 PM	[2014/3/8 15:00
10	2014-03-09	2014/3/9	29Feb2014	16:00	9:08 PM	1	2014/3/9 16:00
11	1898-03-10	1898/3/10	10Mar2014	17:00	10:09 PM	3/10/1898	5:00:00 PM
12							

To import date or time values from a MS Excel file, above. SAS LIBNAME access and excel engine read date values using DATE9.format. Time value without date part will be treated as decimal.

As for '1898/3/10', it cannot be treated as a positive integer in Excel so will be treated as TEXT, then into SAS, it will be missing, due to DATE1 is a numeric variable.

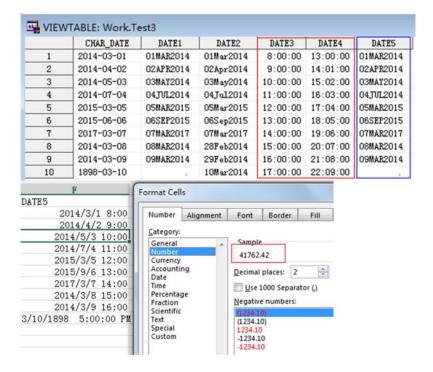


in variable DATE3 and DATE4 SAS will treat the number as date value, We need to force SAS to read our datetime variables in Excel as datetime. format in SAS,



To solve this problem, we use a libname option: scanTime=yes, then, Time values are assigned the TIME8.format.

libname cc 'd:\date.xlsx' SCANTIME=YES ;

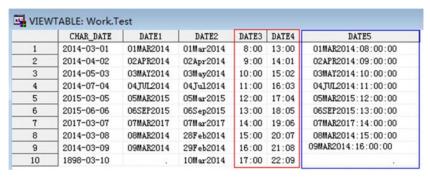


SAS reads date values (positive integer part) as date9. format, the time part (fractional part) of TIME5. is omitted. dbSasType= will be very useful to solve this problem, just define:

dbSasType= (DATE5=datetime). Then in the further step, we define the proper format to the variables which contains time part.

```
libname cc 'd:\date.xlsx' SCANTIME=YES;
data test;
  retain char_date date1-date5;
  format date5 datetime20. date3 date4 time5.;
  set cc.'sheet1$'n ((dbsastype=(DATE5=datetime));
run;
libname cc clear;
```

Now it looks much better,



3. What if one variable contains more than one type in Excel?

mixed= yes | no

What if one variable contains both numeric and character in Excel?

SAS cannot tolerant one variable has different types. By default, SAS reads the data just as it is stored in Excel. One variable's type will be determined by the first eight observations. In these 8 rows, if the type is mixed and {COUNT (Character observations) >= COUNT (Numeric observations)}, it will be set character, otherwise it will be set Numeric, and the character value will be assigned missing.

We can use mixed=yes option in the LibName statement like that:

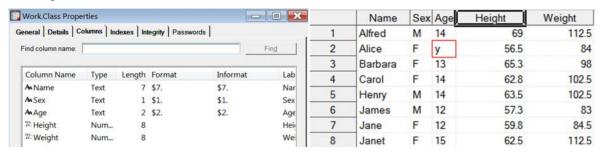
```
libname workbk1'd:\workbook.xlsx' mixed=yes;
```

compare the difference:

mixed= no; (default)



mixed=yes;



4. How to add row numbers which determine one variable's type

TypeGuessRows

In SAS excel engine, we need to change the windows registry settings, the locations are not same in each windows or MS office version.

For windows7 x64 with MS office 2013 64-bit, it is in

HKEY_LOCAL_MACHINE / Software / Microsoft /Office/15.0 /AccessConnectivityEngine /Engines/ Excel

Change the value of TypeGuessRows to 0, which by default, is 8 in hexadecimal.

Then all the rows will be scanned and checked.

But we should know that changes also affect other software that uses the Microsoft Jet provider to access Excel file data, including accessing Excel data in a Microsoft Access database.

5. Read long character

SCANTEXT = YES | NO

Specifies whether to read the entire data column and use the length of the longest string found as the SAS column width.

YES: scans entire data column and uses the longest string value to determine SAS column width.

NO: does not scan the columns and default width is set to 255.

Default is YES.

dbMax_text

If SCANTEXT = YES, SAS determines its length according to the longest value in it. Unless the length is greater than 1024 characters or we will never worry about it will be truncated!

Of course we can set greater length than 1024, dbMax_text=32767 option in the LibName statement.

Indicates the length of the longest character string, where n is an integer between 256 and 32,767.

Default is 1024.

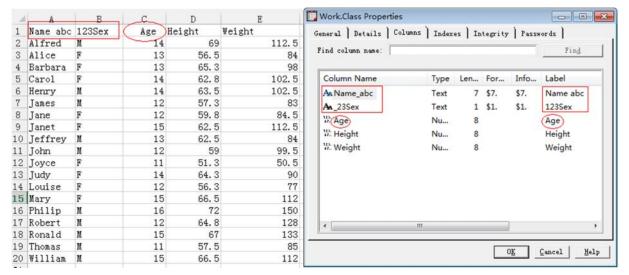
6. How to get the right name

HEADER= YES | NO

To determines whether the first row of data in a Microsoft Excel range (or spreadsheet) are column names.

By default, it is YES, if we choose NO, SAS assigns variable names F1, F2, F3, and so on.

The variable's name in SAS must begin with letter or underscore, contains at most 32 characters ([0-9a-zA-Z] and underscore only). But in Excel, there are fewer restricts. When SAS get names from column names, Non-standard places in column name will be replaced by underscore.



First, we changed some columns' name, then import them into SAS. We will find the blank among the name and the digit in the front were replaced by '_', the blanks in the front of ' Age' were omitted. SAS retains and stores the raw information in variable's label.

dbSasLabel= none

If we don't want to create any label, we can use libname statement option or dataset option: dbSasLabel=none.

7. Any difference Compared to "proc import"

An example to show **proc import** syntax:

```
PROC IMPORT OUT= WORK.demo DATAFILE= "d:\workbook.xlsx"
                                   /*identify DBMS type to read file*/
DBMS=EXCEL | xls |xlsx REPLACE;
  RANGE="CLASS$A1:J13";
                          /*In sheet CLASS identify the rectangular set of cells to
                             import, you can also use a named range*/
                          /*identify an entire spreadsheet to be read, choose only
 /*SHEET=CLASS; */
                               one statement between RANGE and SHEET */
                           /*create SAS variable names from the column names in the
  GETNAMES=YES;
                             input file's first row of data*/
  DATAROW= 2; /*starts reading data from 2<sup>nd</sup>row, valid only for delimited files */
  GUESSINGROWS=32767;
                          /*reads the number of rows before guessing col's type,
                             valid only for delimited files */
  MIXED=YES | NO;
  USEDATE=YES | NO;
  SCANTIME=YES;
RUN;
```

Proc import's functions are quite similar to SAS / ACCESS libname and Excel engine, even some options have the same name and same function, eg. MIXED, USEDATE, SCANTIME. And by default, they both read data according to the original data's type and value. We find the advantage and disadvantage in SAS / ACCESS libname and Excel engine.

Advantage:

SAS / ACCESS libname and Excel engine is more flexible and powerful than "proc import" especially in defining variable's length, format, because it can combine with data step, "proc sql" and etc.

We can directly reference the source data in excel through SAS / ACCESS libname and Excel engine. And its syntax looks much easier than "proc import".

Disadvantage:

LIBNAME which we reference, must be clear manually, or it will exist and exclusive lock on an existing Excel file, many beginners may omit this step.

CONCLUSION

Since MS Excel is widely used, it is necessary for a programmer to learn to transfer data between SAS and MS Excel. SAS / ACCESS libname and Excel engine provide us an easy way to access the data stored in MS Excel, and can retain the original information (value, type) from source data. The application of SAS / ACCESS libname and Excel engine makes it more convenience to import / export the data between SAS and excel, because it can inside many DATA/ PROC step. We can directly reference the data as long as we establish the excel libname.

REFERENCE

So, Your Data are in Excel!, Ed Heaton, Westat

De-mystifying the SAS LIBNAME engine in Microsoft Excel: A practical guide, Choate, P., Martell, C.,

Proceedings of the Thirty-First SAS Users Group International Conference, paper 024-31.

Moving Data and Results Between SAS® and Microsoft Excel, Harry Droogendyk, Stratia Consulting Inc., Lynden, ON, Canada

SAS/ACCESS® 9.3 Interface to PC Files Reference, Cary, NC: SAS Institute Inc. 2011

System Requirements--SAS 9.3 Foundation for Microsoft Windows for x64, Cary, NC: SAS Institute Inc., 2012.

SAS® Certification Prep Guide Base Programming for SAS®9, Third Edition, Cary, NC, USA, SAS Institute Inc., 2011

Contact Information

Name: Lianbo Zhang Enterprise: Fountain-med

Address:

City, State ZIP: Nanjing. 211112

Work Phone:

Fax:

E-mail: zhanglianbo35@126.com

Web:

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.