PharmaSUG 2013 - Paper CC06

# Bring Excel file with SDTM data in multiple sheets to *SAS*®

Mindy Wang, Independent Consultant, North Potomac, MD

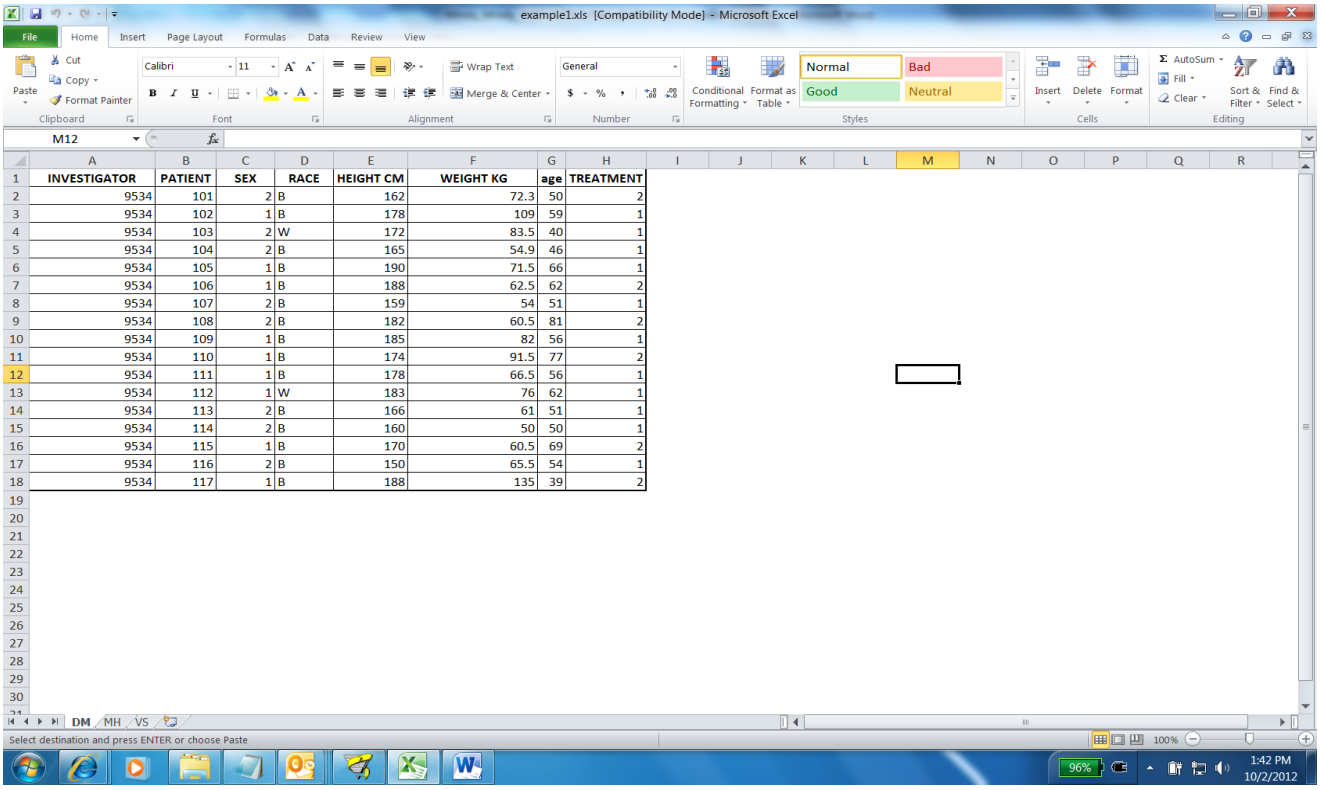
## ABSTRACT

In today's work place, Excel files seem to be the most common files that we deal with. Sometimes we encounter data come in as an Excel file with multiple spread sheets. This paper illustrates three approaches as to bring the spread sheets with SDTM data into SAS, where each sheet becomes a SAS data set. This first method is using DDE (Dynamic Data Exchange). The second method is using a simple macro program to import the multiple spreadsheets one by one. This third method is setting up the Excel file as a SAS library and bring in each sheet as a member in the library. The last method is definitely very convenient and easy to use when there are many sheets involved.

## INTRODUCTION

This paper illustrates three methods to bring in Excel file with multiple spreadsheets. Following is the original Excel file.

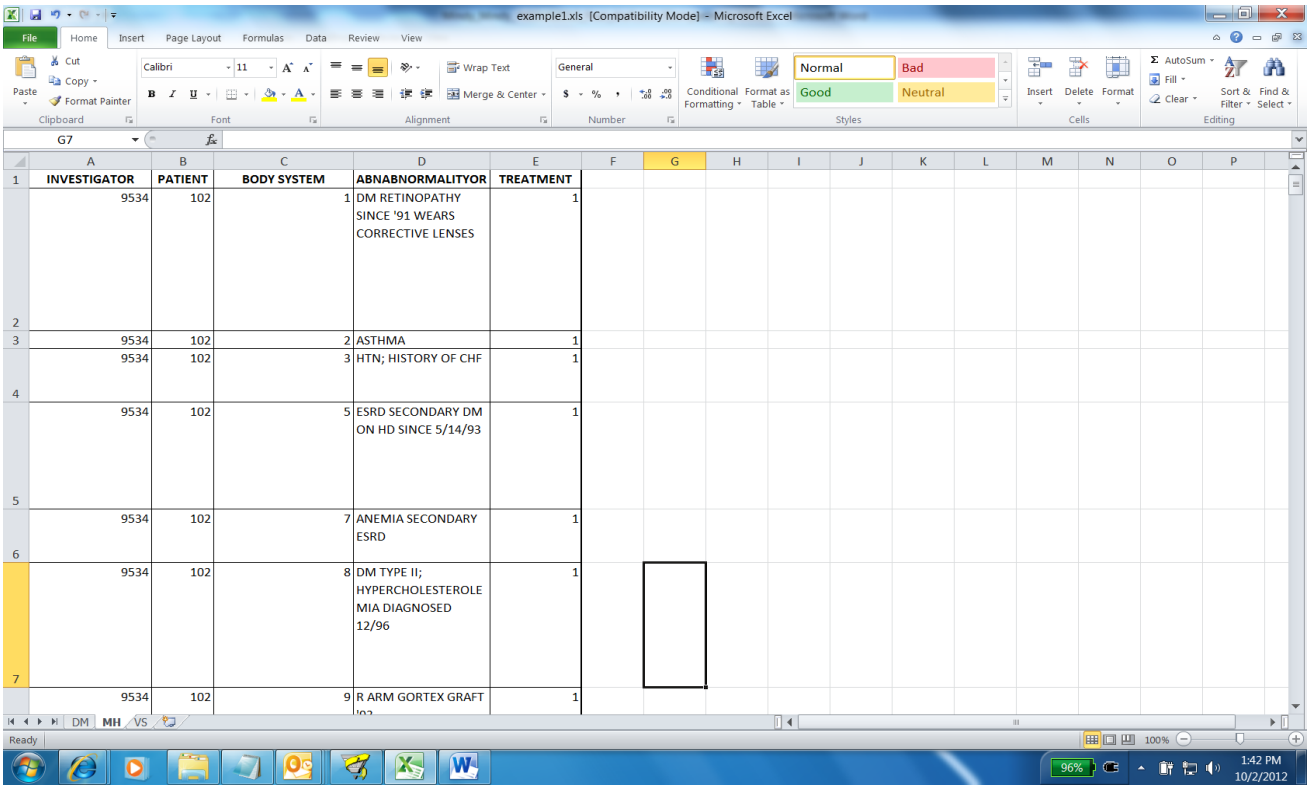
Display1 is the screen capture of the first sheet of the Excel file.




| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | INVESTIGATOR | PATIENT | SEX | RACE | HEIGHT CM | WEIGHT KG | age | TREATMENT |
| 2 | 9534 | 101 | 2 | B | 162 | 72.3 | 50 | 2 |
| 3 | 9534 | 102 | 1 | B | 178 | 109 | 59 | 1 |
| 4 | 9534 | 103 | 2 | W | 172 | 83.5 | 40 | 1 |
| 5 | 9534 | 104 | 2 | B | 165 | 54.9 | 46 | 1 |
| 6 | 9534 | 105 | 1 | B | 190 | 71.5 | 66 | 1 |
| 7 | 9534 | 106 | 1 | B | 188 | 62.5 | 62 | 2 |
| 8 | 9534 | 107 | 2 | B | 159 | 54 | 51 | 1 |
| 9 | 9534 | 108 | 2 | B | 182 | 60.5 | 81 | 2 |
| 10 | 9534 | 109 | 1 | B | 185 | 82 | 56 | 1 |
| 11 | 9534 | 110 | 1 | B | 174 | 91.5 | 77 | 2 |
| 12 | 9534 | 111 | 1 | B | 178 | 66.5 | 56 | 1 |
| 13 | 9534 | 112 | 1 | W | 183 | 76 | 62 | 1 |
| 14 | 9534 | 113 | 2 | B | 166 | 61 | 51 | 1 |
| 15 | 9534 | 114 | 2 | B | 160 | 50 | 50 | 1 |
| 16 | 9534 | 115 | 1 | B | 170 | 60.5 | 69 | 2 |
| 17 | 9534 | 116 | 2 | B | 150 | 65.5 | 54 | 1 |
| 18 | 9534 | 117 | 1 | B | 188 | 135 | 39 | 2 |

Display 1. Screen Capture of the First Sheet of the Excel File

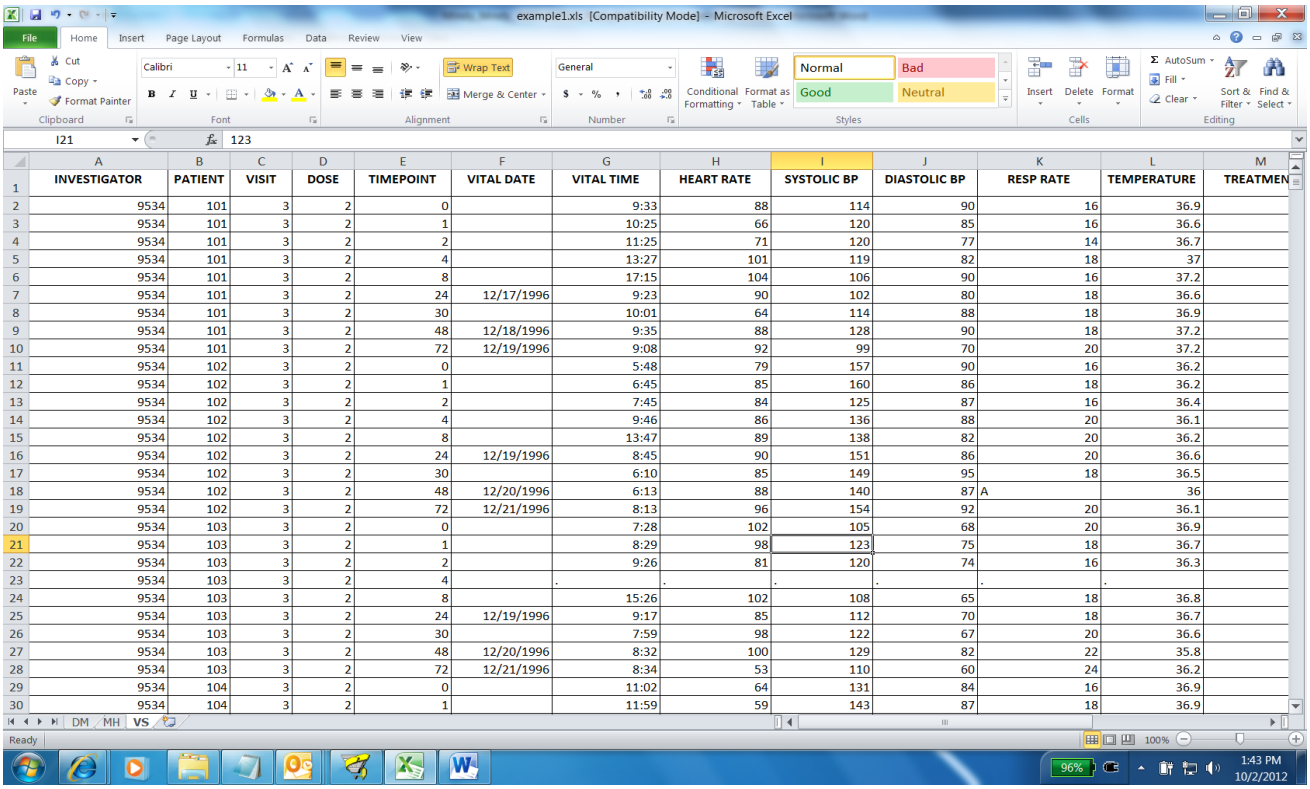
Display 2 is the screen capture of the second sheet of the Excel file.




| INVESTIGATOR | PATIENT | BODY SYSTEM | ABNABNORMALITYOR | TREATMENT |
|---|---|---|---|---|
| 9534 | 102 | 1 | DM RETINOPATHY SINCE '91 WEARS CORRECTIVE LENSES | 1 |
| 9534 | 102 | 2 | ASTHMA | 1 |
| 9534 | 102 | 3 | HTN; HISTORY OF CHF | 1 |
| 9534 | 102 | 5 | ESRD SECONDARY DM ON HD SINCE 5/14/93 | 1 |
| 9534 | 102 | 7 | ANEMIA SECONDARY ESRD | 1 |
| 9534 | 102 | 8 | DM TYPE II; HYPERCHOLESTEROLEMIA DIAGNOSED 12/96 | 1 |
| 9534 | 102 | 9 | R ARM GORTEX GRAFT | 1 |

Display 2. Screen Capture of the Second Sheet of the Excel File

Display 3 is the screen capture of the third sheet of the Excel file.




| | A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | INVESTIGATOR | PATIENT | VISIT | DOSE | TIMEPOINT | VITAL DATE | VITAL TIME | HEART RATE | SYSTOLIC BP | DIASTOLIC BP | RESP RATE | TEMPERATURE | TREATMEN |
| 2 | 9534 | 101 | 3 | 2 | 0 | | 9:33 | 88 | 114 | 90 | 16 | 36.9 | |
| 3 | 9534 | 101 | 3 | 2 | 1 | | 10:25 | 66 | 120 | 85 | 16 | 36.6 | |
| 4 | 9534 | 101 | 3 | 2 | 2 | | 11:25 | 71 | 120 | 77 | 14 | 36.7 | |
| 5 | 9534 | 101 | 3 | 2 | 4 | | 13:27 | 101 | 119 | 82 | 18 | 37 | |
| 6 | 9534 | 101 | 3 | 2 | 8 | | 17:15 | 104 | 106 | 90 | 16 | 37.2 | |
| 7 | 9534 | 101 | 3 | 2 | 24 | 12/17/1996 | 9:23 | 90 | 102 | 80 | 18 | 36.6 | |
| 8 | 9534 | 101 | 3 | 2 | 30 | | 10:01 | 64 | 114 | 88 | 18 | 36.9 | |
| 9 | 9534 | 101 | 3 | 2 | 48 | 12/18/1996 | 9:35 | 88 | 128 | 90 | 18 | 37.2 | |
| 10 | 9534 | 101 | 3 | 2 | 72 | 12/19/1996 | 9:08 | 92 | 99 | 70 | 20 | 37.2 | |
| 11 | 9534 | 102 | 3 | 2 | 0 | | 5:48 | 79 | 157 | 90 | 16 | 36.2 | |
| 12 | 9534 | 102 | 3 | 2 | 1 | | 6:45 | 85 | 160 | 86 | 18 | 36.2 | |
| 13 | 9534 | 102 | 3 | 2 | 2 | | 7:45 | 84 | 125 | 87 | 16 | 36.4 | |
| 14 | 9534 | 102 | 3 | 2 | 4 | | 9:46 | 86 | 136 | 88 | 20 | 36.1 | |
| 15 | 9534 | 102 | 3 | 2 | 8 | | 13:47 | 89 | 138 | 82 | 20 | 36.2 | |
| 16 | 9534 | 102 | 3 | 2 | 24 | 12/19/1996 | 8:45 | 90 | 151 | 86 | 20 | 36.6 | |
| 17 | 9534 | 102 | 3 | 2 | 30 | | 6:10 | 85 | 149 | 95 | 18 | 36.5 | |
| 18 | 9534 | 102 | 3 | 2 | 48 | 12/20/1996 | 6:13 | 88 | 140 | 87 | A | 36 | |
| 19 | 9534 | 102 | 3 | 2 | 72 | 12/21/1996 | 8:13 | 96 | 154 | 92 | 20 | 36.1 | |
| 20 | 9534 | 103 | 3 | 2 | 0 | | 7:28 | 102 | 105 | 68 | 20 | 36.9 | |
| 21 | 9534 | 103 | 3 | 2 | 1 | | 8:29 | 98 | 123 | 75 | 18 | 36.7 | |
| 22 | 9534 | 103 | 3 | 2 | 2 | | 9:26 | 81 | 120 | 74 | 16 | 36.3 | |
| 23 | 9534 | 103 | 3 | 2 | 4 | | . | . | . | . | . | . | |
| 24 | 9534 | 103 | 3 | 2 | 8 | | 15:26 | 102 | 108 | 65 | 18 | 36.8 | |
| 25 | 9534 | 103 | 3 | 2 | 24 | 12/19/1996 | 9:17 | 85 | 112 | 70 | 18 | 36.7 | |
| 26 | 9534 | 103 | 3 | 2 | 30 | | 7:59 | 98 | 122 | 67 | 20 | 36.6 | |
| 27 | 9534 | 103 | 3 | 2 | 48 | 12/20/1996 | 8:32 | 100 | 129 | 82 | 22 | 35.8 | |
| 28 | 9534 | 103 | 3 | 2 | 72 | 12/21/1996 | 8:34 | 53 | 110 | 60 | 24 | 36.2 | |
| 29 | 9534 | 104 | 3 | 2 | 0 | | 11:02 | 64 | 131 | 84 | 16 | 36.9 | |
| 30 | 9534 | 104 | 3 | 2 | 1 | | 11:59 | 59 | 143 | 87 | 18 | 36.9 | |

Display 3. Screen Capture of the Third Sheet in the Excel File

## DDE METHOD

The first method is using DDE (Dynamic Data Exchange). For DDE to work, first we need to have the Excel spreadsheet open, when we run the SAS program. Also we need to specify the specific variables and data types (such as numeric or character), as well as the ranges of rows and columns which contain data in the SAS program. Then when we bring the next sheet to SAS, we need to open the next sheet in Excel and type in all the variable information in the SAS program. While this method is working, if the Excel file contains many spreadsheets, this method can be very tedious. We need to manually open each sheet and for the very minimum, we need to know all the variable names and data types (numeric or character) in each sheet. There is still room for improvement.

Following is the program code using DDE:

```
FILENAME in1 DDE "Excel|E:\NESUG2012\[example2.xls]DM!r1c1:r300c1000";
DATA DM;
  INFILE in1 DSD DLM='09'x NOTAB MISSOVER FIRSTOBS=2;
  INPUT INVESTIGATOR PATIENT SEX RACE $ HEIGHT_CM WEIGHT_KG age TREATMENT;
RUN;

FILENAME in2 DDE "Excel|E:\NESUG2012\[example2.xls]MH!r1c1:r300c1000";
DATA MH;
   INFILE in2 DSD DLM='09'x NOTAB MISSOVER FIRSTOBS=2;
   INPUT INVESTIGATOR PATIENT BODY_SYSTEM ABNORMALITY $ TREATMENT;
RUN;
..............
```

## USE MACRO TO EXPORT THE FILE

We can write a simple the MACRO program to bring in the multiple sheets one by one to SAS using the following SAS codes. This program has the advantage of not needing to open the Excel spread sheet one by one as the DDE method mentioned above. Also, we don't need to type in the variable names and types. However in this program, we still need to put in the sheet name one by one. There is still room for improvement.

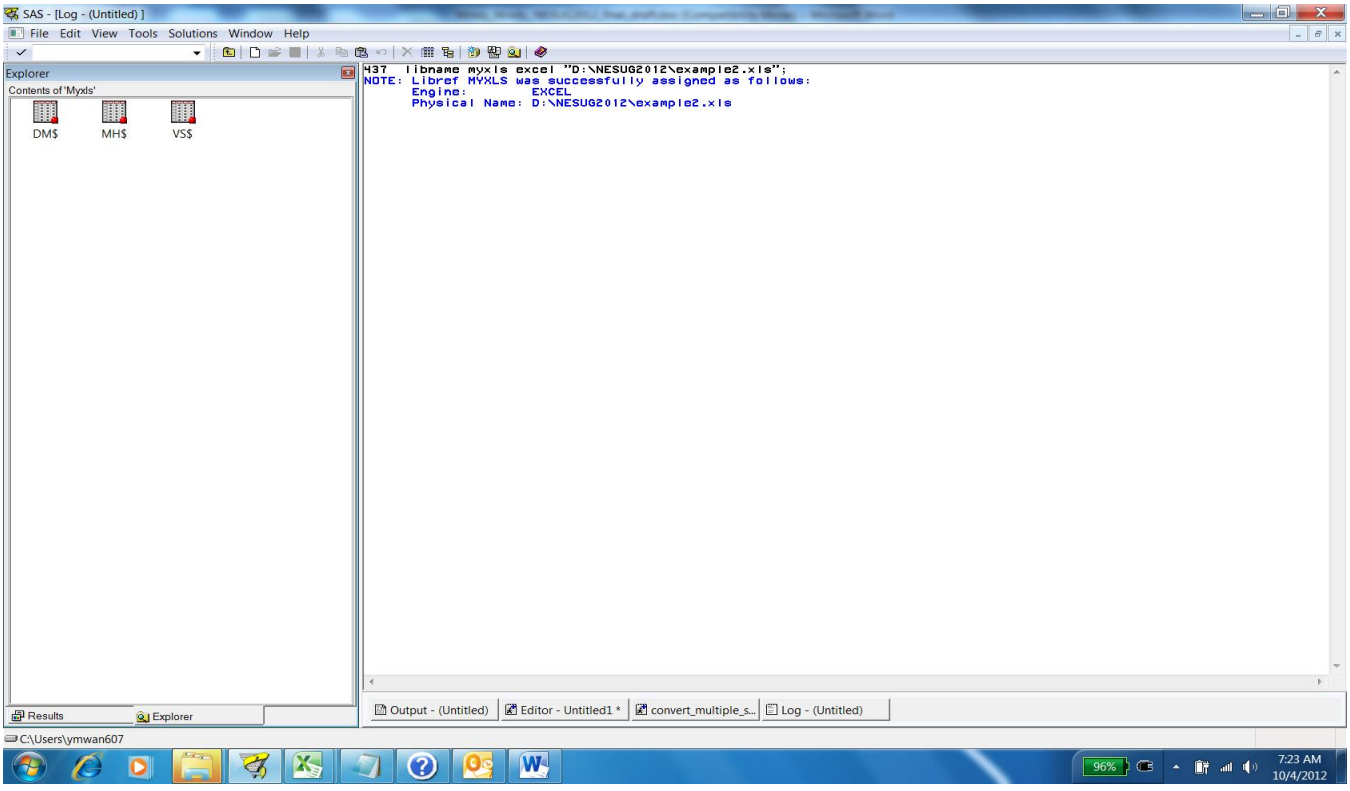
Following is the program code using macro to export the file:

```
OPTIONS mprint;

%MACRO imp (insheet=);
       PROC IMPORT OUT=work.&insheet
              DATAFILE= "E:\NESUG2012\example2.xls"
              DBMS=Excel replace;
              SHEET="&insheet.$";
              GETNAME=YES;
              MIXED=NO;
              SCANTEXT=YES;
              USEDATE=YES;
              SCANTIME=YES;
       RUN;
%MEND imp;

%imp(insheet=DM)
%imp(insheet=MH)
%imp(insheet=VS)
```

## USE LIBNAME TO POINT TO THE EXCEL FILE

In SAS version 9.1 or up, there is an option to treat Excel file as a library and all the sheets in the Excel file as data sets in the library. With this improvement, it is much easier to bring the multiple spreadsheets to SAS. If you don't mind that the data sets coming from Excel end with $ at the end of the data set names, then you are done by setting up the LIBNAME. Each Excel spreadsheet is now a SAS data set in the library. In this example, they are DM$, MH$, and VS$.

Display 4 is screenshot of the library where the SAS data set names have a $ at the end.




Display 4. Screen Capture of the SAS Data Set Names with $ at the End

However, if you need to have a data set name without the $ sign at the end, there are still some work need to be done. With the LIBNAME method, we can use the SQL procedure to query dictionary.tables to find out how many sheets the Excel file has and save it as a macro variable. We can also save the sheet names as a macro variable.

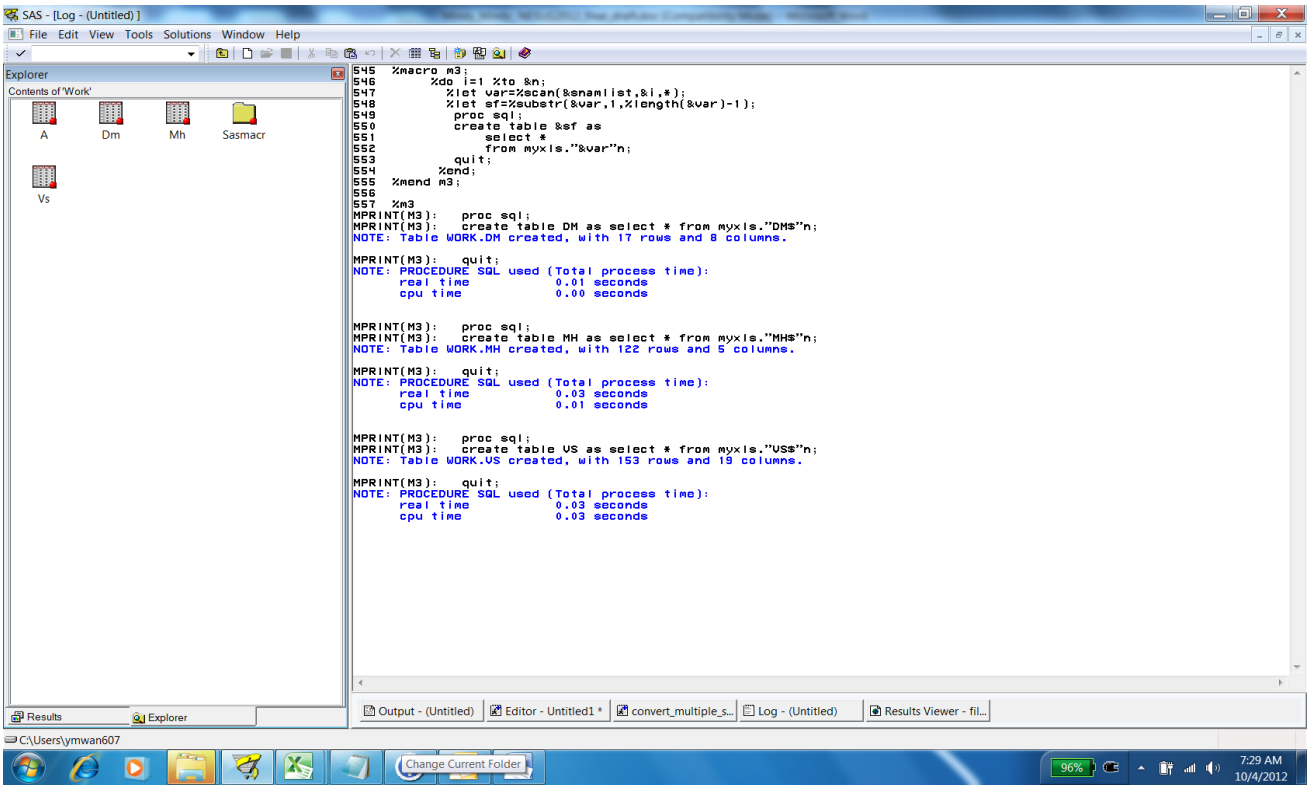
Then with a simple macro program all the import procedures can be done to multiple sheets. Macro programs m1, m2, and m3 are three alternatives the data sets without $ sign at the end can be done.

Following are the code that brings the spread sheets to SAS without the $ at the end of the data set name:

```
LIBNAME myxls Excel "E:\NESUG2012\example2.xls";

PROC SQL;
CREATE TABLE a AS
SELECT *
FROM dictionary.tables
WHERE LIBNAME="MYXLS";
QUIT;

PROC SQL;
SELECT memname
INTO :snamlist SEPARATED BY '*'
FROM a;
QUIT;

PROC SQL;
SELECT count(memname)
INTO :n
FROM a;
QUIT;

%PUT &snamlist;
%PUT &n;

OPTIONS mprint;

%MACRO m1;
       %DO i=1 %TO &n;
       %LET var=%SCAN(&snamlist,&i,*);
              PROC IMPORT OUT= WORK.%SUBSTR(&var,1,%length(&var)-1)
                   DATAFILE= "E:\NESUG2012\example2.xls"
                   DBMS=EXCEL REPLACE;
                   RANGE="&var";
                   GETNAMES=YES;
                   MIXED=NO;
                   SCANTEXT=YES;
                   USEDATE=YES;
                   SCANTIME=YES;
              RUN;
       %END;
%MEND m1;
%m1

/*ALTERNATIVELY USE MACRO PROGRAM m2 INSTEAD OF m1*/
%MACRO m2;
       %DO i=1 %TO &n;
        %LET var=%SCAN(&snamlist,&i,*);
         %LET sf=%SUBSTR(&var,1,%length(&var)-1);
                    DATA &sf;
                           SET myxls."&var"n;
                    RUN;
       %END;
%MEND m2;
%m2

/*ALTERNATIVELY USE MACRO PROGRAM m3 INSTEAD OF m1*/
%MACRO m3;
        %DO i=1 %TO &n;
           %LET var=%scan(&snamlist,&i,*);
```

```
            %LET sf=%SUBSTR(&var,1,%length(&var)-1);
               PROC SQL;
               CREATE TABLE &sf AS
                      SELECT *
                      FROM myxls."&var"n;
               QUIT;
         %END;
%MEND m3;
%m3
```

Display 5 is the screenshot of the data sets without $ at the end of data set name.




Display 5. Screen Capture of the SAS Data Set Names without $ at the End

## CONCLUSION

Of the methods shown, the last method is preferred since it is the most dynamic. In the last method, the number of spreadsheets, names/types of variables, and row/column ranges do not need to be known. This allows the method to be applied to different projects more easily. With PROC SQL, we pull out the total number of sheets and the sheet names in the Excel file and save them as macro variables. Once the macro is set up, the only change we need to do to make it work in another project is to change the LIBNAME statement and the file name in the DATAFILE= to point to the right Excel file. Much of the typing is largely reduced and less likely errors might occur due to manual manipulation.

## RECOMMENDED READING


- SAS® Macro Language 1: Essentials Course Notes
- Carpenter, Art (2004) Carpenter's Guide to SAS® Macro Language Second Edition
- Burlew, Michele M. (2006) SAS® Macro Programming Made Easy
- Shostak, Jack (2005) SAS® Programming in the Pharmaceutical Industry

## CONTACT INFORMATION

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

Name: Mindy Wang
Enterprise: Independent Consultant
Address: 14412 Stonebridge View Dr.
City, State ZIP: North Potomac, MD 20878
Work Phone: 240-855-3479
E-mail: ymindywang@yhaoo.com