

MWSUG 2019 - Paper xxxx
The XLSX Engine with PROC SQL Dictionary tables
Michael Harper, Independent

ABSTRACT

With the XLSX libname engine, individual worksheet tabs become logical tables you can reference in SAS

INTRODUCTION

The XLSX engine became available in SAS 9.4 M2. Here is the simple syntax:

```
Libname libref XLSX "C:\MyFiles\Workbook.xlsx" ;
```

Or Libname xlsxref xlsx "/folders/myfolders/Workbooks/Newdata.xlsx";

There are many legacy alternatives, such as PROC IMPORT, export the data from Excel as .csv then infile input dlm=',' - or ODS EXCEL. There was also the PCFILES engine that allowed you to lift a named range from the worksheet, assuming you had the SAS/ACCESS Interface to PC Files installed on your Linux or UNIX server. With PCFILES, you can directly access PC data from Linux and UNIX. I have struggled with all of these in the last 20 years.

INPUT

With the XLSX reference, the individual worksheet tabs become logical tables you can reference in SAS - whatever the platform you are running on - allowing you to read from, and write to, the worksheets in an Excel workbook. Metadata is available in the DICTIONARY tables using PROC SQL. This presents an opportunity to loop through reading the sheets with SAS macro logic without knowing the sheet names in advance. As output to Excel, you can create multiple sheets in a workbook using macro code.

```
proc sql noprint;
  select memname into :tab1 - :tab3 /* the names of 3 sheet tabs */
  from DICTIONARY.Tables
  where upcase(libname)='XLSXREF'
  ;
  /* how many columns in the Class sheet? */

  select count (distinct name) into :vcnt trimmed
  from DICTIONARY.Columns
  where upcase(libname)='XLSXREF'
  and upcase(memname)='CLASS'
  ;

%put vcnt=&vcnt.;
```

```

select name into :var1 - :var&vcnt.      /* what are the column names? */
  from DICTIONARY.Columns
  where upcase(libname)='XLSXREF'
  and upcase(memname)='CLASS';
quit;

%put _user_;

libname xlsxref; /* clear the libref */

```

OUTPUT

Imagine writing 99 sheets in a single workbook for 99 branches with macro code around a proc or data step.

```

libname xlsxref xlsx "/folders/myfolders/Workbooks/Newdata.xlsx";

%macro loop(parm=branch);
data xlsxref.&branch.;
  /* make a worksheet in the Excel file */
  set monthend;
  Where branch=&branch.;
run;
%mend loop;

%do i=1 %to 99;
  %loop(parm=&i.);
%end;

```

CONCLUSION

I want SAS to be simple. With the DICTIONARY tables and the XLSX engine, SAS and Excel can coexist in our development.

REFERENCES

"PROC SQL: Beyond the Basics using SAS" - Kirk Paul Lafler, © 2004 SAS

Vince DelGobbo has hosted many workshops and papers on this subject –
Including ODS, XML (tagsets), and EXCELXP

<http://support.sas.com/resources/papers/proceedings09/152-2009.pdf>

<https://blogs.sas.com/content/sasdummy/2015/05/20/using-libname-xlsx-to-read-and-write-excel-files/> - from Chris Hemedinger May 2015

CONTACT INFORMATION

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

Michael Harper
515-279-4663
MJHarper424@gmail.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.