Paper 74924-2011

Exporting Variable Labels as Column Headers in Excel using SAS® Chaitanya Chowdagam, MaxisIT Inc., Metuchen, NJ

ABSTRACT

Excel output is the desired format for most of the ad-hoc reports requested for an easy interpretation and exploration of data. However, to make these reports more informative variable labels could be used instead of variable names as column header. This is harder to achieve by the conventional SAS® Export procedure, PROC EXPORT. This paper will provide easy and quick methods of exporting SAS® data to multiple Excel sheets with variable labels as column headers using EXCEL Libname and ODS Markup.

INTRODUCTION

Excel is a universal platform for simple data analysis, sharing information and data interpretation. There are numerous ways to export data from SAS into Excel, and these procedures improve with evolving versions of SAS. Companies which still uses older versions of SAS 8.2 and older uses PROC EXPORT and DDE to do the job without any easy options of customizing the reports. Though DDE allows SAS to run VBA macros within the excel workbook and data formatting, it is not worth doing some laborious work for a simple formatted EXCEL output. Stepping to SAS 9 and later versions, SAS has come up with EXCEL Libname and ODS Markup, a middle ground between the plain data dump using PROC EXPORT and highly laborious DDE. The goal of this paper is to discuss these two methods that are available for SAS 9 and later versions, which are fairly simple to get the variable labels as column headers in EXCEL outputs.

EXCEL LIBNAME

Excel Libname is better than a simple PROC EXPORT & other methods available in the following ways:

- 1) It can be done with a simple data step and a Libname statement.
- 2) Variable Labels can be generated as column headers.
- 3) Multiple sheets can be generated dynamically in one single spread sheet.
- 4) Can avoid formatting issues with a smile SQL code dynamically generated if needed?

GETTING STARTED

Test Data Used for Demonstration

```
data Test;
  attrib name length=$10 label='Name of the Subject'
        gender length= $8 label='Gender at the time of Birth';
  input name $ gender $;
  cards;
  Tom Male
  Mary Female
  George Male
  Christine Female
  Jim Male
  Beth Female
  ;
run;
```

THE BASIC EXCEL LIBNAME SYNTAX

LIBNAME libref <engine-name> <physical-file-name> libname-options>;

<SAS Code>

LIBNAME libref CLEAR;

The code below will generate variable names as column headers using Excel LIBNAME and a simple DATA STEP with DBLABEL dataset option.

```
Libname myxlbook 'C:\Path\Libname.xls';

*To Replace the Sheet With new data existing data must be deleted;
proc datasets lib=myxlbook;
    delete Names Gender;
run;
quit;

*Sheet-1 (Name of the sheet-Names);
data myxlbook.Names(dblabel=yes);
    set test;
run;

*Sheet-2 (Name of the sheet-Gender);
data myxlbook.Gender(dblabel=yes);
    set test;
run;

*Indicate the sheet in the sheet in
```

The output generated using the above code is shown is Figure 1.

DBLABEL=YES

During output to Excel, SAS variable labels, instead of variable names, are written out to column headers by changing to the data set option "DBLABEL=YES". All labels written in this way must be unique. This allows long variable labels with spaces and special characters to be written from SAS into Excel column headers (Explore more about the other dataset options).

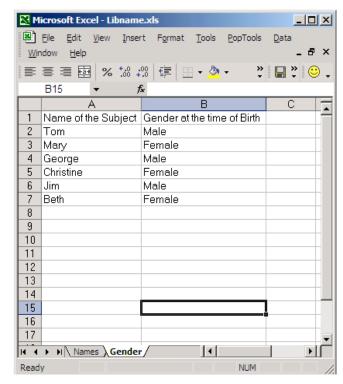


Figure 1: Output using Excel Libname engine

To replace the same excel output with new data, one need to delete the existing data and rewrite file with new data (for this PROC DATASETS procedure could be used).

```
proc datasets lib=myxlbook;
   delete Names Gender;
run;
quit;
```

The above code will delete the data present in the each sheet that has to be replaced with the new data. The log will refer the deleted sheets as sheet-name\$ as shown in the figure 2.

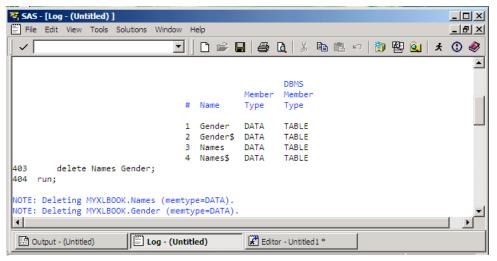


Figure 2: Log Generated while deleting the data in the excel work book

THINGS TO REMEMBER

When writing to Excel, workbook should not be open.

Delete old data before rewriting the workbook.

Variables in the SAS dataset will be exported in the same order.

ODS TAGSETS (supported by Microsoft Excel (Version 2002 and later).

Output Delivery System with Tagsets generates XML outputs; Tagsets with ExcelXP generates XML output that can be opened in Excel. ODS Tagsets is a better way to generate multiple sheets with more of customizing options.

- 1. No formatting issues loss and change of data during transformation of SAS datasets to Excel, since the data is preserved as xml file.
- Customizing outputs by Adding filters to different fields lock the headers, dimensions of each cell, and adding titles footnotes in the same spreadsheet.
- 3. Using different layouts customizing the template using PROC TEMPLATE.
- 4. Create Multiple Excel Sheets.
- 5. Traffic lighting to pick issues and also look at the results in one glance.

THE BASIC ODS TAGSETS SYNTAX

ods listing close;

ods tagsets.ExcelXP path='PATH' file='file-name.xls' style= style-name <Tagset options>;

<SAS Code - Reporting Procedures>

ods tagsets.ExcelXP close;

ods listing;

Using the test data, the code below generates Excel output shown in figure3, Generally any reporting procedure like PROC PRINT, PROC REPORT, PROC TABULATE can be used in ODS tagsets. But to get the variable labels as column headers PROC PRINT with split option will generate the desired outcome.

```
ods listing close;
ods tagsets.ExcelXP path="C:\Path" file="ods.xls" style=sansPrinter
options (autofilter='all'
embedded titles ='yes' autofit height = 'yes' frozen headers='yes'
frozen headers='3');
ods tagsets. ExcelXP options (sheet name='Subject Names' absolute column width='10,11'
);
      proc print data = Test split= ' ';
             var name gender;
       run;
ods tagsets. ExcelXP options (sheet name='Gender' absolute column width='10,11');
      proc print data = Test split= ' ';
       var gender;
       run;
ods tagsets. ExcelXP close;
ods listing;
```

LIMITATION

- It is an xml file, which will take a round of procedure to import it back to SAS datasets once the spread sheets are generated.
- Older versions than MS Excel 2002 cannot handle xml Extension files.

For those who are using SAS version 9.2, variable labels as headers can be achieved quite simply by using the label statement in the PROC EXPORT.

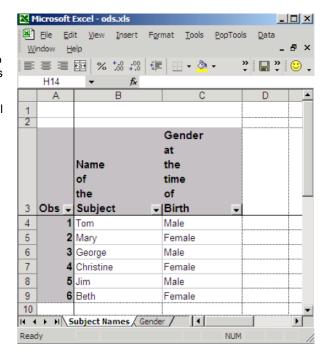


Figure 3: Output generated by ODS TAGSETS

CONCLUSION

While using SAS 9 and later versions, the EXCEL LIBNAME and ODS TAGSETS discussed in this paper greatly simplifies the task of generating multiple excel sheets and customized reports with variables labels as column headers. The ODS Tagsets have additional advantages that includes traffic lighting of the data and advanced customizing options, while EXCEL libname provides familiar and simpler SAS syntax and the usual SAS error log reporting. However, if the SAS version 9.2 is used, the label statement could be used to generate similar outputs.

REFERENCES

SAS 9.1.3 XP Platform SAS Institute Inc., Cary, NC

SAS OnlineDoc 9.1.3 for the Web SAS Institute Inc., Cary, NC

SAS/ACCESS 9.1 Interface to PC Files: Reference

Chapter 2 - The LIBNAME Statement for PC Files on Windows

DelGobbo, V. 2007. "Creating and importing multi-sheet excel workbooks the easy way with SAS". *Proceedings of the SAS Global Forum 31*, Paper 115. Available http://www2.sas.com/proceedings/sugi31/115-31.pdf

DelGobbo, V. 2004, "From SAS® to Excel via XML",

Available http://support.sas.com/rnd/papers/sugi29/ExcelXML.pdf

ACKNOWLEDGMENTS

The author would like to thank Mr. Carey Smoak & Mr. Mario Widel, Roche Molecular Systems and the management of MaxisIT Inc. for their support and guidance.

CONTACT INFORMATION

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

Chaitanya Chowdagam MaxisIT Inc., 203 Main Street, Metuchen, NJ 08840 (516)-448-3918 chowdagam@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.

Other brand and product names are trademarks of their respective companies.