

Paper RV-05

Printable Spreadsheets Made Easy: Utilizing the SAS® Excel XP Tagset

Rick Andrews, Office of the Actuary, CMS, Baltimore, MD

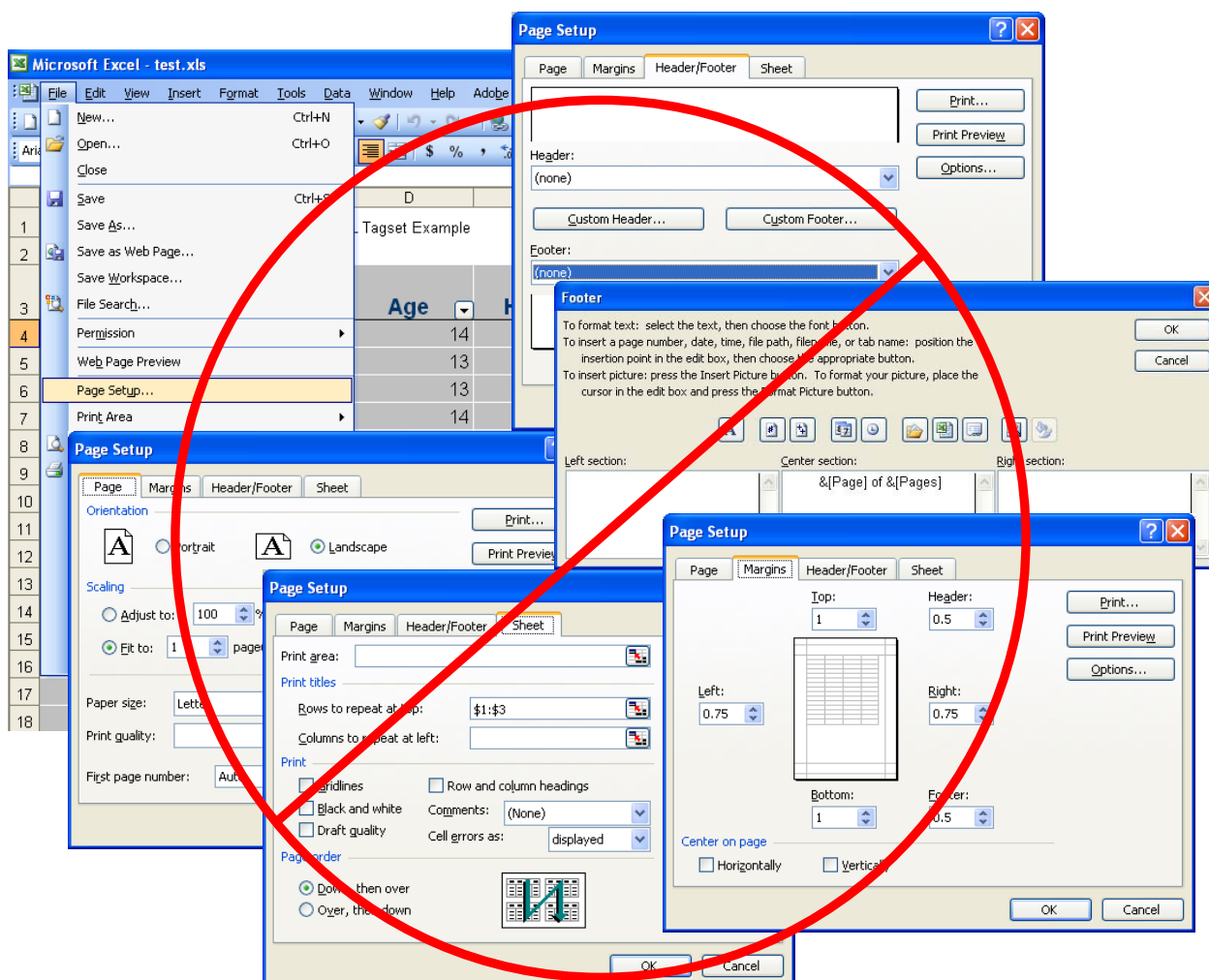
ABSTRACT

The SAS System offers myriad techniques for exchanging data to and from Microsoft Excel. Depending on the task at hand SAS Access® or the Output Delivery System (ODS) might be good choices; Dynamic Data Exchange (DDE) or the old standby, Comma-Separated Values (CSV). This paper describes a method of creating multi-tab, print-ready reports using the Excel XP tagset available in version 9.1. This feature of Base SAS can greatly minimize the manual and repetitious task of preparing headers, footers, and various other formatting needs.

INTRODUCTION

The tagset utilizes the eXtended Markup Language (XML); an open standard for the definition, transmission, validation, and interpretation of data. The standard was developed by the Worldwide Web Consortium (W3C) to provide an efficient way to manage self-documenting data files (ref, 200_). Knowledge of XML is not required to use the Excel XP tagset. The SAS code necessary is very similar to most other ODS processes and only a handful of options are needed to create spreadsheets ready for publication.

The advantage of using the tagset is to reduce formatting time. Configuring a spreadsheet once may take only a few moments, though if the same spreadsheet needs changing repeatedly, or if a similar document needs creating for multiple iterations, the formatting can be come very cumbersome.



REQUIREMENTS

The techniques presented in this paper utilize technologies implemented in Base SAS 9.1 or later, on any supported operating system and hardware, and Microsoft® Excel 2002 or later. The current version of the ExcelXP tagset has undergone various revisions since initial release. The latest tagset should be downloaded from the SAS Research & Development (R&D) website located at: <http://support.sas.com/rnd/base/ods/odsmarkup>

Notice the current version of the tagset at the writing of this paper is 1.86; see Figure 1, ExcelXP Tagset Download. Also located on this site are links to various examples, demos, and tutorials to give even the most novice individual a head-start with the exciting new world of markup capabilities.

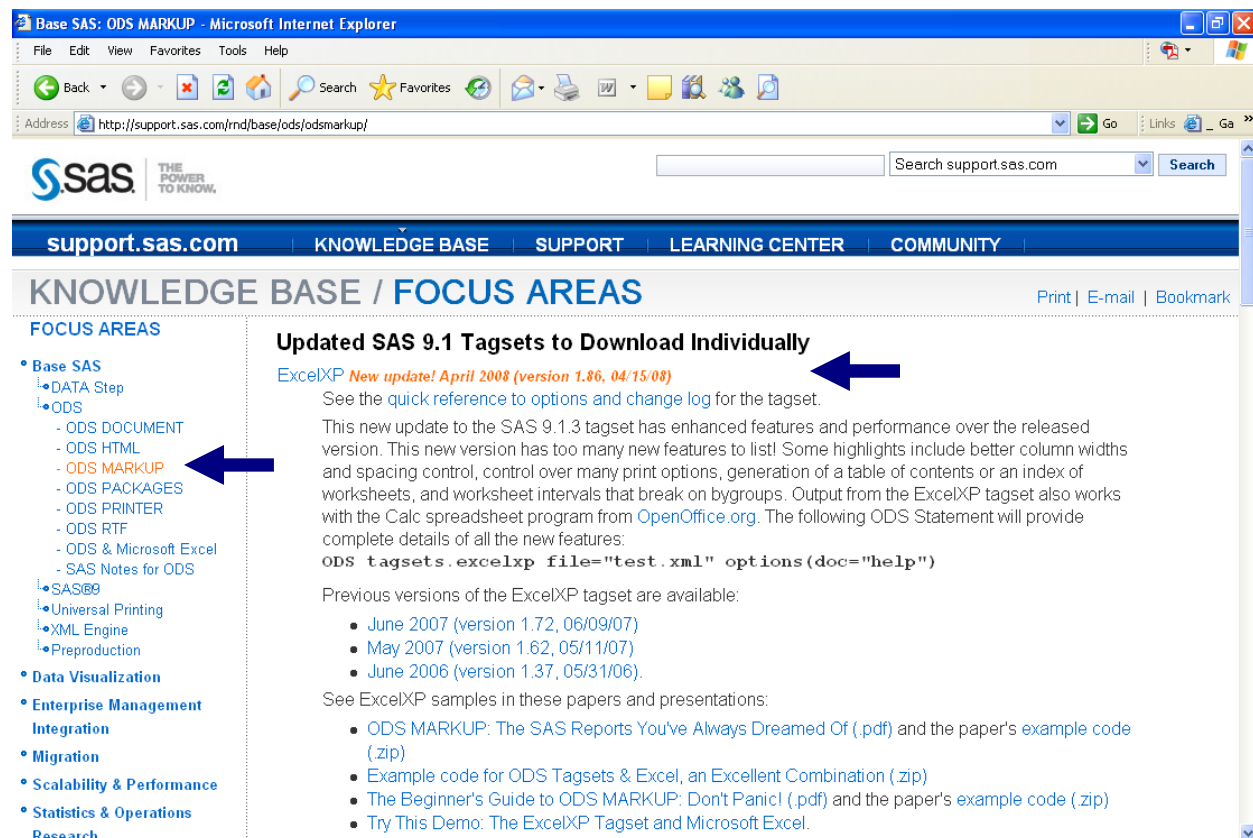


Figure 1: ExcelXP Tagset Download

UPDATING TAGSET

The ExcelXP tagset is created using the Template procedure, though absolutely no knowledge of the PROC is required. Copy and paste the code located at the website listed in Figure 1 into a SAS program editor and run it. This will create or update the current version in the SASHELP.TMPLMST catalog. The developers at SAS have done a very nice job of documenting the history and usage of the tagset directly within the code. The `log_note` shown below indicates the version and when it was updated.

```
proc template;
...
  define Tagset Tagsets.ExcelXP;
    parent = Tagsets.ExcelBase;
  end;
...
  log_note = "NOTE: This is the Excel XP tagset (SAS 9.1.3, v1.86, 04/15/08)";
...
run; #
```

COMPARISON

Creating Excel spreadsheets from SAS has been available since, well, forever. If a customer only need rows and columns of data a CSV file may be all that is necessary and no code required as there is a handy export wizard to guide through the process. If data need formatting in a special way DDE or ODS HTML can be used. It has been said that trying to compare DDE, ODS HTML, and the Excel XP tagset is like comparing apples and oranges (or comparing Excel XP apples from a tree growing in the backyard with DDE/HTML oranges from a tree in South America, where instead of picking the fruit from the branch, they need to be shipped via boat, train, and truck to the grocery store AND are still not available until purchased).

The end result is Excel, but the difference is HOW to get there. In this case, the journey is everything.

DDE

DDE technology uses the Excel 4 Macro (XLM) language, which is a predecessor to Visual Basic for Applications (VBA), Open Database Connectivity (ODBC) and Object Linking and Embedding Databases (OLE-DB).

PRO - This is currently the only method that can be used to perform very specific updating to individual cells within a new or existing spreadsheet.

CON - Both SAS and Excel need to be open for DDE commands to populate the Excel spreadsheet and the computer is tied up during processing.

ODS HTML

Opens, manages, or closes the Hypertext Markup Language (HTML) destination, which produces HTML 4.0 output that contains embedded stylesheets. Changing the HTML extension to XLS opens the page in Excel.

PRO - Allows the inclusion of SAS/GRAPH® images and can be created on any operating system.

CON - Cannot use this technique to update a single cell in an existing spreadsheet and there is not an options statement used in the HTML destination.

EXCEL XP TAGSET

Microsoft® announced the spreadsheet version of the eXtended Markup Language (XML) in Office 10 (Office 2000), which allows the definition of elements based on the needs of the task in lieu of hard-coded within the software.

PRO - Can execute in batch job from SAS end and Excel does NOT have to be open for this method to work.

CON - Cannot use this technique to update a single cell in an existing spreadsheet and does not support insertion of graphic images.

These PROs and CONs are not all-inclusive and advances in newer versions of Base SAS may have changed those presented here. They are given as examples of where one option may not provide the flexibility and another might be ease of use. The choice to use one method over the other truly depends on the situation.

The Excel XP tagset is a new technology and cannot be found in the SAS OnLineDoc®. This should not however be a significant consideration when deciding to use this technique. Much documentation has been provided directly in the tagset itself and the SAS Help Desk is very responsive to questions.

USAGE

In its simplest form, the Excel XP tagset is used very much like the ODS HTML destination. Figure 2: Excel XP Tagset vs. ODS HTML shows the differences in the syntax are the opening and closing of the destinations. The output is slightly different as the default STYLE for each is different and the default name of the tab created using Excel XP is "Table 1 - Data Set SASHELP.CLAS" denoting the SAS data set that was used versus HTML, which created a tab called "HTML_Example".

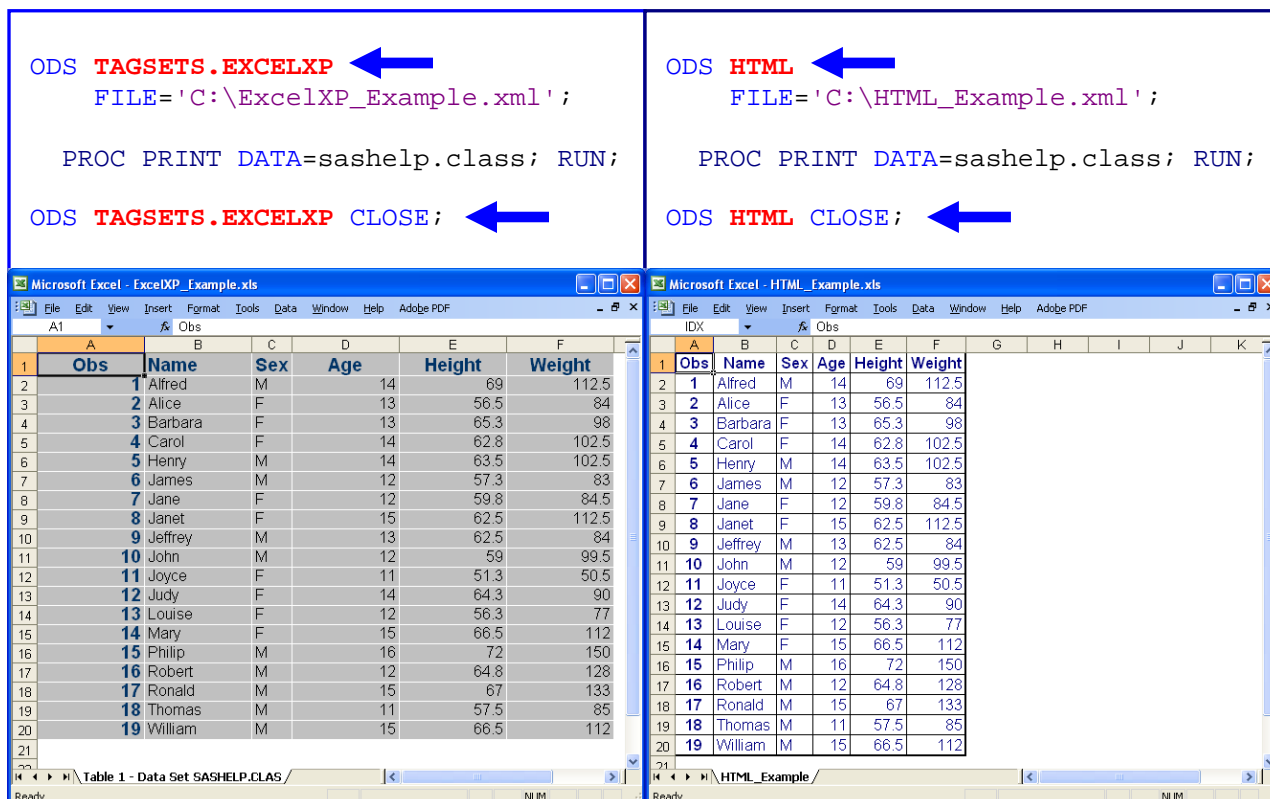


Figure 2: Excel XP Tagset vs. ODS HTML

This is where much of the similarity ends. Each can accept ODS options such as STYLE=minimal, though only the Excel XP tagset has an OPTIONS statement containing over fifty element used in formatting the spreadsheet. One of the most useful of these options is called Doc, which tells SAS to output the available help to the Log. Notice in Figure 3: ODS OPTIONS vs. Excel XP OPTIONS, the ODS STYLE option is used exactly as is would in the HTML destination, though an OPTIONS statement has been added that contains the Doc option with the 'Help' parameter surrounded by parentheses.

Below are the values available for the Doc option as obtained from the SAS Log:

Doc: No default value.

Help: Displays introductory text and available options in full detail.

Quick: Displays introductory text and an alphabetical list of options, their current value, and short description.

Settings: Displays config/debug settings.

Changelog: Lists the changes in reverse chronological order.

All: Shows the output from all the help options.

```

ODS TAGSETS.EXCELXP

FILE='C:\SESUG\ExcelXP_Example2.xml'
STYLE = minimal
OPTIONS ( Doc = 'Help' );

PROC PRINT DATA=sashelp.class; RUN;

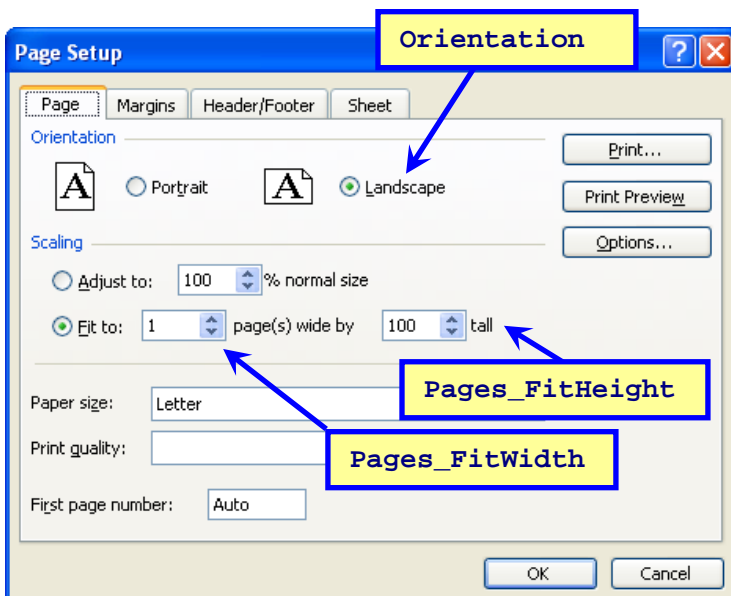
ODS TAGSETS.EXCELXP CLOSE;
    
```

Figure 3: ODS OPTIONS vs. Excel XP OPTIONS

PAGE SETUP

Getting familiar with the fifty-plus options available can be a daunting task. This paper describes only some of the possibilities and how they relate to the Excel spreadsheet. One of the great nuances of the Excel XP tagset is the ability to easily setup the printing options.

As shown in Figure 4: Page Setup Options, the options for setting the Orientation and Scaling are being displayed. This is a rather easy task to perform if only one file were being created, though if there are many files to setup the task could become rather cumbersome and prone to error. The code below signifies how to perform these options using the tagset.



```
ODS TAGSETS.EXCELXP
FILE='C:\SESUG\ExcelXP_Example3.xml'
STYLE=minimal
OPTIONS ( Orientation    = 'landscape'
          FitToPage      = 'yes'
          Pages_FitWidth  = '1'
          Pages_FitHeight = '100' );

PROC PRINT DATA=sashelp.class; RUN;

ODS TAGSETS.EXCELXP CLOSE;
```

Notice the Doc = 'Help' option was removed and replaced with Orientation, FitToPage, Pages_FitWidth, and the Pages_FitHeight. Some of the options like Orientation only have two possible parameters; landscape or portrait, while others like the Pages_FitHeight option is determined by necessity.

Also note the parameters and their values are separated by an equal sign and all options surrounded with parentheses.

Figure 4: Page Setup Options

MARGINS

The margins of the printable area of the spreadsheet are set using a SAS OPTIONS statement in lieu of an Excel XP option, as shown in Figure 5: Margins.

```
OPTIONS LeftMargin  = .5in
        RightMargin = .5in
        TopMargin   = .5in
        BottomMargin = .5in ;
```

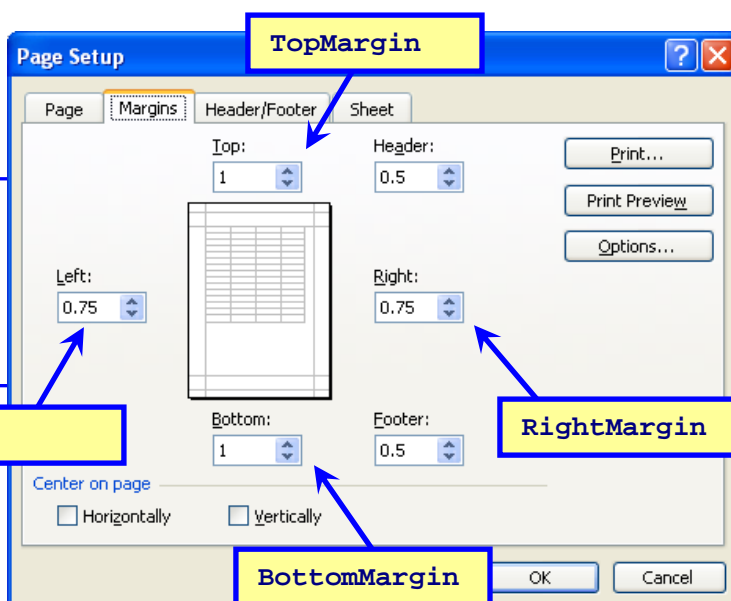


Figure 5: Margins

HEADERS & FOOTERS

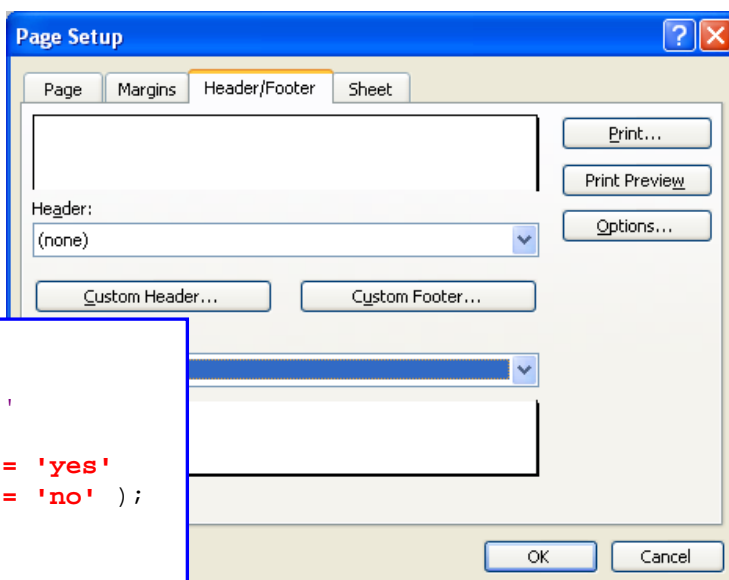
The headers and footers are set by the Title and Footnote statements respectively. Two Excel XP options exist to control whether a title or footnote is shown within the spreadsheet itself or in the printable section alone. The following code will create a file containing a title in the worksheet itself and a footnote only when printed.

```
ODS TAGSETS.EXCELXP
FILE='C:\ExcelXP_Example4.xml'
STYLE=minimal
OPTIONS ( Embedded_Titles    = 'yes'
          Embedded_Footnotes = 'no' );

TITLE1 'SAS is Great';
TITLE2 'SAS is Good';
FOOTNOTE1 'Let us Thank';
FOOTNOTE2 'Jim we Should';

PROC PRINT DATA=sashelp.class; RUN;

ODS TAGSETS.EXCELXP CLOSE;
```



Notice there is no footer in the worksheet as indicated in Figure 7: Worksheet with NO Footer and there is indeed a footer in Figure 8: Print Preview with Footer. This is useful when a title within the worksheet itself is needed, though a footnote only need exist in footer.

Figure 6: Titles & Footnotes

Obs	Name	Sex	Age	Height
1	Alfred	M	14	69
2	Alice	F	13	56.5
3	Barbara	F	13	65.3
4	Carol	F	14	62.8
5	Henry	M	14	63.5
6	James	M	12	57.3
7	Jane	F	12	59.8
8	Janet	F	15	62.5
9	Jeffrey	M	13	62.5
10	John	M	12	59
11	Joyce	F	11	51.3
12	Judy	F	14	64.3
13	Louise	F	12	56.3
14	Mary	F	15	66.5
15	Philip	M	16	72
16	Robert	M	12	64.8
17	Ronald	M	15	67
18	Thomas	M	11	57.5
19	William	M	15	66.5

Figure 7: Worksheet with NO Footer

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84
10	John	M	12	59	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90
13	Louise	F	12	56.3	77
14	Mary	F	15	66.5	112
15	Philip	M	16	72	150
16	Robert	M	12	64.8	128
17	Ronald	M	15	67	133
18	Thomas	M	11	57.5	85
19	William	M	15	66.5	112

Figure 8: Print Preview with Footer

PAGE of PAGES

If Embedded_Footnotes are on, the Print_Footer will be used as the footer for printing. Everything about the appearance of the footer can be controlled with this value. The easiest way to create a header or footer is to first do it in Excel, save the workbook as an XML Spreadsheet, open the file in Notepad, and search for <header or <footer. The exact syntax can then be used in SAS.

Below are some of the options listed in the Doc = 'Help' of the Excel XP tagset:

Newline: 
 Page Number: &P
 Pages: &N
 Date: &D
 Time: &T
 File Path: &Z&
 File: &F
 Sheet Name: &A
 Underline: &U
 Font Size: &8

Other options include changing the font and other characteristics such as bold and italic. The following will create a header or footnote that is left justified with an Arial font, is bold, italic and underlined.

Many possibilities exist to create almost any header or footnote that can be imagined. Though watch out for spaces between the "&" as this can cause unexpected results. It is usually best to create the desired result in SAS first and view the XML syntax as described above.

Notice in Figure 9: Page of Pages Syntax the FOOTNOTE statements have been removed. If they had not the footnotes would appear twice; once at the end of each report and another at the end of each page.

This can be a bit confusing at first. To completely understand the result, create a report with and without both the FOOTNOTE statement and the Print_Footer option to get a feel of the correct syntax.

The screenshot shows the 'Footer' dialog box in SAS. It has three sections: Left section, Center section, and Right section. The Center section contains the text '&[Page] of &[Pages]'. Below the dialog box, the SAS code is displayed:

```
TITLE; FOOTNOTE;

ODS TAGSETS.EXCELXP
FILE='C:\SESUG\ExcelXP_Example5.xml'
STYLE=minimal
OPTIONS ( Center_Horizontal = 'yes'
          Embedded_Titles    = 'yes'
          Embedded_Footnotes = 'yes'
          Print_Footer       = 'Let us Thank
          &#13; Jim we Should &#13; Page: &P of Pages:
          &N' );

TITLE1 'SAS is Good';
TITLE2 'SAS is Great';

PROC PRINT DATA=sashelp.class; RUN;

ODS TAGSETS.EXCELXP CLOSE;
```

Figure 9: Page of Pages Syntax

The screenshot shows the 'Microsoft Excel - ExcelXP_Example5.xls' preview window. It displays the output of the SAS code, including the titles, the data table, and the footer. The footer text is 'Let us Thank Jim we Should Page: 1 of Pages: 1'. A blue arrow points to the footer text.

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69	112.5
2	Alice	F	13	56.5	84
3	Barbara	F	13	65.3	98
18	Thomas	M	11	57.5	85
19	William	M	15	66.5	112

Figure 10: Page of Pages Preview

ROW REPEAT

Another useful feature of the Excel XP tagset is the Row Repeat option. This will identify the rows to be repeated when the report is printed. The syntax for this option is located below.

Row_Repeat

Column_Repeat

```
ODS TAGSETS.EXCELXP
FILE='C:\SESUG\ExcelXP_Example7.xml'
STYLE=minimal
OPTIONS ( Embedded_Titles = 'yes'
          Row_Repeat      = '1-3' );

TITLE 'Row Repeat Option';

PROC PRINT DATA=sashelp.air; RUN;

ODS TAGSETS.EXCELXP CLOSE;
```

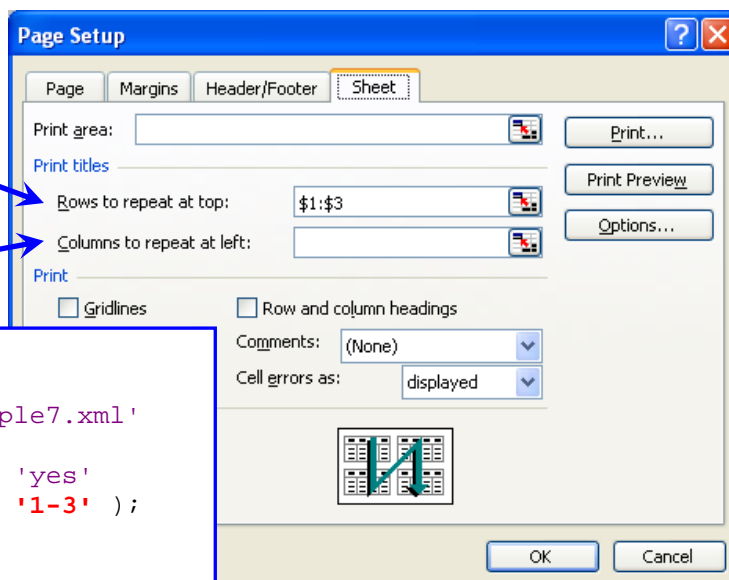


Figure 11: Row Repeat

MULTIPLE WORKSHEETS

One of the most useful aspects of the tagset is the ability to create multiple tabs or worksheets within the same workbook. In this example there are two worksheets being created and given the names of Shoes and Class respectively. This is accomplished using the Sheet_Name option.

The key is to open the Excel XP tagset, then using the Sheet_Name option give the first report a name. Then before closing the destination provide SAS with another ODS TAGSETS.ExcelXP destination and provide a second sheet name.

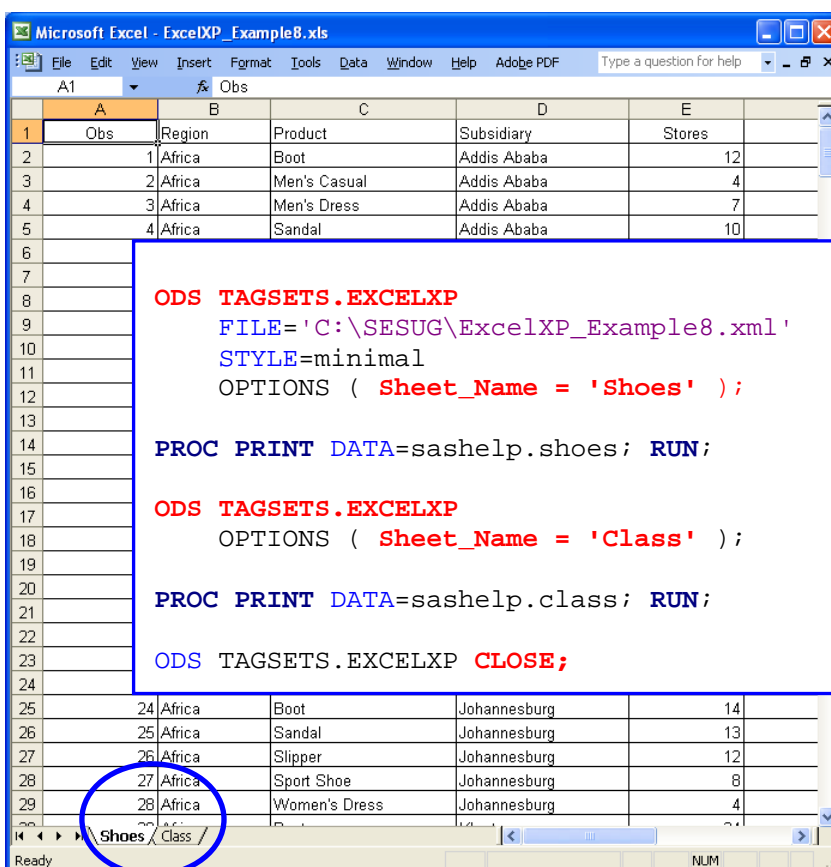


Figure 12: Multiple Worksheets

SHEET INTERVAL

If the desired result is to have both reports on the same worksheet the Sheet_Interval option can be used without opening the Excel XP tagset again.

Other parameters for this option include Table, Page, Bygroup, and Proc. See the Doc = 'Help' for more information.

```
ODS TAGSETS.EXCELXP
FILE='C:\SESUG\ExcelXP_Example9.xml'
STYLE=minimal
OPTIONS( Sheet_Name      = 'Shoes & Class'
        Sheet_Interval = 'none' );

PROC PRINT DATA=sashelp.shoes; RUN;
PROC PRINT DATA=sashelp.class; RUN;

ODS TAGSETS.EXCELXP CLOSE;
```

Figure 13: Two Reports on Same Worksheet

EXCEL vs. XML

As previously mentioned, the Excel XP Tagset creates an XML document that can be read directly into Excel. A disadvantage in using XML is the size of the file that is created. An XML document can easily become 80 to 90 percent larger than a true Excel file in binary format.

If file space is an issue, the XML document can simply be opened in Excel and saved as a true workbook. If automation is a consideration there is a way to open Excel in a batch mode using SAS without having to open the XML document. The key is to create a temporary Visual Basic Script (VBS) and simply execute it.

The steps are: **1)** Create an Excel object; **2)** Tell the system not to make Excel visible (hence batch mode); **3)** Tell the system not to display an alerts; **4)** Open the XML document; **5)** Save the document as a true Excel file; **6)** Close the workbook; and **7)** Quit the Excel application.

```
1. Set objExcel = CreateObject("Excel.Application")
2. objExcel.Visible = FALSE
3. objExcel.DisplayAlerts = FALSE
4. Set objWorkbook = objExcel.Workbooks.Open("C:\SESUG\ExcelXP_Example9.xml")
5. objExcel.ActiveWorkbook.SaveAs "C:\SESUG\ExcelXP_Example9.xlsx", 51
6. objExcel.ActiveWorkbook.Close
7. objExcel.Quit
```

In the example above, an XLSX file was created, though others are available as well.

51 = xlsx	- xlOpenXMLWorkbook	(without macro's)
52 = xism	- xlOpenXMLWorkbookMacroEnabled	(with macro's)
50 = xlsb	- xlExcel12	(Binary Workbook)
56 = xls	- xlExcel8	(97-2003 format)

```
DATA _NULL_;
FILE 'C:\userfiles\ExcelBatch.vbs';
PUT 'Set objExcel = CreateObject("Excel.Application")';
PUT 'objExcel.Visible = FALSE';
PUT 'objExcel.DisplayAlerts = FALSE';
PUT 'Set objWorkbook = objExcel.Workbooks.Open("C:\userfiles\ExcelBatch.xml")';
PUT 'objExcel.ActiveWorkbook.SaveAs"C:\userfiles\ExcelBatch.xlsx",51';
PUT 'objExcel.ActiveWorkbook.Close';
PUT 'objExcel.Quit';
RUN;

X "C:\userfiles\ExcelBatch.vbs";      * Execute VB Script *;
X del "C:\userfiles\ExcelBatch.vbs";  * Delete VB Script *;
X del "C:\userfiles\ExcelBatch.xml";  * Delete XML File *;
```

Figure 14: Visual Basic Script to save XML to XLS

CONCLUSION

Many other options exist such as turning on the Auto Filter and Freezing Headers and Columns. Other options include, though are not limited to, adding formats to the data, creating drill-downs, and writing formulas in Excel. There will always be a need for the DDE, ODS HTML, and other methods of exporting data to Excel and now with the dynamic new ODS Excel XP tagset even more opportunities exist for create print-ready reports.

REFERENCES

#

DelGobbo, V.. "Techniques for SAS® Enabling Microsoft Office in a Cross-Platform Environment". SUGI (2002), <http://www2.sas.com/proceedings/sugi27/p174-27.pdf>.

DelGobbo, V. "A Beginner's Guide to Incorporating SAS® Output in Microsoft Office Applications". SUGI (2003). <http://www2.sas.com/proceedings/sugi28/052-28.pdf>.

DelGobbo, V.. "Creating AND Importing Multi-Sheet Excel Workbooks the Easy Way with SAS® ". SUGI (2006), <http://www2.sas.com/proceedings/sugi31/115-31.pdf>.

Gebhart, Eric S. "ODS MARKUP: The SAS® Reports You've Always Dreamed Of," SUGI (2004), <http://www2.sas.com/proceedings/sugi30/085-30.pdf>.

Gebhart, Eric S. "The Devil Is in the Details: Styles, Tips, and Tricks That Make Your Microsoft Excel Output Look Great!," , SGF (2008), <http://www2.sas.com/proceedings/forum2008/036-2008.pdf>.

Cisternas, Miriam and Ricardo Cisternas. "Reading and Writing XML files from SAS®," SUGI 29 (2003), <http://www2.sas.com/proceedings/sugi29/119-29.pdf>.

Friebel, Anthony. "XML? We do that!" SUGI 28 (2002), <http://www2.sas.com/proceedings/sugi28/173-28.pdf>.

Hoyle, Larry. "Reading Microsoft Word XML files with SAS®," SUGI 31 (2005), <http://www2.sas.com/proceedings/sugi31/019-31.pdf>.

Palmer, Michael. "XML in the DATA Step," SUGI 29 (2003), <http://www2.sas.com/proceedings/sugi28/025-28.pdf>.

Pratter, Frederick. "Beyond HTML: Using the SAS System® Version 8.2 with XML," SUGI 27 (2001), <http://www2.sas.com/proceedings/sugi27/p002-27.pdf>.

Pratter, Frederick. "Using the SAS® Output Delivery System and PROC TEMPLATE to Create XHTML Files," SAS Global Forum 2007, <http://www2.sas.com/proceedings/forum2007/118-2007.pdf>.

Shoemaker, Jack N. "XML Primer for SAS® Programmers," SUGI 30 (2004), <http://www2.sas.com/proceedings/sugi30/240-30.pdf>.

Bodt, M. "Talking to PC Applications Using Dynamic Data Exchange". *Observations™*, vol. 5, no. 3, pp. 18–27, 1996.

Parker, C. "Generating Custom Excel Spreadsheets using ODS". *Proceedings of the Twenty-Eighth Annual SAS Users Group International Conference*, paper 12, 2003.

Roper, C. A. "Intelligently Launching Microsoft Excel from SAS, using SCL functions ported to Base SAS". *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference*, paper 97, 2000a.

Roper, C. A. "Using SAS and DDE to execute VBA macros in Microsoft Excel". *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference*, paper 98, 2000b.

Schreier, H. "Getting Started with Dynamic Data Exchange". *Proceedings of the Sixth Annual Southeastern SAS Users Group Conference*, pp. 207–215, 1998.

Stetz, M. "Using SAS Software and Visual Basic for Applications to Automate Tasks in Microsoft Word: an Alternative to Dynamic Data Exchange". *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference*, paper 21, 2000.

Vyverman, K. "Using Dynamic Data Exchange to Pour SAS Data into Microsoft Excel".
Proceedings of the Twenty-Sixth Annual SAS Users Group International Conference, paper 11, 2001.

SAS Institute Inc. 2004. "Chapter 9: TEMPLATE Procedure: Creating a Style Definition". SAS 9.1 Output Delivery System, User's Guide. Cary, NC: SAS Institute Inc.

SAS Institute Inc. "ODS MARKUP Resources", <http://support.sas.com/rnd/base/topics/odsmarkup/>.

SAS Institute, "Technical Support Document #325 – The SAS System and DDE".
<http://ftp.sas.com/techsup/download/technote/ts325.pdf>, updated 1999.

SAS Institute, "Technical Support Document #589b – Use PC-SAS 8.x and Higher to Read and Write Excel Files".
<http://ftp.sas.com/techsup/download/technote/ts325.pdf>, updated 2004.

Wikipedia, "Dynamic Data Exchange",
http://en.wikipedia.org/wiki/Dynamic_Data_Exchange, updated 2004.

Microsoft Corporation. "XML Spreadsheet Reference".
Available [http://msdn2.microsoft.com/en-us/library/aa140066\(office.10\).aspx](http://msdn2.microsoft.com/en-us/library/aa140066(office.10).aspx).

ACKNOWLEDGMENTS

The author wishes to thank the following, the SAS Help Desk, Vincent DelGobbo, David Kelley, Eric Gebhart, and Cynthia Zender for their tireless efforts to answer seemingly endless questions.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Rick Andrews

Office of the Actuary
Centers for Medicare and Medicaid Services
7500 Security Boulevard
Baltimore, MD 21244
Phone: (410) 786-6395
E-mail: Richard.Andrews2@cms.hhs.gov

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.