

Paper 102-28

Another Shot at the Holy Grail: Using SAS® to Create Highly-Customized Excel Workbooks

Ted Conway, Ted Conway Consulting, Inc., Chicago, IL

ABSTRACT

A different approach to the age-old problem of creating highly-customized Excel workbooks in an automated fashion is presented in this paper. The technique departs somewhat from more widely documented approaches in that it does not employ DDE, ODS, EXPORT, or SAS2XCEL and requires only Base SAS. This technique may be of interest to all skill levels and uses Base SAS on the PC.

INTRODUCTION

Over the years, many different solutions to the problem of getting data from SAS into Excel have been developed and presented at SUGI and other forums.

Things have progressed a great deal from the early days of the SAS2XCEL macro and PROC EXPORT, which provided welcome but rudimentary ways to essentially just dump data into an Excel worksheet.

As time went on, a DDE interface was provided that allowed developers to exercise some real control over worksheet creation and formatting. Indeed, this technique continues to be used by many to this day, despite the drawbacks of waning support from Microsoft and a dwindling knowledge base.

Most recently, SAS introduced ODS, which does a pretty good job of creating a nice looking Excel worksheet without too much effort.

THE PROBLEM

The above approaches will probably provide reasonable and cost-effective solutions for most of your needs. However, it's very likely that you'll eventually run into situations where your business units' reporting demands will exceed the capabilities of these techniques, much in the same way that one sometimes has to depart from a "vanilla" PROC REPORT or TABULATE solution.

Faced with precise Excel formatting requirements involving scores or even hundreds of fields, many simply throw up their hands and give up, resorting to time-consuming and error-prone manual reformatting - even daily - to meet their clients' needs.

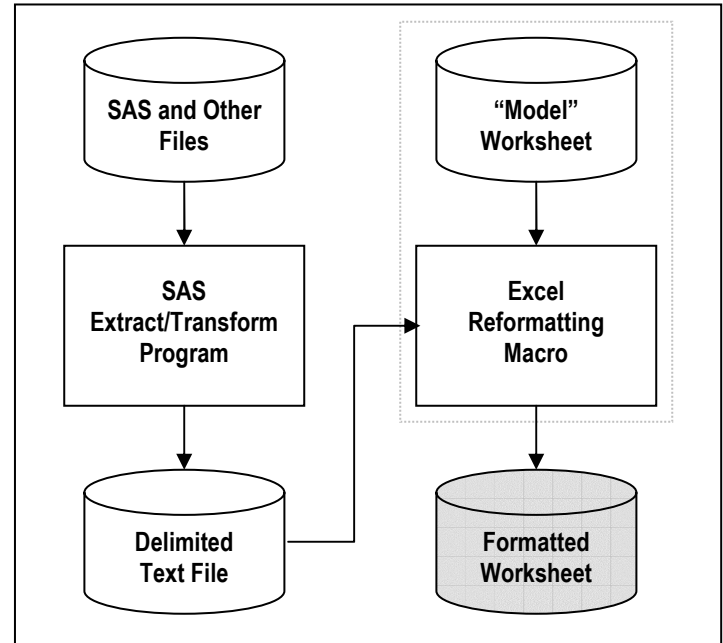
However, there is another way.

A DIFFERENT APPROACH

With apologies to the *Book of Luke*, we're simply going to *Render therefore unto SAS the things which be SAS*, and unto Excel the things which be Excel's.

Now what exactly does *that* mean?

What it simply means is that - for complicated requirements - one is often better off letting SAS take care of the things it is best at - e.g., analyzing, restructuring and summarizing data - and letting Excel take care of the things it is best suited for - i.e., worksheet-specific formatting!

PROCESSING OVERVIEW

1. Write a **SAS Extract/Transform program** that analyzes, restructures, and summarizes **SAS or other input data** into the desired results, writes out a **delimited text file** and launches Excel.
2. Create a **"model" Excel worksheet** with the desired formats, fonts, borders, etc.
3. Record and manually "tweak" an **Excel reformatting macro** that will be run automatically when the "model" worksheet is opened.

Requirements will vary with each worksheet, but a typical Excel reformatting macro:

- Reads the delimited text file
- Copies the "model" worksheet formats
- Auto fits and resizes columns
- Sets print parameters
- Freezes panes
- Saves the **formatted worksheet**
- Closes the workbook and Excel
- Returns to SAS for any subsequent processing (e.g., e-Mail, FTP, etc.)

A WORKING EXAMPLE

The following sections illustrate the aforementioned techniques using the **SASHELP.SHOES** dataset provided with SAS LE.

ShoesToExcel.sas – SAS Program

```

*--> Summarize Data To Yield Desired Totals;

proc tabulate data=sashelp.shoes out=_data_;
class region subsidiary;
var stores sales returns;
tables (region*(subsidiary all), stores sales returns);

*--> Sort Summarized Results In Desired Sequence;

data;
set;
regionseq=substr(_type_,1,1);
subsidiaryseq=substr(_type_,2,1);
proc sort;
by descending regionseq region descending subsidiaryseq subsidiary;

*--> Write Summarized Results To A Delimited Text File For Excel;

data _null_;
set;
by descending regionseq region;
file 'c:\windows\temp\shoes.txt' noprint notitles;
if _n_=1 then
  put '|Region|Subsidiary|Stores|Sales|Returns';
put _type_ +(-1) '|@';
if first.region then
  put region@;
else
  put ' '|@;
put (subsidiary stores_sum sales_sum returns_sum)+(-1) '|';
run;

*--> Launch Excel To Create & Review Worksheet;

options xsync noxwait;
x erase c:\windows\temp\shoes.xls;
x c:\shoesmodel.xls;
x c:\windows\temp\shoes.xls;

```

ShoesModel.xls – “Model” Excel Worksheet

	A	B	C	D	E	F
1	type	Region	Subsidiary	Stores	Sales	Returns
2		XXXXXXXXXX	XXXXXXXXXX	1,234,567	\$1,234,567.00	\$1,234,567.00

Workbook_Open – Excel Open Macro

```

Private Sub Workbook_Open()

'--> Generate Worksheet From Reformatted Text File Created By SAS

GenerateWorksheet

End Sub

```

GenerateWorksheet – Excel Reformatting Macro

```

Sub GenerateWorksheet()

'--> Read Delimited Text File

Workbooks.OpenText Filename="C:\windows\temp\shoes.txt",
Origin:=xlWindows, _
  StartRow:=1, DataType:=xlDelimited, TextQualifier:=xlNone, _
  ConsecutiveDelimiter:=False, Tab:=False, Semicolon:=False, Comma:=False, _
  Space:=False, Other:=True, OtherChar=""

'--> Copy Formats From "Model" Worksheet

Windows("shoesmodel.xls").Activate
Cells.Select
Range(Selection, ActiveCell.SpecialCells(xlLastCell)).Select
Selection.Copy
Windows("shoes.txt").Activate
Cells.Select
Selection.PasteSpecial Paste:=xlFormats, Operation:=xlNone, SkipBlanks:= _
  False, Transpose:=False
Application.CutCopyMode = False

'--> Highlight Subtotals (Grey) And Final Totals (Black)

lastrow = ActiveCell.SpecialCells(xlLastCell).Row
For r = 2 To lastrow
  If Cells(r, 1).Value = 0 Then
    Range(Cells(r, 2), Cells(r, 6)).Select
    Selection.Interior.ColorIndex = 1
    Selection.Font.Bold = True
    Selection.Font.ColorIndex = 2
  End If
  If Cells(r, 1).Value = 10 Then
    Range(Cells(r, 3), Cells(r, 6)).Select
    Selection.Interior.ColorIndex = 15
    Selection.Font.Bold = True
  End If
Next
Cells(1, 1).EntireColumn.Delete

'--> Autofit Columns And Make All Numeric Columns The Same Width

Cells.EntireColumn.AutoFit
maxwidth = 0
For c = 3 To 5
  If Cells(1, c).ColumnWidth > maxwidth Then maxwidth = Cells(1, c).ColumnWidth
Next
Range("c1:e1").ColumnWidth = maxwidth

'--> Set Print Attributes

ActiveSheet.PageSetup.PrintTitleRows = "$1:$1"
ActiveSheet.PageSetup.CenterHeader = "Shoes Example"
ActiveSheet.PageSetup.CenterHorizontally = True
ActiveSheet.PageSetup.Orientation = xlPortrait
ActiveSheet.PageSetup.PaperSize = xlPaperLetter

'--> Freeze Panes And Update Worksheet Name

Range("A2").Select
ActiveWindow.FreezePanes = True
ActiveSheet.Name = "Shoes Example"

'--> Save Reformatted Text File As An Excel Workbook

ActiveWorkbook.SaveAs Filename="C:\windows\temp\Shoes.xls",
FileFormat:=xlNormal, _
  Password="", WriteResPassword="", ReadOnlyRecommended:=False, _
  CreateBackup:=False

'--> Close Workbook And Excel To Return To SAS

ActiveWindow.Close
Application.SendKeys "%{F4}", True

End Sub

```

Shoes.xls – Formatted Worksheet

	A	B	C	D	E
1	Region	Subsidiary	Stores	Sales	Returns
50	United States	Chicago	150	\$1,565,585.00	\$57,523.00
51		Los Angeles	92	\$737,312.00	\$25,312.00
52		Minneapolis	141	\$1,099,937.00	\$37,073.00
53		New York	131	\$1,489,207.00	\$46,701.00
54		Seattle	103	\$611,945.00	\$20,893.00
55			617	\$5,503,986.00	\$187,502.00
56	Western Europe	Copenhagen	52	\$693,116.00	\$24,523.00
57		Geneva	95	\$447,208.00	\$14,268.00
58		Heidelberg	105	\$967,739.00	\$36,038.00
59		Lisbon	99	\$898,345.00	\$30,431.00
60		London	113	\$762,009.00	\$27,410.00
61		Madrid	17	\$200,642.00	\$7,334.00
62		Paris	86	\$621,877.00	\$19,324.00
63		Rome	75	\$282,064.00	\$10,407.00
64			642	\$4,873,000.00	\$169,755.00
65			4,601	\$33,851,566.00	\$1,172,092.00

CONCLUSION

When it comes to getting the ultimate in spreadsheet formatting flexibility, 'thinking outside the box' (to get things inside the box!) can be as simple as letting SAS and Excel each do their own things.

CONTACT INFORMATION

Ted Conway currently works for Ted Conway Consulting, Inc. (guess how he got that job!) in Chicago, Illinois. He can be reached at tedconway@aol.com.

TRADEMARKS

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.