

Paper 035-31

Sur La Table:**Creating Microsoft Excel PivotTables in a Jiffy from SAS® Data**

Ted Conway, Chicago, IL

ABSTRACT

Creating good looking Excel Pivot Tables to hold your nicely summarized SAS data can be surprisingly labor intensive. This paper presents a technique and macro that combine Base SAS with Visual Basic Scripting to make it possible to fully automate the creation of custom formatted Excel Pivot Tables from SAS data sets.

INTRODUCTION

Looking for the best way to create Excel pivot tables automatically from SAS, Phil Mason and colleague Chris Brooks considered complex solutions such as ODS with MSO XML directives, straight XML, DDE, and so on before Brooks had his Eureka! moment.

The result? A very elegant, simple method which Mason was kind enough to post on SAS-L.

Newsgroups: comp.soft-sys.sas

From: woodstr...@GMAIL.COM (Phil Mason) - Find messages by this author

Date: Thu, 2 Jun 2005 09:49:14 +0100

Local: Thurs, Jun 2 2005 3:49 am

Subject: SASTip 133 - Create a pivot table from SAS

Daily Tip 133

02/06/2005 08:24:00

Create a pivot table from SAS

A colleague and I were looking at the best way to automatically create a pivot table in EXCEL automatically from SAS. We considered solutions such as ODS with MSO XML directives, straight XML, DDE, and so on - but these were all very complex. He finally came up with the following simple method.

We use a SAS program to create a spreadsheet and then call a Visual Basic Script. The Visual Basic Script does the following:

- open the spreadsheet
- add a new sheet for pivot table
- create a pivot table using wizard
- set the fields to be used in the table

The SAS program could be extended to make a macro which creates the VBS file. This could then make it parameter driven to work for all data.

SAS Program

* create EXCEL spreadsheet ;

```
proc export data=sashelp.class outfile="c:\sas\class.xls" dbms=excel;
quit;
```

* call VB script to make the pivot table ;

```
data _null_;
x 'c:\sas\pivot.vbs';
```

VB Script Program

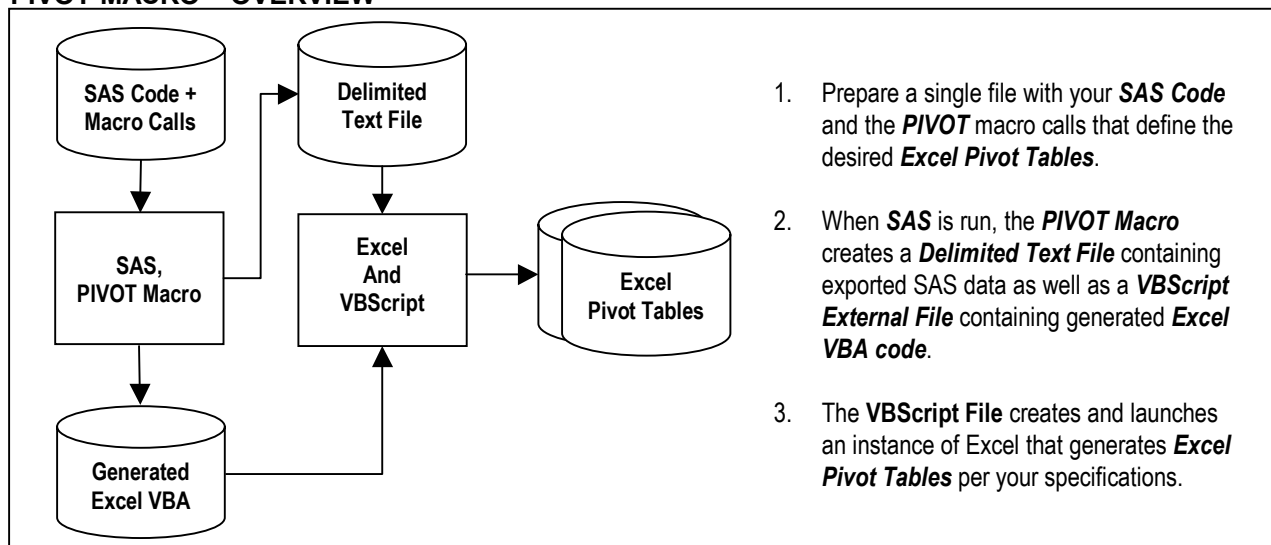
```
Set XL = CreateObject("Excel.Application")
XL.Visible=True
XL.Workbooks.Open "c:\sas\class.xls"
XLlastcell= xl.cells.specialcells(11).address
XL.Sheets.Add.name = "PivotTable"
xldata="class"
XL.Sheets(xldata).select
XL.ActiveSheet.PivotTableWizard SourceType=xlDatabase,XL.Range-"A1" & "." & xllastcell),"Pivottable!R1C1",-xldata
XL.ActiveSheet.PivotTables(xld-ata).PivotFields("Name").Orientation = 1
XL.ActiveSheet.PivotTables(xld-ata).PivotFields("Age").Orientation = 1
XL.ActiveSheet.PivotTables(xld-ata).PivotFields("Sex").Orientation = 1
XL.ActiveSheet.PivotTables(xld-ata).PivotFields("Height").Orientation = 4
XL.ActiveWorkbook.ShowPivotTableFieldList = False
```

Thanks to Chris Brooks, National Statistics, UK for this tip

Pretty clever, eh? I thought so! I also took note of Mason's suggestion: "The SAS program could be extended to make a macro which creates the VBS file. This could then make it parameter driven to work for all data."

Hey, I thought – that would make a great SUGI Paper! And that's a rather long-winded introduction to the **PIVOT** macro that's presented below!

PIVOT MACRO – OVERVIEW



PIVOT MACRO – SAMPLE USAGE

Before delving into the guts of the **PIVOT** macro, let's first take a look at how one might use it.

At the **workbook** level, one specifies a **file name** for the workbook that will be created as well as the **SAS data set** and **variables** that will provide the source of the data for the pivot tables.

At the **worksheet** level, one specifies a **name** for the worksheet (i.e., Excel tab) that will be created.

At the **field** level, one specifies a **field name** together with the field's **orientation** (**row**, **column**, or **data**), an optional Excel number **format**, and – for data fields – the type of statistical measure (**Sum**, **Average**, or **Count**).

Specify the **resize** option if you wish to autofit the pivot table's columns, and **create** when you're ready to generate the actual workbook. Easy enough, mais non?

```

%pivot(workbook, name=c:\temp\class.xls, data=sashelp.class, var=Name Age Sex Height);

%pivot(worksheet, name=TestPivotTable3);
%pivot(field, name=Sex, Orientation=Row);
%pivot(field, name=Age, Orientation=Row, Format="00");
%pivot(field, name=Height, Orientation=Data, Stat=Sum, Format="###0.00");
%pivot(resize);

%pivot(worksheet, name=TestPivotTable2);
%pivot(field, name=Sex, Orientation=Row);
%pivot(field, name=Height, Orientation=Data, Stat=Sum, Format="###0.00");
%pivot(field, name=Height, Orientation=Data, Stat=Average, Format="###0.0");
%pivot(field, name=Height, Orientation=Data, Stat=Count, Format="###0");
%pivot(resize);

%pivot(worksheet, name=TestPivotTable1);
%pivot(field, name=Sex, Orientation=Page);
%pivot(field, name=Name, Orientation=Row);
%pivot(field, name=Height, Orientation=Data, Stat=Sum, Format="###0.00");
%pivot(resize);

%pivot(create);
  
```

PIVOT MACRO – CODE

The **PIVOT** macro is certainly messier than the macro calls on the previous page, but at least the code still fits on one page!

After initially generating a text file from the SAS data, the **PIVOT** macro merely generates the VBScript code that's needed to effect whatever has been requested via the macro parameters.

```
%macro pivot(type, data=, var=, name=, orientation=, stat=, format=);

%if &type=workbook %then %do;
  %global gblworkbook;
  %let gblworkbook=&name;
  data _null_;
  set sashelp.class;
  file "c:\temp\RawData.txt";
  if _n_=1 then put "% sysfunc(translate(&var, '|', ' '))";
  put %scan(&var, 1) (%substr(&var, %index(&var, %str( ))) (+(-1) '|'));
  data _null_;
  file "c:\temp\class.vbs";
  put 'Set XL = CreateObject("Excel.Application") / 'XL.Visible=True' /
      'XL.Workbooks.OpenText "C:\temp\RawData.txt", 437, 1, 1, -4142, False, False, False, False, True, "|";
  %end;

%if &type=worksheet %then %do;
  put "XL.Sheets.Add.name = ""&name"" /
      "XL.ActiveSheet.PivotTableWizard SourceType=xlbase, xl.sheets("""RawData"").UsedRange, ""&name!R1C1"", ""pvttbl""";
  %end;

%if &type=field %then %do;
  %if &orientation=Data %then %do;
    put "XL.ActiveSheet.PivotTables("""pvttbl"").AddDataField XL.ActiveSheet.PivotTables("""pvttbl"").PivotFields("""&name""), ""&Stat of &name"" ,
    "
    %if &stat=Sum %then "-4157";
    %if &stat=Count %then "-4112";
    %if &stat=Average %then "-4106";
    %end;
  %else %do;
    put "XL.ActiveSheet.PivotTables("""pvttbl"").PivotFields("""&name"").Orientation = "
    %if &orientation=Page %then "3";
    %if &orientation=Row %then "1";
    %if &orientation=Column %then "2";
    %end;;
  %if &format^= %then
    %if &stat^= %then
      put "XL.ActiveSheet.PivotTables("""pvttbl"").PivotFields("""&stat of &name"").numberformat = ""&format"";
    %else
      put "XL.ActiveSheet.PivotTables("""pvttbl"").PivotFields("""&name"").numberformat = ""&format"";
    %end;
  %end;

%if &type=resize %then
  put "XL.ActiveSheet.Columns.AutoFit";

%if &type=create %then %do;
  put "XL.ActiveWorkbook.SaveAs ""&gblworkbook"", -4143";
  run;
  x 'c:\temp\class.vbs';
  %end;
%mend;
```

PIVOT MACRO – SAMPLE GENERATED VBSCRIPT

For our particular examples, the **PIVOT** macro generated the below VBScript code.

```
Set XL = CreateObject("Excel.Application")
XL.Visible=True
XL.Workbooks.OpenText "C:\temp\RawData.txt", 437, 1, 1, -4142, False, False, False, False, True, ""
XL.Sheets.Add.name = "TestPivotTable3"
XL.ActiveSheet.PivotTableWizard SourceType=xlbase, xl.sheets("RawData").UsedRange, "TestPivotTable3!R1C1", "pvttbl"
XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Sex").Orientation = 1
XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Age").Orientation = 1
XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Age").numberformat = "00"
XL.ActiveSheet.PivotTables("pvttbl").AddDataField XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Height"), "Sum of Height", -4157
XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Sum of Height").numberformat = "#,##0.00"
XL.ActiveSheet.Columns.AutoFit
XL.Sheets.Add.name = "TestPivotTable2"
XL.ActiveSheet.PivotTableWizard SourceType=xlbase, xl.sheets("RawData").UsedRange, "TestPivotTable2!R1C1", "pvttbl"
XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Sex").Orientation = 1
XL.ActiveSheet.PivotTables("pvttbl").AddDataField XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Height"), "Sum of Height", -4157
XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Sum of Height").numberformat = "#,##0.00"
XL.ActiveSheet.PivotTables("pvttbl").AddDataField XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Height"), "Average of Height", -4106
XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Average of Height").numberformat = "#,##0.0"
XL.ActiveSheet.PivotTables("pvttbl").AddDataField XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Height"), "Count of Height", -4112
XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Count of Height").numberformat = "#,##0"
XL.ActiveSheet.Columns.AutoFit
XL.Sheets.Add.name = "TestPivotTable1"
XL.ActiveSheet.PivotTableWizard SourceType=xlbase, xl.sheets("RawData").UsedRange, "TestPivotTable1!R1C1", "pvttbl"
XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Sex").Orientation = 3
XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Name").Orientation = 1
XL.ActiveSheet.PivotTables("pvttbl").AddDataField XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Height"), "Sum of Height", -4157
XL.ActiveSheet.PivotTables("pvttbl").PivotFields("Sum of Height").numberformat = "#,##0.00"
XL.ActiveSheet.Columns.AutoFit
XL.ActiveWorkbook.SaveAs "c:\temp\class.xls", -4143
```

Aren't you glad you didn't have to hand-code it?

PIVOT MACRO – SAMPLE OUTPUT (EXCEL PIVOT TABLES)

The Excel Pivot Tables and data worksheet created by the **PIVOT** macro example are shown in the following pages.

CONCLUSION

The **PIVOT** macro provides a way of creating Excel pivot tables automatically from SAS without having to resort to complex solutions such as ODS with MSO XML directives, straight XML, DDE, and so on.

For the purposes of this paper, the feature set of the **PIVOT** macro has been kept rather simple.

Feel free to extend it to include other Excel features – just remember to share your enhancements on SAS-L!

CONTACT INFORMATION

Ted Conway resides 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.

PIVOT MACRO – SAMPLE OUTPUT (EXCEL PIVOT TABLES)

Sex	(All)	
		Sum of Height
		Name Total
		Alfred 89.00
		Alice 56.50
		Barbara 85.30
		Carol 62.80
		Henry 63.50
		James 57.30
		Jane 59.80
		Janet 62.50
		Jeffrey 62.50
		John 59.00
		Joyce 51.30
		Judy 64.30
		Louise 56.30
		Mary 66.50
		Philip 72.00
		Robert 64.80
		Ronald 67.00
		Thomas 57.50
		William 66.50
		Grand Total 1,184.40

Sex	Data	Total
F	Sum of Height	545.30
	Average of Height	60.6
	Count of Height	9
M	Sum of Height	639.10
	Average of Height	63.9
	Count of Height	10
Total Sum of Height		1,184.40
Total Average of Height		62.3
Total Count of Height		19

PIVOT MACRO – SAMPLE OUTPUT (EXCEL PIVOT TABLES) – CONTINUED

The screenshot shows an Excel PivotTable titled "Sum of Height". The PivotTable is structured with "Sex" as the row field and "Age" as the column field. The values are summed. The Grand Total is 1,184.40.

Sex	Age	Total
F	11	51.30
	12	116.10
	13	121.80
	14	127.10
	15	129.00
F Total		545.30
M	11	57.50
	12	181.10
	13	62.50
	14	132.50
	15	133.50
M Total		639.10
Grand Total		1,184.40

The screenshot shows the source data for the PivotTable. The columns are Name, Age, Sex, and Height. The data is as follows:

Name	Age	Sex	Height
Alfred	14	M	69
Alice	13	F	56.5
Barbara	13	F	65.3
Carol	14	F	62.8
Henry	14	M	63.5
James	12	M	57.3
Jane	12	F	59.8
Janet	15	F	62.5
Jeffrey	13	M	62.5
John	12	M	59
Joyce	11	F	51.3
Judy	14	F	64.3
Louise	12	F	56.3
Mary	15	F	66.5
Philip	16	M	72
Robert	12	M	64.8
Ronald	15	M	67
Thomas	11	M	57.5
William	15	M	66.5