

## Writing to excel from SAS, DDE vs ODS

Robert Ellsworth, Ellsworth Stewart Consulting Inc.

### ABSTRACT

This paper will demonstrate 4 methods for populating excel report. The paper will compare methods using DDE versus writing CSV and using excel macros versus SAS Add-in for Microsoft Office versus ODS. The pros and cons of each method will be highlighted.

### INTRODUCTION

Despite the proliferation of presentation tools the business still wants reports in Excel. The reports the business is looking for has complicated formatting as just doesn't follow standard SAS reporting structure. Lots of businesses are moving away from SAS on the desktop as a cost saving measure. In this paper we will look at possible solutions to writing your reports in Excel without cutting and pasting data.

### THE PROBLEM

- Business area wants reports in Excel
- Reports require significant formatting
- Moving away from SAS on the desktop

### POSSIBLE SOLUTIONS

- Use DDE to write to Excel
- Use CSV files with Excel macros
- Use SAS Add-in For Microsoft Office
- ODS to Excel

### DDE –DYNAMIC DATA EXCHANGE

DDE allows a SAS program to:

- Write directly to cells in an Excel workbook
- Execute Excel macros
- Execute Excel functions (eg. save as)

### USING DDE TO WRITE TO EXCEL

To use DDE to control where and what is written to a formatted Excel workbook you need to execute some or all of the following steps.

- Start excel.
- Create a filename to the target cells.

- Write the data.
- Refresh a pivot table.
- Run a VBA macro.
- Save the workbook.

## STARTING EXCEL

In order to write to Excel with SAS using dde you need to open an Excel workbook. You can use the x command in SAS to open a workbook. If the path or file name of the Excel workbook has spaces in it, it must be enclosed in quotes. Setting the options noxwait and noxsync causes SAS not to wait for the x command to finish (ie until the workbook is closed). The sleep command following the open allows excel time to start before you write to it.

```
Options noxwait noxsync;
X ""C:\prdsales\Expense Report tmp1.xlsx"";
Data _null_;
  x=sleep(10);
run;
```

**Figure 1 Starting Excel from SAS**

## CREATING A FILENAME

SAS needs a filename to write data and commands to Excel. For data this filename must identify the sheet and the cell range where the data is to be written. It is recommended that you set the lrecl to maximum to avoid line wrap.

```
Filename cmds dde 'excel|system';
```

**Figure 2 Filename for Excel Commands**

```
Filename data dde "excel|sheet1!r1c1:r10c10" lrecl=32000;
```

**Figure 3 Filename for data**

## CONTROL CELL RANGE

In the filename statement we specify the cell range where the data is to be written. You can use a macro variable to set the cells. Make sure you take into account the row you start on when calculating the range. There can be no spaces in the cell range (ie. R123C15).

```
Proc sql noprint;
  Select trim(left(put(count(*)+1,8.)))||"c15" into: nobs from cust;
quit;
filename data dde "excel|&sht!r1c1:r&nobs" lrecl=32000;
```

**Figure 4 Define calculated data range**

## WRITING DATA

Now we have a filename to write to. We can then use that filename in a data step. When writing from a data set to an Excel sheet we want to control moving to the next column. The notab option on the file

statement suppresses SAS default column control. The default is to change columns with each white space including spaces in the data. You can use `_n_ = 1` to set column titles.

```
Data _null_;
  file data notab;
  set cust;
  if _n_ = 1 then put "customer name" '09'x "customer address";
  put name '09'x addr1;
run;
```

**Figure 5 Code to write data to sheet**

## SAVING WORKBOOK

After you have finished writing to the workbook you need to save the result. There are 2 options save or save.as. Save updates the workbook in place. Save.as creates a new workbook.

```
Filename cmds dde 'excel|system';
Data _null_;
  File cmds;
  Put '[SAVE()]';
  Put '[QUIT()]';
run;
```

**Figure 6 Code to save a workbook**

```
Filename cmds dde 'excel|system';
Data _null_;
  File cmds;
  Put "[SAVE.as('C:\prdsales\Expense Report.xlsx')]";
  put '[QUIT()]';
run;
```

**Figure 7 Code to save workbook to new name**

## PIVOT TABLE REFRESH

Objects in your excel template need to be updated or refreshed when the data is changed. You first must go to the sheet with the object on it. Then select a cell in the pivot table so it is active. Then execute the refresh command.

```
Filename cmds dde 'excel|system';
Data _null_;
  File cmds;
  Put '[workbook.activate("Pivot")]';
  Put '[select("R7C2")]';
  Put '[pivot.refresh()]';
run;
```

**Figure 8 Code to refresh pivot table on sheet named Pivot**

## RUN VBA MACRO

If you have a VBA macro to format your workbook (delete blank line, update charts, ...) you want to be able to launch it from SAS once you have updated the data. Here is code to run the macro.

```

filenamecmds dde 'excel|system';
data _null_;
filecmds;
put '[RUN("Macro1")]';
run;

```

**Figure 9 Code to execute Excel macro**

## EXAMPLE OF USING DDE

Putting all the step together here is a program that opens a template, writes data to the data sheet, updates the pivot table, and save the workbook to a new name

```

%let rptdate = %sysfunc(intnx(month,%sysfunc(today()),-1,e),monyy7.);
%let rptmth = %sysfunc(intnx(month,%sysfunc(today()),-1,e),monname3.);
Options noxwait noxsync;
X ""C:\prdsales\Expense Report tpl.xlsx"";
Data _null_;
  x=sleep(10);
run;
proc sql noprint;
  select trim(left(put(count(*)+1,8.)))||"c15"into:nobs
  from sashelp.prdsale;
quit;
filename data dde "excel|data!r1c1:r&nobs" lrecl=32000;
data _null_;
  file data notab;
  set sashelp.prdsale;
  where put(month,monname3.) = "&rptmth";
  if _n_ = 1 then
    put "country" '09'x "Product" '09'x "Actual" '09'x
      "Predict" '09'x "Month";
  Put country '09'x product '09'x actual '09'x predict '09'x month;
run;
filename cmds dde 'excel|system';
data _null_;
  file cmds;
  put '[workbook.activate("Pivot")]';
  put '[select("R7C2")]';
  put '[pivot.refresh()]';
  x=sleep(10);
  put "[SAVE.as("""C:\prdsales\Expense Report &rptdate..xlsx"")]";
  put '[QUIT()]';
run;

```

**Figure 10 Sample program to write to Excel using dde**

## RESULTING DATA SHEET

After running the above program here is the resulting data sheet.

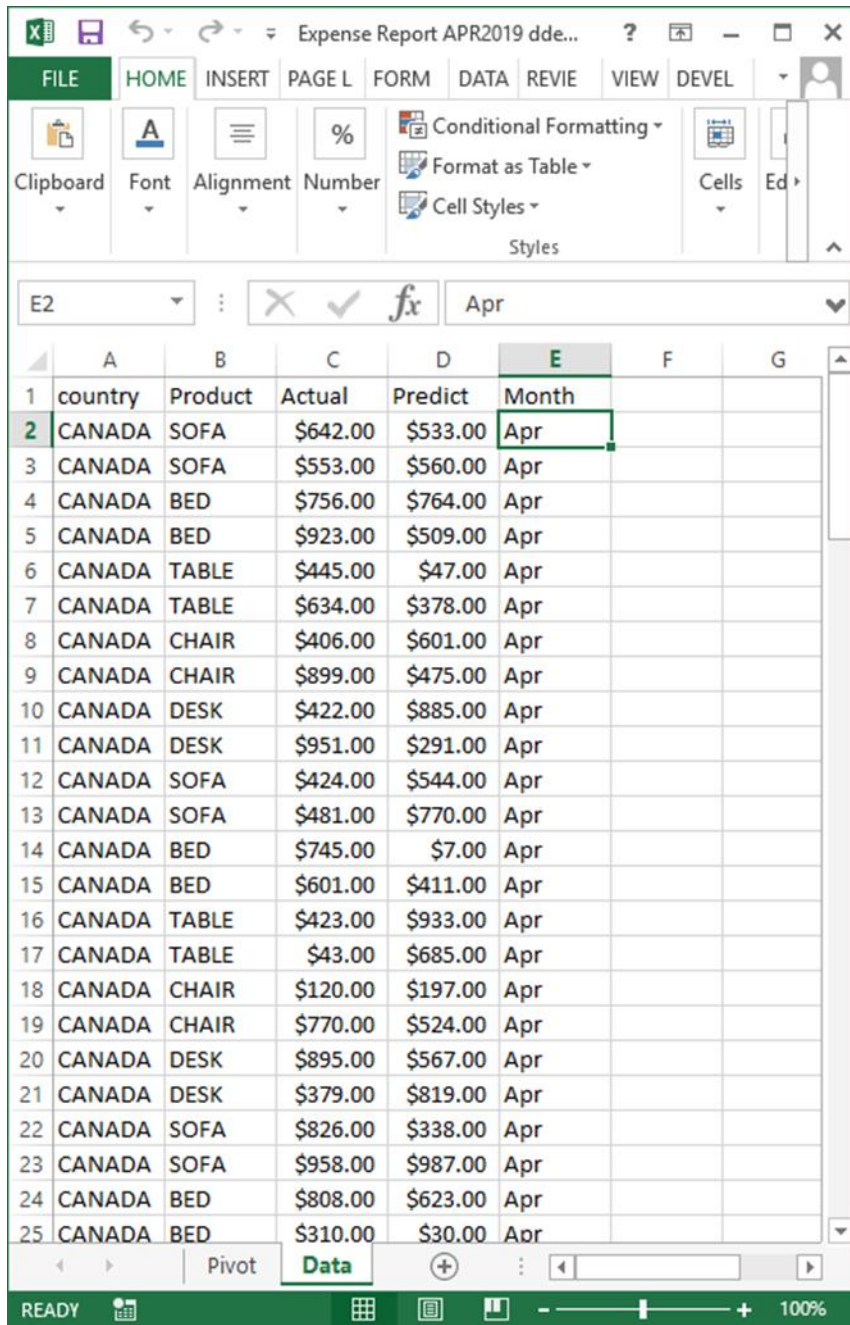


Figure 11 Data sheet resulting from SAS program using DDE

### PIVOT TABLE

After running the above program here is the resulting pivot table.

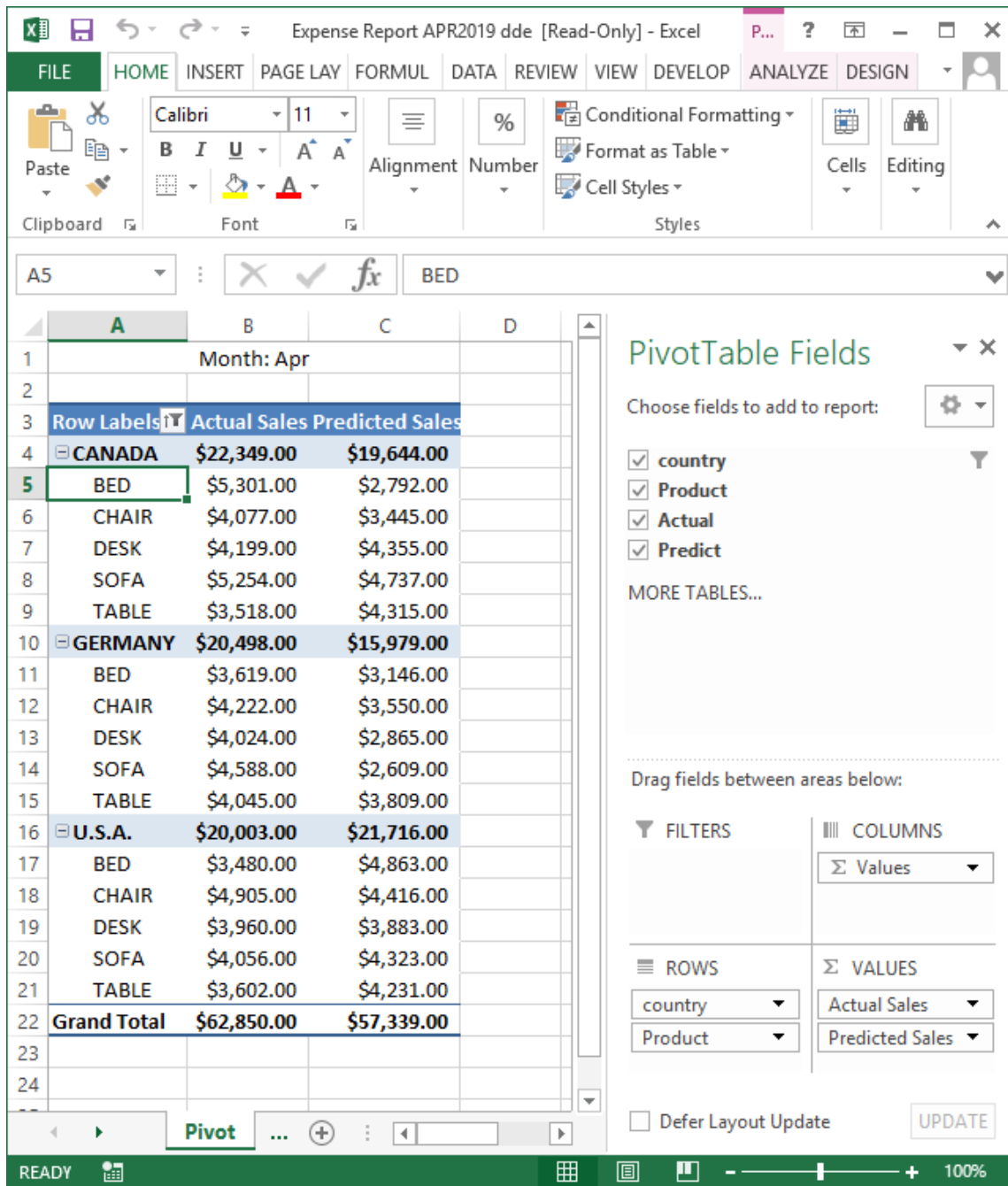


Figure 12 Pivot table resulting from SAS program using DDE

## USING CSV WITH EXCEL MACRO

In order to build our report using CSV file with an Excel macro. We use SAS to output the data to a CSV file and then start excel and run a VBA macro to load the data into the report. The step for the process are as follows:

- Use SAS to write data to CSV file and processing commands to an additional file
- Open Excel

- Run the report creation macro

```

%let rptdate = %sysfunc(intnx(month,%sysfunc(today()),-1,e),monyy7.);
%let rptmth = %sysfunc(intnx(month,%sysfunc(today()),-1,e),monname3.);
Data _null_;
  File 'C:\prdsales\Expense Report data.csv' delimiter = ',' dsd;
  Set sashelp.prdsale;
  Where put(month,monname3.) = "&rptmth";
  If _n_ = 1 then put "country,Product,Actual,Predict,Month";
  Put country product actual predict month;
run;
data _null_;
  file 'C:\prdsales\Expense Report cntl.csv' delimiter = ',' dsd;
  put "tpl,C:\prdsales\Expense Report tpl.xlsx";
  put "data,C:\prdsales\Expense Report data.csv";
  put "copy,data";
  put "rfsh,pivot";
  put "saveas,C:\prdsales\Expense Report &rptdate..xlsx";
  put "quit";
run;

```

**Figure 13 Program to write CSV file and processing commands**

## DATA SHEET

After running the above program here is the resulting data sheet.

The screenshot shows an Excel spreadsheet titled "Expense Report data [Read-Only] - E...". The ribbon is set to "HOME". The active cell is A1, containing the text "country". The spreadsheet data is as follows:

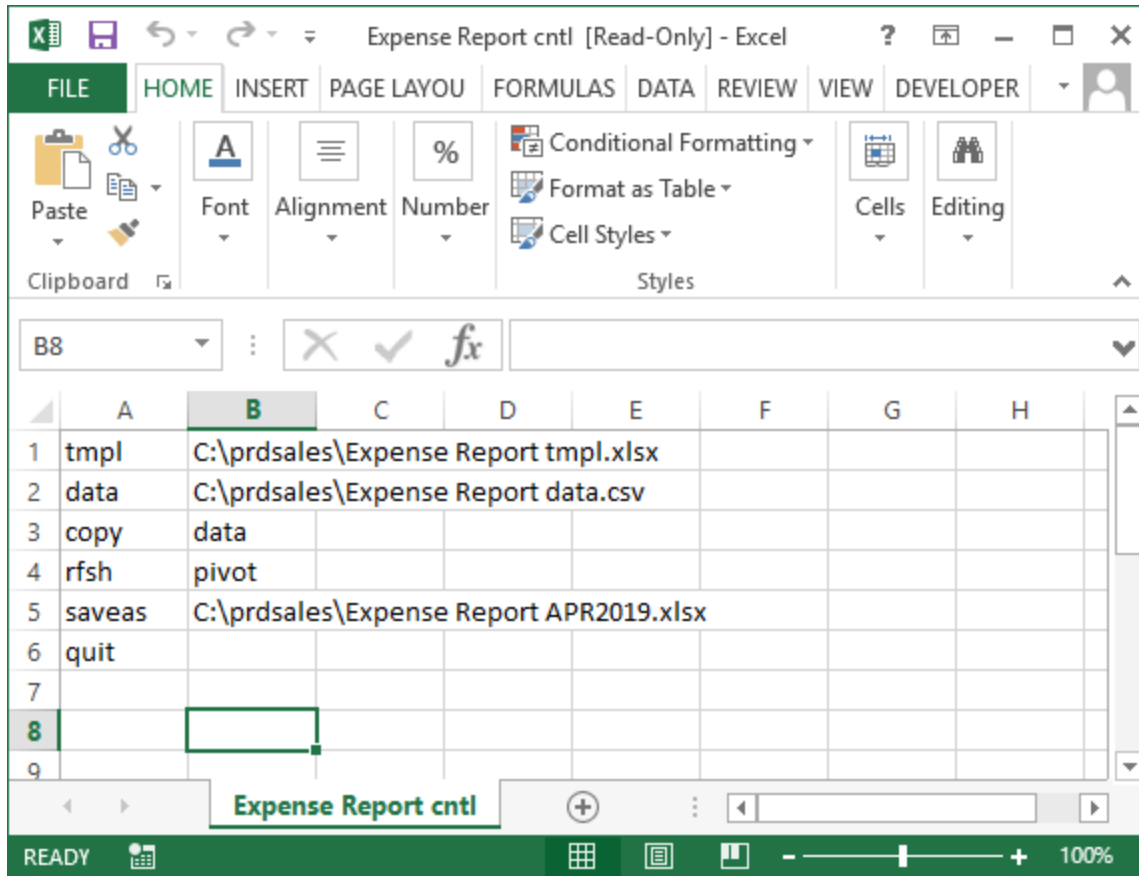
	A	B	C	D	E	F	G	H
1	country	Product	Actual	Predict	Month			
2	CANADA	SOFA	\$642.00	\$533.00	Apr			
3	CANADA	SOFA	\$553.00	\$560.00	Apr			
4	CANADA	BED	\$756.00	\$764.00	Apr			
5	CANADA	BED	\$923.00	\$509.00	Apr			
6	CANADA	TABLE	\$445.00	\$47.00	Apr			
7	CANADA	TABLE	\$634.00	\$378.00	Apr			
8	CANADA	CHAIR	\$406.00	\$601.00	Apr			
9	CANADA	CHAIR	\$899.00	\$475.00	Apr			
10	CANADA	DESK	\$422.00	\$885.00	Apr			
11	CANADA	DESK	\$951.00	\$291.00	Apr			
12	CANADA	SOFA	\$424.00	\$544.00	Apr			
13	CANADA	SOFA	\$481.00	\$770.00	Apr			
14	CANADA	BED	\$745.00	\$7.00	Apr			
15	CANADA	BED	\$601.00	\$411.00	Apr			
16	CANADA	TABLE	\$423.00	\$933.00	Apr			
17	CANADA	TABLE	\$43.00	\$685.00	Apr			
18	CANADA	CHAIR	\$120.00	\$197.00	Apr			
19	CANADA	CHAIR	\$770.00	\$524.00	Apr			
20	CANADA	DESK	\$895.00	\$567.00	Apr			

Figure 14 Data sheet resulting from SAS program to write CSV

### CONTROL CSV

After running the above program here is the resulting processing commands sheet.





**Figure 15 Processing commands sheet resulting from SAS program to write processing commands**

## VBA TO FORMAT REPORT

Not only do we need a SAS program to generate the CSV data sheet and CSV processing commands file. We need a vba macro that will interpret the processing commands and build the new report.

```

Sub format_report()
    x = Range("a1").End(xlDown).Row
    cntl = ActiveWorkbook.Name

    For i = 1 To x
        Windows(cntl).Activate
        Cmd = Cells(i, 1).Value
        cparm = Cells(i, 2).Value

        If cmd= "tpl" Then
            tname = opn(cparm)
        ElseIf cmd = "data" Then
            dname = opn(cparm)
        ElseIf cmd = "copy" Then
            cpy dname, tname, cparm
        ElseIf cmd = "rfs" Then
            rfsht name, cparm
        ElseIf cmd= "saveas" Then

```

```

        sav tname, cparm
    ElseIf cmd= "quit" Then
        Application.Quit
    End If
Next i
End Sub
Function opn(fn) As String
    Workbooks.OpenFilename:=fn
    opn = ActiveWorkbook.Name
End Function

Sub cpy(dname, tname, sn)
    Windows(dname).Activate
    Cells.Select
    Application.CutCopyMode= False
    Selection.Copy
    Windows(tname).Activate
    Sheets(sn).Select
    Cells.Select
    ActiveSheet.Paste
End Sub

Sub rfsh(tname, sn)
    Windows(tname).Activate
    Sheets(sn).Select
    ActiveWorkbook.RefreshAll
End Sub

Sub sav(tname, rn)
    Windows(tname).Activate
    ActiveWorkbook.SaveAsFilename:=rn, _
        FileFormat:=xlOpenXMLWorkbook, CreateBackup:=False
End Sub

```

**Figure 16 VBA macro to format CSV report**

## **PIVOT TABLE**

After running the above VBA macro here is the resulting pivot table.

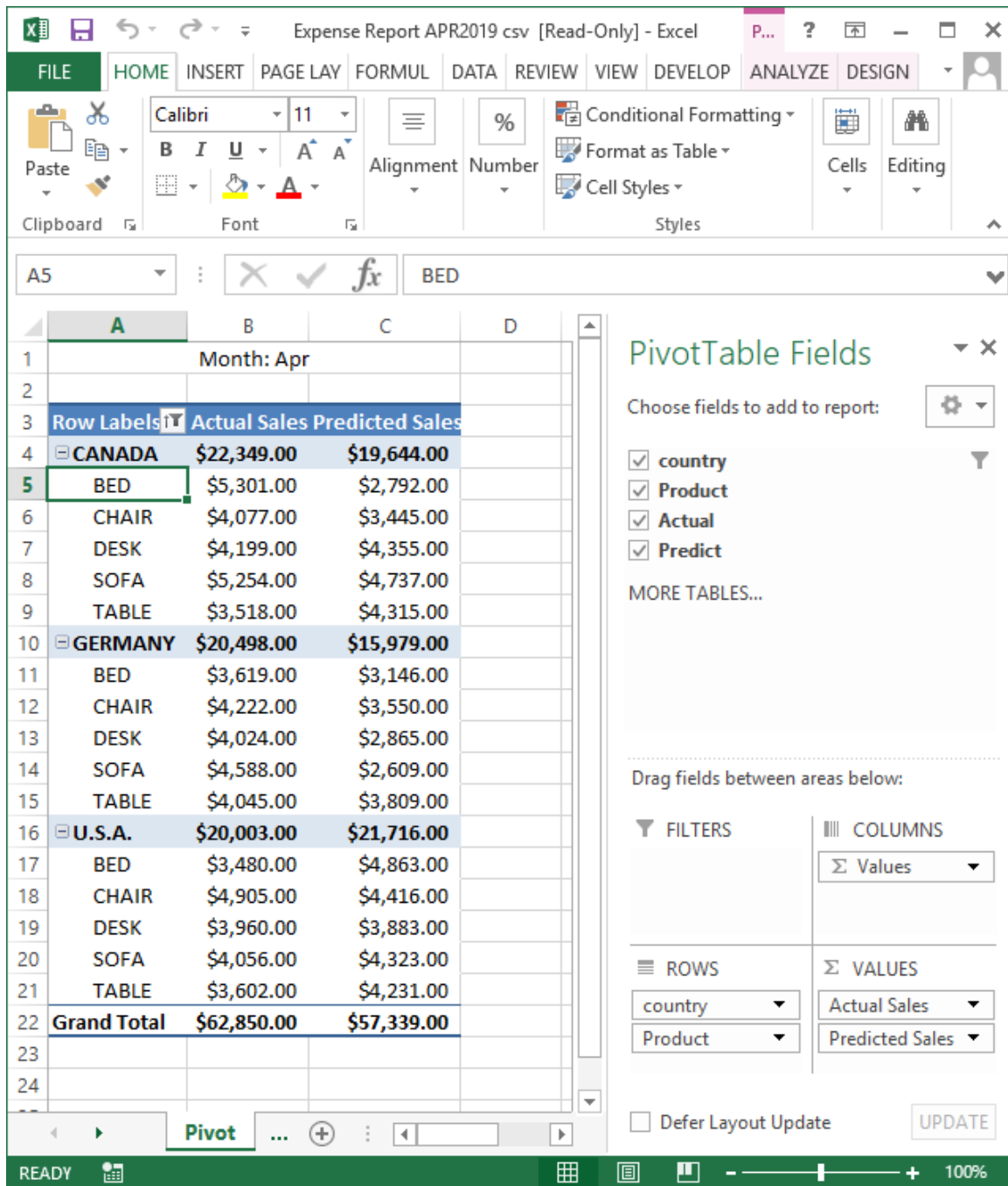


Figure 17 Pivot table resulting from SAS program and VBA macro

## USING SAS ADD-IN FOR MICROSOFT OFFICE

In order to build our report using SAS Add-in for Microsoft Office. We use SAS output the data refresh the dataset that is the basis of the report and then start excel and refresh the data into the report. The step for the process are as follows:

- In SAS update the dataset that will be the basis of your report
- Open Excel template for the report

- Refresh datasets and save the updated report

```

%let rptmth= %sysfunc(intnx(month,%sysfunc(today()),-1,e),monname3.);
Libname prdsales 'c:\prdsales';
Data prdsales.prdsale;
  Set sashelp.prdsale;
  Where put(month,monname3.) = "&rptmth";
run;

```

Figure 18 Code to refresh the dataset for SAS addin

## DATA SHEET

After running the above program here is the resulting data sheet.

	ACTUAL	PREDICT	COUNTRY	REGION	DIVISION	PRODTYPE	PRODUCT	QUARTER	YEAR	MONTH
1	\$642.00	\$533.00	CANADA	EAST	EDUCATION	FURNITURE	SOFA	2	1993	Apr
2	\$553.00	\$560.00	CANADA	EAST	EDUCATION	FURNITURE	SOFA	2	1994	Apr
3	\$756.00	\$764.00	CANADA	EAST	EDUCATION	FURNITURE	BED	2	1993	Apr
4	\$923.00	\$509.00	CANADA	EAST	EDUCATION	FURNITURE	BED	2	1994	Apr
5	\$445.00	\$47.00	CANADA	EAST	EDUCATION	OFFICE	TABLE	2	1993	Apr
6	\$634.00	\$378.00	CANADA	EAST	EDUCATION	OFFICE	TABLE	2	1994	Apr
7	\$406.00	\$601.00	CANADA	EAST	EDUCATION	OFFICE	CHAIR	2	1993	Apr
8	\$899.00	\$475.00	CANADA	EAST	EDUCATION	OFFICE	CHAIR	2	1994	Apr
9	\$422.00	\$885.00	CANADA	EAST	EDUCATION	OFFICE	DESK	2	1993	Apr
10	\$951.00	\$291.00	CANADA	EAST	EDUCATION	OFFICE	DESK	2	1994	Apr
11	\$424.00	\$544.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA	2	1993	Apr
12	\$481.00	\$770.00	CANADA	EAST	CONSUMER	FURNITURE	SOFA	2	1994	Apr

Figure 19 data sheet resulting from using SAS Addin

## PIVOT TABLE

After refreshing the pivot table in Excel here is the resulting pivot table.

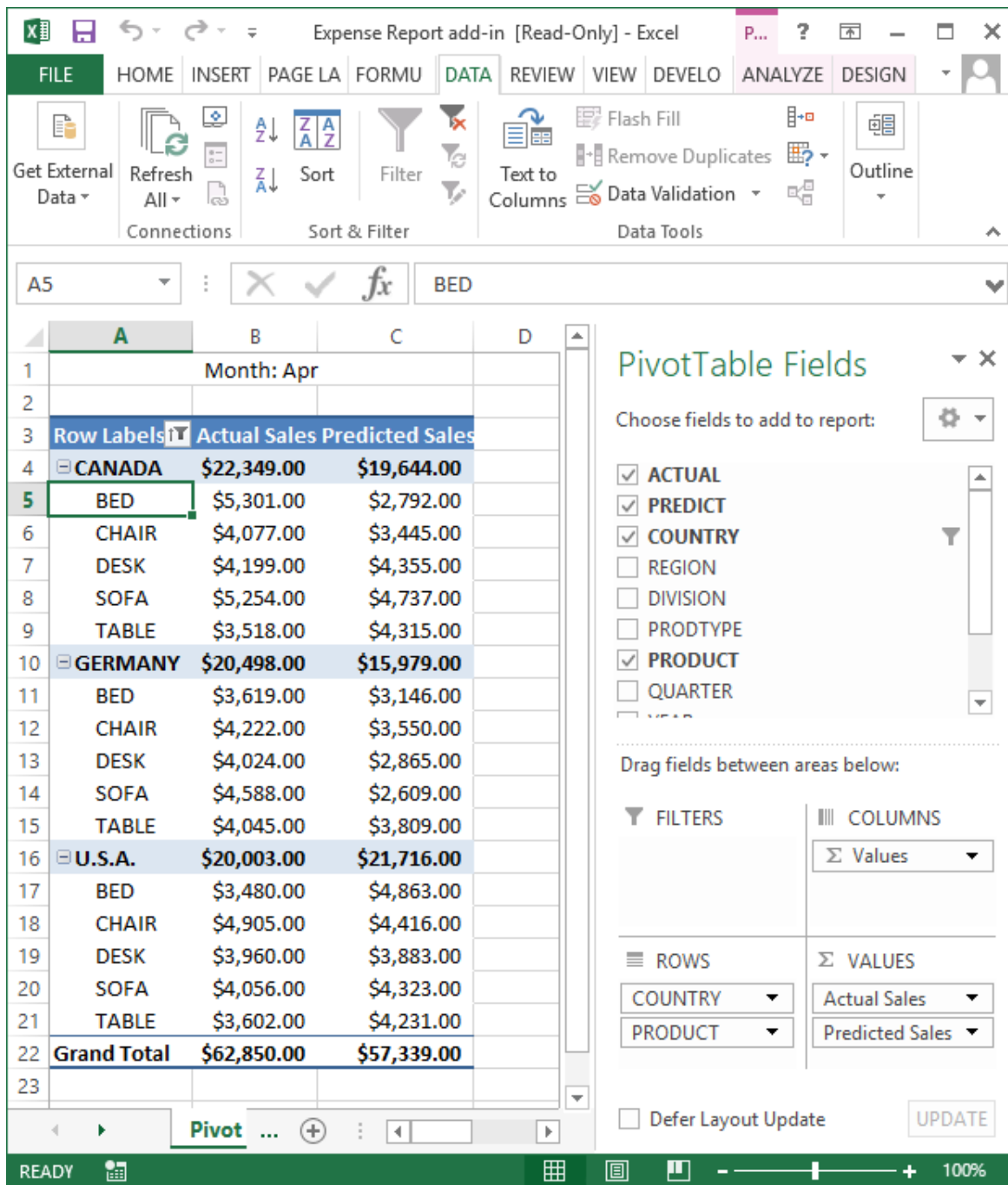


Figure 20 Pivot table resulting from using SAS Addin

## ODS TO EXCEL

ODS can be used to write to Excel. The Excel option for ODS is new in SAS 9.4, prior to 9.4 you could use ODS html or ODS xml and open the result in excel. Using ODS Excel with SAS reporting procedures (print, report, tabulate, ...), SAS can create reports in excel workbook. There are a large number of options for formatting and content. The following examples are from a blog by Chevell Parker that covers the format and content options. <https://blogs.sas.com/content/sgf/2017/02/20/tips-for-using-the-ods-excel-destination/>

Here is an example ODS Excel program

```
ods excel file="c:\temp.xlsx"
  options (start_at="3,3"
    frozen_headers="5"
    frozen_rowheaders="3"
    autofilter="1-5"
    sheet_name="Sales Report"
    row_repeat="2"
    embedded_titles="yes");
proc print data=sashelp.orsales;
  title "Sales Report for the Year 1999";
run;
ods excel close;
```

Figure 21 ODS Excel program

## ODS PARAMETERS

START\_AT= option enables you to select the placement of the initial table or graph on the worksheet. In Microsoft Excel, by default, a table or graph begins in position A1. However, pinning the table or graph in that position does not always provide the best visual presentation.

FROZEN\_HEADERS= option locks the table header in your table while the FROZEN\_ROWHEADERS= option locks row headers. Both of these options lock your headers and row headers so that they remain visible as you scroll through the values in the table rows and columns.

AUTOFILTER= option enables you to add filters to tables headers so that you can filter based on the value of a particular column.

SHEET\_NAME= option enables you to add more meaningful text to the worksheet tab.

ROW\_REPEAT= option makes your report more readable by repeating the rows that you specify in the option. If this report is ever printed, specifying the rows to repeat, in this case the column headers would allow for easy filtering of the data.

EMBEDDED\_TITLE= option specifies that the title that is specified in the TITLE statement should be displayed.

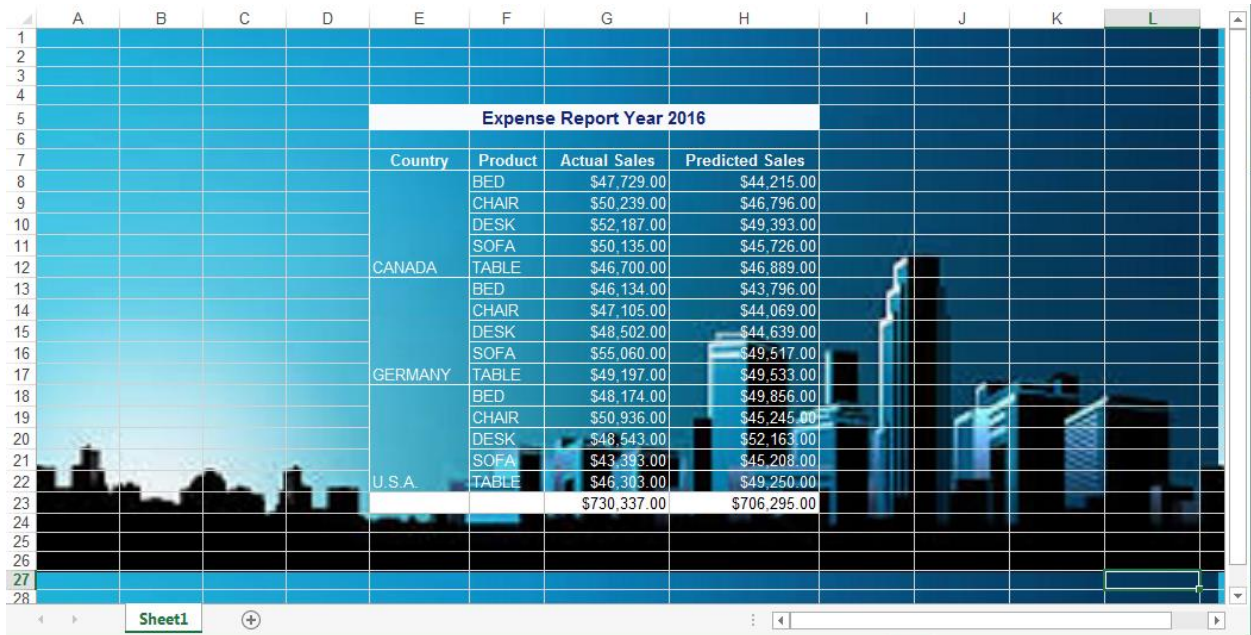
## ODS OUTPUT

	Year	Quarter	Product Line	Product Category	Product Group	Quantity	Profit	Total Retail Price
1	1999	1999Q1	Children	Children Sports	A-Team, Kids	286	4980.15	8990.90
2	1999	1999Q1	Children	Children Sports	Bathing Suits, Kids	98	1479.95	2560.40
3	1999	1999Q1	Children	Children Sports	Eclipse, Kid's Clothes	588	9348.95	18768.80
4	1999	1999Q1	Children	Children Sports	Eclipse, Kid's Shoes	334	7136.80	14337.20
5	1999	1999Q1	Children	Children Sports	Lucky Guy, Kids	303	7163.00	12996.20
6	1999	1999Q1	Children	Children Sports	N.D. Gear, Kids	755	19153.05	34250.50
7	1999	1999Q1	Children	Children Sports	Olissons, Kids	209	1975.35	3339.30
8	1999	1999Q1	Children	Children Sports	Orion Kid's Clothes	14	288.80	580.40
9	1999	1999Q1	Children	Children Sports	Osprey, Kids	454	7334.70	13219.60
10	1999	1999Q1	Children	Children Sports	Tracker Kid's Clothes	1243	21847.85	40049.50
11	1999	1999Q1	Children	Children Sports	Ypsilon, Kids	139	3020.85	5354.70
12	1999	1999Q1	Clothes & Shoes	Clothes	Eclipse Clothing	2938	84962.50	170206.10
13	1999	1999Q1	Clothes & Shoes	Clothes	Green Tomato	171	4706.85	7846.20
14	1999	1999Q1	Clothes & Shoes	Clothes	Knitwear	1554	79951.69	140077.94
15	1999	1999Q1	Clothes & Shoes	Clothes	LSF	335	16878.00	32535.50

Figure 22 Output of ODS example program

## ODS BACKGROUND IMAGE

There are formatting options that allow you define a background image for your report.



Expense Report Year 2016			
Country	Product	Actual Sales	Predicted Sales
CANADA	BED	\$47,729.00	\$44,215.00
	CHAIR	\$50,239.00	\$46,796.00
	DESK	\$52,187.00	\$49,393.00
	SOFA	\$50,135.00	\$45,726.00
	TABLE	\$46,700.00	\$46,889.00
GERMANY	BED	\$46,134.00	\$43,796.00
	CHAIR	\$47,105.00	\$44,069.00
	DESK	\$48,502.00	\$44,639.00
	SOFA	\$55,060.00	\$49,517.00
	TABLE	\$49,197.00	\$49,533.00
U.S.A.	BED	\$48,174.00	\$49,856.00
	CHAIR	\$50,936.00	\$45,245.00
	DESK	\$48,543.00	\$52,163.00
	SOFA	\$43,393.00	\$45,208.00
	TABLE	\$46,303.00	\$49,250.00
		\$730,337.00	\$706,295.00

Figure 23 ODS report with background image

## ODS CONTENT IMAGES

There are formatting options that allow you define images different field values in your report.




	A	B	C	D	E	F	G	H
1	sas THE POWER TO KNOW.							
2	Sales Report for 2017							
3								
4		<b>Country</b>	<b>Product</b>	<b>Actual Sales</b>	<b>Predicted Sales</b>			
5		CANADA	BED	\$47,729.00	\$44,215.00			
6			CHAIR	\$50,239.00	\$46,796.00			
7			DESK	\$52,187.00	\$49,393.00			
8			SOFA	\$50,135.00	\$45,726.00			
9			TABLE	\$46,700.00	\$46,889.00			
10		GERMANY	BED	\$46,134.00	\$43,796.00			
11			CHAIR	\$47,105.00	\$44,069.00			
12			DESK	\$48,502.00	\$44,639.00			
13			SOFA	\$55,060.00	\$49,517.00			
14			TABLE	\$49,197.00	\$49,533.00			
15		U.S.A.	BED	\$48,174.00	\$49,856.00			
16			CHAIR	\$50,936.00	\$45,245.00			
17			DESK	\$48,543.00	\$52,163.00			
18			SOFA	\$43,393.00	\$45,208.00			
19			TABLE	\$46,303.00	\$49,250.00			
20				\$730,337.00	\$706,295.00			
21								
22								

Figure 24 ODS report with images tied to field values

## CONCLUSION

### WRITING TO EXCEL USING DDE

#### Advantages

- Data collection and report generation can be 1 program (1 step execution)
- Uses the full power of Excel reporting constructs
- Minimal changes to existing report to implement.

#### Disadvantages

- Must run SAS on windows platform for at least some of process.
- Cannot be using PC while DDE portion of program is running.
- Some VBA skills may be required.

### USING SAS ADD-IN.

#### Advantages

- SAS Platform independent.
- Uses the full power of Excel reporting constructs
- No VBA skills required.



## **Disadvantages**

- 2 step process, must gather data then run Excel and update the report.
- Need to change existing reports to reference the SAS data source.

## **USING CSV WITH EXCEL MACRO.**

### **Advantages**

- SAS Platform independent.
- Uses the full power of Excel reporting constructs
- Minimal changes to existing report to implement.

### **Disadvantages**

- 2 step process, must gather data then run Excel and update the report.
- Need to change existing reports to reference the SAS data source.
- VBA skills required.

## **ODS TO EXCEL.**

### **Advantages**

- SAS Platform independent.
- No VBA skills required.
- Data collection and report generation can be 1 program (1 step execution)

### **Disadvantages**

- Does not uses the full power of Excel reporting constructs
- Need to change existing reports.

## **REFERENCES**

Parker Chevell, "Tips for Using the ODS Excel Destination", blog, last revised February 20, 2017. Available at <https://blogs.sas.com/content/sgf/2017/02/20/tips-for-using-the-ods-excel-destination/>

## **CONTACT INFORMATION**

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

Robert Ellsworth  
Ellsworth Stewart Consulting Inc.  
416-414-1172  
rob@escorp.ca