

Customized Excel Output Using the Excel Libname

GHSUG 2009-04-24

Harry Droogendyk

Stratia Consulting Inc.

Introduction



- Many ways to create Excel output
 - delimited text files
 - ODS HTML
 - ExcelXP tagset
 - DDE (*arghhhh !!!*)
 - Excel libname
- Excel libname can also *read* Excel data
 - no time for that today

Excel Libname – Steps



1. Create customized Excel file
2. Define "named range" in Excel sheet
3. Process SAS data
 - massage into the structure of range created
4. Excel libname
 - clear the Excel named range
 - load the Excel named range
 - done

Not necessary to have Excel on the machine

Excel Libname – Syntax



Typical syntax with addition of <engine name>

```
libname xls <engine-name> physical-file-name;
```

```
libname xls excel 'c:\temp\report.xls';
```

- file doesn't exist ?
 - SAS will create

How to Use ?

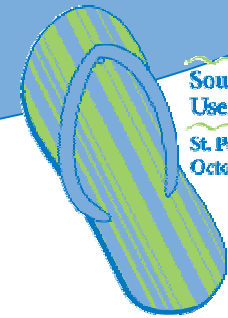


Excel libname can be used like any other:

```
data xls.class;  
    set sashelp.class;  
run;  
proc copy in    = sashelp  
          out   = xls;  
    select prdsale shoes;  
run;
```

How to Use ?

SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008



SAS Explorer displays:

The screenshot shows the SAS Explorer window with the title 'Explorer'. Below the title bar is a tab labeled 'Contents of 'Xls''. The main area contains a table with three columns: 'Name', 'Size', and 'Type'. The table lists several SAS tables: PRDSALE, PRDSALE\$, SHOES, SHOES\$, class, and class\$. Each row has a small icon to the left of the name.

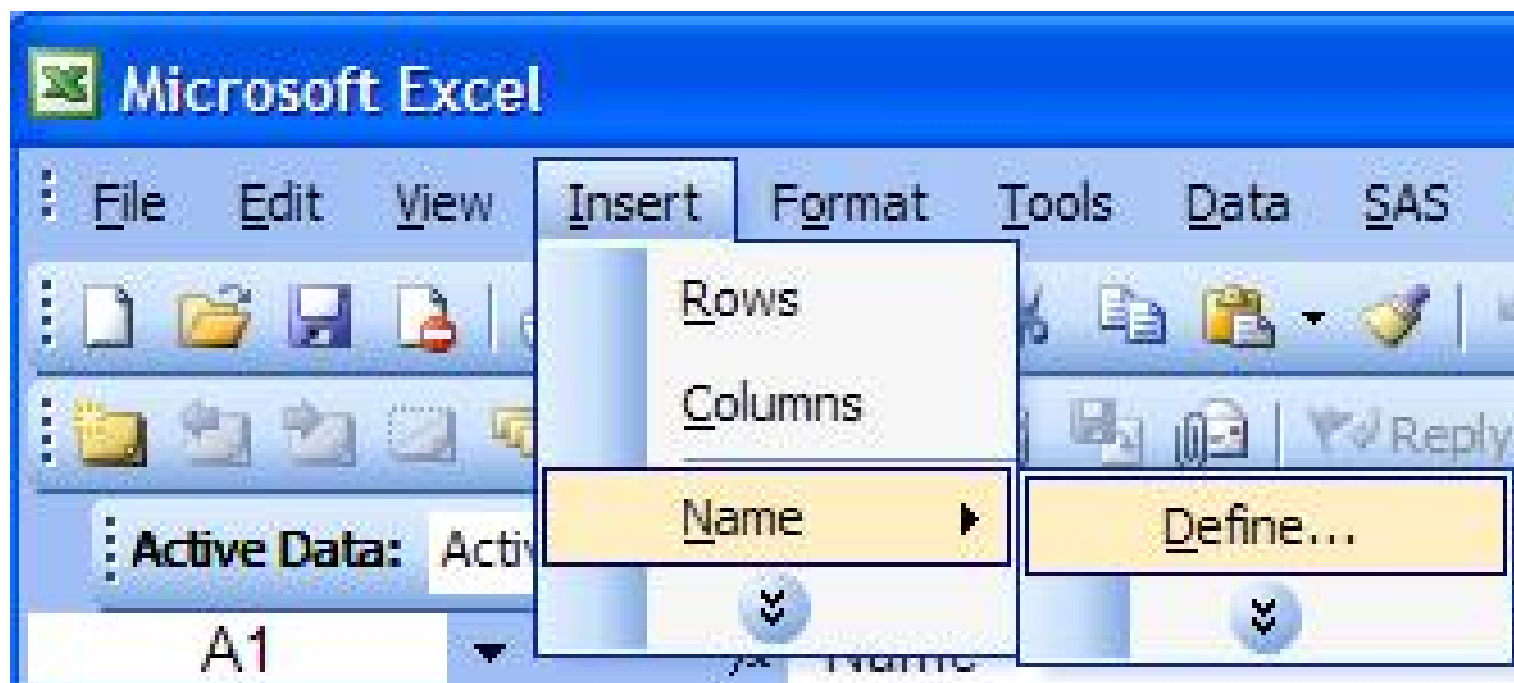
Name	Size	Type
PRDSALE		Table
PRDSALE\$		Table
SHOES		Table
SHOES\$		Table
class		Table
class\$		Table

The screenshot shows a SAS report window titled 'report.xls'. It displays a data table with columns A through E. The first row contains headers: Name, Sex, Age, Height, and Weight. The second row contains data for Alfred: M, 14, 69, 112.5. The third row contains data for Alice: F, 13, 56.5, 84. Below the table is a navigation bar with buttons for navigating between reports.

	A	B	C	D	E
1	Name	Sex	Age	Height	Weight
2	Alfred	M	14	69	112.5
3	Alice	F	13	56.5	84

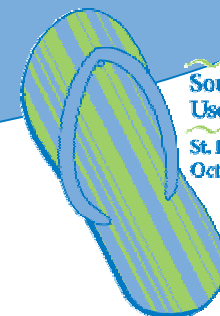
Named Ranges

SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008



Named Ranges

SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008




Define Name [X]

Names in workbook:

class
class
PRDSALE
SHOES

OK
Close
Add
Delete

Refers to:
=class!\$A\$1:\$E\$20



Capabilities / Limitations



Capabilities:

- create workbooks, sheets
- named ranges
 - delete contents, populate, append, read

Limitations:

- formatting
- delete entire workbook, sheets, named ranges
- deal with formulas

Formatted Excel Report



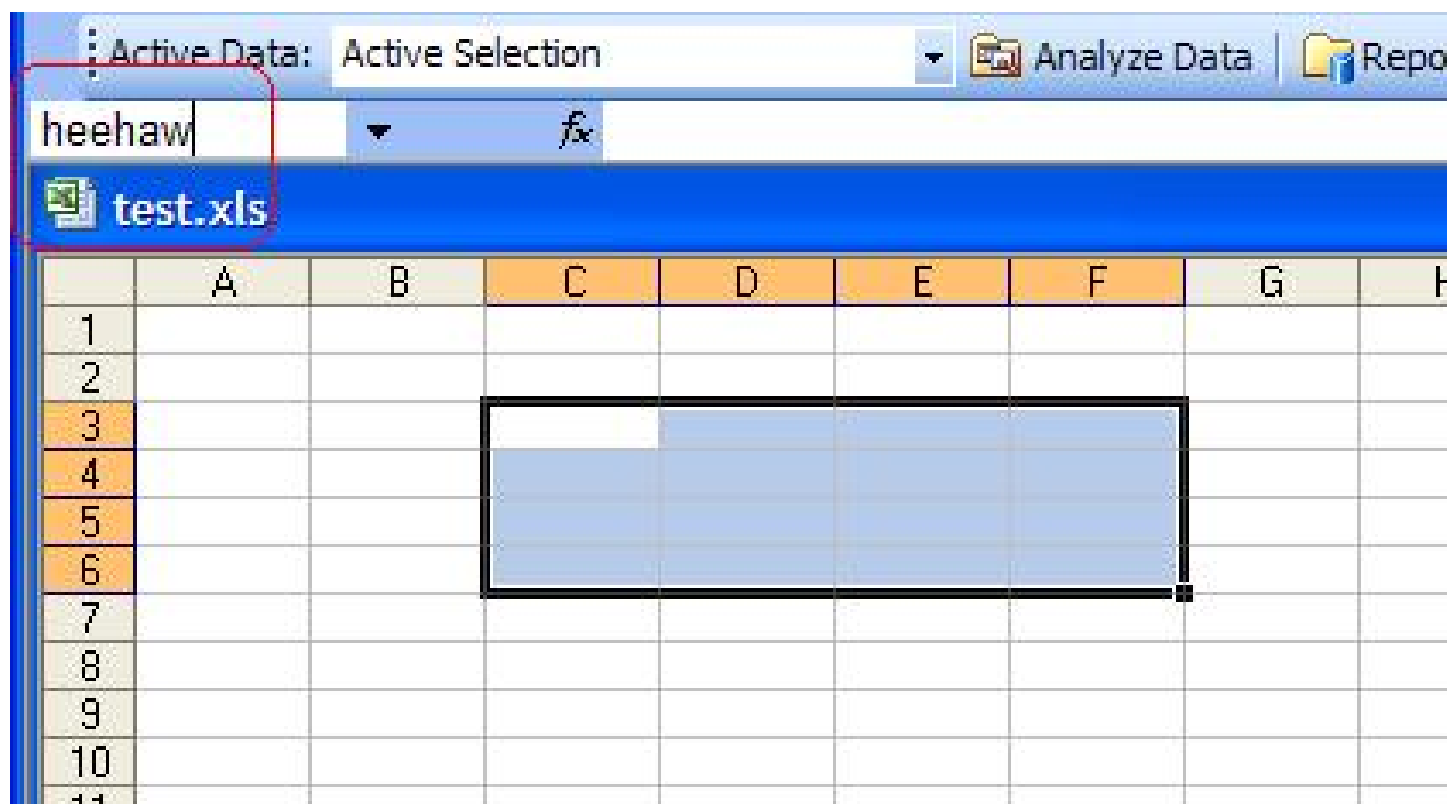
Separate processing from presentation!

- SAS crunches data
- Excel to develop pretty, useful output
- create an Excel "template"
- populate the "template" with SAS

Defining Named Ranges



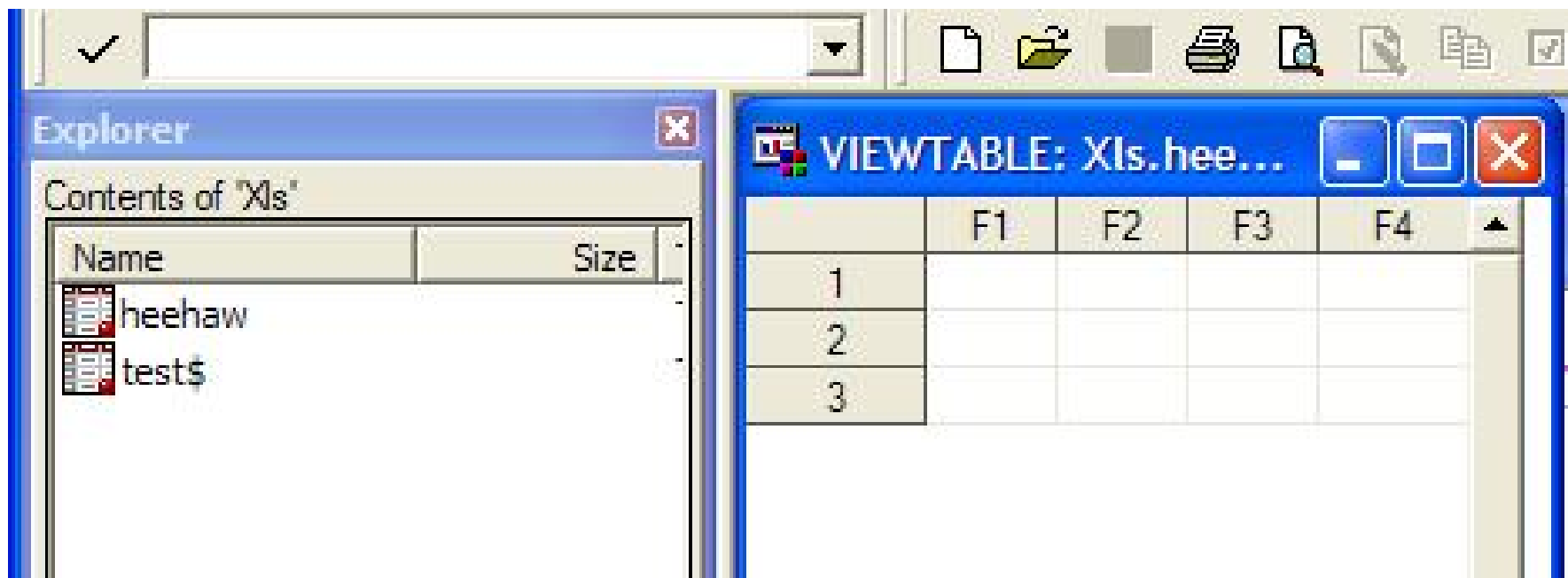
- could use Insert, Name, Define
- easier to use mouse



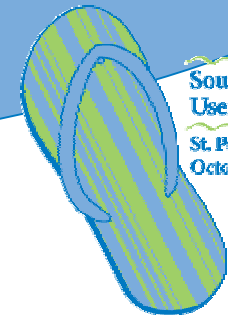
Defining Named Ranges



```
libname xls excel 'c:\temp\test.xls' ver=2003;
```



Populate Named Range



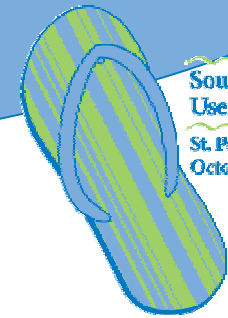
SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

```
proc datasets lib = xls nolist;  
    delete heehaw;  
quit;
```

```
data xls.heehaw;  
    set sashelp.class ( obs = 4  
                        keep = name sex age height );  
run;
```

```
libname xls clear; * very important !! ;
```

Populate Named Range



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

The screenshot shows an Excel spreadsheet window titled 'test.xls'. The spreadsheet has columns A through G and rows 1 through 9. A named range is defined in the range C3:F7, which is highlighted with a thick black border. The data within this named range is as follows:

Name	Sex	Age	Height
Alfred	M	14	69
Alice	F	13	56.5
Barbara	F	13	65.3
Carol	F	14	62.8

The spreadsheet interface includes standard Excel window controls (minimize, maximize, close) in the top right corner and navigation buttons (back, forward, etc.) in the bottom left corner. The status bar at the bottom shows the active sheet is 'test'.

Formatted Report

SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008



dept_salary_report_template.xls

	A	B	C	D	E	F
2	Departmental Salaries					
3	Dept	No. Employees	Total Salaries	Minimum Salary	Maximum Salary	Average Salary
5						#DIV/0!
6						#DIV/0!
7						#DIV/0!
8						#DIV/0!
9						#DIV/0!

Define Name

Names in workbook:

- date
- dept_salary

Refers to:

=Sheet1!\$F\$1:\$F\$2

Sheet1

Formatted Report



```
options noxwait xsync;
x 'copy c:\temp\dept_salary_report_template.xls
  c:\temp\dept_salary_report.xls';

libname xls excel 'c:\temp\dept_salary_report.xls'
  ver=2003;

proc datasets lib = xls nolist;
  delete date dept_salary;      * case sensitive ;
quit;
data xls.date;
  format t yymmddd10.;
  t = today();
run;
```


Formatted Report

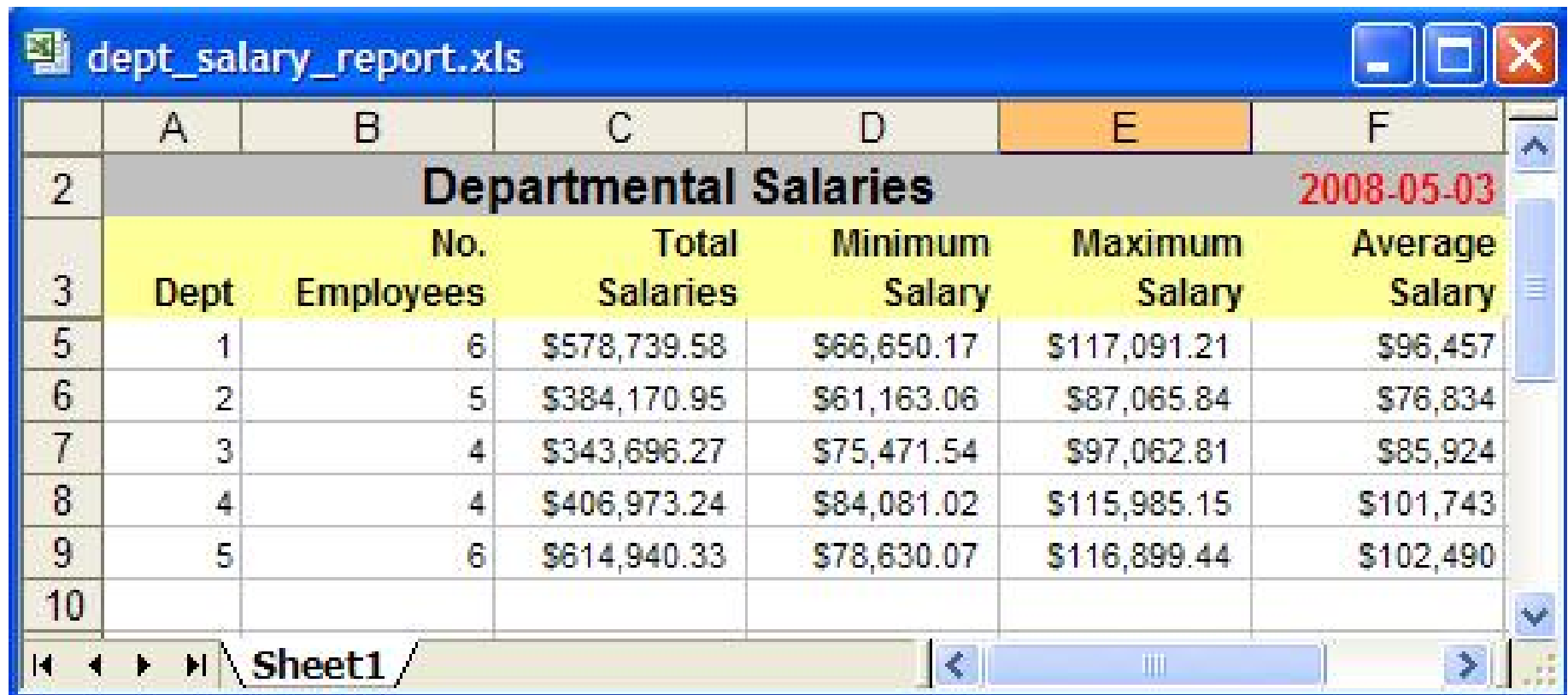


* RETAIN used to ensure variables are in correct order for Excel report;

```
data xls.dept_salary;  
    retain dept employees total_salary  
    min_salary max_salary;  
    set dept_salaries_sum;  
run;  
  
libname xls clear;
```

Formatted Report

SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008



The image shows an Excel spreadsheet titled "dept_salary_report.xls". The spreadsheet displays a report of departmental salaries. The title "Departmental Salaries" is centered in row 2, with the date "2008-05-03" in cell F2. The data is organized into columns: Dept (A), No. Employees (B), Total Salaries (C), Minimum Salary (D), Maximum Salary (E), and Average Salary (F). Rows 5 through 9 contain data for departments 1 through 5. Row 10 is empty. The spreadsheet interface includes standard Excel window controls and a sheet tab labeled "Sheet1".

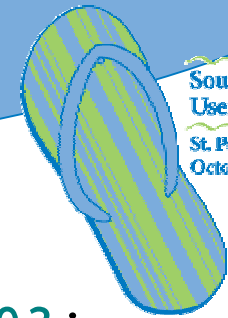
	A	B	C	D	E	F
2	Departmental Salaries					2008-05-03
3	Dept	No. Employees	Total Salaries	Minimum Salary	Maximum Salary	Average Salary
5	1	6	\$578,739.58	\$66,650.17	\$117,091.21	\$96,457
6	2	5	\$384,170.95	\$61,163.06	\$87,065.84	\$76,834
7	3	4	\$343,696.27	\$75,471.54	\$97,062.81	\$85,924
8	4	4	\$406,973.24	\$84,081.02	\$115,985.15	\$101,743
9	5	6	\$614,940.33	\$78,630.07	\$116,899.44	\$102,490
10						

Appending Pivot Data



- not all reports are created from scratch
- monthly process
 - calculates a number of metrics
 - appends it to previously existing Excel data
 - refresh the pivot sourced from the data

Appending Pivot Data



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

```
libname xls excel 'c:\temp\append.xls' ver=2003;
```

```
proc datasets lib = xls nolist;
```

```
  delete output;
```

```
quit;
```

```
data xls.output;
```

```
  set sashelp.prdsale
```

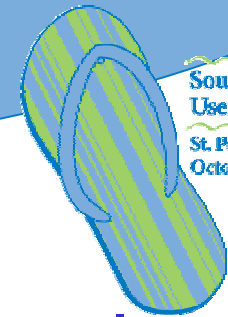
```
    ( where = ( put(month,yymmnn6.) ne '199412' ) );
```

```
  mm = month(month);
```

```
run;
```

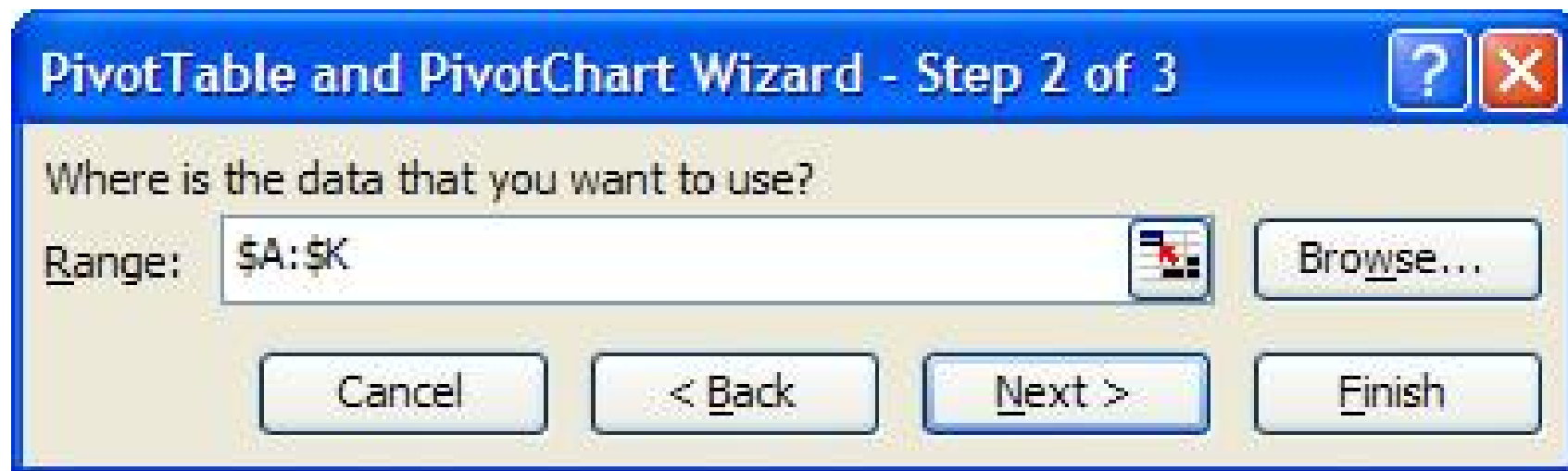
```
libname xls clear;
```

Appending Pivot Data



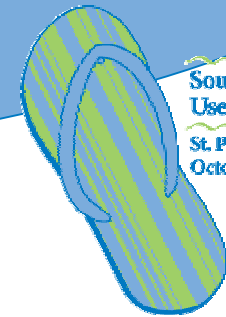
SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

- open append.xls when initial load completes
- use the PivotTable wizard to define:
 - source of the PivotTable data
 - dimension & fact variables



Appending Pivot Data

SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008



append.xls

	A	B	C	D
1	COUNTRY	(All)		
2	REGION	(All)		
3	DIVISION	(All)		
4	YEAR	1994		
5				
6	Actual Sales	Prod Type		
7	Mth	FURNITURE	OFFICE	Grand Total
8	1	\$12,956	\$20,748	\$33,704
9	2	\$11,033	\$16,680	\$27,713
10	3	\$10,350	\$17,996	\$28,346
11	4	\$13,280	\$18,989	\$32,269
12	5	\$12,673	\$20,673	\$33,346
13	6	\$11,501	\$16,243	\$27,744
14	7	\$11,494	\$17,112	\$28,606
15	8	\$10,786	\$19,615	\$30,401
16	9	\$12,745	\$17,297	\$30,042
17	10	\$10,088	\$15,284	\$25,372
18	11	\$13,411	\$17,983	\$31,394
19	Grand Total	\$130,317	\$198,620	\$328,937
20				
21				

Appending Pivot Data



```
libname xls excel 'c:\temp\append.xls' scan_text = no  
ver=2003;
```

```
data this_month;  
    set sashelp.prdsale  
        ( where = ( put(month,yymmnn6.) eq '199412' ) );  
    mm = month(month);  
run;
```

```
proc append base = xls.output  
            data = this_month;  
run;
```

```
libname xls clear;
```

Appending Pivot Data

SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008



append.xls

	A	B	C	D
1	COUNTRY	(All)		
2	REGION	(All)		
3	DIVISION	(All)		
4	YEAR	1994		
5				
6	Actual Sales	Prod Type		
7	Mth	FURNITURE	OFFICE	Grand Total
8	1	\$12,956	\$20,748	\$33,704
9	2	\$11,033	\$16,680	\$27,713
10	3	\$10,350	\$17,996	\$28,346
11	4	\$13,280	\$18,989	\$32,269
12	5	\$12,673	\$20,673	\$33,346
13	6	\$11,501	\$16,243	\$27,744
14	7	\$11,494	\$17,112	\$28,606
15	8	\$10,786	\$19,615	\$30,401
16	9	\$12,745	\$17,297	\$30,042
17	10	\$10,088	\$15,284	\$25,372
18	11	\$13,411	\$17,983	\$31,394
19	12	\$11,413	\$20,510	\$31,923
20	Grand Total	\$141,730	\$219,130	\$360,860
21				

Odds and Sods



- Excel libname engine options
 - dslabels=yes, see Online Docs for more
- Excel97 is default
 - 9.1.3
 - for latest available functionality use VER=2003
 - 9.2
 - supports Excel 2007 and .xlsx & .xlsb
 - defaults to VER=2007 for .xlsx & .xlsb
- RETAIN before the SET to re-order columns

Wrap



- Excel libname engine is versatile
 - Excel need not be installed, eg. servers
 - PC File Server on UNIX
- separates processing from presentation
 - leverage the strengths of each
 - not limited to ODS options

Credits



See the paper:

- Paul Choate and Carol Martel
- Winfried Jakob
- SAS Online Docs

Author



Harry Droogendyk

conf@stratia.ca

www.stratia.ca