

## Paper 140-2009

**SAS® and Microsoft Excel: Two Extreme Worlds. Learn How SAS Works with Excel Data and How Excel Exploits the Power of SAS**

Henri Theuwissen, BI Knowledge Sharing, Bierbeek, Belgium

**ABSTRACT**

Microsoft Excel is still one of the most popular tools for reporting and data exchange. SAS® applications often receive input through Excel files and must return data or information in Excel format. This paper illustrates the openness between SAS and Microsoft Excel. There are several easy ways to exchange data between SAS and Excel in both directions. SAS provides several ways to access data from Excel: the IMPORT procedure and the SAS/ACCESS® LIBNAME solution are illustrated.

Many SAS applications have to provide results in Excel format. The paper illustrates the EXPORT procedure, the SAS/ACCESS LIBNAME engine, the %DS2CSV macro, and the power of the Output Delivery System (ODS). ODS examples are included using several styles, like MARKUP and the ExcelXP tagset.

Through the SAS® Add-In for Microsoft Office, Excel users can work directly on SAS data from within their Excel sheets.

**INTRODUCTION**

Despite of a large range of Business Intelligence tools like SAS, Microsoft Excel is still the most used reporting tool. Excel lacks a lot of functionality and flexibility, which can be offered through SAS. Combining the power of SAS with Excel increases the productivity of the business users.

This paper gives a global overview of a wide range of possibilities to work with data or with information:

- using SAS data in Excel or working with Excel data in SAS.
- building reports, graphs and analytics in SAS and making the results available in Excel or working in Excel, executing tasks and procedures in SAS and returning the results back into Excel.

An overview will be given of different solutions and a comparison between these solutions is discussed, allowing to select the most beneficial one.

**ACCESSING DATA****SAS/ACCESS LIBNAME ENGINE**

A SAS libref can be assigned to non-SAS data. It is not restricted to RDBMSs data like Oracle or DB2, but is also available for Microsoft Excel files. The libref points to the ".xls" or ".xlsx" file and every sheet in the file is treated as a single data set. Through the LIBNAME engine, SAS will work directly on the Excel data, without the need to build a (temporary) SAS data set.

Consider an Excel file with three sheets. To make this Excel file available to SAS, execute a LIBNAME statement. To indicate that the libref refers to non-SAS data, override the automatic default engine, by specifying "EXCEL" as engine parameter in the LIBNAME statement.

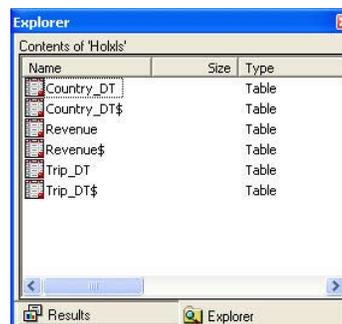
```
LIBNAME HOLXLS EXCEL "C:\SGF2009\HOLIDAYS.XLS" ;
```

HOLXLS appears in the list of active libraries in the Explorer window. The small globe on the icon indicates that the library points to external, non-SAS data.

Open the library HOLXLS. You receive the list of data sets (sheets) in the library. Notice that every sheet appears twice: once with the name as within Microsoft Excel, and once with the name with a trailing "\$". Excel has "named ranges", allowing data to be placed anywhere in a sheet. SAS reads by default from "named ranges", but can also read from spreadsheets directly. The file with trailing "\$" contains the spreadsheet, the other one contains the "named ranges".



**Active Libraries**



**Files in Library HOLXLS**

Open the data set REVENUE by selecting either "Revenue" or "Revenue\$" in the Explorer window. The content of the Excel sheet is presented.

	Trip_Type	Continent	Country	Nbr_of_Trips	Trip_Revenue	Tour_Traveller_Count
1	Beach Holiday	Africa	Egypt	30	74345.5	
2	Beach Holiday	Asia	Maldives	38	183298.5	
3	Beach Holiday	Asia	Sri Lanka	33	99842	
4	Beach Holiday	Asia	Thailand	67	205459	
5	Beach Holiday	North America	Mexico	28	99872	
6	Tour	Africa	Egypt	69	534862	403
7	Tour	Africa	South Africa	47	611161	305
8	Tour	Asia	China	36	361029.5	219
9	Tour	Asia	Sri Lanka	42	479253	303

**Excel sheet "Revenue" in SAS**

### **Reading from and Writing to Excel**

Through the SAS/ACCESS LIBNAME engine you can read from and write to Excel files. The Excel SAS/ACCESS LIBNAME engine cannot simply replace an existing sheet. To replace a sheet in an Excel file, first remove the sheet, using the DATASETS procedure for example. To add data to an existing sheet, specify the option SCAN\_TEXT = NO in the LIBNAME statement.

Within the next example, a new sheet named "Continent\_DT" is created in the Excel file, starting from another sheet, named "Country\_DT" in the same Excel file.

```
PROC SORT DATA = HOLXLS.COUNTRY_DT (KEEP = CONTINENT_CODE)
      OUT = HOLXLS.CONTINENT_DT NODUPKEY;
      BY CONTINENT_CODE;
RUN;
```

The result is verified in Microsoft Excel.

	A	B	C	D
1	Continent_Code			
2	AF			
3	AS			
4	EU			
5	NA			
6	PA			
7	SA			
8				

*Excel sheet "Continent\_DT"*

### **Combining Excel Data with SAS Data**

Consider the following data: the Microsoft Excel file - on which the libref HOLXLS is assigned – contains a translation table named COUNTRY\_DT, with country names and country codes. A SAS table named AT\_COUNTRY contains revenue data per country code.

	Country_Code	Continent_Code	Country_E
1	ARG	SA	Argentina
2	AUS	PA	Australia
3	AUT	EU	Austria
4	BHS	SA	Bahamas
5	BEL	EU	Belgium
6	BRA	SA	Brazil
7	KHM	AS	Cambodia
8	CAN	NA	Canada
9	CAF	AF	Central African Republic

*Excel sheet "Country\_DT" in Holidays.xls*

	Country	Base_Price	Child_Discount	Single_Supplement
1	CHN	1,029,148	9,920	8,385
2	CUB	1,505,506	5,050	13,815
3	ECU	1,528,085	15,570	33,670
4	EGY	1,941,538	18,447	23,790
5	IDN	1,465,403	10,377	24,500
6	IND	1,774,748	10,265	15,090
7	LKA	1,716,751	18,416	18,494
8	MDV	669,264	7,806	13,404
9	MEX	1,880,283	17,289	17,518

*SAS data set AT\_COUNTRY*

Use these two data sources to build a new SAS data set, showing revenue by country.

```
PROC SQL;
  CREATE TABLE COUNTRY_REVENUE AS
  SELECT COUNTRY_E AS COUNTRY_NAME,
         BASE_PRICE, CHILD_DISCOUNT, SINGLE_SUPPLEMENT
  FROM   BIKS.AT_COUNTRY S,
         HOLXLS.COUNTRY_DT X
  WHERE  S.COUNTRY = X.COUNTRY_CODE;
QUIT;
```

	Country_E	Base_Price	Child_Discount	Single_Supplement
1	China	1,029,148	9,920	8,385
2	Cuba	1,505,506	5,050	13,815
3	Ecuador	1,528,085	15,570	33,670
4	Egypt	1,941,538	18,447	23,790
5	India	1,774,748	10,265	15,090
6	Indonesia	1,465,403	10,377	24,500
7	Maldives	669,264	7,806	13,404
8	Mexico	1,880,283	17,289	17,518
9	Peru	2,351,354	11,655	26,705
10	South Africa	1,889,286	8,545	8,900
11	Sri Lanka	1,716,751	18,416	18,494

**SAS data set COUNTRY\_REVENUE**

## THE IMPORT AND EXPORT PROCEDURES

Access to SAS data in a SAS session or SAS batch program is always faster than access to non-SAS data. When an application needs to process the same Excel data multiple times, it might be useful to create a copy of the Excel file into a real SAS data set. The IMPORT procedure creates a SAS data set from an Excel sheet; the EXPORT procedure creates an Excel sheet from a SAS data set.

Both procedures can be invoked by executing SAS code, using the Import Wizard or Export Wizard in a SAS session or by using the Import Data task in SAS/Enterprise Guide®.

### Importing Data into SAS

Consider the Excel file "BIKS\_Info.xls", containing several sheets.

	A	B	C	D	E	F
1	Item_Code	Item_E	Item_D	Item_F		
2	E	Excursion	Excursie	Excursion		
3	F	Flight	Vlucht	Vol		
4	H	Hotel	Hotel	Hôtel		
5	T	Tour	Rondreis	Circuit		
6	X	Extension	Uitbreiding	Extension		
7						

**Excel Sheet "Trip\_Item\_Codes"**

Execute the IMPORT procedure to create a SAS data set WORK.ITEM\_DIMENSION from the sheet named "Trip\_Item\_Codes".

```
PROC IMPORT OUT = WORK.ITEM_DIMENSION
            DBMS = EXCEL REPLACE
            FILE = "C:\SGF2009\BIKS_INFO.XLS" ;
            SHEET = "TRIP_ITEM_CODES" ;
RUN;
```

- FILE = names the source Excel file.
- OUT = names the SAS data set to create.
- DBMS = EXCEL indicates that the IMPORT procedure treats Excel data.

- The REPLACE option indicates to override the SAS data set, if it already exists.
- By default SAS will create the SAS data set, based on the first sheet in the Excel file. Specify a SHEET statement to process another sheet.

	Item_Code	Item_E	Item_D	Item_F
1	E	Excursion	Excursie	Excursion
2	F	Flight	Vlucht	Vol
3	H	Hotel	Hotel	Hôtel
4	T	Tour	Rondreis	Circuit
5	X	Extension	Uitbreiding	Extension

**SAS data set ITEM\_DIMENSION**

Many spreadsheets contain columns with mixed data entry (character values and numeric values). When reading data from Excel, SAS scans every column to determine the data type.

Consider another Excel sheet, containing a column with a mix of numeric and character data (column named "BIKS\_Code").

	A	B	C
1	Trip_Country	BIKS_Code	Date
2	Egypt	A10	18-Jan-09
3	Thailand	20	25-Feb-09
4	Cuba	Q70	14-May-09
5	Mexico	Q80	18-Jun-09
6	Vietnam	23	15-Nov-09

**Excel sheet "Country"**

Create a SAS data set WORK.COUNTRY\_INFO from this Excel sheet, using the IMPORT procedure.

```
PROC IMPORT OUT = COUNTRY_INFO
            DBMS = EXCEL REPLACE
            FILE = "C:\SGF2009\COUNTRY_INFO.XLS" ;
            SHEET = "COUNTRY" ;
RUN;
```

Examining the result shows that not all data are imported! For the numeric values in the column "BIKS\_Code", a missing value appears in the SAS data set.

	Trip_Country	BIKS_Code	Date
1	Egypt	A10	18JAN2009
2	Thailand	.	25FEB2009
3	Cuba	Q70	14MAY2009
4	Mexico	Q80	18JUN2009
5	Vietnam	.	15NOV2009

**SAS data set COUNTRY\_INFO**

To solve this problem / error, you can specify a MIXED = YES statement in the IMPORT procedure. Notice that the option MIXED = YES can also be specified in a LIBNAME statement, using the Excel engine.

```
PROC IMPORT OUT = COUNTRY_INFO
  DBMS = EXCEL REPLACE
  FILE = "C:\SGF2009\COUNTRY_INFO.XLS" ;
  SHEET = "COUNTRY" ;
  MIXED = YES ;
RUN ;
```

	Trip_Country	BIKS_Code	Date
1	Egypt	A10	18JAN2009
2	Thailand	20	25FEB2009
3	Cuba	Q70	14MAY2009
4	Mexico	Q80	18JUN2009
5	Vietnam	23	15NOV2009

**SAS data set COUNTRY\_INFO**

### Exporting SAS Data Sets to Excel

The EXPORT procedure enables creating Excel data from within a SAS session or SAS program.

Consider the SAS dataset BIKS.REVENUE\_FACTS. Use the EXPORT procedure to build a sheet named "Global Revenue" in the Excel file named "Revenue\_Facts.xls" from this SAS data set.

	Trip_ID	Trip_Item_ID	Traveller_ID	Trip_Departure	Revenue
1	B/EGY-01*20041208	E/EGY-F-11	NLD_4743_1	08DEC2004	27
2	B/EGY-01*20041208	F/HRG	NLD_4743_1	08DEC2004	32C
3	B/EGY-01*20041208	H/EGY-E-01	NLD_4743_1	08DEC2004	53E
4	B/EGY-01*20041208	E/EGY-F-11	NLD_4743_2	08DEC2004	27
5	B/EGY-01*20041208	F/HRG	NLD_4743_2	08DEC2004	32C
6	B/EGY-01*20041208	H/EGY-E-01	NLD_4743_2	08DEC2004	53E
7	B/EGY-01*20041216	E/EGY-F-09	NLD_1055_1	16DEC2004	23
8	B/EGY-01*20041216	E/EGY-F-10	NLD_1055_1	16DEC2004	42
9	B/EGY-01*20041216	E/EGY-F-11	NLD_1055_1	16DEC2004	27

**SAS Table BIKS.REVENUE\_FACTS (partial)**

```
PROC EXPORT DATA = BIKS.REVENUE_FACTS
  DBMS = EXCEL REPLACE
  FILE = "C:\SGF2009\REVENUE_FACTS.XLS" ;
  SHEET = "GLOBAL REVENUE" ;
RUN ;
```

- FILE = names the Excel file to create or to modify.
- REPLACE indicates to override the sheet, when it already exists.
- By default the sheet receives the name of the SAS data set as label. Use the SHEET = statement to assign another label to the sheet.
- SAS date values are presented as dates in Excel.

	A	B	C	D	E
1	Trip_ID	Trip_Item_ID	Traveller_ID	Trip_Departure	Revenue
2	B/EGY-01*20041208	E/EGY-F-11	NLD_4743_1	12/8/2004	27
3	B/EGY-01*20041208	F/HRG	NLD_4743_1	12/8/2004	320
4	B/EGY-01*20041208	H/EGY-E-01	NLD_4743_1	12/8/2004	536
5	B/EGY-01*20041208	E/EGY-F-11	NLD_4743_2	12/8/2004	27
6	B/EGY-01*20041208	F/HRG	NLD_4743_2	12/8/2004	320
7	B/EGY-01*20041208	H/EGY-E-01	NLD_4743_2	12/8/2004	536
8	B/EGY-01*20041216	E/EGY-F-09	NLD_1055_1	12/16/2004	23
9	B/EGY-01*20041216	E/EGY-F-10	NLD_1055_1	12/16/2004	42
10	B/EGY-01*20041216	E/EGY-F-11	NLD_1055_1	12/16/2004	27
11	B/EGY-01*20041216	F/HRG	NLD_1055_1	12/16/2004	320
12	B/EGY-01*20041216	H/EGY-E-01	NLD_1055_1	12/16/2004	713
13	B/EGY-01*20041216	E/EGY-F-09	NLD_1055_2	12/16/2004	23

Excel file "Revenue\_Facts.xls"

## THE %DS2CSV MACRO

Microsoft Excel is not available on platforms like Unix, but still users want to create output that can be treated in Excel on their PC. SAS provides the macro %DS2CSV to build ".csv" (comma separated values) output for Excel. The macro has a set of named parameter, with default values. You need to specify at least the name of the source data set (DATA = parameter), the output file name (CSVFILE = parameter) and the option RUNMODE = B.

Use the %DS2CSV macro to create a ".csv" file from the SAS data set BIKS.HOLIDAYS.

```
%DS2CSV (CSVFILE = "C:\SGF2009\DS_HOLIDAYS.CSV",
        DATA    = BIKS.HOLIDAYS,
        RUNMODE  = B);
```

Partial file "DS\_Holidays.csv"

```
"Trip_Description","Trip_Type","Continent","Country","Trip_Revenue","Tour_Traveller_Count","Tour_Score"
"Discovery of China","Tour","Asia","China","10380.5","7","57"
"Big Tour of China","Tour","Asia","China","6642","4","79"
"Discovery of China","Tour","Asia","China","11586","8","75"
"Big Tour of China","Tour","Asia","China","7072","4","82"
"Discovery of China","Tour","Asia","China","5370","4","52"
"Big Tour of Cuba","Tour","South America","Cuba","6542","4","58"
"Discovery of Cuba","Tour","South America","Cuba","10881","6","39"
```

The following example illustrates a few other parameters: use the SAS data set named BIKS.HOLIDAYS to create the Excel file named "DS\_Holidays.csv". Modify the default output through parameters in the macro call:

- When the sheet already exists, override its content with new data. (OPENMODE = option)
- By default the first line in the output file contains the name of the variables. Within this example this first line is not generated. (COLHEAD = option)
- By default all variables of the SAS data set are written to the output file, in the same sequence as their sequence in the observations. Use the VAR = parameter to select the variables to export and to define their sequence in the output file.
- By default all observations are exported. Specify a WHERE = parameter to subset the data to records with the value "Thailand" for the variable COUNTRY.

```
%DS2CSV (CSVFILE = "C:\SGF2009\DS_HOLIDAYS.CSV",
        OPENMODE = REPLACE,
        COLHEAD = N,
        VAR      = TRIP_DESCRIPTION TRIP_TYPE TRIP_REVENUE,
        DATA    = BIKS.HOLIDAYS,
        WHERE    = COUNTRY = 'Thailand',
        RUNMODE  = B);
```

**Partial file "DS\_Holidays.csv"**

```
"Discovery of Thailand","Tour","7750"
"Discovery of Thailand","Tour","5939"
"Big Tour of Thailand","Tour","8231.5"
"Discovery of Thailand","Tour","5151"
"Big Tour of Thailand","Tour","8859"
"Discovery of Thailand","Tour","5342.5"
"Discovery of Thailand","Tour","7197"
"Discovery of Thailand","Tour","7164"
"Discovery of Thailand","Tour","10096"
"Discovery of Thailand","Tour","15161"
"Discovery of Thailand","Tour","6302"
```

## CREATING RESULTS IN EXCEL FORMAT

Instead of exchanging data between SAS and Excel, reports, graphs or analysis results can be prepared in SAS and published in Excel. The business users can concentrate on analyzing the results in Excel instead of spending (too much) time on working on the lay-out of the information. Moreover, SAS is more self-documenting than Excel, which makes maintenance a lot easier.

## THE OUTPUT DELIVERY SYSTEM (ODS)

By default SAS generates output to the LISTING destination. This output uses mono-space fonts, and was designed for traditional line-printers. The Output Delivery System (ODS) introduces more flexibility in output generation with a wide range of formatting options. ODS enables generating output that is treated by other applications like Excel, Acrobat, Web Browsers, etc.

ODS allows creating output to different destinations. To start building output for a specific destination, execute:

```
ODS destination <FILE = output_file> <additional_parameters>;
```

To stop generating output to a specific destination, execute:

```
ODS destination CLOSE;
```

- *destination* names the output destination. Valid values include: RTF, PDF, HTML, PRINTER, LISTING, CSV, MARKUP, ...
- *FILE = output\_file* names the physical name of the output file. This can be replaced by a *FILEREF =* option.
- *additional\_parameters* (optional) specify for example a style to use. A style is a template describing the formatting of the output. The *STYLE =* option can refer to SAS supplied styles or user-defined styles.

The following examples generate two reports, for which the code is defined in a macro:

```
%MACRO TABRPT;
  TITLE "Trip Revenue";
  PROC TABULATE DATA = BIKS.HOLIDAYS;
    CLASS CONTINENT TRIP_TYPE;
    VAR    TRIP_REVENUE;
```

```

TABLE CONTINENT = " ",
TRIP_TYPE = " " * TRIP_REVENUE = " " * SUM = " " * F = COMMA11.;
RUN;
%MEND;

%MACRO REPRPT;
TITLE "Revenue Analysis per Country";
PROC REPORT DATA = BIKS.HOLIDAYS_AG NOWD;
WHERE TRIP_TYPE LIKE "T%";
COLUMNS COUNTRY TRIP_REVENUE NBR_OF_TRIPS AVG_REVENUE;
DEFINE COUNTRY / GROUP;
DEFINE TRIP_REVENUE / SUM FORMAT = COMMA8.;
DEFINE NBR_OF_TRIPS / SUM;
DEFINE AVG_REVENUE / COMPUTED FORMAT = COMMA8.;
COMPUTE AVG_REVENUE;
AVG_REVENUE = TRIP_REVENUE.SUM / NBR_OF_TRIPS.SUM;
ENDCOMP;
RUN;
%MEND;

```

### The ODS CSV Destination

Use the ODS CSV destination to create two reports, and examine the result in Excel.

```

ODS CSV FILE = "C:\SGF2009\BIKS_CSV.CSV";
%TABRPT;
%REPRPT;
ODS CSV CLOSE;

```

- The ODS CSV statement opens the CSV destination and names the output file. All statements following the ODS statement will generate their output to that destination, until the ODS CSV CLOSE statement is encountered.
- The ODS CSV CLOSE statement ends the process of output generation to this destination.

	A	B	C	D
1		Beach Holiday	Tour	
2	Africa	74346	1146023	
3	Asia	487600	2660959	
4	North America	99872	586675	
5	South America		1599534	
6				
7	Country	Trip_Revenue	Nbr_of_Trips	Avg_Revenue
8	China	361030	36	10029
9	Cuba	513039	39	13155
10	Ecuador	513411	27	19015
11	Egypt	534862	69	7752
12	Mexico	586675	42	13968
13	Peru	573085	47	12193
14	South Africa	611161	47	13003
15	Sri Lanka	479253	42	11411
16	Thailand	1810677	150	12071
17				

*ODS CSV output in Excel*

- Notice that the result of both procedures is written to one Excel sheet.
- Notice that the result of the TITLE statements is not written to the CSV file. Using ODS CSVALL instead of ODS CSV will add the titles.

### The ODS MARKUP Destination

The ODS MARKUP destination is used to create XML from SAS output. This XML can be used in a Web Browser or in Microsoft Excel. The TAGSET = option in the ODS MARKUP statement specifies the value of a tagset. A tagset is a template that defines how to create a markup language output type. Valid values include HTML, CSV, CSVALL.

The example below illustrates the creation of a “.csv” file, containing two reports – with titles – within one sheet.

```
ODS MARKUP TAGSET = CSVALL FILE = "C:\SGF2009\BIKS_CSVALL.CSV" ;
%TABRPT ;
%REPRPT ;
ODS MARKUP CLOSE ;
```

	A	B	C	D
1	Trip Revenue			
2				
3		Beach Holiday Tour		
4	Africa	74346	1146023	
5	Asia	487600	2650959	
6	North America	99872	586675	
7	South America		1599534	
8				
9	Revenue Analysis per Country			
10				
11	Country	Trip_Revenue	Nbr_of_Trips	Avg_Revenue
12	China	361030	36	10029
13	Cuba	513039	39	13155
14	Ecuador	513411	27	19015
15	Egypt	534862	69	7752
16	Mexico	586675	42	13968
17	Peru	573085	47	12193
18	South Africa	611161	47	13003
19	Sri Lanka	479253	42	11411
20	Thailand	1810677	150	12071
21				

ODS MARKUP output in Excel

### The ODS Tagsets.ExcelXP Destination

The ODS TAGSETS.EXCELXP destination allows generating multiple worksheets per Excel file. This ODS destination generates output in the SpreadsheetML format. The EXCELXP destination allows additional parameters to improve the layout of the output.

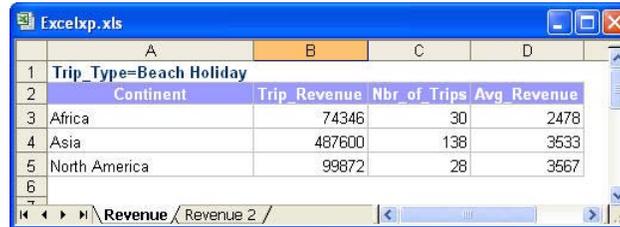
The examples for the ODS TAGSETS.EXCELXP build a report using the REPORT procedure, executing the following statements. A BY statement indicates to build a report for each value of the variable “TRIP\_TYPE”:

```
%MACRO RPRT ;
TITLE "Revenue Analysis per Country" ;
PROC REPORT DATA = BIKS.HOLIDAYS_AG NOWD ;
  BY TRIP_TYPE ;
  COLUMNS CONTINENT TRIP_REVENUE NBR_OF_TRIPS AVG_REVENUE ;
  DEFINE CONTINENT / GROUP ;
  DEFINE TRIP_REVENUE / SUM FORMAT = COMMA8. ;
  DEFINE NBR_OF_TRIPS / SUM ;
  DEFINE AVG_REVENUE / COMPUTED FORMAT = COMMA8. ;
  COMPUTE AVG_REVENUE ;
    AVG_REVENUE = TRIP_REVENUE.SUM / NBR_OF_TRIPS.SUM ;
  ENDCOMP ;
RUN ;
%MEND ;
```

Create an Excel sheet using the ODS TAGSETS.EXCELXP destination. As a result of the BY statement (with two distinct values for TRIP\_TYPE) two sheets are created, one for each value of the BY-variable. The BY-line is shown

in each sheet. Through the OPTIONS option, the name of the first sheet is specified ("Revenue"). Subsequent sheets receive the same name, followed by a sequence number.

```
ODS TAGSETS.EXCELXP FILE = "C:\SGF2009\EXCELXP.XLS"
                      STYLE = SASWEB
                      OPTIONS (SHEET_NAME = 'Revenue');
%RPRT;
ODS TAGSETS.EXCELXP CLOSE;
```

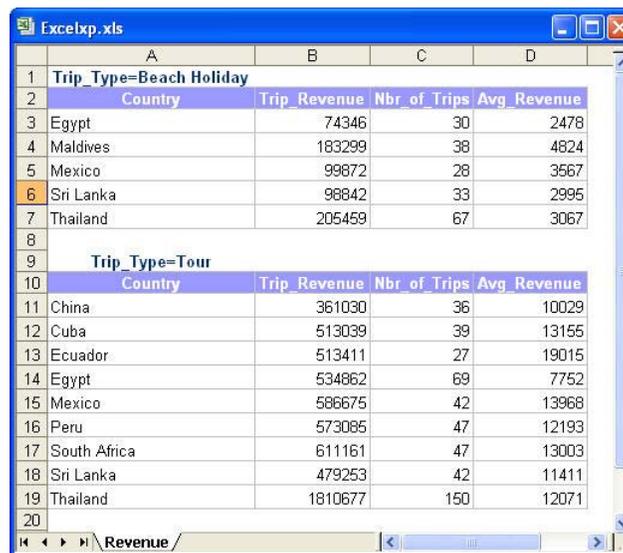


Trip_Type=Beach Holiday			
Continent	Trip Revenue	Nbr of Trips	Avg Revenue
Africa	74346	30	2478
Asia	487600	138	3533
North America	99872	28	3567

*ODS Tagsets.ExcelXP output in Excel*

Modify the options in the ODS statement. The SHEET\_INTERVAL = "NONE" option indicates that, although a BY statement is used, all output must be created in one sheet.

```
ODS TAGSETS.EXCELXP FILE = "C:\SGF2009\EXCELXP.XLS"
                      STYLE = SASWEB
                      OPTIONS (SHEET_INTERVAL = "NONE"
                              SHEET_NAME      = "Revenue");
%RPRT;
ODS TAGSETS.EXCELXP CLOSE;
```



Trip_Type=Beach Holiday			
Country	Trip Revenue	Nbr of Trips	Avg Revenue
Egypt	74346	30	2478
Maldives	183299	38	4824
Mexico	99872	28	3567
Sri Lanka	98842	33	2995
Thailand	205459	67	3067

Trip_Type=Tour			
Country	Trip Revenue	Nbr of Trips	Avg Revenue
China	361030	36	10029
Cuba	513039	39	13155
Ecuador	513411	27	19015
Egypt	534862	69	7752
Mexico	586675	42	13968
Peru	573085	47	12193
South Africa	611161	47	13003
Sri Lanka	479253	42	11411
Thailand	1810677	150	12071

*ODS Tagsets.ExcelXP output in Excel*

A final example on the ODS TAGSETS.EXCELXP destination specifies options to:

- Allow users to filter on each column within Excel (the AUTOFILTER = option).
- Keep the header lines presented when scrolling down in the Excel sheet.

```

ODS TAGSETS.EXCELXP FILE = "C:\SGF2009\EXCELXP.XLS"
                        STYLE = SASWEB
                        OPTIONS (AUTOFILTER      = "ALL"
                                FROZEN_HEADERS  = "YES"
                                SHEET_NAME     = "Revenue");
TITLE "Revenue Analysis per Country";
PROC PRINT DATA = BIKS.HOLIDAYS_AG NOOBS;
  VAR TRIP_TYPE CONTINENT COUNTRY TRIP_REVENUE;
  FORMAT TRIP_REVENUE COMMA8.;
RUN;
ODS TAGSETS.EXCELXP CLOSE;

```

Review the output in Excel and scroll down three lines. Notice that the headers are frozen.

	A	B	C	D
1	Trip Type	Continent	Country	Trip Reveni
4	Beach Holiday	Asia	Sri Lanka	98842
5	Beach Holiday	Asia	Thailand	205459
6	Beach Holiday	North America	Mexico	99872
7	Tour	Africa	Egypt	534862
8	Tour	Africa	South Africa	611161
9	Tour	Asia	China	361030
10	Tour	Asia	Sri Lanka	479253
11	Tour	Asia	Thailand	1810677
12	Tour	North America	Mexico	586675
13	Tour	South America	Cuba	513039
14	Tour	South America	Ecuador	513411
15	Tour	South America	Peru	573085

**ODS Tagsets.ExcelXP output in Excel**

Use the drop down icons to filter the Excel on TRIP\_TYPE = "Tour" and CONTINENT = "Asia".

	A	B	C	D
1	Trip Type	Continent	Country	Trip Reveni
9	Tour	Asia	China	361030
10	Tour	Asia	Sri Lanka	479253
11	Tour	Asia	Thailand	1810677
16				

3 of 14 records found

**ODS Tagsets.ExcelXP output in Excel**

### **The ODS HTML Destination**

The ODS HTML destination is by default used to build output to view in a Web Browser. By specifying the ".xls" or ".xlsx" file type suffix in the file name, an Excel output is created.

The following example illustrates the creation of a PROC REPORT output, enhanced with exception highlighting by using formats.

```

PROC FORMAT;
  VALUE BACK    0 - 550000 = "RED"
              650000 - HIGH = "GREEN"
              OTHER = "WHITE";

ODS HTML FILE = "C:\SGF2009\REPORT.XLS"
  STYLE = SASDOCPRINTER;
PROC REPORT DATA = BIKS.HOLIDAYS NOWD;
  COLUMN CONTINENT COUNTRY TRIP_REVENUE;
  DEFINE CONTINENT / GROUP;
  DEFINE COUNTRY / GROUP;
  DEFINE TRIP_REVENUE / ANALYSIS SUM FORMAT = COMMA12. "Revenue"
  STYLE (COLUMN) = {BACKGROUND = BACK.};

RUN;
ODS HTML CLOSE;

```

Continent	Country	Revenue
Africa	Egypt	609,208
	South Africa	611,161
Asia	China	361,030
	Maldives	183,293
	Sri Lanka	578,095
	Thailand	2,016,136
North America	Mexico	886,547
South America	Cuba	513,033
	Ecuador	513,411
	Peru	573,085

**ODS HTML to Excel file, with Exception Highlighting**

The STYLE (COLUMN) option specifies the background color of the TRIP\_REVENUE column. The calculated value is translated using a format, named BACK (translating 2016136 into "GREEN" for example). As a result the Excel file is created with a dynamic color layout, based on the cell values.

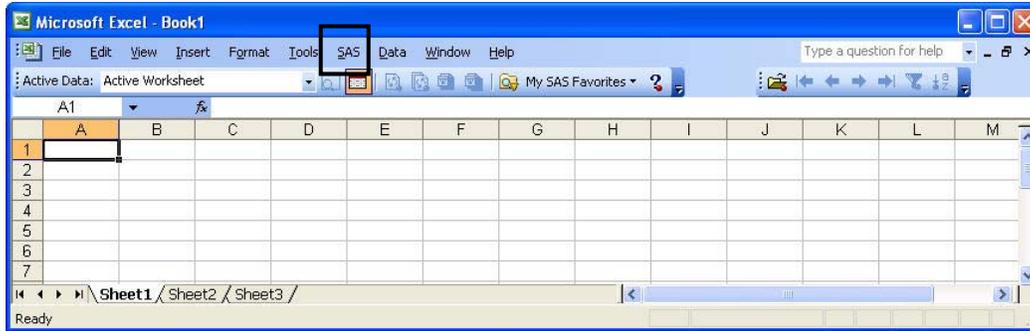
## SAS® ADD-IN FOR MICROSOFT OFFICE

SAS Add-In for Microsoft Office is an add-in that enables users to transparently leverage the power of SAS directly from Excel, Word or Powerpoint. It allows Excel users to access directly data in SAS data sets or provides Excel users access to SAS procedures to be executed on SAS data sets, returning results into their Excel files.

### ACCESSING SAS DATA IN EXCEL

SAS Add-In for Microsoft Office is capable of accessing very large data sources, eventually bypassing the data size limit of Excel.

SAS Add-In for Microsoft Office is installed as an integrated menu option within Microsoft Office applications (Excel, Word, Powerpoint). The traditional standard menu is extended with an option named "SAS".



*Excel with SAS® Add-In for Microsoft Office installed*

SAS tasks are also available through two SAS toolbars.



*SAS Data Analysis Toolbar*

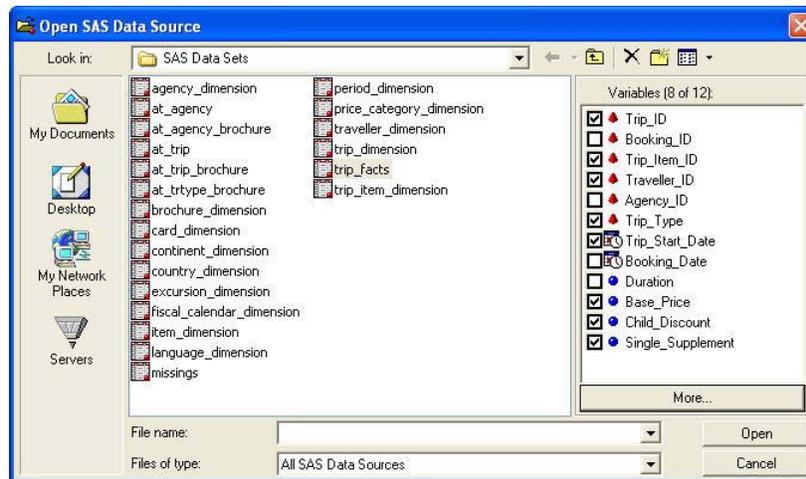


*SAS Analysis Tools Toolbar*

### **Example: Working with SAS Data Sets**

The SAS data set TRIP\_FACTS contains 12 variables and more than 150,000 observations. Analyze this SAS data set in Excel.

- Select the Open icon on the SAS toolbar.
- Navigate to the location of the SAS data and select the data set. The list of variables of the data set is shown in the right list box. By default all variables are selected.
- Deselect the variables that are not required to be shown in Excel.
- Select the Open button to display the data in Excel.



	A	B	C	D	E	F	G
4991	4990	B/THA-03*20050824	F/HKT	NLD_3148_1	B	24-Aug-2005	630
4992	4991	B/THA-03*20050824	H/THA-KL-01	NLD_3148_1	B	24-Aug-2005	1034
4993	4992	B/THA-03*20050904	F/HKT	NLD_1177_1	B	04-Sep-2005	604
4994	4993	B/THA-03*20050904	H/THA-KL-02	NLD_1177_1	B	04-Sep-2005	894
4995	4994	B/THA-03*20050904	F/HKT	NLD_1177_2	B	04-Sep-2005	604
4996	4995	B/THA-03*20050904	H/THA-KL-02	NLD_1177_2	B	04-Sep-2005	894
4997	4996	B/THA-03*20050906	F/HKT	NLD_1618_1	B	06-Sep-2005	604
4998	4997	B/THA-03*20050906	H/THA-KL-02	NLD_1618_1	B	06-Sep-2005	860
4999	4998	B/THA-03*20050906	F/HKT	NLD_1618_2	B	06-Sep-2005	604
5000	4999	B/THA-03*20050906	H/THA-KL-02	NLD_1618_2	B	06-Sep-2005	860
5001	5000	B/THA-03*20050926	F/HKT	NLD_1416_1	B	26-Sep-2005	604
5002							

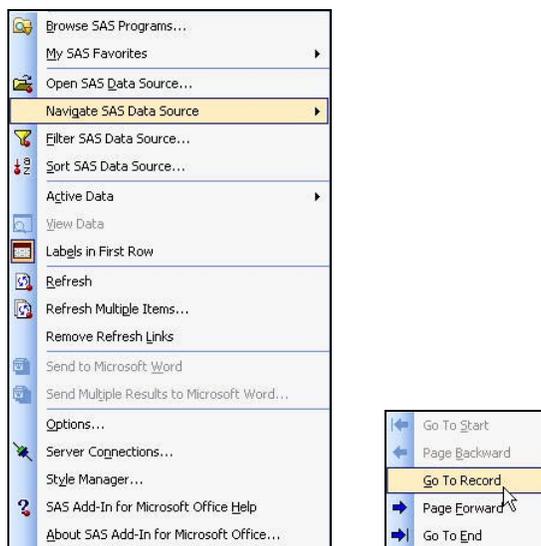
**SAS data set TRIP\_FACTS in Excel (partial)**

When the data set exceeds the data size limit of the Excel version, you can navigate easily through portions of the data set, using the SAS toolbar.



Navigating to specific parts of the SAS data set, or to a specific observation is also available from the pull-down menu:

- Select "SAS" on the pull-down menu.
- Select "Navigate SAS Data Source" from the drop-down list.
- Select an item from the pop-up list.

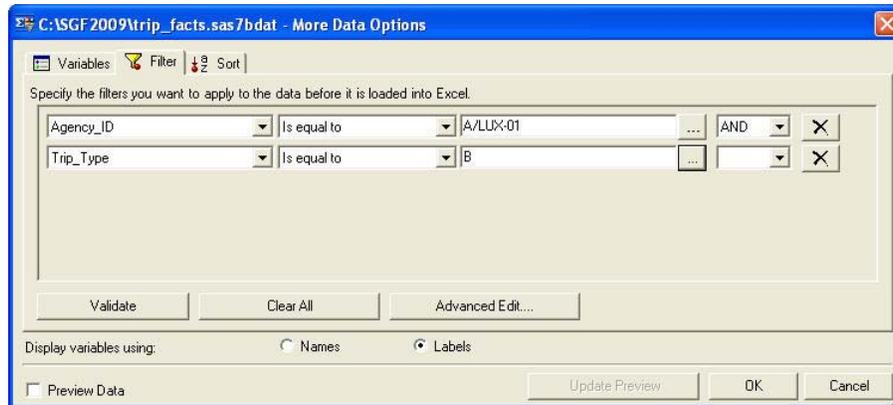


**Drop-down lists from "SAS" pull-down menu item**

### Filtering SAS Data Sets

A point-and-click interface allows building filters, applied on the SAS data set before bringing the data into Excel. Afterwards, Excel filters can be added to further subset the data in Excel.

- Select the Filter icon on the SAS Data Analysis toolbar.
- Use the drop-down arrows and selection lists to build the filter expression.



Advantages of filtering the data in SAS include:

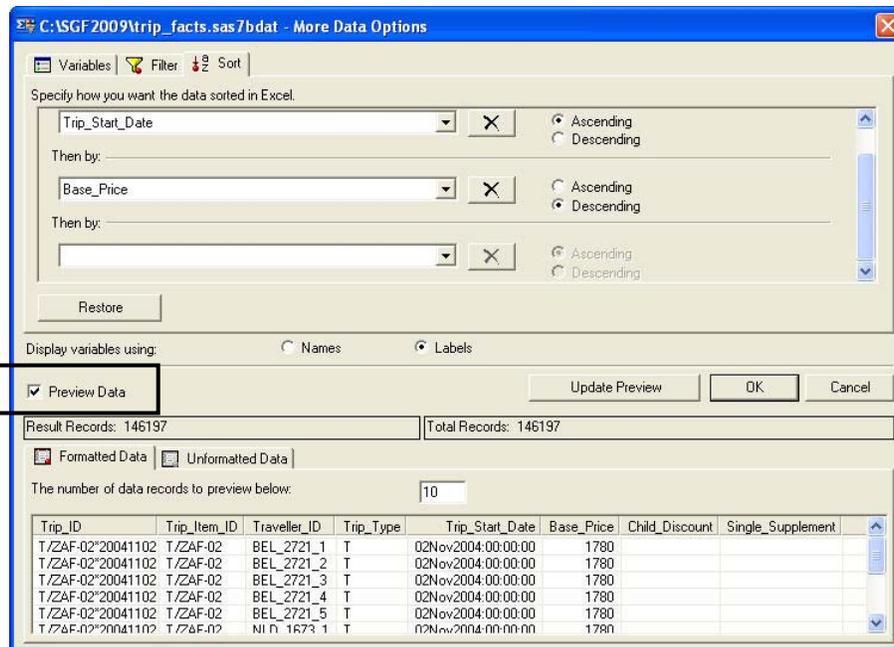
- Apply filters on variables that are not transferred to Excel.
- Build complex filters using AND and OR relations.
- The Advanced Edit... button opens the Advanced Expression Editor window enabling to type any filter expression.
- The filters are applied on SAS data sets in SAS, without any size limits.

### Sorting SAS Data Sets

Data can be sorted before presenting that data in Excel. The sort is executed in SAS and can be extended by a sort in Excel.

- Select the Sort icon on the SAS Data Analysis toolbar.
- Specify one or more variables to sort on and define the sort as ascending or descending.
- Select the Preview Data check box to get a preview of the result. The size of the subset to display in the preview can easily be adapted.





## EXECUTING SAS TASKS IN EXCEL

Business users can exploit the full power of SAS to build reports, charts, data analysis results on their data from within their familiar Excel environment. Almost all procedures are available through a point-and-click environment.

### Example

The SAS data set HOLIDAYS contains information about holiday trips from a travel agency. Use this SAS data set to build a tabular report, showing revenue figures, calculated per continent and per trip type. Execute this task from within an Excel sheet, and return the result in the same Excel file, in a new sheet.

- Open the SAS data set HOLIDAYS in Excel.

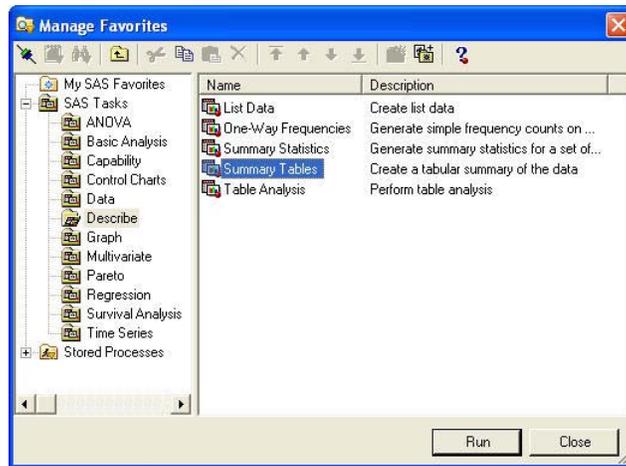
obs	Trip_Description	Trip_Type	Continent	Country	Trip_Revenue	Tour_Traveller_Count	Tour_Score
1	Discovery of China	Tour	Asia	China	10380.5	7	57
2	Big Tour of China	Tour	Asia	China	6642	4	79
3	Discovery of China	Tour	Asia	China	11586	8	75
4	Big Tour of China	Tour	Asia	China	7072	4	82
5	Discovery of China	Tour	Asia	China	5370	4	52
6	Big Tour of Cuba	Tour	South America	Cuba	6542	4	58
7	Discovery of Cuba	Tour	South America	Cuba	10881	6	39
8	Discovery of China	Tour	Asia	China	10302	7	72
9	Discovery of Cuba	Tour	South America	Cuba	6448	4	62
10	Discovery of Cuba	Tour	South America	Cuba	6448	4	62

SAS data set HOLIDAYS in Excel

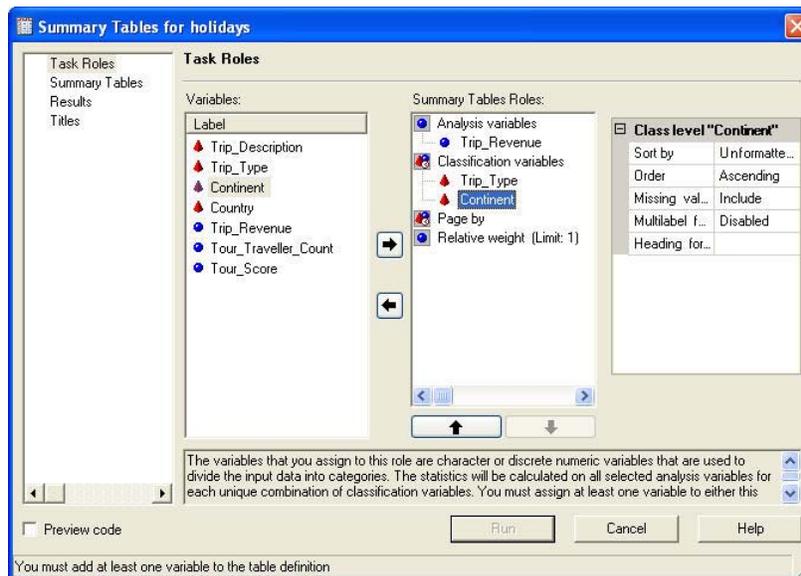
- Select My SAS Favorites from the SAS Analysis Tools toolbar
- 



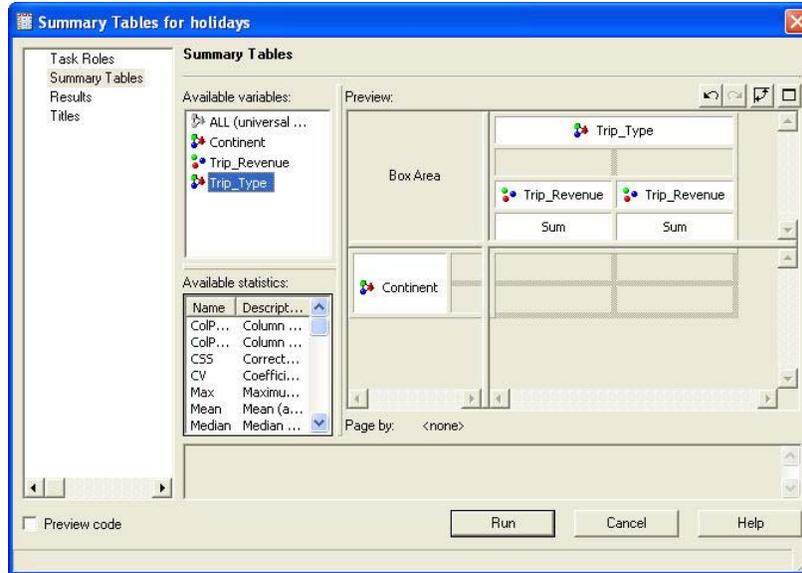
- A list of standard SAS tasks and user-defined SAS tasks is shown. (Notice that this list is comparable with the list that you receive in SAS Enterprise Guide). Select Summary Tables.



- Select Task Roles in the left list box and define variables for analysis and for classification. Optionally specify additional options for each variable.



- Select Summary Tables in the left list box. Use drag-and-drop to draw the layout of the final tabular report. Optionally request totals, subtotals, percentages, etc.



- Select the Run button to build the SAS code and execute this code. The result is shown in a new sheet in the Excel file.



**PROC TABULATE Output in Excel**

## CONCLUSION

Users have several possibilities to work with Excel data in SAS. The following table compares three methods:

Method	Advantages	Disadvantages
SAS/Access LIBNAME Engine	Easy Transparent	Performance: when data are required in several tasks, data are each time read from Excel
IMPORT / EXPORT Procedure	A SAS copy is created, with positive impact on performance.	A SAS copy is created, resulting in extra space requirements
%DS2CSV macro	Available on platforms like Unix, that do not run Microsoft Office	More complex coding

SAS Add-In for Microsoft Office allows working with (large) SAS data sets directly in Excel. Benefits include:

- Subset, sort the data in SAS.

- Navigate through large tables.

SAS ODS offers several possibilities to create information by means of reports and graphs to be displayed in Excel. Additional features and parameters enable passing display options to Excel, like freezing columns or exception highlighting.

Finally the SAS Add-In for Microsoft Office offers the full power of SAS from within Excel, to build documented, powerful applications.

## CONTACT INFORMATION

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

Theuwissen Henri  
BI Knowledge Sharing  
Sterrenlaan 40  
B-3360 Bierbeek  
Belgium  
Phone: +32 495 54 52 53  
Fax: +32 16 46 37 74  
E-mail: [Henri.Theuwissen@BIKnowledgeSharing.be](mailto:Henri.Theuwissen@BIKnowledgeSharing.be)  
Web: [www.BIKnowledgeSharing.be](http://www.BIKnowledgeSharing.be)

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.