

Maintaining Formats when Exporting Data from SAS into Microsoft Excel

Nate Derby & Colleen McGahan

Stakana Analytics, Seattle, WA
BC Cancer Agency, Vancouver, BC

Club des Utilisateurs SAS de Québec
11/1/16

Outline

- 1 Introduction
- 2 Solutions
 - The ExcelXP Tagset
 - Dynamic Data Exchange (DDE)
 - The LIBNAME Engine
- 3 Custom Formatting
 - Exporting SAS Data into Excel
 - Optimal Methods: What Do We Want to Do?
- 4 Conclusions

Introduction

Many typical ways of exporting data from SAS into Excel destroy the data formats.

Creating Data Formats

```
DATA class;  
  SET sashelp.class;  
  FORMAT age 3. height weight 6.2;  
  IF name = 'Thomas' THEN age = .;  
RUN;
```

SAS Dataset

	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.00	112.50
2	Alice	F	13	56.50	84.00
3	Barbara	F	13	65.30	98.00
4	Carol	F	14	62.80	102.50
5	Henry	M	14	63.50	102.50
6	James	M	12	57.30	83.00
7	Jane	F	12	59.80	84.50
8	Janet	F	15	62.50	112.50
9	Jeffrey	M	13	62.50	84.00
10	John	M	12	59.00	99.50
11	Joyce	F	11	51.30	50.50
12	Judy	F	14	64.30	90.00
13	Louise	F	12	56.30	77.00
14	Mary	F	15	66.50	112.00
15	Philip	M	16	72.00	150.00
16	Robert	M	12	64.80	128.00
17	Ronald	M	15	67.00	133.00

Exporting SAS Data

Now let's export it via PROC EXPORT and the ExcelXP tagset:

SAS Code

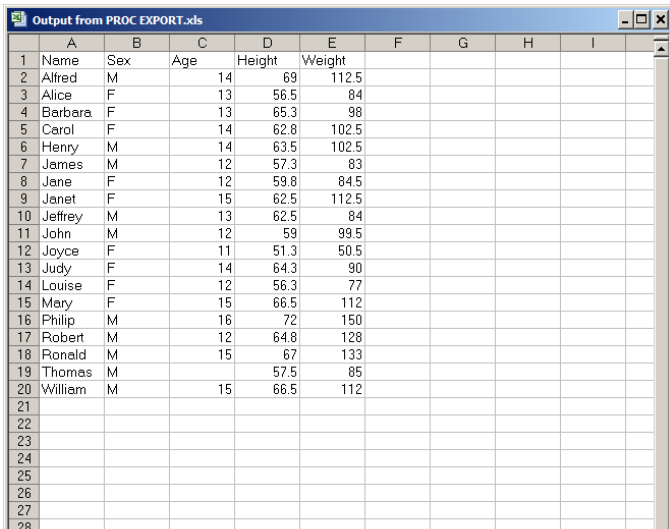
```
PROC EXPORT DATA=class
  OUTFILE="&outroot\Output from PROC EXPORT.xls";
RUN;

ODS tagsets.ExcelXP
  FILE="&outroot\Output from ExcelXP.xls";

PROC PRINT DATA=class;
RUN;

ODS tagsets.ExcelXP CLOSE;
```

PROC EXPORT Output



	A	B	C	D	E	F	G	H	I
1	Name	Sex	Age	Height	Weight				
2	Alfred	M	14	69	112.5				
3	Alice	F	13	56.5	84				
4	Barbara	F	13	65.3	98				
5	Carol	F	14	62.8	102.5				
6	Henry	M	14	63.5	102.5				
7	James	M	12	57.3	83				
8	Jane	F	12	59.8	84.5				
9	Janet	F	15	62.5	112.5				
10	Jeffrey	M	13	62.5	84				
11	John	M	12	59	99.5				
12	Joyce	F	11	51.3	50.5				
13	Judy	F	14	64.3	90				
14	Louise	F	12	56.3	77				
15	Mary	F	15	66.5	112				
16	Philip	M	16	72	150				
17	Robert	M	12	64.8	128				
18	Ronald	M	15	67	133				
19	Thomas	M		57.5	85				
20	William	M	15	66.5	112				
21									
22									
23									
24									
25									
26									
27									
28									

PROC EXPORT Output

The screenshot shows a spreadsheet window titled "Output from PROC EXPORT.xls". The data is as follows:

	A	B	C	D	E	F	G	H	I
1	Name	Sex	Age	Height	Weight				
2	Alfred	M	14	69	112.5				
3	Alice	F	13	56.5	84				
4	Barbara	F	13	65.3	98				
5	Carol	F							
6	Henry	M							
7	James	M							
8	Jane	F							
9	Janet	F							
10	Jeffrey	M							
11	John	M							
12	Joyce	F							
13	Judy	F							
14	Louise	F							
15	Mary	F							
16	Philip	M							
17	Robert	M							
18	Ronald	M							
19	Thomas	M							
20	William	M							
21									
22									
23									
24									
25									
26									
27									
28									

The "Format Cells" dialog box is open, showing the "Number" tab. The "Category" list includes General, Number, Currency, Accounting, Date, Time, Percentage, Fraction, Scientific, Text, Special, and Custom. The "Sample" field displays "69". A message states: "General format cells have no specific number format." The "OK" and "Cancel" buttons are at the bottom.

ExcelXP Tagset Output



The screenshot shows an Excel spreadsheet window titled "Output from ExcelXP.xls". The spreadsheet contains a table with 18 rows of data. The columns are labeled A through F, corresponding to the variables: Obs, Name, Sex, Age, Height, and Weight. The data is as follows:

	A	B	C	D	E	F
	Obs	Name	Sex	Age	Height	Weight
1	1	Alfred	M	14	69	112.5
2	2	Alice	F	13	56.5	84
3	3	Barbara	F	13	65.3	98
4	4	Carol	F	14	62.8	102.5
5	5	Henry	M	14	63.5	102.5
6	6	James	M	12	57.3	83
7	7	Jane	F	12	59.8	84.5
8	8	Janet	F	15	62.5	112.5
9	9	Jeffrey	M	13	62.5	84
10	10	John	M	12	59	99.5
11	11	Joyce	F	11	51.3	50.5
12	12	Judy	F	14	64.3	90
13	13	Louise	F	12	56.3	77
14	14	Mary	F	15	66.5	112
15	15	Philip	M	16	72	150
16	16	Robert	M	12	64.8	128
17	17	Ronald	M	15	67	133

ExcelXP Tagset Output

The screenshot shows an Excel spreadsheet titled "Output from ExcelXP.xls". The data is organized in columns A through F:

	A	B	C	D	E	F
1		Obs	Name	Sex	Age	Height
2		1	Alfred	M	14	112.5
3		2	Alice	F	13	84
4		3	Barbara	F	13	98
5		4	Carol			2.5
6		5	Henry			2.5
7		6	James			83
8		7	Jane			4.5
9		8	Janet			2.5
10		9	Jeffrey			84
11		10	John			9.5
12		11	Joyce			0.5
13		12	Judy			90
14		13	Louise			77
15		14	Mary			12
16		15	Philip			50
17		16	Robert			28
18		17	Ronald			33

The "Format Cells" dialog box is open, showing the "Number" tab. The "Category" list includes General, Number, Currency, Accounting, Date, Time, Percentage, Fraction, Scientific, Text, Special, and Custom. The "Sample" field displays "69". A message states: "General format cells have no specific number format." The "OK" and "Cancel" buttons are visible at the bottom.

SAS Formats vs. Excel Formats

SAS Formats vs. Excel Formats

SAS format	Excel format	Excel format name
\$8.	@	Text
8.2	0.00	Number, 2 decimal places
z8.2	00000.00	(none)
percent8.2	0.00%	Percentage, 2 decimal places
mmddyy8.	mm/dd/yy	Date, type "03/14/01"
comma12.2	#, ##0.00	Number, 2 decimal places, with ...

We need to translate SAS formats into Excel formats!

ExcelXP Tagset Solution

SAS Code

```
ODS tagsets.ExcelXP  
  FILE="&outroot\Output from ExcelXP, Numeric Formatting.xls";  
  
PROC PRINT DATA=class;  
  VAR name sex age;  
  VAR height weight / STYLE={TAGATTR='format=0.00'};  
RUN;  
  
ODS tagsets.ExcelXP CLOSE;
```

ExcelXP Tagset Solution

Output from ExcelXP, Numeric Formatting.xls

	A	B	C	D	E	F
1	Obs	Name	Sex	Age	Height	Weight
2	1	Alfred	M	14	69.00	112.50
3	2	Alice	F	13	56.50	84.00
4	3	Barbara	F	13	65.30	98.00
5	4	Carol	F	14	62.80	102.50
6					63.50	102.50
7					57.30	83.00
8					59.80	84.50
9					62.50	112.50
10					62.50	84.00
11					59.00	99.50
12					51.30	50.50
13					64.30	90.00
14					56.30	77.00
15					66.50	112.00
16					72.00	150.00
17					64.80	128.00
18					67.00	133.00

Format Cells

Category: **Number**

Sample: 69.00

Decimal places: 2

Use 1000 Separator (,)

Negative numbers: -1234.10

Number is used for general display of numbers. Currency and Accounting offer specialized formatting for monetary value.

ExcelXP Tagset Solution with PROC TEMPLATE

SAS Code

```
PROC TEMPLATE;  
  DEFINE STYLE styles.mystyle;  
    PARENT = styles.default;  
    STYLE data_num from data / TAGATTR='format:0.00';  
  END;  
RUN;  
  
ODS tagsets.ExcelXP  
  FILE="&outroot\Output from ExcelXP, Numeric Formatting.xls";  
  
PROC PRINT DATA=class;  
  VAR name sex age;  
  VAR height weight / STYLE( data )=data_num;  
RUN;  
  
ODS tagsets.ExcelXP CLOSE;
```

Dealing with Missing Values

SAS Code

```
OPTIONS MISSING='';

ODS tagsets.ExcelXP
  FILE="&outroot\Output from ExcelXP, Numeric Formatting.xls";

PROC PRINT DATA=class;
  VAR name sex age;
  VAR height weight / STYLE( data )=data_num;
RUN;

ODS tagsets.ExcelXP CLOSE;

OPTIONS MISSING='.';
```

Dynamic Data Exchange (DDE) Solution

- DDE = SAS opens Excel, tells it what to do.
- You have to tell Excel **every single step**.
- Best solution: The %exportToXL macro (free!).

SAS Code

```
%LET exroot = c:\...\exportToXL;  
  
OPTIONS SASAUTOS=( "&exroot" ) MAUTOSOURCE;  
  
%exportToXL( DSIN=class, SAVEPATH=&outroot,  
  SAVENAME=Output from DDE );
```

Dynamic Data Exchange (DDE) Solution

The screenshot shows an Excel spreadsheet titled "Output from DDE.xls" with a table of data. The table has columns A through M and rows 1 through 24. The data is as follows:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Name	Sex	Age	Height	Weight									
2	Alfred	M	14	69.00	112.50									
3	Alice	F	13	56.50	84.00									
4	Barbara	F	13	65.30	98.00									
5	Carol	F	14	62.80	102.50									
6	Henry	M	14	63.50	102.50									
7	James	M	12	57.30	83.00									
8	Jane	F	12	59.80	84.50									
9	Janet	F	15	62.50	112.50									
10	Jeffrey	M	13	62.50	84.00									
11	John	M	12	59.00	99.50									
12	Joyce	F	11	51.30	50.50									
13	Judy	F	14	64.30	90.00									
14	Louise	F	12	56.30	77.00									
15	Mary	F	15	66.50	112.00									
16	Philip	M	16	72.00	150.00									
17	Robert	M	12	64.80	128.00									
18	Ronald	M	15	67.00	133.00									
19	Thomas	M		57.50	85.00									
20	William	M	15	66.50	112.00									
21														
22														
23														
24														

The "Format Cells" dialog box is open, showing the "Number" category selected. The "Sample" field displays "69.00". The "Decimal places" is set to 2. The "Use 1000 Separator" checkbox is unchecked. The "Negative numbers" list shows three options: "-1234.10", "1234.10", and "(1234.10)". The "Number" category is selected in the list. The dialog box also includes tabs for "Alignment", "Font", "Border", "Patterns", and "Protection".

Number is used for general display of numbers. Currency and Accounting offer specialized formatting for monetary value.

The LIBNAME Solution

- Requires the SAS/ACCESS for PC Files package.
- We “cheat” by (manually) formatting the Excel template ahead of time.
- We then pour the data into the template.

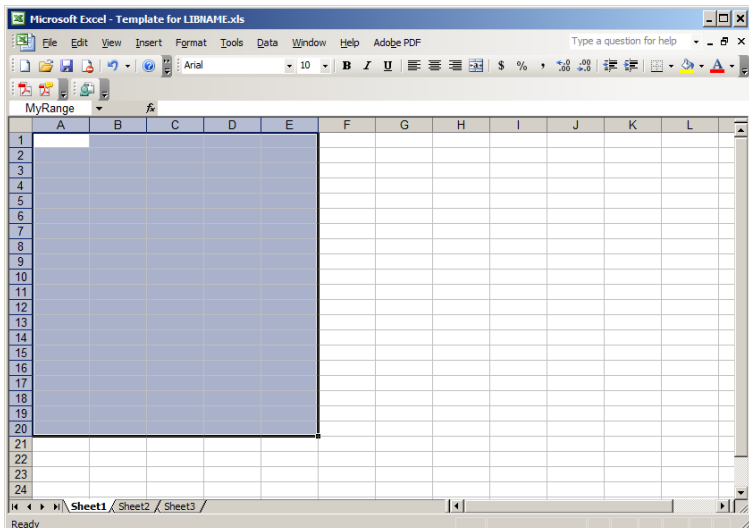
The LIBNAME Solution

The screenshot shows the Microsoft Excel interface with the 'Format Cells' dialog box open. The dialog box has several tabs: 'Number', 'Alignment', 'Font', 'Border', 'Patterns', and 'Protection'. The 'Number' tab is selected. In the 'Number' tab, the 'Category' list on the left has 'Number' selected. The 'Decimal places' is set to 2. The 'Use 1000 Separator (,)' checkbox is unchecked. Under 'Negative numbers:', three options are shown: '-1234.10' (selected), '(1234.10)', and '(1234.10)'. The background shows a spreadsheet with columns A through L and rows 1 through 24. Column D is highlighted in blue. The status bar at the bottom indicates 'Ready'.

The LIBNAME Solution

The screenshot displays the Microsoft Excel interface. The title bar reads "Microsoft Excel - Template for LIBNAME.xls". The menu bar includes "File", "Edit", "View", "Insert", "Format", "Tools", "Data", "Window", "Help", and "Adobe PDF". The toolbar shows various icons for file operations and formatting. The spreadsheet grid is visible, with a dashed box highlighting the range A1:E20. A "Define Name" dialog box is open in the foreground, showing a list of names in the workbook with "MyRange" selected. The "Refers to:" field contains the formula "=Sheet1!\$A\$1:\$E\$20". The dialog box has buttons for "OK", "Close", "Add", and "Delete". The status bar at the bottom shows "Point".

The LIBNAME Solution



The LIBNAME Solution

SAS Code

```
LIBNAME workbook PCFILES  
  PATH="&outroot\Output from LIBNAME.xls";  
  
PROC DATASETS LIBRARY=workbook NOLIST;  
  DELETE MyRange;  
QUIT;  
  
DATA workbook.MyRange;  
  SET class;  
RUN;  
  
LIBNAME workbook CLEAR;
```

The LIBNAME Solution

Output from LIBNAME.xls

	A	B	C	D	E	F	G	H	I	J	K	L
1	Name	Sex	Age	Height	Weight							
2	Alfred	M	14	69.00	112.50							
3	Alice	F	13	56.50	84.00							
4	Barbara	F	13	65.30	98.00							
5	Carol	F	14	62.80	102.50							
6	Henry	M	14	63.50	102.50							
7	James	M	12	57.30	83.00							
8	Jane	F	12	59.80	84.50							
9	Janet	F	15	62.50	112.50							
10	Jeffrey	M	13	62.50	84.00							
11	John	M	12	59.00	99.50							
12	Joyce	F	11	51.30	50.50							
13	Judy	F	14	64.30	90.00							
14	Louise	F	12	56.30	77.00							
15	Mary	F	15	66.50	112.00							
16	Philip	M	16	72.00	150.00							
17	Robert	M	12	64.80	128.00							
18	Ronald	M	15	67.00	133.00							
19	Thomas	M		57.50	85.00							
20	William	M	15	66.50	112.00							
21												
22												
23												
24												

Format Cells

Number Alignment Font Border Patterns Protection

Category: General Number Currency Accounting Date Time Percentage Fraction Scientific Text Special Custom

Sample: 69.00

Decimal places: 2

Use 1000 Separator (,)

Negative numbers: -1234.10 1234.10 (1234.10) (1234.10)

Number is used for general display of numbers. Currency and Accounting offer specialized formatting for monetary value.

OK Cancel

Many Options for Exporting SAS into Excel

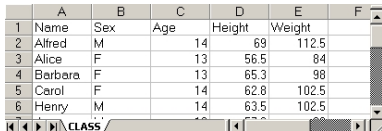
Many Options:

- PROC EXPORT
- PROC DBLOAD
- ODS HTML/MOffice2K
- ODBC
- OLE DB
- Stored Process
- Export Wizard
- DDE
- LIBNAME Statement
- via a CSV/TXT file
- ODS ExcelXP
- Add-in for MS Office

Many Options for Exporting SAS into Excel

Most allow for little/no *custom-formatting* of Excel output.

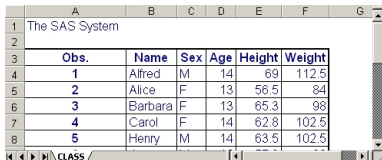
- PROC EXPORT does no formatting.
- ODS HTML does little formatting.



A screenshot of an Excel spreadsheet showing the output of PROC EXPORT. The data is presented in a standard table format with columns labeled Name, Sex, Age, Height, and Weight. The rows contain data for six individuals: Alfred, Alice, Barbara, Carol, and Henry. The spreadsheet interface includes a grid, column headers (A-F), and a status bar at the bottom.

	A	B	C	D	E	F
1	Name	Sex	Age	Height	Weight	
2	Alfred	M	14	69	112.5	
3	Alice	F	13	56.5	84	
4	Barbara	F	13	65.3	98	
5	Carol	F	14	62.8	102.5	
6	Henry	M	14	63.5	102.5	

PROC EXPORT



A screenshot of an Excel spreadsheet showing the output of ODS HTML. The data is presented in a table with a header row and a data row. The header row includes 'Obs.', 'Name', 'Sex', 'Age', 'Height', and 'Weight'. The data rows contain information for five individuals: Alfred, Alice, Barbara, Carol, and Henry. The spreadsheet interface includes a grid, column headers (A-G), and a status bar at the bottom.

	A	B	C	D	E	F	G
1	The SAS System						
2							
3	Obs.	Name	Sex	Age	Height	Weight	
4	1	Alfred	M	14	69	112.5	
5	2	Alice	F	13	56.5	84	
6	3	Barbara	F	13	65.3	98	
7	4	Carol	F	14	62.8	102.5	
8	5	Henry	M	14	63.5	102.5	

ODS HTML

Why Care about Custom Formatting?

May be perfectly fine to do no formatting:

- Just want the results (e.g., fitting statistical model).

However, special formatting is often desired:

- Sharing data with a client.
- Sharing data with a manager/executive.
- Sharing data with someone other than SAS programmer.

We can do this manually, but that takes time.

- Especially important for periodic reports!

Two Methods for Automating Custom Formatting

Two general methods:

- *Export SAS data onto pre-formatted Excel template.*
 - Needs time to (manually) set up template, but
 - Same template can be used over and over again.
- *Let SAS custom-format Excel worksheet.*
 - Needs time to write code, but
 - Same code can be used over and over again.

Choice depends on **whether template or code easier to set up code**, and a few other things ...

Criteria for Classifying Methods

Two criteria:

- **Customization:**

- *Full*: Can do (almost) anything that can be done manually.
- *Partial*: Can modify some aspects, not all.

- **Automation:**

- *Full*: Can do everything by pushing RUN.
- *Partial*: Something manually must be done afterwards.

NB: Full automation may include **manually** formatting an Excel template.

Classification of Different Methods

	Partial Customization	Full Customization
Partial Automation		Stored Processes Excel Add-In
Full Automation	ODS HTML ODS HTML _n ODS MSOffice2K	ExcelXP Tagset Dynamic Data Exchange (DDE) LIBNAME Engine

What Do We Want to Do?

	ExcelXP	DDE	LIBNAME
Work with PC SAS	X	X	X
Work with older versions of PC SAS or Excel		X	
Work with Enterprise Guide	X		X
Make side-by-side tables		X	X
Export onto a pre-formatted worksheet		X	X
Do "almost anything" to the worksheet		X	
Work with graphical output within Excel		X	X
Work "quickly"	X		X
Work without opening/installing Excel	X		X
Work with OpenOffice.org Calc	X		
Modify the code to export to other (RTF, HTML)	X		
Modify the method to suit our own purposes	Hard	Moderate	Easy

Conclusions

- Many ways of exporting data from SAS into Excel destroy data formats.
 - SAS and Excel speak different languages for data formats.
- This can be fixed in three ways:
 - ExcelXP Tagset with the TAGATTR style.
 - Dynamic Data Exchange with %exportToXL macro.
 - The LIBNAME engine with pre-formatted template.
- The above ways are also optimal for *custom formatting* output.

Further Resources

Too many to list – see the paper!

Nate Derby: `nderby@stakana.com`

Colleen McGahan: `cmcgahan@bccancer.bc.ca`