

Using the SAS® Output Delivery System (ODS) and the TEMPLATE Procedure to Replace Dynamic Data Exchange (DDE)

Peter Timusk, Centre for Special Business Projects, Statistics Canada

ABSTRACT

Many papers have been written over the years that describe how to use Dynamic Data Exchange (DDE) to pass data from SAS® to Excel. This presentation aims to show you how to do the same exchange with the SAS Output Delivery System (ODS) and the TEMPLATE Procedure.

INTRODUCTION

This paper demonstrates techniques using the TEMPLATE procedure and the STYLE element to format Excel workbooks output with SAS Output Delivery System (ODS). There have been many papers over the years explaining how to format Excel workbooks using the Dynamic Data Exchange (DDE) method of communicating with Excel from SAS. The author has often found these papers online and used these DDE techniques to deliver publication standard Excel workbooks containing statistics from Statistics Canada surveys.

Clients prefer Excel workbooks and in using Excel templates these clients have an ability to define statistical estimates and population and sub-population requirements. The production programmer has only to calculate the statistics in SAS and then place the results in the client's Excel template for publication as an Excel workbook.

More SAS user papers these days show how to use ODS and PROC TEMPLATE to achieve the same formatting results with Excel workbooks. This paper contributes to these papers and will demonstrate some formatting techniques important in the author's programming statistics for publication in Excel workbooks. It is hoped that future SAS programmers will find many published example papers for using ODS, as we can now find for DDE.

USING THE TEMPLATE PROCEDURE

Throughout we will show a simple PROC TEMPLATE and a DEFINE STYLE to make a new STYLE element using various STYLE attributes to format Excel. This is the technique and code used by, Nate Derby and Colleen McGahan in their 2013 SAS Global Forum paper: *Maintaining Formats when Exporting Data from SAS® into Microsoft® Excel®*, also with the Class dataset.

We will first apply a numeric format as these authors did, then extend this to a font format, then a cell format and finally a font justification format. Then we will use all of these together to end the examples. For many more possible formats see the references section.

FORMATING NUMBERS

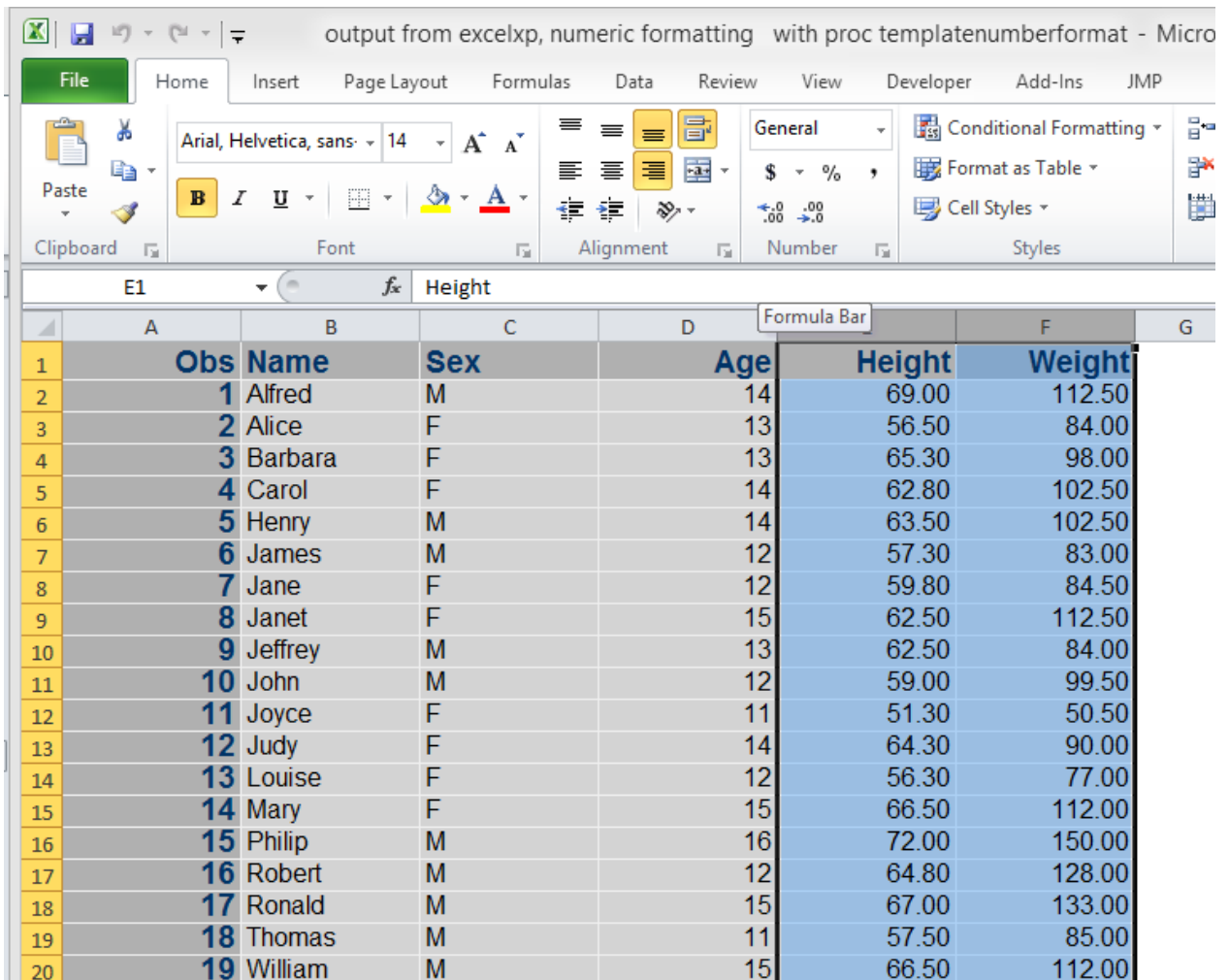
Here we use PROC TEMPLATE and a STYLE element to format two variables height and weight from the Class dataset with the STYLE element "data_num" to display two decimal places in the Excel output:

```
proc template;
  define style styles.mystyle;
    parent = styles.default;
    style data_num from data / tagattr='format:0.00' ;
  end;
quit;
ods tagsets.excelxp file="&folder output from excelxp, numeric formatting
with proc templatenumberformat.xls" style=mystyle ;
```

```
proc print data=sashelp.class;
  var name sex age;
  var height weight / style( data )=data_num;
run;
ods tagsets.excelxp close;
```

(NOTE: the code in this paper is drawn from examples in (Derby & McGahan, 2013), (DelGobbo, 2012) and SAS documentation and then extended to other formats.)

The format used is set using a tagattr and this string 'format:0.00'



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

Display 1 Excel with Numeric Formatting

FORMATING FONTS

Here we use the PROC TEMPLATE and a STYLE element to format the variable age from the Class dataset with the STYLE element "super_up" to appear as superscripted data points in the Excel output:

```

proc template;
  define style styles.mystyle;
    parent = styles.default;
    style super_up from data / fontsize= 8pt verticalalign=top;
  end;
quit;
ods tagsets.excelxp file="&folder output from excelxp, font formatting
with proc templatesuperup.xls" style=mystyle ;
proc print data=sashelp.class;
  var name sex;
  var age / style( data )=super_up;
run;
ods tagsets.excelxp close;

```

Superscripting using the attribute “super” seems possible in a STYLE element only for text not data, so we use a combination of the attributes fontsize to make a smaller font size and an align attribute verticalalign to align the font at the top of the cell to in effect superscript data points. In our work we usually included a letter grading in a column next to an estimate indicating its coefficient of variation and amount of imputed data contained in the estimate, as an indicator of an estimates’ quality. In the Excel this is displayed as a superscripted letter grade in a column next to the estimate. Each quality letter is calculated individually for each estimate it corresponds to, so is in effect data itself.

output from excelxp, font formatting with proc templatesuperup - Micro

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

Display 2 Excel with Superscripted Data Formatting

MORE STYLE ELEMENTS

FORMATING CELLS

Here we format a cell with borders. Again we use the PROC TEMPLATE and a STYLE element to format the cells in an Excel workbook with a border:

```
proc template;
  define style styles.mystyletable;
    parent = styles.default;
    style borders from data / borderleftwidth=1pt
      borderrightwidth=1pt borderbottomwidth=1pt;
  end;
quit;
ods tagsets.excelxp file="&folder output from excelxp, numeric formatting
with proc templateborderscell.xls" style=mystyletable;
proc print data=sashelp.class;
  var name sex age height weight/ style( data )=borders ;
run;
ods tagsets.excelxp close;
```

	A	B	C	D	E	F
		Obs	Name	Sex	Age	Height
1		1	Alfred	M	14	69
2		2	Alice	F	13	56.5
3		3	Barbara	F	13	65.3
4		4	Carol	F	14	62.8
5		5	Henry	M	14	63.5
6		6	James	M	12	57.3
7		7	Jane	F	12	59.8
8		8	Janet	F	15	62.5
9		9	Jeffrey	M	13	62.5
10		10	John	M	12	59
11		11	Joyce	F	11	51.3
12		12	Judy	F	14	64.3
13		13	Louise	F	12	56.3
14		14	Mary	F	15	66.5
15		15	Philip	M	16	72
16		16	Robert	M	12	64.8
17		17	Ronald	M	15	67
18		18	Thomas	M	11	57.5
19		19	William	M	15	66.5

Display 3 Excel with Cell Border Formatting

Here we set the cell borders to equal 1pt with various border width attributes.

FORMATING JUSTIFICATION OF FONTS

Here we use the PROC TEMPLATE and a STYLE element to format the justification of data values. Here we justify left the age variable in the Class dataset. We can justify left or right or center with justify attributes.

```
proc template;
  define style styles.mystyletable;
    parent = styles.default;
    style just_left from data / just=left;
  end;
quit;
ods tagsets.excelxp file="&folder output from excelxp, font formatting
  with proc templatejustifyleft.xls" style=mystyletable ;
proc print data=sashelp.class;
  var name sex;
  var age / style( data )=just_left ;
  var height weight ;
```

```
run;
ods tagsets.excelxp close;
```

output from excelxp, font formatting with proc templatejustifyleft - Micro

	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
18	18	Thomas	M	11	57.5	85
19	19	William	M	15	66.5	112

Display 4 Excel with Left Justify Formatting

Here we have justified the variable age to the left.

FORMATING MULTIPLE ATTRIBUTES AT THE SAME TIME

Here we use the PROC TEMPLATE and a STYLE element to use all the previous formats and apply these all to the last column the weight variable column.

```
proc template;
  define style styles.mystyletable;
    parent = styles.default;
    style all_formats from data / just=left tagattr='format:0.00'
    fontsize=6pt verticalalign=top borderleftwidth=1pt
    borderrightwidth=1pt borderbottomwidth=1pt;
  end;
quit;
ods tagsets.excelxp file="&folder output from excelxp, all formatting
with proc templateallformats.xls" style=mystyletable ;
```

```

proc print data=sashelp.class;
  var name sex age height;
  var weight/ style( data )=all_formats ;
run;
ods tagsets.excelxp close;

```

	A	B	C	D	E	F
	Obs	Name	Sex	Age	Height	Weight
1	1	Alfred	M	14	69	112.50
2	2	Alice	F	13	56.5	84.00
3	3	Barbara	F	13	65.3	98.00
4	4	Carol	F	14	62.8	102.50
5	5	Henry	M	14	63.5	102.50
6	6	James	M	12	57.3	83.00
7	7	Jane	F	12	59.8	84.50
8	8	Janet	F	15	62.5	112.50
9	9	Jeffrey	M	13	62.5	84.00
10	10	John	M	12	59	99.50
11	11	Joyce	F	11	51.3	50.50
12	12	Judy	F	14	64.3	90.00
13	13	Louise	F	12	56.3	77.00
14	14	Mary	F	15	66.5	112.00
15	15	Philip	M	16	72	150.00
16	16	Robert	M	12	64.8	128.00
17	17	Ronald	M	15	67	133.00
18	18	Thomas	M	11	57.5	85.00
19	19	William	M	15	66.5	112.00

Display 5 Excel with All Example Formats Together

CONCLUSION

The examples show how easily Excel output can be formatted with a TEMPLATE procedure and a STYLE element. In past user papers, this type of formatting was completed using Dynamic Data Exchange. This paper demonstrated a few examples of formatting Excel workbook produced by ODS Excel XP and a TEMPLATE procedure and a STYLE element. There are many other ways to achieve this formatting depending on the procedure used inside the ODS statements, such as using STYLE elements in a REPORT procedure, (Fraeman, 2009. p. 8) or with GRAPH procedures. As well one can use OPTIONS on the ODS statement or adjust tagsets with the TEMPLATE procedure.

This paper was mostly written with SAS 9.2 and with SAS 9.4 a new ODS output EXCEL is now available in a beta version. This paper has shown some formatting using only the STYLE element inside a TEMPLATE procedure for the tagset ExcelXP. Please see the references for full SAS documentation of the STYLE elements available and representative papers covering ODS and Excel. You may also want to

check other sessions at this year's SAS Global forum and demos that will undoubtedly cover ODS and Excel with more depth and using other techniques.

REFERENCES

- DelGobbo , Vincent.2012. An Introduction to Creating Multi-Sheet Microsoft Excel Workbooks the Easy Way with SAS®. *Proceedings of the SAS Global 2012 Conference*. Orlando, FL. Available at <http://support.sas.com/resources/papers/proceedings12/150-2012.pdf>
- Derby , Nate. McGahan, Colleen. 2013. Maintaining Formats when Exporting Data from SAS® into Microsoft® Excel®, *Proceedings of the SAS Global 2013 Conference*. San Francisco, CA. Available at <http://support.sas.com/resources/papers/proceedings13/316-2013.pdf>
- Fraeman , Kathy Hardis. 2009. Formatting Tips and Techniques for Printable Excel Tables Created with the SAS® Excel XP ODS Tagset and PROC REPORT. *North East SAS User Group (NESUG)* Burlington, VT. Available at <http://www.lexjansen.com/nexus/nexus09/np/NP03.pdf>
- Gebhart, Eric. 2010. ODS ExcelXP: Tag Attr Is It! Using and Understanding the TAGATTR= Style Attribute with the ExcelXP Tagset. *Proceedings of the SAS Global 2010 Conference*. Seattle, WA. Available at <http://support.sas.com/resources/papers/proceedings10/031-2010.pdf>
- Haworth , Lauren . 2006. PROC TEMPLATE: The Basics *Proceedings of the SUGI31 Conference*. San Francisco, CA. Available at <http://www2.sas.com/proceedings/sugi31/112-31.pdf>
- Helbers, Lawrence. Vinokurov, Alex. 2002. Output to Excel: DDE and Beyond. *North East SAS User Group (NESUG)* Buffalo, NY. Available at <http://www.lexjansen.com/nexus/nexus02/ad/ad007.pdf>
- Hemedinger, Chris. Experimenting with ODS EXCEL to create spreadsheets from SAS. *The SAS Dummy*. August 29, 2014. Available at <http://blogs.sas.com/content/sasdummy/2014/08/29/experimenting-with-ods-excel-to-create-spreadsheets-from-sas/>
- SAS Institute. 2010. SAS 9 ODS Excel XP Tip Sheet. Available at http://support.sas.com/rnd/base/ods/odsmarkup/TipSheet_ods_xl_xp.pdf

ACKNOWLEDGMENTS

I was greatly aided in writing this paper by questions and discussions on the SAS-L mailing list.

SAS Institute staff answered queries online this proved helpful and SAS Institute staff papers were also helpful when my coding reached problems.

I need to acknowledge my management at my workplace for supporting my SAS use and allowing me to learn more about SAS as part of my work.

RECOMMENDED READING

- *ODS The Basics and Beyond*
- *PROC TEMPLATE made easy : a guide for SAS users*
- *SAS® (9.2, 9.3, 9.4) Output Delivery Procedures Guide: Template Procedure: Creating a Style Template: Style Statement.*

CONTACT INFORMATION

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

Peter Timusk
Centre for Special Business Projects, Statistics Canada
peter.timusk@statcan.gc.ca
www.statcan.gc.ca

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.