

**Paper 973**  
**ODS TAGSETS.EXCELXP and ODS EXCEL SHOWDOWN**  
Christopher J. Boniface, U.S. Census Bureau;

## **ABSTRACT**

Do you create Excel files from SAS? Do you use ODS TAGSETS.EXCELXP or ODS EXCEL? The ExcelXP tagset and ODS Excel destinations are compared face to face. There's gonna be a showdown! We provide quick tips for each and show how to create Excel files for our Special Census program. The pros of each destination are explored. We'll show the added benefits of ODS Excel. We'll display how to create XML files with the ExcelXP tagset. We'll present how to use TAGATTR formats with the ExcelXP tagset to ensure that leading and trailing zeroes in Excel are preserved. Secondly, we'll demonstrate how to create the same Excel file with ODS Excel with SAS formats instead of TAGATTR formats. Additionally, we'll see that ODS Excel creates native Excel files. One of the drawbacks of an XML file created with the ExcelXP tagset is that a popup message is displayed in Microsoft Excel each time you open it. Lastly, we'll present differences with the `absolute_column_width=` option between the two destinations.

## **INTRODUCTION**

A Special Census is a basic enumeration of population, housing units, group quarters and transitory locations, conducted by the U.S. Census Bureau at the request of a governmental unit. The Census Bureau conducts Special Censuses between decennial censuses, starting in years ending in "2" and ending in years ending in "8". When local officials believe there has been a significant population change in their community due to growth or annexation, a Special Census may be in order. The Census Bureau produces Excel files that show population and housing counts by block and one-page demographic profiles for the governmental unit and associated tracts or part tracts. Standard information includes age, sex, race, Hispanic origin, occupancy or vacancy status, type of vacancy, and tenure for housing units. Most governmental units prefer to receive these data in Excel or PDF formats.

The Census Bureau tabulates and produces these reports in SAS without any post-processing or manual touch-ups. The reports are Excel files produced using PROC REPORT with ODS using SAS 9.4 Maintenance Release 3. The focus of this paper is on the differences between the ExcelXP tagset and ODS EXCEL for the following topics:

- 1) XML file with the ExcelXP Tagset versus native Excel file (.xlsx) with ODS EXCEL
- 2) ExcelXP Tagset Formatting with or without TAGATTR formats
- 3) ODS Excel Formatting with SAS Formats
- 4) `absolute_column_width` differences between the ExcelXP tagset and ODS Excel

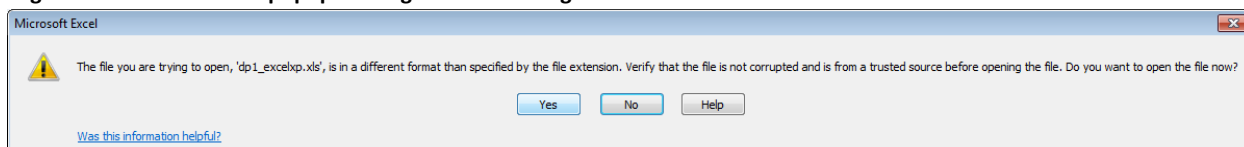
## **XML file versus Native Excel File (.xlsx)**

The main difference between the ExcelXP tagset and ODS Excel is that they create two different output files. The ExcelXP tagset creates a Microsoft XML spreadsheet file, which can be used with Excel 2002 and later. The most important thing to realize is that the ExcelXP tagset does not create a native Excel format (.xlsx). Starting in SAS 9.4 maintenance release 3, the ODS Excel destination creates native Excel (.xlsx) files that can be used with Microsoft Office 2010 or later. Another difference between the two destinations is on the ODS statement where you specify the output file to create. One of the options on the ODS ExcelXP tagset statement is the `path=` option, which specifies the output directory where the XML file will be created. The `path=` option is not available with the ODS Excel destination. According to Chevell Parker of SAS Institute Inc. from his paper, "A Ringside Seat: The ODS Excel Destination versus the ODS ExcelXP Tagset", the option is not used in the Excel destination because all files are local to the workbook when you use the destination. The code below compares the syntax between the ODS tagsets.ExcelXP and ODS Excel statements. The ExcelXP tagset destination uses the `path` and `file` options. The ODS Excel destination uses just the `file=` option. The `file=` option is where you specify the name of the final output file. With ODS Excel, the output file is created in the local directory where the program is run.

```
ods tagsets.excelxp path="/special_census/output" file="dp1_excelxp.xls"
ods excel file="dp1_excel.xlsx"
```

Notice, that with the above ODS tagsets.ExcelXP statement, the extension of the filename on the file= option is .xls. The file will be created with the .xls extension, but the file created is not a native Excel file. It is an XML file in reality. Either way (.xml or .xls), when you try to open the file in Microsoft Excel you get a popup message. Figure 1 shows the popup message that you get every time you try to open the native XML file in Microsoft Excel. The file will open up fine once you click on “Yes”, but do you really want to see this popup message for every XML file created with the ExcelXP tagset? You may have hundreds of XML files created and you don’t want to click “Yes” several hundred times to open each file. Thus, when creating hundreds of pseudo Excel (really XML) files with the ExcelXP tagset, you probably will have to run some post processing to avoid these popups. A Visual Basic script needs to be run after the fact to convert the XML file(s) to native Excel files (.xlsx) in order to get rid of the popup messages.

**Figure 1. Microsoft Excel popup message when creating an XML file with ODS EXCELXP**



The ODS Excel destination is the solution. It creates a native Excel file (.xlsx) for use in Microsoft Excel 2010 or later. When you open a native Excel file (.xlsx) it opens up immediately with no time consuming popup messages.

## ODS EXCELXP - Formatting With or Without TAGATTR

One of the initial issues discovered when creating Excel files for our Special Census deliverables was that leading and trailing zeroes were omitted from the report. Excel does not honor SAS formats with the ExcelXP tagset. For example, sometimes a percent might be 5.0 in our Demographic Profile or a county code might have a leading zero in our Block Report. The program below shows our original ExcelXP tagset code using PROC REPORT with SAS formats for our Demographic Profile. Table 1 shows the output without commas and with no trailing zeroes.

```
ods tagsets.excelxp path="&outp2" file="dp1_&caseid._excelxp.xls"
  style=printer options(header_data_associations="yes" autofit_height='yes')
options(embedded_titles="yes" embedded_footnotes='yes' absolute_column_width="37,9,9");

proc report data=section1 nowd spanrows split='|'
  style(header)=[fontweight=bold fontsize=10pt font=(Times, 8pt) ] ;
  column name ("Special Census" number pc);
  define name / "Subject" f=$table1f. font=(TimesNewRoman, 10pt)];
  define number / 'Number' f=comma7.0 ;
  define pc / 'Percent' f=4.1 ;
```

Subject	Special Census	
	Number	Percent
<b>SEX AND AGE</b>		
<b>Total population</b>	<b>20420</b>	<b>100</b>
Under 5 years	2007	9.8
5 to 9 years	2121	10.4
10 to 14 years	1838	9
15 to 19 years	1555	7.6
20 to 24 years	1123	5

**Table 1. ODS tagset.ExcelXP Demographic Profile output using SAS Formats**

In this example, the percent column is numeric in our dataset. As you can see, Microsoft Excel has stripped out the trailing zeroes in the Percent column for ages 10-14 and 20 to 24. This led to us post-processing the file to preserve the commas and trailing zeroes. With the ExcelXP tagset, SAS formats are ignored in favor of Excel formats. However, TAGATTR formats can be used with the ExcelXP tagset in your SAS code to give you control of the formatting. You can override the automatic mapping of SAS formats to the Excel formats by using the TAGATTR= attribute to pass a custom Excel format. The following SAS code takes advantage of two such TAGATTR formats in our ExcelXP tagset code. The TAGATTR formats are specified as style overrides on the define statements for the "number" and "pc" columns. Table 2 shows the output with commas and trailing zeroes preserved.

```
ods tagsets.excelxp path="&outp2" file="dp1_&caseid._excelxp.xls"
  style=printer options(header_data_associations="yes" autofit_height='yes')
options(embedded_titles="yes" embedded_footnotes='yes' absolute_column_width="37,9,9");

proc report data=section1 nowd spanrows split='|'
  style(header)=[fontweight=bold fontsize=10pt font=(Times, 8pt) ] ;
  column name ("Special Census" number pc);
  define name / "Subject" f=$table1f. font=(TimesNewRoman, 10pt)];
  define number / 'Number' style(column)=[tagattr="format:##0" font=(TimesNewRoman, 10pt)] ;
  define pc / 'Percent' style(column)=[tagattr="format:##0.0" font=(TimesNewRoman, 10pt)] ;
```

Subject	Special Census	
	Number	Percent
<b>SEX AND AGE</b>		
<b>Total population</b>	<b>20,420</b>	<b>100</b>
Under 5 years	2,007	9.8
5 to 9 years	2,121	10.4
10 to 14 years	1,838	9.0
15 to 19 years	1,555	7.6
20 to 24 years	1,021	5.0

**Table 2. ODS tagset.ExcelXP Demographic Profile - TAGATTR Formats preserve comma's & trailing zeroes**

Similarly, the Block Excel file uses TAGATTR formats to preserve leading zeroes. With the Demographic Profile, the TAGATTR formats preserved the trailing zeroes of the numeric Percent column. For the Block file, there are character variables for the geography columns, which sometimes have leading zeroes. Without the TAGATTR formats, the leading zeroes are stripped out of the Excel file with the ExcelXP tagset. However, when using TAGATTR formats with these character variables, the leading zeroes are preserved. Below is the SAS code using the ExcelXP tagset with TAGATTR formats for the geographic character variables. Table 3A below shows the output Block file with the leading zeroes preserved for the FIPS County Code column and the Census Tract Code column. Table 3B below shows the output using the same program (code not shown), but without TAGATTR formats as style overrides. Without the TAGATTR formats, the leading zeroes are stripped off these same two columns.

```
ods tagsets.excelxp path="&outp2" file="blks-&caseid._excelxp.xls"
  style=printer options(header_data_associations="yes" autofit_height='yes')
options(orientation='landscape' absolute_column_width="6,6,7,6,6,6,6,6");

/* Use TAGATTR formats with ODS EXCELXP to preserve leading zeroes for character variables */
proc report data=outp.tabs_profile nowd spanrows split='|'
  style(header)=[fontsize=0.75 fontweight=bold ]
  style(column)=[font=(Arial, 8pt)];
  column tabblkst tabblkcou cousubfp placefp tabtractce tabblock count_Sum count17under_Sum;
  define tabblkst / "FIPS|State|Code" style(column)=[tagattr="format:0##0"];
  define tabblkcou / "FIPS|County|Code" style(column)=[tagattr="format:0#0"];
  define cousubfp / "FIPS|County|Subdivision|Code" style(column)=[tagattr="format:0###0"];
```

```
define placefp / "FIPS|Place|Code" order=data flow style(column)=[tagattr="format:0###0"];
define tabtractce / "Census|Tract|Code" style(column)=[tagattr="format:0####0"];
```

FIPS State Code	FIPS County Code	FIPS County Subdivision Code	FIPS Place Code	Census Tract Code	Census Block Code	Total population	Age 17 years and under
019	049	90300	83910	050809	1001C	0	0
019	049	90300	83910	050809	1043A	0	0

Table 3A. ODS EXCELXP Block File - TAGGATR Formats preserve leading zeroes

FIPS State Code	FIPS County Code	FIPS County Subdivision Code	FIPS Place Code	Census Tract Code	Census Block Code	Total population	Age 17 years and under
019	49	90300	83910	50809	1001C	0	0
019	49	90300	83910	50809	1043A	0	0

Table 3B. ODS EXCELXP Block File – No TAGGATR Formats. Leading Zeroes Stripped

For a list of Excel custom-format values that you can add by using the TAGTATTR= attribute, see the Microsoft article "Create or delete a custom number format." ([support.office.com/en-us/article/Create-ordelete-a-custom-number-format-78f2a361-936b-4c03-8772-09fab54be7f4](https://support.office.com/en-us/article/Create-ordelete-a-custom-number-format-78f2a361-936b-4c03-8772-09fab54be7f4)).

## ODS EXCEL - Formatting With SAS Formats

With the arrival of the ODS Excel destination, you do not necessarily need to use TAGATTR formats to preserve leading or trailing zeroes. With ODS EXCEL, SAS formats can be used instead of the TAGATTR formats to preserve leading or trailing zeroes. This is exciting news, since SAS programmers are more familiar with SAS formats than TAGATTR formats. It is still possible to use TAGATTR formats with ODS Excel. According to Chevell Parker of SAS Institute Inc. from his paper, "A Ringside Seat: The ODS Excel Destination versus the ODS ExcelXP Tagset", SAS formats with the Excel destination when you export output to Excel. SAS formats are automatically mapped to comparable Excel formats that are assigned to the column. The program below uses ODS Excel with SAS formats to produce the Demographic Profile. Table 4 below shows the output. Notice that the SAS format, comma7., is used for the Number column and that the commas are preserved. Moreover, the SAS 4.1 format is used for the Percent column and the trailing zeroes are preserved for ages 10 to 14 and 20 to 24.

```
ods excel file="dp1_&caseid._excel.xlsx" style=printer
options(embedded_titles="yes" embedded_footnotes='yes' absolute_column_width="67,16,16");

/* generate excel output with ODS Excel. Use SAS formats instead of TAGATTR formats*/
proc report data=section1 nowd spanrows split='|'
    style(header)=[fontweight=bold fontsize=10pt font=(Times, 8pt) ] ;
column name ("Special Census" number pc);
define name / "Subject" f=$table1f. order=data flow;
define number / 'Number' f=comma7. ;
define pc / 'Percent' f=4.1;
```

Subject	Special Census	
	Number	Percent
<b>SEX AND AGE</b>		
<b>Total population</b>	<b>20,420</b>	<b>100</b>
Under 5 years	2,007	9.8
5 to 9 years	2,121	10.4
10 to 14 years	1,838	9.0
15 to 19 years	1,555	7.6
20 to 24 years	1,021	5.0

**Table 4. ODS EXCEL output using SAS formats instead of TAGATTR formats**

Similarly, with our Block file, the TAGATTR formats are not needed when using ODS Excel. Instead, SAS formats can be used with ODS Excel. In the program below, the character formats, \$3. and \$6., are used for the FIPS County Code and Census Tract Code, respectively. Table 5 below shows that the leading zeroes for these two columns are preserved using ODS Excel with SAS formats. However, notice that the lengths of the columns are narrower and the column headers are truncated compared to the output from ODS EXCELXP. The column widths are narrower with ODS Excel despite using the same values for the absolute\_column\_width that are used with the ExcelXP tagset. Also, notice that some of the column headers are chopped off.

```

/* Use SAS formats with ODS EXCEL. Leading zeroes are preserved for character variables */
ods excel file="blks-&caseid._excel.xlsx" style=printer
options(orientation='landscape' absolute_column_width="6,6,7,6,6,6,6,6");
proc report data=outp.tabs_profile nowd spanrows split='|'
style(header)=[fontsize=0.75 fontweight=bold ] style(column)=[font=(Arial, 8pt)];
column tabblkst tabblkcou cousubfp placefp tabtractce tabblock count_Sum count17under_Sum;
define tabblkst / "FIPS|State|Code" f=$2. order=data flow ;
define tabblkcou / "FIPS|County|Code" f=$3. order=data flow ;
define cousubfp / "FIPS|County|Subdivision|Code" f=$5. order=data flow ;
define placefp / "FIPS|Place|Code" order=data f=$5. flow ;
define tabtractce / "Census|Tract|Code" f=$6. order=data flow ;

```

FIPS State Code	FIPS County Code	FIPS Subdivision Code	FIPS Place Code	Census Tract Code	Census Block Code	Age 17 and under population
19	049	90300	83910	050809	1001C	0
19	049	90300	83910	050809	1043A	0

**Table 5. ODS EXCEL output using SAS formats, column widths smaller than same ODS EXCELXP widths**

### Absolute\_Column\_Width=

The absolute\_column\_width= option with both the ExcelXP tagsets and with ODS Excel will adjust the column widths of our Excel tables. According to the SAS® 9.4 Output Delivery System: User's Guide, Fourth Edition, you need to list widths to use for columns instead of allowing SAS to determine the column width (measured widths). The number-list is a comma separated list of numbers. This option was used initially with the ExcelXP tagset to establish the column widths for the tables. However, when using these same column widths in ODS Excel, the columns and the overall table are less wide than our output from the ExcelXP tagset. If you were to take a ruler and compare the table width of the output in Table 5 (ODS Excel) to that of Table 3A (ExcelXP tagset), you can quickly see that the width in table 5 is much less than the width of Table 3A.

According to SAS Institute Technical Support, the measurement for absolute\_column\_width is not quite right in SAS 9.4M3. The ABSOLUTE\_COLUMN\_WIDTH option works properly only for a very small set of fonts in the Normal style. A font size larger than expected (>10 or 12 points) causes the column widths to be miscalculated.

It has been updated for SAS 9.4M4 where it is corrected along with the ability to specify the width using a unit of dimension such as PX, MM, CM, or IN. However, for now with ODS Excel in SAS 9.4M3, a work-around is needed to adjust the column widths, so that the tables have the same overall width. Increasing the width of each column in the absolute\_column\_width option with ODS Excel, gives similar columns widths to what we had with the ExcelXP tagset. Additionally, the column headers are not truncated as you can see in Table 6 below. The below code has the adjusted values for the absolute\_column\_width= option when using ODS Excel. As you can see, the overall width of the table is now similar to the overall width of the same table when using the ExcelXP tagset (Table 3A).

```
ods excel file="blks-&caseid._excel.xlsx" style=printer
  options(orientation='landscape' absolute_column_width="11,11,12,11,11,11,11,11");
```

FIPS State Code	FIPS County Code	FIPS County Subdivision Code	FIPS Place Code	Census Tract Code	Census Block Code	Total population	Age 17 years and under
19	049	90300	83910	050809	1001C	0	0
19	049	90300	83910	050809	1043A	0	0

Table 6. ODS EXCEL output using SAS formats, with larger adjusted absolute column widths than we used with ODS EXCELXP

## CONCLUSION

You can use ODS tagsets.ExcelXP or ODS Excel to create great looking Excel files. The ExcelXP tagset will create an XML file that can be opened in Microsoft Excel. However, a popup message will be displayed in Microsoft Excel each time you open this XML file to indicate that the file is not a native Excel file. The ODS Excel destination will create a native Excel file (.xlsx) without the popup message when you open it. The TAGATTR formats can be used with the ExcelXP tagset to control the formatting in Microsoft Excel, specifically to preserve leading and trailing zeroes. Additionally, SAS formats can be used instead of TAGATTR formats with the ODS Excel destination. Moreover, leading and trailing zeroes can be preserved when using SAS formats with the ODS Excel destination. Lastly, there are some differences with the column widths of the Excel tables between the ExcelXP tagset and ODS Excel when using the absolute\_column\_width= option. Lastly, when using the same values for the absolute\_column\_width= option, overall table width is less for native Excel files created with ODS Excel than an XML file created with the ExcelXP tagset in SAS 9.4M3.

## REFERENCES

SAS® 9.4 Output Delivery System: User's Guide, Fourth Edition

Chevell Parker, SAS Institute Inc. 2015. " A Ringside Seat: The ODS Excel Destination versus the ODS ExcelXP Tagset

Microsoft Corporation. 2016. "Create or delete a custom number format." Microsoft Office Online. Available at [support.office.com/en-us/article/Create-or-delete-a-custom-numberformat-78f2a361-936b-4c03-8772-09fab54be7f4](https://support.office.com/en-us/article/Create-or-delete-a-custom-numberformat-78f2a361-936b-4c03-8772-09fab54be7f4).

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

Name: Christopher J. Boniface  
 U.S. Census Bureau  
 Washington D.C. 20233  
 Work Phone: (301)763-5769  
 E-mail: [christopher.j.boniface@census.gov](mailto:christopher.j.boniface@census.gov)

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.