



Try This Demo: The ExcelXP Tagset and Microsoft Excel

- [Introduction](#)
- [Skip Spacing](#)
- [Suppress Bylines](#)
- [Table of Contents and Index of Tables](#)
- [Data-Driven Worksheets](#)
- [Manual Multiple Worksheets](#)
- [Autofilters](#)
- [Data-Specified Formulas](#)
- [Formats and Formulas](#)
- [Embedded Titles and Footnotes](#)
- [Default Column Width and Width Fudge](#)
- [Rotated Column Headers](#)
- [Flyovers](#)

Introduction

This topic provides example code that uses the ExcelXP tagset to generate XML output. You can run the code in SAS 9.1.3 or later release, and then open the XML files in Microsoft Excel 2000 or later. The output also works with the Calc spreadsheet program from OpenOffice.org.

To run these examples in SAS, first download the latest ExcelXP tagset from the [ODS MARKUP](#) page. That page also links to documentation for using and customizing tagsets.

See additional ExcelXP samples in these papers and presentations:

- [ODS MARKUP: The SAS Reports You've Always Dreamed Of \(.pdf\)](#) and the paper's [example code \(.tgz\)](#)
- [Example code for ODS Tagsets & Excel, an Excellent Combination \(.zip\)](#)
- [The Beginner's Guide to ODS MARKUP: Don't Panic! \(.pdf\)](#) and the paper's [example code \(.tgz\)](#).

Skip Spacing

You can adjust the spacing between objects in a report by using the SKIP_SPACE= option. The value of this option should be a comma-separated list of numbers that correspond to the spacing after tables, bylines, titles, footers, and pagebreaks.

The default values are:

- Table : 1
- Byline : 0
- Title : 1
- Footer : 1
- PageBreak : 1

```
ods tagsets.excelxp file="spacing.xls" style=statistical
  options( skip_space='3,2,0,0,1' sheet_interval='none'
           suppress_bylines='no');
```

```
proc sort data=sashelp.class out=class;
  by age;
run;
```

```
proc print data=class;
```

```

        by age;
run;

ods tagsets.excelxp close;

```

Suppress Bylines

You can use the SUPPRESS_BYLINES= option to turn the bylines on and off.

```

ods tagsets.excelxp file="bylines.xls" style=statistical
  options( suppress_bylines='yes' sheet_interval='none' );

proc sort data=sashelp.class out=class;
  by age;
run;

proc print data=class;
  by age;
run;

ods tagsets.excelxp options( suppress_bylines='no' sheet_interval='none' );

proc print data=class;
  by age;
run;

ods tagsets.excelxp close;

```

Table of Contents and Index of Tables

You can generate a table of contents and an index of tables by using the CONTENTS= and INDEX= options, respectively. The table of contents and index are hyperlinked to the corresponding worksheets in the report.

```

ods tagsets.excelxp file="toc.xls" style=statistical
  options( contents='yes' index='yes' );

proc print data=sashelp.class; run;

proc contents data=sashelp.class; run;

proc means data=sashelp.class; run;

ods tagsets.excelxp close;

```

Data-Driven Worksheets

By default, a new worksheet is generated for each table. You can override this and create a new worksheet for each proc, bygroup, or never. The example here creates a new worksheet for each bygroup.

```

ods tagsets.excelxp file='multisheet.xls' style=statistical
  options( sheet_interval='bygroup' );

proc sort data=sashelp.class out=class;
  by age;
run;

proc print data=class;
  by age;
run;

```

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

Manual Multiple Worksheets

Use the SHEET_NAME= and SHEET_INTERVAL= options to specify the name of a worksheet as well as the interval in create new worksheets.

```
data prdsale;
  set sashelp.prdsale;
  difference = actual-predict;
run;

proc sort data=prdsale; by country region division year; run; quit;

title; footnote;

*;
* Create a workbook with multiple tables per worksheet, and specify
* the name for the worksheets. Autofilters will be applied to the
* second table in each worksheet. The SUM statement of PROC PRINT
* will result in a subtotal row, but note that the SUBTOTAL function
* is not used for the DIFFERENCE column because a formula was specified on
* the SUM statment. In this case, the user-specified formula takes
* precedence over the auto subtotal.
*;

ods listing close;

ods tagsets.excelxp file='multitable.xml' style=statistical
  options(auto_subtotals='yes' default_column_width='7, 10, 10, 7, 7'
    frozen_rowheaders='yes' sheet_interval='none' sheet_name='Canada'
    autofilter='all' autofilter_table='2');

*;
* The output from the following two procs will be in a single worksheet
* with a user-specified name of 'Canada'.
*;

proc tabulate data=prdsale;
  where country eq 'CANADA' and year eq 1993;
  var predict actual;
  class region division prodtype;
  table
    region*(division*prodtype all={label='Division Total'}) all={label='Grand Total'
    predict={label='Total Predicted Sales'}*f=dollar10.*sum={label=''}
    actual={label='Total Actual Sales'}*f=dollar10.*sum={label=''}};
run; quit;

proc print data=prdsale noobs label split='*';
  where country eq 'CANADA' and year eq 1993;
  id country region division;
  var prodtype product quarter month year;
  sum predict / style={tagattr='format:Currency'};
  sum actual / style={tagattr='format:Currency'};
  sum difference / style={tagattr='format:Currency formula:RC[-1]-RC[-2]'};
  label prodtype = 'Product*Type'
        predict = 'Predicted*Sales'
        actual = 'Actual*Sales';
run; quit;

ods tagsets.excelxp options(sheet_interval='none' sheet_name='Germany');

*;
* The output from the following two procs will be in a single worksheet
* with a user-specified name of 'Germany'.
```

```

*;

proc tabulate data=prdsale;
  where country eq 'GERMANY' and year eq 1993;
  var predict actual;
  class region division prodtype;
  table
    region*(division*prodtype all={label='Division Total'}) all={label='Grand Total'
    predict={label='Total Predicted Sales'}*f=dollar10.*sum={label=''}
    actual={label='Total Actual Sales'}*f=dollar10.*sum={label=''};
run; quit;

proc print data=prdsale noobs label split='*';
  where country eq 'GERMANY' and year eq 1993;
  id country region division;
  var prodtype product quarter month year;
  sum predict / style={tagattr='format:Currency'};
  sum actual / style={tagattr='format:Currency'};
  sum difference / style={tagattr='format:Currency formula:RC[-1]-RC[-2]'};
  label prodtype = 'Product*Type'
        predict  = 'Predicted*Sales'
        actual   = 'Actual*Sales';
run; quit;

ods tagsets.excelxp options(sheet_interval='none' sheet_name='United States');

*;
* The output from the following two procs will be in a single worksheet
* with a user-specified name of 'United States'.
*;

proc tabulate data=prdsale;
  where country eq 'U.S.A.' and year eq 1993;
  var predict actual;
  class region division prodtype;
  table
    region*(division*prodtype all={label='Division Total'}) all={label='Grand Total'
    predict={label='Total Predicted Sales'}*f=dollar10.*sum={label=''}
    actual={label='Total Actual Sales'}*f=dollar10.*sum={label=''};
run; quit;

proc print data=prdsale noobs label split='*';
  where country eq 'U.S.A.' and year eq 1993;
  id country region division;
  var prodtype product quarter month year;
  sum predict / style={tagattr='format:Currency'};
  sum actual / style={tagattr='format:Currency'};
  sum difference / style={tagattr='format:Currency formula:RC[-1]-RC[-2]'};
  label prodtype = 'Product*Type'
        predict  = 'Predicted*Sales'
        actual   = 'Actual*Sales';
run; quit;

ods tagsets.excelxp close;

ods listing;

```

Autofilters

Filters enable you to visually subset your data within Excel. The values for `AUTOFILTER=` are ALL, a column number (e.g., 3), or a range of column numbers (e.g., 3-5).

The example below specifies ALL, which causes Excel to display a drop-down menu that contains all of the values with column. When you select one of these values, the rows that don't match that value collapse.

```
ods listing close;

ods tagsets.excelxp file='filters.xls' style=statistical
  options(autofilter='all');

proc print data=sashelp.class; run;

ods tagsets.excelxp close;

ods listing;
```

Data-Specified Formulas

If a data point starts with an equal sign, that cell is evaluated as a formula in Excel.

```
ods tagsets.excelxp file="dataeq.xml" options(zoom='75' debug_level='5');

/* Interestingly, this first example doesn't work, although the XML looks
 * perfectly fine. Excel doesn't seem to like absolute cell
 * references. If the formulas are fixed, and saved, the
 * resulting XML replaces all absolute references with relative
 * ones.
 */

data foo;
  length a b 8 c $20;
  input a b c $;
  cards;
  1 2 3
  2 3 =A3+B3
  3 4 =A4+B4
  . . =SUM(C2:C4)
  ;
run;

proc print noobs;
run;

data foobar;
  length a b 8 c $20;
  input a b c $;
  cards;
  1 2 3
  2 3 =RC[-2]+RC[-1]
  3 4 =RC[-2]+RC[-1]
  . . =SUM(R[-3]C:R[-1]C)
  ;
run;

proc print noobs;
  var a b;
  var c / style(head) = {flyover="Hello World"}
           style(data) = {cellwidth=10pt};
run;

ods tagsets.excelxp close;
```

Formats and Formulas

You can use Excel formats and formulas by embedding them in the TAGATTR= style option.

```
data prdsale;
```

```

        set sashelp.prdsale;
        difference = actual-predict;
run;

proc sort data=prdsale; by country region division year; run; quit;

title; footnote;

ods listing close;

ods tagsets.excelxp file='formulas.xls' style=statistical;

proc print data=prdsale noobs label split='*';
    where country eq 'CANADA' and year eq 1993;
    id country region division;
    var prodtype product quarter month year;
    sum predict / style={tagattr='format:Currency'};
    sum actual / style={tagattr='format:Currency'};
    sum difference / style={tagattr='format:Currency formula:RC[-1]-RC[-2]'};
    label prodtype = 'Product*Type'
           predict  = 'Predicted*Sales'
           actual   = 'Actual*Sales';
run; quit;

ods tagsets.excelxp close;

ods listing;

```

Embedded Titles and Footnotes

Use the EMBEDDED_TITLES= and EMBEDDED_FOOTNOTES= options.

```

ods listing close;

ods tagsets.excelxp file="embed.xls" style=statistical
    options( embedded_titles='yes' embedded_footnotes='yes' );

title "My First Title";
title3 "My Third Title";

proc print data=sashelp.class (obs=5); run;

footnote "My First Footnote";
footnote3 "My Third Footnote";

ods tagsets.excelxp close;

ods listing;

```

Default Column Width and Width Fudge

You can set the default character widths of columns by using the DEFAULT_COLUMN_WIDTH= option. The WIDTH_F option is also used to adjust the column width based on font parameters. The default width fudge is 0.75.

```

title; footnote;

*;
*  Illustrate the use of split characters and column justification.
*  Use WIDTH_FUDGE= to get slightly wider columns.
*;

ods listing close;

```

```
ods tagsets.excelxp file='report.xml' style=statistical options(width_fudge='0.8');

proc print data=sashelp.class noobs label split='*';
  var name          / style={just=r};
  var age sex       / style={just=c};
  var height weight / style={just=l};
  label name      = 'Student*Name'
        age       = '*Age'
        sex       = '*Gender'
        height    = 'Height*(inches)'
        weight    = 'Weight*(pounds)';
run; quit;

*;
* PROC REPORT is not setting column widths.  So we must force them with
* DEFAULT_COLUMN_WIDTH=.  Reset WIDTH_FUDGE= to the default value of "0.75".
*;

ods tagsets.excelxp options(default_column_width="7.5, 7.5, 5, 7.5, 7.5" width_fudge=

proc report data=sashelp.class nowindows split='*';
  column name sex age height weight;
  define name / display 'Student*Name' left style={just=l};
  define sex / display '*Gender' right style={just=r};
  define age / display '*Age' center style={just=c};
  define height / display 'Height*(inches)' center style={just=c};
  define weight / display 'Weight*(pounds)' center style={just=c};
run; quit;

ods tagsets.excelxp close;

ods listing;
```

Rotated Column Headers

You can use the `vert_excelxp` tagset to create tables with rotated column headers.

```
/*-----*/
/*-- Column widths will be too wide because the tagset thinks --*/
/*-- the headers are horizontal. --*/
/*-----*/

proc template;
  define style styles.mystyle;
    parent=styles.default;
    style vertical_header from header /
      tagattr = 'rotate:45'
  ;
end;
run;

ods tagsets.excelxp
  style=mystyle
  file="test.xls"
  options(absolute_column_width="4,8,4,3,4,5"
    row_heights="30"
  );

proc print data=sashelp.class;
  var name / style(header) = vertical_header;
  var age sex;
  var weight height / style(header) = vertical_header;
run;

ods _all_ close;
```

Flyovers

You can create a tooltip for a cell by using the FLYOVER= style attribute.

```
proc sort data=sashelp.prdsale out=prdsale;
  by country region division year;
run; quit;

title; footnote;

ods listing close;

ods tagsets.excelxp file='flyover.xls' style=statistical;

proc print data=prdsale noobs label split='*';
  where country eq 'CANADA' and year eq 1993;
  id country region division;
  var prodtype product quarter month year;
  sum predict / style={tagattr='format:Currency' flyover='Predicted Total'};
  sum actual / style={tagattr='format:Currency' flyover='Actual Total'};
  label prodtype = 'Product*Type'
         predict  = 'Predicted*Sales'
         actual   = 'Actual*Sales';
run; quit;

ods tagsets.excelxp close;

ods listing;
```