

ExcelXP Tagset Help

SAS 9.1.3, v1.32, 2/6/2006 Release

This tagset/destination creates Microsoft's spreadsheetML XML. It is used specifically for importing data into Excel.

Each table will be placed in its own worksheet within a workbook. This destination supports ODS styles, traffic lighting, and custom formats.

Numbers, currency and percentages are correctly detected and displayed. Custom formats can be given by supplying a style override on the TagAttr style attribute.

By default, titles and footnotes are part of the spreadsheet, but are part of the header and footer.

Also by default, printing will be in 'Portrait'. The orientation can be changed to 'Landscape'.

The specification for this xml is here.

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnexcl2k2/html/odc_xlsmlinss.asp

See Also:

<http://support.sas.com/rnd/base/topics/odsmarkup/>

<http://support.sas.com/rnd/papers/index.html#excel2006>

<http://support.sas.com/rnd/papers/index.html#office2005>

These are the options supported by this tagset.

Sample usage:

```
ods tagsets.excelxp file='test.xml' data='test.ini' options(doc='help');  
ods tagsets.excelxp options(doc='quick');  
ods tagsets.excelxp options(embedded_titles='no' orientation='landscape');
```

Doc: No default value.

Values: 'Help', 'Quick'.

'Help' displays introductory text and options and 'Quick' displays available options.

Orientation: Default value: 'Portrait'

Values: 'Portrait', 'Landscape'.

Tells Excel how to format the page when printing. The only other value is 'landscape'. Also available as a macro variable.

Embedded_Titles: Default value: 'No'

Values: 'Yes', 'No'.

If 'Yes', titles will appear in the worksheet. By default, titles are a part of the print header and footer. Also available as a macro variable.

Embedded_Footnotes: Default value: 'No'

Values: 'Yes', 'No'.

If 'Yes', footnotes will appear in the worksheet. By default, footnotes are a part of the print header and footer. Also available as a macro variable.

Embed_Titles_Once: Default value: 'No'

Values: 'Yes', 'No'.

If 'Yes', embedded titles will only appear at the top of each worksheet.

Print_Header: Default value: ""

If embedded titles are on, this value will be used as the header for printing. Everything about the appearance of the 3 part header can be controlled with this value. The Excel syntax for this string follows. Of course the easiest way to create a header or footer is to do it in Excel. Then save the workbook to XML. Search the XML for '<Header' or '<footer'. A simple cut and past will make it a part of your SAS program.

```
&L <Left header> &C <Center header> &R <Right Header>
```

A very simple example is this:

```
&LLeft header text&CCenter header text&RRight Header Text
```

Newlines can be introduced by inserting `` within the text. Other special values follow.

```
Newline:      &#13;
Page Number:  &P
Pages:        &N
Date:         &D
Time:         &T
File Path:    &Z&F
File:         &F
Sheet Name:   &A
Underline:    &U  One to start underlining, another to stop it.
Font Size:    &8  Persists until changed.
```

The font size can be controlled by placing the font size, in points, right before the text. This is a left sided header with a font size of 8.

```
&L&&8This is a test;
```

The font, bold and italic can be changed using this syntax.

```
&quot; <font name>,<Bold> <Italic> &quot;
```

This example changes the font, turns on bold and italic, changes the font size and turns underline on and off.

```
&L&&quot;Palatino,Bold Italic&quot;&9&UThis is a test&U
```

This is a complete example, showing the various possibilities:

```
&L&&quot;Palatino,Bold Italic&quot;&9&UThis is underlined &U
This is not &#13;&#12This is bigger&CThis is the Center&#13;Page:
&P&#13;Pages: &N&#13;Date: &D&#13;Time: &T&#13;Path:
&Z&F&#13;File: &F&#13;Sheet: &A&R&#14This is bigger and on
the right&#13;&P
```

Print_Footer: Default value: ""

If embedded footers are on, this value will be used as the footer for printing. Everything about the appearance of the 3 part footer can be controlled with this value. The syntax for this value is the same as that for the Print_Header option.

Suppress_Bylines: Default value: 'No'

Values: 'Yes', 'No'.

If 'Yes', BY lines will not appear in the worksheet. This is useful with PROC PRINT because turning off BY lines will defeat the tagset's BY group processing abilities when using PROC PRINT.

Zoom: Default value: '100'

Values: number (percentage).

This value determines the zoom level on the worksheet.

Scale: Default value: '100'

Values: number (percentage).

This value determines the scale level for printing.

DPI: Default value: '300'

Values: number.

This value determines the dots per inch for printing.

Center_Vertical: Default value: 'No'

Values: 'Yes', 'No'.

This value controls vertical centering for printing.

Center_Horizontal: Default value: 'No'

Values: 'Yes', 'No'.

This value controls horizontal centering for printing.

Row_Repeat: Default value: 'None'

Values: 'None', number, range, 'Header'.

If a number is specified, that row will be repeated across pages if a worksheet breaks across pages when printing. If a range such as '3-5' is specified, that range of rows will be repeated. If 'Header' is specified, the table headers for the first table of the worksheet will be repeated.

Column_Repeat: Default value: 'None'

Values: 'None', number, range, 'Header'.

If a number is specified, that column will be repeated across pages if a worksheet breaks across pages when printing. If a range such as '3-5' is specified, that range of columns will be repeated. If 'Header' is specified, the columns that contain the row headers for the first table of the worksheet will be repeated.

Frozen_Headers: Default value: 'No'

Values: 'Yes', 'No', number.

If 'Yes', the rows down to the bottom of the headers will be frozen when the table data scrolls. This includes any titles created with the Embedded_Titles option. If a number is given, that is the row count that will be frozen. Also available as a macro variable.

Frozen_RowHeaders: Default value: 'No'

Values: 'Yes', 'No', number.

If 'Yes', the header columns on the left will be frozen when the table data scrolls. If a number is given, that is the column count that will be frozen. Also available as a macro variable.

AutoFilter: Default value: 'None'

Values: 'None', 'All', range.

If 'All', an AutoFilter will be applied to all columns. If a range such as '3-5' is specified, the AutoFilter will be applied to the in that range of columns.

AutoFilter_Table: Default value: '1'

Values: number.

If sheet interval is anything but 'Table' or 'Bygroup', this value determines which table gets the AutoFilter applied. If the sheet interval is 'Table', or 'Bygroup', the only table in the worksheet gets the AutoFilter, regardless of this setting.

Formulas: Default value: 'Yes'

Values: 'Yes', 'No'.

By default, data values that start with an '=' will become formulas instead of cell values. This behavior can be turned off by setting this option to 'No'. Excel only understands relative column references in its XML. A formula like SUM(C2,C3) or A2+B3 will not work. An equivalent might be SUM(R[-2]C,R[-1]C) or RC[-2]+RC[-1]. See the PROC PRINT example in the "Using Style Attributes" section.

Width_Fudge: Default value: '0.75'

Values: 'None', number.

By default, this value is used along with Width_Points and column width to calculate an approximate width for the table columns.

$$\text{width} = \text{Data_Font_Points} * \text{Number_Of_Chars} * \text{Width_Fudge}$$

If 'None' is specified, this feature is turned off.

Width_Points: Default value: 'None'

Values: 'None', number.

By default, the point size from the data or header style classes are used to calculate a pseudo column width. The column width is calculated from the given column width or the length of the column's header text, if the header is bigger. In the case where the header length is used, the header's point size is also used. This value overrides that point size. This value is used along with Width_Fudge and column width to calculate an approximate width for the table columns.

$$\text{width} = \text{Width_Points} * \text{Number_Of_Chars} * \text{Width_Fudge}$$

Default_Column_Width: Default value: 'None'

Values: 'None', number, list of numbers.

Most procedures provide column widths, but occasionally a column will not have a width. Excel will resize the column to fit any numbers but will not auto-size for character string headings. In the case that a column does not have a width, this value will be used instead. The value should be the width in characters. If the value of this option is a comma separated list, each number will be used for the column in the same position. If the table has more columns, the list will start over again.

Absolute_Column_Width: Default value: 'None'

Values: 'None', number, list of numbers.

This option works similarly to the Default_Column_Width option. The difference is that these widths will be used regardless of any column widths the procedure might provide. The value should be the width in characters. If the value of this option is a comma separated list, each number will be used for the column in the same position. If the table has more columns, the list will start over again.

Row_Heights: Default value: '0, 0, 0, 0, 0'

Values: list of numbers.

This option controls how tall the rows will be for each type of row. The numbers are in points. By default, the values will be taken from the font size used for the row. The font sizes are collected from the style class definitions for each item. The table row height is defined by the font size in the header style.

The parameters of this option are positional, but not all values must be specified. A value of 0 means the height should be taken from the style class. The first value is the height for table rows. The second value is the row height for bylines. The third is for titles, the fourth is for footers, the fifth is the pagebreak height, and the last value is the height for paragraph skip. The default values are:

Table: 0
Byline: 0
Title: 0
Footer: 0
PageBreak: 0
Parskip: 0

Row_Height_Fudge: Default value: '4'

Values: number.

This value is added to the row height for each row. The additional height makes the spreadsheet easier to read.

Autofit_Height: Default value: 'No'

Values: 'Yes', 'No'

If 'Yes', no row heights will be specified. This allows the auto fit height feature of Excel to do its job... sometimes not so well.

Sheet_Interval: Default value: 'Table'

Values: 'Table', 'Page', 'Bygroup', 'Proc', 'None'.

This option controls how many tables will go in a worksheet. In reality only one table is allowed per worksheet. To get more than one table, the tables are actually combined into one.

Specifying a sheet interval will cause the current worksheet to close. It is recommended that this always be the first option, to insure that the options following it apply to the new worksheet, rather than the previous worksheet.

Sheet_Name: Default value: 'None'

Values: 'None', string.

Worksheet names can be up to 31 characters long. This name will be used in combination with a worksheet counter to create a unique name. Specifying 'None' will result in the tagset controlling the worksheet name.

Sheet_Label: Default value: 'None'

Values: 'None', string.

This option is used in combination with the various worksheet naming, heuristics, which are based on the sheet interval. This string will be used as the first part of the name instead of the predefined string it would normally use.

These are the defaults:

```
'Proc ' total_Proc_count - label
'Page ' total_page_count - label
'By ' numberOfWorksheets byGroupLabel - label
'Table ' numberOfWorksheets - label
```

Specifying 'None' will result in the tagset controlling the worksheet name.

Contents_Workbook: Default value: 'Contents, Index'

Values: 'Contents', 'Index', 'Workbooks', 'All'.

If set to 'All', the contents file will contain 3 worksheets, a list of workbooks, a hierarchical table of contents, and a list of worksheets.

Contents: Default value: 'No'

Values: 'Yes', 'No'.

If set to 'Yes', the first worksheet will contain a table of contents with links to each worksheet in the workbook.

Index: Default value: 'No'

Values: 'Yes', 'No'.

If set to 'Yes', the first worksheet will contain a table of contents with a single link to each worksheet in the workbook. If both this option and the Contents option are specified, then the index of worksheets will be the second worksheet and it will be named 'Worksheets'.

Missing_Align: Default value: 'Right'

Values: 'Left', 'Center', 'Right'.

Sets the horizontal alignment for missing values. By default a 'datamissing' style class is created from the 'data' style class. The 'datamissing' style class is created in 3 versions, one for each justification. When a style class has the string 'data' in its name, the data value is checked. If the data value is missing, then the 'datamissing' style class will be used instead. A 'datamissing' style class can be provided in the style definition. If found, the tagset will use that style class as a basis for the 3 'datamissing' styles classes.

Auto_SubTotals: Default value: 'No'

Values: 'Yes', 'No'.

If 'Yes', this option causes a subtotal formula to be placed in the subtotal cells on the last table row of the PRINT procedure's tables. **WARNING:** This does not work with SUMBY; it works only if the totals happen once per table.

Convert_Percentages: Default value: 'Yes'

Values: 'Yes', 'No'.

If 'Yes', remove percent symbol, apply excel percent format, and multiply by 100. This causes percentage values to display as numeric percentages in Excel. If 'No', percentage values will be untouched and will appear as strings in Excel. This option will be deprecated in a future release, when it is no longer needed.

Currency_Symbol: Default value: '\$'

Values: string.

Used for detection of currency formats and for removing those symbols so Excel will like them. This option will be deprecated in a future release, when it is no longer needed.

Currency_Format: Default value: 'Currency'

Values: Excel format.

The currency format for Excel to use for monetary data. Another possible value is 'Euro Currency'. This option will be deprecated in a future release, when it is no longer needed.

Decimal_Separator: Default value: '.'

Values: string.

The character used for the decimal point. This option will be deprecated in a future release, when it is no longer needed.

Thousands_Separator: Default value: ','

Values: string.

The character used for indicating thousands in numeric values. Used for removing those symbols from numeric values so Excel will like them. This option will be deprecated in a future release, when it is no longer needed.

Numeric_Test_Format: Default value: '12.'

Values: SAS numeric format.

Used for determining if a value is numeric or not. Other useful values might be the COMMAX. or NLNUM. formats. This option will be deprecated in a future release, when it is no longer needed.

Minimize_Style: Default value: 'No'

Values: 'Yes', 'No'.

If set to 'Yes', the style definitions will be filtered so that only the most necessary definitions are included. This can have the reverse effect if style attribute overrides are used on PROC statements. It is best to define a new style with the appropriate overrides built in. The procedure can use the new style, but without individual attribute overrides. The result is a much smaller style section. In that case, this option should be set to 'No'.

Skip_Space: Default value: '1, 0, 1, 1, 1'

Values: list of numbers.

This option controls how much space follows the different types of output that can occur within a worksheet. The number given is a multiplier that is used against the height given in the 'parskip' style class. In the absence of the 'parskip' style class, the font size from the 'header' style class is used. The parameters of this option are positional, but not all values must be specified. The first value is for the space following each table. The second value is the space following BY lines. The third is for titles, the fourth is for footers and the last value is the space following page breaks if the PageBreaks option is turned on and a 'pagebreak' style element exists.

The default values are:

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

PageBreaks: Default value: 'No'

Values: 'Yes', 'No'.

If set to 'Yes', page breaks will be inserted into the style definition. The 'pagebreak' style class will be used to define what that page break looks like. A sample style class definition looks like this.

```
style pagebreak / cellheight=8 foreground=black tagattr="HorzStripe"
```

It is not necessary to define a 'pagebreak' style class. In its absence, a blank row will be inserted.

Configuration_File: Default value: ''

Values: Any valid file name.

The name of the configuration file to read. This is a .ini formatted file as written to the data file, if one was specified. If specified, the options for the configuration will be loaded on top of any options given on the ODS statement. A file may contain more than one configuration section. Only the first section that matches the configuration name will be loaded. See also, Configuration_Name.

Configuration_Name: Default value: 'default'

Values: string.

The name of the configuration to read or write in the .ini file.

Debug_Level: Default value: '0'

Values: 0, 1, 2, 3, 4, or 5.

Determines what level of debugging information should be printed to the SAS Log. Higher values cause more information to be printed.

Using Style Elements

There are a few style attributes that can be used to good effect in the ExcelXP tagset. The TagAttr attribute can be used to add formulas and formats. Cellwidth can be used to control the column widths. Flyover can be used to add comments to cells. A URL on a cell will cause it to be a link. Additionally, formulas can be given as the actual data values.

An alternative to setting widths is to use the cellwidth / width style attributes. This value will be used regardless of any other column width calculations. Cellwidth can be specified in any of these units: inch, centimeter, millimeter, points or pixel. If a cell width for column is given more than once, the first width is used. This can happen when there is more than one table per worksheet.

The following example shows formulas as data, comment text on a header, and absolute control of column widths.

```
ods tagsets.excelxp file='test.xml' options(zoom='75');

data test;
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 data=test noobs;
  var a b;
  var c / style(head) = {flyover='Hello World'}
         style(data) = {cellwidth=50pt};
run; quit;

ods tagsets.excelxp close;
```

TagAttr: Default value: "

Values: <Excel format> or <format: Excel format> <formula: Excel formula>

This is not a tagset option but a style attribute that the tagset will use to set formulas and column formats. The format and formulas given must be a valid to Excel. A single value without a keyword is interpreted as a format. Both a formula and format can be specified together using keywords. There should be no spaces except for those between the two values. The keyword and value must be separated by a ':'. For example:

```
tagattr='format:###.## formula:SUM(R[-4]C:R[-1]C)'
```

Margins:

Margins can be set two ways: with the system options or through style classes. The system options take precedence over the style attributes. In the style, the margins must be set on the 'body' style class.

Setting the margins with the options statement is the easiest thing to do:

```
options topmargin=1in
        bottommargin=1in
        leftmargin=.5in
        rightmargin=.5in;
```

As a style class definition, the 'body' class might look like this:

```
style body from body / topmargin=.5in leftmargin=.25in;
```

This approach is more reusable, since each program that uses the style automatically gets the margins.

Center/NoCenter system options

Setting the center or nocenter options will cause titles, footnotes and BY lines to be centered or left justified within the worksheet, respectively. If the nocenter option is set, the cells are not merged. This causes Excel to do a better job of printing when the BY line, title or footnote text is long.
