

Understanding the ExcelXP Tagset

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.

Understanding the ExcelXP Tagset Handout

Copyright © 2009 by SAS Institute Inc., Cary, NC 27513, USA. All rights reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

Prepared on 12Jun2009

Understanding the EXCELXP Tagset

EXCELXP is a tagset destination that creates Microsoft's spreadsheetML. This destination is specifically used to create an XML file that can be opened in Excel 2002 or later.

The EXCELXP destination is available starting in SAS 9.1.

SAS 9.1.3 comes with the v1.28 tagset, 08/29/05.

SAS 9.2 comes with the v1.75 tagset, 07/26/07.

The SAS supplied EXCELXP tagset is stored in the SASHELP.TMPLMST item store in the TAGSETS folder. To view the tagset, type ODSTEMP on the SAS command line, to open the Templates window. Double click on SASHELP.TMPLMST and then double click on the TAGSETS folder.

The latest EXCELXP tagset can be downloaded from
<http://support.sas.com/rnd/base/topics/odsmarkup/excltags.tpl>

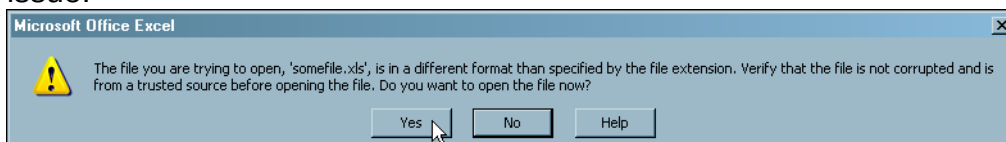
For more information on how to download and install updated versions of the ExcelXP tagset template, refer to this Tech Support note:
<http://support.sas.com/kb/32/394.html>

As of December 2008, the latest version is v1.86, 04/15/08. After downloading, submit the PROC TEMPLATE code to great a new version of EXCELXP. By default, the EXCELXP tagset will be stored in the SASUSER.TEMPLAT item store if using the default ODS PATH statement.

Basic Invocation:

```
ods tagsets.excelxp file='____.xml' . . . ;  
SAS code . . . ;  
ods tagsets.excelxp close;
```

An .XLS extension can be used instead of the .XML extension. However, a Spreadsheet Markup Language XML document is still created, not an Excel workbook. Excel 2007 detects that you have created an XML file and prompts/warns you about the issue.



If using the .XLS extension, you will be able to have Excel open the file automatically when expanding the Results window and double-clicking on the report. However, you must click 'Yes' at this prompt in order to open the file.

Additional Resources:

Office 2003: XML Reference Schemas (SpreadsheetML- Excel 2003)

<http://www.microsoft.com/downloads/details.aspx?familyid=fe118952-3547-420a-a412-00a2662442d9&displaylang=en>

Updated SAS Tagsets to Download Individually

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

Try This Demo: The ExcelXP Tagset and Microsoft Excel

http://support.sas.com/rnd/base/topics/odsmarkup/excelxp_demo.html

Documentation:

To display introductory text and available options in full detail in the SAS log add options(doc='help') to the first ODS statement.

```
ods tagsets.excelxp file='____.xml' options(doc='help') . . .;
```

Short Descriptions of the Supported Options:

Name	Default value	Description
ASCII_DOTS	yes	Turn off/on leading dots in textual 'batch' output
AUTOFILTER	none	Turn on auto filter for all columns or a range of columns.
AUTOFILTER_TABLE	1	Which table on the worksheet should get the filters.
ABSOLUTE_COLUMN_WIDTH	none	List of widths to use for each column in a table no matter what.
AUTOFIT_HEIGHT	no	If yes no row heights will be specified.
AUTO_SUBTOTALS	no	Add a subtotal function to the summary line of proc print.
BLACKANDWHITE	no	This value turns on black and White for printing.
CENTER_VERTICAL	no	This value controls vertical centering for printing
CENTER_HORIZONTAL	no	This value controls horizontal centering for printing
COLUMN_REPEAT	none	Repeat columns across pages when printing.
CONTENTS	no	Create a worksheet that will contain a table of contents
CONTENTS_WORKBOOK	Contents, Index	Create a workbook of with a table of contents and/or an index of workbooks and/or an index of worksheets
CONVERT_PERCENTAGES	yes	Remove percent symbol, apply excel percent format, and multiply by 100.
CURRENCY_SYMBOL	\$	Used for detection of currency formats and for removing symbols so Excel will see currency as numbers
CURRENCY_FORMAT	Currency	The currency format specified for excel to use.
DECIMAL_SEPARATOR	.	The character used for the decimal point.
DEFAULT_COLUMN_WIDTH	none	List of widths to use for each column in a table, if there are no widths
DOC	none	Documentation for this tagset. Valid values are: Help, Options, Quick, Settings and Changelog.
DPI	300	This value determines the dots per inch for printing
DRAFTQUALITY	no	This value turns on draft quality for printing.
EMBED_TITLES_ONCE	no	If 'Yes' Embedded titles will only appear at the top of each worksheet.
EMBEDDED_FOOTNOTES	no	Put footnotes in the worksheet.
EMBEDDED_TITLES	no	Put titles in the worksheet.
FITTOPAGE	no	Fit to Page when printing.
FORMULAS	yes	Data values that start with an '=' will become formulas
FROZEN_HEADERS	no	Freeze rows from scrolling with the scrollbar.
FROZEN_ROWHEADERS	no	Freeze columns from scrolling with the scrollbar.
GRIDLINES	no	This value turns on gridlines for printing.
INDEX	no	Create a worksheet that will contain a index of worksheets
MINIMIZE_STYLE	no	Minimize the styles written to the stylesheet, can cause unloadable XML files.
MISSING_ALIGN	r	Sets the alignment for missing values.
NUMERIC_TEST_FORMAT	12.	Used for determining if a value is numeric or not.
ORIENTATION	Portrait	Print orientation for the worksheet, Portrait or Landscape
PAGE_ORDER_ACROSS	no	If set to yes, the worksheet page order will be set to print across, then down.
PAGEBREAKS	no	Insert page break lines in the worksheet.
PAGES_FITWIDTH	1	This value determines the number of pages to fit the worksheet across when printing.
PAGES_FITHEIGHT	1	This value determines the number of pages down to fit the worksheet when printing.
PRINT_FOOTER	None	If Embedded footers are on, this value will be used as the footer for printing.
PRINT_FOOTER_MARGIN	none	This is the footer margin as set in the page setup dialog window.
PRINT_HEADER	None	If Embedded titles are on, this value will be used as the header for printing.
PRINT_HEADER_MARGIN	none	This is the header margin as set in the page setup dialog window.
ROW_HEIGHT_FUDGE	4	A fudge value to add to the row height for each row.
ROW_HEIGHTS	0,0,0,0,0,0	Positional list of point sizes to use for row heights.

Name	Default value	Description
ROW_REPEAT	none	Repeat rows across pages when printing.
ROWCOLHEADINGS	no	This value turns on row and Column headings for printing.
SCALE	100	This value determines the scale level for printing
SHEET_INTERVAL	Table	Interval to divide the output between worksheets, Table, Page, Bygroup, Proc or None.
SHEET_NAME	none	Worksheet name to use for the next worksheet.
SHEET_LABEL	none	Replace the prefix of the worksheet name with this value.
SKIP_SPACE	1,0,1,1,1	Multiplier for the space that follows the different types of output
SUPPRESS_BYLINES	no	Suppresses bylines in the worksheet.
THOUSANDS_SEPARATOR	,	The character used for indicating thousands in numeric values.
WIDTH_FUDGE	0.75	This value is used along with Width_Points and column width to calculate an approximate width for the table columns.
WIDTH_POINTS	none	Over-ride value for width calculations
ZOOM	100	This value determines the zoom level on the worksheet.
DEBUG_LEVEL	0	Numeric value to turn on various debug messages
CONFIGURATION_FILE	none	An ini file to read option settings from.
CONFIGURATION_NAME	none	A section name in an ini file that holds option settings

Additional Resources:

Printable Spreadsheets Made Easy: Utilizing the SAS® Excel XP Tagset

Rick Andrews, UnitedHealth Group, Cary,NC

<http://www.nesug.org/proceedings/nesug08/ap/ap06.pdf>

```

/* ExcelXP1.sas */

ods tagsets.excelxp file='excelxp1a.xml'
options(doc='help');

ods tagsets.excelxp close;

*****;

ods listing close;
ods tagsets.excelxp file='excelxp1b.xml' style=statistical;

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)';
title 'R&D Report 1';
footnote;
run;

proc report data=sashelp.class nowindows split='*';
column name sex age height weight;
define name / display 'Student*Name' style={just=l};
define sex / display '*Gender' style={just=r};
define age / display '*Age' style={just=c};
define height / display 'Height*(inches)' style={just=c};
define weight / display 'Weight*(pounds)' style={just=c};
title '&L &D &C R&D Report 2 &R Page &P';
footnote;
run;

ods tagsets.excelxp options(absolute_column_width="8, 8, 5, 8, 20"
embedded_titles='yes'
print_header='&L &D &C R&D Report 3 &R Page &P');

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

ods tagsets.excelxp close;
ods listing;

```

```

/* ExcelXP2.sas */

ods tagsets.excelxp file='excelxp2a.xml'
options(doc='help');

ods tagsets.excelxp close;

*****;

ods listing close;
ods tagsets.excelxp file='excelxp2b.xml' style=sasweb
options(absolute_column_width='20');

proc report data=sashelp.shoes nowindows split='*';
  where region='United States';
  column subsidiary product stores sales inventory returns;
  title;
  footnote;
run;

ods tagsets.excelxp options(absolute_column_width='15'
                           sheet_name='Group');

proc report data=sashelp.shoes nowindows split='*';
  where region='United States';
  column subsidiary product stores sales inventory returns;
  define subsidiary / order;
  break after subsidiary / page;
  title;
  footnote;
run;

ods tagsets.excelxp options(sheet_interval='proc'
                           sheet_name='United States'
                           skip_space='2,0,0,0,0');

proc report data=sashelp.shoes nowindows split='*';
  where region='United States';
  column subsidiary product stores sales inventory returns;
  define subsidiary / order;
  break after subsidiary / page;
  title;
  footnote;
run;

ods tagsets.excelxp options(autofilter='1-3'
                           frozen_headers='yes'
                           sheet_name='None');

proc report data=sashelp.shoes nowindows split='*';
  where region='United States';
  column subsidiary product stores sales inventory returns;
  title;
  footnote;
run;

ods tagsets.excelxp close;
ods listing;

```

```

/* ExcelXP3.sas */

ods tagsets.excelxp file='excelxp3a.xml'
  options(doc='help');

ods tagsets.excelxp close;

*****;

options topmargin=1in bottommargin=1in
  leftmargin=1in rightmargin=1in;

ods listing close;
ods tagsets.excelxp file='excelxp3b.xml' style=seaside;

ods tagsets.excelxp options(width_fudge='1'
  width_points='9'
  default_column_width='10');

proc report data=sashelp.shoes nowindows split='*';
  where region='United States';
  column subsidiary product stores sales inventory returns;
  title;
  footnote;
run;

ods tagsets.excelxp options(orientation='landscape'
  zoom='150'
  fittopage='yes'
  gridlines='yes');

proc report data=sashelp.shoes nowindows split='*';
  where region='United States';
  column subsidiary product stores sales inventory returns;
  title;
  footnote;
run;

ods tagsets.excelxp close;
ods listing;
*****;

```

INI Files:

An INI file is an initialization or configuration file that generally contains configuration data for Microsoft Windows-based applications.

The EXCELXP tagset can create an INI file, use an existing INI file, or point to a section within an INI file.

Creating an INI File with EXCELXP:

```

ods tagsets.excelxp file='excelxp4b.xml' style=rsvp
  data='config1.ini'
  options(absolute_column_width='10'
    sheet_interval='proc');

```

Using an INI File with EXCELXP:

```

ods tagsets.excelxp file='excelxp4c.xml' style=rsvp
  options(configuration_file='config1.ini');

```

Pointing to a Section within an INI File with EXCELXP:

```

ods tagsets.excelxp file='excelxp4d.xml' style=rsvp
    options(configuration_file='config2.ini'
            configuration_name='standard1');

/* ExcelXP4.sas */

ods tagsets.excelxp file='excelxp4a.xml'
    options(doc='help');

ods tagsets.excelxp close;

*****;

options topmargin=0.5in bottommargin=0.5in
    leftmargin=0.5in rightmargin=0.5in;

ods listing close;
ods tagsets.excelxp file='excelxp4b.xml' style=rsvp
    data='config1.ini'
    options(absolute_column_width='10'
            sheet_interval='proc'
            sheet_name='United States'
            skip_space='2,0,0,0,0'
            autofilter='1-3'
            frozen_headers='yes'
            orientation='landscape'
            zoom='75');

proc report data=sashelp.shoes nowindows split='*';
    where region='United States';
    column subsidiary product stores sales inventory returns;
    define subsidiary / order;
    break after subsidiary / page;
    title 'Top';
    footnote 'Bottom';
run;

ods tagsets.excelxp close;
ods listing;

*****;

options topmargin=0.5in bottommargin=0.5in
    leftmargin=0.5in rightmargin=0.5in;

ods listing close;
ods tagsets.excelxp file='excelxp4c.xml' style=rsvp
    options(configuration_file='config1.ini');

proc report data=sashelp.shoes nowindows split='*';
    where region='United States';
    column subsidiary product stores sales inventory returns;
    define subsidiary / order;
    break after subsidiary / page;
    title 'Top';
    footnote 'Bottom';
run;

ods tagsets.excelxp close;
ods listing;

*****;

options topmargin=0.5in bottommargin=0.5in
    leftmargin=0.5in rightmargin=0.5in;

ods listing close;
ods tagsets.excelxp file='excelxp4d.xml' style=rsvp
    options(configuration_file='config2.ini'
            configuration_name='standard1');

proc report data=sashelp.shoes nowindows split='*';
    where region='United States';
    column subsidiary product stores sales inventory returns;
    define subsidiary / order;

```

```

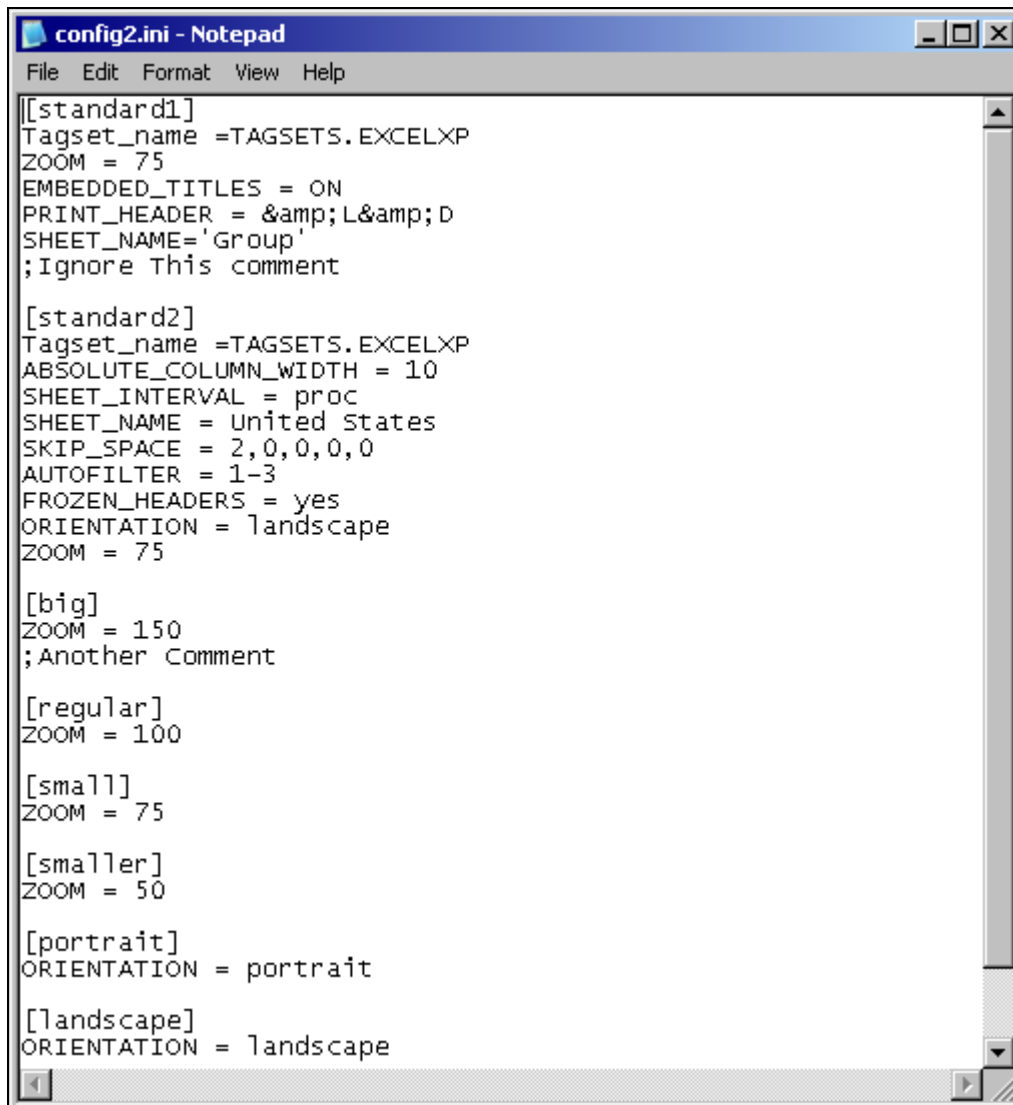
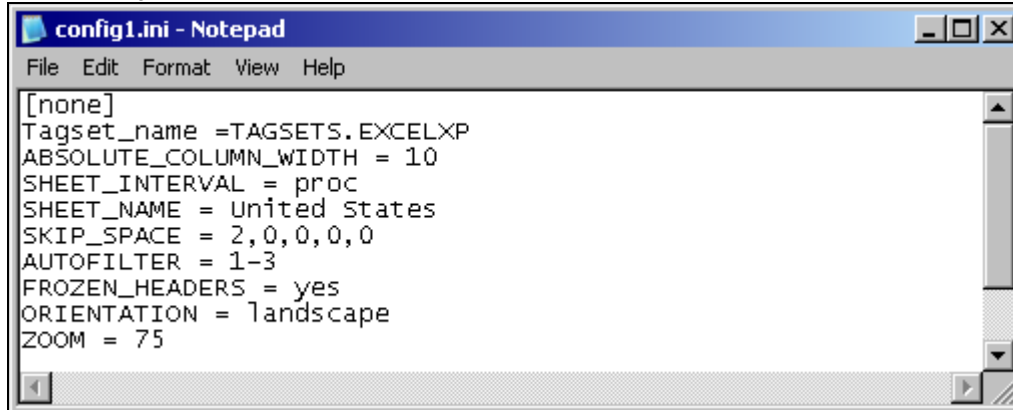
break after subsidiary / page;
title 'Top';
footnote 'Bottom';
run;

```

```

ods tagsets.excelxp close;
ods listing;

```



TAGATTR:

TAGATTR is a style attribute that can be defined on a STYLE statement in a style template or defined on a STYLE= option in the PRINT, REPORT, or TABULATE procedure.

Documentation:

TAGATTR='string' specifies text to insert in the HTML. The string must be valid HTML for the context in which the style element is created. Many style elements are created between <TD> and </TD> tags. To determine how a style element is created, look at the source for the output.

TAGATTR can be used in conjunction with the EXCELXP tagset. FORMAT, TYPE, FORMULA, ROTATE, and HIDDEN are examples of elements that can be specified in the TAGATTR style attribute for EXCELXP.

A single value without a keyword is interpreted as a format.

```
style(column)={tagattr='$#,###.00'}
```

A keyword and a value must be separated by a :. There should be no space between the : and the value.

```
style(column)={tagattr='format:0.0%'}
```

Additional Resources:

Try This Demo: The ExcelXP Tagset and Microsoft Excel

http://support.sas.com/rnd/base/topics/odsmarkup/excelxp_demo.html

ODS Markup, Tagsets, and Styles!

<http://www2.sas.com/proceedings/forum2007/225-2007.pdf>

ODS MARKUP: The SAS® Reports You've Always Dreamed Of

<http://www2.sas.com/proceedings/sugi30/085-30.pdf>

```

/* TagAttr1.sas */

data ticket;
  input sample $ dest $ type $ amt pct date : mmddyy10.;
  cards;
00010 CHICAGO TEL 100 .10 12/01/2005
00222 CHICAGO TEL 200 .49 12/01/2005
00330 GENEVA WEB 300 .30 12/02/2005
00040 GENEVA WEB -400 .22 12/03/2005
00550 LONDON TEL 500 .78 12/05/2005
00066 LONDON TEL 600 .54 12/05/2005
00777 LONDON WEB 700 .19 12/06/2005
00088 PARIS TEL 800 .88 12/07/2005
00990 PARIS WEB 900 .23 12/07/2005
;
run;

title;
footnote;
ods listing close;
ods tagsets.excelxp file='tagattr1.xml' style=statistical;

proc report data=ticket nowd;
  column dest sample date amt pct;
  define date / format=date9.;
run;

/* SAS dates reference January 1, 1960.
Excel dates reference January 1, 1900.
To create an Excel date, add 21,916 to a SAS date.
There are 21,916 days between January 1, 1900 and January 1, 1960.
December 01, 2005 is 16,771 in SAS and 38,687 in Excel. */

proc report data=ticket nowd;
  column dest sample date exceldate amt pct;
  define sample / style(column)={tagattr='00000'};
  define date / style(column)={tagattr='format:m/d/y'};
  define exceldate / style(column)={tagattr='format:m/d/y'};
  define amt / style(column)={tagattr='$#,###.00'};
  define pct / style(column)={tagattr='format:###%'};
  compute exceldate;
    exceldate=date.sum+21916;
  endcomp;
run;

/* To Determine Excel Formats (TagAttr0.xml):
1. Open a blank Excel workbook.
2. In the first row, put individual values with proper formats.
3. Save file as type XML Spreadsheet.
4. Open XML document in Notepad. */

proc report data=ticket nowd;
  column dest sample date exceldate amt pct;
  define sample / style(column)={tagattr='format:@'};
  define date / noprint;
  define exceldate / style(column)={tagattr='format:[$-F800]dddd\, \ mmmm\ dd\, \ yyyy'};
  define amt / style(column)={tagattr='format:&quot;$$&quot;;#,##0.00;[Red]&quot;$$&quot;;#,##0.00'};
  define pct / style(column)={tagattr='format:0.0%'};
  compute exceldate;
    exceldate=date.sum+21916;
  endcomp;
run;

ods tagsets.excelxp close;
ods listing;

```

- /* To Determine Excel Formats (TagAttr0.xml):
1. Open a blank Excel workbook.
 2. In the first row, put individual values with proper formats.
 3. Save file as type XML Spreadsheet.
 4. Open XML document in Notepad. */

	A	B	C	D	E
1	000123	Friday, October 26, 1900	\$600.00	10.0%	
2					
3					

```

<style ss:ID="s21">
  <NumberFormat ss:Format="@"/>
</style>
<style ss:ID="s22">
  <NumberFormat ss:Format="[$-F800]dddd\, \ mmmm\ dd\, \ yyyy"/>
</style>
<style ss:ID="s24">
  <NumberFormat ss:Format="0.0%"/>
</style>
<style ss:ID="s25">
  <NumberFormat ss:Format=""$$";#,##0.00; [Red]&quot;$$";#,##0.00"/>
</style>
</styles>
<worksheet ss:Name="Sheet1">
  <table ss:ExpandedColumnCount="6" ss:ExpandedRowCount="1" x:FullColumns="1"
    x:FullRows="1">
    <column ss:Index="2" ss:width="115.5"/>
    <column ss:Index="4" ss:width="48.75"/>
    <row>
      <cell ss:StyleID="s21"><data ss:Type="String">000123</data></cell>
      <cell ss:StyleID="s22"><data ss:Type="DateTime">1900-10-26T00:00:00.000</data></cell>
      <cell ss:StyleID="s25"><data ss:Type="Number">600</data></cell>
      <cell ss:StyleID="s24"><data ss:Type="Number">0.1</data></cell>
      <cell ss:Index="6" ss:StyleID="s21"/>
    </row>
  </table>

```

Traffic Lighting:

Traffic lighting is the act of highlighting individual cells based on the cell's value.

Traffic lighting can be performed in ODS destinations that use style attributes such as HTML, MSOFFICE2K, PDF, RTF, and EXCELXP.

Two common methods for performing traffic lighting:

- Using CALL DEFINE in a COMPUTE block in the REPORT procedure
- Referencing user-defined formats in the STYLE= option in the PRINT, REPORT, or TABULATE procedures

Additional Resources:

ODS FAQs

<http://support.sas.com/faq/032/FAQ03257.html>

<http://support.sas.com/faq/035/FAQ03597.html>

```

/* TrLight1.sas */

data new;
  input year $ qtr $ num;
  cards;
2006 Q1 27
2006 Q2 40
2006 Q3 21
2006 Q4 12
2007 Q1 25
2007 Q2 45
2007 Q3 15
2007 Q4 25
;
run;

proc format;
  value fore low -< 20 = 'white'
             20 -< 40 = 'black'
             40 - high = 'white';
  value back low -< 20 = 'red'
             20 -< 40 = 'white'
             40 - high = 'green';
run;

ods listing close;
ods tagsets.excelxp file='trlight1.xls' style=sasweb;

options nodate nonumber;
footnote;

proc report data=new nowd;
  compute num;
    if 0<=num.sum<20 then
      call define(_col_, "style", "style={foreground=white background=red}");
    else if 20<=num.sum<40 then
      call define(_col_, "style", "style={foreground=black background=white}");
    else if 40<=num.sum<100 then
      call define(_col_, "style", "style={foreground=white background=green}");
  endcomp;
  title 'Report 1 - Call Define';
run;

proc report data=new nowd;
  compute num;
    count+1;
    if mod(count,2) then
      call define(_row_, "style", "style={background=pink}");
  endcomp;
  title 'Report 2 - Call Define';
run;

proc report data=new nowd;
  define num / style(column)={foreground=fore. background=back.};
  title 'Report 3 - Format with PROC REPORT';
run;

proc print data=new;
  var year qtr;
  var num / style(column)={foreground=fore. background=back.};
  title 'Report 4 - Format with PROC PRINT';
run;

proc tabulate data=new;
  class year qtr;
  var num;
  table qtr, year*num*(n sum*{style={foreground=fore. background=back.}});
  title 'Report 5 - Format with PROC TABULATE';
run;

ods _all_ close;
ods listing;

```

