ABSTRACT

The SAS Excel XP ODS Tagset can be used to generate multi-sheet Excel workbooks directly from SAS. The output of PROC REPORT sent to Excel with the Tagset can be nicely formatted for printing using a variety of programming tips and techniques specific to PROC REPORT, such as the use of COMPUTE blocks. These techniques allow the programmer to apply a variety of formats, both automatically and conditionally, to specific table headers, rows, columns, and even individual cells. Other formatting techniques available through the Excel XP ODS Tagset include page setup, headers and footers, titles, the insertion of line feeds and special characters in text with the use of PROTECTSPECIALCHARS = OFF, and the use of in-line formatting and ODS ESCAPECHAR.

INTRODUCTION

The Excel XP ODS Tagset enables the creation of multi-sheet Excel workbooks directly from SAS, where each sheet can be a different SAS-generated table or report. Generating many tables in a single Excel workbook file has advantages over generating separate and individual output files for each report. The main advantages of having many reports in one multi-sheet Excel workbook are:

- Organization of tables
- Version control of tables

The many tables stored in a single Excel workbook can be appropriately ordered and more easily managed than if the tables were individual files. All tables included in a multi-sheet Excel workbook also must be generated at the same time, so the problem of accidentally using different tables generated at different times is no longer an issue.

One of my initial concerns about generating multi-sheet Excel workbooks containing SAS report output was that the Excel output reports could not be made to look as nice as individual .RTF files. A variety of well-documented techniques are available to improve the formatting of .RTF files, such as in-line formatting and the use of ODS ESCAPECHAR.

When I first started trying to reproduce .RTF tables in multi-sheet Excel files created by the ExcelXP Tagset, I couldn’t get the tables to look nearly as nice as individual tables generated in .RTF format. I could find lots of documentation about the basics of using the Tagset, but I couldn’t find much information about how to improve the appearance and format of the Tagset’s output. However, I was so impressed by the ease of using the ExcelXP Tagset and by the convenience of having dozens (and sometimes hundreds) of individual tables stored in one document, I was determined to figure out how to make the tables produced by the Tagset look just as good as those produced in .RTF format. I also wanted SAS to do the formatting work; I didn’t want any aspect of the formatting process to have to be manually adjusted.

Reports included as individual sheets in the Excel workbook can be generated with PROC PRINT, PROC TABULATE, or PROC REPORT. I tend to use PROC REPORT to generate tables in the Excel workbooks because of specific features found in PROC REPORT and the amount of control available to format the document. The ExcelXP Tagset also has ways to improve the appearance and use of Excel workbooks meant to be used directly as Excel data, rather than as a “printable” table. However, the focus of this paper is on creating “printable” Excel tables.

THE SAS EXCEL XP TAGSET

The ExcelXP Tagset is a piece of software that can be downloaded for free on the SAS website at http://support.sas.com/rnd/base/ods/odsmarkup

As of writing this paper, the most recent version of the Tagset is version 1.86 for SAS version 9.1 or later. Many excellent publications about how to use the Tagset are included as references at the end of this paper.
PROC TEMPLATE

PROC TEMPLATE is very useful in setting up the basic formatting of tables created by the ExcelXP Tagset. A paper that discusses the basics of using PROC TEMPLATE is also included as a reference.

The SAS code for the PROC TEMPLATE used for the examples in this paper is given below.

```sas
proc template;
  define style Styles.exp;
  parent = styles.minimal;

  replace fonts / 'TitleFont2' = ("Arial",10pt,Bold Italic)
    'TitleFont' = ("Arial",10pt,Bold Italic)
    'StrongFont' = ("Arial",10pt,Bold)
    'EmphasisFont' = ("Arial",10pt,Italic)
    'FixedEmphasisFont' = ("Arial",10pt,Italic)
    'FixedStrongFont' = ("Arial",10pt,Bold)
    'FixedHeadingFont' = ("Arial",10pt,Bold)
    'BatchFixedFont' = ("SAS Monospace, Courier",6.7pt)
    'FixedFont' = ("Arial",10pt)
    'headingEmphasisFont' = ("Arial",10pt,Bold Italic)
    'headingFont' = ("Arial",10pt,Bold)
    'docFont' = ("Arial",10pt);

/*-------------------------*/
/* Style for Excel table */
/*-------------------------*/
  style table from table / font_face = "Arial"
    font_weight = medium
    font_style = roman
    font_size = 1
    cellpadding = 5

; /*-------------------------*/
/* Style for Excel titles */
/*-------------------------*/
  style systemtitle from titlesandfooters / just = left
    font_face = "Arial"
    font_weight = bold
    font_size = 2

; /*-------------------------*/
/* Style for Excel footnotes */
/*-------------------------*/
  style systemfooter from titlesandfooters / just = left
    font_face = "Arial"
    font_weight=medium
    font_size = 1;

end;
run;
quit;
```

MULTI-SHEET EXCEL XP TAGSET OUTPUT

The multi-sheet Excel workbook generated by the programming examples included in this paper is shown below.
These two tables were created and put into this multi-sheet Excel workbook directly from SAS without any additional modification. In print-preview mode, these tables would appear as:
Table 1. Baseline Characteristics of Treatment Groups

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Drug A</th>
<th>Drug B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age years (mean (SD), median, range)</td>
<td>49.2 (12.8)</td>
<td>49.9 (9.7)</td>
</tr>
<tr>
<td></td>
<td>50.0, 18.0-72.0</td>
<td>51.0, 20.0-72.0</td>
</tr>
<tr>
<td>Age categorized (n, %)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18-39</td>
<td>344 (25.3%)</td>
<td>262 (15.7%)</td>
</tr>
<tr>
<td>40-49</td>
<td>309 (22.7%)</td>
<td>513 (30.7%)</td>
</tr>
<tr>
<td>50-64</td>
<td>530 (39.0%)</td>
<td>812 (48.6%)</td>
</tr>
<tr>
<td>65+</td>
<td>177 (13.0%)</td>
<td>83 (5.0%)</td>
</tr>
<tr>
<td>Gender (n, %)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>575 (42.3%)</td>
<td>637 (38.1%)</td>
</tr>
<tr>
<td>Female</td>
<td>785 (57.7%)</td>
<td>1033 (61.9%)</td>
</tr>
<tr>
<td>Baseline comorbidities (n, %)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hypertension</td>
<td>773 (56.8%)</td>
<td>1123 (67.2%)</td>
</tr>
<tr>
<td>Coronary artery disease</td>
<td>242 (17.8%)</td>
<td>246 (14.7%)</td>
</tr>
<tr>
<td>Diabetes</td>
<td>208 (15.1%)</td>
<td>362 (21.7%)</td>
</tr>
<tr>
<td>Heart failure</td>
<td>71 (5.2%)</td>
<td>64 (3.8%)</td>
</tr>
<tr>
<td>Malignancy</td>
<td>80 (5.9%)</td>
<td>76 (4.6%)</td>
</tr>
</tbody>
</table>

SD = Standard deviation
### Table 2. Baseline Predictors of Adverse Event

<table>
<thead>
<tr>
<th>Effect</th>
<th>Odds Ratio (95% CI)</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drug treatment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drug A</td>
<td>3.40 (2.37-4.89)</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>Drug B</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18-&lt;40</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>40-&lt;50</td>
<td>1.30 (0.60-2.83)</td>
<td>0.51</td>
</tr>
<tr>
<td>50-&lt;65</td>
<td>4.01 (2.08-7.72)</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>&gt;=65</td>
<td>13.00 (6.44-26.23)</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>Gender</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>0.64 (0.46-0.89)</td>
<td>0.0089</td>
</tr>
<tr>
<td>Baseline hypertension</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>1.18 (0.81-1.71)</td>
<td>0.4</td>
</tr>
<tr>
<td>Baseline coronary artery disease</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>1.43 (0.98-2.07)</td>
<td>0.062</td>
</tr>
<tr>
<td>Baseline diabetes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>1.87 (1.29-2.73)</td>
<td>0.0011</td>
</tr>
<tr>
<td>Baseline heart failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>2.07 (1.23-3.46)</td>
<td>0.0058</td>
</tr>
<tr>
<td>Baseline malignancy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td>2.17 (1.31-3.60)</td>
<td>0.0027</td>
</tr>
</tbody>
</table>

CI = Confidence interval

1 Odds Ratio < 1
The SAS data used as input to PROC REPORT for the above reports are shown below.

<table>
<thead>
<tr>
<th>Obs</th>
<th>patchar</th>
<th>stats1</th>
<th>stats2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Age years (mean (SD), median, range)</td>
<td>49.2 (12.8), 50.0, 18.0-72.0</td>
<td>49.9 (9.7), 51.0, 20.0-72.0</td>
</tr>
<tr>
<td>2</td>
<td>Age categorized (n, %)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>18-39</td>
<td>344 (25.3%)</td>
<td>262 (15.7%)</td>
</tr>
<tr>
<td>4</td>
<td>40-49</td>
<td>309 (22.7%)</td>
<td>513 (30.7%)</td>
</tr>
<tr>
<td>5</td>
<td>50-64</td>
<td>530 (39.0%)</td>
<td>812 (48.6%)</td>
</tr>
<tr>
<td>6</td>
<td>65+</td>
<td>177 (13.0%)</td>
<td>83 (5.0%)</td>
</tr>
<tr>
<td>7</td>
<td>Gender (n, %)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Male</td>
<td>575 (42.3%)</td>
<td>637 (38.1%)</td>
</tr>
<tr>
<td>9</td>
<td>Female</td>
<td>785 (57.7%)</td>
<td>1033 (61.9%)</td>
</tr>
<tr>
<td>10</td>
<td>Baseline comorbidities (n, %)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Hypertension</td>
<td>773 (56.8%)</td>
<td>1123 (67.2%)</td>
</tr>
<tr>
<td>12</td>
<td>Coronary artery disease</td>
<td>242 (17.8%)</td>
<td>245 (14.7%)</td>
</tr>
<tr>
<td>13</td>
<td>Diabetes</td>
<td>206 (15.1%)</td>
<td>362 (21.7%)</td>
</tr>
<tr>
<td>14</td>
<td>Heart failure</td>
<td>71 (5.2%)</td>
<td>64 (3.8%)</td>
</tr>
<tr>
<td>15</td>
<td>Malignancy</td>
<td>80 (5.9%)</td>
<td>76 (4.6%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>effect_name</th>
<th>orprt</th>
<th>pvalprt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Drug treatment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Drug A</td>
<td>3.40 (2.37-4.89)</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>3</td>
<td>Drug B</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Age</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>18-&lt;40</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>40-&lt;50</td>
<td>1.30 (0.60-2.83)</td>
<td>0.51</td>
</tr>
<tr>
<td>7</td>
<td>50-&lt;65</td>
<td>4.01 (2.08-7.72)</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>8</td>
<td>&gt;=65</td>
<td>13.00 (6.44-26.23)</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>9</td>
<td>Gender</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Male</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Female</td>
<td>0.64 (0.46-0.89)</td>
<td>0.0089</td>
</tr>
<tr>
<td>12</td>
<td>Baseline hypertension</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>No</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Yes</td>
<td>1.18 (0.81-1.71)</td>
<td>0.40</td>
</tr>
<tr>
<td>15</td>
<td>Baseline coronary artery disease</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>No</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Yes</td>
<td>1.43 (0.98-2.07)</td>
<td>0.062</td>
</tr>
<tr>
<td>18</td>
<td>Baseline diabetes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>No</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Yes</td>
<td>1.87 (1.29-2.73)</td>
<td>0.0011</td>
</tr>
<tr>
<td>21</td>
<td>Baseline heart failure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>No</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Yes</td>
<td>2.07 (1.23-3.46)</td>
<td>0.0058</td>
</tr>
<tr>
<td>24</td>
<td>Baseline malignancy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>No</td>
<td>Reference</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Yes</td>
<td>2.17 (1.31-3.60)</td>
<td>0.0027</td>
</tr>
</tbody>
</table>

These SAS datasets were created by report programs (not shown) where the input was the patient-level data and the output was the summary datasets shown above.
Two types of text formatting were added to these SAS data sets when they were created in the report programs. These types of text formatting will allow text indentation and within-cell linefeeds in the final Excel report.

**TEXT INDENTATION**
Both of these SAS datasets include character variables that are indented to the right (PATCHAR for Table 1 and EFFECT_NAME for Table 2). These indentations were created simply by concatenating the string 'A0A0A0A0'x at the beginning of the character variable.

```sas
patchar = catx('','A0A0A0A0', (put(&rowvar,&&&rowvar.fmt)));
```

No additional programming is needed in PROC REPORT to produce the character indentation.

**FORCED LINEFEED WITHIN A CELL**
The summary statistics for the continuous variable AGE in Table 1 include MEAN (SD), and MEDIAN, RANGE. The 5 character text string &\#10; will cause Excel to insert a forced within cell line feed between those items.

```sas
patchar = "&label"||" (mean (SD),&\#10;median, range)";

stats&colnum = trim(left(put(mean&colnum,8.1)) || '|' ||
trim(left(put(std&colnum,8.1)) || '|') ||
"&\#10;" ||
trim(left(put(median&colnum,8.1)) || ', ' ||
trim(left(put(min&colnum,8.1)) || '-' ||
trim(left(put(max&colnum,8.1)));
```

For the linefeed to work, an option PROTECTSPECIALCHARS = OFF is needed for those variables in the PROC REPORT CODE. The use of this option will be demonstrated with the PROC REPORT code.

**PART 1 – SETTING UP THE EXCEL TAGSET PROGRAM**
The code to set up the Excel Tagset, with the selected options used in the examples, is given below:

```sas
%inc "&pgmloc\EXP_template.sas";
%inc "&pgmloc\ExcelXP.sas";
odc listing close;

ods Tagsets.excelxp file="&rptloc\NESUG.xls" style=styles.exp
   options(fittopage = 'yes'
   embedded_titles = 'yes'
   embedded_footnotes = 'yes'
   zoom = '100'
   orientation='Landscape'
   row_repeat = 'header'
   Pages_FitHeight = '100'
   center_horizontal = 'yes'
   center_vertical = 'no'
   autofit_height = 'yes'
   print_header='&C&"Arial"&10CONFIDENTIAL'
   print_footer='&L&"Arial"&10Treatment with Drug A and DrugB&\#13Tables Prepared for NESUG&C&"Arial"&10Page &P&"Arial"&10&D');
```

The ExcelXP Tagset has many set-up options, many of which parallel Excel's page set-up options.
The options of EMBEDDED_TITLES and EMBEDDED_FOOTNOTES allow the SAS titles and footnotes to be included as part of each table.

The PRINT_HEADER and PRINT_FOOTER options define the Excel headers and footnotes that will be included on every page in the workbook. Excel headers and footnotes can have left, center, and right section, and can include automatic variables, such as the current date included in the right footnote section as shown in the example.

Once the ODS Excel Tagset has been defined and opened, calls can be made to the Tagset that define each individual sheet.

PART 2 – SHEET 1 AND PROC REPORT CODE FOR TABLE 1
Table 1 is a simple comparison of treatment group table. The PROC REPORT code to generate the Excel table is given below:

```
ods Tagsets.excelxp options(sheet_name="Tab 1 Baseline Char");
title1 "Table 1. Baseline Characteristics of Treatment Groups";
footnote1 "SD = Standard deviation";
%getcounts(dsn=patvars, countvar=tx_group, prefix=n);
proc report data=allprt_tab1
   nowindows split='*' missing
   style(header)={font_weight=bold
                  font_size=10pt
                  just=center
                  protectspecialchars=off}
   style(column)={font_size=10pt just=center}
;  
column patchar stats1 stats2 c;
define patchar / display "Characteristic \n" style={cellwidth=7cm};
define stats1 / display "Drug A*n = &n1"   style={cellwidth=3cm};
define stats2 / display "Drug B*n = &n2"   style={cellwidth=3cm};
define c    / noprint;
compute c;
  if index(patchar,"n, %") > 0 or index(patchar,"mean (SD)") > 0 then do;
    call define("patchar", "style", "style=[font_weight=bold just=left]");
    call define(_row_, "style", "style=[protectspecialchars=off]"));
  end;
  else if index(patchar,"#10;") > 0 or index(stats1,"#10;") > 0 then do;
    call define("patchar", "style", "style=[just=left]");
    call define(_row_, "style", "style=[protectspecialchars=off]"));
  end;
  else call define("patchar", "style", "style=[just=left]");
endcomp;
run;
```

The first line of the above code calls the Tagset and defines a new sheet. Formatting techniques shown in the above example include using PROC REPORT’s compute blocks to conditionally format individual cells, and the use of macro variables to include counts in each column header.
USING COMPUTE BLOCKS TO CONDITIONALLY FORMAT CELLS

In the table’s column with the header “Characteristics” I wanted the header to be centered, but the text in the cells to be left justified. I also wanted all non-indented data that gave the name of the variable described below to be in bold, but the text representing data values to be plain text.

To allow the use of a COMPUTE block in PROC REPORT to conditionally format individual cells or rows, I included a non-printed dummy variable C in the PROC REPORT code.

The CALL DEFINE statements in the COMPUTE C block conditionally format individual cells or table rows. To enable the forced linefeeds within an Excel table cell, style= [protectspecialchars=off] was included in the cell’s style. If this option was not added, the text &\#10; would appear in each cell, instead of a forced linefeed.

ADDING COUNTS TO COLUMN HEADERS

Table 1 has the total number of patients in each treatment group as part of the column headers. These patient counts are stored as macro variables and generated in the Tagset program by a call to the following macro:

```sas
%macro getcounts(dsn=, countvar=, prefix=n);
  %let dsid  = %sysfunc(open(&dsn,i));
  %let varnum = %sysfunc(varnum(&dsid,&countvar));
  %let fmtname = %sysfunc(varfmt(&dsid, &varnum));
  %let rc   = %sysfunc(close(&dsid));

  proc format library=library cntlout=__fmtvals0;
    run;
  data __fmtvals (keep = start);
    set __fmtvals0;
    fmtname = trim(left(fmtname))||'.';
    if fmtname = "&fmtname"
      run;

  proc datasets nolist;
    delete __fmtvals0;
  run;

  data _null_; set __fmtvals;
    call symputx("&prefix"||trim(left(start)),0,'g');
  run;

  proc freq data=&dsn noprint;
    where &countvar > .;
    tables &countvar / out=&countvar;
  run;

  data _null_; set &countvar;
    call symputx("&prefix"||trim(left(put(&countvar,8.))),count,'g');
  run;

  %global &prefix.n;
  proc sql noprint;
    select sum(count) into :&prefix.n
      from &countvar;
    quit;

  %let &prefix.n = %left(&&prefix.n);
  %put &;&&prefix.n;
%mend getcounts;
```
PART 3 – SHEET 2 AND PROC REPORT CODE FOR TABLE 2

Table 2 displays the odds ratios from a logistic regression. The PROC REPORT code to generate the Excel table is given below:

```sas
ods escapechar = "^";
ods Tagsets.excelxp options(sheet_name="Tab 2 Logistic Reg");
title1 "Table 2. Baseline Predictors of Adverse Event";
footnote1 "CI = Confidence interval";
footnote2 "^{super 1} Odds Ratio < 1";
proc report data=allprt_tab2
nowindows split='*' missing
style(header)={font_weight=bold font_size=10pt just=center}
style(column)={font_size=10pt}
;
column effect_name orprt pvalprt c;
define effect_name / display "Effect" style={just=left cellwidth=8cm};
define orprt / display "Odds Ratio (95% CI)" style={just=center cellwidth=5cm protectspecialchars=off};
define pvalprt / display "P-Value" style={just=center cellwidth=4cm};
define c / noprint;
compute orprt;
  if substr(orprt,1,1) = '0' then orprt = catx('',orprt,"^{super 1}");
endcomp;
compute c;
  if orprt = '' then call define(_row_, "style",
      "style=[background=CXDCDCDC font_weight=bold]"));
endcomp;
run
```

This PROC REPORT also uses a dummy variable C that is used in a COMPUTE C block. This block will shade an entire row and make the text bold for the header rows with no values for odds ratios.

USING ODS ESCAPECHAR TO ADD INLINE FORMATTING—CONDITIONAL SUPERSCRIPTS

The above example of PROC REPORT uses ODS ESCAPECHAR and a COMPUTE block to conditionally insert a superscript “1” next to any odds ratio that is less than 1.0. This inline formatting works both in the SAS titles and footnotes, and in the contents of the cells. Inline formatting can be used for applications other than superscripting text and opens up a whole range of formatting possibilities reports generated by the ExcelXP Tagset.

PART 4 – ENDING THE PROGRAM

The final two lines of the program to generate the Excel workbook with the two sheets are given below:

```sas
ods Tagsets.excelxp close;
ods listing;
```
CONCLUSION
The ExcelXP Tagset is an easy-to-use program that enables SAS to generate multi-sheet Excel workbooks. The advantages of storing multiple tables and reports in a single Excel file are organization of tables, ease of use, and consistent version control of tables. Formatting tricks and techniques can be used to make the printable tables generated by the ExcelXP tag as nicely formatted as individual .RTF files.

REFERENCES:

EXCEL XP TAGSET


PROC TEMPLATE

INLINE FORMATTING

ACKNOWLEDGMENTS
SAS and other SAS Institute, Inc. products 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 registered trademarks or trademarks of their respective companies.

I would like to thank Eric Gebhart of the SAS Institute and Jack Hamilton for their helpful answers to my questions about using the ExcelXP Tagset.

CONTACT INFORMATION
Please contact the author with any comments or questions:

Kathy H. Fraeman
United BioSource Corporation
7101 Wisconsin Avenue, Suite 600
Bethesda, MD 20832
(240) 235-2525 voice
(301) 654-9864 fax
kathy.fraeman@unitedbiosource.com
ENTIRE PROGRAM

/*******************************************************************************/
/* Program: T_Excel_Tables.sas */
/* Developer: Kathy Fraeman */
/* Date: 18June2009 */
/* Platform: SAS 9.1.3 */
/* Drug/Protocol: NESUG Paper */
/* Description: Creates Excel spreadsheet with two tables */
/* Macros Used: %GETCOUNTS */
/* Input: Data sets that will be used to generate tables */
/* ALLPRT_TAB1 */
/* ALLPRT_TAB2 */
/* Patient-level data file for counts of patients in TX groups */
/* PATVARS */
/* Output: NESUG.XLS */
/* Modification History: */
/*******************************************************************************/

%inc "setup.sas" / source2;
%inc '"pgmloc\Table1.sas";
%inc '"pgmloc\Table2.sas";
%inc '"pgmloc\EXP_template.sas";
%inc '"pgmloc\ExcelXP.sas";
ods listing close;
ods tagsets.excelxp file="&rptloc\NESUG.xls" style=styles.exp
options(fittopage = 'yes'
       embedded_titles = 'yes'
       embedded_footnotes = 'yes'
       zoom = '100'
       orientation='Landscape'
       )
row_repeat = 'header'
Pages_FitHeight = '100'
center_horizontal = 'yes'
center_vertical = 'no'
autfit_height = 'yes'
print_header= '&C&"Arial"&10CONFIDENTIAL'
print_footer= '&L&"Arial"&10Treatment with Drug A
and Drug B
&13Tables Prepared for NESUG&\&P&"Arial"&10Page
&D');

/******************************************************************************/
/* SHEET 1 -- Table 1 (Baseline Characteristics of Treatment Groups) */
/******************************************************************************/
ods tagsets.excelxp options(sheet_name="Tab 1 Baseline Char");
title1 "Table 1. Baseline Characteristics of Treatment Groups";
footnote1 "SD = Standard deviation";
%getcounts(dsn=patvars, countvar=tx_group, prefix=n);
proc report data=allprt_tab1
    nowindows split='*' missing
    style(header)={font_weight=bold font_size=10pt
                   just=center protectspecialchars=off}
    style(column)={font_size=10pt just=center}
    ;
column patchar stats1 stats2 c;
define patchar  / display "Characteristic 
"   style={cellwidth=7cm};
define stats1  / display "Drug A*n = &n1"      style={cellwidth=3cm};
define stats2  / display "Drug B*n = &n2"      style={cellwidth=3cm};
define c     / noprint;
/*----------------------------------------------------*/
/* Adjust the formatting of specific rows and cells */
/*----------------------------------------------------*/
compute c;
    if index(patchar,"n," %") > 0 or index(patchar,"mean (SD)") > 0 then do;
        call define("patchar", "style", "style=[font_weight:bold just=left]");
        call define(_row_, "style", "style=[protectspecialchars=off]");
    end;
    else if index(patchar,"#10;") > 0 or index(stats1,"#10;") > 0 then do;
        call define("patchar", "style", "style=[just=left]");
        call define(_row_, "style", "style=[protectspecialchars=off]");
    end;
    else call define("patchar", "style", "style=[just=left]");
endcomp;
run;

/******************************************************************************/
/* SHEET 2 -- Table 2 (Baseline Predictors of Adverse Event) */
/******************************************************************************/
ods escapechar = "^";
ods tagsets.excelxp options(sheet_name="Tab 2 Logistic Reg");
title1 "Table 2. Baseline Predictors of Adverse Event";
footnote1 "CI = Confidence interval";
footnote2 "^\{super 1\} Odds Ratio < 1";

proc report data=allprt_tab2
nowindows split='*' missing
style(header)={font_weight=bold font_size=10pt just=center}
style(column)={font_size=10pt}
;
column effect_name orprt pvalprt c;
define effect_name  / display  "Effect" style={just=left cellwidth=8cm};
define orprt     / display  "Odds Ratio (95% CI)"
    style={just=center cellwidth=5cm protectspecialchars=off};
define pvalprt    / display  "P-Value" style={just=center cellwidth=4cm};
define c       / noprint;
/*---------------------------------------------------------------*/
/* Put a superscript 1 on every odds ratio that is less than 1 */
/*---------------------------------------------------------------*/
compute orprt;
  if substr(orprt,1,1) = '0' then orprt = catx('',orprt,"^{super 1}");
endcomp;

/*---------------------------------------------------------------*/
/* Adjust the formatting of specific rows and cells */
/*---------------------------------------------------------------*/
compute c;
  if orprt = '' then
    call define(_row_, "style",
      "style=[background=CXDCDCDC font_weight=bold]");
endcomp;
run;

/** End of program, close the tagset */
/** End of program, close the tagset */
ods tagsets.excelxp close;
ods listing;