ABSTRACT
PROC REPORT is a powerful reporting procedure, whose output can be “practically perfect” when you add ODS STYLE= overrides to your PROC REPORT code. This hands-on workshop will feature several PROC REPORT programs that produce default output for ODS HTML, RTF and PDF destinations. Workshop attendees will learn how to modify the defaults to change elements of PROC REPORT output, such as HEADER cells, DATA cells, SUMMARY cells and LINE output using ODS STYLE= overrides. In addition, attendees will learn how to apply conditional formatting at the column or cell level and at the row level using PROC FORMAT techniques and CALL DEFINE techniques. Other topics include: table attributes that control interior table lines and table borders, use of logos in output and producing “Page x of y” page numbering. The rest of this document is divided into two parts: the slides used in the Hands-On Workshop presentation and the attendee worksheet.

CONTACT INFORMATION
Your comments and questions are valued and encouraged. Contact the author:

Cynthia L. Zender
SAS Institute, Inc.
Work Phone: 919-531-9012 (Mountain Time Zone)
E-mail: Cynthia.Zender@sas.com
Today's Agenda

- Modify PROC REPORT defaults to change elements of PROC REPORT output
  - HEADER cells
  - DATA cells
  - SUMMARY cells
  - LINE output using ODS STYLE= overrides
- Apply conditional formatting at the column, cell or row level
- Modify table attributes that control interior table lines and table borders
- Use logos or images in output
- Produce "Page x of y" page numbering for RTF and PDF output.
Chapter 6: Using Options with ODS

1) Basic PROC REPORT, ODS Style= and ESCAPECHAR

2) Style= Statement Level Overrides

3) Overriding Style for LINES and CALL DEFINE

4) Inserting Images and Performing Traffic Lighting

Proc Report Review – Pen and Paper Activity

Match the statement with its function in PROC REPORT:

A: DEFINE
B: COLUMN
C: DISPLAY
D: COMPUTE/ENDCOMP
E: ACROSS

1: defines a block of code to be executed
2: specifies that variable values should be listed without ordering or grouping
3: specifies that a column is created for every variable value
4: specifies how to use a report item
5: specifies which report items are used on the report
Review Material for PROC REPORT Syntax

The following material can be used as reference throughout the Workshop. The correct answer to the review quiz can be found on Slide # 20, Page 10.

About the REPORT Procedure

PROC REPORT produces
- detail reports in data storage order
- detail reports in sorted order from unsorted data
- summary reports based on grouping variables
- summary rows (group subtotals and overall report totals) for both detail and summary reports
- statistics for analysis variables
- custom text.
Using the Report Procedure

General form of the REPORT procedure:

```
PROC REPORT DATA=library.filename <option(s)>;
  COLUMN column-specification(s);
  DEFINE report-item / <usage> <attribute(s)> <option(s)>;
  COMPUTE report-item </ type-specification>;
  . . . select SAS language elements . . .
  ENDCOMP;
  BREAK location break-variable </ option(s)>;
  RBREAK location </ option(s)>;
RUN;
```

Reviewing the PROC REPORT Statement

```
PROC REPORT DATA=library.filename NOWD <option(s)>;
```

- The NOWINDOWS | NOWD option specifies how to generate the report.
- **The NOWD option is important if you are using the Output Delivery System.**
- With the NOWD option, the report is sent to the LISTING window and open ODS destinations in non-interactive mode.
- Without the NOWD option, the report is generated in the REPORT window, an interactive environment in which to develop a report.
Reviewing the PROC REPORT Statement

Selected PROC REPORT statement options:

<table>
<thead>
<tr>
<th>To Do This</th>
<th>Use This</th>
</tr>
</thead>
<tbody>
<tr>
<td>Underline all column headers and the spaces between them.</td>
<td>HEADLINE*</td>
</tr>
<tr>
<td>Write a blank line beneath all column headers.</td>
<td>HEADSKIP*</td>
</tr>
<tr>
<td>Use formatting characters to add line-drawing characters to the report.</td>
<td>BOX*</td>
</tr>
<tr>
<td>Specify the number of panels on each page of the report.</td>
<td>PANELS=*</td>
</tr>
<tr>
<td>Specify the split character. The default split character is a forward slash (/).</td>
<td>SPLIT=</td>
</tr>
<tr>
<td>Specify the length of a line of the report.</td>
<td>LS=*</td>
</tr>
<tr>
<td>Specify the number of lines in a page of the report</td>
<td>PS=*</td>
</tr>
</tbody>
</table>

* Listing destination only

Reviewing the COLUMN Statement

COLUMN column-specification(s);

column-specification(s) is one or more of the following:

- report items such as the name of a data set variable, computed variable, or a statistic
- a comma operator to nest or stack columns
- parentheses to group columns
- headers to span columns
- an alias for a column.
COLUMN Statement Examples

Some COLUMN statement examples:
- list report items such as the name of a data set variable, computed variable, or a statistic

```
column Region District N Sales;
```
- list an alias for a column.

```
column Region Sales Sales=SLmedian;
```

Comma and Parentheses Examples

The COLUMN statement also supports use of:
- a comma operator to nest or stack columns

```
column Product Sales,N Sales,Mean;
```
- parentheses and a comma operator to nest or stack columns

```
column Region Product,(Sales N);
```
- parentheses and headers to span columns.

```
column Region ('-Cosmetic Division-' Product Sales);
```
Specifying a DEFINE Statement

The DEFINE statement specifies HOW to use a report item:

```
DEFINE report-item / <usage> <attribute(s)> <option(s)> ;
```

where
- `report-item` is a data set variable, a statistic, column alias, or a computed column that has been listed in the COLUMN statement
- `usage` is DISPLAY, ORDER, GROUP, ACROSS, ANALYSIS, or COMPUTED
- `attributes` define the format, width, spacing, statistic for analysis variable, or sort sequence of the column
- `options` specify the label and justification.

Specifying DEFINE Statement Usages

<table>
<thead>
<tr>
<th>To Do This</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present the data in the order in which it is stored.</td>
<td>DISPLAY</td>
</tr>
<tr>
<td>Present the data in the default order or the order specified by the ORDER= option.</td>
<td>ORDER</td>
</tr>
<tr>
<td>Summarize analysis variables based on the group variables and consolidate into one row all observations that have the same value for the group variable.</td>
<td>GROUP</td>
</tr>
<tr>
<td>Use the variable values as the column headers in the report.</td>
<td>ACROSS</td>
</tr>
<tr>
<td>Summarize these variables when there is a GROUP variable and/or a BREAK or RBREAK statement.</td>
<td>ANALYSIS</td>
</tr>
<tr>
<td>Compute the variable value from statements within a compute block.</td>
<td>COMPUTED</td>
</tr>
</tbody>
</table>
**Identifying DEFINE Statement Options**

DEFINE statement options alter the report appearance for a particular column.

Selected DEFINE statement options:

<table>
<thead>
<tr>
<th>To Modify</th>
<th>Use This Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Justification</td>
<td>LEFT</td>
</tr>
<tr>
<td>Column headings</td>
<td>&quot;Variable Label&quot;</td>
</tr>
<tr>
<td>Order of data values</td>
<td>ORDER=</td>
</tr>
<tr>
<td>Printing of column</td>
<td>NOPRINT</td>
</tr>
<tr>
<td>Number of blanks to the left of a column</td>
<td>SPACING= *</td>
</tr>
<tr>
<td>Column width</td>
<td>WIDTH= *</td>
</tr>
<tr>
<td>Consider missing values as valid values for the item</td>
<td>MISSING</td>
</tr>
<tr>
<td>Format data</td>
<td>FORMAT=</td>
</tr>
</tbody>
</table>

* The order of the options after the slash does not matter.

* Listing destination only

**Creating Column Summaries**

To create a summary for the entire report, use the RBREAK statement.

```
RBREAK BEFORE | AFTER / <options>;
```

To create a summary for groups of data, use the BREAK statement.

```
BREAK BEFORE | AFTER break-variable / <options>;
```

The `break-variable` is a group or order variable. When the value of the `break-variable` changes, PROC REPORT creates a summary line.
Using RBREAK and BREAK Options

<table>
<thead>
<tr>
<th>To Do This</th>
<th>Use This Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double-underline each value.</td>
<td>DUL*</td>
</tr>
<tr>
<td>Double-overline each value.</td>
<td>DOL*</td>
</tr>
<tr>
<td>Overline each value.</td>
<td>OL*</td>
</tr>
<tr>
<td>Underline each value.</td>
<td>UL*</td>
</tr>
<tr>
<td>Start a new page after the last break line.</td>
<td>PAGE</td>
</tr>
<tr>
<td>Write a blank line for the last break line.</td>
<td>SKIP*</td>
</tr>
<tr>
<td>Write a summary line in each group of break lines.</td>
<td>SUMMARIZE</td>
</tr>
<tr>
<td>Suppress the printing of the value of the break</td>
<td>SUPPRESS</td>
</tr>
<tr>
<td>variable in the summary line and of any underlining</td>
<td></td>
</tr>
<tr>
<td>or overlining in the break lines in the column that</td>
<td></td>
</tr>
<tr>
<td>contains the break variable.</td>
<td></td>
</tr>
</tbody>
</table>

* Listing destination only

Customizing Text for Breaks

```
COMPUTE BEFORE | AFTER <variable> <_PAGE_> /;
    executable statements
ENDCOMP;
```

- When a variable is not specified, the compute block is executed either before or after the report is created.
- The _PAGE_ option specifies that the compute block is executed either before or after the page break. There must be a BREAK statement with the PAGE option in order to use the _PAGE_ option in a COMPUTE statement.
- The executable statements can calculate variables, use IF...THEN logic, write out custom text, or use most DATA step statements.
### Using a COMPUTE Statement

<table>
<thead>
<tr>
<th>To execute the compute block</th>
<th>Use This Target</th>
<th>Use This Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>at a break immediately after the last row of a set of rows. If there is a default summary on that variable, immediately after the creation of the preliminary summary line.</td>
<td>Report-item</td>
<td>AFTER</td>
</tr>
<tr>
<td>near the bottom of each page, immediately before any footnotes. (LISTING only)</td>
<td><em>PAGE</em>*</td>
<td>AFTER</td>
</tr>
<tr>
<td>at the end of the report.</td>
<td>No target</td>
<td>AFTER</td>
</tr>
<tr>
<td>at a break immediately before the last row of a set of rows. If there is a default summary on that variable, immediately after the creation of the preliminary summary line.</td>
<td>Report-item</td>
<td>BEFORE</td>
</tr>
<tr>
<td>immediately after printing any titles. (LISTING only)</td>
<td><em>PAGE</em>*</td>
<td>BEFORE</td>
</tr>
<tr>
<td>at the end of the report.</td>
<td>No target</td>
<td>BEFORE</td>
</tr>
</tbody>
</table>

If the target is _PAGE_, you can specify a justification option, LEFT, RIGHT, or CENTER.

### Proc Report Review – Solutions

The correct answers are:

A/4: DEFINE - specifies HOW to use a report item

B/5: COLUMN - specifies WHICH report items are used on the report

C/2: DISPLAY - specifies that variable values should be listed without ordering or grouping

D/1: COMPUTE/ENDCOMP - defines a block of code to be executed

E/3: ACROSS - specifies that a column is created for every variable value
Your Turn: Submit Demo Programs

- Turn to your worksheet and run the programs outlined on #1 and #2.

For more information about PROC REPORT options that only work in the LISTING destination, see these web sites:
- http://support.sas.com/kb/2/549.html

Summary of Demo Results

ODS destinations do not all use PROC REPORT Options the same way.

<table>
<thead>
<tr>
<th>Statement or Option</th>
<th>LISTING</th>
<th>HTML</th>
<th>PDF</th>
<th>RTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEADLINE/HEADSKIP</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>WIDTH</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>SPACING</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>DOL/DUL</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CENTER</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SKIP</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>LS</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Repeat Character in Spanning Header</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>NUMBER and PAGENO=1</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
### Summary of Global System Options

<table>
<thead>
<tr>
<th>Option</th>
<th>HTML</th>
<th>PDF</th>
<th>RTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>NODATE</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NUMBER</td>
<td>NONUMBER</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PAGENO=</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>ORIENTATION=</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>CENTER</td>
<td>NOCENTER</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TOPMARGIN=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>BOTTOMMARGIN=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LEFTMARGIN=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RIGHTMARGIN=</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

### STYLE= Option

The `STYLE=` option specifies the style template to use. Style templates describe how to display the presentation aspects of the output.

To get a listing of the style templates supplied by SAS, submit the following code:

```sas
proc template;
   list styles / store=sashelp.tmplmst;
run;
```
The ODS ESCAPECHAR statement sets an escape character that is used to introduce special sequences for in-line formatting.

```
ods escapechar = 'escape-character';
```

Choose a unique character that does not occur in your code or output.

Examples of ODS ESCAPECHAR statement:

```
ods escapechar = '!';
ods escapechar = '*';
ods escapechar = '~';
ods escapechar = '^';
ods escapechar = '#';
```

The ODS ESCAPECHAR statement can be omitted entirely if you use one of the following as the escape character:
- '\03'x
- (*ESC*)
## ODS ESCAPECHAR

<table>
<thead>
<tr>
<th>Categories of Special Sequences</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Performs a function to get specialized formatting effects within a cell.</td>
</tr>
<tr>
<td>Style</td>
<td>Modifies the style of the current cell based on the specified style attributes.</td>
</tr>
<tr>
<td>Raw Text</td>
<td>Inserts raw text into the current cell.</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>Inserts special codes, line breaks, line wraps, and non-breaking space.</td>
</tr>
</tbody>
</table>

The escape character followed by a function creates a specialized formatting effect within a cell.

### Function

<table>
<thead>
<tr>
<th></th>
<th>HTML</th>
<th>PDF</th>
<th>RTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>{super text}</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>{sub text}</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>{dagger}</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>{thispage}</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>{lastpage}</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>{pageof}</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

The escape character followed by a function creates a specialized formatting effect within a cell.
Your Turn: Different Styles and Page X of Y

Turn to your worksheet and do items 3 and 4:

• Submit the Demo01e_different_styles.sas program to use different STYLE= options.

• Submit the Demo01f_page_x_of_y.sas program to use the ODS ESCAPECHAR option as shown below:

```sas
<after ODS invocation statements>
ods escapechar='~';
footnote j=r 'Page ~{pageof}';
footnote2 j=c 'Page ~{thispage} of ~{lastpage}';
```

Practically Perfect Tip #1

If you are creating RTF output, you can use ODS ESCAPECHAR and Style attributes to insert RTF control strings into your result output.

```sas
footnote j=l font='Arial' h=10.1pt
"^S={protectspecialchars=off "
"pretext='\brdrt\brdrs\brdrw1 '}
"\~";
```

Tip01.sas
References

http://www.sas.com/rnd/base/topics/templateFAQ/reoperation.html
http://www.w3schools.com/css/css_reference.asp#list
Chapter 6: Using Options with ODS

1) Basic PROC REPORT, ODS Style= and ESCAPECHAR

2) Style= Statement Level Overrides

3) Overriding Style for LINES and CALL DEFINE

4) Inserting Images and Performing Traffic Lighting

The STYLE= Statement Level Override

General form of the STYLE= statement level override option:

```
STYLE(component)=
{attribute-1=value-1 … 
attribute-n=value-n}
```

With PROC REPORT, you provide the component name for the STYLE= override option directly in selected PROC REPORT statements.

The STYLE= option is placed after a slash (/) in all the statements except the PROC REPORT statement.
### Using PROC REPORT Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Affects This Part of Report</th>
<th>Use in This Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>Report as a whole and table structure attributes</td>
<td>PROC REPORT</td>
</tr>
<tr>
<td>Header (HDR)</td>
<td>Column header cells</td>
<td>PROC REPORT DEFINE</td>
</tr>
<tr>
<td>Column</td>
<td>Data cells</td>
<td>PROC REPORT DEFINE</td>
</tr>
<tr>
<td>Summary</td>
<td>Summary lines generated by BREAK or RBREAK statements</td>
<td>PROC REPORT RBREAK</td>
</tr>
<tr>
<td>Lines</td>
<td>Lines generated by LINE statements</td>
<td>PROC REPORT COMPUTE</td>
</tr>
<tr>
<td>Calldef</td>
<td>Cells identified by a CALL DEFINE statement</td>
<td>PROC REPORT</td>
</tr>
</tbody>
</table>

### Style Attributes and Color Values

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Sample Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOREGROUND BACKGROUND</td>
<td>white, red, yellow, black, cxFFFFFF, cxFF0000, cxFFFF00, cx000000</td>
</tr>
<tr>
<td></td>
<td>h000FF00, h07880FF, h0B480FF, h0000000</td>
</tr>
<tr>
<td>BORDERCOLOR</td>
<td>black, cx000000, h0000000</td>
</tr>
</tbody>
</table>
Visual Guide to Report Components

**Header**

<table>
<thead>
<tr>
<th>Region</th>
<th>Subsidiary</th>
<th>Product</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asian Pacific</td>
<td>Cosmetic Products</td>
<td>HeadGel Lotion</td>
<td>1,095,962</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WinkAway Cream</td>
<td>529,495</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nose Spray</td>
<td>998,180</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SleepTight Tablets</td>
<td>223,870</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,965,586</td>
<td></td>
</tr>
</tbody>
</table>

**Summary**

<table>
<thead>
<tr>
<th>Region</th>
<th>Subsidiary</th>
<th>Product</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eastern Europe</td>
<td>Cosmetic Products</td>
<td>HeadGel Lotion</td>
<td>1,000,007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WinkAway Cream</td>
<td>865,769</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nose Spray</td>
<td>294,815</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SleepTight Tablets</td>
<td>377,529</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3,000,070</td>
<td></td>
</tr>
</tbody>
</table>

**Demo02.sas**

Font Related Attributes and Sample Values

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Sample Values</th>
</tr>
</thead>
</table>
| FONT_FACE     | "Times New Roman"  
                Arial  
                Helvetica  
                'Courier New'  
                "Arial, Helvetica, Helv" |
| FONT_SIZE     | 5, 10 pt, 1 cm, 0.25 in                                                   |
| FONT_STYLE    | italic, roman                                                             |
| FONT_WEIGHT   | medium, bold, light                                                        |
| FONT_WIDTH    | compressed, narrow, wide                                                   |
Alternate Font Attribute

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Sample Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>FONT</td>
<td>('Arial, Helvetica, Helv', 2)</td>
</tr>
<tr>
<td></td>
<td>(&quot;Times, Times New Roman&quot;, 3 cm, bold)</td>
</tr>
<tr>
<td></td>
<td>(Verdana, 4, bold italic)</td>
</tr>
<tr>
<td></td>
<td>('Courier New, Courier', 2, italic)</td>
</tr>
</tbody>
</table>

Your Turn: Using STYLE= Option

Turn to your worksheet and submit the Demo02.sas program (Item #5). Then review the code and answer the following questions:

a) What style attribute is set to CYAN?
b) What style attribute is set to RED?
### Other Table/Cell Style Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Sample Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>CELLWIDTH</td>
<td>150, 2 in, 5 cm, 20%</td>
</tr>
<tr>
<td>CELLHEIGHT</td>
<td></td>
</tr>
<tr>
<td>CELLSPACING</td>
<td>0, 7, .1 in, 1 cm, em, ex, pt</td>
</tr>
<tr>
<td>CELLPADDING</td>
<td></td>
</tr>
<tr>
<td>JUST</td>
<td>left, dec, center, right</td>
</tr>
<tr>
<td></td>
<td>l, d, c, r</td>
</tr>
<tr>
<td>VJUST</td>
<td>top, middle, bottom</td>
</tr>
<tr>
<td></td>
<td>t, m, b</td>
</tr>
<tr>
<td>URL</td>
<td>'www.sas.com'</td>
</tr>
<tr>
<td></td>
<td>&quot;<a href="http://support.sas.com">http://support.sas.com</a>&quot;</td>
</tr>
</tbody>
</table>

### Other Style Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Sample Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPUTWIDTH</td>
<td>150, 3 in, 20 cm, 80%</td>
</tr>
<tr>
<td>ASIS</td>
<td></td>
</tr>
<tr>
<td>PROTECTSPECIALCHARS</td>
<td></td>
</tr>
<tr>
<td>NOBREAKSPACE</td>
<td></td>
</tr>
<tr>
<td>INDENT</td>
<td>n ( \text{(number of units to indent)} )</td>
</tr>
</tbody>
</table>
### Table Structure, Interior Lines and Exterior Frame

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Sample Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>CELLSPLACING</td>
<td>0, 7, .1 in, 1 cm, em, ex, pt</td>
</tr>
<tr>
<td>BORDERWIDTH</td>
<td></td>
</tr>
<tr>
<td>RULES</td>
<td>ALL, GROUPS, NONE, COLS, ROWS</td>
</tr>
<tr>
<td>FRAME</td>
<td>ABOVE, BELOW, BOX, HSIDES, VSIDES, LHS, RHS, VOID</td>
</tr>
</tbody>
</table>

#### RULES Style Attribute

The RULES style attribute specifies the types of rules, or interior table lines, to use in a table.

<table>
<thead>
<tr>
<th>RULES Value</th>
<th>Creates interior table lines:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>between all rows and columns</td>
</tr>
<tr>
<td>GROUPS</td>
<td>between the table header and the table and between the table and the table footer, if there is one</td>
</tr>
<tr>
<td>NONE</td>
<td>no rules anywhere</td>
</tr>
<tr>
<td>COLS</td>
<td>between all columns</td>
</tr>
<tr>
<td>ROWS</td>
<td>between all rows</td>
</tr>
</tbody>
</table>
FRAME Style Attributes

FRAME specifies the type of frame to use on a table.

<table>
<thead>
<tr>
<th>FRAME Value</th>
<th>Creates this kind of frame around the table</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABOVE</td>
<td>a border at the top</td>
</tr>
<tr>
<td>BELOW</td>
<td>a border at the bottom</td>
</tr>
<tr>
<td>BOX</td>
<td>borders at the top, bottom, and both sides</td>
</tr>
<tr>
<td>HSIDES</td>
<td>borders at the top and bottom</td>
</tr>
<tr>
<td>VSIDES</td>
<td>borders at the left and right sides</td>
</tr>
<tr>
<td>LHS</td>
<td>a border at the left side</td>
</tr>
<tr>
<td>RHS</td>
<td>a border at the right side</td>
</tr>
<tr>
<td>VOID</td>
<td>no borders</td>
</tr>
</tbody>
</table>

Practically Perfect Tip #2

There is an interaction between CELLSPACING, BORDERWIDTH and the RULES and FRAMES style attributes.

```html
background=cyan
cellspacing=0
bordercolor=red
borderwidth=0
rules=rows frame=box
```

CELLSPACING=0 prevents the CYAN table background from showing around each table cell and BORDERWIDTH=0 prevents the RED border color from being used for the interior table lines.
Practically Perfect Tip #3

When dealing with very wide tables:

- HTML – tables can be as wide as required.
- RTF & PDF – table width is controlled by the viewer or rendering software. Use these techniques to make wide tables fit:
  - Set ORIENTATION option to LANDSCAPE in OPTIONS statement.
  - Change FONT_SIZE attribute.
  - Change CELLPADDING style attribute.
Chapter 6: Using Options with ODS

1) Basic PROC REPORT, ODS Style= and ESCAPECHAR

2) Style= Statement Level Overrides

3) Overriding Style for LINES and CALL DEFINE

4) Inserting Images and Performing Traffic Lighting

Other Methods to Change Style Attributes

<table>
<thead>
<tr>
<th>Region</th>
<th>Subsidiary</th>
<th>Product</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific</td>
<td>OTC Remedies</td>
<td>Dizzi Nose Sprays</td>
<td>196,196</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StepTight Tablets</td>
<td>223,790</td>
</tr>
<tr>
<td>Asia Pacific</td>
<td>OTC Remedies</td>
<td>Dizzi Nose Sprays</td>
<td>949,414</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StepTight Tablets</td>
<td>377,029</td>
</tr>
<tr>
<td>Eastern Europe</td>
<td>OTC Remedies</td>
<td>Dizzi Nose Sprays</td>
<td>972,340</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StepTight Tablets</td>
<td>744,254</td>
</tr>
<tr>
<td>North America</td>
<td>OTC Remedies</td>
<td>Dizzi Nose Sprays</td>
<td>1,236,136</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StepTight Tablets</td>
<td>774,254</td>
</tr>
<tr>
<td>North America</td>
<td>OTC Remedies</td>
<td>Dizzi Nose Sprays</td>
<td>2,019,470</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StepTight Tablets</td>
<td>980,317</td>
</tr>
<tr>
<td>South America</td>
<td>OTC Remedies</td>
<td>Dizzi Nose Sprays</td>
<td>393,050</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StepTight Tablets</td>
<td>317,681</td>
</tr>
<tr>
<td>South America</td>
<td>OTC Remedies</td>
<td>Dizzi Nose Sprays</td>
<td>895,283</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StepTight Tablets</td>
<td>217,681</td>
</tr>
<tr>
<td>Western Europe</td>
<td>OTC Remedies</td>
<td>Dizzi Nose Sprays</td>
<td>1,368,682</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StepTight Tablets</td>
<td>560,898</td>
</tr>
</tbody>
</table>

Grand Total: $5,650,818
Changing **STYLE(LINES)**

The LINES component changes the style of a line written in a COMPUTE block:

```plaintext
proc report data=salesdata nowd
  style(report)=
    {background=white cellspacing=0
     bordercolor=black borderwidth=2
     rules=rows frame=box}
  style(header)=
    {background=purple foreground=white}
  style(column)=
    {background=white}
  style(lines)=
    {font_weight=bold font_size=14pt
     just=r vjust=b foreground=black}
  style(summary)=
    {background=purple foreground=white};
```

CALL DEFINE

Inside a COMPUTE block, you can use the CALL DEFINE statement to change STYLE attributes:

```plaintext
compute after Region;
  if upcase(_break_) = 'REGION' and
    index(Region,'Europe') gt 0 then
    call define(_ROW_, 'style',
      {background=green foreground=white});
endcomp;
```
CALL DEFINE Syntax

A CALL DEFINE statement sets the value of an attribute for a particular column or row in the current row.

```
CALL DEFINE (column-id, attribute-name, value);
```

The CALL DEFINE statement has three arguments.

### Examples

<table>
<thead>
<tr>
<th>A COLUMN-ID can be</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>a character literal (in quotes)</td>
<td>'EmpID' 'Salary.sum'</td>
</tr>
<tr>
<td>a name of the form <em>Cn</em> (in quotes), where n is the column number</td>
<td>'<em>C1</em>' '<em>C5</em>'</td>
</tr>
<tr>
<td>the automatic variable <em>COL</em></td>
<td><em>COL</em></td>
</tr>
<tr>
<td>This variable identifies the column to which the compute block is attached.</td>
<td></td>
</tr>
<tr>
<td>the automatic variable <em>ROW</em></td>
<td><em>ROW</em></td>
</tr>
<tr>
<td>This variable identifies the row to which the compute block is attached.</td>
<td></td>
</tr>
</tbody>
</table>
### CALL DEFINE Syntax

#### Possible attribute-name values

<table>
<thead>
<tr>
<th>attribute-name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'STYLE'</td>
<td>specifies the style element</td>
</tr>
<tr>
<td>'URL'</td>
<td>makes the contents of each cell of the column a link to the specified Uniform Resource Locator (URL)</td>
</tr>
<tr>
<td>'FORMAT'</td>
<td>specifies a format for the column</td>
</tr>
</tbody>
</table>

#### Example values

<table>
<thead>
<tr>
<th>attribute-name</th>
<th>Example values</th>
</tr>
</thead>
<tbody>
<tr>
<td>'STYLE'</td>
<td>'STYLE={background=white foreground=red}'</td>
</tr>
<tr>
<td></td>
<td>'STYLE={foreground=xFF3300 font_weight=bold}'</td>
</tr>
<tr>
<td>'URL'</td>
<td>'MyHTMLPage.html'</td>
</tr>
<tr>
<td></td>
<td>MYURL (where MYURL is a variable equal to a URL)</td>
</tr>
<tr>
<td>'FORMAT'</td>
<td>'dollar12.2'</td>
</tr>
<tr>
<td></td>
<td>'myformat.'</td>
</tr>
</tbody>
</table>
Another CALL DEFINE Example

<table>
<thead>
<tr>
<th>Region</th>
<th>Subsidiary</th>
<th>Product</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asian Pacific</td>
<td>Cosmetic Products</td>
<td>HairGro Lotion</td>
<td>1,035,952</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WrinkAway Creme</td>
<td>529,345</td>
</tr>
<tr>
<td>Asian Pacific</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eastern Europe</td>
<td>OTC Remedies</td>
<td>Drizzle Nose Spray</td>
<td>584,815</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SleepTight Tablets</td>
<td>377,529</td>
</tr>
</tbody>
</table>

```sas
if upcase(_break_) ne 'REGION' then do;
  if mod(cntr,2) ne 0 then
    call define(_ROW_,'style',
      'style={background=cxeeeeeee}');
  end;
else if upcase(_break_) eq 'REGION' then do;
  call define(_ROW_,'style','style=Header');
  end;
end;
```

Another LINE Example

<table>
<thead>
<tr>
<th>Region</th>
<th>Subsidiary</th>
<th>Product</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asian Pacific</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eastern Europe</td>
<td>OTC Remedies</td>
<td>Drizzle Nose Spray</td>
<td>584,815</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SleepTight Tablets</td>
<td>377,529</td>
</tr>
</tbody>
</table>

```sas
compute after Region /
  style={foreground=white background=white
  cellheight=1px font_size=1px};
  line ' ';
endcomp;
```

```sas
  225,576
<table>
<thead>
<tr>
<th>Region</th>
<th>Subsidiary</th>
<th>Product</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asian Pacific</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eastern Europe</td>
<td>Cosmetic Products</td>
<td>HairGro Lotion</td>
<td>1,550,952</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WrinkAway Creme</td>
<td>442,724</td>
</tr>
<tr>
<td>Eastern Europe</td>
<td>OTC Remedies</td>
<td>Drizzle Nose Spray</td>
<td>584,815</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SleepTight Tablets</td>
<td>377,529</td>
</tr>
</tbody>
</table>
```
Other CALL DEFINE Examples

<table>
<thead>
<tr>
<th>Region</th>
<th>Subsidiary</th>
<th>Product</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>Cosmetic Products</td>
<td>HairGro Lotion</td>
<td>1,356,795</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WrinkAway Creme</td>
<td>1,065,312</td>
</tr>
<tr>
<td></td>
<td>OTC Remedies</td>
<td>Drizzle Nose Spray</td>
<td>1,235,135</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SleepTight Tablets</td>
<td>774,289</td>
</tr>
<tr>
<td>North America</td>
<td></td>
<td></td>
<td>$4,420,571</td>
</tr>
<tr>
<td>South America</td>
<td>Cosmetic Products</td>
<td>HairGro Lotion</td>
<td>379,630</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WrinkAway Creme</td>
<td>355,897</td>
</tr>
<tr>
<td></td>
<td>OTC Remedies</td>
<td>Drizzle Nose Spray</td>
<td>493,317</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SleepTight Tablets</td>
<td>333,065</td>
</tr>
<tr>
<td>South America</td>
<td></td>
<td></td>
<td>$2,116,197</td>
</tr>
</tbody>
</table>

Practically Perfect Tip #4

When you need to perform highlighting based on two or more variable values, use the CALL DEFINE statement.

```
if sales.sum gt 500000 and Product eq 'WrinkAway Creme' then
  call define(_ROW_,'style','style={background=yellow}');
```

Tip04.sas
Chapter 6: Using Options with ODS

1) Basic PROC REPORT, ODS Style= and ESCAPECHAR
2) Style= Statement Level Overrides
3) Overriding Style for LINES and CALL DEFINE
4) Inserting Images and Performing Traffic Lighting

Traffic Lighting
Traffic lighting is highlighting individual cells based on the cell’s value. With CALL DEFINE, you can highlight cells, rows or columns based on multiple values.

Example:

Red for bad results.
Yellow for neutral results.
Green for good results.
Adding Traffic Lighting with Formats

Two items must be completed in order to modify individual cells within a column:

1. Create a format referencing the style attribute values with PROC FORMAT.

2. Refer to the format in the STYLE option in PROC REPORT for the appropriate column.

Adding Traffic Lighting

Create the format with the appropriate value for the style attribute you want to change.

```sas
proc format;
  value tlite low <-750000 = 'light red'
    750000 - 1200000 = 'light yellow'
    1200000 <- 2000000 = 'light green'
    other = 'cx6495ED';
run;
```
Traffic Lighting with Format

Refer to the format in the STYLE= option or CALL DEFINE:

```sas
define Sales/ sum 'Sales' f=comma16.
style(column)={background=tlite.};
```

<table>
<thead>
<tr>
<th>Region</th>
<th>Subsidiary</th>
<th>Product</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>North America</td>
<td>Cosmetic Products</td>
<td>HairGlo Lotion</td>
<td>1,365,756</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WindAway Cream</td>
<td>1,065,312</td>
</tr>
<tr>
<td>OTC Remedies</td>
<td></td>
<td>Drizzle Nose Spray</td>
<td>1,285,356</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SleepTight Tablets</td>
<td>774,264</td>
</tr>
<tr>
<td>North America</td>
<td></td>
<td></td>
<td>4,420,577</td>
</tr>
</tbody>
</table>

Your Turn: Using the Correct Format Ranges

- Turn to your worksheet and submit programs Demo06.sas and Demo06_diff_format.sas. (Item 7) Review the output and note how the summary line changes whether there is an "OTHER" category or not.
Table/Cell Style Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREIMAGE POSTIMAGE</td>
<td>'C:\Logo.jpg'</td>
</tr>
<tr>
<td>BACKGROUNDIMAGE</td>
<td>&quot;PharmaSUG.jpg&quot;</td>
</tr>
<tr>
<td>PREHTML POSTHTML</td>
<td>&quot;&lt;HR size=3&gt;&quot;</td>
</tr>
<tr>
<td></td>
<td>'&lt;a href=&quot;www.sas.com&quot;&gt;SAS Web Page&lt;/a&gt;'</td>
</tr>
<tr>
<td>PRETEXT POSTTEXT</td>
<td>'This is a draft.'</td>
</tr>
<tr>
<td></td>
<td>&quot;Continue&quot;</td>
</tr>
</tbody>
</table>

Using the PREIMAGE Style Attribute

The location of the image file for HTML is the place where the file will be found when the HTML file is opened by the browser:

```sas
proc report data=salesdata nowd
style(report)={preimage='PharmaSUG.jpg'};
title 'Demo 07 - Using Images';
```
Practically Perfect Tip #5

Even though some PROC REPORT options are not used by all ODS destinations, you can simulate many of the options using the HTMLSTYLE attribute with CSS style properties and values.

The HTMLSTYLE attribute is only used for HTML destination output. However, starting in SAS 9.2, new style attributes will allow you to change attributes like border top and border bottom in order to simulate PROC REPORT options for RTF and PDF destinations, too.

Practically Perfect Tip #6

You can also put a corporate logo into the SAS TITLE statement using ODS ESCAPECHAR.
About the Speaker

Speaker  Cynthia Zender

Company  SAS Institute Inc.

Telephone  (919) 531-9012 (MST)

Comments & E-Mail  Cynthia.Zender@sas.com
**Proc Report Review – Solutions**

The correct answers are:

A/4: DEFINE - specifies HOW to use a report item

B/5: COLUMN - specifies WHICH report items are used on the report

C/2: DISPLAY - specifies that variable values should be listed without ordering or grouping

D/1: COMPUTE/ENDCOMP - defines a block of code to be executed

E/3: ACROSS - specifies that a column is created for every variable value
1. Submit program Demo00_setup.sas to establish the libref and working directory for class.
   a. The working directory where output will be written is: C:\Workshp\Zender\output

2. Submit programs Demo01a.sas, Demo01b.sas and Demo01c.sas and review the output files created.
   This chart documents how PROC REPORT options work in the various ODS destinations, as illustrated in these three programs.
   a. To learn about the PANELS= option and the COLUMNS= option submit program Demo01d.sas.
      i. PANELS= is a LISTING only destination option
         ii. COLUMNS= is an option that works in ODS PDF and ODS RTF destinations, but not in the ODS HTML destination.

<table>
<thead>
<tr>
<th>Statement</th>
<th>LISTING</th>
<th>HTML</th>
<th>RTF</th>
<th>PDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEADLINE/HEADSKIP</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>WIDTH</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>SPACING</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>DOL/DUL</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>CENTER</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SKIP</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>LS</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Repeat Character in Spanning Header</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>NUMBER and PAGENO=1</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>HEADLINE/HEADSKIP</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

In addition the FLOW option is a LISTING destination only option.

3. To see different style templates for each of the destinations, run program Demo01e_different_styles.sas
   a. Style=JOURNAL is designed especially for use for journal publication and black and white results.

4. To learn how to do Page X of Y page numbering using ODS ESCAPECHAR functions, submit program Demo01f_page_x_of_y.sas.
   a. Note that the {PAGEOF} function only works in the RTF destination, while {THISPAGE} and {LASTPAGE} work in both the PDF and RTF destination.

5. Submit program Demo02.sas and note which style attributes are used for these colors:
   a. What style attribute is set to CYAN?
   b. What style attribute is set to RED?

6. Demo03.sas illustrates the use of the CALL DEFINE statement. Demo04.sas shows how to make output with alternating rows in different colors. Demo05.sas shows the use of the URL= option to create a hyperlink in report output. Note that you will not be able to run Demo05.sas during the workshop if there is no Internet connectivity.

7. Submit programs Demo06.sas and Demo06_diff_format.sas. Compare the summary line in both outputs. Note how the OTHER specification was needed so the summary line was the same color as the Header style background.
Other Programs

Tip01.sas: This program shows how to insert "raw" RTF control strings into your RTF output file.

Tip02.sas: This program shows the interaction between CELLPACING, BORDERWIDTH and the RULES and FRAMES style attributes.

Tip03.sas: This program shows how to use various techniques to make wide tables fit on an RTF or PDF page: set ORIENTATION option to LANDSCAPE in OPTIONS statement; change FONT_SIZE attribute; and/or change the CELLPADDING style attribute.

Tip04.sas: When you need to perform highlighting based on two or more variable values in PROC REPORT, use the CALL DEFINE statement in a COMPUTE BLOCK.

Tip05.sas: Even though some PROC REPORT options are not used by all ODS destinations, you can simulate many of the options using the HTMLSTYLE attribute with CSS style properties and values. This program shows how to use some of the CSS style properties to simulate PROC REPORT options.

Tip06.sas: This program illustrates how to insert a logo at the top of a report using the PREIMAGE attribute in a TITLE statement.

Xtra_column_statement.sas: This program illustrates different ways you can use a COLUMN statement in PROC REPORT.

Xtra_page_xofy.sas: This program illustrates the way to use Page X of Y ODS ESCAPECHAR functions in RTF and PDF destinations.