

Paper 193-2008

SAS® ODS Technology for Today's Decision Makers

Sunil Gupta, Quintiles, Thousand Oaks, CA

ABSTRACT

Have you ever wondered why there has been so much excitement about ODS ever since it was introduced? How does ODS significantly enhance the presentation quality of reports and create files that were never before possible from SAS? Have you ever thought about how ODS can directly improve your daily activities and what it actually has to offer you as a programmer, statistician or manager?

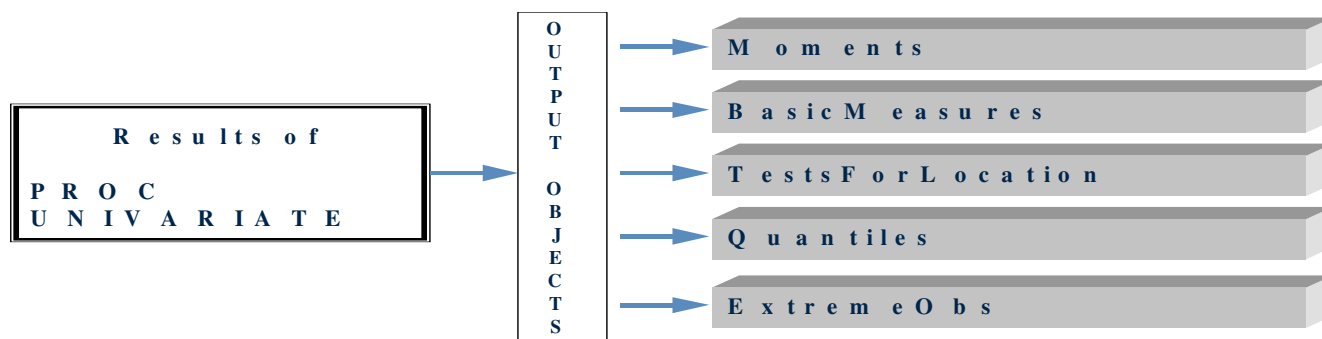
If you have ever had any of these questions about ODS, then this presentation is for you because it provides something for everyone at all levels. As a programmer, you can now create most any file types directly from SAS. In addition, the level of control and flexibility to format the output enables a higher degree of customization. As a statistician, you will want to take advantage of ODS for more efficient data analysis. The tools available in ODS facilitate multiple iterations of an analysis and ease of locating significant results. As a manager, you have various options to view and analyze your data and reports. Managers can access complex hierarchy of drill-down HTML or PDF files for a better understanding of the data or custom excel files with ODS's latest technology. This paper does not include new features of SAS version 9.13.

INTRODUCTION

- What's Unique about ODS?
- Creating Custom Styles using the Style Editor
- Creating Excel Files
- Efficient Data Analysis with ODS
- Custom Formats in RTF files
- Applying Style Syntax in Proc Tabulate, Proc Print
- Last Words

WHAT'S UNIQUE ABOUT ODS?

- Creating Output Objects (Reference to Output Object Names)
- Creating Output Files with Style (Reference to 'SAS-Supplied' Styles)

Creating Output Objects**Identifying Output Objects: List File**

Output Added: (One of the several objects created)

```

Name:          Basic Measures
Label:         Basic Measures of Location and Variability
Template:     base.univariate.Measures
Path:         Univariate.weight.Basic Measures
Label Path:   'The Univariate Procedure'. 'weight'. 'Basic
              Measures of Location and Variability'

```

Object Reference by SAS Procedure

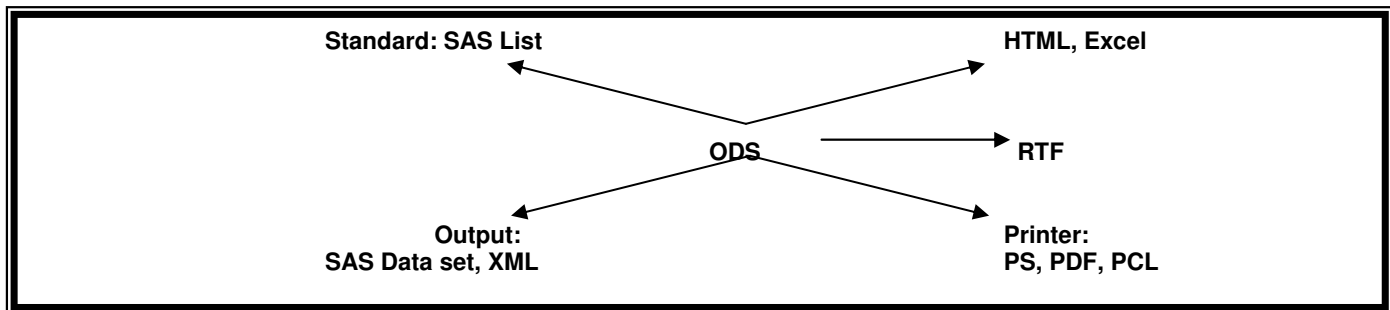
Below is an object reference table for selected SAS procedures. It can be helpful in identifying the object to specify in the ODS SELECT option.

OBJECT REFERENCE BY SAS PROCEDURE

PROCEDURE	NAME	PATH
Proc_name;	Obj_name	Proc_name.X.Obj_name
Var X;		
Run;		
Freq	OneWayFreqs	Freq.X.OneWayFreqs
Means	Summary	Means.Summary
SQL	SQL_Results	SQL.SQL_Results
Univariate	Moments	Univariate.X.Moments
	Basic Measures	Univariate.X.BasicMeasures
	TestsForLocation	Univariate.X.TestsForLocation
	TestsForNormal	Univariate.X.TestsForNormal
	Quantiles	Univariate.X.Quantiles
	Extreme	Univariate.X.ExtremeObs
	Frequency	Univariate.X.Frequency
	Plots	Univariate.X.Plots

Creating Output Files with Style

- Enhance report using predefined color, font, size, ..
- Easy to use
- “SAS-Supplied” Styles are available with installation



Creating HTML Files with Style

Using ODS is easy with basic syntax.

```

ODS HTML FILE = 'c:\demog_style.htm' STYLE = barrettsblue;

PROC UNIVARIATE DATA=DEMOG;
  VAR WEIGHT;
RUN;

ODS HTML CLOSE;
  
```

Output using Barrettsblue Style

The UNIVARIATE Procedure
Variable: weight (Weight)

Moments			
N	25	Sum Weights	25
Mean	198.68	Sum Observations	4967
Std Deviation	44.5568177	Variance	1985.31
Skewness	-0.4326447	Kurtosis	-0.2970718
Uncorrected SS	1034491	Corrected SS	47647.44
Coeff Variation	22.4264232	Std Error Mean	8.91136353

Basic Statistical Measures	
Location	Variability

Below is a collection of existing styles that can be used with the STYLE= option. Note that the default style for the LISTING and the HTML destinations is different from the default style for the PRINTER and the RTF destinations. In addition, there are several styles that are most appropriate for specific destinations such as the Printer.

PURPOSE	NAME
Screen Display for Viewing (HTML)	BarrettsBlue, Beige, Brick, D3D, Default
Editing Output (RTF)	RTF, Minimal, Theme
Printing Output (PRINTER: PDF, PS)	Printer, FancyPrinter

"SAS Supplied" STYLES

NAME	DESCRIPTION
BarrettsBlue	Blue header background, light table background
Beige	Beige header text, white text in table
Brick	Brick color header text, white text in table
Brown	Brown title, black header, light table background
D3D	White header, bold table border
Default	Dark blue header, shade table background (Default for LISTING and HTML Destinations)
Minimal	No color, light text in table
NoFontDefault	Black header text, white background table
Printer	Printer Style (Default for PRINTER Destination)
RTF	RTF Style (Default for RTF Destination)
Statdoc	Blue header, black text in table
Theme	Dark header, dark table
FancyPrinter	Printer Style
SansPrinter	Printer Style
SasdocPrinter	Printer Style
SerifPrinter	Printer Style

CREATING CUSTOM STYLES USING THE STYLE EDITOR

The Style Editor from the Enterprise Guide is an essential tool to facilitate the development of custom templates. With the user interface, it becomes easier to learn and modify style attributes without knowing the syntax. Although there are some limitations with this approach, it offers a very nice tool to get familiar with style attributes.

Creating Custom Styles using the Style Editor in Enterprise Guide

- Tools
- Style Editor

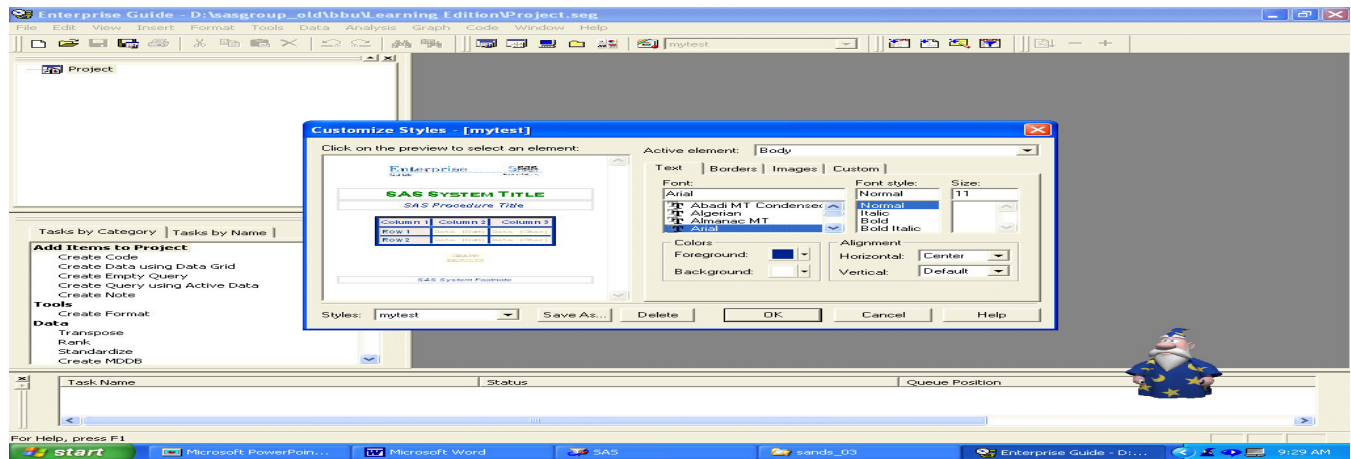
Click on preview item to select a style element

(System Title, Proc Title, Table, Header, Row Header, Data, System Footer)

Text tab: font, font style, size, color, alignment
 Border tab: attributes, color, style, width, margin, padding
 Images, Custom tabs
 Save as (new style)

- Tools
- Options
- Results
- Style (to set as default style)

The nice user interface makes it easy to change style attributes and see the results. Once the custom style is created, it can be accessed to create custom output.



The fonts used in styles define items such as titles and headers. The font definition consists of the following items: font face, font size, font weight, font style, and font width.

FONT STYLE ELEMENT

FONT WEIGHT	FONT STYLE	FONT WIDTH
Medium	Italic	Normal
Bold	Roman	Compressed
Demi_bold	Slant	Extra_compressed
Extra_bold		Narrow
Light		Wide

CREATING EXCEL FILES

A useful not so known feature of ODS is the ability to create excel files. By using the HTML destination with the .xls extension, ODS creates excel files. As a manager, you know how easy it is to analyze and share data in excel files.

```
ODS HTML FILE = 'c:\demog.xls' style = minimal;
PROC UNIVARIATE DATA=DEMOG;
  VAR WEIGHT;
RUN;
ODS HTML CLOSE;
```

Creating Excel file – demog.xls

The SAS System			
The UNIVARIATE Procedure			
Variable: weight (Weight)			
Moments			
N	25	Sum Weights	25
Mean	198.68	Sum Observations	4967
Std Deviation	44.65682	Variance	1985.31
Skewness	-0.4326447	Kurtosis	-0.2970718
Uncorrected SS	1034491	Corrected SS	47647.44
Coeff Variation	22.426423	Std Error Mean	8.9113635
Basic Statistical Measures			
Location		Variability	
Mean	198.68	Std Deviation	44.65682
Median	199	Variance	1985
Mode		Range	159
		Interquartile Range	55
Tests for Location: Mu0=0			
Test	Statistic		p Value
Student's t	t	22.29513	Pr > t < .0001
Sign	M	12.5	Pr >= M < .0001
Signed Rank	S	162.5	Pr >= S < .0001
Quantiles (Definition 5)			
Quantile	Estimate		
100% Max	270		
99%	270		
95%	264		
90%	267		
75% Q3	228		
50% Median	199		

EFFICIENT DATA ANALYSIS WITH ODS

As a statistician, you can use ODS tools to improve your productivity.

- Selecting Objects
- Creating Output Data sets
- Creating Multiple Data sets with the MATCH_ALL option
- Using Traffic Lighting conditions to identify significance
- Creating Plots and Tables as RTF file (Reference to drill-down graph)

Selecting Output Objects By Name

```
ODS HTML FILE = 'c:\select_demog.htm' ;
```

```
ODS HTML SELECT BASICMEASURES;
ODS HTML SHOW;
```

```
PROC UNIVARIATE DATA=DEMOG;
  VAR WEIGHT;
RUN;
```

```
ODS HTML CLOSE;
```

Selecting Multiple Output Objects from BY Statements

```
PROC SORT DATA=DEMOG;          /* 1 - Active, 2 - Placebo */
```

```
  BY DRUG;
RUN;
```

```
ODS HTML FILE='c:\select_weight_demog.htm';
```

```
ODS HTML SELECT
```

```
UNIVARIATE .BYGROUP1 .WEIGHT .BASICMEASURES
UNIVARIATE .BYGROUP2 .WEIGHT .BASICMEASURES;
```

Selecting Multiple Output Objects from BY Statements (cont.)

```
PROC UNIVARIATE DATA=DEMOG;
  BY DRUG;
  VAR WEIGHT;
RUN;

ODS HTML CLOSE;
```

Object Name: **BASICMEASURES**

Object Path Name: **UNIVARIATE.BYGROUP1.WEIGHT.BASICMEASURES**

The screenshot shows the SAS output for the UNIVARIATE procedure, specifically for the variable 'weight' (Weight) in the 'Drug=Active' group. The output is titled 'The UNIVARIATE Procedure Variable: weight (Weight)'. Below the title, there is a section for 'Drug=Active' which contains a table of 'Basic Statistical Measures'. This table is divided into two columns: 'Location' and 'Variability'.

Basic Statistical Measures			
Location		Variability	
Mean	191.5789	Std Deviation	43.40419
Median	198.0000	Variance	1884
Mode		Range	146.00000
		Interquartile Range	51.00000

Object Name: **BASICMEASURES**

Object Path Name: **UNIVARIATE.BYGROUP2.WEIGHT.BASICMEASURES**

The screenshot shows the SAS output for the UNIVARIATE procedure, specifically for the variable 'weight' (Weight) in the 'Drug=Placebo' group. The output is titled 'The UNIVARIATE Procedure Variable: weight (Weight)'. Below the title, there is a section for 'Drug=Placebo' which contains a table of 'Basic Statistical Measures'. This table is divided into two columns: 'Location' and 'Variability'.

Basic Statistical Measures			
Location		Variability	
Mean	221.1667	Std Deviation	44.14710
Median	226.0000	Variance	1949
Mode		Range	108.00000
		Interquartile Range	85.00000

Creating Output Data Sets

```

                (Object Name) (Data set Name)
ODS OUTPUT BASICMEASURES = MEASURE;

PROC UNIVARIATE DATA=DEMOG;
  VAR WEIGHT;
RUN;

ODS OUTPUT CLOSE;

```

Creating Output Data Sets: List File

It is always a good idea to print the data set contents to see the variable names and data structure. Note that this is the same information that is normally displayed when applying proc univariate.

```

* Proc print of MEASURE Output Data Set;
  Measure as SAS Data set

Obs      Var  Loc
Name Measure LocValue  VarMeasure          VarValue

1  weight  Mean    198.6800 Std Deviation      44.55682
2  weight  Median  199.0000 Variance            1985
3  weight  Mode     .          Range              159.00000
4  weight           _          Interquartile Range 56.00000

```

Creating Output Data Sets

- Works with any SAS Procedure
- All results are available in the data set
- Options are available for creating multiple data sets

Note: Requires having information about the object. Can use the ODS TRACE statement to get this information.

One of the useful options when saving objects as data sets is the MATCH_ALL option. Instead of having to specify each data set created, you can reference the macro variable &measure_dsn.

Creating Multiple Data Sets with the MATCH_ALL option

```

ODS OUTPUT BASICMEASURES          /* object name */
  (MATCH_ALL = MEASURE_DSN        /* macro name */
  PERSIST = PROC)                 /* maintain selection list across procedures */
  = MEASURE;
  /* data set names - measure, measure1, measure2 */

PROC UNIVARIATE DATA=DEMOG;
  VAR WEIGHT HEIGHT;
RUN;

PROC UNIVARIATE DATA=DEMOG;
  VAR AGE;
RUN;

ODS OUTPUT CLOSE;

%PUT Macro variable measure_dsn = &MEASURE_DSN;

```

Log File

```

101 PROC UNIVARIATE DATA=DEMOG;
102 VAR WEIGHT HEIGHT;
103 RUN;

NOTE: The data set WORK.MEASURE has 4 observations and 7 variables.
NOTE: The data set WORK.MEASURE1 has 4 observations and 7 variables.
NOTE: PROCEDURE UNIVARIATE used (Total process time):
      real time          1.92 seconds
      cpu time           0.05 seconds

104 PROC UNIVARIATE DATA=DEMOG;
105 VAR AGE;RUN;

NOTE: The data set WORK.MEASURE2 has 4 observations and 7 variables.
NOTE: PROCEDURE UNIVARIATE used (Total process time):

Macro variable measure_dsn = measure measure1 measure2

```

Creating Multiple Data Sets with the MATCH_ALL option

```

data allmeans;
  set &measure_dsn;
  /* set measure measure1 measure2; */
  where locmeasure = 'Mean';
  keep varname locmeasure locvalue;
run;

```

By knowing the variable name of interest, with the match_all option, it becomes easy to save only the mean values of selected variables.

Proc Print

Obs	Var Name	Loc Measure	LocValue
1	weight	Mean	198.6800
2	height	Mean	66.8400
3	age	Mean	56.5280

Another useful tool is the feature to specify traffic lighting conditions to identify significance in the data. With the format statement having colors as labels, the PROC REPORT syntax can accept the format names being equal to the background style attribute. The background style attribute should be set equal to the format name.

Using Traffic Lighting Conditions to identify significance

```

proc format;
  value traffic low - 99 = 'green' /* cx006600 */
               100 - high = 'orange' /* cxFF9900 */ ;
quit;

ods html file = 'c:\print_results.htm';

proc report data=allmeans nowd;
  columns varname locvalue;
  define varname / 'Variable' ;
  define locvalue / 'Mean' style = {background = traffic.};
run;

ods html close;

```


The result enables us to quickly identify data values for example that are greater than 100.

Output Using Traffic Lighting Conditions

The SAS System

Variable	Mean
weight	198.6800
height	66.8400
age	56.5280

When using traffic lighting conditions, these are some key points to consider:

Key Points to Remember

- Create format with color labels ex. 'cx006600'
- (See Lauren Haworth's paper, SAS with Style: Creating your own ODS Style Template, for more information.)
- Works best for HTML destination
- Best to use for background color
- Works only for selected procedures – Proc Print, Proc Report, Proc Tabulate

In addition to text, ODS enables you to combine plots with text in your output file.

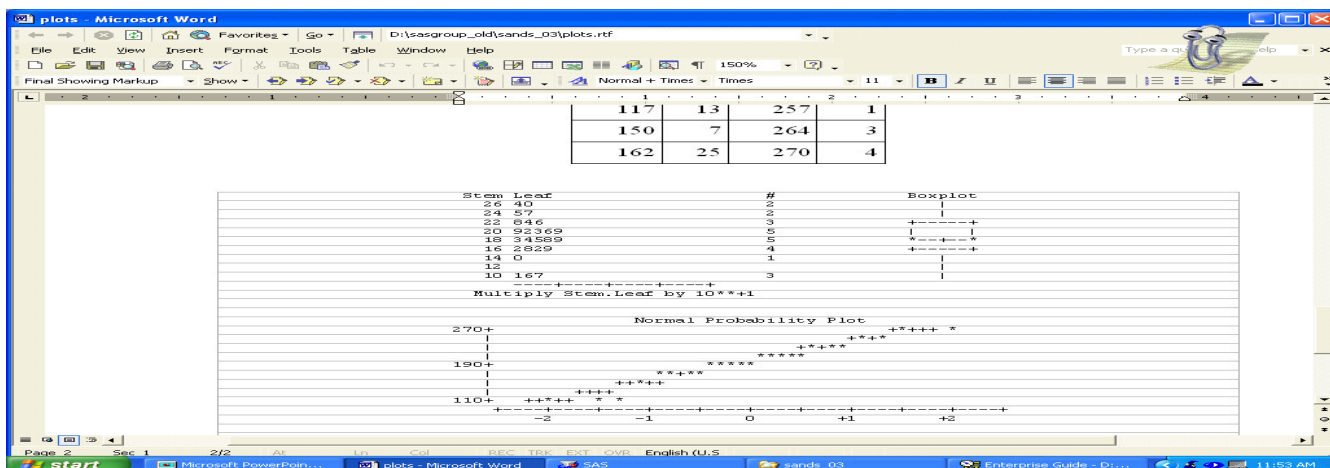
Creating Plots and Tables as RTF File

```
ods rtf file = 'c:\plots.rtf';

proc univariate data = demog plots;
  var weight;
run;

ods rtf close;
```

Output for Creating Plots and Tables as RTF File



CUSTOM FORMATS IN RTF FILES

For industries such as the pharmaceutical industry, there is a greater focus on creating custom RTF files. ODS provides tools to easily create customized RTF files.

Customize RTF File

ODS ESCAPECHAR = '^';

- In-Line Formatting
Font, Bold, *Italic*, Size, Superscript, Subscript, RTF control words
- Header/Footer sections
Titles and Footnotes, Page X of Y
- Page-Break Control
- Using PROC TABULATE, REPORT and PRINT Style options

In-Line Formatting as Style Options

- Font, **Bold**, *Italic*
Style=[font_face='arial'];
Style=[font_weight=bold];
Style=[font_style=italic];
- Size
Style=[font_size=1.3fs];

Below is the syntax for specifying styles in titles. Just after the escapechar value, the style attribute can be specified before the text to be affected. Note that the {pageof} tag is a SAS Version 9.13 option.

Header/Footer Sections - Titles

```
Title font=arial bold
      "Table 1. Baseline Table: Sex, Race^{super a}      Page ^{pageof} ";

      Font:      font=<courier|times|arial|helvetica>
                  <bold|median|light>
                  <italic|roman|slant>;
      Height:    height=8pt;
      Justify:    j=<left|right|center>;
      Superscript: ^{super &text};      Subscript: ^{sub &text};
      Page X of Y: ^{pageof};      * SAS Version 9.13;
      "{Page} {\field{\*\fldinst{ PAGE }}} \~{of} \~{\field{\*\fldinst { NUMPAGES }}}}"
      * SAS Version 8.2;
```

Note that the syntax for specifying styles in footnotes is different from that of titles. Styles are turned on and then turned off. This syntax is similar to the style syntax in Proc Print, Proc Report and Proc Tabulate.

Header/Footer Sections – Footnotes

```
Footnote1 "^S={font_face=arial font_style=italic} ^{super a}Race: Non-White consists of
           Black, Hispanic, and Native American^S={}" ;

Footnote2 "^S={font_face=arial font_style=italic}Program: /stat/druga/program1.sas ^S={}" ;
```

Font:

```
"^S={font_face=arial}&text      ^S={}";
"^S={font_weight=bold}&text    ^S={}";
"^S={font_style=italic}&text   ^S={}";
```

ODS is powerful enough to enable most any Microsoft Word format feature including *italics*, underline and **bold**.

RTF Control Words

	<u>Effect</u>	<u>Control Word</u>
	<i>italics</i>	\i
	underline	\ul
➤	bullet	\bullet
	bold	\b
	_{subscript}	\sub
	^{superscript}	\super
	strike	\strike

This references other formats that can be specified in SAS programs:

(<http://msdn.microsoft.com/library/default.asp?url=/libraryen-us/dnrftspec/html/rftspec.asp>)

The process of specifying RTF control words in SAS programs is as follows:

1. Carrot '^' symbol is best as the escapechar symbol since the RTF syntax uses the '\' symbol.
2. Use the following formula and conditions: "text" || '^R"\rtf-control-word text \rtf-control-word0 "'
 - a. Double quotes (" ") around "text" strings.
 - b. Concatenate all strings with || operator.
 - c. Use single quote (to set ^R) and then double quote to start rtf-control-word string ("rtf-control-word). This turns on the Microsoft Word style setting.
 - d. The text to be affected.
 - e. Repeat and close each rtf-control-word string with 0, double quote (") then single quote ('), (\rtf-control-word0 "'). This turns off the Microsoft Word style setting.

Title and Data Step with RTF Control Words

```
ods escapechar = '^';

title1 '^R/RTF"\ul " Title contains underlined text';

data rtfcontrol;
  a = "The text uses RTF control words " ||
    '^R"\i italic text \i0 "' || ' regular text ' ||
    '^R"\ul underlined text \ul0 "' ||
    '^R"\strike strike text \strike0 "' ;
run;

ods rtf file = 'c:\rtf_control.rtf';

proc print data=rtfcontrol;
run;

ods rtf close;
```

Output using RTF Control Words

The result can be seen below. For the first time, SAS programs can control word formats.

Title contains underlined text

Obs	a
1	The text uses RTF control words <i>italic text</i> regular text <u>underlined text</u> strike text .

Since word files can be landscape or portrait, SAS now has options to specify the orientation of your output file. You do not need to worry about setting the line size or page length for output files. Another option that helps the presentation layout is the center option.

General RTF Settings

```
OPTIONS ORIENTATION=PORTRAIT nodate center nonumber;

ODS ESCAPECHAR = '^';

ODS RTF FILE = 'C:\baseline_tables.rtf'
  STYLE = MINIMAL;          /* Basic MS Word table, no color */

SAS Procedure(s)

ODS RTF CLOSE;
```

Proc Format used for publication tables

```
proc format;
  value gender 0 = 'Female'
              1 = 'Male';
  value race 0 = 'White'
             1 = 'Non-White';
  picture pctpct low - 99.4 = '99%'
                99.5 - high = '999%' (prefix=' (');
quit;
```

SAS makes it easy to apply styles at the procedure level instead of having to create a custom style definition or table template. Styles can be placed on most any TABULATE procedure statement to provide a variety of effects.

APPLYING STYLE SYNTAX IN PROC TABULATE

Using PROC TABULATE - Table 1

```
%let mystyle =%str(font_face="arial" font_weight=bold);

PROC TABULATE data=demog missing formchar=' -----'
  style=[font_face="arial"];          /* Style for all data */
                                     /* Style for column and row header */
class gender race drug/style=[&mystyle];
format gender gender. race race.;
table (gender race), drug=' '*
      (n='N'*f=7. pctn<gender race>=' %' *f=pctpct. )
      / box=[label='Baseline '          /* Style for box label */
           style=[&mystyle] ] rts=43; run;
```

In the title, the text is arial and bold. In addition, a superscript 'a' and Page X of Y is also specified. Notice that the footnotes are arial and italics with a superscript 'a'.

Output Using PROC TABULATE - Table 1

Table 1. Baseline Table: Sex, Race^a Page 1 of 3

Baseline	Active		Placebo	
	N	%	N	%
Sex				
Female	3	15%	1	16%
Male	16	84%	5	83%
Race				
White	5	26%	3	50%
Non-White	14	73%	3	50%

^aRace: Non-White consists of Black, Hispanic, and Native American
Program: /stat/druga/program1.sas

APPLYING STYLE SYNTAX IN PROC TABULATE with CLASSLEV and KEYWORD statements

Using PROC TABULATE - Table 2

```
%let mystyle =%str(font_face="arial" font_weight=bold);
PROC TABULATE data=demog missing formchar='-----'
  style=[font_face="arial"];          /* Style for all data */
                                     /* Style for column and row header */

class gender race drug/style=[&mystyle];
format gender gender. race race.;
classlev gender race drug/style=[&mystyle];
keyword n pctn/style=[&mystyle];

table (gender race), drug=' '*
      (n='N'*f=7. pctn<gender race>=' %' *f=pctpct. )
      / box=[label='Baseline '          /* Style for box label */
          style=[&mystyle] ] rts=43;

run;
```

With the CLASSLEV and KEYWORD statements and style options, you can get the arial font for the sub levels of Sex and Race along with the N and % column headers. This table looks more professional now.

Output Using PROC TABULATE - Table 2

Table 2. Baseline Table: Sex, Race^a Page 2 of 3

Baseline	Active		Placebo	
	N	%	N	%
Sex				
Female	3	15%	1	16%
Male	16	84%	5	83%
Race				
White	5	26%	3	50%
Non-White	14	73%	3	50%

^a Race: Non-White consists of Black, Hispanic, and Native American
 Program: /stat/druga/program1.sas

For those who do not use the TABULATE procedure, you will be happy to know that you can also apply styles in the PRINT procedure.

APPLYING STYLE SYNTAX IN PROC PRINT

```
PROC PRINT data=demog (obs=5) noobs
  style(header) = [font_face="arial" font_weight=bold]
  style(data) = [font_face="arial"];          /* Style for headers */
  style(data) = [font_face="arial"];          /* Style for all data */
  var patient drug gender race age height weight;
run;
```

Output Using PROC PRINT - Table 3

Table 3. Sample Proc Print Page 3 of 3

patient	drug	gender	race	age	height	weight
001	Active	Male	Non-White	67.9	74.4	257
002	Active	Male	White	36.7	63.1	168
003	Placebo	Male	White	74.6	69.6	264
004	Placebo	Male	Non-White	73.8	63.2	270
005	Active	Male	Non-White	57.8	67.8	209

^a Race: Non-White consists of Black, Hispanic, and Native American
 Program: /stat/druga/program1.sas

SUMMARY

Today's ODS Technology offers something for everyone including Programmers, Statisticians, and Managers. The power and flexibility of ODS allow for customizing output, using the style editor to create new styles, creating excel files, creating SAS data sets, and directly applying style syntax in Proc Tabulate, Proc Report, and Proc Print procedures. Using the traffic lighting feature enables you to quickly identify significant values. With RTF control words, you can generate text such as underline, *italic*, or **bold**.

REFERENCES

- Gupta, Sunil, Quick Results with the Output Delivery System, WUSS 2004
- Gupta, Sunil. 2003, *Quick Results with the Output Delivery System*. Cary, NC: SAS Institute Inc.
- Hamilton, Paul, ODS to RTF: Tips and Tricks, SUGI 28
- Lauren Haworth, SAS with Style: Creating your own ODS Style Template, Genentech Inc.
- McNeill, Sandy, Changes & Enhancements for ODS by Example (Through version 8.2), SAS Institute, Inc.
- McNeill, Sandy, What's New in the Output Delivery System, version 9.0, SAS Institute, Inc.
- McNeill, Sandy, Ray Pass, PROC REPORT: Doin' It in Style
- O'Connor, Dan, Next Generation Data _NULL_ Report Writing Using ODS OO Features, SUGI 28, SAS Institute, Inc.
- Parker, Chevell, Generating Custom Excel Spreadsheets Using ODS, SUGI 28, SAS Institute, Inc.
- SAS version 8.2 and 9.0 On-Line Doc
- Schellenberger, Brian, ODS LAYOUT: Arranging ODS Output as You See Fit, SUGI 28, SAS Institute, Inc.
- Schellenberger, Brian, Presentation Quality Tabular Output via ODS, SAS Institute, Inc.
- Slaughter, Susan and Lora Delwiche, ODS for Reporting with Style

TRADEMARK INFORMATION

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.

ABOUT THE AUTHOR

The author welcomes your comments and suggestions.

Sunil K. Gupta

Associate Director of Statistical Programming

Quintiles

325 E. Hillcrest Dr., Suite 200

Thousand Oaks, CA 91360

Phone: (805)-557-7700

E-mail: Sunil.Gupta@Quintiles.com

Sunil is the Associate Director, Statistical Programming at Quintiles. He has been using SAS® software for over 14 years and is a SAS Base Certified Professional. He has participated in over 6 successful FDA submissions. His projects with pharmaceutical companies include the development of a Macro-Based Application for Report Generation and Customized Plots and Charts. He is also the author of *Quick Results with the Output Delivery System*, developer of over five SAS programming classes, developer of Clinical Trial Reporting Templates for quick generation of tables, lists and graphs and was a SAS Institute Quality Partner™ for over 5 years. Most recently, he released his new book, *Data Management and Reporting Made Easy with SAS Learning Edition 2.0*, and has co-authored the book *Sharpening Your SAS Skills*.

