Paper 94-27

What does the style definition look like?

Wei Cheng, ISIS Pharmaceuticals, Inc., Carlsbad, CA

ABSTRACT

The SAS® Output Delivery System (ODS) has a number of built-in output style definitions. ODS also provides tools to create your own custom-defined style definitions. This paper presents a SAS program to generate output in HTML, PDF, and RTF file format with a "Table of Contents" linking to each available style definition. Some advanced ODS techniques including PROC TEMPLATE, ODS statements and options, PROC REPORT, and SAS macro facility are used and covered.

INTRODUCTION

The SAS Output Delivery System (ODS) can send output to different destinations including Listing, HTML, Printer (PDF, PS, PCL), SAS data sets, etc. You can use the build-in output style definitions, or you can use your own custom-defined style definitions to specify the color, border, font, etc. ODS uses a default style definition for each output destination, unless you specify a style definition using STYLE= option in the ODS statement.

You need to know what the available styles look like before you choose the right style definition or define your own style definition. You can view the different style definitions by changing the STYLE= option in the ODS statement. There are at least seventeen built-in style definitions with the SAS V8.2, and you may have many custom-defined style definitions. It will be tedious and time-consuming to change the STYLE= option to view all the style definitions for all the output destinations.

This paper will go through the code step by step to build a program that can generate output for each style definition in HTML, RTF, and PDF file format. You only need to open the "Table of Contents" file to view all the style definitions available in your SAS system. The complete program is included as an appendix.

DEFINE THE DIRECTORY PATH TO STORE THE OUTPUT

```
   %let dir = C:\sugi\sugi27\styles;
```

This macro variable stores the directory path for all the output generated by the program. It should be modified from run to run, and it is the only part that needs modification when you run the program on your computer.

CREATE A CUSTOM-DEFINED STYLE DEFINITION

In case you don't have any custom-defined style definition, the program will create one calls SUGI27 for you.

```
proc template;
   define style sugi27;
   parent = Styles.sasdocPrinter;
   replace fonts /
      'TitleFont2' = ("Arial, Helvetica", 8pt,
Bold)
      'TitleFont' = ("Arial, Helvetica", 8pt,
Bold)
      'StrongFont' = ("ITC Bookman, Times Roman",
8pt, Bold)
      'EmphasisFont' = ("ITC Bookman, Times
Roman", 8pt, Italic)
      'FixedEmphasisFont' = ("Courier", 8pt,
Italic)
      'FixedStrongFont' = ("Courier", 8pt, Bold)
      'FixedHeadingFont' = ("Courier", 8pt,
Bold)
      'BatchFixedFont' = ("SAS Monospace,
Courier", 8pt)
      'FixedFont' = ("Courier", 8pt)
      'headingEmphasisFont' = ("ITC Bookman,
Times Roman", 8pt, Bold Italic)
      'headingFont' = ("ITC Bookman, Times
Roman", 8pt, Bold)
      'docFont' = ("ITC Bookman, Times Roman",
8pt);
   replace Table from Output /
      background = white
      frame = VOID
      cellpadding = 4pt
      cellspacing = 0.75pt
      borderwidth = 0.75pt;
   end;
run;
```

By default, the custom-defined style definitions will be saved in SASUSER.TEMPLAT, verses the built-in style definitions which are stored in SASHELP.TMPLMST. If you don't like this default place for your custom-defined style definitions, you can use the ODS PATH statement to change it.

GET A LIST OF ALL THE AVAILABLE STYLE DEFINITIONS

By using PROC TEMPLATE and ODS OUTPUT statement we can import the names of all the built-in style definitions into a SAS data set:

```
ods output stats = slist(where = (upcase(path)
ne 'STYLES') keep = path);

proc template;
   list styles;
run;

ods output close;
```

Using the same technique to get the names of custom-defined style definitions:

```
ods output stats = slist2(keep = path);

proc template;
   list / store = sasuser.templat;
run;

ods output close;
```

Then, we can put the names of all available style definitions together:

```
data slist;
```

```
   length path $ 50;
   set slist
        slist2;
   path = scan(path,-1,'.');
```

And put the names of style definitions into a series of macro variables:

```
proc sql noprint;
   select compress(path) into :style1 -
:style999 from slist;
quit;
```

PROC SQL generates a macro variable &SQLOBS to count the number of available style definitions, which will be used later in the program.

CREATE A SAMPLE DATA SET FOR STYLE DEMONSTRATION

We need to have a sample data set to generate the output in different output destinations, so we create one, MYTEST:

```
Data MyTest;
   input Patient $ Age Gender $ Race $;
   cards;
   001 43 Male Caucasian
   002 50 Male Hispanic
   003 36 Female Hispanic
   004 52 Male Caucasian
   005 33 Male Caucasian
   006 48 Female Caucasian
;
```

We also create one data set BLANK for the front page of HTML destination. You will see its usage later.

```
data blank;
   text = 'Please click on the Table of Contents
on the right to choose a style to view.';
run;
```

SET UP ENVIRONMENT

Clear the existing title and footnote, and set system options:

```
title;
footnote;
options nodate;
run;
```

And supress the procedure title in the output:

```
ods noptitle;
```

HTML OUTPUT DESTINATION

The following macro will display each style definition in a HTML file and generate a “Table of Contents” file. You only need to open the “HTMLstylesFrame.htm" to view each style definition in HTML file format.

```
%Macro DisplayStylesInHTML;

ods html path = "&dir"(url = NONE)
          frame = "HTMLstylesFrame.htm"
          (title = "What does the style look
like?")
          contents = "HTMLstylesContents.htm"
           body = "HTMLstylesblank.htm"
           style = sasweb;
ods proclabel "Home";
title "What does the style look like?";
proc report data = blank nowd;
   define text / ' ' style = [font_size = 15];
run;
%Do C = 1 %to &sqlobs;
   ods html body = "HTMLstyle_&&Style&C...htm"
style = &&Style&C;
   title2 "This Style is &&Style&C";
   ods proclabel "This Style is &&Style&C";
   proc report data = MyTest nowd headline;
   run;
%End;
ods html close;

%Mend DisplayStylesInHTML;

%DisplayStylesInHTML;
```

RTF OUTPUT DESTINATION

The following macro will display each style definition in RTF file format:

```
%Macro DisplayStylesInRTF;

%Do C = 1 %to &sqlobs;
   ods rtf file =
"&dir.\RTFstyle_&&Style&C...rtf" style =
&&Style&C;
   title 'What does the style look like?';
   title2 "This Style is &&Style&C";
   proc report data = MyTest nowd headline;
   run;
   ods rtf close;
%End;

%Mend DisplayStylesInRTF;

%DisplayStylesInRTF;
```

PDF OUTPUT DESTINATION

The following macro will display each style definition in PDF file format:

```
%Macro DisplayStylesInPDF;

%Do C = 1 %to &sqlobs;
   ods pdf file =
"&dir.\PDFstyle_&&Style&C...pdf" style =
&&Style&C notoc;
   title 'What does the style look like?';
   title2 "This Style is &&Style&C";
   proc report data = MyTest nowd headline;
   run;
   ods pdf close;
%End;

%Mend DisplayStylesInPDF;

%DisplayStylesInPDF;
```

“TABLE OF CONTENTS”

The last step is to create “Table of Contents” for RTF and PDF files. CALL DEFINE statement in PROC REPORT is used to hyperlink the “Table of Contents” to each style definition.

```
data slist;
   length pdfname rtfname $ 100;
    set slist;
    pdfname = "file:PDFstyle_" ||
    compress(path) || '.pdf';
    rtfname = "file:RTFstyle_" ||
    compress(path) || '.rtf';
run;

ods pdf file = "&dir.\PDFstyle_Table of
Contents.pdf" notoc;
title 'What does the style look like?';
title2 "Table of Contents";
proc report data = slist nowd headline;
   column pdfname path;
   define pdfname / noprint;
   define path / width = 50 left "Style";
   compute path;
      call define(_col_, "URL", pdfname);
   endcomp;
run;
ods pdf close;

ods rtf file = "&dir.\RTFstyle_Table of
Contents.rtf";
title 'What does the style look like?';
title2 "Table of Contents";
proc report data = slist nowd headline;
   column rtfname path;
   define rtfname / noprint;
   define path / width = 50 left "Style";
   compute path;
      call define(_col_, "URL", rtfname);
   endcomp;
run;
ods rtf close;
```

You can open the “RTFstyle_Table of Contents” to view each style definition in RTF file format, and open the “PDFstyle_Table of Contents” to view each style definition in PDF file format.

CONCLUSION

This paper shows a method to view available style definitions using the technique of ODS. SAS users can then take advantage of these advanced ODS features to customize the report with the built-in style definitions or custom-defined style definitions. The program can be adapted to other output destinations with a few modifications.

REFERENCES

SAS Institute, Inc., *The Complete Guide to the SAS® Output Delivery System, Version 8*, Cary, NC: SAS Institute, Inc., 1999.

Haworth, Lauren E., *Output Delivery System: The Basics*, Cary, NC: SAS Institute Inc., 2001.

CONTACT INFORMATION

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

Wei Cheng
Isis Pharmaceuticals, Inc.
2292 Faraday Ave.
Carlsbad, CA 92008
Work Phone: (760) 603-3807
Fax: (760) 603-2588
Email: wcheng@isisph.com
Web: http://www.geocities.com/prochelp

APPENDIX: THE SAS CODE

```
*****************************************;
**  PROGRAM: STYLE_DEMO_SUGI27.SAS       **;
**  AUTHOR:  WEI CHENG     JUNE, 2001    **;
**  NOTE:    THIS PROGRAM WILL GENERATE  **;
**           THE STYLE DEMO IN A         **;
**           DIRECTORY. IT ILLUSTRATES   **;
**           THE USE OF AVAILABLE STYLES.**;
*****************************************;

* DIRECTORY TO HOLD ALL THE DEMOS *;
***********************************;

%let dir = C:\sugi\sugi27\styles;

* CREATE A CUSTOM DEFINED STYLE SUGI27 *;
***************************************;

ods path sasuser.templat (UPDATE)
         sashelp.tmplmst (READ);

proc template;
   define style sugi27;
   parent = Styles.sasdocPrinter;
   replace fonts /
      'TitleFont2' = ("Arial, Helvetica", 8pt,
Bold)
      'TitleFont' = ("Arial, Helvetica", 8pt,
Bold)
      'StrongFont' = ("ITC Bookman, Times
Roman", 8pt, Bold)
      'EmphasisFont' = ("ITC Bookman, Times
Roman", 8pt, Italic)
      'FixedEmphasisFont' = ("Courier", 8pt,
Italic)
      'FixedStrongFont' = ("Courier", 8pt, Bold)
      'FixedHeadingFont' = ("Courier", 8pt,
Bold)
      'BatchFixedFont' = ("SAS Monospace,
Courier", 8pt)
      'FixedFont' = ("Courier", 8pt)
      'headingEmphasisFont' = ("ITC Bookman,
Times Roman", 8pt, Bold Italic)
      'headingFont' = ("ITC Bookman, Times
Roman", 8pt, Bold)
      'docFont' = ("ITC Bookman, Times Roman",
8pt);
   replace Table from Output /
      background = white
      frame = VOID
      cellpadding = 4pt
      cellspacing = 0.75pt
      borderwidth = 0.75pt;
   end;
run;

* GET A LIST OF THE STYLE DEFINITIONS *;
***************************************;

ods listing close;

ods output stats = slist(where = (upcase(path)
ne 'STYLES') keep = path);

proc template;
   list styles;
run;

ods output close;

**** FOR USER DEFINED STYLES ****;

ods output stats = slist2(keep = path);

proc template;
   list /store = sasuser.templat;
run;

ods output close;

data slist;
   length path $ 50;
   set slist
        slist2;
   path = scan(path,-1,'.');
run;

proc sql noprint;
   select compress(path) into :style1 -
:style999 from slist;
quit;

* CREATE A SAMPLE DATA SET *;
****************************;

Data MyTest;
   input Patient $ Age Gender $ Race $;
   cards;
   001 43 Male Caucasian
   002 50 Male Hispanic
   003 36 Female Hispanic
   004 52 Male Caucasian
   005 33 Male Caucasian
   006 48 Female Caucasian
;

data blank;
   text = 'Please click on the Table of Contents
on the right to choose a style to view.';
run;

* SET UP ENVIRONMENT *;
**********************;

ods noptitle;
title;
footnote;
options nodate;
run;

* HTML OUTPUT DESTINATION *;
***************************;

%Macro DisplayStylesInHTML;

ods html path = "&dir"(url = NONE)
          frame = "HTMLstylesFrame.htm"
          (TITLE = "What does the style look
like?")
         contents = "HTMLstylesContents.htm"
         body = "HTMLstylesblank.htm"
          style = sasweb;
%* BODY= IS REQUIRED IF THE HTML DESTINATION IS
          CLOSED;
ods proclabel "Home";
title "What does the style look like?";
proc report data = blank nowd;
   define text / ' ' style = [font_size = 15];
run;
%Do C = 1 %to &sqlobs;
   ods html body = "HTMLstyle_&&Style&C...htm"
             style = &&Style&C;
   title2 "This Style is &&Style&C";
   ods proclabel "This Style is &&Style&C";
```

```
   proc report data = MyTest nowd headline;
   run;
%End;
ods html close;

%Mend DisplayStylesInHTML;

%DisplayStylesInHTML;

* RTF OUTPUT DESTINATION *;
*************************;

%Macro DisplayStylesInRTF;

%Do C = 1 %to &sqlobs;
   ods rtf file =
    "&dir.\RTFstyle_&&Style&C...rtf" style =
    &&Style&C;
   title 'What does the style look like?';
   title2 "This Style is &&Style&C";
   proc report data = MyTest nowd headline;
   run;
   ods rtf close;
%End;

%Mend DisplayStylesInRTF;

%DisplayStylesInRTF;

* PDF OUTPUT DESTINATION *;
*************************;

%Macro DisplayStylesInPDF;

%Do C = 1 %to &sqlobs;
    ods pdf file =
    "&dir.\PDFstyle_&&Style&C...pdf" style =
    &&Style&C notoc;
    title 'What does the style look like?';
    title2 "This Style is &&Style&C";
    proc report data = MyTest nowd headline;
    run;
    ods pdf close;
%End;

%Mend DisplayStylesInPDF;

%DisplayStylesInPDF;

* TABLE OF CONTENTS *;
*********************;

data slist;
   length pdfname rtfname $ 100;
   set slist;
   pdfname = "file:PDFstyle_" ||
   compress(path) || '.pdf';
   rtfname = "file:RTFstyle_" ||
   compress(path) || '.rtf';
run;

ods pdf file = "&dir.\PDFstyle_Table of
Contents.pdf" notoc;
title 'What does the style look like?';
title2 "Table of Contents";
proc report data = slist nowd headline;
   column pdfname path;
   define pdfname / noprint;
   define path / width = 50 left "Style";
   compute path;
      call define(_col_, "URL", pdfname);
   endcomp;
run;
ods pdf close;

ods rtf file = "&dir.\RTFstyle_Table of
Contents.rtf";
title 'What does the style look like?';
title2 "Table of Contents";
proc report data = slist nowd headline;
   column rtfname path;
   define rtfname / noprint;
   define path / width = 50 left "Style";
   compute path;
      call define(_col_, "URL", rtfname);
   endcomp;
run;
ods rtf close;
```