

ODS: A Primer

Matthew Grover, S-Street Consulting, Austin, TX

Abstract:

The objective of this paper is to provide an overview of the features provided by the Output Delivery System (ODS). ODS was first released in version 7 of SAS and in subsequent releases has seen increased functionality. This paper focuses on 3 key features of ODS.

- 1.) The ability to isolate distinct pieces of information created by a procedure into separate datasets. For example, output from Proc Univariate can be separated into Basic Measures, TestsForLocations, Quantiles, and ExtremeObs.
- 2.) Destinations, an additional feature of ODS, provide a variety of choices (HTML, RTF, MARKUP, XML, PCL) for redirecting output.
- 3.) PROC Template, which provides further customization of ODS output content (column order, data formats) and style (font size, font face, color).

Rationale

The Output Delivery System was specifically designed to overcome the output limitations found in early versions of the SAS System. Prior to the existence of ODS, SAS was often used to crunch numbers, but the reporting format was extremely limited, still very much rooted in its mainframe ancestry. As a result, SAS output was often exported to another software tool to be converted into presentation quality output.

Pre-ODS Limitations:

- Some procedures did not provide the option of saving the procedure results to an output dataset, only a listing was created.
- Writing to another file format (RTF, PDF, HTML) was difficult and tedious, usually involving complex code that was hard to maintain
- Style for output was non-existent, mono-spaced fonts were the only option
- Access to a specific piece of result data was not simple; the solution to this problem usually involved writing the output to disk and then reading it back into SAS and parsing the lines to find the desired results.

ODS Solutions:

- ODS provides the ability to select or exclude specific pieces of information created by a procedure
- ODS logically separates procedural output into separate data tables.
- ODS supports writing to various output destinations; HTML, RTF, PDF.
- ODS provides styles to control the look of the output.
- ODS via table definitions allows control of output content.

I. Output Objects

Output Objects are created when a procedure is executed. The output object is actually a combination of the data produced by the procedure and a table definition that describes fonts sizes and types, order of data, data formats, and column header size. An output object, therefore, contains the results of the procedure and also contains information regarding how the results should be displayed.

Most SAS procedures create output objects in tabular format. However, reporting procedures (PROC REPORT, PROC TABULATE, PROC PRINT) do not produce output objects. The reporting procedures primary function is to display output that has already been created by a prior data statement or SAS procedure.

The number of individual output objects that are created is procedure dependent. PROC Univariate produces the output objects; *BasicMeasures*, *TestsforLocation*, *Quantiles*, *ExtremeObs*, *Moments*; and PROC MEANS produces *Summary* as its only output object. By default, all output objects get created and no ODS messages are written to the log.

To make use of an output object, a programmer must first know the output objects created by a specific procedure.

ODS TRACE statement writes to the log the name of each output object created by that procedure.

Syntax: ODS trace /options

Example 1: Determining Output Objects

```
ods trace on;
```

```
proc corr data=sashelp.shoes;  
var sales returns;  
run;
```

```
proc means data=sashelp.shoes;  
var sales ;  
run;
```

```
ods trace off;
```

Log Snippet:

Proc Corr: Output Objects

Output Added:

```
-----  
Name:      VarInformation  
Label:     Variables Information  
Template:  base.corr.VarInfo  
Path:      Corr.VarInformation  
-----
```

Output Added:

```
-----  
Name:      SimpleStats  
Label:     Simple Statistics  
Template:  base.corr.UniStat  
Path:      Corr.SimpleStats  
-----
```

Output Added:

```
-----  
Name:      PearsonCorr  
Label:     Pearson Correlations  
  
Template:  base.corr.StackedMatrix  
Path:      Corr.PearsonCorr  
-----
```

PROC Means: Output Objects

Output Added:

```
-----  
Name:      Summary  
Label:     Summary statistics  
Template:  base.summary  
Path:      Means.Summary  
-----
```

Once the names of the output objects created by procedure is known, additional ODS statements can be issued to control what objects are actually written to open destinations.

ODS SELECT – selects what output objects get written to open destinations.

Syntax: ODS Select *outputObject*

Example 2: Selecting Output Objects

```
ods select simplestats;
```

```
ods output simplestats=work.stats;
```

```
proc corr data=sashelp.shoes;  
var sales returns;  
run;
```

```
ods output close;
```

ODS EXCLUDE – excludes output objects from being written to open destinations

Syntax: ODS Exclude *outputObject*

Example 3: Excluding Output Objects

```
ods exclude varInformation  
PearsonCorr;
```

```
ods output simplestats =work.stats;
```

```
proc corr data=sashelp.shoes;  
var sales returns;  
run;
```

```
ods output close;
```

II. Destinations

An ODS destination determines where the results of procedure will be written. The basic syntax for specifying a destination is straightforward. The real difference in destinations is predicated upon the large number of destination dependent options. The default destination is listing and it remains as an open destination unless explicitly closed. As of version 8.2, ODS supports writing to multiple destinations simultaneously.

DISPLAY - Opens a window for testing and debugging purposes.

HTML - Creates Hyper Text Markup Language, ideal for web publishing

ODS LISTING - Standard Listing output

ODS MARKP - Replace ODS XML and now supports creation of custom tagsets, not just XML

ODS OUTPUT - Creates a SAS data set

ODS PRINTER - Sends output directly to the printer

ODS PDF - Great for providing printer friendly output on the web. Replaces ODS PRINTER PDF

ODS PS - Replaces ODS PRINTER PS, now a real destination.

ODS PCL - Replaces ODS PRINTER SAS PRINTER=PCL4, designed to be easier to use.

ODS RTF - Rich Text Format (RTF) is useful for importing into a word processor like Microsoft Word

To write to a destination, it must first be opened. To open a destination, issue the ODS statement with a destination and an option to specify where the destination should be written, normally a file. To close a destination issue the ODS statement, destination type, and the work 'close'. It is important to note that once a destination is opened, it remains open until it is closed or modified. The following examples demonstrate the opening and closing HTML destinations.

Syntax: ODS <destinations> /options

Example 4: HTML Destination

```
/* Full-fledged html output */

ods html body='c:\temp\body.html'
      frame='c:\temp\frame.html'

contents='c:\temp\contents.html'
      page='c:\temp\page.html';

proc corr data=sashelp.shoes;
var sales returns;
run;

ods html close;
```

```
/* Write ALL output objects to one
file */
```

```
ods html body='c:\temp\corr.html';

proc corr data=sashelp.shoes;
var sales returns;
run;

ods html close;
```

See Figure 1 in Appendix for Output.

```
/* Write only SimpleStatics out to a
file */
```

```
ods select SimpleStatistics;

ods html body='c:\temp\stats.html';

proc corr data=sashelp.shoes;
var sales returns;
run;
```

```
ods html close;
```

See Figure 2 in Appendix for Output.

III Templates

Templates are definitions that can contain information regarding table attributes and style attributes. Each template definition contains individual elements or attributes.

SAS provides default template definitions which are stored in the template store SASHELP.TMPLMST. A template store is special item store that holds template definitions created with PROC TEMPLATE. Template stores can contain multiple levels often referred to as directories. A directory can hold a template or subdirectory. When using ODS to send output to a destination, SAS first searches the template store SASUSER.TEMPLAT and then looks for the template in template store SASHELP.TMPLMST. As part of its functionality, ODS allows users to define and store templates in their own template store. Unless templates exist in SASUSER.TEMPLAT data store, the default templates located in SASHELP library are used. The search path for templates can be modified to include user created template stores.

ODS PATH – sets the path for template stores.

Templates support the object-oriented concept of inheritance. The parent statement in PROC

TEMPLATE contains the name of the parent that template definition. Templates can be created without using an existing parent, but most frequently a programmer will just modify one of the default templates provided with SAS. Templates are created, deleted, modified with PROC TEMPLATE. SAS currently supports the following template definitions;

- Table
- Column
- Header
- Footer
- Style
- Tagset

This paper will focus on showing how to create simple style and table definitions

Table Definitions:

A table definition can be utilized to control the content of the output. Column order, data formats, column headers, stacking of observations, and table headers and footers are all related to content.

The following steps will outline the process for creating a custom table definition

- Create code and examine the default output
- Determine the template definition associated with the output object
- Write the existing template definition to file for modification
- Modify the template definition as needed
- Save the template definition to template store that is in the ODS path

Example 5: Steps to Creating a Table Definition

Step 1: Evaluate the output from default template

```
ods html
body='c:\temp\SummaryStats.html' ;

title 'Shoes Store Statistics';

proc means n mean median
data=sashelp.shoes;
var sales returns;
run;

ods html close;
```

Step 2: Determine output object template

```
ods trace on;

proc means n mean median
data=sashelp.shoes;
var sales returns;
run;

ods trace off;
```

Log Snippet:

Output Added:

```
-----
Name:          Summary
Label:         Summary statistics
Template:      base.summary
Path:          Means.Summary
-----
```

Step 3: Save the template elements to a file.

```
proc template;
  source base.summary/
file='C:\temp\summarysource.sas';
run;
```

Step 4: Create your own custom template.

```
ods path sasuser.templat(update)
sashelp.tmplmst(read);

Proc template;
define table Base.Summary;
  notes "Summary Stats";
  column variable (label) (n) (median)
(mean) ;

  header table_header1;
  define table_header1;
    text " Weekly Numbers";
    space = 1;
    just = C;
    spill_margin;
  end;
  define label;
    header = "Financials";
    id;
    generic;
  end;
define n;
```

```

        header = "N";
        generic;
    end;
    define mean;
        header = "Mean";
        generic;
    end;
    define median;
        header = "Median";
        generic;
    end;
    define footer foot_table;
        text ' 9/20/2002';
    end;

end;
run;

```

Step 5: Using your own custom template

```

ods noptitle;

ods path sasuser.templat(update)
sashelp.tmplmst(read);

ods html
body='c:\temp\SummaryStats.html';

title 'Shoes Store Statistics';
proc means n mean median
data=sashelp.shoes;
var sales returns;
run;

ods html close;

```

See Figure 3 in Appendix for Output.

Styles:

Style definitions are probably the most commonly modified template definition. Like their name implies, style definitions control font size, font face, font style, foreground colors and background colors. SAS provides a variety of styles automatically.

The default html style definition is **STYLES.DEFAULT** found in the **SASHELP.TMPLST** template store. To look at source for the default style, use the **SOURCE** statement.

Example 6: Listing source for default style.

```
proc template;
```

```

source styles.default /
    file='C:\temp\stylesdefault.sas';
run;

```

STYLES.DEFAULT is not the only style found in the **SASHELP.TMPLMST** template store. Use the **LIST** statement to get a list of all styles that are available.

Example 7: Listing all default styles.

```
proc template;
list styles;
run;
```

Log Snippet:

Listing of: SASHELP.TMPLMST

Path Filter is: Styles
Sort by: PATH/ASCENDING

Obs	Path	Type
1	Styles	Dir
2	Styles.BarrettsBlue	Style
3	Styles.Beige	Style
4	Styles.Brick	Style
5	Styles.Brown	Style
6	Styles.D3d	Style
7	Styles.Default	Style
8	Styles.Minimal	Style
9	Styles.NoFontDefault	Style
10	Styles.Printer	Style
11	Styles.Rtf	Style
12	Styles.Statdoc	Style
13	Styles.Theme	Style
14	Styles.fancyPrinter	Style
15	Styles.sansPrinter	Style
16	Styles.sasdocPrinter	Style
17	Styles.serifPrinter	Style

With the table definition statement, the code from the previous example now produces the exact desired content, however, the style of the output leaves a lot to be desired. Using the code from the previous example, we will work to create a style definition that we can associate with HTML destination to give our output a whole new look.

Color Style:

Example 8: Defining a new color style:

```
ods path sasuser.templat(update)
sashelp.tmplmst(read);
```

```

Proc template;
  define style styles.mycolors;
    parent=styles.default;
    replace color_list/
      'fgB2'=cx004488
      'fgB1'=cx6495ED
      'fgA4'=cxFFF0F5
      'bgA4'=cxF0F0F0
      'fgA3'=cx000000
      'bgA3'=cxFFF8DC
      'fgA2'=cx669999
      'bgA2'=cxEEE8AA
      'fgA1'=cx2F4F4F
      'bgA1'=cx5F9EA0
      'fgA'  =cx8B008B
      'bgA'  =cxFFF8DC
    ;
  end;

```

```

ods html
body='c:\temp\SummaryStats.html'
style=styles.mycolors;

```

```

title 'Shoes Store Statistics';

```

```

proc means n mean median
data=sashelp.shoes;
var sales returns;
run;

```

```

ods html close;

```

Font Style:

Example 9: Defining a new font style:

```

ods path sasuser.templat(update)
sashelp.tmplmst(read);

```

```

proc template;
  define style Styles.myfonts;
    parent = styles.default;
    replace fonts /
      'TitleFont2' = ("Matisse ITC",8,Bold
Italic)
      'TitleFont' = ("Matisse ITC ",5,Bold
Italic)
      'StrongFont' = ("Arial",4,Bold)
      'EmphasisFont' = ("Arial",3,Italic)
      'FixedEmphasisFont' = ("Courier
New",2,Italic)

```

```

'FixedStrongFont' = ("Courier
New",2,Bold)
'FixedHeadingFont' = ("Courier New",2)
'BatchFixedFont' = ("Courier New",2)
'FixedFont' = ("Courier New",1)
'headingEmphasisFont' =
("Arial",4,Bold Italic)
'headingFont' = ("CopperPlate
GothicBold",2,Bold)
'docFont' = ("Arial",1);
end;
run;

```

```

ods html
body='c:\temp\SummaryStatsFonts.html'
style=styles.myfonts;

```

```

title 'Shoes Store Statistics';

```

```

proc means n mean median
data=sashelp.shoes;
var sales returns;
run;

```

```

ods html close;

```

See Figure 4 in Appendix for Output.

Conclusion:

ODS is an extremely powerful tool that will allow SAS users to easily create presentation quality output form within the SAS System. For information on ODS see the books listed in the reference section and check out the ODS FAQ on the SAS website.

References:

SAS Institute Inc. (1999). The Complete Guide to the SAS Output Delivery System, Version 8, SAS Institute Inc. Cary, NC

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

Olinger, Chris, "Twisty Little Passages, All Alike – ODS Templates Exposed", PharmaSUG '99, SAS Institute Inc., Cary, NC

McNeill, Sandy, "Changes and Enhancements for ODS by Example (through Version 8.2), SUGI 27, SAS Institute Inc. Cary, NC.

APPENDIX

Figure 1: Default Output from PROC CORR

Shoes Store Statistics

2 Variables

Sales Returns

Shoes Store Statistics

Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum	Label
Sales	395	85700	29107	33851566	325 00000	1298717	Total Sales
Returns	395	2967	4612	1172092	10 00000	57362	Total Returns

Pearson Correlation Coefficients, N = 395

Prob > |r| Under H0: Rho = 0

	Sales	Returns
Sales Total Sales	1.00000	0.96484 < .0001
Returns Total Returns	0.96484 < .0001	1.00000

Figure 2: Output with just one Output Object Selected

Shoes Store Statistics							
Simple Statistics							
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum	Label
Sales	395	85700	129107	33851566	325 00000	1298717	Total Sales
Returns	395	2967	4612	1172092	10 00000	57362	Total Returns

Figure 3: Using the New Table Definition

Shoes Store Statistics			
The MEANS Procedure			
Weekly Numbers			
Financials	N	Mean	Median
Total Sales	395	85700	88942.0

Weekly Numbers			
Financials	1	Mean	Median
Total Sales	39	35700	38912
Total Returns	20	2867.2	1438.00
9/20/2002			

Figure 4: Using the new Font Style

Shoes Store Statistics

Weekly Numbers			
Financials	1	Mean	Median
Total Sales	39	35700	38912
Total Returns	20	2867.2	1438.00
9/20/2002			