

Making the Output Delivery System (ODS) Work for You

William Fehlner, SAS Institute (Canada) Inc., Toronto, Ontario

ABSTRACT

Over the years, a variety of options have been offered in order to give a SAS® user access to SAS procedure output. In Version 7 of SAS Software, the Output Delivery System (ODS) has come of age and offers a standard means to capture output and convert it into formatted reports, html files or SAS data sets.. This tutorial introduces basic techniques for using ODS

- to transfer output values to a SAS data set,
- to create a custom print file,
- to generate an html file for a web site and
- to customize output prior to download to a spreadsheet.

Only basic knowledge of the SAS system is assumed. You should be familiar with DATA step basics (e.g. formats, labels) and be able to create reports from some procedures (e.g. univariate) when default options are used. For additional technical details, a list of references is provided as a starting point for learning more.

INTRODUCTION

Prior to Version 7, SAS procedures that produced printed output (that is, output destined for the procedure output file) generated output that was designed for a traditional line-printer. As a consequence, the Printto procedure has been used to reroute procedure output to a text file. In the past, many lines of SAS data step code have been invested in converting this output to other styles, or extracting key information from the standard report. These programs are no longer required.

In Version 7, procedure output is much more flexible. There is no longer a requirement to post process your output in order to extract information or to reformat a report. The Output Delivery System (ODS) enhances your ability to manage procedure output. Tools are available to select specific components of output, to control the report format with templates, and to enhance html output with styles. ODS tools can also be used to enhance reports created with a SAS data step.

ODS is part of base SAS software. The Template procedure is used to create and modify both templates and styles. Templates and styles exist

as a hierarchy in which a child template (style) can inherit or override aspects of the parent template (style) behaviour.

ODS is available as part of base SAS software on all platforms supported by SAS Institute. The sample output shown in this tutorial are screen shots from SAS software running under Windows 95. Sample code used in this tutorial is from the sample library. Full directions for accessing this library are in the reference section.

DEFAULT BEHAVIOUR

ODS is designed so that when SAS code written prior to version 7 is run without change in Version 7, it will continue to generate the same output as in the past. Even in this situation however, ODS provides an additional benefit not available prior to Version 7. In particular, when SAS procedures are run interactively, they also generate a table of contents displayed in the Results window of display manager. Although the Output Manager window is still available, the Results window extends its functionality significantly.

Consider the following SAS code from the SAS sample library (to find it, select "base SAS" and then select "SAS language and Procedures: Usage, Chapter 1"):

```
title 'Sample from a Normal Distribution';
title2 'with Mean=50 and Standard Deviation=10';
footnote;
data normal;
  drop n;
  do n=1 to 10000;
    x=10*rannor(53124)+50;
    output;
  end;

run;
proc univariate normal data=work.normal;
  var x;
run;
```

When this code is run interactively, the display in Figure 1 is generated.

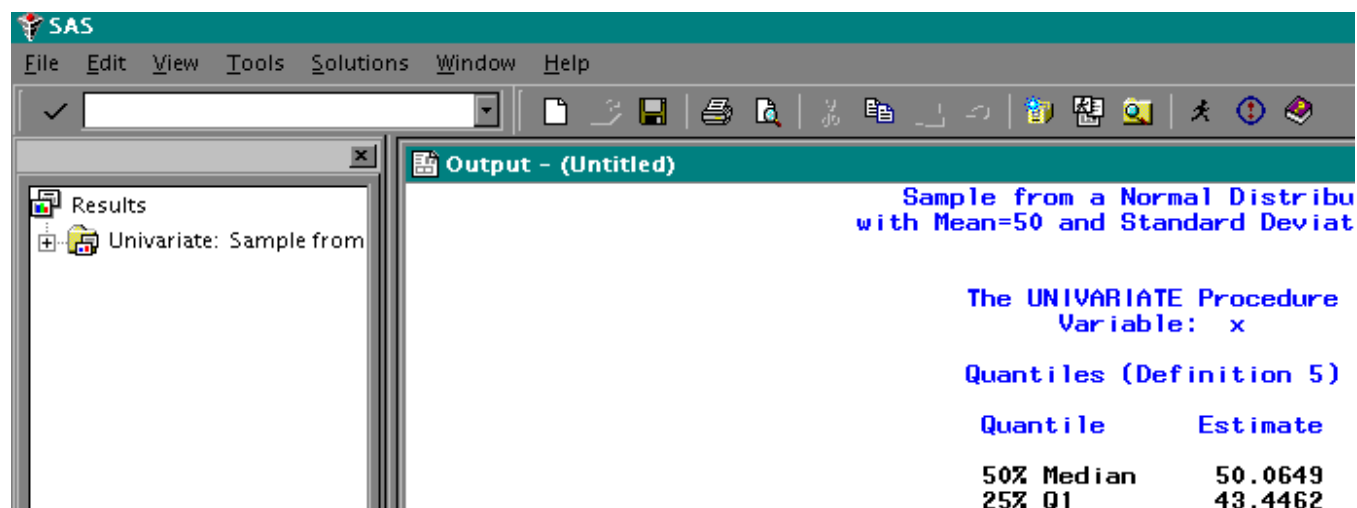


Figure 1. Interactive results for default settings.

Note that initially, the explorer on the left is unexpanded, and the Output window on the right displays the last page of the most recent output. Figure 2 shows the explorer after all folders are opened.

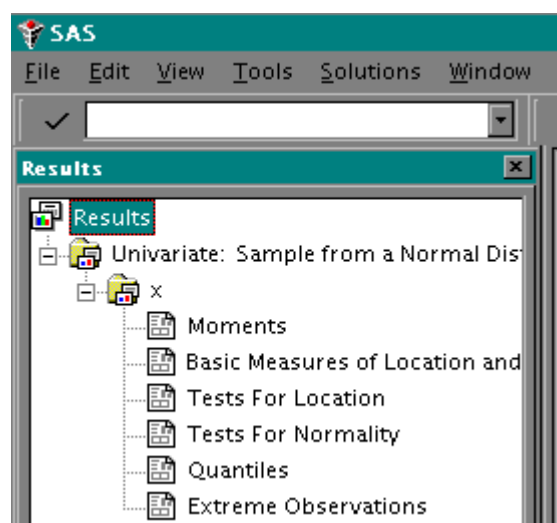


Figure 2. Expanded Results window from Figure 1.

The Results window now displays a hierarchy of output folders and output objects created by the Univariate procedure. The top-level folder contains all of the output from the procedure. The second level represents variables on the VAR statement. In this case, there is one folder for the *x* variable, the only variable on this VAR statement. Six output objects have been created for this variable, and each output object is linked to the corresponding segment of the report in the Output window. When you double click on an output object in the Results window, the corresponding report segment scrolls to the top of the Output window. Additional levels in the folder hierarchy will be created when you use a BY statement with a SAS procedure.

ODS STATEMENTS

In order to move beyond default behaviour, ODS provides three means of controlling the output from a SAS procedure - ODS statements, templates, and styles. The starting point is the set of ODS statements. One or more of these statements are used to

- Choose a result destination (or destinations).
 - ODS LISTING ...
 - ODS HTML ...
 - ODS OUTPUT ...
- Select desired output objects to create.
 - ODS SELECT ...
 - ODS EXCLUDE ...
- Control the type of messages written in the Log window.
 - ODS SHOW ...
 - ODS TRACE ...
 - ODS VERIFY ...
- Set a search path for template(s) to format listing and html results.
 - ODS PATH ...
- Select a style to control aspects of html results.
 - ODS HTML STYLE= ...

The default action in each case when ODS statements are not used is to

- route output to the Listing destination.
- create all possible output objects.
- not write any ODS messages to the Log window.
- look first in SASUSER for custom templates. If custom versions are not found, then look in SASHELP for the templates shipped with SAS software.
- use the Default style if none is specified in the template.

The next few sections explore how to develop solutions by using ODS statements to override the defaults.

IDENTIFYING OUTPUT OBJECTS

In order to do more than simply display an output object, you must be able to determine its name, path and template. This is the purpose of the ODS TRACE statement. Trace can be turned on or off at any time during a SAS session or SAS batch job. When trace is on, it writes to the SAS log a record of each output object that is created.

Figure 3 shows a part of the log produced by the following code:

```
ODS TRACE on;
proc univariate normal data=work.normal;
  var x;
run;
ODS TRACE off;
```

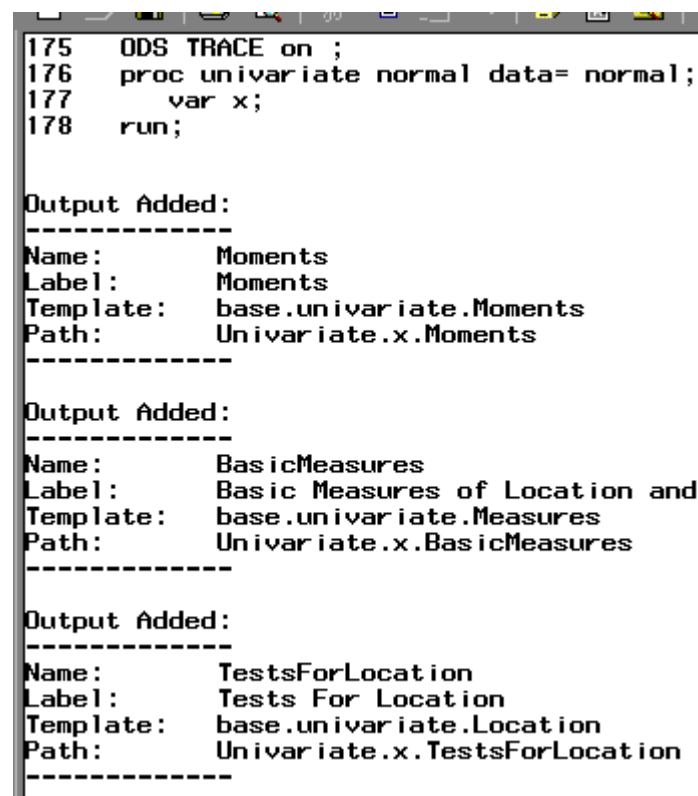


Figure 3. Record of output objects in the SAS log

To understand how to use this trace, consider the third output object that appears in the log. Label: "Tests for Location" identifies the output object in the Results window and the matching report segment in the Output window. Path: "Univariate.x.TestsForLocation" is used to uniquely identify this output object in other ODS statements. Name: "TestsForLocation" can be used in place of the full path to identify the output object when there is only one output object with a given label, as in the current example. Finally, Template: "base.univariate.Location" is the full name of the template that ODS used to format the output object. The section "Using Templates" has an example that modifies a template by means of the Template procedure.

TRANSFER OUTPUT VALUES TO A SAS DATA FILE

A major advantage of ODS is that numeric values created by a SAS procedure are stored at maximum precision in output objects. Thus there is no loss of accuracy when values are routed to a SAS data file from an output object. The ODS OUTPUT statement makes the connection. Consider the following example:

```
ODS OUTPUT TestsForLocation = work.capture;
proc univariate normal data=work.normal;
  var x;
run;
```

The ODS OUTPUT statement copies data values from the output object with the name "TestsForLocation" to a SAS data file called "work.capture". Figure 4 shows the contents of work.capture. Note the number of significant figures in the Student's t value.

VarName	Test	Te	Stat	pType	pValue
x	Student's t	t	504.3515590326	Pr > t	<.0001
x	Sign	M	5000	Pr >= M	<.0001
x	Signed	S	25002500	Pr >= S	<.0001

Figure 4. Maximum precision in a SAS data file.

SELECTING OUTPUT OBJECTS

Once output object names and paths have been identified (see section "Identifying Output Objects"), a subset of output objects can be selected for output. Either the ODS SELECT or the ODS EXCLUDE statement accomplishes this task. When the output object name is used, all output objects with that name are selected. When a path is used, a single unique output object is selected.

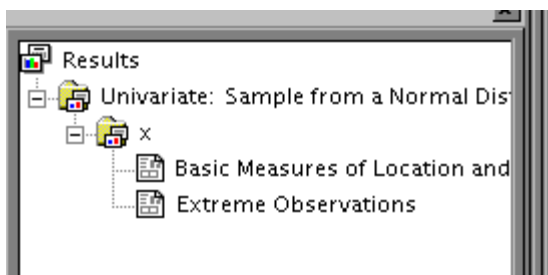


Figure 5. Results when output objects are selected.

Consider the following example where two output objects are kept. Figure 5 shows the Results window when the following code is run.

```

ODS SELECT BasicMeasures ExtremeObs;
proc univariate normal data=work.normal;
  var x;
run;
    
```

CREATING HTML OUTPUT

The output of a SAS procedure can be rendered as a set of html pages with little more effort than it takes to request ODS to create listing output. The same templates that control the listing output set the general formatting rules for the html pages. It is also possible to modify or enhance the Default style that controls attributes of html output.

The ODS HTML statement opens, manages, or closes the HTML destination. If the destination is open, you can create HTML procedure output (output that is written in Hyper Text Markup Language). A simple form of the ODS HTML statement opens up from one to four files for output. Consider the following example:

```

ODS SELECT BasicMeasures TestsForLocation
  ExtremeObs;
ODS HTML body = 'listingOutput.html'
  contents = 'tableOfContents.html'
  frame = 'frameTiesItTogether.html'
  page = 'tableOfPages.html';
proc univariate normal data=work.normal;
  var x;
run;
ODS HTML close;
    
```

The second html file, the table of contents, serves the same purpose for the browser as the Results window does for the SAS display manager. The third html file, the frame, is used to link the output pages, the table of contents and the table of pages (the fourth html file) into a single display. Only the first html file, the body, is required. Figure 6 shows how a browser renders this set of four html files.

<p>Table of Contents</p> <hr/> <p>I. The Univariate Procedure</p> <ul style="list-style-type: none"> <input type="checkbox"/> x <ul style="list-style-type: none"> <input type="checkbox"/> Basic Measures of Location and Variability <input type="checkbox"/> Tests For Location <input type="checkbox"/> Extreme Observations <hr/> <p>Table of Pages</p> <hr/> <p>I. The Univariate Procedure</p> <ul style="list-style-type: none"> <input type="checkbox"/> Page 1 	<p style="text-align: center;">Sample from a Normal Distribution with Mean=50 and Standard Deviation=10</p> <p style="text-align: center;">The UNIVARIATE Procedure</p> <p style="text-align: center;">Variable: x</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th colspan="4">Basic Statistical Measures</th> </tr> <tr> <th colspan="2">Location</th> <th colspan="2">Variability</th> </tr> </thead> <tbody> <tr> <td>Mean</td> <td>50.03237</td> <td>Std Deviation</td> <td></td> </tr> <tr> <td>Median</td> <td>50.06492</td> <td>Variance</td> <td>9</td> </tr> <tr> <td>Mode</td> <td>.</td> <td>Range</td> <td>7</td> </tr> <tr> <td></td> <td></td> <td>Interquartile Range</td> <td>1</td> </tr> </tbody> </table>	Basic Statistical Measures				Location		Variability		Mean	50.03237	Std Deviation		Median	50.06492	Variance	9	Mode	.	Range	7			Interquartile Range	1
Basic Statistical Measures																									
Location		Variability																							
Mean	50.03237	Std Deviation																							
Median	50.06492	Variance	9																						
Mode	.	Range	7																						
		Interquartile Range	1																						

Figure 6. Browser rendering of ODS HTML output files

The three ODS destinations are independent of each other. If listing output is not required, then use an

```
ODS LISTING off;
```

statement to close the listing destination and the same statement with the "on" option to reopen it. Similarly, if the HTML destination has been opened, an

```
ODS HTML off;
```

statement closes it. SAS data files can be requested whether or not one or both of the other two destinations are open.

USING TEMPLATES

Many procedures that fully support ODS provide templates that enable you to customize each piece of output that the procedure produces. In a preceding section on "Identifying Output Objects the ODS TRACE statement was used to identify a template associated with a particular piece of output. This section demonstrates how to access the template code, make simple modifications, and store the result.

Templates that are supplied by SAS Institute, Inc. are stored in a template store in the SASHELP library. By default, the Template procedure places custom templates in SASUSER. Again by default, when a template is required, ODS searches for a template in SASUSER before SASHELP, unless the search path has been altered with the ODS PATH statement. An example of an ODS PATH statement that adds a personal template store to the search path is given at the end of this section.

In order to search a template store, click the mouse menu button on the Results icon in the Results window. Select "Templates" from the popup menu and then expand the Sashelp.Tmplmst folder several levels until you open the procedure folder of interest. Figure 7 shows the Templates window at this stage.

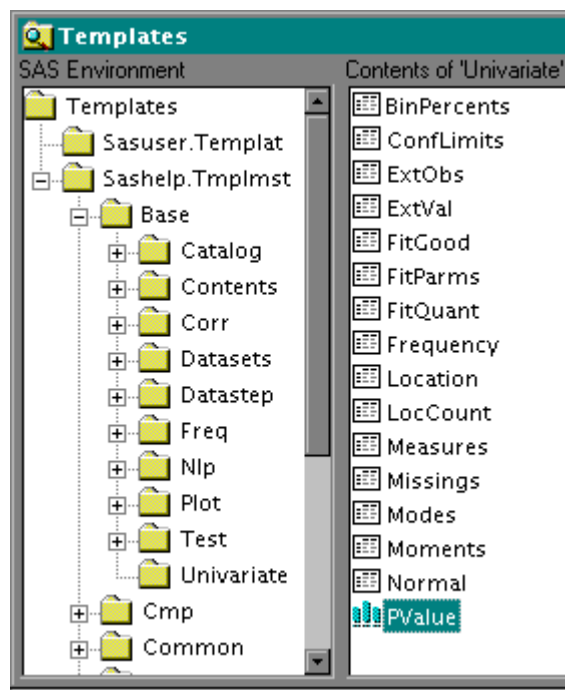


Figure 7. Partial List of templates for the Univariate procedure

To make changes to a template use the following process.

- Double click on a template icon to open the Template Browser window.
- Use File → Save as to copy the text into a text file.
- Include the text into the Program editor.
- In order to place the new template in your personal template store, add an ODS PATH statement that places your template store ahead of the default stores (see example below).
- Make appropriate changes to header text, formats, etc.
- Submit the code. If your personal template store does not yet exist, it will be created.

The following code segments are taken from base.univariate.measures.

```
proc template;
  define table Base.Univariate.Measures;
    notes
    'Basic measures of location and variability';
    dynamic WgtHdr;
    column LocMeasure LocValue VarMeasure
           VarValue;
    header h1 h2 h3;
    translate _val=._ into '';
    .
    .
  define h2;
    text 'Location';
    end = LocValue;
    start = LocMeasure;
  end;
  .
  .
  define LocValue;
    space = 5;
    print_headers = OFF;
  end;
  .
  .
end;
run;
```

The key words to look for initially are "header", "footer", "text" and "format". To modify an attribute, or override its default value, insert the attribute name and attribute value within the appropriate define - end block of code. For example,

- to reverse the position of the display of 'Location' values and 'Variability' values, modify the header statement.
- to change header content, modify the text statement in the define h2 block.
- To override the default format for the LocValue column, add a format statement to the define LocValue block.

The modified statements are as follows:

```
ODS PATH mylib.mystore(update)
         Sasuser.Templat(update)
         Sashelp.tmplmst(read) ;
Proc template;
  .
  header h1 h3 h2;
  .
  define h2;
    text 'Measures of Location';
    end = LocValue;
    start = LocMeasure;
  end;
  .
  define LocValue;
    space = 5;
    format = 7.3;
    print_headers = OFF;
  end;
  .
run;
```

USING STYLES

A style determines the overall look of the document that uses it. Each style is a collection of style elements, each of which affects a particular part of the document. Creating custom styles is very similar to creating custom templates, since styles are stored within a template store and are created by the Template procedure.

The primary difference between templates and styles is that the code for a style may include a Parent statement. This allows you to inherit all the attributes of a parent style. Thus you need only code explicit changes or extensions to the parent. For example, all styles but one in the list of styles shipped with SAS software have the Default style as a parent or grandparent. The one exception, the Minimal style, provides a starting point for the adventurous type who wants to create a style from scratch.

In order to see a list of styles shipped with the SAS system, open the template window, expand the Sashelp.Tmplmst folder and open the style folder. As with templates, a double click on a style icon will display the Template procedure code used to create the style. This code also can be copied, modified and run to create a custom version of the style.

The online documentation contains several pages of detailed information on style attributes that you can set. Directions to this material on the SAS OnlineDoc™ CD are given in the Reference section.

ODS AND THE DATA STEP

Extensions to the FILE statement and the PUT statement permit ODS to be accessed from within a SAS data step used to write a report. In this way, a template can be given responsibility for all aspects of an output line in the report. The bottom line is that the SAS data step does what it does best, complex calculations. The template focuses on data column attributes, not on individual locations in the file output buffer. Thus you can build complex reports with less effort than in the past.

The extension to the FILE statement is used to bind a template definition to variables in the data step Program Data Vector. As the following example shows, Template procedure code has many similarities to Report procedure code. In effect, the viewer definition (template) has been isolated from the model definition (data step).

In this example, a SAS data file is available that contains three variables, first name, last name and account balance. The object is to create a multi-panel report listing the full name and balance.

```
option linesize=85 nodate;
title ;

proc template;
  define table level1.level2.level3.showbal;

    column name balance;
    header sptitle1;
    panels=2;

    define name;
      header = 'Last, First Name';
      format=$20.;
    end;
    define balance;
      header = 'Account Balance';
      format=dollar10.0;
    end;

    define sptitle1;
      text "Customer Account Balances";
      underline;
    end;
  end;
run;

data _null_;
  length name $30;
  file print ods=
    (template= 'level1.level2.level3.showbal');
```

```
set work.balances;
name = trim(lname) || ', ' || fname;
put _ods_;
run;
```

There are several items to note in this code.

- The template name can have as many parts as desired. In this case, the template store holds a folder Level1, containing folder Level2, containing folder Level3, containing the template showbal.
- By default, the Header statement defines a header which spans the entire page (in other words, a title).
- The Panels statement creates a multi-panel report.
- The ODS option on the File statement in the data step refers to the template by the same number of levels it has in the Template procedure code.
- The _ods_ keyword on the Put statement requests the output buffer to be formatted in accord with the instructions in the ODS template.
- The ODS template also controls the overall page formatting as well as the individual formatting of data values.

CONCLUSION

Why is there so much excitement around the Output Delivery System?

- Output management is universal. SAS procedures, in Version 7 and later, produce the values while ODS handles the display.
- ODS provides integrated support for output destined for the Web.
- Results of SAS procedures are available in maximum precision.

The challenge is to reorient your thinking. If you choose, you can take control of the style of your output and reports.

REFERENCES

SAS Institute Inc. (1998) *SAS OnlineDoc™, Version 7-0*, Cary, NC: SAS Institute Inc. ODS documentation is contained in the *SAS Procedures Guide*; Concepts; Fundamental Concepts for Using Base SAS Procedures; Language Concepts.

Access to the SAS sample library is available from within a Version 7 display manager session, or directly from your web browser. In both cases, you connect to the SAS Institute web site.

From display manager,

help → SAS on the Web → Sample Programs

From web browser,

http://www.sas.com/service/techsup/sample_library_html

TRADEMARK INFORMATION

SAS, and SAS OnlineDoc are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION

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

William Fehlner
 SAS Institute (Canada) Inc.
 BCE Place, 181 Bay Street, Suite 2220
 P. O. Box 819
 Toronto, Ontario M5J 2T3
 Work Phone: (416) 307-4513
 Fax: (416) 363-5399
 e-mail: CANWRF@can.sas.com