

# Output Delivery System (ODS) Tips and Techniques

Kirk Paul Lafler, Software Intelligence Corporation, Spring Valley, California

## Abstract

Are you looking for ways to improve or enhance the appearance of your SAS<sup>®</sup> output? Output Delivery System (ODS) can help turn tired-looking output into great-looking information that has a purpose. Gone are the days when the only available formatting choice was the boring output listing that contained lifeless monospace fonts. ODS introduces exciting features that improve the appearance of your output. Using built-in format engines, ODS provides SAS users with a powerhouse of exciting capabilities to produce publishing-quality output.

This presentation provides an introduction to ODS, including basic operation and syntax; how specialized output can be created, including RTF, PDF, HTML, Microsoft Excel spreadsheets, and SAS data sets; and how selection (or exclusion) lists can be constructed to handle basic content customizations.

## Introduction

Technology is moving at an astounding pace. With many new features and enhancements introduced in Version 8, users can finally sound the trumpets for what could be the most exciting time for output delivery since the introduction of the color monitor and printer. Never again will SAS users be confined to boring monospace output. Instead, output delivery has entered a new age taking full advantage of font characteristics, color, a variety of output layouts, and numerous other features. This paper and presentation will introduce users to the many new and exciting features found in ODS. Sprinkled throughout will be numerous tips, tricks, and techniques that will, hopefully, make the learning process a little easier.

Attendees will learn numerous tips, tricks, and techniques in handling output including the advantages / disadvantages with traditional SAS output; advantages of ODS; how ODS handles raw data; how ODS combines raw data with table definitions; how to open, close, and manage output destinations; how the Results window stores links to ODS output; how selection and exclusion lists are used; how to create output data sets, RTF files, HTML output, and postscript files; and how ODS is integrated with the DATA step.

## Output Delivery Basics

The SAS<sup>®</sup> Output Delivery System (**ODS**) provides many ways to format output. It controls the way output is accessed and formatted. Although ODS continues to support the creation of traditional SAS listing or monospace output (i.e., Listing), it provides many new features and greater flexibility when working with output.

An early version of ODS appeared in Version 6.12 as a way to address the inherent weaknesses found in traditional SAS output. It enables “quality” looking output to be produced without having to import it into word processors such as MS-Word. In Version 8 and 9, many new output formatting features and options are introduced for SAS users to take advantage of. Users have a powerful and easy way to create and access formatted procedure and DATA step output.

### **Tip #1 – ODS and “Batch” Use**

Many of the ODS features found in the interactive side of the SAS Display Manager System (DMS) can also be used in batch processing. ODS has been designed to make exciting new formatting options available to users. In a windowing environment, ODS can send output to the following destinations: the output window (DMS), the listing file, HTML, SAS dataset, rich text format (RTF), postscript file, external output file (non-SAS file), or output device. The only exception for batch processing is having output sent to the output window.

### **Tip #2 – What if I’m Still Not Using the latest Version of SAS Software**

First introduced in Version 6.12, ODS offered users the capability to format output to destinations other than traditional line printers. Version 6.12 introduced the ability to deploy output to the web, the creation of SAS datasets and rich text format (RTF) files, and DATA step interaction. ODS was designed to address the inherent weaknesses found in traditional SAS output. It enables the creation of “quality” looking output without having to import it into word processors such as MS-Word. New output enhancements were introduced in Version 8 and then in Version 9, including the ability to create postscript files and output customizations. To take full advantage of the power offered in ODS, it is recommended that users upgrade to the latest Version as early as possible to take advantage of these features.

**Tip #3 – ODS and System Resources**

A very important efficiency consideration is to remember that ODS currently supports the following destinations: 1) Listing (Default), 2) HTML, 3) rich text format (RTF), 4) postscript, and 5) Output. (Note: It also provides support in the DATA step.) Each ODS destination can be open or closed at the same time. For each open destination, ODS sends output object(s) to it. System resources are used when a destination is open. As a result, make sure any and all unwanted open destinations are closed to conserve on resources.

**Tip #4 – Closing Destinations before and after use**

The Listing destination is open by default at SAS invocation, while the other destinations are closed. If nothing is done to suppress output to the Listing destination, your SAS programs automatically produce Listing output, just as they always have in the SAS System. If you needed to suppress printed output from being sent to the Listing destination (or DMS Output window) before the execution of a procedure step, the following ODS statement would be issued:

```
ODS Listing Close;  
Proc univariate data=movies;  
Run;  
ODS Listing;
```

By closing the Listing destination before the procedure code, the SAS System is actually suppressing output to that destination until it is reopened. The preceding example shows that at the end of the procedure step, the Listing destination is reopened by specifying ODS Listing; so output from subsequent steps can be sent to the Listing destination.

**Tip #5 – Deleting Output from the Results Window**

The Results window identifies procedure output that has been produced, providing users with an improved way to manage their output. It is customarily a good thing to remove unwanted output displayed in this window to conserve on system resources. The Results window is opened by specifying the command **ODSRESULTS** on the DMS command line or by selecting **View Results** from the pull-down menu.

To delete procedure output, use the following steps:

1. Select the procedure folder you want to remove.
2. Click the Delete button on the task bar.
3. Select "Yes" to confirm the deletion of the procedure output folder.

## Tracing Output

**Tip #6 – Tracing Procedure Output**

Output producing procedures often create multiple pieces or tables of information. In order to discriminate between the various pieces of information, it is advantageous to know the names assigned to each piece of information. The ability to display the names of individual pieces of information generated on output is referred to as **tracing**. The ODS statement syntax ODS trace ON; causes the SAS System to turn the trace feature on and print results to the SAS Log.

```
ODS Trace ON;  
Proc univariate data=movies;  
Run;  
ODS Trace Off;
```

The trace results display information about the data component, the table definition, and the output object. For example, the trace results display the following output objects to the SAS Log: 1) Moments, 2) BasicMeasures, 3) TestForLocation, 4) Quantiles, and 5) ExtremeObs. A sample trace record containing each output object's name, label, template, and path is displayed for the Univariate procedure. Note that for each output object, the name, label, template, and path is displayed.

```

Output Added:
-----
Name:    Moments
Label:   Moments
Template: base.univariate.Moments
Path:    Univariate.age.Moments
-----

Output Added:
-----
Name:    BasicMeasures
Label:   Basic Measures of Location and Variability
Template: base.univariate.Measures
Path:    Univariate.age.BasicMeasures
-----

Output Added:
-----
Name:    TestsForLocation
Label:   Tests For Location
Template: base.univariate.Location
Path:    Univariate.age.TestsForLocation
-----

Output Added:
-----
Name:    Quantiles
Label:   Quantiles
Template: base.univariate.Quantiles
Path:    Univariate.age.Quantiles
-----

Output Added:
-----
Name:    ExtremeObs
Label:   Extreme Observations
Template: base.univariate.ExtObs
Path:    Univariate.age.ExtremeObs
-----

```

## Selecting Output With ODS

A selection or exclusion list exists for each open ODS destination. These lists determine which output objects to send to ODS destinations. To accomplish this, ODS verifies whether an output object is included in a destination's selection or exclusion list. If it does not appear in this list, then the output object is not sent to the ODS destination. If an output object is included in the list, ODS determines if the object is included in the overall list. If it does not appear in this list, then the output object is not sent to the ODS destination. If an output object is included in the overall list then ODS sends it to the selected destination.

### **Tip #7 – Selecting Desired Pieces of Information**

Once you know the individual names of each output object (from the trace), you can then select the desired object for reporting purposes. The syntax is:

**ODS select *output-component-name*;**

where ***output-component-name*** is the name of the desired output object. To select just the output object Moments from the Univariate procedure, the following syntax is specified:

```

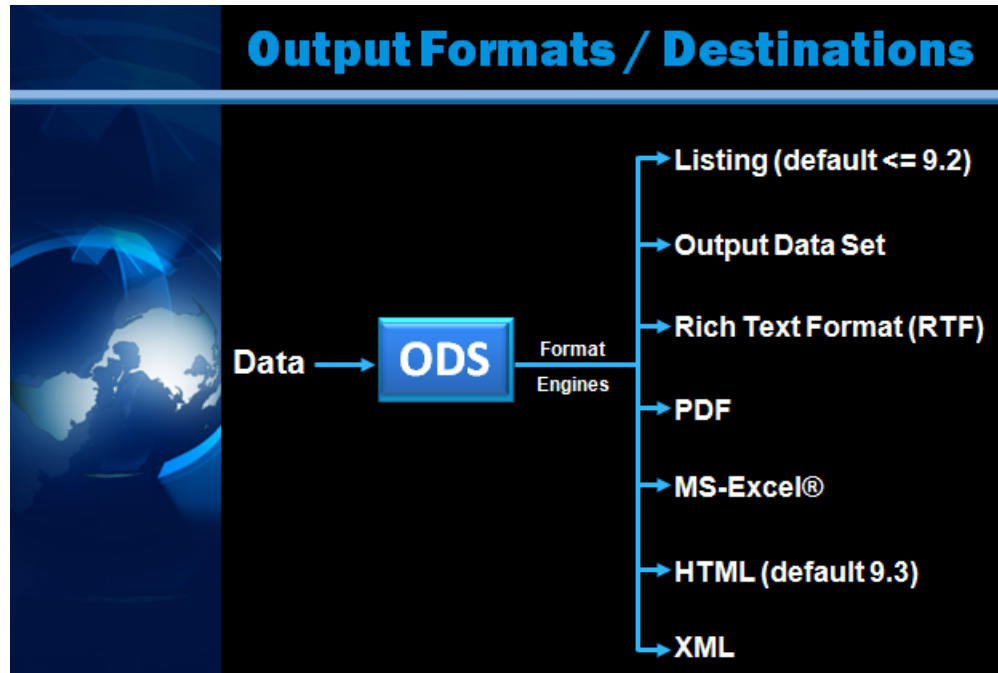
ODS Select Moments;
Proc univariate data=movies;
Run;

```

## Creating Unique Output With ODS

Output Delivery System (ODS) can be used to create a variety of output formats. ODS statements are classified as global statements and are processed immediately by the SAS System. ODS statement options control what format engine(s) are turned on and in effect during the step or until another ODS statement is specified.

ODS has built-in format engines (e.g., Listing, Output, RTF, PDF, DATA Step, HTML, and XML). Specifying an ODS statement and destination at a particular point in a program is important, because output-producing PROC and DATA steps will respond by sending output to the open destination.



## Creating SAS Output Data Sets

### **Tip #8 – SAS Output Data Sets**

Occasionally, output results are needed in a SAS data set rather than in printed form such as the Listing destination. Redirecting SAS procedure output to a data set is relatively simple with ODS. The syntax is:

```
ODS Output output-table-name =
           user-defined-table-name;
< SAS Code >
```

where *output-table-name* is the name of the desired output table (component) containing the information you want written to a data set, such as *Moments* in the UNIVARIATE procedure. *User-defined-table-name* is the name you supply for the newly created data set. It can be defined as either a temporary or permanent data set. Once an object is selected, specify the object in the ODS OUTPUT statement. For example, the Moments from the Univariate procedure is selected and output to a SAS data set in the following code.

```
ODS Listing Close;
ODS Output Moments = Movie_Moments;
Proc univariate data=movies;
Run;
ODS Listing;
```

When the OUTPUT destination is no longer needed, it can be closed with the following ODS statement:

```
ODS OUTPUT CLOSE;
```

## Creating Rich Text Format (RTF) Output

### **Tip #9 – Creating Rich Text Format (RTF) with ODS**

Rich text format (RTF) is text consisting of formatting attributes codes, such as boldface, italics, underline, etc. It is principally used to encapsulate text and formatting attributes during copy-and-paste operations. Because word-processing programs use RTF rather than ASCII when handling data, the need to reformat is a thing of the past. The syntax to create RTF output is:

**ODS RTF FILE = 'user-specified-file-name';**

where user-specified-file-name references a complete and fully-qualified output location for the creation and storage of the RTF file, data, and codes. For example, the following code creates an RTF file using the Univariate procedure output. (**Note:** The RTF extension is required).

```
ODS Select Moments = moments;
ODS RTF FILE='ods-rtf-univariate.rtf';
Proc univariate data=movies;
  Title1 'Delivering RTF Output';
Run;
ODS RTF Close;
```

The results of the RTF output are displayed below:

**Delivering RTF Output**  
*The UNIVARIATE Procedure*  
**Variable: Year**

Moments			
<b>N</b>	22	<b>Sum Weights</b>	22
<b>Mean</b>	1982.9090 9	<b>Sum Observations</b>	43624
<b>Std Deviation</b>	15.212492	<b>Variance</b>	231.4199 13
<b>Skewness</b>	-2.1106473	<b>Kurtosis</b>	4.418072 09
<b>Uncorrected SS</b>	86507286	<b>Corrected SS</b>	4859.818 18
<b>Coeff Variation</b>	0.7671805 1	<b>Std Error Mean</b>	3.243314 2

## Customizing Page Numbers with Functions

### **Tip #10 – Creating Customized Rich Text Format (RTF) with Functions**

Page numbers can be customized in RTF output by specifying an escape character with an ODS RTF statement and any or all of the following functions:

- ✓ {thispage}
- ✓ {lastpage}
- ✓ {pageof}

In the following code, an escape character is specified with ODS, and the functions: {thispage}, {lastpage}, and {pageof} are specified in the title and footnote statements.

```
ods Listing close;
ods escapechar='^';
ods RTF file='c:\univariate-output.rtf';
proc univariate data=movies;
  title 'Page ^{thispage} of ^{lastpage}';
  footnote '^{pageof}';
run;
ods RTF close;
ods Listing;
```

The results of the RTF output are displayed below:

The screenshot displays the following tables:

**Moments**

		Sum Weights	
N	22	Sum Observations	2748
Mean	124.909091	Variance	667.419913
Std Deviation	25.8344714	Kurtosis	5.51199999
Skewness	1.45414494	Corrected SS	337.266
Uncorrected SS	337.266	Std Error Mean	20.682619
Coeff Variation	20.682619		

**Basic Statistical Measures**

Location	Variability	
Mean	124.9091	Std Deviation 25.83447
Median	119.0000	Variance 667.41991
Mode	127.0000	Range 97.00000
		Interquartile Range 24.00000

**Tests for Location: Mu0=0**

Test	Statistic	p Value
Student's t	t 22.67806	Pr >  t  <.0001
Sign	M 11	Pr >=  M  <.0001
Signed Rank	S 126.5	Pr >=  S  <.0001

**Quantiles (Definition 5)**

Quantile	Estimate
100% Max	194
99%	194
95%	177
90%	170
75% Q3	130
50% Median	119
25% Q1	106
10%	101
5%	98

**Tip #11 – Creating and Naming Postscript Files**

ODS enables the creation of a postscript file from SAS output to help ensure that formatting and content is preserved while maintaining complete printer independence. The next example shows the creation of a postscript file from Print procedure output.

```
ODS Listing Close;
ODS Printer Postscript;
Proc print data=movies noobs n;
  Title1 'Example Creating a Postscript File';
Run;
ODS Printer Close;
ODS Listing;
```

Since the postscript file was created without assigning a filename, ODS assigns the name: **SASPRT.PS**. Frequently, a user-defined filename is desired rather than the default name. In these cases, the ODS statement and **File=** option can be used to assign a name to the postscript file.

```
ODS Listing Close;
ODS Printer Postscript File='Beginning_tutorials.ps';
Proc print data=movies noobs n;
  Title1 'Example Creating a Postscript File';
Run;
ODS Printer Close;
ODS Listing;
```

**Tip #12 – The Custom Reporting Interface – ODS and the DATA Step**

To provide greater capabilities when working with custom output created in the DATA step, ODS is integrated into the DATA step. Two new options are necessary to take advantage of ODS: 1) the ODS option in the FILE statement and 2) the **\_ODS\_** option in the PUT statement. These two options are used to direct the results of a DATA step to ODS. For example, the following code directs output from a DATA step through the RTF format engine to create a RTF file.

```
ODS RTF FILE='ods-DATA-step.rtf';
Title1 'Great Movies';
Data NULL;
Set movies;
File print ods;
Put _ods_;
Run;
ODS RTF Close;
```

The results from the ODS RTF file specification in the DATA step code is illustrated below.



Title	Length	Category	Year	Studio	Rating
Brave Heart	177	Action Adventure	1995	Paramount Pictures	R
Casablanca	103	Drama	1942	MGM / UA	PG
Christmas Vacation	97	Comedy	1989	Warner Brothers	PG-13
Coming to America	116	Comedy	1988	Paramount Pictures	R
Dracula	130	Horror	1993	Columbia TriStar	R
Dressed to Kill	105	Drama Mysteries	1980	Filmways Pictures	R
Forrest Gump	142	Drama	1994	Paramount Pictures	PG-13
Ghost	127	Drama Romance	1990	Paramount Pictures	PG-13
Jaws	125	Action Adventure	1975	Universal Studios	PG
Jurassic Park	127	Action	1993	Universal Pictures	PG-13
Lethal Weapon	110	Action Cops & Robber	1987	Warner Brothers	R
Michael	106	Drama	1997	Warner Brothers	PG-13
National Lampoon's Vacation	98	Comedy	1983	Warner Brothers	PG-13
Poltergeist	115	Horror	1982	MGM / UA	PG
Rocky	120	Action Adventure	1976	MGM / UA	PG

## Sending Data / Output / Results to Excel Spreadsheet Files

### **Tip #13 – Several Approaches Exist to Create Excel Spreadsheet Files from SAS Data, Output and Results**

There are several ways to send data / output / results to Excel. Four approaches will be presented:

- ✓ Using SAS Display Manager (with a right-click)
- ✓ Using Output Delivery System (ODS)
- ✓ Using the EXPORT procedure
- ✓ Using a LIBNAME statement

#### **Using SAS Display Manager**

SAS 9 permits the creation of an Excel file using SAS Display Manager. To create an Excel file in this way, follow these steps:

- ✓ Select the desired SAS data set
- ✓ Right-click on the selected data set name
- ✓ From the selection list, select 'View in Excel'
- ✓ Once data is transferred to Excel, save the Excel file to the desired location

#### **Using ODS in a PROC Step**

SAS 9 permits the creation of an Excel file using ODS. For example, the XLS extension is specified in the file name of the ODS statement with PROC PRINT.

```
ods Listing close;
ods HTML FILE='c:\Movies-PROCPRINT.XLS';
proc print data=movies noobs;
  title;
run;
ods HTML close;
ods Listing;
```



**Using ODS in a DATA Step**

An Excel file can be created with ODS in a DATA step. For example, the XLS extension is specified in the file name of the ODS statement with the DATA step follows.

```
ods Listing close;
ods HTML FILE='c:\Movies-DATAstep.XLS';
  data _null_;
    set movies;
    file print ods;
    put _ods_;
  run;
ods HTML close;
ods Listing;
```

**Using the EXPORT Procedure**

Exporting data to Excel with the EXPORT procedure is an effective way to transfer data. For example, exporting “Comedy” movies from the Movies data set to Excel with the PROC EXPORT statement and WHERE= data set option is illustrated:

```
PROC EXPORT DATA=movies
  (WHERE=(category="Comedy" ) )
  OUTFILE='c:\Movies-Export.xls'
  DBMS=EXCEL;
RUN;
```

**Using a LIBNAME Statement**

Beginning in SAS 9, a LIBNAME statement can be used to export a data set to an Excel file. For example, to export the data in the Movies data set to an Excel 97 spreadsheet, the following LIBNAME statement is used:

```
LIBNAME MYXLS PCFILES PATH="c:\Movies-Libname.XLS";
DATA MYXLS.Movies;
  SET Movies;
RUN;
```

**Output Delivery Goes Web**

The Web offers incredible potential that impacts all corners of society. With its increasing popularity as a communications medium, Web publishers have arguably established the Web as the greatest medium ever created. Businesses, government agencies, professional associations, schools, libraries, research agencies, and a potpourri of society’s true believers have endorsed the Web as an efficient means of conveying their messages to the world.

The Web is not a static environment, but a dynamic medium capable of distributing content anywhere and at anytime. The 24/7 model permits information to be refreshed and updated continuously as new material becomes available. A primary objective of Web publishers everywhere is to engage visitors with timely and interesting content that brings them back for more.

**Tip #14 – Pagesize / Linesize Settings**

The Options PS= and LS= have no effect when used with the HTML destination (opposed to most other output-producing steps that generate output to a print destination). If the PS= and/or LS= options are used with the HTML destination, they are simply ignored. The SAS System creates a type of “streaming” or continuous output and adds elevator bars (horizontal and/or vertical) for easy navigation.

The SAS System does provide a way for users to paginate through output displayed in a body file. The HTML destination provides a way to designate an optional description of each page of the body file. The PAGE= file (when specified) recognizes each new page of output produced by ODS. What ODS does is create a section called **Table of Pages** containing links to the body file for easy navigation through output.

**Tip #15 – Deploying Output to the Web**

With the popularity of the Internet, you may find it useful to deploy selected pieces of output on your web site. ODS makes deploying procedure output to the web a simple process. Syntactically correct HTML code is automatically produced and made ready for deployment using one of the Internet browser software products (e.g., Internet Explorer, Netscape Navigator, etc.). As a result, the SAS System and the HTML destination create a type of “streaming” or continuous output by adding elevator bars (horizontal and/or vertical) for easy navigation.

**Tip #16 – Creating HTML Destination Files with ODS**

Four types of files can be created with the ODS HTML destination: 1) body, 2) contents, 3) page, and 4) frame. Each file is described below.

The **Body** file contains the results from the procedure embedded in ODS-generated HTML code. Horizontal and vertical scroll bars are automatically placed on the generated page, if necessary.

The **Contents** file consists of a link to each HTML table within the body file. It uses an anchor tag to link to each table. By using your browser software, you can view the contents file directly or as part of the frame file.

The **Page** file consists of a link to each page of ODS created output. By using your browser, you can view the page file directly or as part of the frame file.

The **Frame** file displays the body file and the contents file, the page file, or both. The next example shows the creation of Web-ready Univariate procedure output using the HTML format engine with the body=, contents=, page=, and frame= options.

```
ODS Listing Close;
ODS HTML body='ods-body.htm'
      contents='ods-contents.htm'
      page='ods-page.htm'
      frame='ods-frame.htm';
proc univariate data=movies;
  Title1 'Creating HTML Output with ODS';
run;
ODS HTML Close;
ODS Listing;
```

A snippet of the HTML output appears below.

The screenshot displays the HTML output generated by SAS. On the left, there are two navigation panels: 'Table of Contents' and 'Table of Pages'. The 'Table of Contents' panel lists '1. The Univariate Procedure' with sub-links for 'Length', 'Moments', 'Basic Measures of Location and Variability', and 'Tests For Location'. The 'Table of Pages' panel lists '1. The Univariate Procedure' with sub-links for 'Page 1' and 'Page 2'. The main content area has a title 'Creating HTML Output with ODS' and a subtitle 'HTML FRAME File containing UNIVARIATE Ou'. Below the title is the text 'The UNIVARIATE Procedure Variable: Length'. A table titled 'Moments' is displayed, showing various statistical measures for the variable Length.

Moments			
<b>N</b>	22	<b>Sum Weights</b>	
<b>Mean</b>	124.909091	<b>Sum Observations</b>	27
<b>Std Deviation</b>	25.8344714	<b>Variance</b>	667.4199
<b>Skewness</b>	1.45414494	<b>Kurtosis</b>	1.714538
<b>Uncorrected SS</b>	357266	<b>Corrected SS</b>	14015.81
<b>Coeff Variation</b>	20.682619	<b>Std Error Mean</b>	5.507927

**Tip #17 – Creating PDF Output**

To share output electronically, Adobe provides user with a proprietary format called PDF. The objective of PDF is to enable the printing of output exactly as it is seen. The significance of PDF output is that it is a great way to format output for Web deployment since it is completely independent of any printer destination. Essentially, PDF output is displayed and printed the way its creator intended. To create PDF output from the results of the UNIVARIATE procedure, the ODS PDF option can be specified as follows.

**Code**

```
ODS Listing Close;
ODS PDF FILE='ods-univariate.pdf';
proc univariate data=movies;
  title1 'Creating PDF Output with ODS';
run;
ODS PDF Close;
ODS Listing;
```

**Tip #18 – Combining Output Results**

With the streaming capabilities of HTML output, results can be combined so they appear on the same screen (or page). Rather than having output controlled by one or more page breaks, HTML automatically displays output without page boundaries. The following example code illustrates combined output from the PRINT and MEANS procedures.

**Code**

```
ods html path='c:\sas app'
  body='ods-body-combined.html'
  contents='ods-contents-combined.html'
  page='ods-page-combined.html'
  frame='ods-frame-combined.html';
proc print data=movies noobs n;
  title1 'Movie Classics Listing';
  where rating in ('G', 'PG');
run;
proc means data=movies;
  title1 'Summary of Movie Classics';
  class rating;
run;
ods html close;
```

The results from combining output appear below.

**Output from Combining Output**

Table of Contents	
1	The Print Procedure Data Set ODS18.MOVIES
2	The Means Procedure Summary statistics
Table of Pages	
1	The Print Procedure Page 1
2	The Means Procedure Page 2

Movie Classics Listing					
Title	Length	Category	Year	Studio	Rating
The Wizard of Oz	101	Adventure	1939	MGM / UA	G
Casablanca	103	Drama	1942	MGM / UA	PG
Jaws	125	Action Adventure	1975	Universal Studios	PG
Rocky	120	Action Adventure	1976	MGM / UA	PG
Star Wars	124	Action Sci-Fi	1977	Lucas Film Ltd	PG
Poltergeist	115	Horror	1982	MGM / UA	PG
The Hunt for Red October	135	Action Adventure	1989	Paramount Pictures	PG
N = 7					

Summary of Movie Classics							
The MEANS Procedure							
Rating	N Obs	Variable	N	Mean	Std Dev	Minimum	Maximum
G	1	Length	1	101.0000000		101.0000000	101.0000000
	1	Year	1	1939.00		1939.00	1939.00
PG	6	Length	6	120.3333333	10.7641380	103.0000000	135.0000000
	6	Year	6	1973.50	16.2818918	1942.00	1989.00
PG-13	7	Length	7	127.2857143	33.9004144	97.0000000	194.0000000
	7	Year	7	1991.86	4.9809160	1983.00	1997.00
R	8	Length	8	129.2500000	28.4190580	105.0000000	177.0000000
	8	Year	8	1987.63	5.1806646	1980.00	1995.00

**Tip #19 – Testing Web Output**

Before transferring your Web files to a Web server, users should thoroughly test any HTML code to make sure they are problem-free. Many viewers will not return to Web pages that contain errors or do not work according to design. A word of caution: **Not all Web browser software handles web pages the same way.** Microsoft Internet Explorer may display your web pages differently than Netscape Navigator and others.

Before deploying web-based output to the Web or Intranet, it is important to visually inspect and test your output to see how your Web pages behave. Web validation services are available for users to check web pages for errors or inconsistencies. The following recommendations should be considered before deploying Web output.

1. Check spelling – check the spelling on each of your Web pages before making them available.
2. If possible, use a validation service to identify errors in your use of HTML.
3. Test the Web pages to see how easy they are to access and browse through the information. You should verify that each Web page has a consistent design and layout.
4. Turn off images to test how Web pages will look and what information is displayed when viewers use Web browsers that cannot display in pages or when they turn off images.
5. Verify links to make sure each link takes you to the intended destination and that each link contains information of interest to your viewers.
6. Enlist a test audience to check out your Web pages and to solicit their feedback. This feedback is very important since it enables you to improve the way your Web pages look and operate. It is also important that you compare your test audience's feedback with your own objectives to determine which areas require more work.
7. Test your Web pages with different Web browser software to evaluate how they will look. The two most popular Web browsers are Microsoft Internet Explorer and Netscape Navigator.
8. Test Web pages on different computers because they can look and sound differently when the content consists of animation.
9. Determine the speed of transferring Web page content. If the content is too text-rich or image-rich, the excessive transfer speeds may cause viewers to tune-out rather than tune-in.
10. View your Web pages at different resolutions to determine the amount of information a monitor can display.

**Building Drill-Down Applications**

As a general rule, the best type of user interface design for transaction-based applications is a drill-down user interface opposed to a character-based one. It's referred to as drill-down because a user drills down through the data, layer by layer, until the desired information is found.

The key to building successful drill-down applications requires systems analysts and system designers to understand what users are trying to achieve with the data. These individuals must recognize the tasks users engage in while trying to access the desired information. These tasks are then translated into a series of selection criteria that users should be able to select from.

**Tip #20 – Building a Drill-down Application**

To simplify the process of building a graphical drill-down application in the SAS System, five easy steps are presented.

1. Create a data set containing the location of the HTML link variable.
2. Create driver program using the HTML path with BODY and other optional files (e.g., CONTENTS=, PAGE=, and FRAME= files).
3. Create graph using HTML= option and link variable.
4. Create detail list drill-down pages.

5. Use Web browser (e.g., Internet Explorer, Firefox, Navigator, Safari, etc.) to navigate through resulting application.

### **Example Drill-down Application**

In the following drill-down application, a pie chart is used to display summary information on G-rated and PG-rated movies. A horizontal or vertical bar chart could also be used as the graphical user interface. To display detailed information on a movie category, a user would only need to click on the desired piece of the pie chart that they had interest in, as shown in the illustration below. Control would then be passed, via hyperlinks, to the underlying detail output created in step #4 above.

### **Example Code**

The SAS coding steps used in building a drill-down application under the Windows® platform is presented below.

```
*STEP 1 - Define data set and link variable;
data movie_ratings_g_pg;
  set movies;
  length urlink $30.;
  if upcase(rating) = "G" then urlink="href=g-ratings-body.html";
  else if upcase(rating) = "PG" then urlink="href=pg-ratings-body.html";
run;

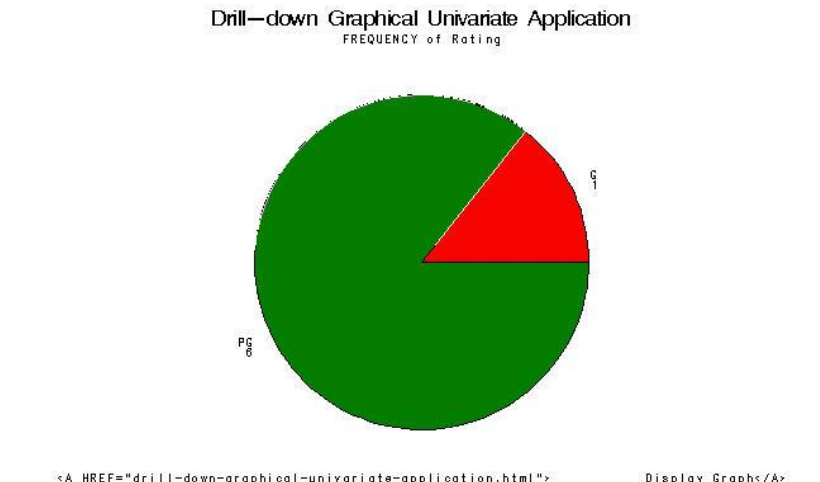
*STEP 2 - Create driver program using HTML path and BODY file;
ods html path='c:\sas app'
  body='drill-down-graphical-application.html';

*STEP 3 - Create graph using HTML= option;
goptions device=gif hsize=7in vsize=4in;
proc gchart data=movie_ratings_g_pg;
  title1 "Drill-down Graphical Application";
  pie rating / html=urlink;
  where upcase(rating) in ('G', 'PG');
run;
quit;

*STEP 4a - Create G-rated drill-down list;
ods HTML path='c:\sas app'
  body='g-ratings-body.html';
proc univariate data = movie_ratings_g_pg ;
  title1 'Creating a Drill-down Application';
  footnote1 '<A HREF="drill-down-graphical-application.html"> Display Graph</A>';
  where upcase(rating)="G";
run;
ods HTML close;

*STEP 4b - Create PG-rated drill-down list;
ods HTML path='c:\sas app'
  body='pg-ratings-body.html';
proc univariate data = movie_ratings_g_pg ;
  title1 'Creating a Drill-down Application';
  footnote1 '<A HREF="drill-down-graphical-application.html"> Display Graph</A>';
  where upcase(rating)="PG";
run;
ods HTML close;
ods Listing;
```

The drill-down results appear below.



## Conclusion

The time has come to turn tired-looking output listings into great looking information. The SAS<sup>®</sup> Output Delivery System (**ODS**) provides new and exciting ways to improve the way output is generated. Gone are the days when the only available formatting choice for output was basic output listings consisting of monospace fonts printed on a white background.

ODS' built-in format engines addresses the inherent weaknesses found in traditional SAS output by enabling "quality" output to be produced without the need of importing it into word processors. The latest version boasts many new format destinations including Postscript, RTF, PDF, HTML, SAS data sets, and DATA step interfaces, while still supporting traditional Listing output. ODS offers users a wonderful opportunity to produce quality and exciting output and results.

## References

- Gupta, Sunil Kumar (2003), Quick Results with the Output Delivery System (Art Carpenter's SAS Software), SAS Institute Inc., Cary, NC, USA.
- Haworth, Lauren, Cynthia L. Zender, and Michele Burlew (2009), Output Delivery System: The Basics and Beyond, SAS Institute Inc., Cary, NC, USA.
- Heffner, William F. (1998), "ODS: The DATA Step Knows," Proceedings of the 23<sup>rd</sup> Annual SAS Users Group International Conference, Cary, NC: SAS Institute Inc.
- Lafler, Kirk Paul (2012), "Quick Results with Output Delivery System (ODS)," Proceedings of the North East SAS Users Group (NESUG) 2012 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.
- Lafler, Kirk Paul (2011), "Quick Results with Output Delivery System (ODS)," Proceedings of the MidWest SAS Users Group (MWSUG) 2011 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.
- Lafler, Kirk Paul (2010), "Output Delivery System (ODS) – Simply the Basics," Proceedings of the MidWest SAS Users Group (MWSUG) 2010 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.
- Lafler, Kirk Paul (2009), "Exploring Output Delivery System Programming Techniques," Proceedings of the Western Users of SAS Software (WUSS) 2009 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.
- Lafler, Kirk Paul (2008), "Output Delivery System Tips and Techniques," San Diego SAS Users Group (SANDS) 1-Day Conference (February 21<sup>st</sup>, 2008), Software Intelligence Corporation, Spring Valley, CA, USA.
- Lafler, Kirk Paul (2007), "Output Delivery System Tips and Techniques," Proceedings of the MidWest SAS Users Group (MWSUG) 2007 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.
- Lafler, Kirk Paul (2006), "Output Delivery System Tips and Techniques," Proceedings of the South Central SAS Users Group (SCSUG) 2006 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.
- Lafler, Kirk Paul (2004), "Creating HTML Output with Output Delivery System," Proceedings of the Western Users of SAS Software (WUSS) 2004 Conference, Software Intelligence Corporation, Spring Valley, CA, USA.
- Lafler, Kirk Paul (2004), "Output Delivery Tips, Tricks, and Techniques," Proceedings of the 2004 North Texas SAS Users Group, Software Intelligence Corporation, Spring Valley, CA, USA.

- Lafler, Kirk Paul (2002), "Output Delivery Goes Web," Proceedings of the 27<sup>th</sup> Annual SAS Users Group International Conference, Software Intelligence Corporation, Spring Valley, CA, USA.
- Lafler, Kirk Paul (2000), The SAS<sup>®</sup> Output Delivery System (ODS) Answer Guide, Revised and Updated, Software Intelligence Corporation.
- Lafler, Kirk Paul (1999), "Delivering Results with the Output Delivery System," Proceedings of the 24<sup>th</sup> Annual SAS Users Group International Conference.
- Olinger, Christopher R. (1998), "ODS for Data Analysis: Output As-You-Like-It in Version 7," Proceedings of the 23rd Annual SAS Users Group International Conference, Cary, NC: SAS Institute Inc.
- Patel, Himesh (1998), "Using SAS/GRAPH<sup>®</sup> Software to Create Graphs on the Web," Proceedings of the 23rd Annual SAS Users Group International Conference, Cary, NC: SAS Institute Inc.
- SAS Institute Inc. (1999), The Complete Guide to the SAS<sup>®</sup> Output Delivery System, Version 7-1, Cary, NC, USA.
- Wehr, Paul (1998), "Building Clinical Information Spaces on the World Wide Web," Proceedings of the 23rd Annual SAS Users Group International Conference, Ann Arbor, MI: STATPROBE, Inc.

## Acknowledgments

The author thanks Alejandro Farias and John Taylor, SCSUG 2014 Conference Chairs, for accepting my abstract and paper; as well as SAS Institute Inc.; and the SCSUG Executive Committee for organizing a great conference!

## Trademark Citations

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

## Author Information

Kirk Paul Lafler is consultant and founder of Software Intelligence Corporation and has been using SAS since 1979. He is a SAS Certified Professional, provider of IT consulting services, trainer to SAS users around the world, and sasCommunity.org emeritus Advisory Board member. As the author of six books including Google<sup>®</sup> Search Complete (Odyssey Press. 2014); PROC SQL: Beyond the Basics Using SAS, Second Edition (SAS Press. 2013); PROC SQL: Beyond the Basics Using SAS (SAS Press. 2004); Kirk has written more than five hundred papers and articles, been an Invited speaker and trainer at four hundred-plus SAS International, regional, special-interest, local, and in-house user group conferences and meetings, and is the recipient of 23 "Best" contributed paper, hands-on workshop (HOW), and poster awards.

Comments and suggestions can be sent to:

Kirk Paul Lafler  
Senior SAS<sup>®</sup> Consultant, Application Developer, Data Scientist, Trainer and Author  
Software Intelligence Corporation  
E-mail: [KirkLafler@cs.com](mailto:KirkLafler@cs.com)  
LinkedIn: <http://www.linkedin.com/in/KirkPaulLafler>  
Twitter: @sasNerd