

Output Delivery Goes Web

Kirk Paul Lafler, Software Intelligence Corporation

Abstract

In an age of Internet frenzy, output delivery is experiencing incredible interest around the globe. The explosion of Internet technologies, especially electronic delivery of data, is on the rise. As digital commerce forces fundamental changes in the nature of business, IT is at the center of these changes. Many organizations are moving toward a customer self-service center model, where information access will be key to survival. As the growth and popularity of the Web surges to unprecedented levels, SAS® users are beginning to look for ways to transform basic monospace output into an electronic world of glitz and color. This advanced tutorial will show attendees how to convert tired and old output listings into beautiful Web pages, combine output results, create PDF output, change Web page labels, build drill-down Web applications, and perform a variety of output customizations. SAS users will see how ODS works with base-SAS software, SAS/GRAPH software, Microsoft Explorer, and HTML to create exciting Web output.

Introduction

The SAS® System Output Delivery System (ODS) controls the formatting of all output objects. It has many great features and provides flexibility when working with output since each output is split into two component objects: ① a *data* component and ② a *template* component. The data object contains the raw data values for each piece of output while the template object contains how the piece should look.

Prior to ODS, SAS users were accustomed to using the basic output available with each procedure or with using techniques of merging SAS output into document editors such as MS-Word. The advantages for using default procedure print formats are that they are familiar to users and easy to use. Beyond that, there is not much motivation for using them, especially when incorporating procedure output into a formatted document.

Inherent limitations exist with listing files: ① monospace fonts are used; ② each output line is rendered at a time; ③ inability to control formatting; and ④ difficult to create HTML code for web deployment. This paper highlights several areas where ODS can turn basic monospace SAS® Listing output into formatted output viewable by all on the Web, Intranet, or Extranet.

The Web as a Dynamic Medium

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 24/7/365 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.

Output Delivery in a Multidimensional World

The availability and demand for information is greater today than at any other time in our history. Each day we are confronted with information about facts, opinions, opinions about facts, facts about opinions, commentary, and not to mention the endless sales pitches in a variety of mediums including print, radio, television, Internet, and countless others. It seems that we cannot get enough information to satisfy this hunger for wanting to know more.

The fact is that there are huge demands for information and there is not any letup in sight. What this means is that an organization's information resources are one of its most valuable assets. To take full advantage of these assets an organization must be able to utilize its information resources effectively to deliver that information to a wide variety of information channels.

Answering the IT Challenge

The ability to take advantage of information resources is a critical IT challenge faced by an organization's customers. Information access and delivery has been greatly improved in Version 8 with ODS. Output can be delivered to print, the Internet, an organization's intranet, as well as other mediums, formats, and information channels. ODS and its new web-based HTML publishing features makes the SAS System a potent tool for effective information delivery by being able to integrate information resources with visual representations.

Another important challenge in Version 8 was to standardize the way the DATA step and each output-producing procedure creates output. This was accomplished by removing the responsibility for formatting output from the DATA step and individual procedures. Major modifications were required so the DATA step and output-producing procedures would only supply the raw data and the name of the table definition containing the formatting instructions. With this accomplished, ODS does the rest by formatting the output to meet the specific needs of the user.

Output Destinations

ODS has been designed to manage and route output to several output destinations: ① Standard SAS listing, ② HTML code for web deployment, ③ Output data set, ④ Rich Text Format (RTF), or ⑤ PDF. By using global ODS statements, output is formatted and routed with simple precision to one or more of these output destinations. ODS improves the way system resources are utilized by permitting output to be routed to one or more output destinations at the same time.

Before output can be routed, users first open the desired output destination(s) by specifying one or more ODS statements and corresponding options. To prevent output from being routed to an undesired destination, one or more ODS statements must also be issued to close open destination(s). Once this is done, output is generated by the specified SAS procedure or DATA step. The syntax for the ODS statement follows:

ODS <output destination>;

Where, as mentioned earlier, <output destination> can be: ① listing, ② rtf, ③ html, ④ output, or ⑤ pdf.

Tracing ODS Output Objects

Output producing procedures often create multiple pieces or tables of information. At times, it may be useful to know the names assigned to each piece of information. By knowing the names of each piece of output, you can better control what tables of information to display in output. The ODS statement syntax to use for procedure code is:

```
ODS trace output;
< SAS procedure code >
ODS trace off;
```

For example, the following code traces the individual objects produced by the Univariate procedure:

```
ODS Trace ON / Listing;
Proc univariate data=libref.movies;
Run;
ODS Trace Off;
```

The trace record displays information about the data component, the table definition, and the output object. For example, the trace record displays the following

output objects to the SAS Listing destination: 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
```

ODS and Natural Selection

A selection or exclusion list is created for each 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 it 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 it is included in the overall list, then ODS sends the object to the destination.

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

ODS select output-object-name;

where **output-object-name** is the name of the desired output object. The following ODS statement selects the output object **Moments** from the Univariate procedure only:

```
ODS Select Moments;
Proc univariate data=libref.movies;
Run;
```

HTML Destination Files

Four types of files are produced with the ODS HTML destination: 1) body, 2) contents, 3) page, and 4) frame. Output is written to one or more of these files by substituting the specific option and filename in the ODS HTML statement. A brief explanation will be presented to describe the purpose of each.

The Body File

The body file consists of output created by your SAS job. When output is routed to the HTML destination, it is placed within one of three HTML tags: 1) TABLE, 2) IMG, or 3) as an HTML table. The nature of the output object determines which of the three tags are used for displaying with a Web browser.

When creating the body file, ODS handles output objects differently depending on the nature of the output. If the output object consists of tabular data without any graphics, ODS inserts the object within TABLE tags. When the object contains a graphic image, it is placed within IMG tags. And when the object does not contain tabular data or any graphic images, it is tagged as an HTML table.

The Contents File

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

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

The frame file displays the body file and the contents file, the page file, or both.

Links and References in ODS

The ODS statement controls how links and references are constructed between one or more HTML destination files. The basic syntax of the ODS statement follows:

ODS HTML ODS-action;

< or >

ODS HTML HTML-file-specification < options >;

When an ODS-action is specified, one or more output objects are selected or excluded, or the HTML destination is closed. The available ODS-actions are: 1) CLOSE, 2) EXCLUDE, 3) SELECT, and 4) SHOW. When an HTML-file-specification is specified, ODS routes one or more pieces of output to a designated file or files. The available options include:

```
ANCHOR='anchor-name'
BASE='string'
GFOOTNOTE <or> NOGFOOTNOTE
GPATH=file-specification
GTITLE <or> NOGTITLE
HEADTEXT='HTML-document-head'
METATEXT='HTML-document-head'
NEWFILE=NONE <or> OUPUT <or> PAGE <or>
PROC
PATH=file-specification
RECORD_SEPARATOR='string' <or> NONE
STYLE='style-definition'
TRANSTAB='translation-table'
```

Deploying Output to the Web

We are all painfully aware that standard SAS output has many limitations. Figure 1 illustrates standard monospace SAS output from the PRINT procedure.

Movie Buff Listing					
Title	Length	Category	Year	Studio	Rating
Brave Heart	177	Action Adventure	1995	Paramount Pictures	R
Carablanca	102	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 Mystery	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 Cop & Robber	1987	Warner Brothers	R
Michael	106	Drama	1997	Warner Brothers	PG-13
Nat. Lampoon's Vacation	94	Comedy	1982	Warner Brothers	PG-13
Poltzgeist	115	Horror	1982	MGM / UA	PG
Rocky	120	Action Adventure	1976	MGM / UA	R
Scarface	170	Action Cop & Robber	1983	Universal Studios	R
Silence of the Lambs	116	Drama Suspense	1991	Orion	R
Star Wars	124	Action Sci-Fi	1977	Lucas Film Ltd	PG
The Hunt for Red October	135	Action Adventure	1989	Paramount Pictures	PG
The Terminator	108	Action Sci-Fi	1984	Live Entertainment	R
The Wizard of Oz	101	Adventure	1939	MGM / UA	G
Titanic	194	Drama Romance	1997	Paramount Pictures	PG-13

Figure 1. Standard SAS Output

With the popularity of the Internet, you may find it useful to deploy all, or selected pieces of output, on your web site. ODS makes deploying standard SAS output to the web a simple process. Syntactically correct HTML code is automatically produced and ready for you to deploy using your favorite Internet browser software. The ODS statement syntax follows:

```
ODS html body = 'user-defined-file-name' ;
< SAS procedure code >;
ODS html close;
```

Creating PDF Output

To share output electronically, Adobe created 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

format for Web deployment since it is completely independent of any printer destination. To create PDF output, the ODS PDF option is specified as follows.

Code

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

Putting It All Together

The SAS System and the ODS HTML destination create a type of "streaming" or continuous output by adding elevator bars (horizontal and/or vertical) for easy navigation. The next illustration shows web-ready SAS output created from the PRINT procedure.

Index	Variable	Type	Value
1	Length	NUM	120
2	Director	CHAR	John Ford
3	Cast	CHAR	John Ford, John Ford
4	Genre	CHAR	Western
5	Year	NUM	1939
6	Box Office	NUM	1,000,000
7	Reviews	CHAR	Excellent
8	Score	NUM	8.5

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.

Code

```
ODS Listing Close;
ODS HTML body='ods-body.htm'
  contents='ods-contents.htm'
  page='ods-page.htm'
  frame='ods-frame.htm';
proc univariate data=libref.movies;
  title1 'Creating HTML Output with ODS';
run;
ODS HTML Close;
ODS Listing;
```

The location of each HTML file created by the previous example code (under the Windows operating system) is:

c:\My Documents\My SAS Files\V8.

The HTML output appears below:

Variable	Type	Value
Length	NUM	120
Director	CHAR	John Ford
Cast	CHAR	John Ford, John Ford
Genre	CHAR	Western
Year	NUM	1939
Box Office	NUM	1,000,000
Reviews	CHAR	Excellent
Score	NUM	8.5

Changing Output Labels

Specifying a PROCLABEL option lets you change the "default" label displayed on procedure output. To change the label produced by the UNIVARIATE procedure from "The Univariate Procedure" to "Movie Classics Statistics", the following code is specified.

Code

```
ods html path='c:\sas app'
  body='ods-body-label.html'
  page='ods-page-label.html'
  contents='ods-contents-label.html'
  frame='ods-frame-label.html';
ods proclabel 'Movie Classics Statistics';
proc univariate data=odslib.movies;
  title1 'Creating HTML Output with ODS';
  title2
    'HTML FRAME File with Changed Labels';
run;
ods html close;
```

Output after Changing Labels

Variable	Type	Value
Length	NUM	120
Director	CHAR	John Ford
Cast	CHAR	John Ford, John Ford
Genre	CHAR	Western
Year	NUM	1939
Box Office	NUM	1,000,000
Reviews	CHAR	Excellent
Score	NUM	8.5

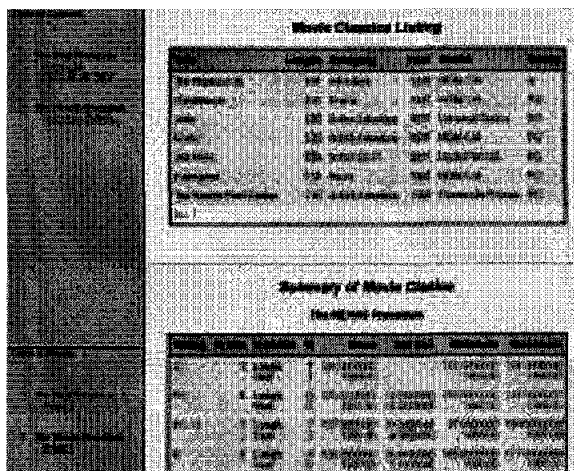
Combining Output Results

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=odslib.movies noobs n;
  title1 'Movie Classics Listing';
  where rating in ('G', 'PG');
run;
proc means data=odslib.movies;
  title1 'Summary of Movie Classics';
  class rating;
run;
ods html close;
```

Output from Combining Output



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.

Two significant design objectives exist for drill-down user interfaces. The first objective is to allow users to get the information they need quickly in order to make good business decisions. The second objective is that only data that is necessary by users is displayed at a time.

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.

Drill-down with Layered Charts

As the amount of data available on the Internet grows exponentially, systems analysts and system designers are faced with huge challenges. First, how should meaningful information be extracted from data? Second, what is the best way to stay current as information changes?

A popular method of visually presenting data while bringing out significant trends and outliers is accomplished with charts. Charts frequently display data as pie charts, vertical and horizontal bar charts, geographical maps, as well as a variety of other visual displays. Besides offering visual trends of data, charts provide convenience and offer a standard way of performing data analysis through drill-down navigation.

Drill-down approaches are extremely flexible, allowing any chart to act as a graphical front-end interface to additional graphical or textual information. Data is frequently displayed by layering one chart on another providing an effective point-and-click user interface. At the top layer data is presented in a highly summarized fashion. As the user clicks on a designated area of the chart, data is displayed in significantly more detail. The user navigates down the various layers of data with one or more clicks of the mouse or a combination of cursor and keyboard commands to see more and more details. It's simple and, above all else, elegant.

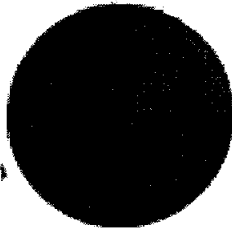
Building a Drill-down Application

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 HTML path with BODY and optional files.
3. Create graph using HTML= option and link variable.
4. Create detail list drill-down pages.
5. Use Web browser 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 to in build 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 odslib.movies;
  length urllink $30.;
  if upcase(rating) = "G" then
    urllink="href=g-ratings-body.html";
  else if upcase(rating) = "PG" then
    urllink="href=pg-ratings-body.html";
run;

*STEP 2 - Create 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=urllink;
  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';
  footnotel '<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';
  footnotel '<A HREF="drill-down-graphical-
application.html">
    Display Graph</A>';
  where upcase(rating)="PG";
run ;
ods HTML close;
ods Listing;
```

Testing Web Output

Before transferring HTML-generated 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 page differently than Netscape Navigator and others.

Before deploying web-based output to the Web, Intranet, or Extranet, it is important to visually inspect and test your output to see how your Web pages look and behave. As a means of assisting in these efforts, Web validation services are available for users to check web pages for errors or inconsistencies. The following recommendations provide a few items to consider before deploying Web output.

1. Check spelling – check the spelling on each of your Web pages before making them available on the Web.
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 will enable 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.

Conclusion

With the growing popularity of the Internet, you may find the Output Delivery System (ODS) useful in deploying selected pieces of output on the Web. ODS makes the deployment of SAS output a simple process. Syntactically correct HTML code is produced consisting of an HTML file, a table of contents, a table of pages, and a frame file for deployment using your favorite Internet browser.

Acknowledgments

I would like to thank the WUSS 2002 Leadership for accepting my abstract and paper, as well as for their hard work and support of a great Conference.

References

- Heffner, William F. (1998), "ODS: The DATA Step Knows," Proceedings of the 23rd Annual SAS Users Group International Conference, Cary, NC: SAS Institute Inc.
- Laffer, Kirk Paul (2001), "Output Delivery Tips, Tricks, and Techniques," Proceedings of the 26th Annual SAS Users Group International Conference.
- Laffer, Kirk Paul (2002), The SAS[®] Output Delivery System (ODS) Answer Guide, Revised and Updated, Software Intelligence Corporation, Spring Valley, CA, USA.
- Laffer, Kirk Paul (1999), "Delivering Results with the Output Delivery System," Proceedings of the 24th Annual SAS Users Group International Conference, Spring Valley, CA, USA.
- 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[®] 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[®] 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.

Trademark Citations

Internet Explorer and MS-Word are the registered trademarks of Microsoft Corporation, Redmond, WA, USA.

Netscape is the registered trademark of Netscape Corporation.

SAS, SAS Alliance Partner, and SAS Certified Professional are registered trademarks of SAS Institute Inc. in the USA and other countries. The ® symbol indicates USA registration.

About the Author

Kirk is a SAS Alliance Partner[®] and SAS Certified Professional[®] with 25 years of SAS software experience. He has written three books and over one hundred articles for professional journals and SAS User Group proceedings. His popular SAS Tips column appears regularly in the SANDS and SESUG Newsletters. His expertise spans application design and development, training, and programming using base-SAS, SQL, ODS, SAS/FSP, SAS/AF, SCL, FRAME, and SAS/EIS software.

The author welcomes comments and can be reached at:

Kirk Paul Lafler
Software Intelligence Corporation
P.O. Box 1390
Spring Valley, California 91979-1390
E-mail: KirkLafler@cs.com
<http://www.software-intelligence.com>
Voice: 619.660.2400