

The Output Delivery System (ODS) from Scratch

Kevin D. Smith, SAS Institute Inc., Cary, NC

ABSTRACT

The Output Delivery System (ODS) is a great way to move beyond reports that just use the listing output. Using ODS, you can generate reports in formats such as HTML, XML, PDF, PostScript, RTF, and Microsoft Excel. This paper tells you what you need to know to generate reports with ODS, from scratch. You'll learn how to generate multiple output formats, simultaneously; how to change the look of your report using styles; how to add text passages; and other "neat" tips (that is, helpful information).

INTRODUCTION

The Output Delivery System was created to enable SAS customers to generate more reports with new features than could be generated via the old-fashioned listing output. Until ODS was released, the output from the SAS System was relegated to plain text and ASCII art drawings of tables. By using ODS, you can now create various types of output; for example, HTML, XML, PDF, PostScript, RTF, Microsoft Excel, and many more.

However, maybe we're "getting ahead" of ourselves. You might not be familiar with some of the formats that ODS can create or know what these formats do. For your quick reference, Table 1 lists the most common ODS output types and shows what each type is typically used for.

Output Type	Description
HTML and XHTML	HTML is the primary format used on the World Wide Web. When you browse the Web, you are looking at HTML files. XHTML is HTML that is formatted as a valid XML file.
XML	XML is a well-defined, plain-text format. Until XML became popular, each file type required its own format, grammar, and parser. XML simplified the exchange of data by creating a file type that is easily processed by various tools.
PDF	PDF is a file format that can be used for printing as well as for online viewing. To view a PDF file, download the free Adobe Acrobat Reader. (Acrobat Reader is, usually, pre-installed on most computers that you buy today.)
PostScript	PostScript is a format that is used for printing. However, it is quickly being replaced by PDF.
RTF	RTF is a format that is used by most word processors.
Excel	The Microsoft Excel format enables you to share your SAS data with Excel users.
LaTeX	LaTeX enables you to copy and paste SAS output into other LaTeX documents which are generally used for scientific documentation.
Data set	Data set enables you to use ODS tables as data sets.

Table 1. Common Types of ODS Output

As you see, ODS can generate output for everything from publishing on the World Wide Web, to printing, to creating spreadsheets. Best of all, **ODS comes with every SAS installation at no additional charge!** You might not even realize it, but you already have access to ODS¹ if you are running SAS[®] 8 or later. Because this includes most SAS customers, let's get started creating some reports that will have you wondering how you survived until now with only the listing output.

CREATING VARIOUS TYPES OF REPORTS

LISTING OUTPUT

Right now, you're probably wondering: Why is there a section about listing output in an ODS paper? Well, believe it or not, the listing output that you "know-and-love" is an ODS destination². To prove it and to demonstrate the first use of the ODS statement, try this little bit of SAS code.

¹ While ODS has been around since SAS 8, not all the output types that are listed here were available at that time. See your SAS Representative if you need to update your software.

² *Destination* is another term for the ODS driver that generates a specific output type, such as PDF, RTF, or HTML. You might hear some people refer to ODS destinations as ODAs. No one really knows what "ODA" means (maybe it's an abbreviation for Output Destination Acronym), so, it's better to use the term *destination*.

```
ods listing close;
proc print data=sashelp.class;
run;
```

What did you get? Absolutely nothing! Except, probably, the following message:

```
WARNING: No output destinations active.
```

That's right. The listing output is, actually, an ODS destination. By using the global ODS statement with the CLOSE option, you turned off the listing output. You can turn it back on with the following code.

```
ods listing;
```

Now, you've already seen two uses of the global ODS statement. In the first example above, the ODS statement closes the destination; in the second example, the ODS statement opens the destination. The general form of the ODS statement is as follows:

```
ods destination-name <option(s)>;
```

where *destination-name* is the name of the ODS destination (for example, listing, PDF, RTF, and so on) and *option(s)* specifies the options that are associated with that destination.

What else can be included in the ODS statement? Most options are specific to the destination, but one option that is available in almost all destinations is the FILE= option, which enables you to send the output to a specific file. For listing output, the FILE= option directs the output to a file rather than displaying it online. Here is an example.

```
ods listing file='myoutput.txt';
proc print data=sashelp.class;
run;
proc contents data=sashelp.class;
run;
```

Instead of displaying the output in your window, you should see all the output in the file MYOUTPUT.TXT³. To make the listing print to the window again, open the listing destination without using the FILE= option. For example,

```
ods listing;
```

Now, that you've mastered the listing destination and the basic form of the ODS statement, let's do something more interesting, such as generating anything other than listing output.

OTHER DESTINATIONS

HTML

Files formatted in HTML are what you see when you browse the World Wide Web⁴. Creating HTML files is just as easy as creating listing output; you just have to turn on the ODS HTML destination by using the following statement:

```
ods html;
```

After you specify the preceding ODS statement, any procedure that generates output will also create HTML output. A message similar to the following appears, which gives you the name of the file that contains the HTML output.

```
NOTE: Writing HTML Body file: sashtml.htm
```

You can open this file in your Web browser to see your HTML report. If you use the SAS Display Manager, the HTML file will probably open up, automatically, in your results window. As an example, the following code generates the output that is shown in Figure 1.

```
ods html;
proc contents data=sashelp.class;
run;
```

³ You might want to use an absolute path name for your files (for example, c:\myoutput.txt) if you use a windowing environment because the files will be generated in the directory where SAS is running, and you might not know where that is.

⁴ If you want all the details about HTML such as its history, the specification, and so on, see the W3C Web site (W3C 2006).

The screenshot shows a SAS Output window titled "The SAS System". It contains two tables. The first table lists dataset properties for SASHELP.CLASS, and the second table lists engine/host dependent information.

The SAS System			
Data Set Name	SASHELP.CLASS	Observations	19
Member Type	DATA	Variables	5
Engine	V9	Indexes	0
Created	Thursday, May 19, 2005 03:25:02 PM	Observation Length	40
Last Modified	Thursday, May 19, 2005 03:25:02 PM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	us-ascii ASCII (ANSI)		

Engine/Host Dependent Information	
Data Set Page Size	4096
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	101

Figure 1. Output from Using the ODS HTML Destination

If you are familiar with the HTML language and you viewed the source of the HTML file, you might notice that the end of the file is missing. While this won't prevent most browsers from displaying the file, the file really should be closed, correctly. To do this, use the ODS statement again. (This is the same as when you closed the listing destination.)

```
ods html close;
```

Here is all the code for the PROC CONTENTS step.

```
ods html;
proc contents data=sashelp.class;
run;
ods html close;
```

The HTML destination supports the FILE= option, which enables you to direct the HTML output into a file of your choosing.

```
ods html file='myreport.html';
```

In addition to the output file, HTML can also generate files to support frames. Frames enable you to divide the Web browser view into multiple panels; each panel contains a separate HTML file. Usually, you use frames when you want to scroll one section of a Web page independently of another. The HTML destination uses this feature to create a table of contents that scrolls independently of the body of the report. In order to do this, you must create three files by using the following options:

FILE= or BODY=

controls the body of the report (including all the tables, titles, notes, and so on).

CONTENTS=

controls the table of contents of the report.

FRAME=

controls the wrapper file that surfaces the files that you specify in the BODY= and CONTENTS= options. You should point your Web browser to the file that you specify in the FRAME= option.

Here is an example of using these three options to create a framed HTML file with a table of contents.

```
ods html file='body.html'
      contents='contents.html'
      frame='myreport.html';
...proc code...
ods html close;
```

When you view the file in your browser, it looks like the file shown in Figure 2.

The SAS System			
Data Set Name	SASHELP.CLASS	Observations	19
Member Type	DATA	Variables	5
Engine	V9	Indexes	0
Created	Thursday, May 19, 2005 03:25:02 PM	Observation Length	40
Last Modified	Thursday, May 19, 2005 03:25:02 PM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	us-ascii ASCII (ANSI)		
Engine/Host Dependent Information			
Data Set Page Size	4096		
Number of Data Set Pages	1		

Figure 2. Output from Using the HTML Destination and the FILE=, CONTENT=, and FRAME= Options

While HTML reports are great for viewing in a window, these reports aren't very good for printing because Web browsers don't split tables across pages very well. For printing, you should use either the PDF or the PostScript destination.

PDF AND POSTSCRIPT

Both PDF and PostScript were created by Adobe to format paginated content. PostScript is an older and more powerful language, but it isn't as well supported for viewing in a windowing environment. PDF requires only the Adobe Acrobat Reader, which comes pre-installed on most computers (a free download is available at www.adobe.com).

The PDF and PostScript destinations work the same way as the listing and HTML destinations (how convenient!). You use the ODS PDF statement to open the PDF destination and the ODS PDF CLOSE statement to close it. For PostScript, you use the ODS PS statement and ODS PS CLOSE statements, respectively. Both of these destinations support the FILE= option to send the output to a file.

```
ods pdf file='myreport.pdf';
proc contents data=sashelp.class;
run;
ods pdf close;
```

Output from the preceding code is shown in Figure 3.

The CONTENTS Procedure

Data Set Name	SASHELP.CLASS	Observations	19
Member Type	DATA	Variables	5
Engine	V9	Indexes	0
Created	Thursday, May 19, 2005 03:25:02 PM	Observation Length	40
Last Modified	Thursday, May 19, 2005 03:25:02 PM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	us-ascii ASCII (ANSI)		

Engine/Host Dependent Information	
Data Set Page Size	4096
Number of Data Set Pages	1
First Data Page	1

Figure 3. Output from Using the PDF Destination and the FILE= Option

If you want a table of contents, use the CONTENTS=YES option as shown in the following statement. The result is shown in Figure 4.

```
ods pdf file='myreport.pdf' contents=yes;
```

Table of Contents

The Contents Procedure	1
SASHELP.CLASS	1
Attributes	1
Engine/Host Information	1
Variables	1

Figure 4. Output from Using the PDF Destination and the CONTENTS= Option**MICROSOFT EXCEL**

The Excel destination is a sub-destination of the Markup destination. The Excel destination is created by a tagset. (This paper doesn't give details about tagsets, for now, it's enough to know that a tagset tells the Markup destination what text should be printed for each part of the report⁵.) To generate Microsoft Excel files for Excel XP, use the following ODS statement.

```
ods tagsets.excelxp file='myreport.xls';
```

The output from a PROC CONTENTS step with the preceding statement will look like the output that is shown in Figure 5.

⁵ The HTML destination is actually a tagset.

	A	B	C	D	E
1	Data Set Name	SASHELP.CLASS	Observations	19	
2	Member Type	DATA	Variables	5	
3	Engine	V9	Indexes	0	
4	Created	Thursday, May 19, 2005 03:25:02 PM	Observation Length	40	
5	Last Modified	Thursday, May 19, 2005 03:25:02 PM	Deleted Observations	0	
6	Protection		Compressed	NO	
7	Data Set Type		Sorted	NO	
8	Label				
9	Data Representation	WINDOWS_32			
10	Encoding	us-ascii ASCII (ANSI)			
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					
31					

Figure 5. Output from PROC CONTENTS Using Excel Destination

By default, each table is put on a separate sheet. You can force all tables onto one sheet by using the SHEET_INTERVAL option, as shown in the following example.

```
ods tagsets.excelxp file='myreport.xls'
options( sheet_interval='none' );
```

	A	B
1	Data Set Name	SASHELP.CLASS
2	Member Type	DATA
3	Engine	V9
4	Created	Thursday, May 19, 2005 03:25:02 PM
5	Last Modified	Thursday, May 19, 2005 03:25:02 PM
6	Protection	
7	Data Set Type	
8	Label	
9	Data Representation	WINDOWS_32
10	Encoding	us-ascii ASCII (ANSI)
11		
12	Engine/Host Dependent Information	
13	Data Set Page Size	
14	Number of Data Set Pages	
15	First Data Page	
16	Max Obs per Page	
17	Obs in First Data Page	
18	Number of Data Set Repairs	
19	File Name	C:\SASv9\sasgen\dev\mva-v920\sas_dvd\src\dntnd\en\sashelp\class.sas7bda
20	Release Created	9.02C1B0
21	Host Created	WIN_PRO
22		
23	Alphabetic List of Variables and Attributes	
24	#	Variable
25	3	Age
26	4	Height
27	1	Name
28	2	Sex
29	5	Weight
30		
31		

Figure 6. Output from PROC CONTENTS Using Excel Destination and SHEET_INTERVAL Option

The Excel destination supports so many options that it's not possible to discuss them in an introductory paper such as this. However, here is a small sampling of the items that these options enable you to control: worksheet generation, tables of contents, table and title spacing, column widths, autofiltering, formulas, formats, and more. For details, see the SAS 9.1.3 *Output Delivery System: User's Guide* (SAS Institute Inc. 2006 (a)) or look at the various Excel XP references that are given on the SAS Customer Support Center Web site at support.sas.com.

YOU WANT MORE?

ODS also provides many destinations that aren't discussed here, but you probably already have an idea of how to use them because the ODS statement works in a similar way for all destinations. For more information about the other ODS destinations and their options, see the *SAS 9.1.3 Output Delivery System: User's Guide* (SAS Institute Inc. 2006 (a)) and the SAS Customer Support Center Web site at support.sas.com.

TIPS AND TRAPS

Here are some tips to help you avoid some problems that you might have when you use ODS.

ODS ... CLOSE;

At the end of your report, always use the ODS ... CLOSE statement (where ... is the name of the destination) to close the ODS destination. Although some destinations (such as HTML) might look okay in a Web browser without your using this statement, the files aren't complete and some important information might not be included in the report.

ALWAYS USE RUN;

If you don't use the RUN statement at the end of your program, your ODS report might not have the output from the last procedure that was executed. For example, without the RUN statement, the output from PROC PRINT would not be included in the report that results from the following program.

```
ods html;  
proc contents data=sashelp.class;  
proc print data=sashelp.class;  
ods html close;
```

The correctly written program follows.

```
ods html;  
proc contents data=sashelp.class;  
run;  
proc print data=sashelp.class;  
run;  
ods html close;
```

RUN MULTIPLE DESTINATIONS SIMULTANEOUSLY

You might not have realized that if you run the programs for the HTML, PDF, and Excel destinations that are shown in the examples, you are running multiple ODS destinations, simultaneously! In fact, there is nothing to stop you from running as many destinations as you want to, simultaneously. Try the following program:

```
ods html file='myreport.html';  
ods pdf file='myreport.pdf';  
ods tagsets.excelxp file='myreport.xls';  
proc contents data=sashelp.class; run;  
ods tagsets.excelxp close;  
ods pdf close;  
ods html close;
```

Typing all those ODS ... CLOSE statements can get tedious, can't it? That brings me to my next tip—the ODS _ALL_ CLOSE statement.

CLOSE MULTIPLE DESTINATIONS SIMULTANEOUSLY

Remembering to close multiple destinations might lead to errors because you might forget one statement. This is why the ODS _ALL_ CLOSE statement was created. Instead of typing the name of a destination between ODS and CLOSE, you can type _ALL_, which specifies that all the ODS destinations be closed, **including listing**. If you usually keep **listing** open, you have to add the ODS LISTING statement after the ODS _ALL_ CLOSE statement, as follows.

```
ods _all_ close;  
ods listing;
```

ADDING STYLE TO YOUR REPORTS

Now that you have the basics of using ODS, you might wonder if you can change the way the reports look. You might think that the blue-and-gray color scheme is alright, but it doesn't really fit with the color scheme of your Web site. Luckily, SAS includes many pre-packaged styles that you can use for your reports. Most destinations use the style that is named Default. However, destinations that are usually used for printing (for example, PDF, PostScript, and RTF) are better without background colors. PDF and PostScript use the style named Printer, and RTF uses the aptly named, RTF style.

The next section shows you how to locate and apply the styles that are provided by SAS to your reports.

LOCATE EXISTING STYLES

Currently, about 50 different styles are shipped with SAS. They are located in the SASHELP.TMPLMST template store. A *template store* is a specially formatted file that is created by PROC TEMPLATE. You can browse these styles by using the Template Browser; type ODSTEMPLATES in the RUN field in the SAS window. The styles are stored in the Styles folder of the SASHELP.TMPLMST template store.

If you don't use the windowing environment in SAS or if you prefer to write code instead of clicking, you can use PROC TEMPLATE to list the styles, as shown in the next example.

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

If you want to see each of these styles, run the code that is given under the heading "Generating a Sample of All Styles" in the Appendix of this paper to generate a sample of the styles in HTML, PDF, and RTF formats.

Now that you know the names of all the styles and know how they look, you can choose which style you want to try for your own reports.

ODS STYLE= OPTION

The preceding section explains how to browse the collection of styles that are provided with SAS, how to obtain a list of the styles by using PROC TEMPLATE, and how to generate a sample of all the styles that are provided. This section shows you how to use these styles in your own reports.

By default, HTML uses the style named Default. This name is an abbreviation of Styles.Default—the Default style is in the Styles folder of SASHELP.TMPLMST. You can specify which style you want by using the STYLE= option in the ODS statement. Here's an example.

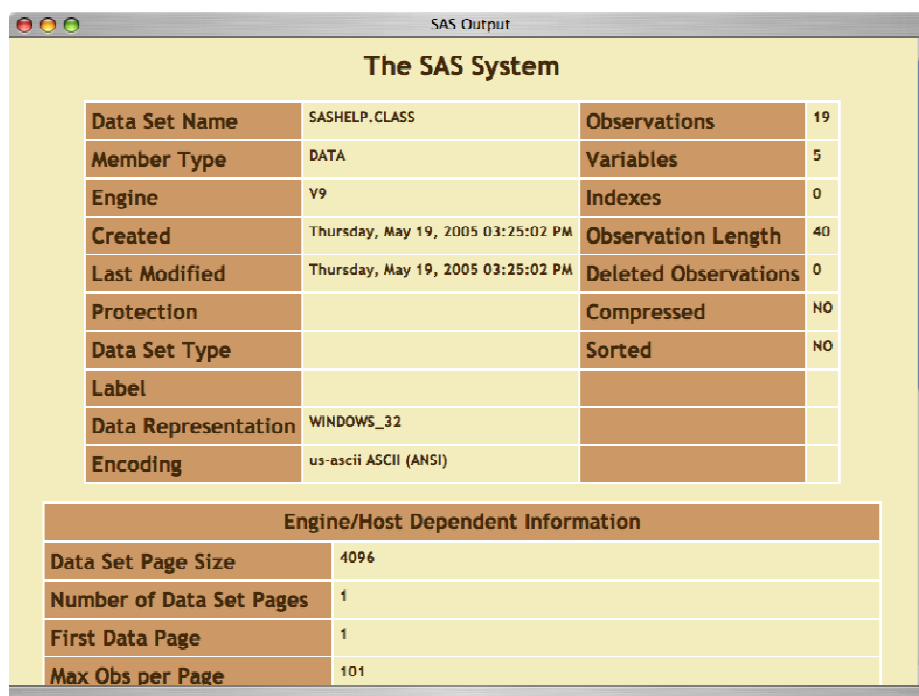
```
ods html style=styles.default;
```

Because Styles.Default is the style that HTML uses by default when you don't specify a value for STYLE=, your report won't look any different when you use the preceding line of code. However, if you use the techniques that are given in the earlier section to locate another style that you want to try (for example, Styles.Gears), you can change the way that your report looks by typing the following statement

```
ods html style=styles.gears;
```

A report that is generated by using the preceding ODS statement is shown in Figure 7⁶.

⁶ You can see some differences between this output and the previous HTML output examples but, to really see the difference, you need to see the output in color. That's your homework for today.



The screenshot shows a SAS Output window with a title bar 'SAS Output'. Inside, there's a section titled 'The SAS System' which contains a table with 12 rows and 4 columns. The first two columns are for attributes and their values, and the next two are for related attributes and their values. Below this is another section titled 'Engine/Host Dependent Information' which contains a table with 4 rows and 2 columns.

The SAS System			
Data Set Name	SASHELP.CLASS	Observations	19
Member Type	DATA	Variables	5
Engine	V9	Indexes	0
Created	Thursday, May 19, 2005 03:25:02 PM	Observation Length	40
Last Modified	Thursday, May 19, 2005 03:25:02 PM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	us-ascii ASCII (ANSI)		

Engine/Host Dependent Information	
Data Set Page Size	4096
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	101

Figure 7. HTML Output from Using the STYLE= Option

Each destination sets its style, independently. Therefore, you can generate, at the same time, PDF, HTML, and other formats that look completely different from each other. However, you might think that, although the styles provided by SAS are nice, none of those styles is exactly what you want. Fortunately, you can define your own; unfortunately, this can be a rather daunting task.

DEFINE YOUR OWN STYLES

Many SAS programmers have had some difficulty when given the task of creating a customized ODS style. Well, it's really not that difficult but defining ODS styles is a bit quirky⁷. Teaching how to create an ODS style is beyond the scope of this paper; however, you can search the online *Proceedings* from previous SAS Conferences (SAS Institute Inc. 2006 (b)) for papers that have been written about this subject.

If you want to see what the code for an ODS style looks like, type ODSTEMPLATES on the command line in a SAS window to open the Template Browser. Expand the SASHELP.TMPLMST folder and select the folder "Styles". Double-click the name of any style to see the PROC TEMPLATE code. You can also see the source of any style by using the PROC TEMPLATE SOURCE statement, as shown next.

```
proc template;
  source <style-name>;
run;
```

To see the source of Styles.Default, use the following code.

```
proc template;
  source styles.default;
run;
```

CUSTOMIZE YOUR REPORTS

Besides specifying a style, you can customize your reports in many ways. For example, you can remove selected tables from a report, modify parts of the table of contents, and create your own ODS destinations. These techniques, and more, are discussed in the following sections.

⁷ SAS 9.2 will provide a graphical interface that will make it much easier to create styles.

ODS SELECT; AND ODS EXCLUDE;

Sometimes, processes generate more information than you want. For example, think back about the use of PROC CONTENTS. If you don't specify any options, PROC CONTENTS prints three tables. The first table contains the data set attributes; the second table contains the engine (host) information; and the third table contains information about the variables. Let's suppose that you are interested only in the third table. Wouldn't it be great if there were a way to turn the other two tables off? Luckily for you, the developers of ODS thought of this, and they created the ODS SELECT and ODS EXCLUDE statements.

The ODS SELECT statement specifies which tables should be printed. The ODS EXCLUDE statement specifies which tables should **not** be printed. Here is the simple form for each statement,⁸

```
ods select <selection(s)>;
ods exclude <selection(s)>;
```

where *selection(s)* specifies one or more paths, labels, label paths, or any combination thereof for the table that is being generated. How do you find the element to specify for *selection(s)*? Use the following ODS TRACE statement.

```
ods trace on / label;
```

With ODS TRACE set to ON, as the table is generated, you see the names, labels, paths, and other information for each table that is printed to the log. The output from the preceding ODS TRACE statement for the third table that is generated by PROC CONTENTS looks like this.

```
Output Added:
-----
Name:      Variables
Label:     Variables
Template:  Base.Contents.Variables
Path:     Contents.DataSet.Variables
Label Path: 'The Contents Procedure'. 'SASHELP.CLASS'. 'Variables'
-----
```

Back to the original problem...you want to print only the last table in a PROC CONTENTS step. With ODS TRACE ON, you see that the name of the table is "Variables". You can use this name in the ODS SELECT statement and print only that table.

```
ods select variables;
proc contents data=sashelp.class;
run;
```

If you run the preceding code, you'll see that only the selected table is printed. Depending on how many tables you are printing, it might be easier to use the ODS EXCLUDE statement instead of the ODS SELECT statement. The code below generates the same output as the previous example.

```
ods exclude attributes enginehost;
proc contents data=sashelp.class;
run;
```

As mentioned previously, you can specify tables by their path, label, or label path. This means that, instead of specifying the name "Variables" for the table in the ODS SELECT statement in the preceding example, you can use either of the following statements.

```
ods select contents.dataset.variables;
ods select 'The Contents Procedure'. 'SASHELP.CLASS'. 'Variables';
```

Because the table names and paths refer to the table names of specific procedures, the ODS SELECT and ODS EXCLUDE lists are cleared after each procedure is run.

⁸ While both the SELECT and EXCLUDE statements are shown here together, you should run only one in your code per procedure. Specifying both can result in unpredictable behavior.

There are more advanced usages of the ODS SELECT and ODS EXCLUDE statements that enable you to specify that the select and exclude lists persist over multiple procedures and to select a specific occurrence of a table. For more information about these techniques, see the *SAS 9.1.3 Output Delivery System: User's Guide* (SAS Institute Inc. 2006 (a)).

OTHER CUSTOMIZATIONS

The following sections describe a few, simple customizations that enable you to interact with a procedure's Table of Contents (TOC) entry and titles.

ODS PROCLABEL=<STRING>;

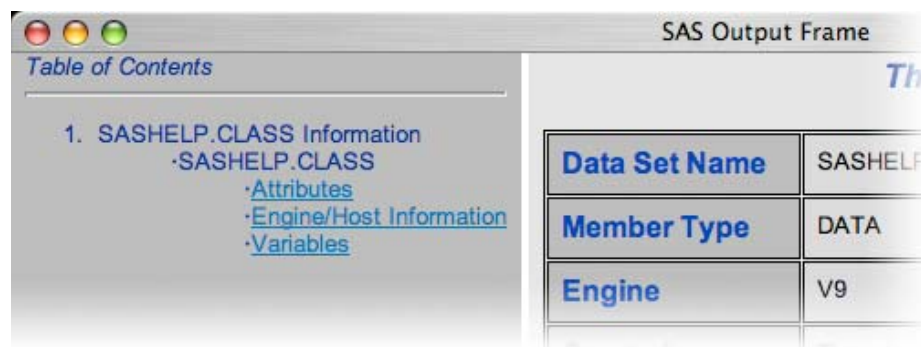
Although ODS doesn't let you control very much in the Table of Contents, it does let you change the procedure's top-level TOC entry⁹ by using the ODS PROCLABEL statement, as shown here,

```
ods proclabel <string>;
```

where *string* is the text that you want to use in place of the default procedure label. For example, the following code changes the default Table of Contents entry, which is "The Contents Procedure", to "SASHELP.CLASS Information".

```
ods proclabel="SASHELP.CLASS Information";
```

If you run PROC CONTENTS in the HTML destination, you'll see output that is similar to the output in Figure 8.



The screenshot shows the SAS Output Frame with two panes. The left pane, titled 'Table of Contents', displays a hierarchical list: '1. SASHELP.CLASS Information' followed by sub-items '·SASHELP.CLASS', '·Attributes', '·Engine/Host Information', and '·Variables'. The right pane shows the output of PROC CONTENTS for the SASHELP.CLASS data set, presented as a table with three rows: 'Data Set Name' (SASHELP), 'Member Type' (DATA), and 'Engine' (V9).

Table of Contents	
1. SASHELP.CLASS Information	
·SASHELP.CLASS	
·Attributes	
·Engine/Host Information	
·Variables	

PROC CONTENTS	
Data Set Name	SASHELP
Member Type	DATA
Engine	V9

Figure 8. Output from Using the HTML Destination and PROC CONTENTS

Just as when you use the ODS SELECT statement, the ODS PROCLABEL statement is cleared after each procedure is run. So, if you want to keep the changes in the top-level TOC entry, you have to issue the ODS PROCLABEL statement for each procedure.

ODS PROCTITLE; AND ODS NOPROCTITLE;

Most procedures print a title at the top of their output that indicates which procedure created those tables. You might not want this information. ODS enables you to turn those titles off by using ODS NOPROCTITLE statement, as shown in the following code.

```
ods noproctitle;
```

After issuing the preceding statement, procedure titles will not appear in your reports. You can print the procedure titles in your reports again by using the following statement.

```
ods proctitle;
```

ADVANCED TECHNIQUES

There aren't any really advanced techniques presented in this paper. However, I want to be sure that you are aware of what you should do after you have learned and are proficient in everything that is given in this paper. The following sections should help.

ODS DOCUMENT

The ODS Document is an ODS destination that enables you to store your reports in an itemstore and then replay the pieces in any order (and for as many times) as you want to. Its effect is similar to the results from using the ODS

⁹ This is the one that usually contains something like "The Contents Procedure".

SELECT and ODS EXCLUDE statements, but the ODS Document is much more powerful. The ODS SELECT and ODS EXCLUDE statements enable you to suppress the printing of specific tables; however, you have to run the procedure multiple times if you want to change the order of tables when the procedure is run. The ODS Document enables you to print a table that is generated by a procedure multiple times without re-invoking that procedure.

Another benefit of the ODS Document is that it enables you to completely control the Table of Contents. Normally, the procedure completely controls the TOC. By using PROC DOCUMENT, you can create a TOC hierarchy from scratch, and insert only the tables that you want from your all your procedure runs. In my opinion, the ODS Document is **the** most under-utilized tool when generating reports in SAS.

PROC TEMPLATE

Many people cringe when they hear the words "PROC TEMPLATE." Although that response might be somewhat deserved, that negative reaction is usually brought about by the reputation of PROC TEMPLATE styles, which are a little problematic to implement. In reality, PROC TEMPLATE does much more than develop styles. PROC TEMPLATE is a framework for creating templates, but the term "templates" is used to describe how everything (that is, tables, styles, graphs, text markup) should look. The "difficulty" with templates is that each of these categories uses a slightly different (sometimes, more than slightly different) syntax. Because of this, it's easier to think of each category of template as having its own language. Table 2 shows each of the current types of templates and how they are used.

Template	Description
Style	Style templates describe what reports should look like, stylistically. This includes fonts, colors, table borders, and page margins.
Table	Table templates describe how tables should be constructed. This includes the content and placement of headers and footers, the content and placement of columns, and style overrides. All SAS procedures, except PROC PRINT, PROC REPORT, and PROC TABULATE, use Table templates to describe how their tables should look. This means that you can change the structure of tables that are generated by SAS procedures by using Table templates.
Tagset	Tagsets are ODS destinations. Tagset templates describe the markup that should be generated for each part of a report. Tagset templates can be used to generate any type of text-based output, such as HTML, CSV (comma-separated values), XML, and LaTeX.
StatGraph (Experimental)	StatGraph templates are very similar to Table templates. However, instead of describing tabular output, StatGraph templates describe a graphical representation. In many cases, StatGraph templates and Table templates can be used interchangeably. As of this writing, StatGraph templates are still experimental.

Table 2. Types of Templates Created by Using PROC TEMPLATE

It is inevitable that more template types will be developed in the future, so, the earlier you begin using PROC TEMPLATE, the easier it will be for you in your career.

A wise man said: *"The true path to ODS enlightenment lies with PROC TEMPLATE."*

Okay, no "wise man" really said that, but that doesn't mean that it isn't true. If you can become proficient in using only half of the types of templates that are described in Table 2, you will release a lot of the power of ODS that remains unseen by many people.

CONCLUSION

Now that you have completed this condensed course in ODS, you'll be able to

- Create reports that you never thought you could.
- Generate HTML for delivery on the Internet.
- Generate PDF for printing.
- Use Microsoft Excel files to share your data with co-workers who don't use SAS.
- Find the styles that are provided with SAS and apply these styles to your own reports.
- Customize your reports by using the ODS SELECT and ODS EXCLUDE statements.

Roman author Publilius Syrus said, "Practice is the best of all instructors." So, use the techniques that are presented in this paper on your own reports, and look forward to using the advanced techniques after you are skilled in the use of the basic techniques.

APPENDIX: GENERATING A SAMPLE OF ALL STYLES

The following SAS program generates a sample report in every style that is provided by SAS in HTML, PDF, and RTF. When the program is finished, open the file INDEX.HTML in your Web browser. Keep in mind that not all styles are designed to work well in every ODS destination.

```
/* This program prints a sample report in HTML, PDF, and RTF */
/* in every style that is specified in ODS path. */

ods listing close;

proc template;

define table vstyle;
  column libname memname style links;
  define links;
    header = 'Samples';
    compute as '
```

REFERENCES

Adobe Systems Inc. 2007. Adobe Web Site. Available at www.adobe.com.

SAS Institute Inc. 2006 (a). *SAS® 9.1.3 Output Delivery System: User's Guide, Volumes 1 and 2*. Cary, NC: SAS Institute Inc. Available at support.sas.com/documentation/onlinedoc/91pdf/sasdoc_913/base_ods_9268.pdf.

SAS Institute Inc. 2006 (b). *SAS Global Forum Proceedings of Previous Conferences*. Available at support.sas.com/events/sasglobalforum/previous/index.html.

SAS Institute Inc. 2007. SAS Customer Support Center Web Site. Available at support.sas.com.

W3C. 2006. W3C Interaction Domain: HTML Web Site. HyperText Markup Language (HTML) Home Page. Available at www.w3.org/MarkUp/.

CONTACT INFORMATION

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

Kevin D. Smith
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
E-mail: Kevin.Smith@sas.com
Web: www.sas.com

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.