Chapter 9

Producing HTML Output

Topics

In this chapter you will learn how to create and view HTML output using **ODS** (**Output Delivery System**) and how to apply styles to ODS output.

- Open and close ODS destinations
- Create a simple HTML file with the output of one or more procedures
- Create HTML output with a linked table of contents in a frame
- Use options to specify links and file paths
- View HTML output
- Apply styles to HTML output

Introduction

- SAS can produce both listing and HTML output.
- HTML output (HTML 4.0 formatting) is created by default in the SAS Windowing environment in the Windows and UNIX operating systems.
- In other operating systems, you can set options to create HTML output.
- You can use ODS to create HTML output and many other types of output, e.g., RTF, PDF, Excel, and many others.

The Output Delivery System (ODS): How Does It Work?

1. ODS creates your output in the form of **output objects**. Each output object contains the results of a procedure or DATA step (the data component) and can also contain information about how to render the results (the table definition).

Data Component   Table Definition

+
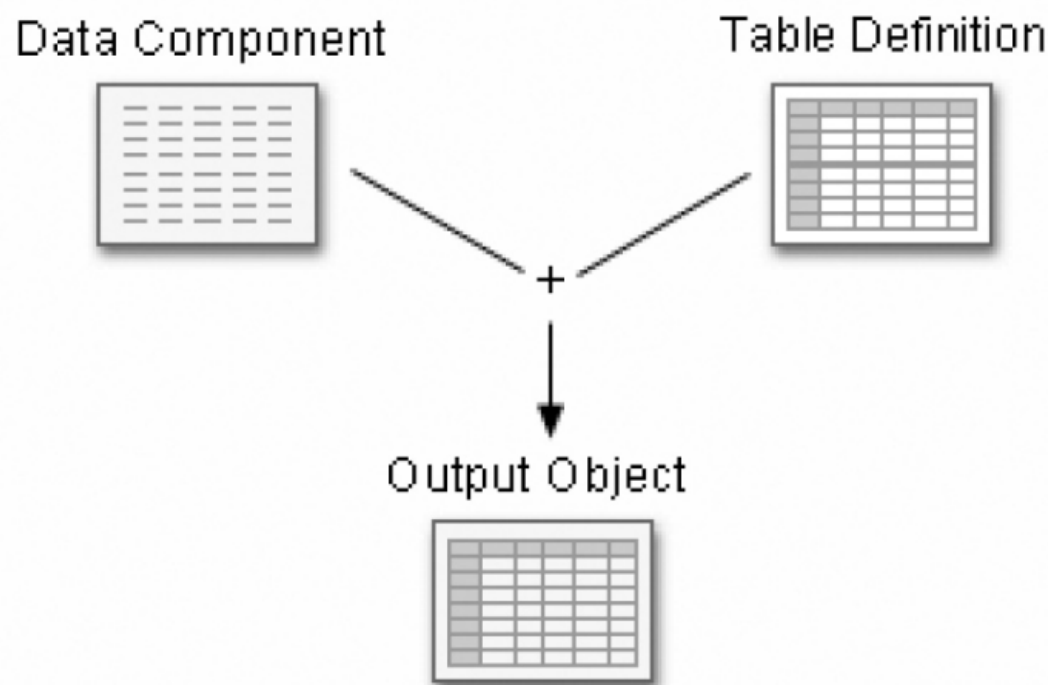
Output Object

The Output Delivery System (ODS): How Does It Work?

2. ODS sends the output object to the **ODS destination(s)** that you specify and creates formatted output as specified by the destination. For example, when the LISTING and HTML destinations are open, ODS creates LISTING and HTML output.

Data Component   Table Definition

+

Output Object

Listing   HTML   ODS Destinations

Listing Output   HTML Output   ODS Output
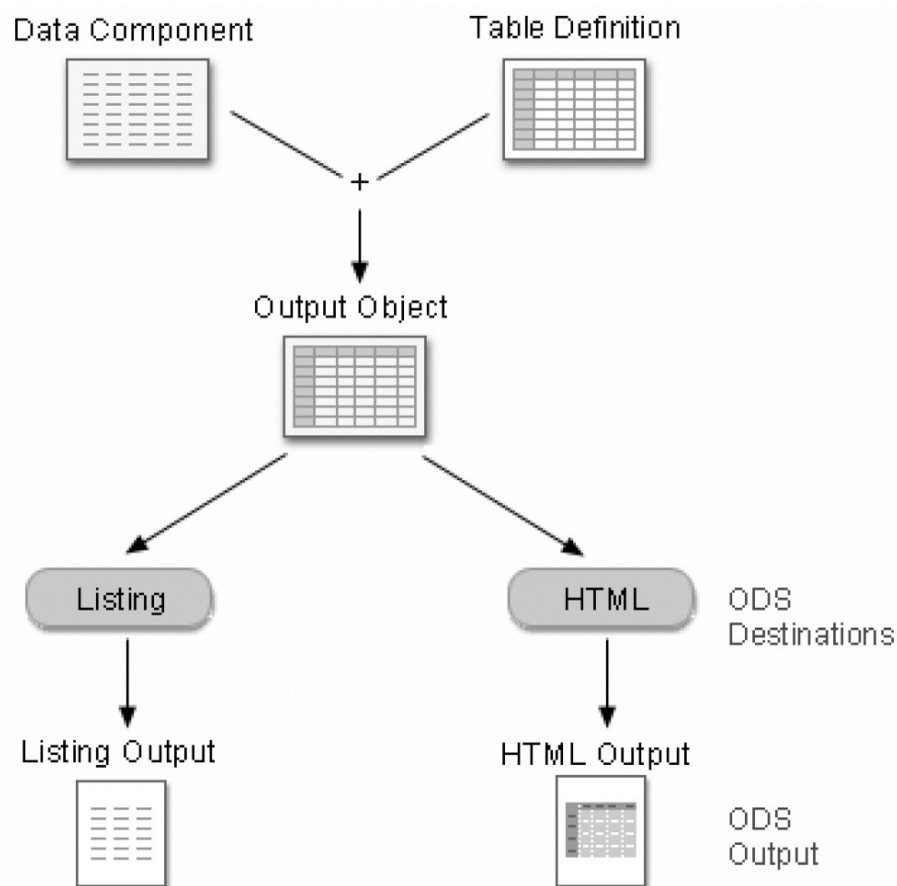
The Output Delivery System (ODS): How Does It Work?

3. ODS creates a **link** to each output object in the Results window and identifies each output object by the appropriate icon.
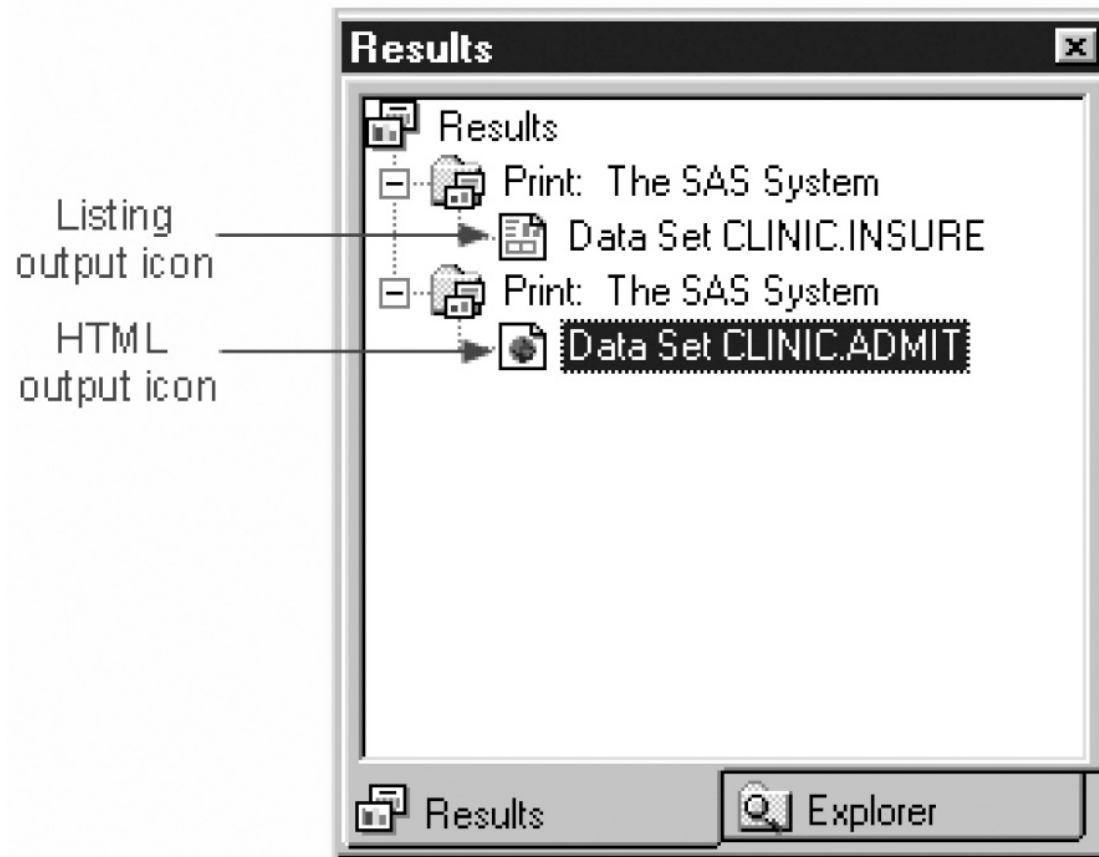
Opening and Closing ODS Destinations

ODS destinations

This destination …	Produces …
HTML	output that is formatted in Hypertext Markup Language (HTML). This is the default destination. You do not have to specify the ODS HTML statement to produce basic HTML output.
LISTING	plain text output that is formatted like traditional SAS procedure (LISTING) output
Markup Languages Family	output that is formatted using markup languages such as Extensible Markup Language (XML)
Document	a hierarchy of output objects that enables you to render multiple ODS output without rerunning procedures
Output	SAS data sets
Printer Family	output that is formatted for a high-resolution printer such as PostScript (PS), Portable Document Format (PDF), or Printer Control Language (PCL) files
RTF	Rich Text Format output for use with MS Word

Opening and Closing ODS Destinations

General form of the ODS statement

ODS *open-destination***;**
ODS *close-destination* **CLOSE;**

where
- *open-destination* is a keyword and any required options for the type of output that you want to create, such as
  - **HTML FILE** *= 'html-file-pathname'*
  - **LISTING**
- *close-destination* is a keyword for the type of output.

Note: the HTML destination is open by default.

Opening and Closing ODS Destinations: Examples

```
/*produce HTML output since the
HTML destination is open by default*/
proc print data=sasuser.mydata;
run;

/*produce HTML and LISTING output*/
ods listing;
proc print data=sasuser.mydata;
run;
ods listing close;

/*produce only LISTING output*/
ods html close; /*close an HTML destination*/
ods listing;
proc print data=sasuser.mydata;
run;
ods listing close;
ods html;
```

```
/*produce output in multiple
formats at once by opening each
ODS destination at the beginning of
the program and use keyword _ALL_
to close all open destinations at
once*/
ods rtf file='C:\SAS\out.rtf';
ods pdf file='C:\SAS\out.pdf';
proc print data=sasuser.admit;
run;
ods _all_ close;
ods html;  /*reopens the HTML
           destination so that ODS
           returns to producing
           HTML output for
           subsequent DATA or
           PROC steps in the current
           SAS session*/
```

Creating HTML Output With Options

To create simple HTML output, you do not have to specify the ODS HTML statement. However, to create HTML output with options specified, you open the HTML destination using the ODS HTML statement.

<p style="text-align:center">ODS HTML BODY=file-specification;</p>
<p style="text-align:center">ODS HTML CLOSE;</p>

where *file-specification* identifies the file that contains the HTML output. The specification can be

- a quoted string which contains the HTML filename (use only the filename to write the file to your <u>current working directory</u> location, such as **C:\My SAS Files)** Example: **ODS HTML BODY=** *"myreport.html";*
- a quoted string which contains the <u>complete directory path</u> and HTML filename Example: **ODS HTML BODY=** *"c:\reportdir\myreport.html";*
- a fileref (unquoted file shortcut) that has been assigned to an HTML file using the FILENAME statement Example: **FILENAME MYHTML** *"c:\reportdir\myreport.html";* **ODS HTML BODY=MYHTML;**
- a SAS catalog entry in the form *entry-name*.html. Note that the catalog name is specified in the PATH= option and the *entry-name*.html value for the BODY= option is <u>unquoted</u>. Example: **ODS HTML PATH=**work.mycat **BODY=**myentry.html;

Creating HTML Output with Options: Example

/*The program below creates PROC PRINT output in an HTML file. The BODY= option specifies the file **G:\STSCI 5010 spring 2017\Output\admit.html** as the file that contains the PROC PRINT results.*/
ods html body='G:\STSCI 5010 fall 2017\Output\admit.html';
proc print data=clinic.admit label;
    var sex age height weight actlevel;
    label actlevel='Activity Level';
run;
ods html close;
ods html;

Note: the keyword **body=** can be replaced with **file=**, which is an alias of **body=**

Obs	Sex	Age	Height	Weight	Activity Level
1	M	27	72	168	HIGH
2	F	34	66	152	HIGH
3	F	31	61	123	LOW
4	F	43	63	137	MOD
5	M	51	71	158	LOW
6	M	29	76	193	HIGH
7	F	32	67	151	MOD
8	M	35	70	173	MOD
9	M	34	73	154	LOW
10	F	49	64	172	LOW
11	F	44	66	140	HIGH
12	F	28	62	118	LOW
13	M	30	69	147	MOD
14	F	40	69	163	HIGH
15	M	47	72	173	MOD
16	M	60	71	191	LOW
17	F	43	65	123	MOD
18	M	25	75	188	HIGH
19	F	22	63	139	LOW
20	F	41	67	141	HIGH
21	M	54	71	183	MOD

Viewing Your HTML Output

- In the Windows environment, when you submit the program, the body file will automatically appear in the Results Viewer (using the internal browser) or your preferred Web browser, as specified in the **Results** tab of the Preferences window.
- In other operating environments, double-click the corresponding link in the Results window to view the HTML output in your Web browser. If you don't have a Web browser, transfer the HTML files to an operating environment where you can view them.
- In <u>SAS Enterprise Guide</u> when you specify a path and filename for your HTML results, it allows you to download and display the HTML file. You can also use Windows Explorer to locate the file that you created and then double-click it to open it in your browser.

Creating HTML Output From Multiple Procedures

You can also use the ODS HTML statement to direct the results from multiple procedures to <u>the same HTML file</u>.

```
ods html body='c:\records\data.html';
proc print data=clinic.admit label;
    var id sex age height weight actlevel;
    label actlevel='Activity Level';
run;
proc tabulate data=clinic.stress2;
    var resthr maxhr rechr;
    table min mean, resthr maxhr rechr;
run;
ods html close;
ods html;
```

The results for both procedures are saved to the file **C:\Records\data.html**. A link for each output object (one for each procedure) appears in the Results window.

Creating HTML Output with a Table of Contents

You can create an HTML file that has <u>a table of contents</u> with links to the output of each specific procedure.

/*contents= creates a file containing a table of contents; frame= creates a file integrating the table of contents and the body file (if you specify FRAME=, you must also specify CONTENTS=.)*/
ods html body= 'c:\records\data.html'
    contents= 'c:\records\toc.html'
    frame= 'c:\records\frame.html';
proc print data=clinic.admit label;
    var id sex age height weight actlevel;
    label actlevel='Activity Level';
run;
proc print data=clinic.stress2;
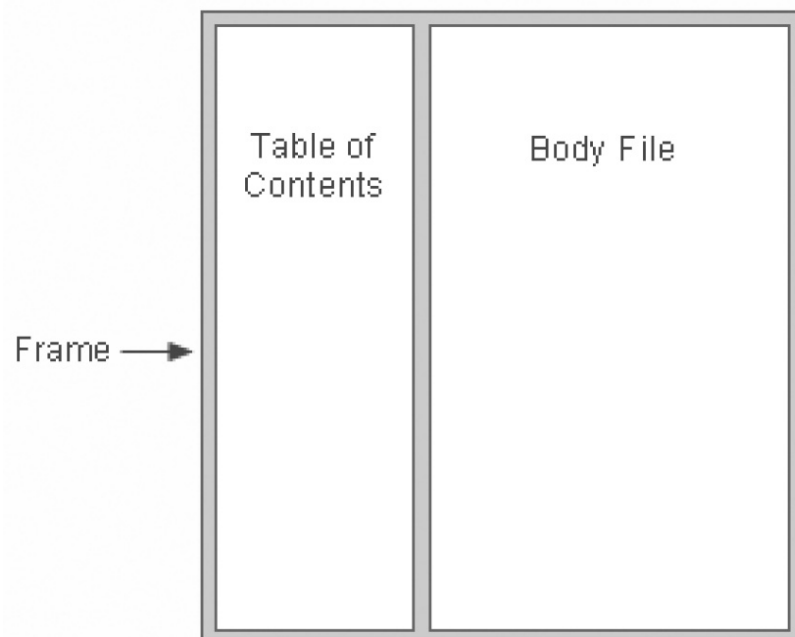   var id resthr maxhr rechr;
run;
ods html close;
ods html;

Viewing Frame Files and Using the Table of Contents

- To view the frame file that integrates the body file and the table of contents, select **File Open** from within the internal browser or your preferred Web browser. Then open the frame file that you specified using FRAME= (here, **frame.html**), which is stored in the Records directory in the Windows environment.
- The table of contents created by the CONTENTS= option contains a numbered heading for each procedure that creates output. Each heading is a link to the output for that procedure.

Using Options to Specify Links and Paths

- In the previous example, the links have the <u>complete pathnames</u>. These links work when you are viewing the HTML files locally. If you want to place these files on a Web server so that other people could access them, you need to use the URL= **suboption** in the BODY= or CONTENTS= file specification to achieve this.
- The general form is
    (URL= 'Uniform-Resource-Locator')
    where *Uniform-Resource-Locator* is the name of *an HTML file* or *the full URL of an HTML file*. ODS uses this URL in all the links and references to the file.
- It can be a complete URL for an <u>absolute link </u>or the HTML filename for a <u>relative link</u>.

Using Relative URLs

In this case, the URL= suboption specifies only the HTML filename. This is the most common style of linking between files because maintenance is easier. The files can be moved as long as they all remain in the same directory or storage location.

ods html body='c:\records\data.html' **(url='data.html')**
        contents='c:\records\toc.html' **(url='toc.html')**
        frame='c:\records\frame.html';

```
<FRAME MARGINWIDTH="4" MARGINHEIGHT="0" SRC="toc.htm"
     NAME="contents" SCROLLING=auto>
<FRAME MARGINWIDTH="9" MARGINHEIGHT="0" SRC="data.htm"
     NAME="body" SCROLLING=auto>
```

Using Absolute URLs

In this case, the URL= suboptions specify complete URLs using Hypertext Transfer Protocol (HTTP). These files can be stored in the same or different locations.

ods html body='c:\records\data.html'
      (url='http://mysite.com/myreports/data.html')
      contents='c:\records\toc.html'
      (url='http://mysite.com/mycontents/toc.html')
      frame='c:\records\frame.html';

```
<FRAME MARGINWIDTH="4" MARGINHEIGHT="0" SRC="http://mysite.com/myreports/data.html"
    NAME="contents" SCROLLING=auto>
<FRAME MARGINWIDTH="9" MARGINHEIGHT="0" SRC="http://mysite.com/myreports/toc.html"
    NAME="body" SCROLLING=auto>
```

The PATH Option

You can use the PATH option to streamline your ODS HTML statement and specify the location where you want to store your HTML output. General form:

PATH=*file-location-specification* <(URL= NONE \\'*Uniform-Resource-Locator'*)>

where *file-location-specification* identifies the location where you want HTML files to be saved. It can be one of the following:
- the complete pathname to an aggregate storage location, such as a directory
- a fileref (file shortcut) that has been assigned to a storage location
- a SAS catalog (*libname.catalog*).

Uniform-Resource-Locator provides a URL for links in the HTML files that ODS generates. If you specify the keyword **NONE**, no information from the PATH= option appears in the links or references.

If you do not use the URL= suboption, information from the PATH= option is added to links and references in the files that are created.

Example: PATH= Option with URL=NONE

The PATH= option directs the files data.html, toc.html, and frame.html to the **C:\Records** directory in the Windows operating environment. The links from the frame file to the body and contents files contain only the HTML filenames data.html and toc.html.

```
ods html path='c:\records' (url=none)
          body='data.html'
          contents='toc.html'
          frame='frame.html';
proc print data=clinic.admit;
run;
proc print data=clinic.stress2;
run;
ods html close;
ods html;
```

Note If you plan to move your HTML files, you should specify URL=NONE

Example: PATH= Option with the URL= Suboption

In the program below, the PATH= option directs the files data.html, toc.html, and frame.html to the **C:\Records** directory. The links from the frame file to the body and contents files contain the specified URL, http://mysite.com/myreports/data.html and http://mysite.com/myreports/toc.html:

```
ods html path='c:\records'  (url='http://mysite.com/myreports/')
         body='data.html'
         contents='toc.html'
         frame='frame.html';
proc print data=clinic.admit;
run;
proc print data=clinic.stress2;
run;
ods html close;
ods html;
```

Changing the Appearance of HTML Output

You can change the appearance of your HTML output by specifying a style in the STYLE= option in the ODS HTML statement. Some of the style definitions that are currently shipped with SAS are

- BarrettsBlue
- Default
- Harvest
- HTMLblue
- Minimal
- Statistical

Example: Changing the Appearance of HTML Output

ods html body='c:\records\data.html'
    style=harvest;
proc print data=clinic.admit label;
    var sex age height weight actlevel;
run;
ods html close;
ods html;

Obs	Sex	Age	Height	Weight	ActLevel
1	M	27	72	168	HIGH
2	F	34	66	152	HIGH
3	F	31	61	123	LOW
4	F	43	63	137	MOD
5	M	51	71	158	LOW
6	M	29	76	193	HIGH
7	F	32	67	151	MOD
8	M	35	70	173	MOD
9	M	34	73	154	LOW
10	F	49	64	172	LOW

The SAS System

Example: Changing the Appearance of HTML Output

To view a full list of the available style definitions, click the Results tab. Then right-click the **Results icon** and select **Templates** from the pop-up menu.

In the Templates window, open Sashelp.tmplmst. Then open the Styles folder. A list of the available style definitions appears in the right panel of the Templates window.
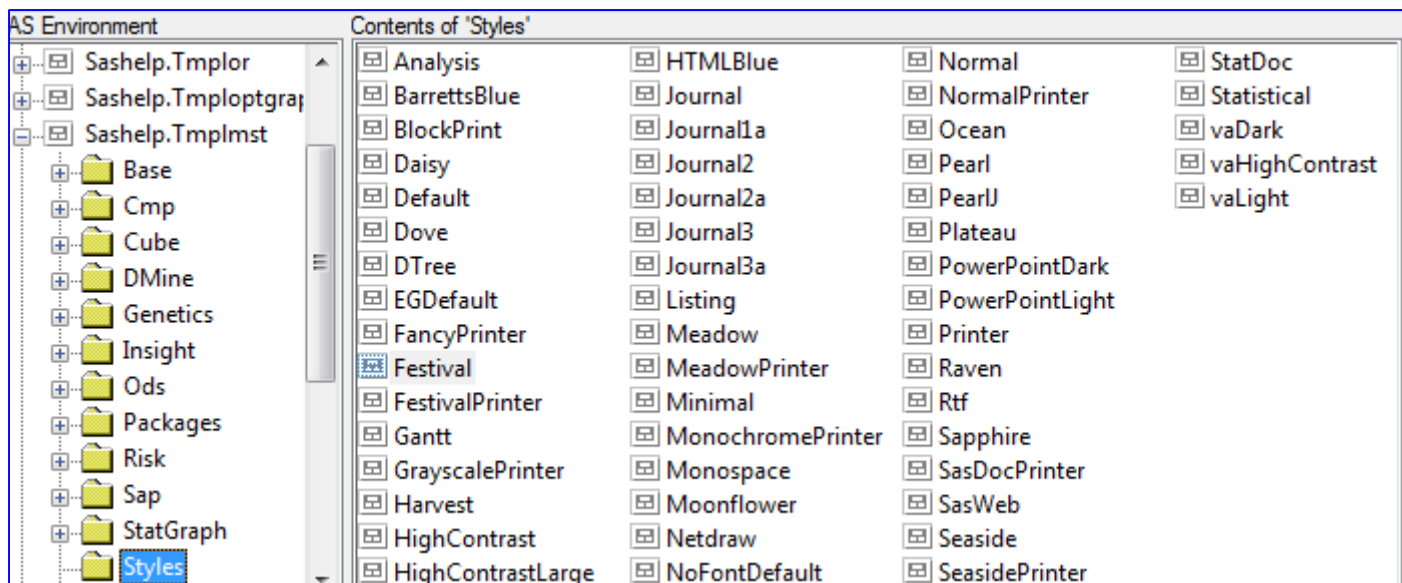
Additional Features: Customizing HTML Output

ODS provides ways for you to customize HTML output using definitions for tables, columns, headers, and so on. These definitions describe how to render the HTML output or part of the HTML output. You can create style definitions using the TEMPLATE procedure. See the SAS documentation for more information.