SAS Certification Prep Guide: Base Programming for SAS 9, Third Edition

by SAS Institute

Chapter 9: Producing HTML Output

Overview

Introduction

In previous chapters, you've seen both traditional SAS LISTING output and HTML output. HTML output is created by default in the SAS Windowing environment in the Windows and UNIX operating systems. If you are not using SAS in the SAS in the SAS Windowing environment in the Windows and UNIX operating systems, then you can set options to create HTML output. The HTML created by default is basic HTML 4.0 with default formatting.

You can use ODS to create many other types of output. You can create RTF, PDF, Excel, and many others using ODS statements. This chapter will discuss HTML output.

Using ODS, you can customize and manage HTML output by submitting programming statements. After you create HTML files, you can view them in the Results Viewer or Internet Explorer, Netscape Navigator, or any Web browser that fully supports HTML 4.0.

This chapter shows you how to create and view HTML output using ODS. You also learn how to apply styles to ODS output.

Caution By default, all code that you submit to SAS with Enterprise Guide has ODS statements included to create HTML output.

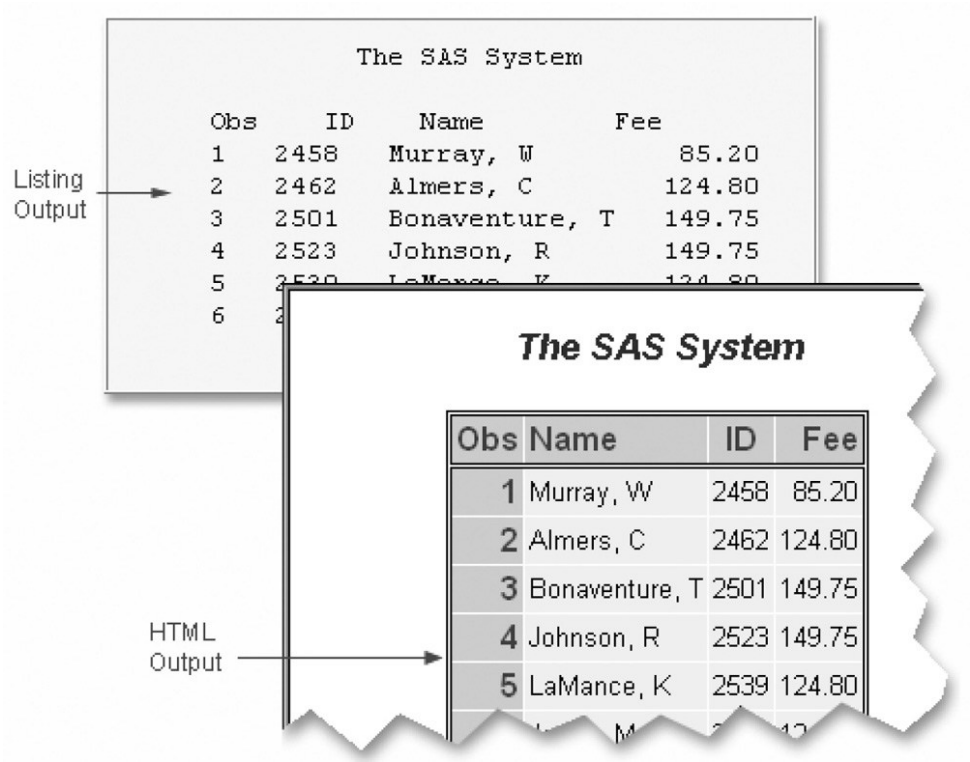


Figure 9.1: LISITING Output versus HTML Output

Objectives

In this chapter, you learn to

- 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.

The Output Delivery System

Before you learn to write ODS programming statements, it is helpful to understand a little about ODS.

Advantages of ODS

ODS gives you formatting options and makes procedure output much more flexible. With ODS, you can easily create output in a variety of formats, including

- HTML output

The SAS System									
Obs	**ID**	**Name**	**Sex**	**Age**	**Date**	**Height**	**Weight**	**ActLevel**	**Fee**
1	2588	Ivan, H	F	22	06/02/97	63	139	LOW	85.20
2	2586	Derber, B	M	25	06/04/97	75	188	HIGH	85.20
3	2458	Murray, W	M	27	06/05/97	72	168	HIGH	85.20
4	2572	Oberon, M	F	28	06/05/97	62	118	LOW	85.20
5	2544	Jones, M	M	29	06/07/97	76	193	HIGH	124.80
6	2574	Peterson, V	M	30	06/08/97	69	147	MOD	149.75

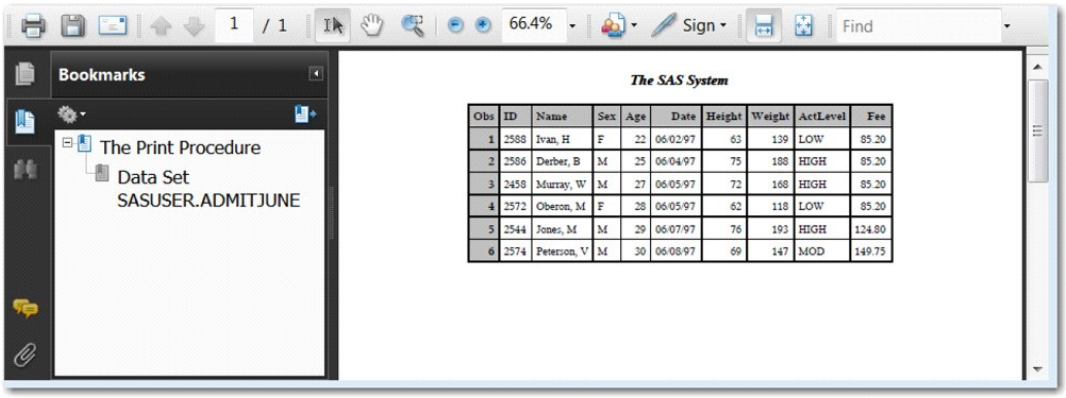
Figure 9.2: HTML Output

- RTF output

The SAS System									
Obs	**ID**	**Name**	**Sex**	**Age**	**Date**	**Height**	**Weight**	**ActLevel**	**Fee**
1	2588	Ivan, H	F	22	06/02/97	63	139	LOW	85.20
2	2586	Derber, E	M	25	06/04/97	75	188	HIGH	85.20
3	2458	Murray, W	M	27	06/05/97	72	168	HIGH	85.20
4	2572	Oberon, M	F	28	06/05/97	62	118	LOW	85.20
5	2544	Jones, M	M	29	06/07/97	76	193	HIGH	124.80
6	2574	Peterson, V	M	30	06/08/97	69	147	MOD	149.75

Figure 9.3: RTF Output

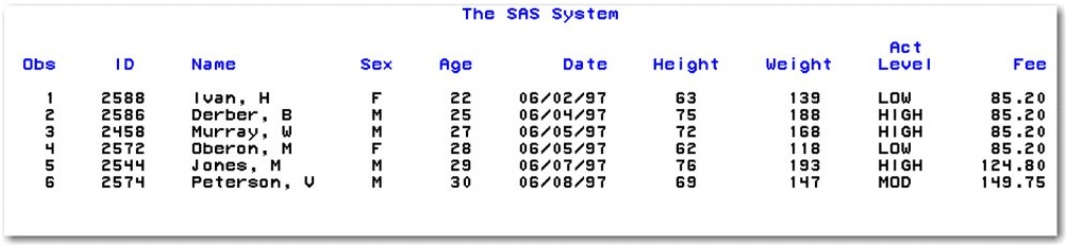
- PDF output



Obs	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee
1	2588	Ivan, H	F	22	06/02/97	63	139	LOW	85.20
2	2586	Derber, B	M	25	06/04/97	75	188	HIGH	85.20
3	2458	Murray, W	M	27	06/05/97	72	168	HIGH	85.20
4	2572	Oberon, M	F	28	06/05/97	62	118	LOW	85.20
5	2544	Jones, M	M	29	06/07/97	76	193	HIGH	124.80
6	2574	Peterson, V	M	30	06/08/97	69	147	MOD	149.75

Figure 9.4: PDF Output

- traditional SAS LISTING output



The SAS System

Obs	ID	Name	Sex	Age	Date	Height	Weight	Act Level	Fee
1	2588	Ivan, H	F	22	06/02/97	63	139	LOW	85.20
2	2586	Derber, B	M	25	06/04/97	75	188	HIGH	85.20
3	2458	Murray, W	M	27	06/05/97	72	168	HIGH	85.20
4	2572	Oberon, M	F	28	06/05/97	62	118	LOW	85.20
5	2544	Jones, M	M	29	06/07/97	76	193	HIGH	124.80
6	2574	Peterson, V	M	30	06/08/97	69	147	MOD	149.75

Figure 9.5: LISTING Output

Also, ODS holds your output in its component parts (data and table definition) so that numerical data retains its full precision.

Let's see how ODS creates output.

How ODS Works

When you submit your ODS statements and the SAS program that creates your output, ODS does the following:

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).

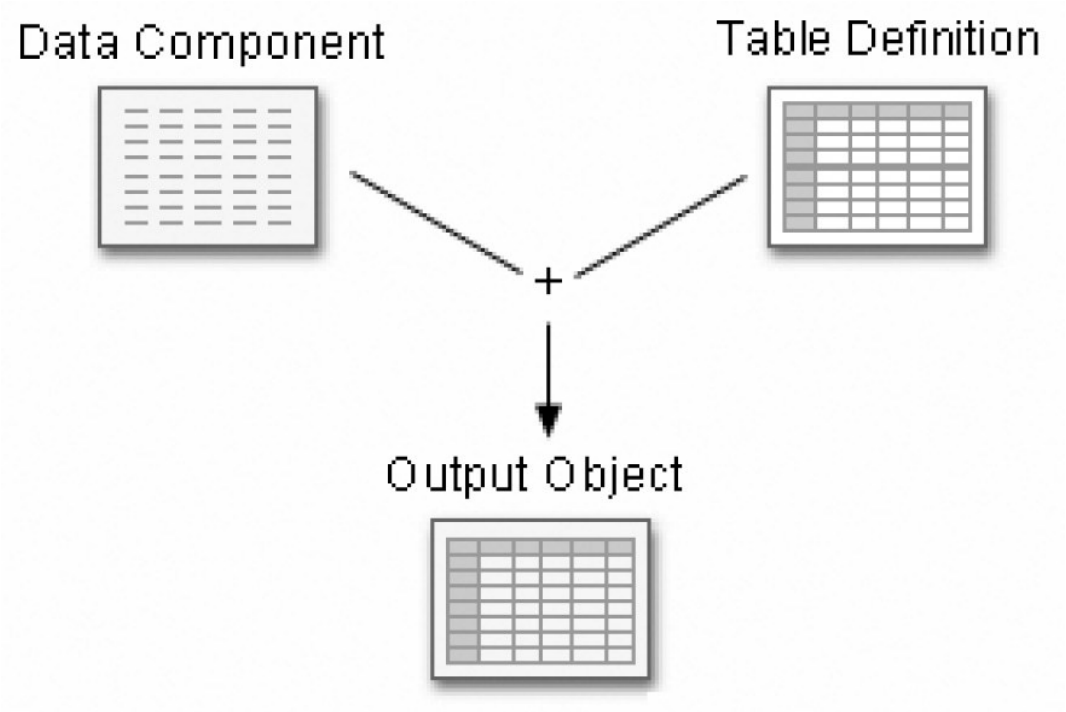


Figure 9.6: ODS Processing: What Goes In

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.

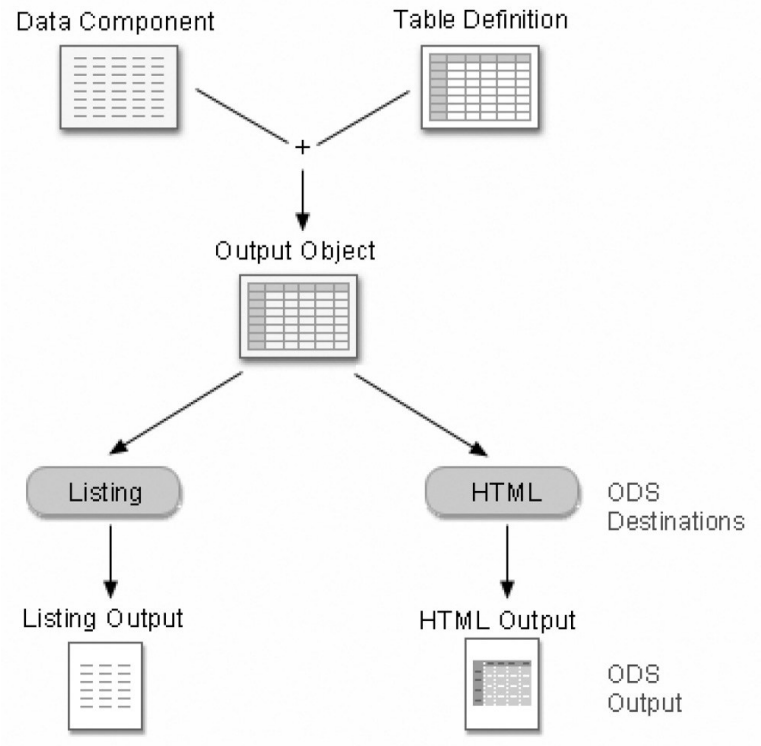


Figure 9.7: ODS Processing: What Comes Out

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

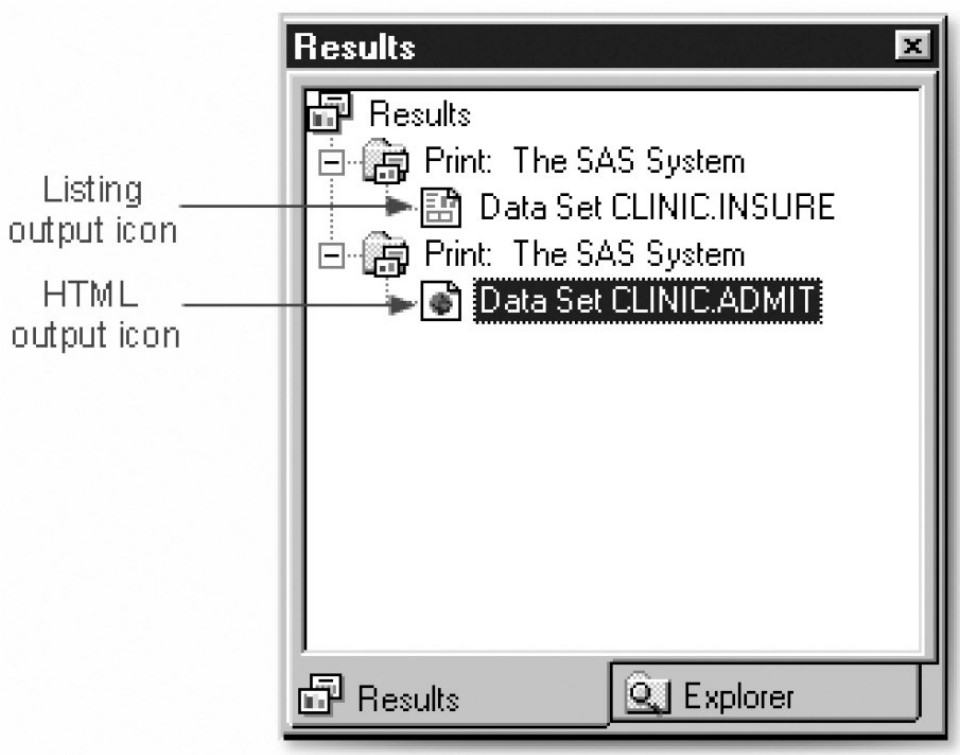


Figure 9.8: Results Window

Let's look at how you open various ODS destinations.

Opening and Closing ODS Destinations

ODS Destinations

You use ODS statements to specify destinations for your output. Each destination creates a specific type of formatted output. The table below lists some of the ODS destinations that are currently supported.

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 Word

Additional Note In this chapter, we'll primarily work with the LISTING destination and the HTML destination. For information about all ODS destinations, please see the SAS documentation for the Output Delivery System.

Using Statements to Open and Close ODS Destinations

For each type of formatted output that you want to create, you use an ODS statement to open the destination. At the end of your program, you use another ODS statement to close the destination so that you can access your output.

General form, ODS statement to open and close destinations:

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.

You can issue ODS statements in any order, depending on whether you need to open or close the destination. Most ODS destinations are closed by default. You open them at the beginning of your program and close them at the end. The exception is the HTML destination, which is open by default.

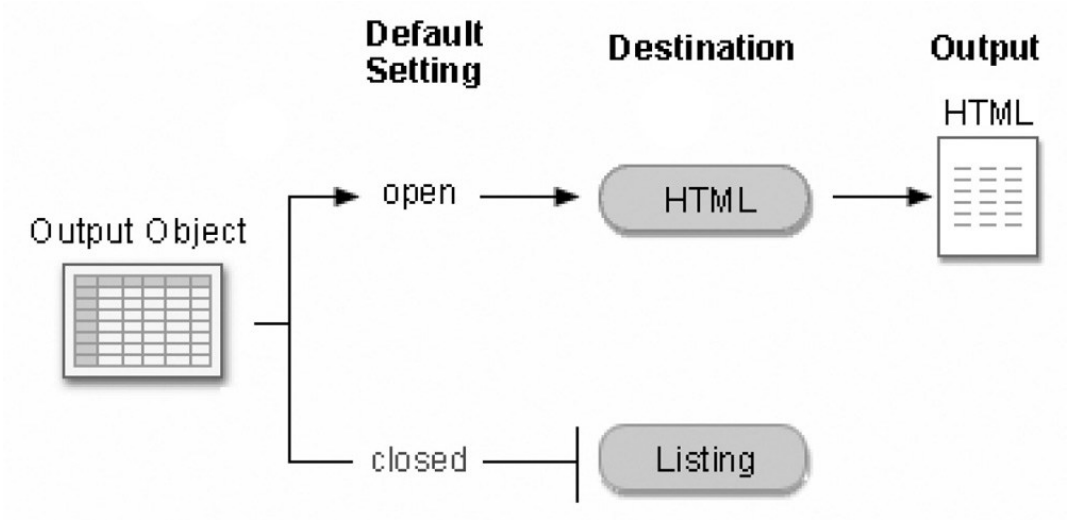


Figure 9.9: Default ODS Destination

Example

The following program creates HTML output because the HTML destination is open by default. No other ODS destinations are open, so no other output objects are produced.

```
proc print data=sasuser.mydata;
run;
```

The following program produces HTML and LISTING output:

```
ods listing;
proc print data=sasuser.mydata;
run;
ods listing close;
```

Additional Note This example is meant to demonstrate how you open and close ODS destinations. You learn the specifics of creating HTML output later in this chapter.

Closing the HTML Destination

As you have learned, the HTML destination is open by default. Because open destinations use system resources, it is a good idea to close the HTML destination at the beginning of your program if you don't want to produce HTML output. For example:

```
ods html close;
```

The HTML destination remains closed until you end your current SAS session or until you re-open the destination. It's a good programming practice to reset ODS to HTML output (the default setting) at the end of your programs. For example:

```
ods html;
```

Example

The following program produces only LISTING output:

```
ods html close;
ods listing;
proc print data=sasuser.mydata;
run;
ods listing close;
ods html;
```

Closing Multiple ODS Destinations at Once

You can produce output in multiple formats at once by opening each ODS destination at the beginning of the program.

When you have more than one open ODS destination, you can use the keyword _ALL_ in the ODS CLOSE statement to close all open destinations at once.

Example

The program below opens the RTF and PDF destinations before the PROC step and closes all ODS destinations at the end of the program:

```
ods rtf file='RTF-file-pathname';
ods pdf file='PDF-file-pathname';

proc print data=sasuser.admit;
run;

ods _all_ close;
ods html;
```

Notice that the last ODS statement reopens the HTML destination so that ODS returns to producing HTML output for subsequent DATA or PROC steps in the current session.

Now that you have learned how to open and close ODS destinations, let's look at creating HTML output with options.

Creating HTML Output with Options

ODS HTML Statement

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.

General form, ODS HTML statement:

ODS HTML BODY=*file-specification*;

ODS HTML CLOSE;

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 current working directory location, such as **`C:\Documents and Settings\`***username*`\My Documents\My SAS Files)` Example: **`ODS HTML BODY=`** *"myreport.html";*
- a quoted string which contains the complete directory path and HTML filename (include the complete pathname if you want to save the HTML file to a specific location other than your working directory) 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 unquoted. Example: **`ODS HTML PATH=`***work.mycat* **`BODY=`***myentry* **`BODY=.HTML;`**

Additional Note **`FILE=`** can also be used to specify the file that contains the HTML output. FILE= is an alias for BODY=.

Additional Note You can also use the **`PATH=`** option to explicitly specify a directory path for your file. This option will be discussed in an upcoming topic.

Example

The program below creates PROC PRINT output in an HTML file. The BODY= option specifies the file **`F:\admit.html`** in the Windows operating environment as the file that contains the PROC PRINT results.

```
ods html body='f:\admit.html';
proc print data=clinic.admit label;
  var sex age height weight actlevel;
  label actlevel='Activity Level';
run;
ods html close;
ods html;
```

The HTML file admit.html contains the results of all procedure steps between the ODS HTML statement and ODS HTML CLOSE statement.

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

Figure 9.10: HTML Output

Viewing Your HTML Output

If you're working 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, you can 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 in your operating environment, you can transfer the HTML files to an operating environment where you can view them.

Creating HTML Output in Enterprise Guide

When you specify a path and filename for your HTML results, SAS Enterprise Guide will allow you to download and display the HTML file. In addition, you can use Windows Explorer to locate the file that you created and then double-click it to open it in your browser.

When you submit a program, two HTML results appear in the Project window. One is the HTML output that is created by SAS Enterprise Guide ODS statements. The other uses the ODS statements from the code that you submitted and creates a temporary file labeled with the path and filename that you designated. It is similar in style to the actual HTML file that is created in the location that you specify.

You can double-click the shortcut to open the HTML results in SAS Enterprise Guide, or you can right-click the shortcut and select **Open with default-browser** to open it in a browser. To see the actual file rather than the temporary file, you can use Windows Explorer to locate the HTML 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 the same HTML file.

The program below generates HTML output for the PRINT and TABULATE procedures. The results for both procedures are saved to the file **C:\Records\data.html** in the Windows environment. A link for each output object (one for each procedure) appears in the Results window.

```
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;
```

The following is a representation of the HTML file containing the results from the program above. The results from the TABULATE procedure are appended. The results also show up in the Results window, where you can select the PRINT procedure output or the TABULATE procedure output.

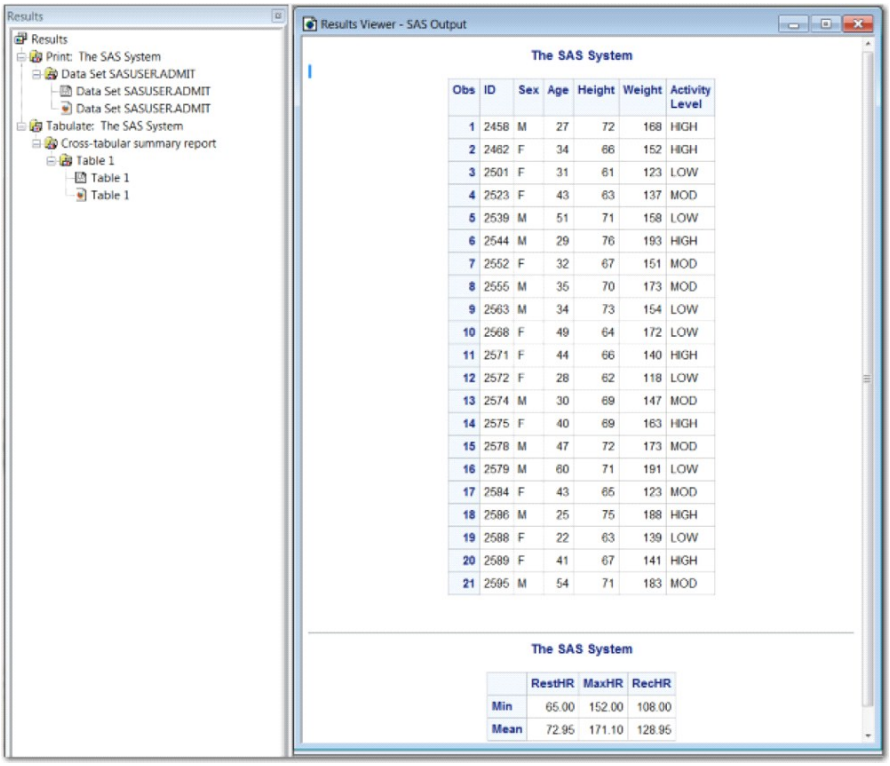


Figure 9.11: HTML Output

Creating HTML Output with a Table of Contents

Overview

So far in this chapter, you've used the BODY= specification to create an HTML file containing your procedure output. Suppose you want to create an HTML file that has a table of contents with links to the output of each specific procedure. You can do this by specifying additional files in the ODS HTML statement.

General form, ODS HTML statement to create a linked table of contents:

ODS HTML

BODY=*body-file-specification*

CONTENTS=*contents-file-specification*

FRAME=*frame-file-specification*;

ODS HTML CLOSE;

where

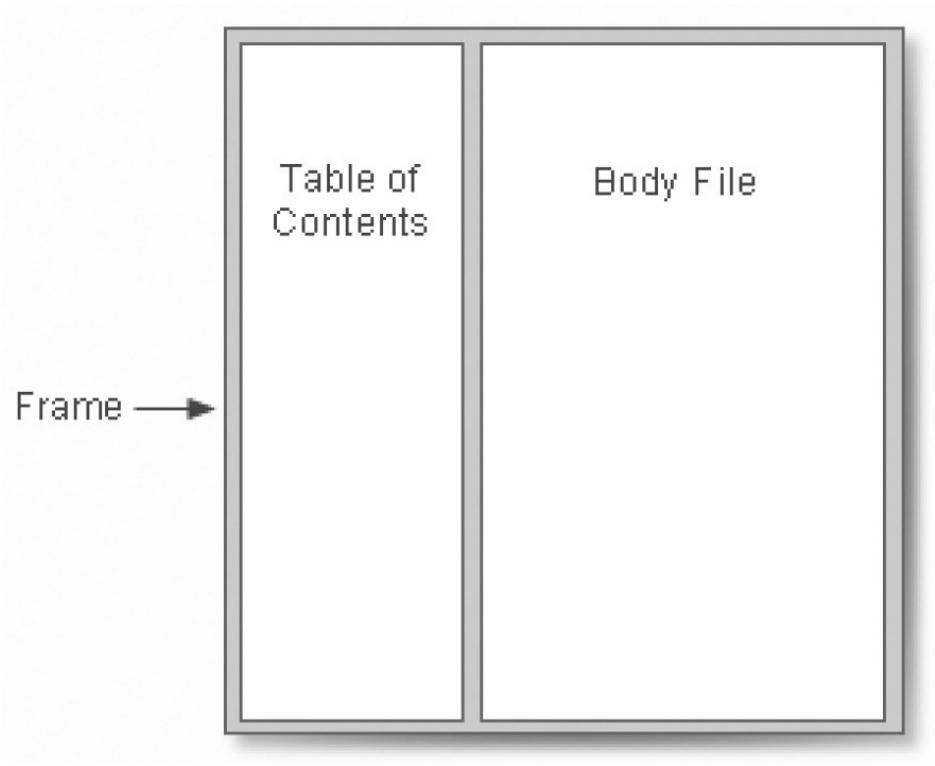
- *body-file-specification* is the name of an HTML file that contains the procedure output.
- *contents-file-specification* is the name of an HTML file that contains a table of contents with links to the procedure output.
- *frame-file-specification* is the name of an HTML file that integrates the table of contents and the body file. If you specify FRAME=, you must also specify CONTENTS=.

Additional Note To direct the HTML output to a specific storage location, specify the complete pathname of the HTML file in the *file-specification*.

Example

In the program below,

- the BODY= specification creates data.html in the **c:\records** directory. The body file contains the results of the two procedures.
- the CONTENTS= specification creates toc.html in the **c:\records** directory. The table of contents file has links to each procedure output in the body file.
- the FRAME=specification creates frame.html in the **c:\records** directory. The frame file integrates the table of contents and the body file.




```
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

The Results window does not display links to frame files. In the Windows environment, only the body file will automatically appear in the internal browser or your preferred Web browser.

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=. In the example above, this file is frame.html, which is stored in the Records directory in the Windows environment.

The frame file, frame.html, is shown below.

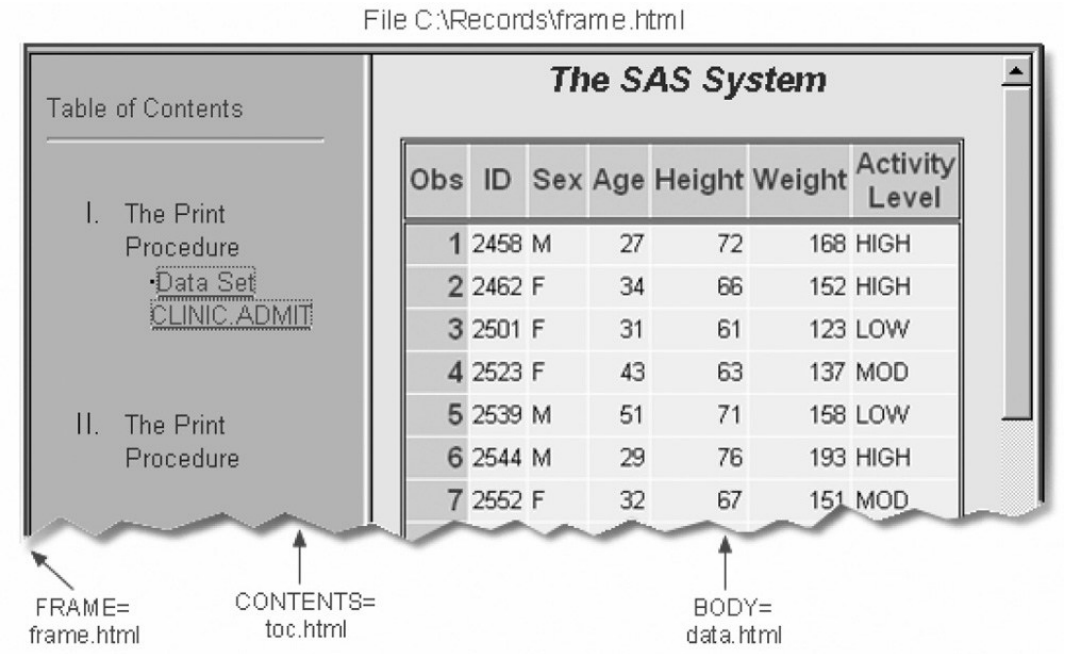


Figure 9.12: Frame File, Frame.html

Using the Table of Contents

The table of contents created by the CONTENTS= option contains a numbered heading for each procedure that creates output. Below each heading is a link to the output for that procedure.

On some browsers, you can select a heading to contract or expand the table of contents.

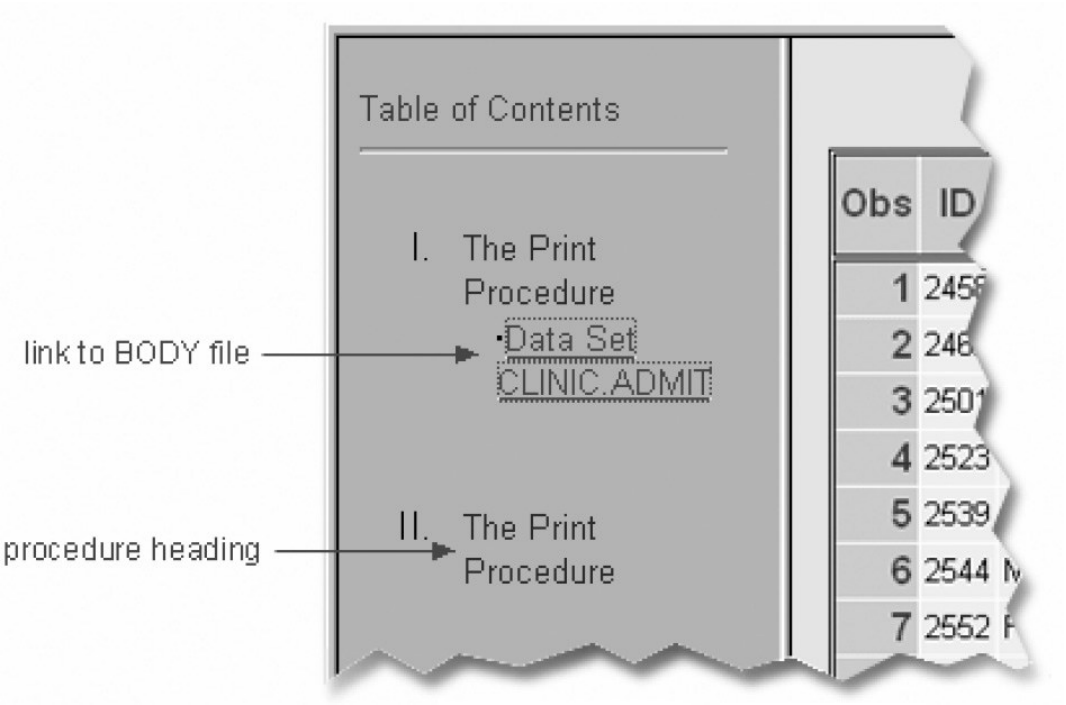


Figure 9.13: Table of Contents

Using Options to Specify Links and Paths

Overview

When ODS generates HTML files for the body, contents, and frame, it also generates links between the files using the

HTML filenames that you specify in the ODS HTML statement. If you specify complete pathnames, ODS uses those pathnames in the links that it generates.

The ODS statement below creates a frame file that has links to **`c:\records\toc.html`** and **`c:\records\data.html`**, and a contents file that has links to **`c:\records\data.html`.**

```
ods html body='c:\records\data.html'
         contents='c:\records\toc.html'
         frame='c:\records\frame.html';
```

A portion of the source code for the HTML file frame.html is shown below. Notice that the links have the complete pathnames specified in the file specifications for the contents and body files.

```
<FRAME MARGINWIDTH="4" MARGINHEIGHT="0" SRC="c:\records\toc.html"
     NAME="contents" SCROLLING=auto>
<FRAME MARGINWIDTH="9" MARGINHEIGHT="0" SRC="c:\records\data.html"
     NAME="body" SCROLLING=auto>
```

Figure 9.14: Source Code for the HTML File Frame.html

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, then the link needs to include either the complete URL for an absolute link or the HTML filename for a relative link.

The URL= Suboption

By specifying the URL= suboption in the BODY= or CONTENTS= file specification, you can provide a URL that ODS uses in all the links that it creates to the file. You can use the URL= suboption in any ODS file specification except FRAME= (because no ODS file references the frame file).

General form, URL= suboption in a file specification:

(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 instead of the file specification in all the links and references that it creates that point to the file.

Additional Note The URL= suboption is useful for building HTML files that might be moved from one location to another. If the links from the contents and page files are constructed with a simple URL (one name), they work as long as the contents, page, and body files are all in the same location.

Example: Relative URLs

In this ODS HTML statement, 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';
```

The source code for frame.html has only the HTML filename as specified on the URL= suboptions for the body and contents files.

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

Figure 9.15: Source Code Frame.html

Example: Absolute URLs

Alternatively, in this ODS HTML statement, 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';
```

As you would expect, the source code for Frame.html has the entire HTTP addresses that you specified in the URL= suboptions for the body and contents files.

```
<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>
```

Figure 9.16: Source Code Frame.html

Additional Note When you use the URL= suboption to specify a complete URL, you might need to move your files to that location before you can view them.

The PATH= Option

So far, you've learned to specify the complete pathname for HTML files in the BODY=, CONTENTS=, and FRAME= specifications when you want to save HTML files to specific locations. You can use the following options to streamline your ODS HTML statement:

PATH= option

specifies the location where you want to store your HTML output

URL=NONE suboption

prevents ODS from using the pathname in any links that it creates in your files

---

General form, PATH= option with the URL= suboption:

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 or partitioned data set
  - 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.

Caution In the z/OS operating environment, if you store your HTML files as members in a partitioned data set, the PATH= value must be a PDSE, not a PDS. You can allocate a PDSE within SAS as shown in this example:

```
filename pdsehtml '.example.htm'

   dsntype=library dsorg=po

   disp=(new, catlg, delete);
```

Also, you should specify valid member names for the HTML files (without extensions).

Example: PATH= Option with URL=NONE

In the program below, 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;
```

This program generates the same files and links as the previous example in which you learned to use the URL= suboption with the BODY= and CONTENTS= file specifications. However, it is a bit simpler to specify the path only once in the PATH= option and to specify URL=NONE.

Additional Note If you plan to move your HTML files, you should specify URL=NONE with the PATH= option to prevent information from the PATH= option from creating URLs that are invalid or incorrect.

Example: PATH= Option without 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 in the Windows operating environment. The links from the frame file to the body and contents files contain the complete pathname, **c:\records\data.html** and **c:\records\toc.html:**

```
ods html path='c:\records'
         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;
```

Example: PATH= Option with a Specified URL

In the program below, 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 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

Style Definitions

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

- Banker
- BarrettsBlue
- Default
- HTMLblue
- Minimal.
- Statistical

General form, STYLE= option:

STYLE=*style-name*

where *style-name* is the name of a valid SAS or user-defined style definition.

Additional Note Don't enclose *style-name* in quotation marks.

Example

In the program below, the STYLE= option applies the Banker style to the output for the PROC PRINT step:

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

The following example shows PROC PRINT output with the Banker style applied.

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	MOB

8	M	35	70	173	MOD
9	M	34	73	154	LOW
10	F	49	64	172	LOW

Figure 9.17: PROC PRINT Output with the Banker Style Applied

Additional Note To view a full list of the available style definitions, click the Results tab on the Explorer window. Then right-click the **Results icon** and select **Templates** from the popup 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 Note Your site might have its own, customized, style definitions.

Additional Features

Customizing HTML Output

You've seen that you can use the STYLE= option to apply predefined styles to your HTML output. However, you might want to further customize your results.

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 online documentation for more information.

Chapter Summary

Text Summary

The Output Delivery System

The Output Delivery System (ODS) makes new report formatting options available in SAS. ODS separates your output into component parts so that the output can be sent to any ODS destination that you specify.

Opening and Closing ODS Destinations

Each ODS destination creates a different type of formatted output. By default, the HTML destination is open and SAS creates simple HTML output. Because an open destination uses system resources, it is a good idea to close the HTML destination if you don't need to create HTML output. Using ODS statements, you can create multiple output objects at the same time. When you have several ODS destinations open, you can close them all using the ODS _ALL_ CLOSE statement.

Creating HTML Output with Option

The HTML destination is open by default. However, you can use the ODS HTML statement to specify options. Use the BODY= or FILE= specification to create a custom named HTML body file containing procedure results. You can also use the ODS HTML statement to direct the HTML output from multiple procedures to the same HTML file.

Creating HTML Output with a Table of Contents

In order to manage multiple pieces of procedure output, you can use the CONTENTS= and FRAME = options with the ODS HTML statement to create a table of contents that links to your HTML output. The table of contents contains a heading for each procedure that creates output.

Specifying a Path for Output

You can also use the PATH= option to specify the directory where you want to store your HTML output. When you use the PATH= option, you don't need to specify the complete pathname for the body, contents, or frame files. By specifying the URL= suboption in the file specification, you can provide a URL that ODS uses in all the links that it creates to the file.

Changing the Appearance of HTML Output

You can change the appearance of your output using the STYLE= option in the ODS HTML statement. Several predefined styles are shipped with SAS.

Additional Features

ODS provides ways for you to customize HTML output using style definitions. Definitions are created using PROC TEMPLATE and describe how to render the HTML output or part of the HTML output.

Syntax

LIBNAME *libref'SAS-data-library'***;**

ODS LISTING CLOSE;

ODS HTML PATH=*file-specification*
<(URL=*'Uniform-Resource-Locator'* | NONE)>

BODY=*file-specification*

CONTENTS=*<file-specification>*

FRAME=*<file-specification>*

STYLE=*<style-name>*;

PROC PRINT DATA=*SAS-data-set*;

RUN;

ODS HTML CLOSE;

ODS LISTING;

Sample Program

```
libname clinic 'c:\data98\patients';
ods html path='c:\records'(url=none)
         body='data.html'
         contents='toc.html'
         frame='frame.html'
         style=brick;
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;
```

Points to Remember

- An open destination uses system resources. Therefore, it is a good idea to close the HTML destination before you create other types of output and reopen the HTML destination after you close all destinations destination.
- You do not need the ODS HTML CLOSE statement to create simple HTML output. However, if you want to create HTML with options, then use the ODS HTML CLOSE statement to close the HTML destination. The ODS HTML CLOSE statement is added *after* the RUN statement for the procedure.
- If you use the CONTENTS= and FRAME= options, open the frame file from within your Web browser to view the procedure output *and* the table of contents.

Chapter Quiz

Select the best answer for each question. After completing the quiz, you can check your answers using the answer key in the appendix.

1. Using ODS statements, how many types of output can you generate at once? ?

a. 1 (only LISTING output)

b. 2

c. 3

d. as many as you want

2. If ODS is set to its default settings, what types of output are created by the code below? ?

```
ods html file='c:\myhtml.htm';
ods pdf file='c:\mypdf.pdf';
```

a. HTML and PDF

b. PDF only

c. HTML, PDF, and LISTING

d. No output is created because ODS is closed by default.

3. What is the purpose of closing the HTML destination in the code shown below? ?

```
ods HTML close;
ods pdf …;
```

a. It conserves system resources.

b. It simplifies your program.

c. It makes your program compatible with other hardware platforms.

d. It makes your program compatible with previous versions of SAS.

4. When the code shown below is run, what will the file **`D:\Output\body.html`** contain? ?

```
ods html body='d:\output\body.html';
proc print data=work.alpha;
run;
proc print data=work.beta;
run;
ods html close;
```

a. The PROC PRINT output for Work.Alpha.

b. The PROC PRINT output for Work.Beta.

c. The PROC PRINT output for both Work.Alpha and Work.Beta.

d. Nothing. No output will be written to **`D:\Output\body.html`**.

5. When the code shown below is run, what file will be loaded by the links in **`D:\Output\contents.html?`** ?

```
ods html body='d:\output\body.html'
         contents='d:\output\contents.html'
         frame='d:\output\frame.html';
```

a. **`D:\Output\body.html`**

b. **`D:\Output\contents.html`**

c. **`D:\Output\frame.html`**

d. There are no links from the file **`D:\Output\contents.html.`**

6. The table of contents created by the CONTENTS= option contains a numbered heading for ?

a. each procedure.

b. each procedure that creates output.

c. each procedure and DATA step.

d. each HTML file created by your program.

7. When the code shown below is run, what will the file **D:\Output\frame.html** display? ?

```
ods html body='d:\output\body.html'
         contents='d:\output\contents.html'
         frame='d:\output\frame.html';
```

a. The file **D:\Output\contents.html.**

b. The file **D:\Output\frame.html.**

c. The files **D:\Output\contents.html** and **D:\Output\body.html.**

d. It displays no other files.

8. What is the purpose of the URL= suboptions shown below? ?

```
ods html body='d:\output\body.html' (url='body.html')
         contents='d:\output\contents.html'
        (url='contents.html')
         frame='d:\output\frame.html';
```

a. To create absolute link addresses for loading the files from a server.

b. To create relative link addresses for loading the files from a server.

c. To allow HTML files to be loaded from a local drive.

d. To send HTML output to two locations.

9. Which ODS HTML option was used in creating the following table? ?

Obs	Sex	Age	Height	Weight	ActLevel
1	M	27	72	168	HIGH
2	F	34	66	152	HIGH
3	F	31	61	123	LOW

a. `format=banker`

b. `format='bbanker'`

c. `style=banker`

d. `style='banker'`

10. What is the purpose of the PATH= option? ?

```
ods html path='d:\output' (url=none)
         body='body.html'
         contents='contents.html'
         frame='frame.html';
```

a. It creates absolute link addresses for loading HTML files from a server.

b. It creates relative link addresses for loading HTML files from a server.

c. It allows HTML files to be loaded from a local drive.

d. It specifies the location of HTML file output.

Answers

1. Correct answer: d

You can generate any number of output types as long as you open the ODS destination for each type of output you want to create.

2. Correct answer: b

HTML output is created by default in the SAS windowing environment for Microsoft Windows and UNIX, so these statements create HTML and PDF output.

3. Correct answer: a

By default, in the SAS windowing environment for Microsoft Windows and UNIX, SAS programs produce HTML output. If you want only RTF output, it's a good idea to close the HTML destination before creating RTF output, as an open destination uses system resources.

4. Correct answer: c

When multiple procedures are run while HTML output is open, procedure output is appended to the same body file.

5. Correct answer: a

The CONTENTS= option creates a table of contents containing links to the body file, **`D:\Output\body.html.`**

6. Correct answer: b

The table of contents contains a numbered heading for each procedure that creates output.

7. Correct answer: c

The FRAME= option creates an HTML file that integrates the table of contents and the body file.

8. Correct answer: b

Specifying the URL= suboption in the file specification provides a URL that ODS uses in the links it creates. Specifying a simple (one name) URL creates a relative link address to the file.

9. Correct answer: c

You can change the appearance of HTML output by using the STYLE= option in the ODS HTML statement. The style name doesn't need quotation marks.

10. Correct answer: d

You use the PATH= option to specify the location for HTML files to be stored. When you use the PATH= option, you don't need to specify the full path name for the body, contents, or frame files.