



临床研究SAS高级编程 ——ODS



Contents

- The output delivery system
- Opening and closing ODS destination
- Producing Dataset
- Producing HTML output
- Producing RTF output
- Producing PDF output



ODS Overview

The Output Delivery System (ODS) gives you greater flexibility in generating, storing, and reproducing SAS procedure and DATA step output with a wide range of formatting options. ODS provides formatting functionality that is not available from individual procedures or from the DATA step alone. ODS overcomes these limitations and enables you to format your output more easily. You can customize the presentation of your SAS output or you can customize the look of a single output object.

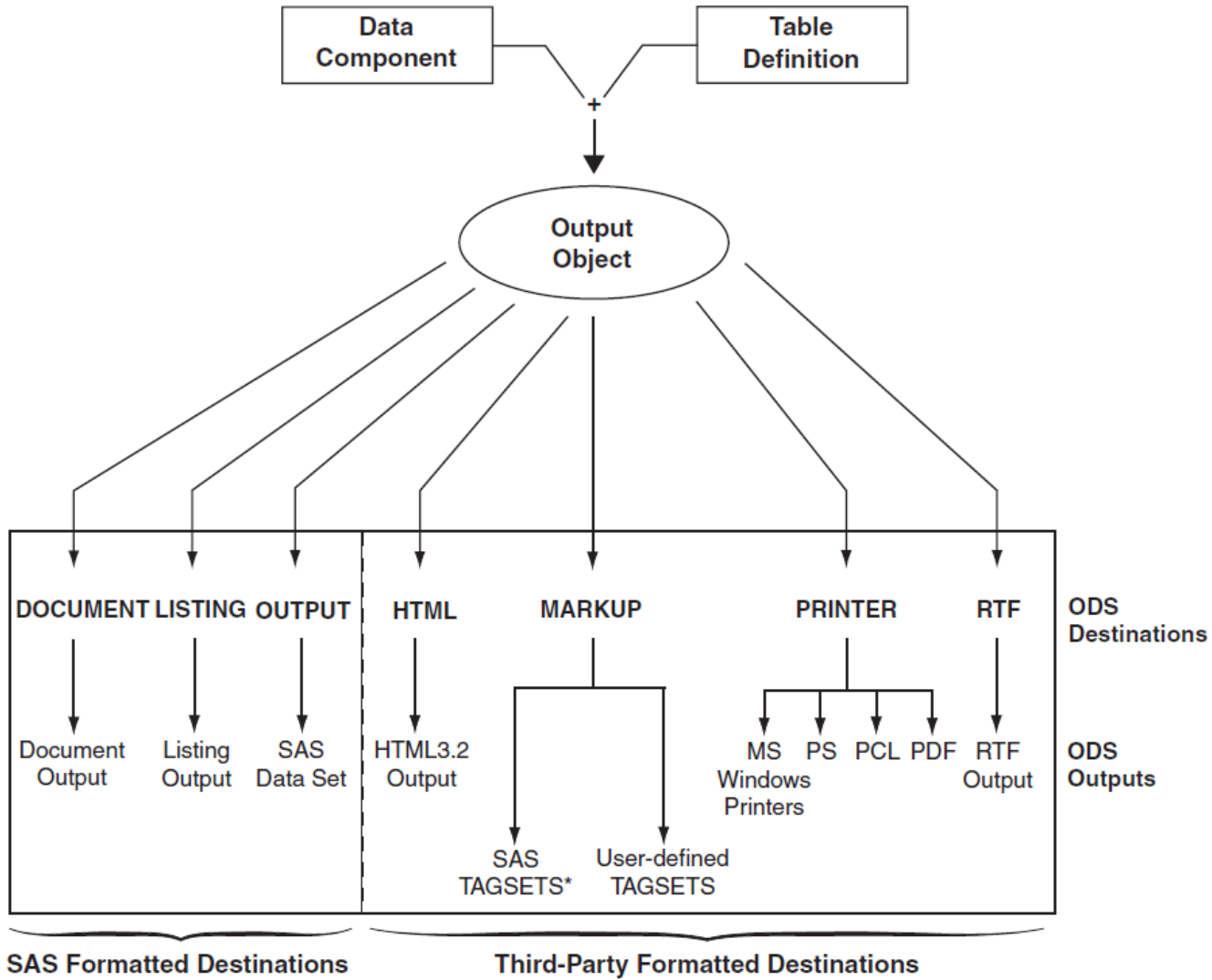
Prior to Version 7, most SAS procedures generated output that was designed for a traditional line-printer. This type of output has limitations that prevents you from getting the most value from your results:

Traditional SAS output is limited to monospace fonts. With today's desktop document editors and publishing systems, you need more versatility in printed output.

Some commonly used procedures do not produce output data sets. Prior to ODS, if you wanted to use output from one of these procedures as input to another procedure, then you relied on PROC PRINTTO and the DATA step to retrieve results.



ODS Output

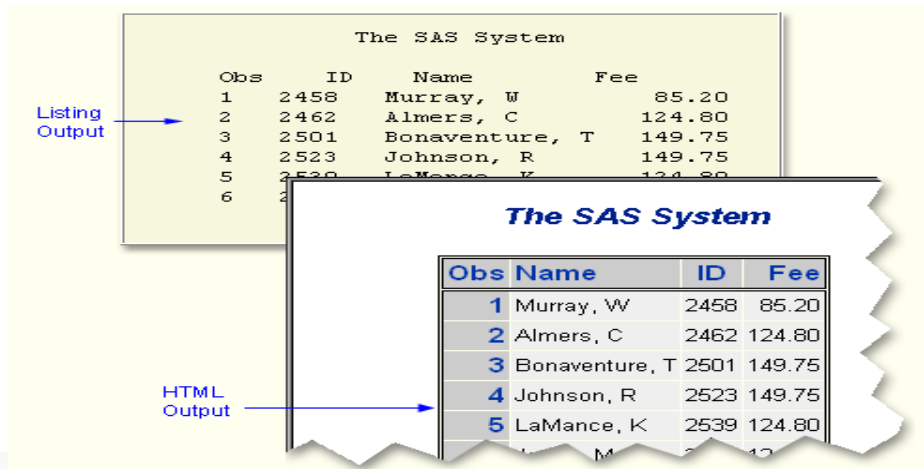




The Output Delivery System (1)

Introduction

- ▶ When you set options to create HTML output, SAS uses **Output Delivery System (ODS)** statements to generate the output.
- ▶ Using ODS, you can create, customize, and manage HTML output.
- ▶ By submitting programming statements. After you create HTML files, you can view them using Internet Explorer, Netscape Navigator, or any Web browser that fully supports HTML3.2.



The Output Delivery System (2)

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

Obs	ID	Name	Fee
1	2458	Murray, W	85.20
2	2462	Almers, C	124.80
3	2501	Bonaventure, T	149.75
4	2523	Johnson, R	149.75
5	2539	LaMance, K	124.80
6	2544	Jones, M	124.80



The Output Delivery System (3)

┌ an output data set of procedure results

VarName	Quantile	Estimate
RestHR	100% Max	80
RestHR	99%	80
RestHR	95%	79
RestHR	90%	78
RestHR	75% Q3	76
RestHR	50% Median	72
RestHR	25% Q1	70
RestHR	10%	68

┌ traditional SAS listing output

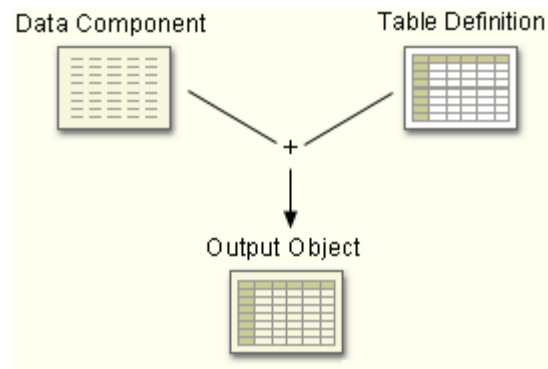
The SAS System			
Obs	ID	Name	Fee
1	2458	Murray, W	85.20
2	2462	Almers, C	124.80
3	2501	Bonaventure, T	149.75
4	2523	Johnson, R	149.75
5	2539	LaMance, K	124.80
6	2544	Jones, M	124.80

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

The Output Delivery System (4)

How ODS works

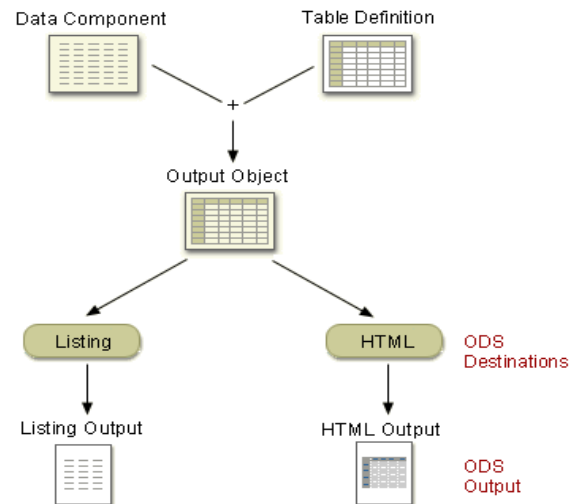
- ▶ When you submit your ODS statements and the SAS program that creates your output, ODS does the following:
 - ┌ ODS creates your output in the form of **output objects**.
Each output object contains the results of a procedure or DATA step (the **data component**) and can also contain information about how to render the results (the **table definition**).



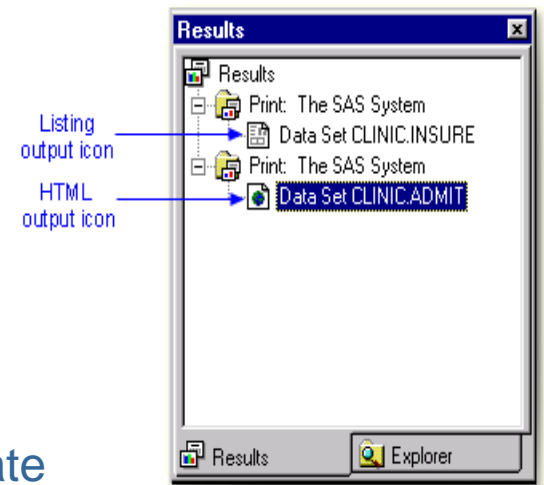


The Output Delivery System (5)

- 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.



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



Opening and Closing ODS Destinations (1)

ODS destinations

- ▶ You use ODS statements to specify destinations for your output and 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)
Listing	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 Microsoft Word

Opening and Closing ODS Destinations (2)

Using statements to open and close destinations

- ▶ For each type of formatted output 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:

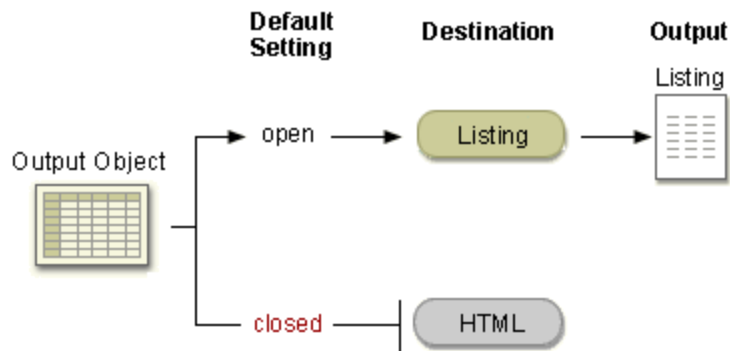
```
ODS open-destination;  
ODS close-destination close;
```

where

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

Opening and Closing ODS Destinations (3)

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



Opening and Closing ODS Destinations (4)

Example

- ▶ The following program creates SAS listing output because the listing destination is open by default. No other ODS destinations are open, so no other output formats are produced.

```
Proc print data=sasuser.mydata;  
Run;
```

- ▶ The following program produces HTML **and** Listing output:

```
Ods html body='c:\mydata.html';  
Proc print data=sasuser.mydata;  
Run;  
Ods html close;
```

Opening and Closing ODS Destinations (5)

Closing the listing destinations

- ▶ Because open destinations use system resources, it's a good idea to close the listing destinations at the **beginning** of your program if you don't want to produce listing output. For example:

```
ods listing close;
```

- ▶ The listing 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 listing output (the default setting) at the end of your programs. For example:

```
ods listing;
```

▶ Example

The following program produces only HTML output:

```
ods listing close;  
ods html body='c:\mydata.html' ;  
Proc print data=sasuser.mydata;  
Run;  
ods html close;  
ods listing;
```

Opening and Closing ODS Destinations (6)

- Closing multiple ODS destinations at once
 - ▶ One of the features of ODS is that 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 HTML and PDF destinations before the PROC step and closes all ODS destinations at the end of the program.

```
ods html file='HTML-file-pathname' ;  
ods html file='PDF-file-pathname' ;  
Proc print data=sasuser.mydata ;  
Run ;  
ods _all_ close ;  
ods listing ;
```

Opening and Closing ODS Destinations (7)

Quiz

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

► 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.

Opening and Closing ODS Destinations (8)

Quiz

What output formats are generated by the following program?

```
ods listing close;  
ods html body='c:\myfile.htm';  
proc print data=sasuser.mydata;  
run;  
ods html close;  
ods listing;
```

- a. HTML and Listing output
- b. HTML output only
- c. Listing output only

► Correct answer: b

! The ODS LISTING CLOSE statement turns off listing output and the ODS HTML statement turns on HTML output. ←



Producing Dataset

- ODS output name=dataset;





Producing HTML Output

Contents

- ▶ Creating simple HTML output
- ▶ Creating HTML output with a table of contents
- ▶ Using options to specify links and paths
- ▶ Changing the appearance of HTML output





Creating Simple HTML Output (1)

- To create HTML output, 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

- ▶ An HTML filename (include the complete pathname if you want to save the HTML file to a specific location)
- ▶ A fileref (file shortcut) that has been assigned to an HTML file.
- ▶ A SAS catalog entry in the form **entry-name.html**
- ▶ **FILE=** can also be used to specify the file that contains the HTML output. **FILE=** is an alias for **BODY=**.



Creating Simple HTML Output (2)

Example

The program below creates PROC PRINT output in an HTML output. The **BODY=** option specifies the file **F:\admit.html** in the windows operation environment as the file that contains the PROC PRINT result.

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

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



Creating Simple HTML Output (3)

Creating HTML output from multiple procedure

- ▶ You can also use the ODS HTML statements 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 listing close;
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 listing;
```



Creating Simple HTML Output (4)

- The following is a representation of the HTML file containing the results from the program above. Notice that the results from each procedure are appended.

The SAS System

Obs	ID	Sex	Age	Height	Weight	Activity Level
1	2458	M	27	72	168	HIGH
2	2462	F	34	66	152	HIGH
3	2501	F	31	61	123	LOW
4	2523	F	43	63	137	MOD
5	2539	M	51	71	158	LOW
6	2544	M	29	76	193	HIGH
7	2552	F	32	67	151	MOD
8	2555	M	35	70	173	MOD
9	2563	M	34	73	154	LOW
10	2568	F	49	64	172	LOW
11	2571	F	44	66	140	HIGH
12	2572	F	28	62	118	LOW
13	2574	M	30	69	147	MOD
14	2575	F	40	69	163	HIGH
15	2578	M	47	72	173	MOD
16	2579	M	60	71	191	LOW
17	2584	F	43	65	123	MOD
18	2586	M	25	75	188	HIGH
19	2588	F	22	63	139	LOW
20	2589	F	41	67	141	HIGH
21	2595	M	54	71	183	MOD

The SAS System

	RestHR	MaxHR	RecHR
Min	65.00	152.00	108.00
Mean	72.95	171.10	128.95





Creating HTML Output with a Table of Contents (1)

- To create an HTML file that has a table of contents with links to the output of each specific procedure. You can specify 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=.
- To direct the HTML output to a specific storage location, specify the complete pathname of the HTML file in the file-specification.



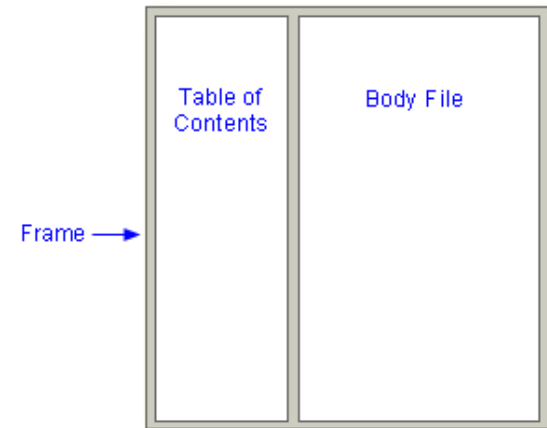
Creating HTML Output with a Table of Contents (2)

Example

In the program below,

- ▶ the **BODY=** specification creates **data.html** in the **c:\records** directory.
- ▶ the **CONTENTS=** specification creates **toc.html** in the **c:\records** directory.
- ▶ the **FRAME=** specification creates **frame.html** in the **c:\records** directory.

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



Creating HTML Output with a Table of Contents (3)

Viewing frame files

- ▶ The **Results window** does not display links to frame files, and 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.

File C:\Records\frame.html

The screenshot shows a web browser window titled "File C:\Records\frame.html". The window is divided into two frames. The left frame, titled "Table of Contents", contains a list of items: "I. The Print Procedure" and "II. The Print Procedure". A link labeled "Data Set CLINIC.ADMIT" is visible under the first item. The right frame, titled "The SAS System", contains a table with the following data:

Obs	ID	Sex	Age	Height	Weight	Activity Level
1	2458	M	27	72	168	HIGH
2	2462	F	34	66	152	HIGH
3	2501	F	31	61	123	LOW
4	2523	F	43	63	137	MOD
5	2539	M	51	71	158	LOW
6	2544	M	29	76	193	HIGH
7	2552	F	32	67	151	MOD

Below the screenshot, three blue arrows point to the frame titles with labels: "FRAME= frame.html" points to the left frame, "CONTENTS= toc.html" points to the right frame, and "BODY= data.html" points to the right frame.

Creating HTML Output with a Table of Contents (4)

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.

Table of Contents	
I. The Print Procedure	
Data Set	
CLINIC.ADMIT	
II. The Print Procedure	

Obs	ID
1	2458
2	246
3	2501
4	2523
5	2539
6	2544
7	2552

Creating HTML Output with a Table of Contents (5)

Quiz

Which file do you need to open to view both the table of contents and the body file created by the program below?

```
ods listing close;
ods html body='c:\sales\seating.html'
      contents='c:\sales\contents.html'
      frame='c:\sales\onorder.html';
proc print data=orders.sofas;
run;
proc print data=orders.chairs;
run;
ods html close;
ods listing;
```

- a. seating.html
- b. contents.html
- c. onorder.html

Correct answer: c

Open the frame file, onorder.html, to view the procedure output and the table of contents.



Creating HTML Output with a Table of Contents (6)

Quiz

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.

Correct answer: b

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



Using Options to Specify Links and Paths (1)

- When ODS generates HTML files for the body, contents, and frame, it also generates links between the file 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 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>
```

Using Options to Specify Links and Paths (2)

- These links work when you are viewing the HTML files locally, but if you wanted 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.
- The URL= suboption is useful for building HTML files that may 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.

Using Options to Specify Links and Paths (3)

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 and 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>
```


Using Options to Specify Links and Paths (4)

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>
```

- 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.

Using Options to Specify Links and Paths (5)

The PATH= Option

- ▶ To streamline your ODS HTML statement, you can also use the PATH= option to specify the location where you want to store your HTML output, and you can use the **URL=NONE** to prevent ODS from using the pathname in any links 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.

Using Options to Specify Links and Paths (6)

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 listing close;
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 listing;
```

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's a bit simpler to specify the path only once in the PATH= option and to specify URL=NONE.

- ▶ 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.

Using Options to Specify Links and Paths (7)

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 listing close;
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 listing;
```

Using Options to Specify Links and Paths (8)

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 listing close;
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 listing;
```

Using Options to Specify Links and Paths (9)

Quiz

- ▶ Which program saves the body, contents, and frame files to the directory **C:\Sales** in the Windows environment?

a.ods listing close;

```
ods html body='c:\sales\seating.html'  
      contents='c:\sales\contents.html'  
      frame='c:\sales\onorder.html';
```

```
proc print data=orders.sofas;  
run;  
proc print data=orders.chairs;  
run;  
ods html close;  
ods listing;
```

b.ods listing close;

```
ods html path='c:\sale'  
      body='seating.html'  
      contents='contents.html'  
      frame='onorder.html';
```

```
proc print data=orders.sofas;  
run;  
proc print data=orders.chairs;  
run;  
ods html close;  
ods listing;
```

c. both programs

- ▶ Correct answer: c

You can use full path names, as shown in the first program, or the PATH= option, as shown in the second program, to specify the destination for the body, contents, and frame files.



Changing the Appearance of HTML Output (1)

- 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

- ▶ Beige
- ▶ Brick
- ▶ Brown
- ▶ D3d
- ▶ Default
- ▶ Minimal

- General form, **STYLE=** option:

STYLE=style-name

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

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

Changing the Appearance of HTML Output (2)

Example: Style=Beige

```
ods html body='c:\records\data.html'  
      style=beige;  
proc print data=work.data;  
run;  
ods html close;
```

Obs	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee
1	2458	Murray, W	M	27	1	72	168	HIGH	85.20
2	2462	Almers, C	F	34	3	66	152	HIGH	124.80
3	2501	Bonaventure, T	F	31	17	61	123	LOW	149.75
4	2523	Johnson, R	F	43	31	63	137	MOD	149.75
5	2539	LaMance, K	M	51	4	71	158	LOW	124.80
6	2544	Jones, M	M	29	6	76	193	HIGH	124.80
7	2552	Reberson, P	F	32	9	67	151	MOD	149.75
8	2555	King, E	M	35	13	70	173	MOD	149.75
9	2563	Pitts, D	M	34	22	73	154	LOW	124.80
10	2568	Eberhardt, S	F	49	27	64	172	LOW	124.80

Changing the Appearance of HTML Output (3)

Example: Style=Brick

```
ods html body='c:\records\data.html'  
      style=brick;  
proc print data=work.data;  
run;  
ods html close;
```

Obs	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee
1	2458	Murray, W	M	27	1	72	168	HIGH	85.20
2	2462	Almers, C	F	34	3	66	152	HIGH	124.80
3	2501	Bonaventure, T	F	31	17	61	123	LOW	149.75
4	2523	Johnson, R	F	43	31	63	137	MOD	149.75
5	2539	LaMance, K	M	51	4	71	158	LOW	124.80
6	2544	Jones, M	M	29	6	76	193	HIGH	124.80
7	2552	Reberson, P	F	32	9	67	151	MOD	149.75
8	2555	King, E	M	35	13	70	173	MOD	149.75
9	2563	Pitts, D	M	34	22	73	154	LOW	124.80
10	2568	Eberhardt, S	F	49	27	64	172	LOW	124.80

Changing the Appearance of HTML Output (4)

Example: Style=Brown

```
ods html body='c:\records\data.html'  
      style=brown;  
proc print data=work.data;  
run;  
ods html close;
```

Obs	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee
1	2458	Murray, W	M	27	1	72	168	HIGH	85.20
2	2462	Almers, C	F	34	3	66	152	HIGH	124.80
3	2501	Bonaventure, T	F	31	17	61	123	LOW	149.75
4	2523	Johnson, R	F	43	31	63	137	MOD	149.75
5	2539	LaMance, K	M	51	4	71	158	LOW	124.80
6	2544	Jones, M	M	29	6	76	193	HIGH	124.80
7	2552	Reberson, P	F	32	9	67	151	MOD	149.75
8	2555	King, E	M	35	13	70	173	MOD	149.75
9	2563	Pitts, D	M	34	22	73	154	LOW	124.80
10	2568	Eberhardt, S	F	49	27	64	172	LOW	124.80

Changing the Appearance of HTML Output (5)

Example: Style=D3D

```
ods html body='c:\records\data.html'  
      style=d3d;  
proc print data=work.data;  
run;  
ods html close;
```

Obs	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee
1	2458	Murray, W	M	27	1	72	168	HIGH	85.20
2	2462	Almers, C	F	34	3	66	152	HIGH	124.80
3	2501	Bonaventure, T	F	31	17	61	123	LOW	149.75
4	2523	Johnson, R	F	43	31	63	137	MOD	149.75
5	2539	LaMance, K	M	51	4	71	158	LOW	124.80
6	2544	Jones, M	M	29	6	76	193	HIGH	124.80
7	2552	Reberson, P	F	32	9	67	151	MOD	149.75
8	2555	King, E	M	35	13	70	173	MOD	149.75
9	2563	Pitts, D	M	34	22	73	154	LOW	124.80
10	2568	Eberhardt, S	F	49	27	64	172	LOW	124.80

Changing the Appearance of HTML Output (6)

Example: Style=Default

```
ods html body='c:\records\data.html'  
      style=default;  
proc print data=work.data;  
run;  
ods html close;
```

The STYLE= specification is optional when you want to produce HTML output in the default style.

Obs	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee
1	2458	Murray, W	M	27	1	72	168	HIGH	85.20
2	2462	Almers, C	F	34	3	66	152	HIGH	124.80
3	2501	Bonaventure, T	F	31	17	61	123	LOW	149.75
4	2523	Johnson, R	F	43	31	63	137	MOD	149.75
5	2539	LaMance, K	M	51	4	71	158	LOW	124.80
6	2544	Jones, M	M	29	6	76	193	HIGH	124.80
7	2552	Reberson, P	F	32	9	67	151	MOD	149.75
8	2555	King, E	M	35	13	70	173	MOD	149.75
9	2563	Pitts, D	M	34	22	73	154	LOW	124.80
10	2568	Eberhardt, S	F	49	27	64	172	LOW	124.80

Changing the Appearance of HTML Output (7)

Example: Style=Minimal

```
ods html body='c:\records\data.html'  
      style=minimal;  
proc print data=work.data;  
run;  
ods html close;
```

Obs	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee
1	2458	Murray, W	M	27	1	72	168	HIGH	85.20
2	2462	Almers, C	F	34	3	66	152	HIGH	124.80
3	2501	Bonaventure, T	F	31	17	61	123	LOW	149.75
4	2523	Johnson, R	F	43	31	63	137	MOD	149.75
5	2539	LaMance, K	M	51	4	71	158	LOW	124.80
6	2544	Jones, M	M	29	6	76	193	HIGH	124.80
7	2552	Reberson, P	F	32	9	67	151	MOD	149.75
8	2555	King, E	M	35	13	70	173	MOD	149.75
9	2563	Pitts, D	M	34	22	73	154	LOW	124.80
10	2568	Eberhardt, S	F	49	27	64	172	LOW	124.80

Changing the Appearance of HTML Output (8)

- 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 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.
- Your site might have its own, customized, style definitions.

Changing the Appearance of HTML Output (9)

Quiz

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

Obs	ID	Name	Sex	Age	Height	Weight
1	2458	Murray, W	M	27	72	168
2	2462	Almers, C	F	34	66	152
3	2501	Bonaventure, T	F	31	61	123
4	2523	Johnson, R	F	43	63	137

- a. format=brown
- b. format='brown'
- c. style=brown
- d. style='brown'

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.





Producing RTF Output (1)

Introduction

- ▶ RTF stands for Rich Text Format, which is Microsoft's open markup language for defining the content of documents.
- ▶ If you have an RTF file, simply clicking on it within a Microsoft Windows environment causes Microsoft Word or WordPad to open the file.
- ▶ There are several ways to create RTF files, and thus Microsoft Word files, with SAS.



Producing RTF Output (2)

● Sending output to the ODS RTF destination

- ▶ Turning on the SAS ODS RTF destination is the easiest way to send your SAS output to an RTF file. With the following SAS code you can create an RTF file:

```
ods rtf file = "demog_table.rtf";  
  proc report data = demog;  
    etc.  
  run;  
ods rtf close;
```



Producing RTF Output (3)

- you can use a different style template supplied by SAS by specifying the **STYLE=** option in the ODS RTF statement. you can define your own ODS style template with **PROC TEMPLATE**.
- Although ODS RTF provides a convenient and efficient way to create attractive SAS output, it has a couple of drawbacks:
 - ▶ **SAS TITLE** and **FOOTNOTE** statements translate into Microsoft Word Header and Footer sections when you use ODS RTF. This is fine unless you have a number of ODS RTF files that you need to integrate into a single document, in which case the headers and footers may conflict with one another in your integrated document. You can use the **ODS RTF BODYTITLE** option to get your titles out of the Microsoft Word header and down into the page space, but that only works if your tables and listings are only one page long. Also, if you use the **BODYTITLE** option with multi-page output, your titles appear on the first page and your footnotes appear on the last page of your output, which is probably not what you want.
 - ▶ PROC REPORT does not use all of its features when you use the RTF destination. For instance, ORDER and GROUP variables are not repeated at the tops of subsequent pages.





RTF Titles and Footnotes

- **BOLD** : bolds the title or footnote
- **COLOR=** supplies the color of the title or footnote
- **BCOLOR=** supplies the background color of the title or footnote
- **FONT=** supplies the font for the title or footnote
- **HEIGHT=** supplies the point size for the title or footnote
- **JUSTIFY=** left-justifies, right-justifies, or centers the title or footnote
- **LINK=** specifies the hyperlink to use for the title or footnote
- **UNDERLIN=** specifies an underline

title bold "this is a bold title";

title2 justify=l "this title is left justified";

title3 link="www.sas.com" "this is my hyperlink";



Enhancing Titles, Footnotes, and More with ESCAPECHAR=

```
ODS escapechar='\';  
proc print data=sashelp.class;  
  title 'this value is superscripted {\super 2} ';  
  title 'this value is subscripted {\sub 2} ';  
run;
```



Adding Images

```
ODS escapechar='^';  
proc print data=sashelp.class;  
  title j=l '^S={preimage="C:\WINDOWS\Desktop\V7image.jpg"} a test';  
run;
```



Inserting Destination-Specific Raw Text

```
ods escapechar='\';  
proc print data=sashelp.class;  
  title '\R/RTF"\tab\tab\tab\tab" this is a test';  
run;
```

```
ods escapechar='\';  
proc print data=sashelp.class;  
  title '\R/RTF"\highlight5" this is a test ';  
run;
```

```
ods escapechar='\';  
proc print data=sashelp.class;  
  title ' more text \R/RTF"\line\ql" more';  
run;  
ods rtf close;
```



Adding RTF Control Words

Style	RTF Control Word	Example Code
Italicize	<code>\i</code>	<code>title '\i italicized title';</code>
Underline	<code>\ul</code>	<code>title '\ul underline title';</code>
Double underline	<code>\dul</code>	<code>title '\dul title';</code>
New line	<code>\line</code>	<code>title 'this is the first \line this is the second ';</code>
Bullet	<code>\bullet</code>	<code>title '\bullet bullet preceding title';</code>
Emboss	<code>\embo</code>	<code>title '\embo embossed title';</code>
Engrave	<code>\impr</code>	<code>title '\impr engraved title';</code>
Subscript	<code>\sub</code>	<code>title 'This is a subscript T\sub 1';</code>
Superscript	<code>\super</code>	<code>title 'This is a subscript T\super 2';</code>
Outline	<code>\outl</code>	<code>title '\outl This is outlined';</code>
Shadow	<code>\shad</code>	<code>title '\shad This is shadowed';</code>
Strike	<code>\strike</code>	<code>title '\strike This is striked';</code>
double strike	<code>\strikedl</code>	
dotted underline	<code>\uld</code>	<code>title '\uld dotted underline';</code>
Wave underline	<code>\ulw</code>	<code>title '\ulw wave underline';</code>
Thick underline	<code>\ulth</code>	<code>title '\ulth thick underline';</code>
foreground color	<code>\cfn</code>	<code>title '\cfn2 foreground color';</code>
Font size in half points	<code>\fs24</code>	<code>title '\fs40 fonts increased';</code>
Highlight	<code>\highlightV</code>	<code>title '\highlight2';</code>
Bold	<code>\b</code>	<code>title '\b bold title';</code>
Left aligned	<code>\ql</code>	<code>title '\ql left aligned.</code>
Right aligned	<code>\qr</code>	<code>title '\qr right aligned.</code>
Centered	<code>\qc</code>	<code>title '\qc left aligned.</code>



Producing PDF Output (1)

Introduction

- ▶ PDF stands for Portable Document Format, which is an open document description language created by Adobe Systems Inc.
- ▶ PDF files have several advantages over RTF and ASCII text files:
 - ┌ PDF files can represent graphics as well as text, which ASCII cannot do.
 - ┌ PDF files can be read with the free Adobe Reader. You do not have to purchase a word processing package to open PDF files.
 - ┌ Large PDF files are stable. This cannot be said of Microsoft Word documents, which become unstable when files get large.
 - ┌ PDF files look the same no matter where you open them. Sometimes Microsoft Word documents can change in appearance depending on your printer or whether you are printing with a PostScript or PCL printer driver.
 - ┌ PDF files are the standard for electronic submission of documents to the FDA.



Producing PDF Output (2)

● Converting a single SAS output to PDF

- ▶ Turning on the SAS ODS PDF destination is the easiest way to send your SAS output to a PDF file. With the following code you can create a PDF file:

```
ods pdf file = "demog_table.pdf";  
proc report data = demog;  
    etc.  
  
run;  
ods pdf close;
```



Producing PDF Output (3)

- you can use a different style template supplied by SAS by specifying the **STYLE=** option in the ODS PDF statement. And you can define your own ODS style template with **PROC TEMPLATE**.
- Finally, PROC REPORT allows you to define formatting elements within the SAS procedure itself, where you can customize the appearance of almost any element on the page.
- Although ODS PDF provides a convenient and efficient way to create attractive SAS output, **PROC REPORT** does not use all of its features when you use the PDF destination. For instance, **ORDER** and **GROUP** variables are not repeated at the tops of subsequent pages when you send PROC REPORT output to the PDF destination.

