Super Demo Overview

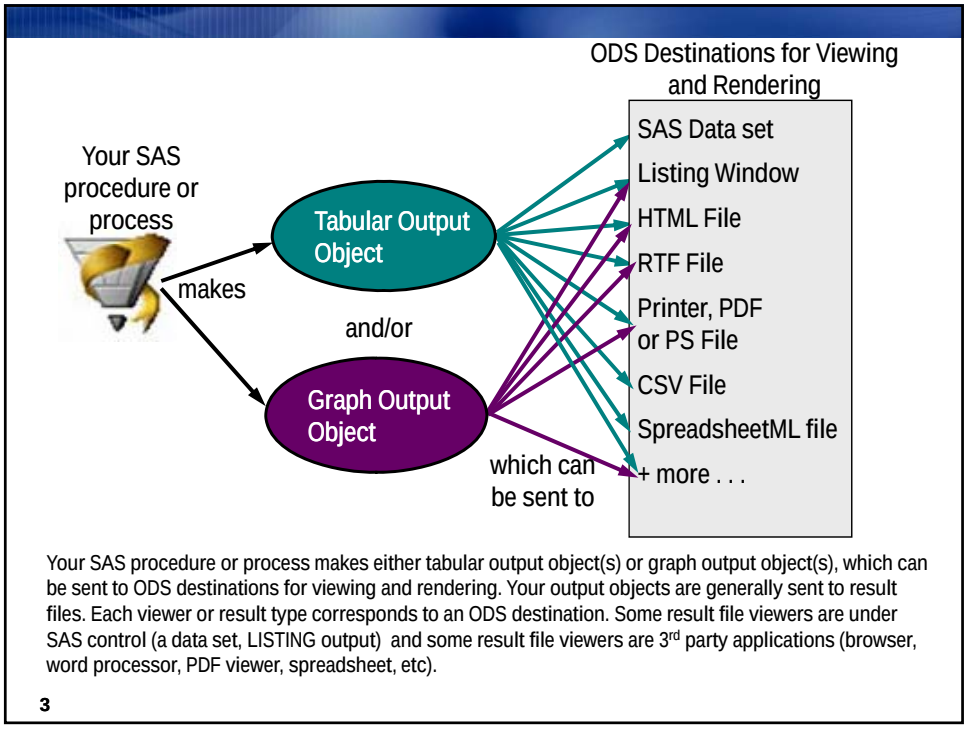
Chapter 1 Introduction to the Output Delivery System

- 1.1 Introduction
- 1.2 Sending a Report to an ODS Destination

1

Getting Started with the Output Delivery System

2

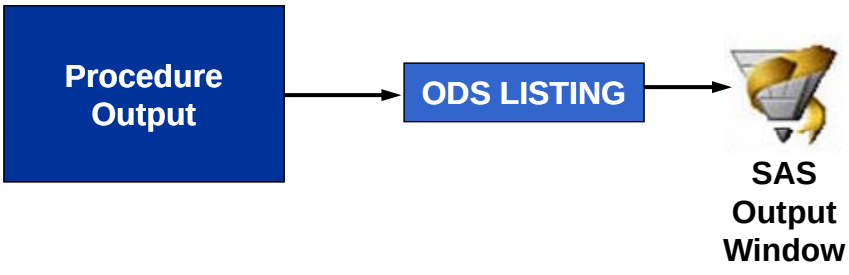


Your SAS procedure or process makes either tabular output object(s) or graph output object(s), which can be sent to ODS destinations for viewing and rendering. Your output objects are generally sent to result files. Each viewer or result type corresponds to an ODS destination. Some result file viewers are under SAS control (a data set, LISTING output) and some result file viewers are 3rd party applications (browser, word processor, PDF viewer, spreadsheet, etc).

3

Sending a Report to the SAS Output Window

The LISTING destination creates output that is placed automatically in the SAS Output Window. In the SAS windowing environment, the SAS Output window is always open unless you close it.



4

Getting Started with the Output Delivery System

5

Basic ODS Syntax Model

You use the same syntax model to open and close ODS destinations in order to create output files.

ODS *destination* FILE = "*filename.ext***" STYLE=*egdefault*;**

SAS code to generate a report

ODS *destination* CLOSE;

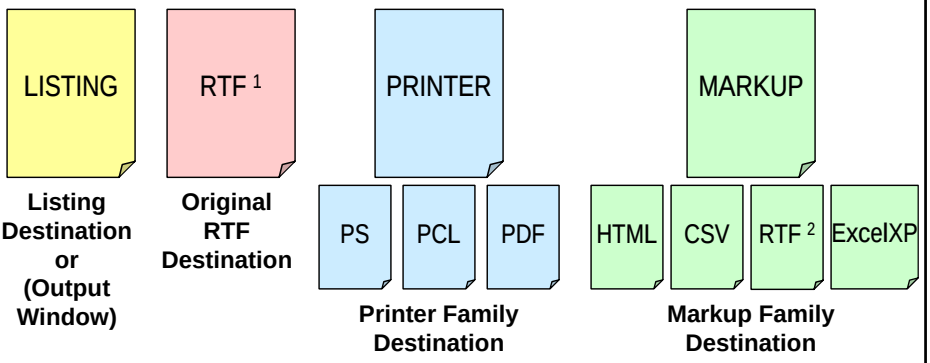
6

ODS Invocation and Destinations

```
ods destination file='filename.ext';

... SAS code ...

ods destination close;
```



Listing Destination or (Output Window)

Original RTF Destination

Printer Family Destination

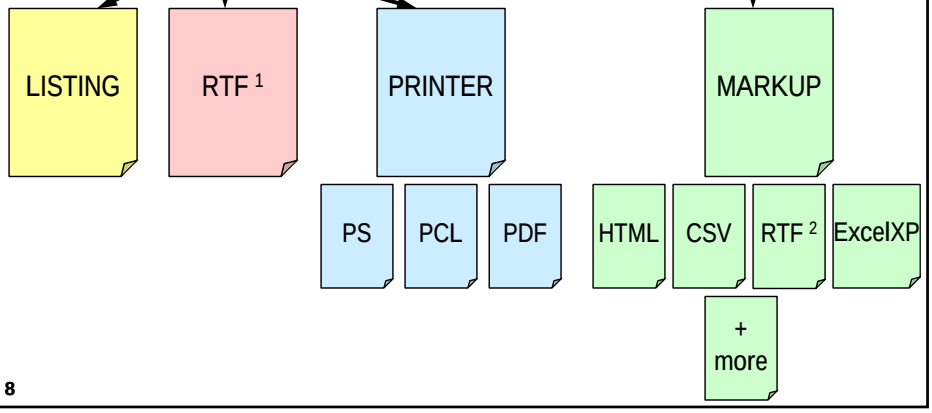
Markup Family Destination

7

ODS Invocation and Destinations

For these destinations, the method of creating the result file is hardcoded in the SAS executable program.

For MARKUP destinations, the method of creating the result file is contained in a TAGSET template.



8

Creating Multiple Output Files

Procedure output can be sent to many destination output files at the same time.

```
ods listing;
ods msoffice2k file='report.html';
ods csvall file='report.csv';

proc tabulate ... ;
  ...additional SAS statements...
run;

ods msoffice2k close;
ods csvall close;
```

9

Creating Multiple Output Files

Procedure output can be sent to many destination output files at the same time.

```
ods listing;❶
ods msoffice2k file='report.html';❷
ods csvall file='report.csv';❸

proc tabulate ... ;
  ...additional SAS statements...
run;

ods msoffice2k close;
ods csvall close;
```

Three destinations are being sent the output from one invocation of the TABULATE procedure.

10

Closing Open Destinations

Every open destination that writes an external file must be closed.

```
ods listing;
ods msoffice2k file='report.html';
ods csvall file='report.csv';

proc tabulate ... ;
  ...additional SAS statements...
run;

ods msoffice2k close;
ods csvall close;
```

The LISTING destination in this example does not create an external file, so it does not need an ODS CLOSE statement. Only the two non-LISTING destinations need to be closed.

11

Alternate Destination Close Method

There is an alternate method to close all open ODS destination files.

```
ods listing;
ods msoffice2k file='report.html';
ods csvall file='report.csv';

proc tabulate ... ;
  ...additional SAS statements...
run;

ods _all_ close;
```

This statement closes all open destinations including the LISTING destination.

12

Alternate Destination Close Method

There is an alternate method to close all open ODS destination files.

```
ods listing;
ods msoffice2k file='report.html';
ods csvall file='report.csv';

proc tabulate ... ;
  ...additional SAS statements...
run;

ods _all_ close;
ods listing;
```

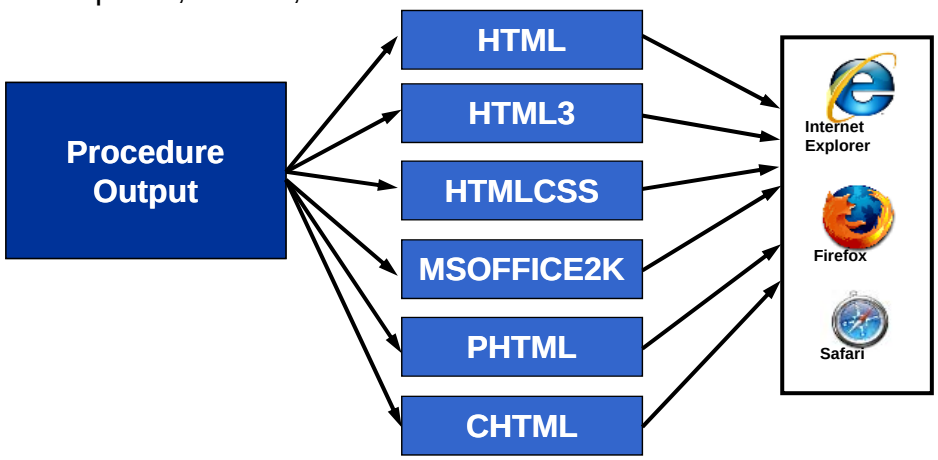
This statement closes all open destinations including the LISTING destination.

This statement reopens the LISTING destination to be ready for the next program submission.

13

Sending a Report to a Web Browser

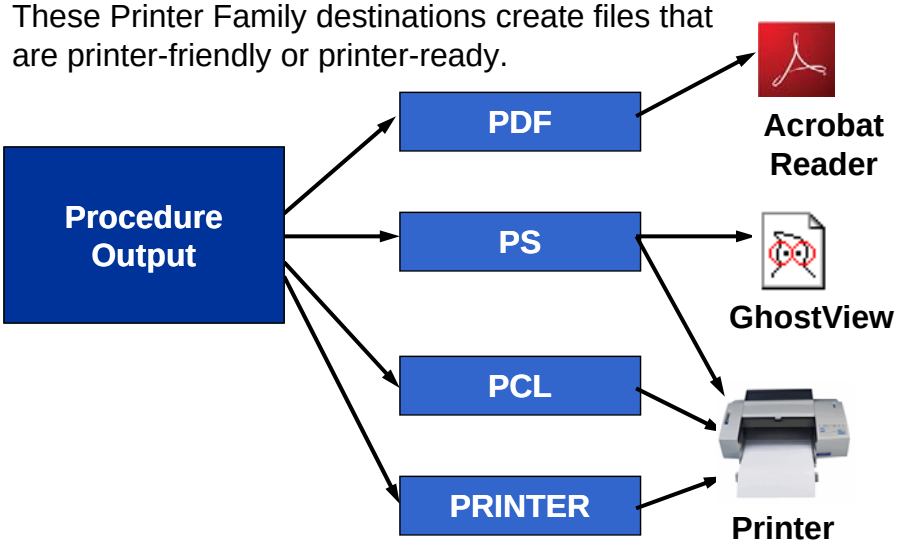
These HTML-based ODS destinations create files that can be opened in a Web browser, such as Internet Explorer, Firefox, or Safari.



14

Sending a Report to a Print-Related Destination

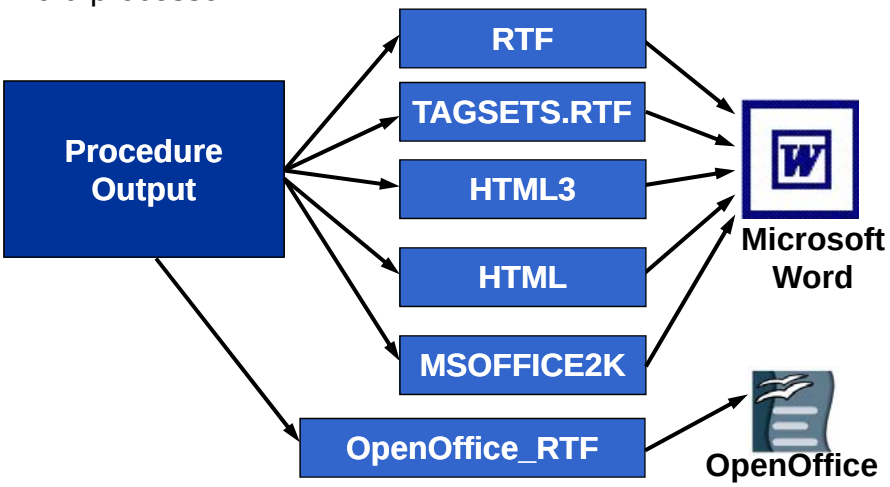
These Printer Family destinations create files that are printer-friendly or printer-ready.



15

Sending a Report to a Word Processor

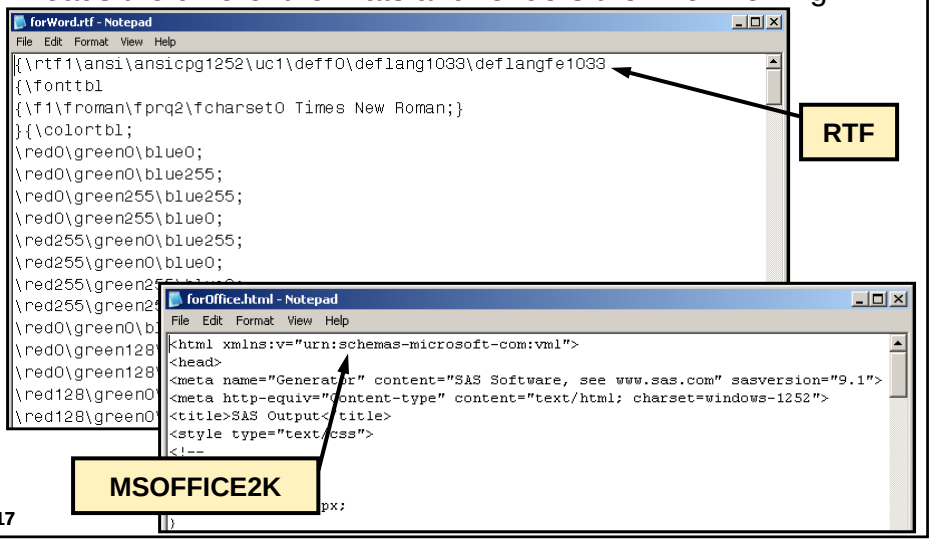
These destinations create files that can be opened in a word processor.



16

How ODS and Word Work Together

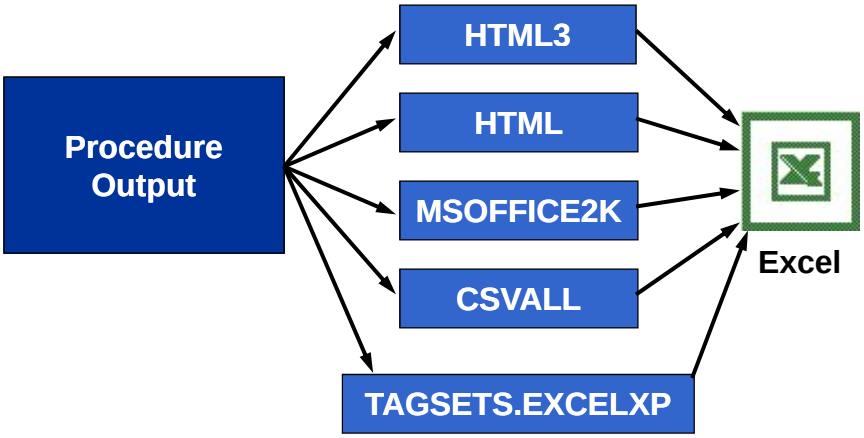
The ODS output file is not a Microsoft Word file. Word reads the different formats and renders them for viewing.



17

Sending a Report to Microsoft Excel

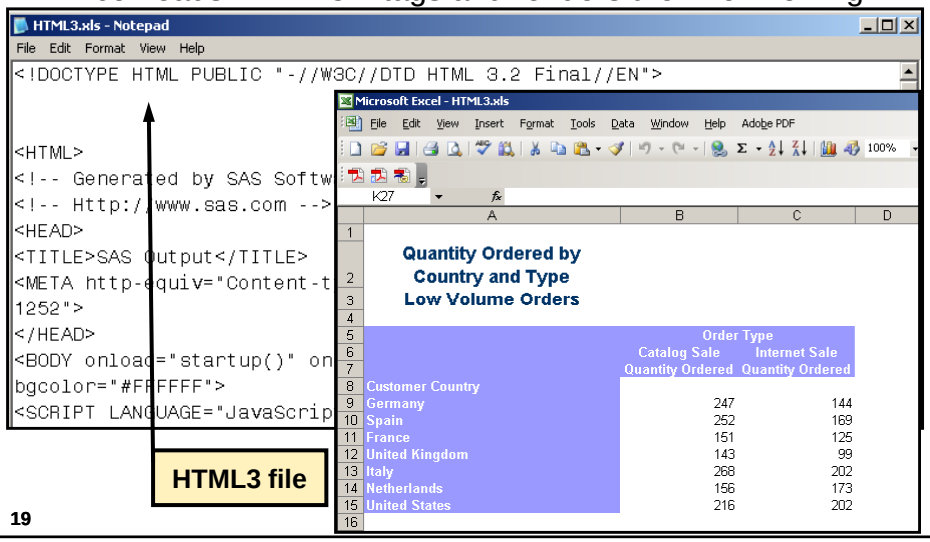
These destinations create files that can be opened in Excel.



18

How ODS and Excel Work Together

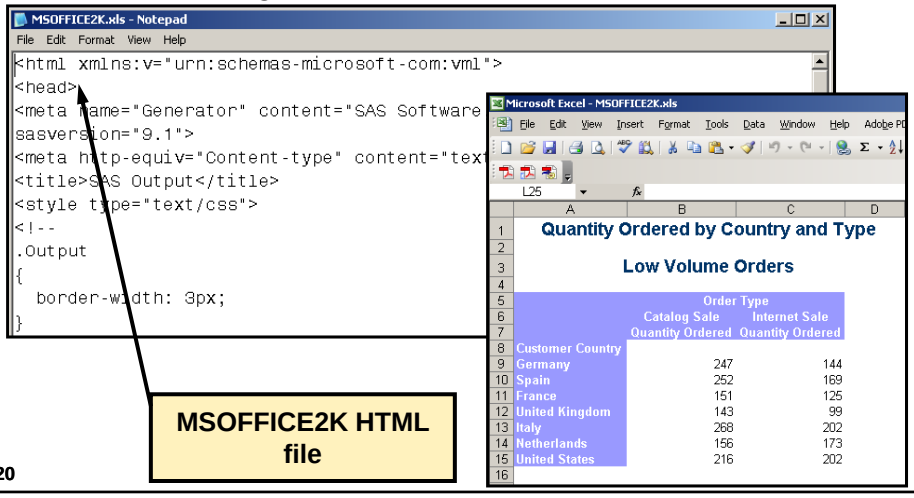
The ODS HTML3 output file is not a Microsoft Excel file. Excel reads HTML 3.2 tags and renders them for viewing.



19

How ODS and Excel Work Together

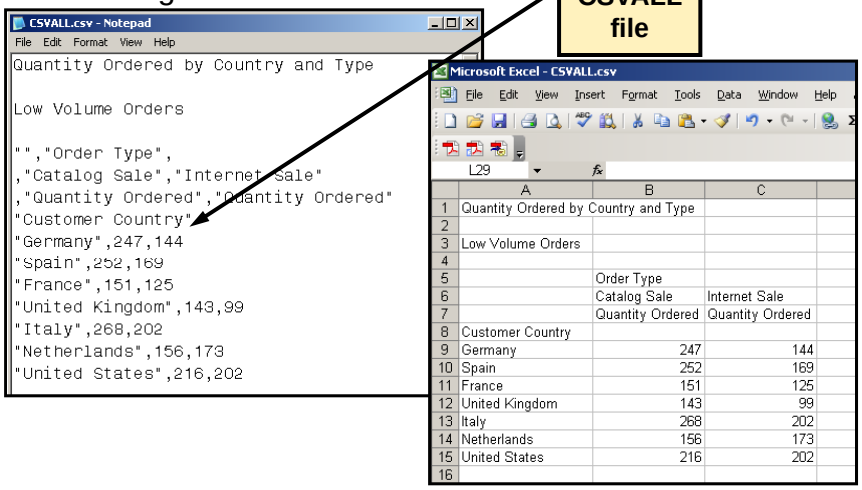
The ODS MSOFFICE2K output file is not a Microsoft Excel file. Excel reads Microsoft HTML tags and renders them for viewing.



20

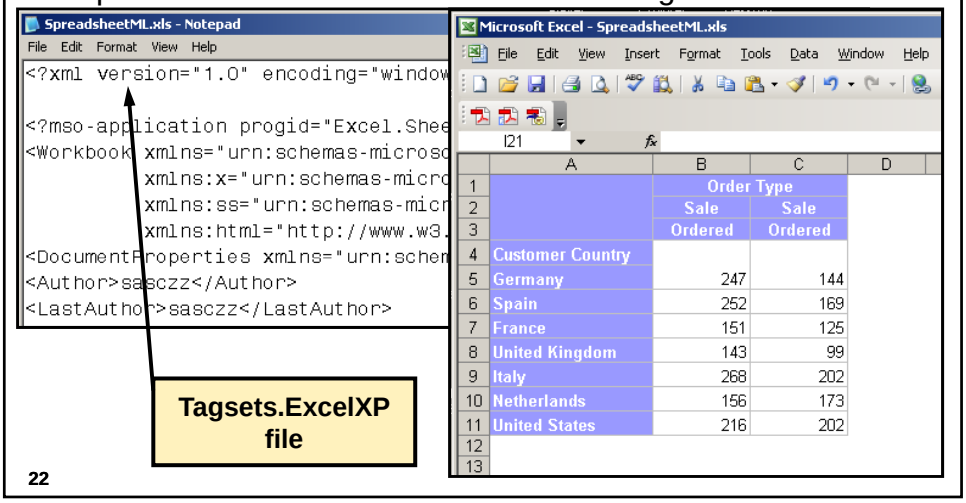
How ODS and Excel Work Together

The ODS CSVALL output file is not a Microsoft Excel file. Excel reads comma-separated values and renders them for viewing.



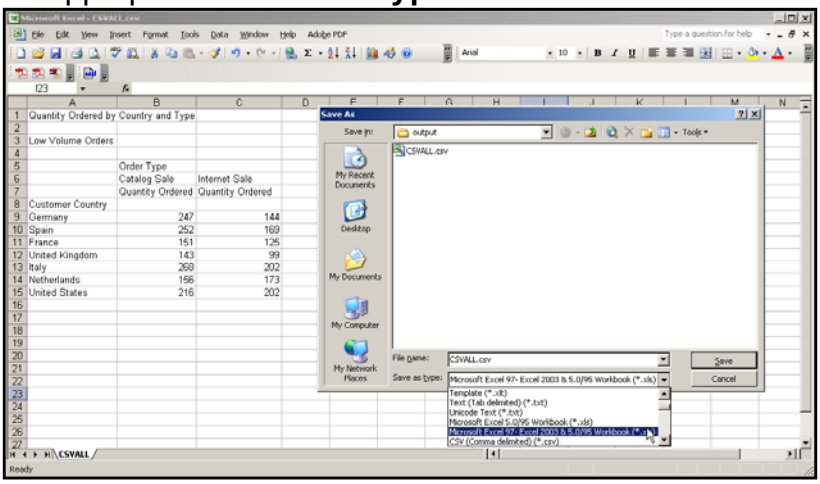
How ODS and Excel Work Together

The ODS TAGSETS.ExcelXP destination creates a Spreadsheet Markup Language XML file. Excel reads Spreadsheet ML and renders it for viewing.



Saving SAS Output as an Excel File

Excel detects whether it opened an HTML, CSV or XML file. To resave the file in Excel proprietary format, select the appropriate **`Save as type`** extension.



23

Specifying a Filename

	HTML	PDF	RTF
BODY= '*filename*' | *fileref*	✓	✓	✓
FILE= '*filename*' | *fileref*	✓	✓	✓

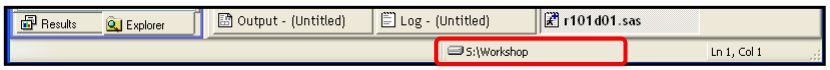
- The filename options specify the file that contains the output.
- '*filename*' is the name of an external file to write to.
- *fileref* is an alias that has been assigned to an external file using the FILENAME statement.
- BODY= and FILE= are interchangeable.
- If you work in a cross-platform environment, then you might also want to use the RS= option to control the carriage control/line feed character that is used for output creation.

24

Using the BODY= Option

```
ods html body='tabrpt.html';
proc tabulate data=orders;
 . . .additional SAS statements. . .
run;
ods html close;
```

```
filename test 'tabrpt.html';
ods html body=test;
proc tabulate data=orders;
 . . .additional SAS statements. . .
run;
ods html close;
```



Without any path information, files are stored in your working directory. In this example, S:\Workshop is the working directory.

25

Using the FILE= Option

```
ods html file='c:\temp\tabrpt.html';
proc tabulate data=orders;
 . . .additional SAS statements. . .
run;
ods html close;
```

```
filename test 'c:\temp\tabrpt.html';
ods html file=test;
proc tabulate data=orders;
 . . .additional SAS statements. . .
run;
ods html close;
```

When you specify full path information in the BODY= or FILE= option, files are stored in that location instead of your working directory.

26

Filename Examples

```
ods html file='tabrpt.html';
proc tabulate data=orders;
 . . .additional SAS statements. . .
run;
ods html close;
```

```
filename test 'tabrpt.
ods html body=test;
proc tabulate data=ord
 . . .additional SAS s
run;
ods html close;
```

Quantity Ordered by Country
Low Volume Orders

	Order Type	
	Catalog Sale	Internet Sale
Customer Country	Quantity Ordered	Quantity Ordered
Germany	247	144
Spain	252	169
France	151	125
United Kingdom	143	99
	268	202
	156	173
United States	216	202

No matter which filename reference method you use, the output files would be the same.

Creating ODS Files on Other Operating Systems

```
ods html file='/usr/mkt/reports/tabrpt.html';
proc tabulate data=orders;
 . . .additional SAS statements. . .
run;
ods html close;
```

UNIX

```
filename test 'XXXXXX.YYYYYY.TABRPT.HTML';
ods html body=test rs=none;
proc tabulate data=orders;
 . . .additional SAS statements. . .
run;
ods html close;
```

z/OS

On other operating systems, you might need to FTP or transfer the result file to an operating system platform that has the appropriate viewer for the file created by ODS. Note that you do **not** need to FTP output when you use SAS Enterprise Guide to communicate with SAS on a server. Also, with SAS Enterprise Guide, you rarely use a BODY= or FILE= option for your programs.

STYLE= Option

	HTML	PDF	RTF
STYLE= *style template*	✓	✓	✓

The STYLE= option specifies the style template to use. Style templates describe how to display the presentation aspects of the output.

To get a listing of the style templates supplied by SAS, submit the following code:

```
proc template;
   list styles / store=sashelp.tmplmst;
run;
```

29

SAS Predefined Style Templates

Templates in Styles Folder			
Analysis*	Education*	Normal*	Theme
Astronomy*	Electronics*	Printer	Torn*
Banker*	Festival*	Rsvp*	Watercolor*
BarrettsBlue	Gears*	RTF	blockPrint*
Beige	Journal*	Sasweb	fancyPrinter
Brick	Magnify*	Science*	sansPrinter
Brown	Meadow*	Seaside*	sasdocPrinter
Curve*	Minimal	Sketch*	serifPrinter
D3d	Money*	Statdoc	
Default	NoFontDefault	Statistical*	

* SAS®9 only

30

STYLE= Option

```
ods html file='useStyle.html' style=Ocean;
proc tabulate data=Orion.customer_orders;
. . . mor
run;
ods html
```

Quantity Ordered by Country and Type
Low Volume Orders

	Order Type	
	Catalog Sale	Internet Sale
Customer Country	Quantity Ordered	Quantity Ordered
Germany	247	144
Spain	252	169
France	151	125
United Kingdom	143	99
Italy	268	202
Netherlands	156	173
United States	216	202

31

STYLE= Option

```
ods pdf file='M6L1DemoB.pdf' style=Rsvp;
proc print data=ia.employees;
   where JobCode='PILOT1';
   title 'Pilot 1 Report';
run;
ods pdf close;
```

Pilot 1 Report

Obs	EmpID	JobCode	Salary
70	E00082	PILOT1	61,000
113	E00145	PILOT1	61,000
175	E00225	PILOT1	64,000
272	E00366	PILOT1	62,000
281	E00377	PILOT1	65,000
404	E00235	PILOT1	65,000
436	E00407	PILOT1	61,000

32

STYLE= Option

```
ods rtf file='M6L1DemoB.rtf' style=Watercolor;
proc print data=ia.employees;
   where JobCode='PILOT1';
   title 'Pilot 1 Report';
run;
ods rtf close;
```

Pilot 1 Report

Obs	EmpID	JobCode	Salary
70	E00082	PILOT1	61,000
113	E00145	PILOT1	61,000
175	E00225	PILOT1	64,000
272	E00366	PILOT1	62,000
281	E00377	PILOT1	65,000
404	E00235	PILOT1	65,000
436	E00407	PILOT1	61,000

33

STYLE= Option

```
ods rtf file='M6L1DemoB.rtf' style=Journal;
proc print data=ia.employees;
   where JobCode='PILOT1';
   title 'Pilot 1 Report';
run;
ods rtf close;
```

Pilot 1 Report

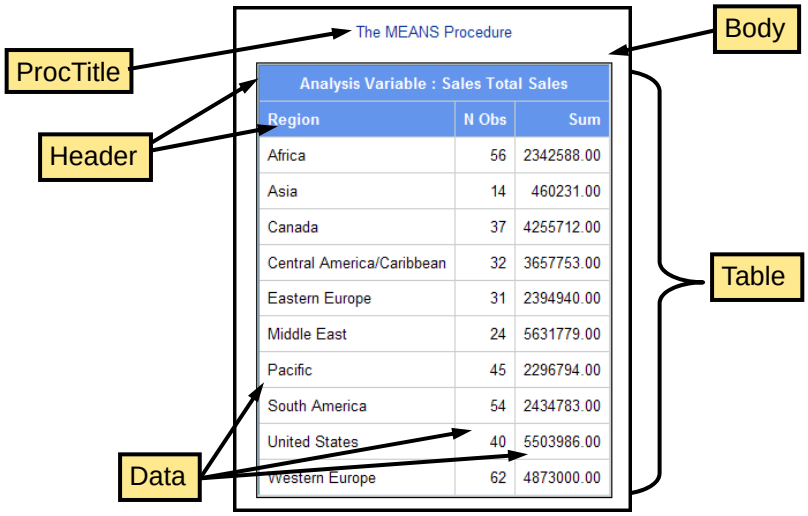
Obs	*EmpID*	*JobCode*	*Salary*
70	E00082	PILOT1	61,000
113	E00145	PILOT1	61,000
175	E00225	PILOT1	64,000
272	E00366	PILOT1	62,000
281	E00377	PILOT1	65,000
404	E00235	PILOT1	65,000
436	E00407	PILOT1	61,000

34

Output viewed in Word Print Preview mode

Style Elements

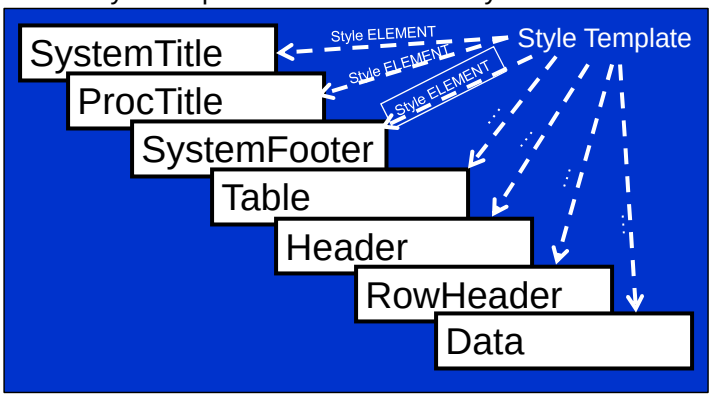
A *style element* is a collection of style attributes that apply to a particular part of the output.



35

Style Template Concepts: Style Elements

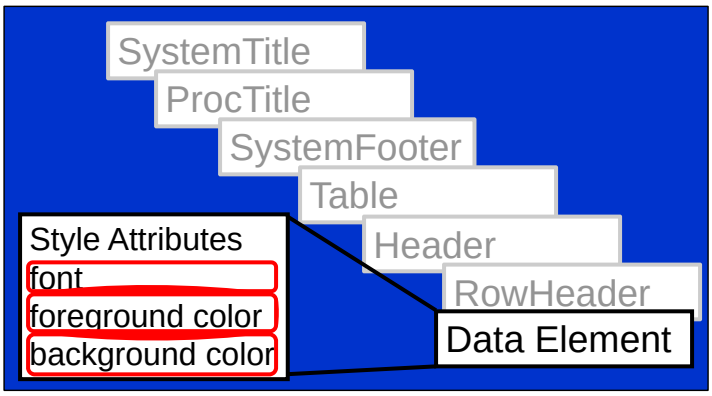
A Style Template is a collection of style ELEMENTS



A STYLE template is a collection of style ELEMENTS. The style elements, conceptually, represent major table and output components, such as SystemTitle or Header.

36

Style Template Concepts: Style Attributes



Each style **ELEMENT** is a collection of style **ATTRIBUTES**. The style attributes represent the individual presentation and output characteristics that can be changed. Style elements their style attributes may work differently in different destinations.

37

Overriding Style Attributes

PROC PRINT, PROC REPORT and PROC TABULATE provide a syntax method for you to override style elements.

38

Employee Report for Level 1 Flight Attendants

EmpID	JobCode	Salary
E00040	FLTAT1	32,000
E00046	FLTAT1	44,000
E00049	FLTAT1	29,000
E00071	FL	
E00084	FL	
E00106	FL	

```
proc print data=ia.employees noobs
  style(header)={font_face='Courier New'
                 font_size=4
                 background=blue
                 foreground=light yellow
                 just=left}
  style(column)={font_face='Courier New'
                 font_size=3
                 background=light yellow
                 foreground=blue}
  style(report)={bordercolor=black
                 borderwidth=2
                 background=red
                 cellspacing=0
                 cellpadding=5
                 rules=rows
                 frame=box};
```

39

```
proc report data=ia.employees nowd
  style(header)={font_face='Courier New' font_size=4
                 background=blue foreground=light yellow
                 just=left}
  style(column)={font_face='Courier New' font_size=3
                 background=light yellow foreground=blue}
  style(summary)={font_face='Courier New' font_size=3
                  font_weight=bold font_style=roman
                  cellheight=50 vjust=B background=blue
                  foreground=light yellow}
  style(report)={bordercolor=black borderwidth=2
                 background=red cellspacing=0 cellpadding=5
                 rules=rows frame=box};
```

Salary Report for
Level 1 Flight Attendant

```
define Salary / 'Annual/Salary'
  style(header)={just=right};
```

CONFIDENTIAL International Airlines

Employee Number
E00040
E00046
E00049
E00071

```
compute before _page_ / Center
  style={preimage='Confidential.jpg'
         font_face="Courier New"
         font_size=5
         font_weight=bold
         background=white
         foreground=blue};
```

40