Getting Started with the SAS® 9.4 Output Delivery System

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2013. *Getting Started with the SAS® 9.4 Output Delivery System*. Cary, NC: SAS Institute Inc.

Getting Started with the SAS® 9.4 Output Delivery System

Copyright © 2013, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2016

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

9.4-P3:odsgs

Contents

About This Book . . . . . *v*
Accessibility . . . . . *xi*

Chapter 1 • Introduction **1**
What Is the Output Delivery System? 1
Benefits of Using ODS 2
How Does ODS Work? 3
Basic Usage 5
ODS-Specific Windows 13
Words to Know 16

Chapter 2 • Learning by Example: Creating Custom Reports with ODS **19**
About the Scenario in This Book 19
Creating the Default Output 19

Chapter 3 • Creating an ODS Document **23**
About the Task That You Will Perform 23
Creating an ODS Document 23
For More Information 26

Chapter 4 • Selecting the Contents of Your Report **27**
About the Tasks That You Will Perform 27
Identify the Output Objects 27
Select the Output Objects 30
For More Information 40

Chapter 5 • Integrating Output with Popular Business Applications and SAS **41**
About the Tasks That You Will Perform 41
Creating RTF Output 42
Creating an E-Book 44
Creating PowerPoint Output 45
Creating PDF Output 47
Creating Enhanced HTML Output 48
Creating Excel Output 50
Combined Program 50
For More Information 54

Chapter 6 • Customizing the Presentation of a Report **55**
About the Tasks That You Will Perform 55
Customized RTF Output 57
Customized E-Book 58
Customized PowerPoint Output 60
Customized PROC TABULATE Output 62
Customized PDF Output 64
Customized HTML Output 66
Customized Excel Output 67
Combined Program 69
For More Information 73

Chapter 7 • Next Steps: A Quick Look at Advanced Features **75**
Working with the TEMPLATE Procedure 75
Working with ODS Graphics 81
Advanced Features of the DOCUMENT Procedure 85
ODS and the DATA Step 89
Working with Cascading Style Sheets (CSS) 91
Working with the ODSTEXT Procedure 93
Working with the ODSLIST Procedure 94
Working with the ODSTABLE Procedure 96
Working with the Report Writing Interface 97
Working with ODS LAYOUT 100
Where to Go from Here 103

Recommended Reading **105**
Index **107**

About This Book

Audience

This book is intended for new or novice users of the SAS Output Delivery System. The documentation assumes familiarity with Base SAS programming and the SAS windowing environment. Although this familiarity is assumed, users who are not familiar will still be able to complete the tasks that are described in this book.

Syntax Conventions for the SAS Language

Overview of Syntax Conventions for the SAS Language

SAS uses standard conventions in the documentation of syntax for SAS language elements. These conventions enable you to easily identify the components of SAS syntax. The conventions can be divided into these parts:

- syntax components
- style conventions
- special characters
- references to SAS libraries and external files

Syntax Components

The components of the syntax for most language elements include a keyword and arguments. For some language elements, only a keyword is necessary. For other language elements, the keyword is followed by an equal sign (=). The syntax for arguments has multiple forms in order to demonstrate the syntax of multiple arguments, with and without punctuation.

keyword
: specifies the name of the SAS language element that you use when you write your program. Keyword is a literal that is usually the first word in the syntax. In a CALL routine, the first two words are keywords.

  In these examples of SAS syntax, the keywords are bold:

  CHAR (*string, position*)

  CALL RANBIN (*seed, n, p, x*);

ALTER (*alter-password*)

BEST *w*.

REMOVE <*data-set-name*>

In this example, the first two words of the CALL routine are the keywords:

CALL RANBIN(*seed*, *n*, *p*, *x*)

The syntax of some SAS statements consists of a single keyword without arguments:

DO;
... SAS code ...
END;

Some system options require that one of two keyword values be specified:

DUPLEX | **NODUPLEX**

Some procedure statements have multiple keywords throughout the statement syntax:

CREATE <UNIQUE> **INDEX** *index-name* **ON** *table-name* (*column-1* <, *column-2*, …>)

argument
specifies a numeric or character constant, variable, or expression. Arguments follow the keyword or an equal sign after the keyword. The arguments are used by SAS to process the language element. Arguments can be required or optional. In the syntax, optional arguments are enclosed in angle brackets (< >).

In this example, *string* and *position* follow the keyword CHAR. These arguments are required arguments for the CHAR function:

CHAR (*string, position*)

Each argument has a value. In this example of SAS code, the argument *string* has a value of 'summer', and the argument *position* has a value of 4:

```
x=char('summer', 4);
```

In this example, *string* and *substring* are required arguments, whereas *modifiers* and *startpos* are optional.

FIND(*string, substring* <, *modifiers*> <, *startpos*>

argument(s)
specifies that one argument is required and that multiple arguments are allowed. Separate arguments with a space. Punctuation, such as a comma (,) is not required between arguments.

The MISSING statement is an example of this form of multiple arguments:

MISSING *character(s)*;

<LITERAL_ARGUMENT> *argument-1* <<LITERAL_ARGUMENT> *argument-2 ...* >
specifies that one argument is required and that a literal argument can be associated with the argument. You can specify multiple literals and argument pairs. No punctuation is required between the literal and argument pairs. The ellipsis (...) indicates that additional literals and arguments are allowed.

The BY statement is an example of this argument:

BY <DESCENDING> *variable-1* <<DESCENDING> *variable-2* …>;

argument-1 <option(s)> <argument-2 <option(s)> ...>
specifies that one argument is required and that one or more options can be associated with the argument. You can specify multiple arguments and associated

options. No punctuation is required between the argument and the option. The ellipsis (...) indicates that additional arguments with an associated option are allowed.

The FORMAT procedure PICTURE statement is an example of this form of multiple arguments:

PICTURE name <(*format-option(s)*)>
<*value-range-set-1* <(*picture-1-option(s)*)>
<*value-range-set-2* <(*picture-2-option(s)*)> …>>;

argument-1=value-1 <argument-2=value-2 ...>
specifies that the argument must be assigned a value and that you can specify multiple arguments. The ellipsis (...) indicates that additional arguments are allowed. No punctuation is required between arguments.

The LABEL statement is an example of this form of multiple arguments:

LABEL *variable-1=label-1 <variable-2=label-2 …>*;

argument-1 <, argument-2, ...>
specifies that one argument is required and that you can specify multiple arguments that are separated by a comma or other punctuation. The ellipsis (...) indicates a continuation of the arguments, separated by a comma. Both forms are used in the SAS documentation.

Here are examples of this form of multiple arguments:

AUTHPROVIDERDOMAIN (*provider-1:domain-1 <, provider-2:domain-2, …>*

INTO :*macro-variable-specification-1 <*, :*macro-variable-specification-2, …>*

Note: In most cases, example code in SAS documentation is written in lowercase with a monospace font. You can use uppercase, lowercase, or mixed case in the code that you write.

Style Conventions

The style conventions that are used in documenting SAS syntax include uppercase bold, uppercase, and italic:

UPPERCASE BOLD
identifies SAS keywords such as the names of functions or statements. In this example, the keyword ERROR is written in uppercase bold:

ERROR <*message*>;

UPPERCASE
identifies arguments that are literals.

In this example of the CMPMODEL= system option, the literals include BOTH, CATALOG, and XML:

CMPMODEL=BOTH | CATALOG | XML |

italic
identifies arguments or values that you supply. Items in italic represent user-supplied values that are either one of the following:

- nonliteral arguments. In this example of the LINK statement, the argument *label* is a user-supplied value and therefore appears in italic:

  LINK *label*;

- nonliteral values that are assigned to an argument.

In this example of the FORMAT statement, the argument DEFAULT is assigned the variable *default-format*:

FORMAT *variable(s)* <*format* > <DEFAULT = *default-format*>;

Special Characters

The syntax of SAS language elements can contain the following special characters:

=

an equal sign identifies a value for a literal in some language elements such as system options.

In this example of the MAPS system option, the equal sign sets the value of MAPS:

MAPS=*location-of-maps*

< >

angle brackets identify optional arguments. A required argument is not enclosed in angle brackets.

In this example of the CAT function, at least one item is required:

CAT (*item-1* <, *item-2*, …>)

|

a vertical bar indicates that you can choose one value from a group of values. Values that are separated by the vertical bar are mutually exclusive.

In this example of the CMPMODEL= system option, you can choose only one of the arguments:

CMPMODEL=BOTH | CATALOG | XML

...

an ellipsis indicates that the argument can be repeated. If an argument and the ellipsis are enclosed in angle brackets, then the argument is optional. The repeated argument must contain punctuation if it appears before or after the argument.

In this example of the CAT function, multiple *item* arguments are allowed, and they must be separated by a comma:

CAT (*item-1* <, *item-2*, …>)

'*value*' or "*value*"

indicates that an argument that is enclosed in single or double quotation marks must have a value that is also enclosed in single or double quotation marks.

In this example of the FOOTNOTE statement, the argument *text* is enclosed in quotation marks:

FOOTNOTE <*n*> <*ods-format-options* '*text*' | "*text*">;

;

a semicolon indicates the end of a statement or CALL routine.

In this example, each statement ends with a semicolon:

```
data namegame;
   length color name $8;
   color = 'black';
   name = 'jack';
   game = trim(color) || name;
run;
```

References to SAS Libraries and External Files

Many SAS statements and other language elements refer to SAS libraries and external files. You can choose whether to make the reference through a logical name (a libref or fileref) or use the physical filename enclosed in quotation marks. If you use a logical name, you typically have a choice of using a SAS statement (LIBNAME or FILENAME) or the operating environment's control language to make the reference. Several methods of referring to SAS libraries and external files are available, and some of these methods depend on your operating environment.

In the examples that use external files, SAS documentation uses the italicized phrase *file-specification*. In the examples that use SAS libraries, SAS documentation uses the italicized phrase *SAS-library* enclosed in quotation marks:

```
infile file-specification obs = 100;
libname libref 'SAS-library';
```

Accessibility

For information about the accessibility of this product, see Accessibility Features of the Windowing Environment for SAS 9.4 at support.sas.com.

Chapter 1

Introduction

What Is the Output Delivery System? 1

Benefits of Using ODS 2

How Does ODS Work? 3
Components of ODS 3
Where Does ODS Put My Output? 4

Basic Usage 5
Overview 5
Base SAS Reporting Procedures 5
Cascading Style Sheets (CSS) 6
The DOCUMENT Procedure 6
ODS Global Statements 7
The Report Writing Interface 11
ODS LAYOUT Statements 11
ODSLIST Procedure 12
ODSTABLE Procedure 12
ODSTEXT Procedure 12
TEMPLATE Procedure 12

ODS-Specific Windows 13
Overview 13
The Documents Window 13
The Templates Window 14
The Template Browser Window 15

Words to Know 16

What Is the Output Delivery System?

The Output Delivery System (ODS) enables you to customize the content of your output, choose how your output is formatted, and customize the appearance of your output. Examples of output formats are EPUB, PDF, RTF, and POWERPOINT.

Important features of ODS include the following:

- ODS destinations produce the following types of output:
  - traditional monospace output
  - an output data set

  - an ODS document that contains a hierarchical file of the output objects
  - a ZIP file
  - output such as PostScript and PDF that is formatted for a high-resolution printer
  - output that is formatted in a markup language such as HTML, HTML 5, and XML
  - RTF output that is formatted for use with Microsoft Word
  - EPUB output that is formatted to be read by a wide variety of e-book readers
  - one or more PowerPoint slides
- ODS provides table templates that define the structure of the output from SAS procedures and from the DATA step. You can customize the output by modifying these table templates or by creating your own.
- ODS provides style templates that you can use to customize features of your output, such as fonts, colors, and line styles.
- ODS enables you to choose an individual output object to be formatted in a different way. For example, PROC UNIVARIATE produces multiple output objects. You can easily create HTML output, one or more output data sets, PDF output, or printer output from one or all of these output objects.
- ODS stores a link to each output object in the Results window of the SAS windowing environment.
- SAS output formatting is centralized in ODS. When ODS destinations are added, they are automatically made available to the DATA step and all procedures that support ODS.
- ODS enables you to produce output for numerous business applications from a single source. This feature saves you time and system resources because you can produce multiple types of output with a single run of your procedure or data query. For example, you can create output formatted for Microsoft PowerPoint, iOS iBooks, Microsoft Excel, Adobe Acrobat, and Microsoft Word.
- ODS enables you to create highly customized output with the Report Writing Interface (RWI) and the DATA step.
- ODS enables you to create text content such as paragraphs and lists.
- ODS enables you to use cascading style sheets (CSS) to customize the appearance of your output.

Benefits of Using ODS

The SAS Output Delivery System (ODS) gives you greater flexibility in generating, storing, and reproducing SAS procedure and DATA step output along with a wide range of formatting options. ODS provides formatting functionality that is not available when using individual procedures or the DATA step without ODS enhancements.

You can use ODS to accomplish the following tasks:

Create reports for popular software applications.
: With ODS, you can use ODS destination statements to create output specifically for software other than SAS and make that output easy to access. For example, you can use the ODS PDF statement to create PDF files for viewing with Adobe Acrobat or

for printing. You can use the ODS EPUB statement to create output for e-book readers. The ODS RTF statement creates output for Microsoft Word. For complete documentation on the ODS destination statements, see "Dictionary of ODS Language Statements" in *SAS Output Delivery System: User's Guide*.

Customize the report contents.
ODS enables you to modify the contents of your output. With ODS, you can embed graphics, select specific cell contents to display, and create embedded links in tables and graphs. You can select specific tables or graphs from procedure output to be printed or you can exclude them. You can create SAS data sets directly from tables or graphics.

Customize the presentation.
ODS enables you to change the appearance of your output. You can change the colors, fonts, and borders of your output. You can customize the layout, format, headers, and style. You can add images and embedded URLs.

Create more accessible SAS output.
The ODS EPUB and ODS EPUB3 destinations are the recommended destinations for creating SAS output that is accessible to the broadest audience. They create e-books that use many of the accessibility features of the EPUB specification. These features allow e-book readers such as iBooks to present e-books so that they adapt to the needs of users with a wide range of abilities. For example, when reading an e-book created by the ODS EPUB and ODS EPUB3 destinations using iBooks on an iPad, users can adjust font size, color schemes, and magnification. They can also access the text using assistive technologies such as the Voiceover screen reader and refreshable braille displays.

How Does ODS Work?

Components of ODS

ODS creates various types of tabular output by combining raw data with one or more table templates to produce one or more output objects. The basic component of ODS functionality is the output object. The PROC or DATA step that you run provides the data component (raw data) and the name of the table template that contains the formatting instructions. The data component and table template together form the output object. There are two types of output objects created by ODS: tabular output objects and graphical output objects. These objects can be sent to any or all ODS destinations, such as PDF, HTML, RTF, or POWERPOINT. By default, in the SAS windowing environment for Windows and UNIX, SAS uses ODS to produce HTML output. By default, in batch mode, SAS produces LISTING output. By specifying an ODS destination, you control the type of output that SAS creates.

You can use ODS to manipulate one or more output objects in many different ways.

- You can use the DOCUMENT procedure to select, rearrange, store, or replay output objects.
- You can use ODS output destinations to create output in many different formats.
- You can use the ODS TRACE statement to determine what output objects are available. Then, you can use the ODS SELECT or ODS EXCLUDE statement to select or exclude the output object from an output destination.

The following figure shows how SAS produces ODS output.

Figure 1.1 *ODS Processing: What Goes in and What Comes Out*

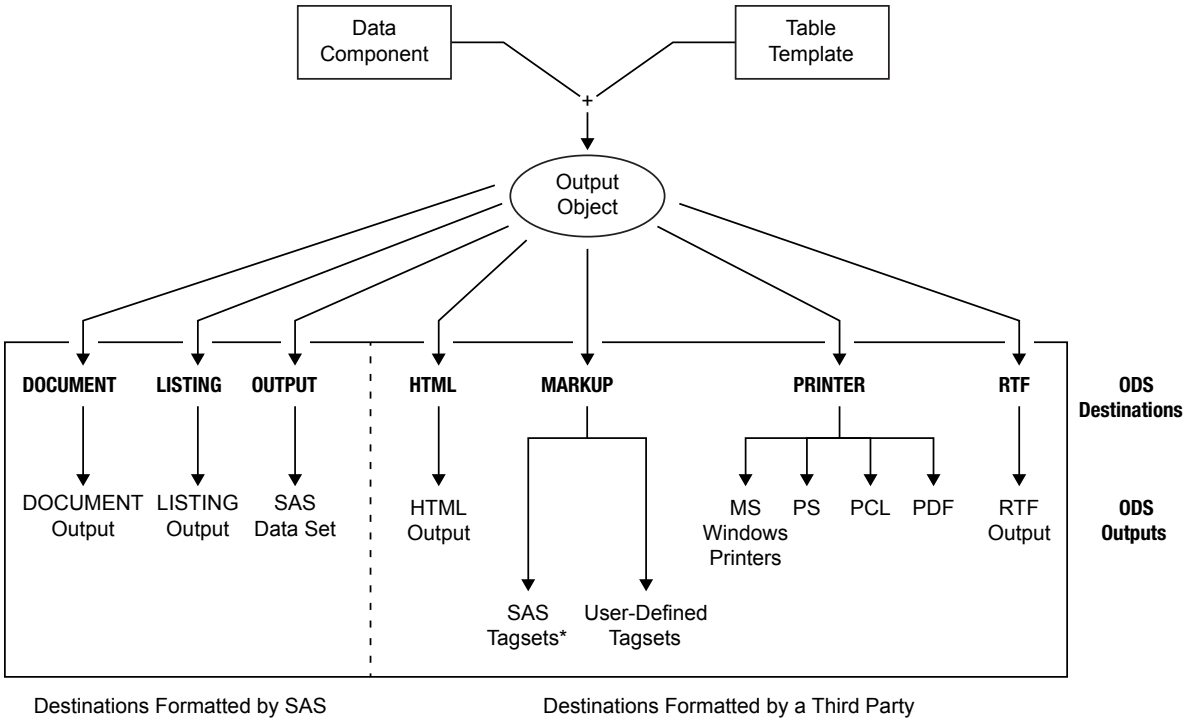


Table 1.1 ** List of Tagsets That SAS Supplies and Supports*

CHTML	CSV	CSVALL	CSVBYLINE
DEFAULT	DOCBOOK	EXCELXP	HTML4
HTMLCSS	HTMLPANEL	IMODE	MSOFFICE2K
PHTML	PYX	TAGSETS.RTF	SASREPORT
WML	WMLOLIST	XHTML	

Table 1.2 ** Additional Diagnostic Tagsets That SAS Supports*

EVENT_MAP	NAMEDHTML	SHORT_MAP	STYLE_DISPLAY
STYLE_POPUP	TEXT_MAP	TPL_STYLE_LIST	TPL_SYLE_MAP

Note: There are also preproduction tagsets. These tagsets can be found at http://support.sas.com. They are not yet supported by SAS.

Where Does ODS Put My Output?

By default, SAS stores output created by ODS in your Work directory. In the SAS windowing environment for Windows and UNIX, after you have opened and closed the HTML destination, your output goes to your current working directory. You can use the

ODS PREFERENCES statement anytime during your SAS session to return to the default behavior. This action is helpful when you are creating multiple graphics and do not want them to accumulate in your current working directory.

Basic Usage

Overview

ODS is used by all SAS software. However, you can explicitly use ODS with the following:

- Base SAS Reporting Procedures on page 5
- Cascading Style Sheets (CSS) on page 6
- DOCUMENT Procedure on page 6
- ODS Global Statements on page 7
- ODSLIST Procedure on page 12
- ODSTABLE Procedure on page 12
- ODSTEXT Procedure on page 12
- TEMPLATE Procedure on page 12

Base SAS Reporting Procedures

The Base SAS reporting procedures, PROC PRINT, PROC REPORT, and PROC TABULATE, enable you to quickly analyze your data and organize it into easy-to-read tables. You can use ODS options with the reporting procedures to give your report another dimension of expression and usability. For example, you can use the STYLE option with a PROC PRINT, PROC REPORT, or PROC TABULATE statement to change the appearance of your report. The following program uses the ODS STYLE option to create the colors in the output below:

```
Title "Height and Weight by Gender and Age";
proc report nowd data=sashelp.class
   style(header)=[background=white];
   col age (('gender' sex),(weight height));
   define age / style(header)=[background=lightgreen];
   define sex / across style(header)=[background=yellow] ' ';
   define weight / style(header)=[background=orange];
   define height / style(header)=[background=tan];
run;
```

Output 1.1 *PROC REPORT Output with Styles Applied*

Height and Weight by Gender and Age

	Gender			
	F		M	
Age	Weight	Height	Weight	Height
253	811	545.3	1089.5	639.1

For complete documentation on the styles and style attributes that you can use with PROC PRINT, PROC TABULATE, and PROC REPORT, see *Base SAS Procedures Guide*.

Cascading Style Sheets (CSS)

Cascading style sheets (CSS) is a style sheet language that you can use with ODS to control the look and formatting of ODS output. A cascading style sheet is an external file that contains label-value pairs that describe the visual aspects of output, such as fonts, colors, borders, and so on. You can then apply the CSS to your ODS output. The CSS language that you can use with ODS is based on the standard CSS syntax found on the Internet at http://www.w3.org/Style/CSS/. However, with ODS, you can apply CSS to many different types of output, such as PDF, RTF, and EXCEL. Using CSS and ODS is an advanced technique. For more information about advanced ODS techniques, see Chapter 7, “Next Steps: A Quick Look at Advanced Features ,” on page 75. For complete documentation about using CSS with ODS, see the chapter about CSS in the *SAS Output Delivery System: Advanced Topics*.

The DOCUMENT Procedure

The combination of the ODS DOCUMENT statement and the DOCUMENT procedure enables you to store a report’s individual components, and then modify and replay the report. The ODS DOCUMENT statement stores the actual ODS objects that are created when you run a report. You can then use PROC DOCUMENT to rearrange, duplicate, or remove output from the results of a procedure or a data query without invoking the procedures from the original report. You can also use PROC DOCUMENT to do the following:

- transform a report without rerunning an analysis or repeating a data query
- modify the structure of output
- display output to any ODS output format without executing the original procedure or DATA step
- navigate the current directory and list entries
- open and list ODS documents
- manage output

- store the ODS output objects in raw form

  Note: The output is kept in the original internal representation as a data component plus a table template.

The DOCUMENT destination has a graphical user interface (GUI), called the Documents window, for performing tasks. However, you can perform the same tasks with batch statement syntax using the DOCUMENT procedure.

For complete documentation about the DOCUMENT procedure, see "The DOCUMENT Procedure" in *SAS Output Delivery System: Procedures Guide*.

ODS Global Statements

Overview

ODS global statements provide greater flexibility to generate, customize, and reproduce SAS procedure and DATA step output. You can use ODS global statements to control different features of ODS. ODS statements can be used anywhere in your SAS program. Some ODS statements remain in effect until you explicitly change them. Other ODS statements are automatically cleared. ODS global statements are organized into two types, output control statements and ODS destination (report) statements. For complete documentation on ODS global statements, see "Dictionary of ODS Language Statements" in *SAS Output Delivery System: User's Guide*..

Output Control Statements

Output Control Statements are statements that provide descriptive information about the specified output objects, and they indicate whether the style template or table template is provided by SAS. The ODS EXCLUDE, ODS SELECT, and ODS TRACE statements are examples of output control statements. Output control statements can do the following:

- select specific output objects for specific destinations
- exclude specific output objects from specific destinations
- specify the location where you want to search for or store style templates or table templates
- verify whether you are using a style template or a table template that is provided by SAS
- provide descriptive information about each specified output object, such as its name, label, template, path, and label path

ODS Destination (Report) Statements

ODS destination (report) statements are statements that enable you to create output that is formatted for third-party software, such as HTML, RTF, and PDF. Or, they enable you to create output that is specific to SAS, such as an ODS document, LISTING output, or a SAS data set. You can use ODS destination statements to generate and modify reports in formats such as HTML, XML, PDF, PostScript, RTF, and Excel. The form for an ODS destination statement is the ODS statement block, which consists of ODS statements that open and close one or more ODS destinations sandwiched around your program. Your results are sent to one or more output destinations.

You can use one or more ODS destination statements, one or more PROC or DATA steps, and an ODS CLOSE statement to form an ODS statement block. An ODS block has the following form:

ODS *output-destination 1 <options(s)>*;

…

ODS *output-destination (n) <options(s)>*

<your SAS program>

ODS *destination close statement 1*;

…

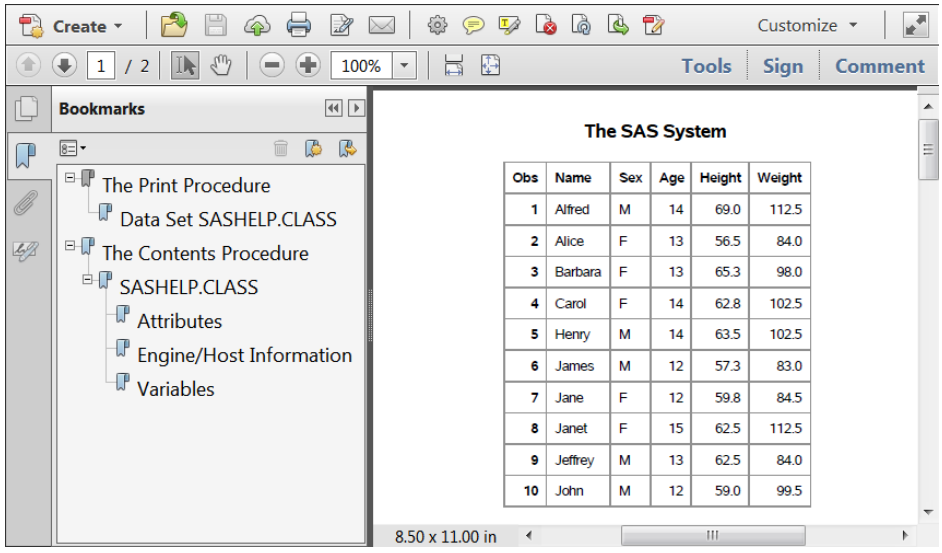
ODS *destination close statement (n)*

In the ODS block, *output-destination* is the name of a valid ODS destination and *option(s)* are options that are valid for that destination. Your SAS program is inserted between the beginning ODS destination statement and the ODS CLOSE statement.

In the following example, the output from PROC PRINT and PROC CONTENTS is sent to the PDF and RTF destinations. The STYLE= option specifies what table template to apply to the output. By default, the PDF opens in Adobe Acrobat and the RTF opens in Microsoft Word.

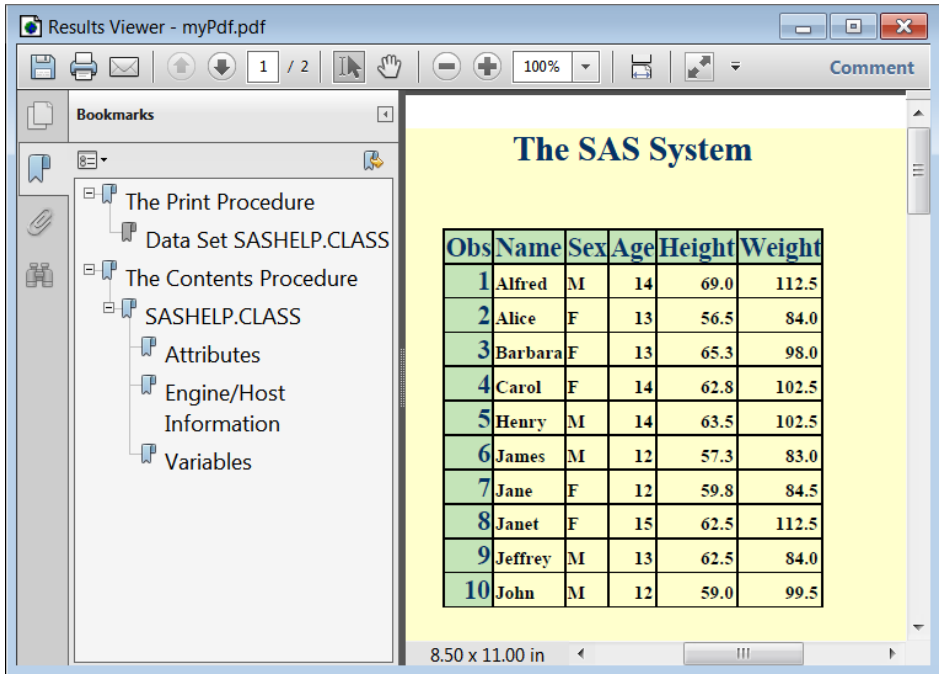
```
options obs=10 nodate;
ods pdf file="myPdf.pdf" style=Banker;
ods rtf file="myRTF.rtf" style=BarrettsBlue text="RTF Output";
proc print data=sashelp.class;
run;

proc contents data=sashelp.class;
run;

ods pdf close;
ods rtf close;
```

Output 1.2 *Default PDF Output*



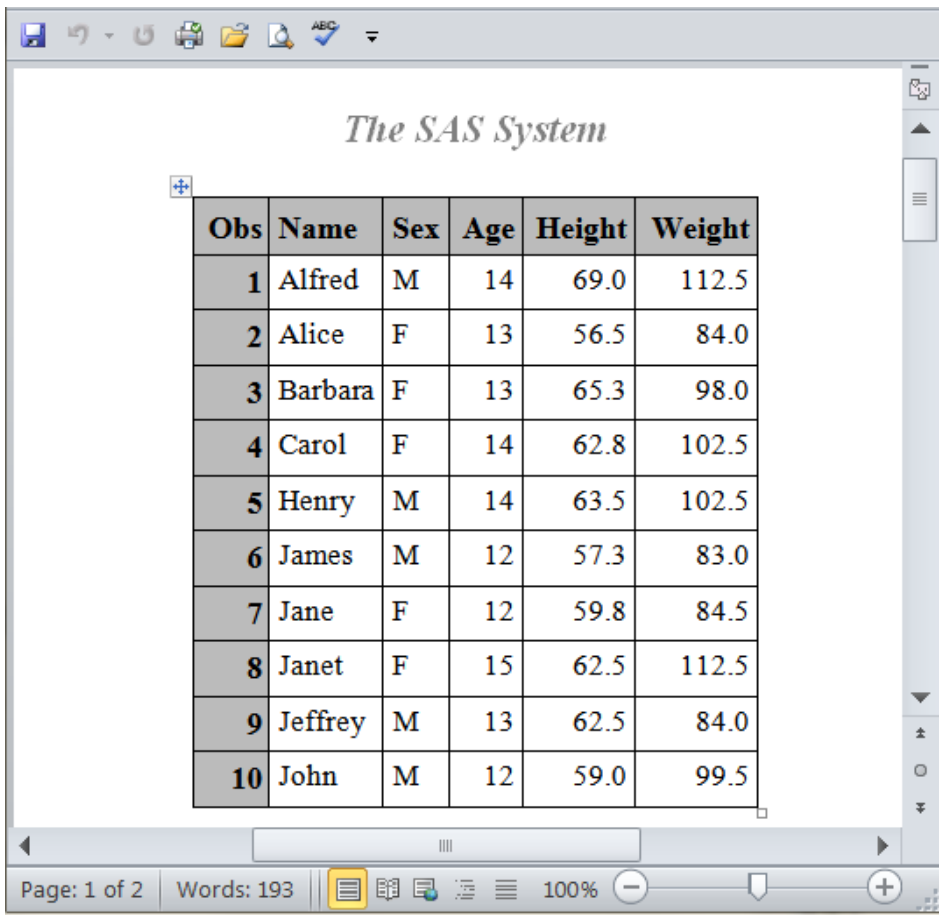
Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84.0
10	John	M	12	59.0	99.5

Output 1.3 *PDF Output with Banker Style Applied*



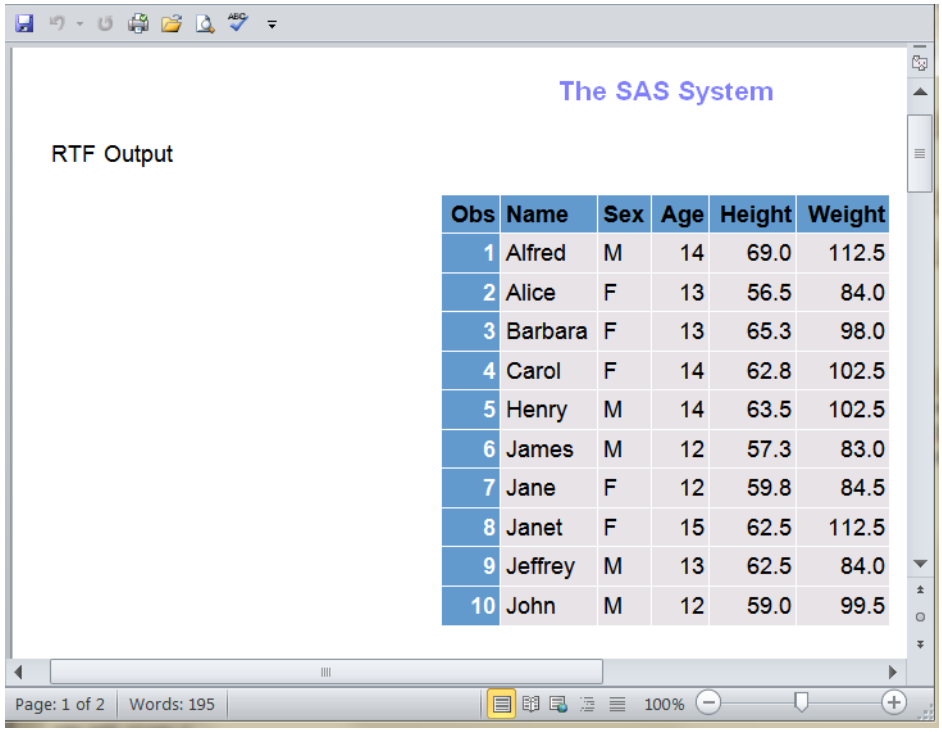
Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84.0
10	John	M	12	59.0	99.5

Output 1.4 *Default RTF Output*



Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84.0
10	John	M	12	59.0	99.5

Output 1.5 *RTF Output with BarrettsBlue Style Applied*



Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84.0
10	John	M	12	59.0	99.5

ODS destinations are organized into two categories.

SAS Formatted Destinations
: These destinations produce output that is controlled and interpreted by SAS, such as a SAS data set, LISTING output, or an ODS document.

Destinations Formatted For a Third Party
: These destinations produce output that is formatted for third-party applications such as e-book readers, PowerPoint, and Microsoft Word. These statements enable you to apply styles or markup languages. You can print to physical printers using page description languages. For example, you can produce output in PostScript, HTML, XML, or in a markup language that you created.

The following table lists the ODS destination categories, the destinations that each category includes, and the formatted output that results from each destination.

Table 1.3 *Destination Category Table*

Category	Destinations	Results
Formatted by SAS	DOCUMENT	ODS document
	LISTING	SAS LISTING output
	OUTPUT	SAS data set
Formatted by a Third Party	EPUB	Output formatted for e-book readers
	EPUB3	Output formatted for e-book readers

Category	Destinations	Results
	HTML	HTML file for online viewing
	MARKUP	Markup language tagsets
	POWERPOINT	PowerPoint slides
	PRINTER	Printable output in one of three different formats: PCL, PDF, or PS (PostScript)
	RTF	Output written in Rich Text Format for use with Microsoft Word 2000

As destinations are added to ODS, they will automatically become available to the DATA step and all procedures that support ODS.

The Report Writing Interface

The ODS Report Writing Interface (RWI) enables you to create highly customized reports from within the DATA step. Because it is fully integrated with ODS, the RWI enables you to combine the powerful programming features from the DATA step with ODS features. This combination gives you flexibility and control over every piece of output in your report. Using the RWI is an advanced technique. For more information about advanced ODS techniques, see Chapter 7, "Next Steps: A Quick Look at Advanced Features ," on page 75. For complete documentation about the RWI, see "Introduction to the Report Writing Interface" in *SAS Output Delivery System: Advanced Topics*.

ODS LAYOUT Statements

ODS LAYOUT statements enable you to create custom reports that easily mix SAS graphics, images, text, and tables, and then arrange them on a page where you like.

- The ODS LAYOUT GRIDDED statement enables you to format in a gridded layout. The ODS LAYOUT GRIDDED statement follows the traditional ODS statement usage in which you wrap (sandwich) your procedure code with a definitive starting and ending location. ODS layout is designed to allow nested layouts (containers) to provide endless customization.
- The ODS LAYOUT ABSOLUTE statement enables you to specify the exact location on the page to place a layout and region container. Each container needs to be explicitly placed to ensure that there is no unintended overlap. The ODS LAYOUT ABSOLUTE statement follows the traditional ODS statement usage in which you wrap (sandwich) your procedure code with a definitive starting and ending location. ODS layout is designed to allow nested layouts (containers) to provide endless customization.

Using the ODS LAYOUT statements is an advanced technique. For more information about advanced ODS techniques, see Chapter 7, "Next Steps: A Quick Look at Advanced Features ," on page 75. For complete documentation about ODS LAYOUT statements, see "Arranging Output with ODS LAYOUT" in *SAS Output Delivery System: User's Guide*.

ODSLIST Procedure

The ODSLIST procedure is used to create bulleted list templates. With PROC ODSLIST, you can do the following:

- create text templates for lists that can be customized and nested an infinite number of times
- use style attributes and formats to customize your content
- use WHERE expressions to specify list item content

With PROC ODSLIST, you can use the DATA= option to bind your data to the template without using a DATA step. PROC ODSLIST can be used with any output destination. However, PROC ODSLIST is essential for creating content for the ODS destination for PowerPoint and e-books. Using the ODSLIST procedure is an advanced technique. For more information about advanced ODS techniques, see Chapter 7, "Next Steps: A Quick Look at Advanced Features ," on page 75. For complete documentation about the ODSLIST procedure, see "The ODSLIST Procedure" in *SAS Output Delivery System: Procedures Guide*.

ODSTABLE Procedure

By default, ODS output is formatted based on the various definitions or templates that the procedure or DATA step specify. However, you can create your own new tabular output templates using the ODSTABLE procedure. ODS uses these templates to produce customized tabular output. With the ODSTABLE procedure, you can create table templates and bind them with the input data set in one statement. You can name your templates and store them in a template store. Using the ODSTABLE procedure is an advanced technique. For more information about advanced ODS techniques, see Chapter 7, "Next Steps: A Quick Look at Advanced Features ," on page 75. For complete documentation about the ODSTABLE procedure, see "The ODSTABLE Procedure" in *SAS Output Delivery System: Procedures Guide*.

ODSTEXT Procedure

The ODSTEXT procedure is used to create text block templates. These text block templates create lists and paragraphs for your output. You can use style attributes and formats to customize your content and WHERE expressions to select your content. With PROC ODSTEXT, you can use the DATA= option to bind your data to the template without using a DATA step. PROC ODSTEXT can be used with any output destination. However, PROC ODSTEXT is essential for creating content for the ODS destination for PowerPoint and e-books. Using the ODSTEXT procedure is an advanced technique. For more information about advanced ODS techniques, see Chapter 7, "Next Steps: A Quick Look at Advanced Features ," on page 75. For complete documentation about the ODSTEXT procedure, see "The ODSTEXT Procedure" in *SAS Output Delivery System: Procedures Guide*.

TEMPLATE Procedure

All SAS procedures produce output objects that ODS delivers to various ODS destinations based on the default specifications for the procedure or based on your own specifications. Output objects are commonly displayed as tables, data sets, or graphs. Each output object has an associated template that is provided by SAS that defines its

presentation format. You can use the TEMPLATE procedure to view or alter a template or to create a new template by changing the headers, formats, column order, and so on.

The TEMPLATE procedure enables you to create or modify a template that you can apply to your output. You can also use the TEMPLATE procedure to navigate and manage the templates stored in template stores. ODS then uses these templates to produce formatted output. Using the TEMPLATE procedure is an advanced technique. For more information about advanced ODS techniques, see Chapter 7, "Next Steps: A Quick Look at Advanced Features ," on page 75.

For complete documentation on the TEMPLATE procedure, see the TEMPLATE procedure in *SAS Output Delivery System: Procedures Guide*.

ODS-Specific Windows

Overview

There are three ODS windows that enable you to manipulate or browse your ODS output and templates.

- Documents Window
- Template Window
- Template Browser Window

The Documents Window

The Documents window displays ODS documents in a hierarchical tree structure.

The Documents window does the following:

- displays all ODS documents, including ODS documents stored in SAS libraries
- organizes, manages, and customizes the layout of the entries contained in ODS documents
- displays the property information of ODS documents
- replays entries
- renames, copies, moves, or deletes ODS documents
- creates shortcuts to ODS documents

To open the Documents window, do one of the following:

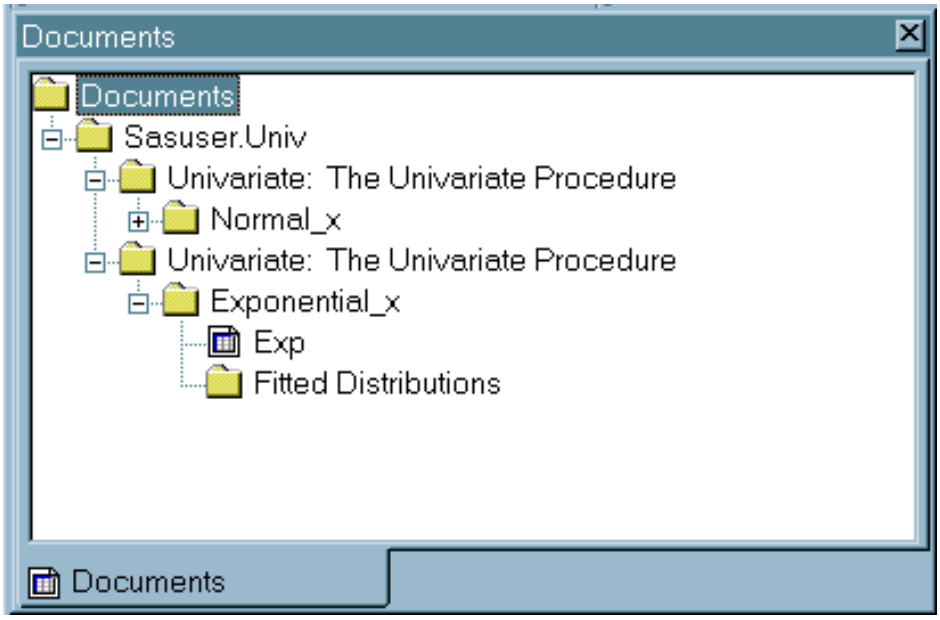
- Select **Results** in the Results window, and then select **View** ⇨ **Documents** from the taskbar.
- Right-click **Results** in the Results window, and then select **Documents**.
- Issue the following command on the command line in the SAS windowing environment:

  ```
  odsdocuments
  ```

This display shows a Documents window that contains an ODS document named Sasuser.Univ. In the display, notice that Sasuser.Univ contains several directory levels. The Exponential_x directory contains the Exp output object. When you double-click on

an output object such as Exp, that output object is replayed in the Results window and sent to all open destinations.

Figure 1.2 *Documents Window*



The Templates Window

Templates contain descriptive information that helps ODS determine the layout of your output. The Templates window enables you to manage all of the templates that are currently available to SAS. Specifically, you can use the Templates window to perform the following tasks:

- Browse ODS styles.
- View template properties.
- Browse template stores and item stores.
- Browse PROC TEMPLATE source code.

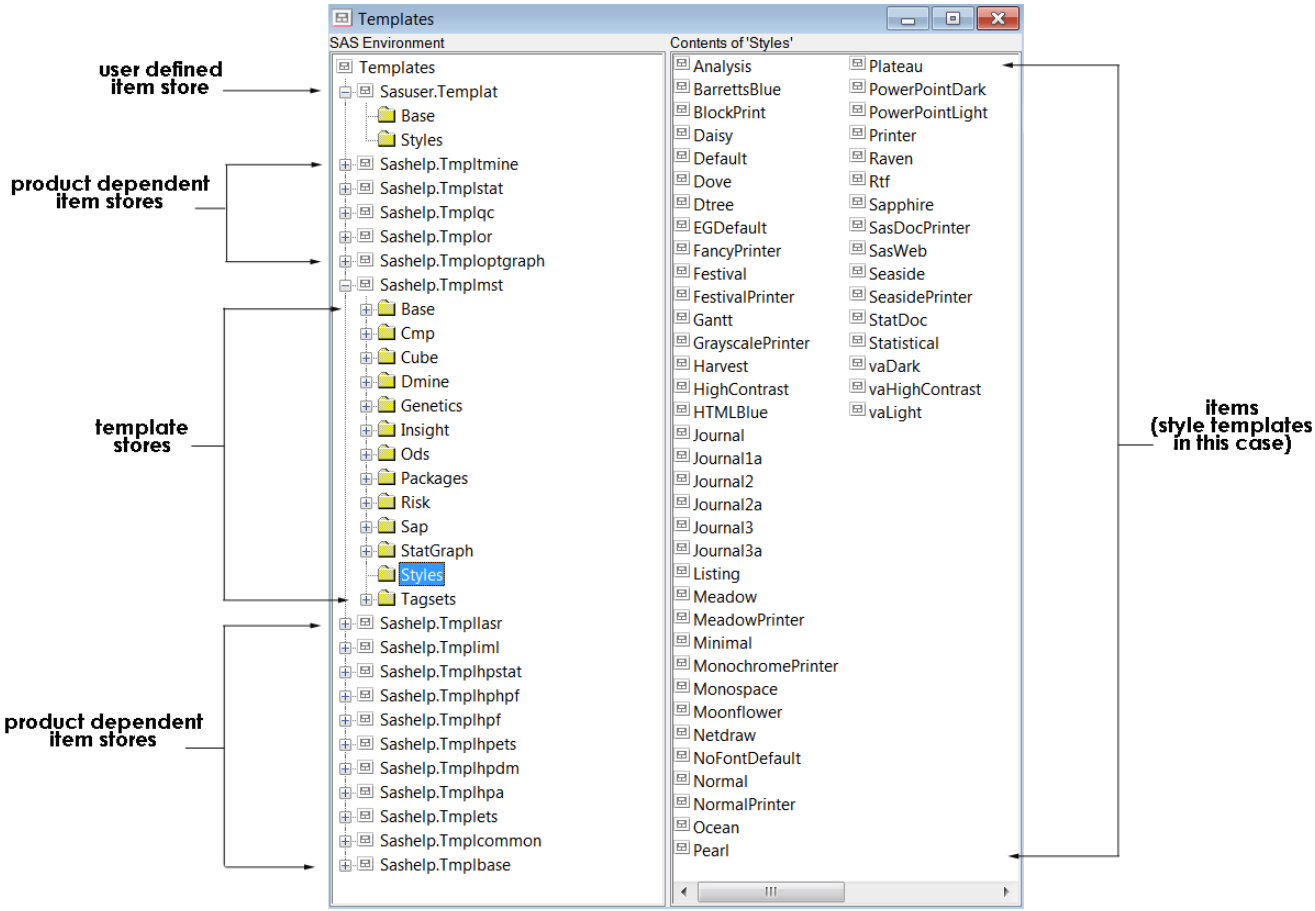
To open the Templates window, do one of the following:

- Select **Results** in the Results window, and then select **View** ⇨ **Templates** from the taskbar.
- Right-click **Results** in the Results window, and then select **Templates**.
- Issue the following command on the command line in the SAS windowing environment:

  ```
  odstemplates
  ```

The hierarchal view on the left side of the Templates window lists the item stores, template stores, and items. The contents of a selected store or directory are displayed on the right side of the window.

Figure 1.3 *Templates Window Showing Item Stores, Template Stores, and Items*



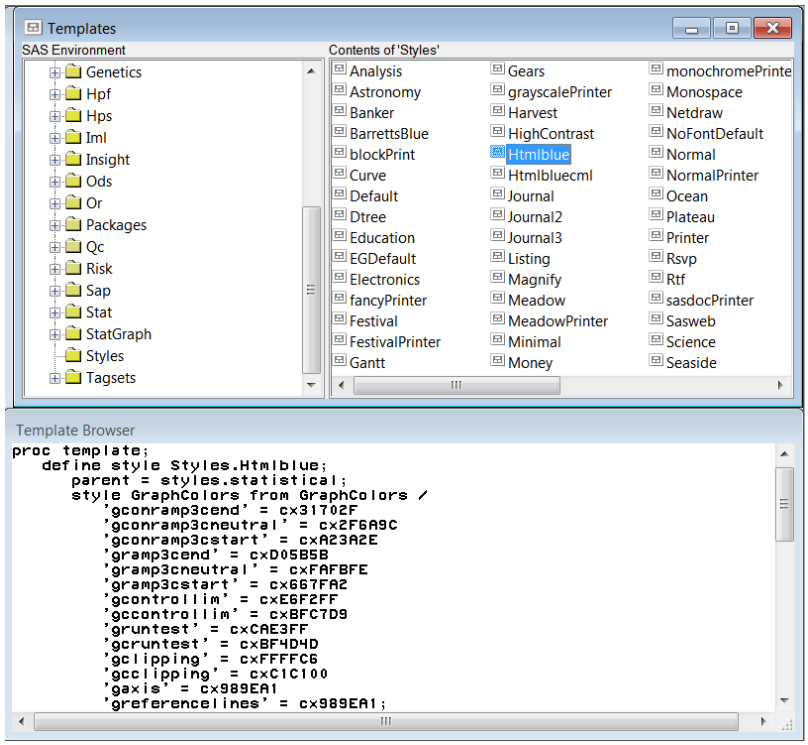
A template store is an item store that contains items that were created by the TEMPLATE procedure. Items that SAS provides are in the Sashelp.Tmplmst item store. By default, compiled templates are stored physically in the Sasuser.Templat item store. You can store items that you create in any template store where you have Write access. A template store can contain multiple levels (directories).

The Template Browser Window

The Template Browser window enables you to view the source code of a template. To open the Template Browser window, double-click on a template or style.

In the following display, the source code for the HTMLBlue style is shown in the Template Browser window.

Figure 1.4 *Templates Window and Template Browser Window*



Words to Know

data components
: a form, similar to a SAS data set, that contains the results (numbers and characters) of a DATA step or PROC step.

item store
: a SAS library member that consists of pieces of information that can be accessed independently. The contents of an item store are organized in a directory tree structure. This structure is similar to the directory structure that is used by UNIX System Services or Windows. For example, a value might be stored in and located by using a directory path (**`root_dir/sub_dir/value`**). The SAS registry, Sasuser.Templat, and Sashelp.Tmplmst are examples of item stores.

template store
: an item store that contains items that were created by the TEMPLATE procedure. Items that SAS provides are in the Sashelp.Tmplmst item store.

ODS destination
: a designation that ODS uses to generate a specific type of output. For example, HTML, XML, LISTING, PostScript, RTF, and SAS data set are ODS destinations.

ODS document
: a hierarchy of output objects created by the DOCUMENT procedure. These output objects are unformatted and placed in a SAS item store.

ODS template
: a definition of how output should appear when it is formatted. An ODS template is stored as a compiled entry in a template store, which is also known as an item store. Common ODS template types include STATGRAPH, STYLE, CROSSTABS, TAGSET, and TABLE.

output object
: data component that is generated by a PROC or DATA step. It can also contain a table template that provides formatting instructions for the data.

table template
: a template that describes how to format the output for a tabular output object. A table template determines the order of table headers and footers, the order of columns, and the overall appearance of the output object. Each table template contains or references table elements.

Chapter 2

Learning by Example: Creating Custom Reports with ODS

About the Scenario in This Book 19
Creating the Default Output 19

About the Scenario in This Book

This book presents an ODS example that is intended to familiarize you with some of the basic features of ODS. From this example, you will learn tasks that will help you enhance and customize SAS output. After this chapter, you do not have to follow the chapters and steps in order.

For the purpose of the scenario in this book, you are a data analyst at a global furniture company. Your manager has asked you for various reports on the data, which you have created. However, you would like to present the information in a more visually pleasing way, and you want to display only the information that is required. This is the first of several presentations of the data, so you would like to save the information in a way that is easy to get to and modify if changes are needed. Your SAS session is being run in the SAS windowing environment in Windows.

Creating the Default Output

The following program creates the original PROC TABULATE, PROC UNIVARIATE, and PROC SGPANEL output. When you run this example program, you are creating ODS output. By default, HTML is created when you run code in the SAS windowing environment for Windows or UNIX. Your output (including your graphics) is sent to your current working directory. This output is viewable in the Results Viewer.

```
options nodate nonumber;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;

proc tabulate data=prdsale;
class region division prodtype;
classlev region division prodtype;
```

```
var actual;
keyword all sum;
keylabel all='Total';
table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];

title 'Actual Product Sales';
title2 '(millions of dollars)';
run;
proc univariate data=prdsale;
by Country;
var actual;
run;
title 'Sales Figures for First Quarter by Product';
proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;
```

When you create this output, you determine that there is more information than you want. Specifically, you need only the Extreme Observations, Quantiles, and Moments tables from the PROC UNIVARIATE output. In addition, you want to make the output

easier to read. The steps in the following chapters show you how to accomplish these tasks.

Output 2.1 *Default Output*

Results Viewer - SAS Output

Actual Product Sales
(millions of dollars)

Region by Division and Type		Product type		Total
		FURNITURE	OFFICE	
		Actual Sales	Actual Sales	Actual Sales
		Sum	Sum	Sum
Region	Division			
EAST	CONSUMER	$72,570	$108,686	$181,256
	EDUCATION	$73,901	$115,104	$189,005
	Total	$146,471	$223,790	$370,261
WEST	Division			
	CONSUMER	$76,209	$105,020	$181,229
	EDUCATION	$67,945	$110,902	$178,847
	Total	$144,154	$215,922	$360,076
Total	Division			
	CONSUMER	$148,779	$213,706	$362,485
	EDUCATION	$141,846	$226,006	$367,852
	Total	$290,625	$439,712	$730,337

Actual Product Sales
(millions of dollars)

The UNIVARIATE Procedure

Chapter 3

Creating an ODS Document

About the Task That You Will Perform 23
Creating an ODS Document 23
For More Information 26

About the Task That You Will Perform

You want to be able to modify your default output later without rerunning your procedures. To accomplish this, you can create an ODS document that contains output from the procedures. The combination of the ODS DOCUMENT statement and the DOCUMENT procedure enables you to store a report's individual components, and then modify and replay the report at a later time. The ODS DOCUMENT statement stores the actual ODS output objects that are created when running a report. You can then use the DOCUMENT procedure to rearrange, duplicate, or remove output from the results without rerunning the procedures from the original report.

Creating an ODS Document

Creating the ODS document with the ODS DOCUMENT statement is the first step toward ODS document functionality. The ODS DOCUMENT statement has the following form:

ODS DOCUMENT *action* | *option*;

In the ODS DOCUMENT statement below, the NAME= option assigns the document the name PrdDocument, and the WRITE option specifies that the document has Write access. Note that using Write access will overwrite existing documents in the library. You must always specify an ODS *destination* CLOSE statement when using ODS global statements. The ODS DOCUMENT CLOSE statement below closes the document so that it can be viewed in the Documents window.

The Documents window displays ODS documents in a hierarchical tree structure. To open the Documents window, issue the following command on the command line:
odsdocuments

```
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
```

```
run;

ods document name=work.prddocument(write);

proc tabulate data=prdsale;
class region division prodtype;
classlev region division prodtype;
var actual;
keyword all sum;
keylabel all='Total';
table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];

title 'Actual Product Sales';
title2 '(millions of dollars)';
run;

proc univariate data=prdsale;
by Country;
var actual;
run;
title 'Sales Figures for First Quarter by Product';
proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;
ods document close;
```

Figure 3.1 *Documents Window*

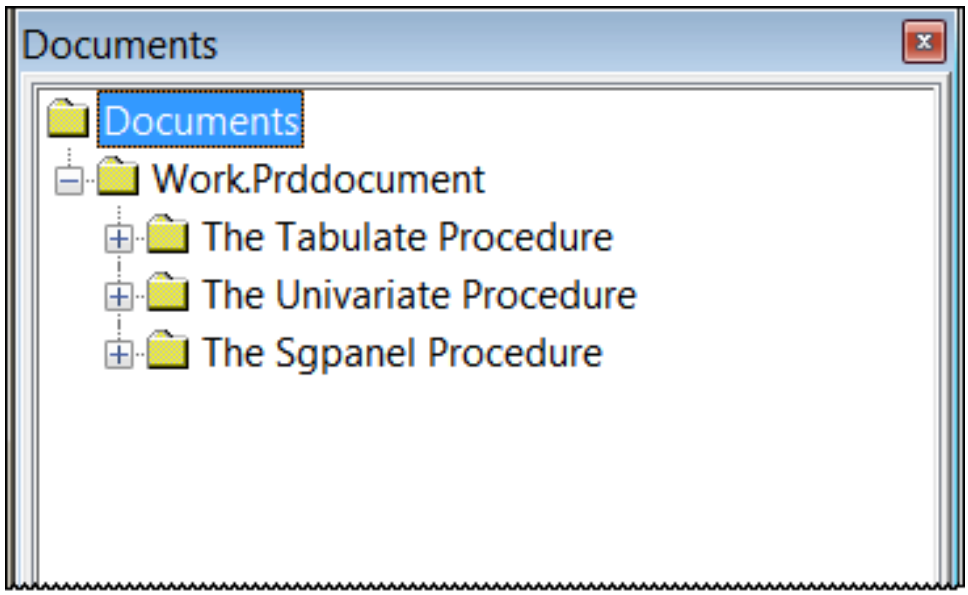
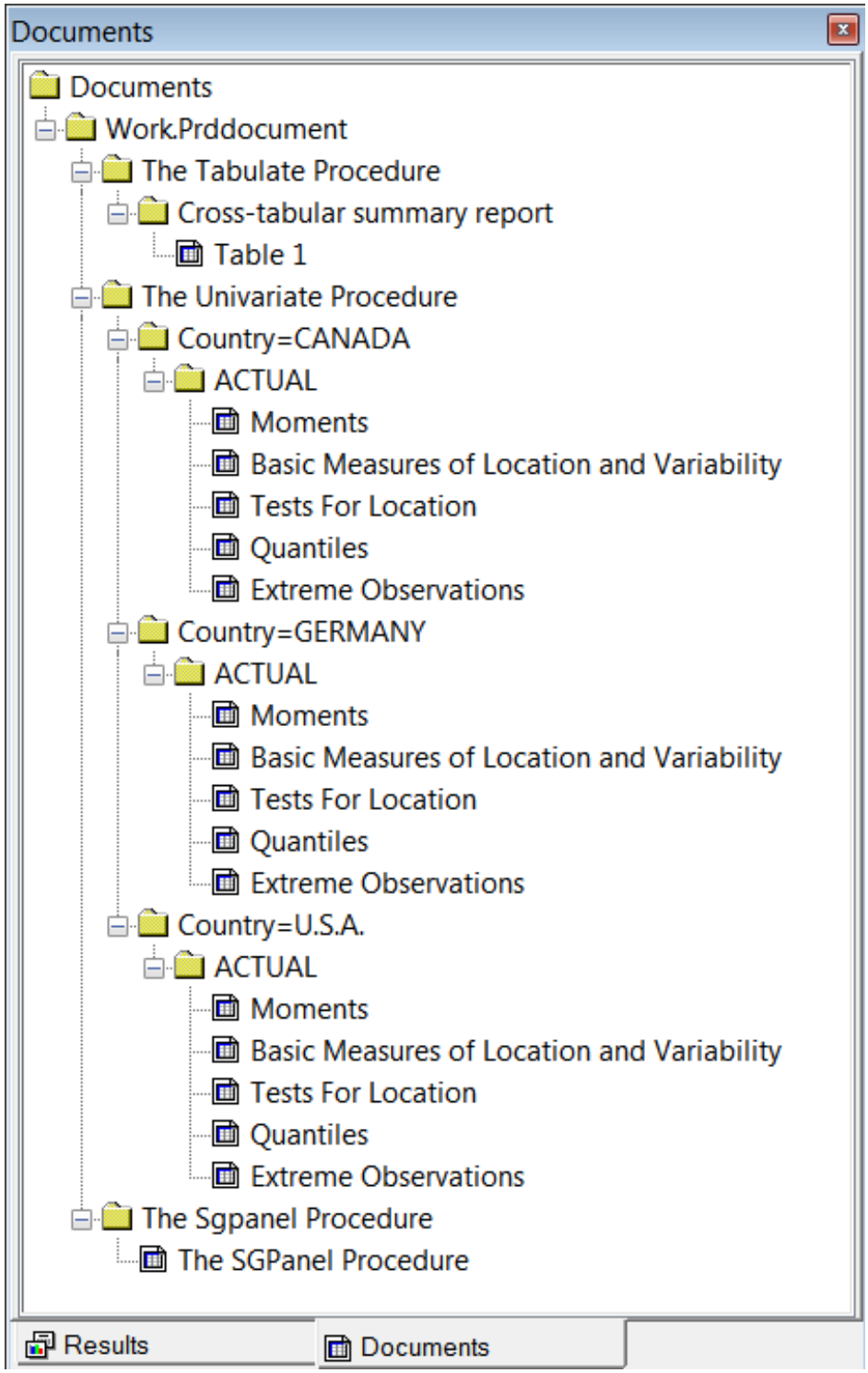


Figure 3.2 *Documents Window Expanded*



For More Information

For more information, see:

- "ODS DOCUMENT Statement" in *SAS Output Delivery System: Procedures Guide*.
- "The DOCUMENT Procedure" in *SAS Output Delivery System: Procedures Guide*.

Chapter 4

Selecting the Contents of Your Report

About the Tasks That You Will Perform 27

Identify the Output Objects 27
Using the ODS TRACE Statement 27
Using the Documents Window 29

Select the Output Objects 30
Using ODS Statements 30
Using the Documents Window 31
Comparing the Two Methods 40

For More Information 40

About the Tasks That You Will Perform

Your content is parceled out into output objects. For your presentation, you do not need all of the output. You want to display the following:

- All of the PROC TABULATE output.
- The Extreme Observations table, Quantiles table, and Moments table for Canada, Germany, and the United States from PROC UNIVARIATE.
- All of the PROC SGPANEL output.

There are two steps to select or exclude an output object. First, you must identify the name, label, or path of the output object. Second, you must use ODS to select or exclude the output object.

Identify the Output Objects

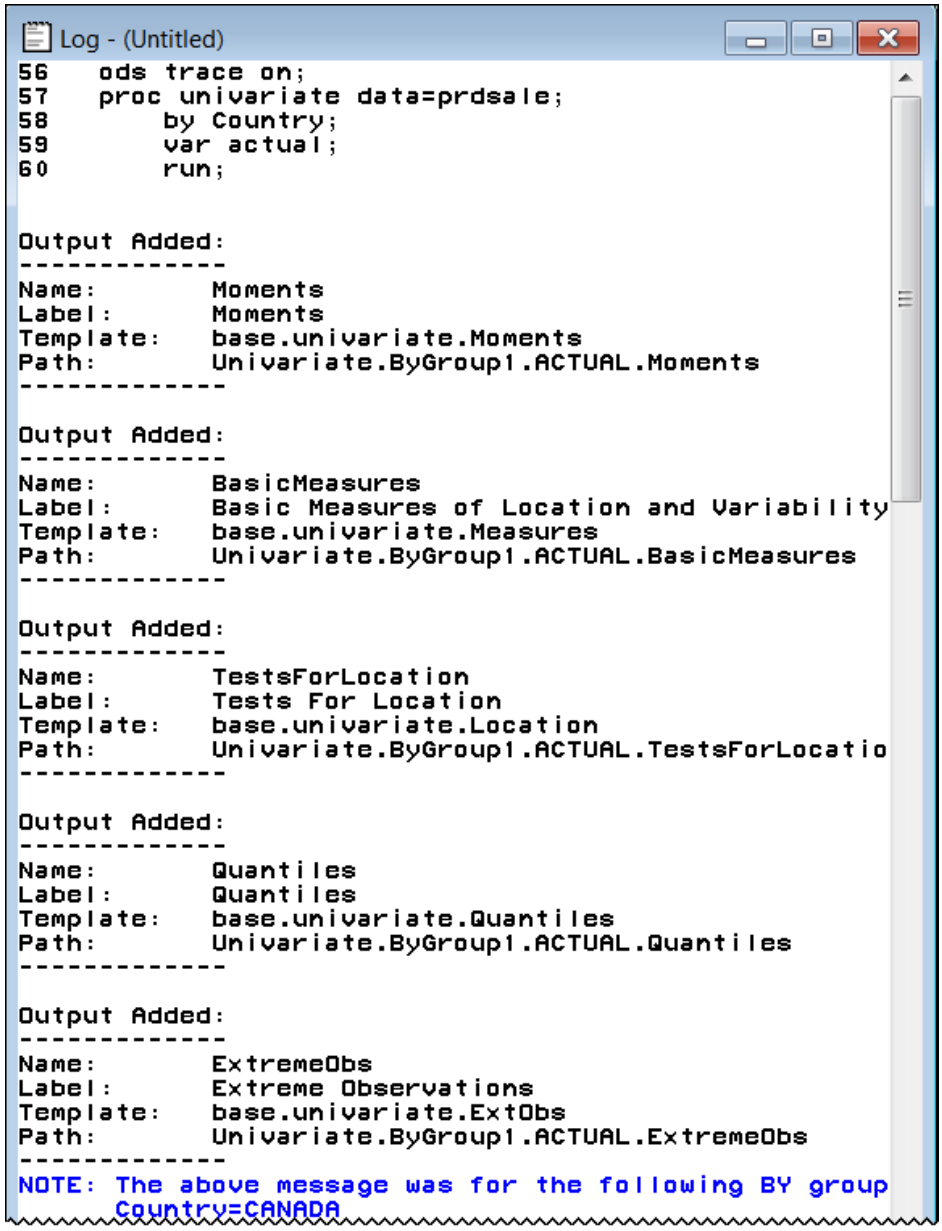
Using the ODS TRACE Statement

The easiest way to identify all of your output objects is with the ODS TRACE statement. Because the ODS TRACE statement is a global statement, you can place it anywhere in your program. By using the ODS TRACE statement, you can see all of the output objects at a glance. Because you want to select or exclude output objects created only by PROC UNIVARIATE, place the ODS TRACE ON statement before the PROC

UNIVARIATE step. Place the ODS TRACE OFF statement after the PROC UNIVARIATE step to stop the generation of trace information.

```
ods trace on;
proc univariate data=prdsale;
    by Country;
    var actual;
    run;
ods trace off;
```

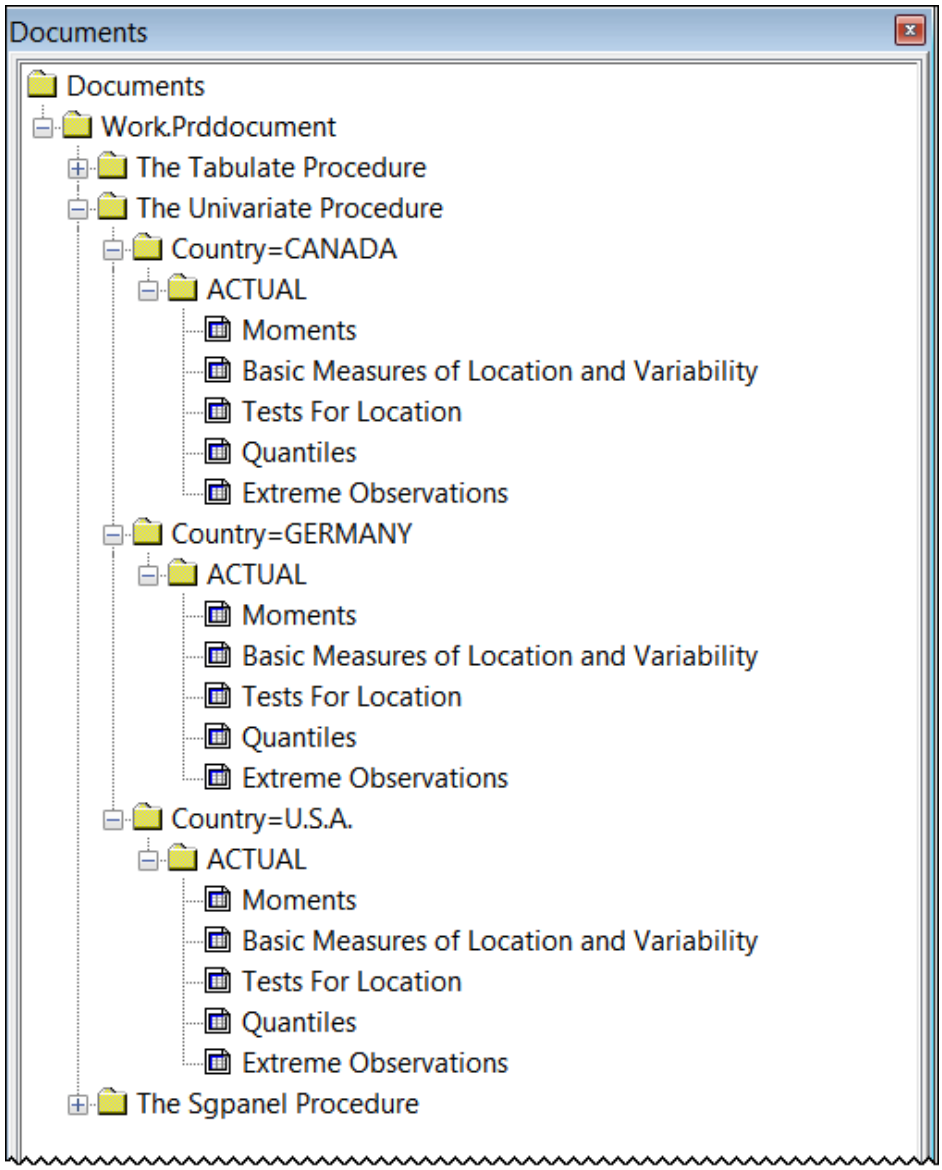
Figure 4.1 *Trace Output Viewed in the SAS Log*



Using the Documents Window

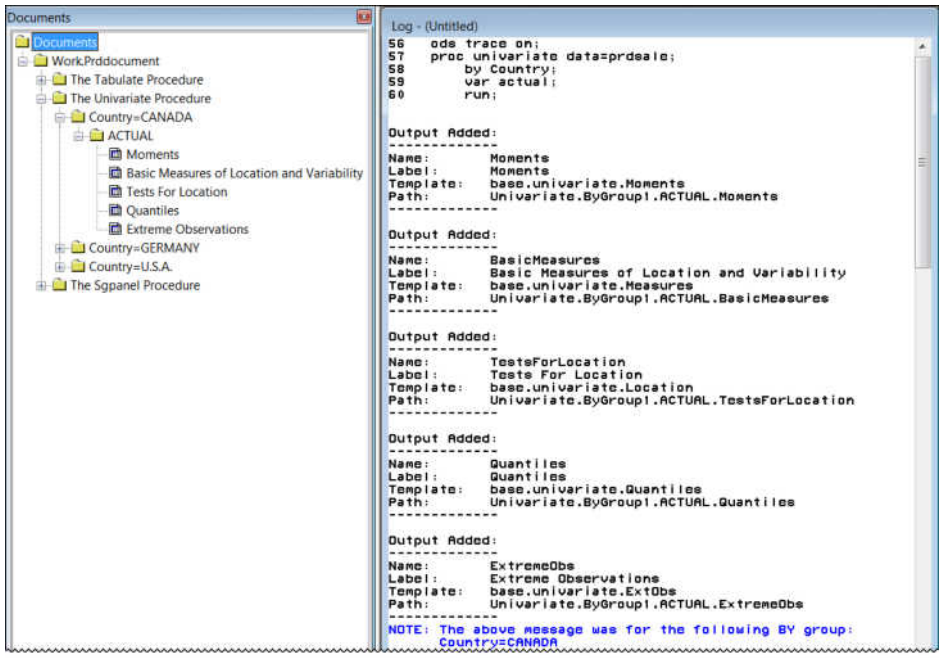
If you have created an ODS document that contains all of the procedure output, you can view the labels of the output objects in the Documents window.

Figure 4.2 *Output Objects Viewed in the Documents Window*



In the figure below, you can see that the labels of the output objects in the trace output correspond to the names given to the output objects in the Documents window.

Figure 4.3 *Comparing the Documents Window and the Trace Output*



Select the Output Objects

Using ODS Statements

Once you have identified all of your output objects, you can use the ODS SELECT statement or the ODS EXCLUDE statement to select or exclude output objects. You can use the name, label, or path to specify the output object in either of these ODS statements.

```
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;
ods html file='your-file-path\HTMLPrdhtml.html';
ods document name=work.prddocument(write);
proc tabulate data=prdsale;
class region division prodtype;
classlev region division prodtype;
var actual;
keyword all sum;
keylabel all='Total';
table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];

title 'Actual Product Sales';
```

```
title2 '(millions of dollars)';
run;
ods select ExtremeObs Quantiles Moments;

proc univariate data=prdsale;
    by Country;
    var actual;
run;
proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;
ods html close;
ods document close;
```

You can use the ODS EXCLUDE statement instead of the ODS SELECT statement. The following ODS EXCLUDE statement gives you the same results:

```
ods exclude BasicMeasures TestsForLocation;
```

Using the Documents Window

If you want to use the Documents window to select your output objects and to store your updated output, you must create a new ODS document.

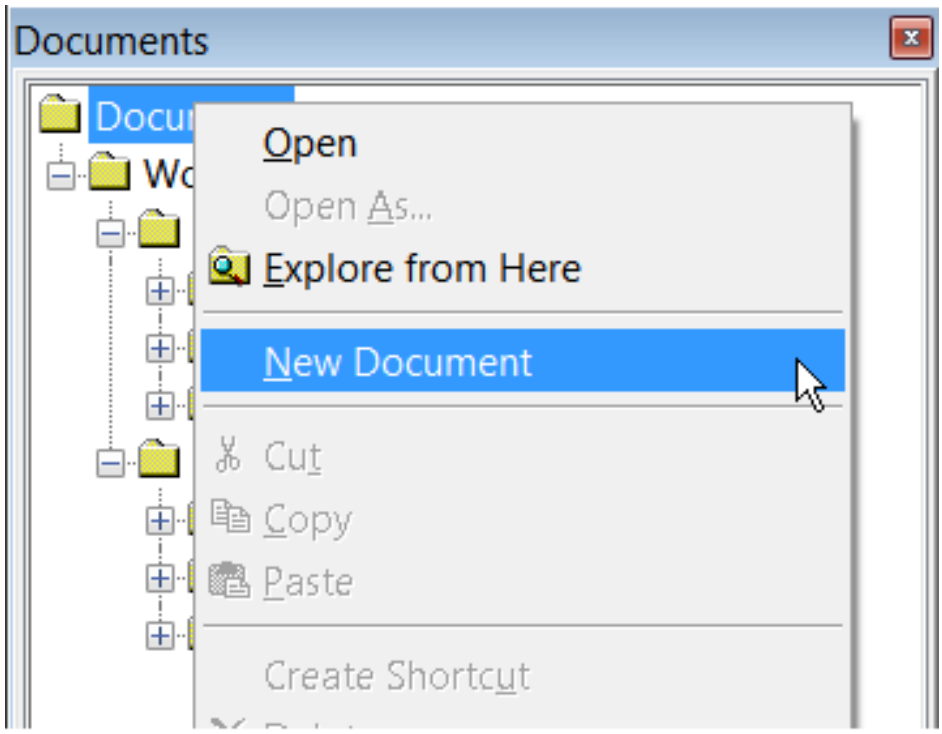
In a previous chapter, you created an original ODS document. You can recall that ODS document.

```
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;

ods document name=work.prddocument(write);

proc tabulate data=prdsale;
class region division prodtype;
classlev region division prodtype;
var actual;
keyword all sum;
keylabel all='Total';
table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];

title 'Actual Product Sales';
title2 '(millions of dollars)';
run;

proc univariate data=prdsale;
by Country;
var actual;
run;
```

```
title 'Sales Figures for First Quarter by Product';
proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;
ods document close;
```

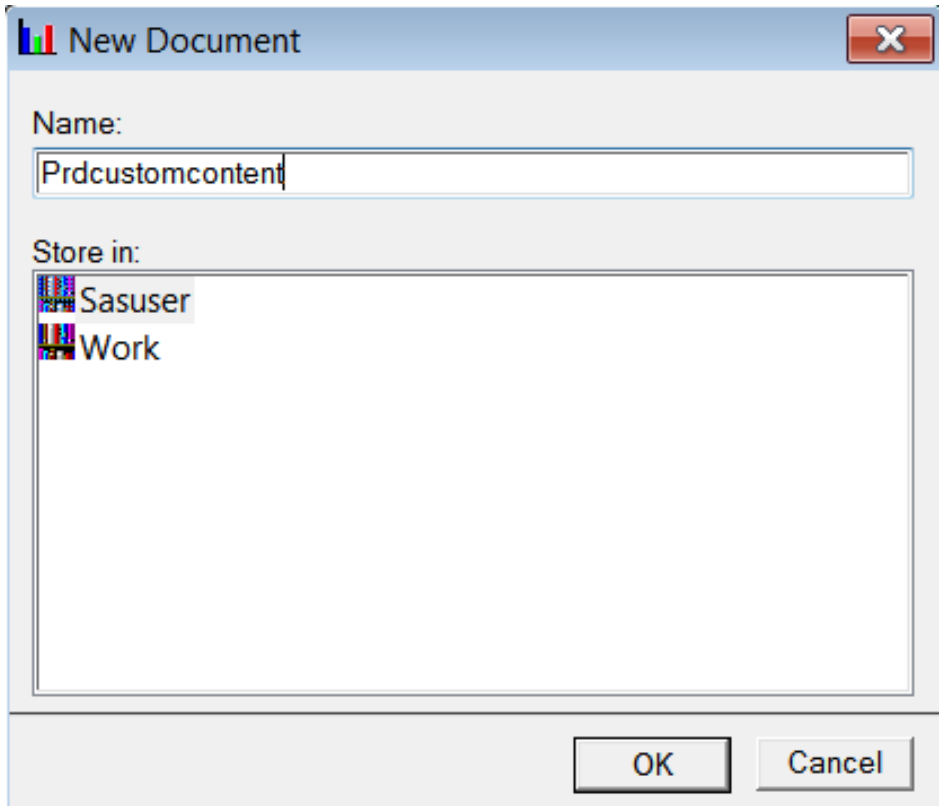
To create a new ODS document, right-click the **Documents** folder at the top of the Documents window, and then select **New Document**.

Figure 4.4 *Creating a New ODS Document*



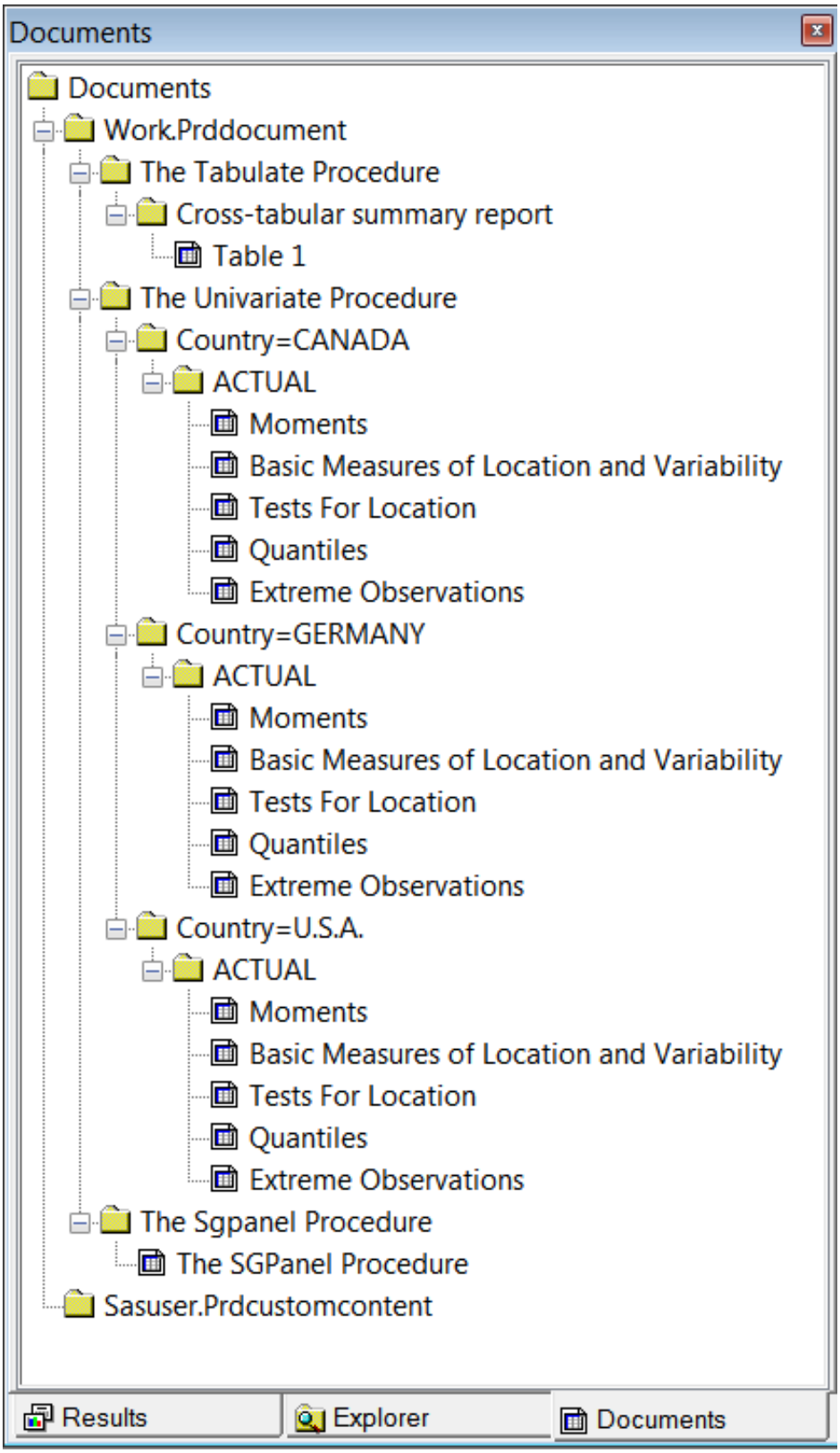
In the New Document window, select a library in which to store the new document, and enter a name for the document. Click **OK**.

Figure 4.5 *Naming a New ODS Document*



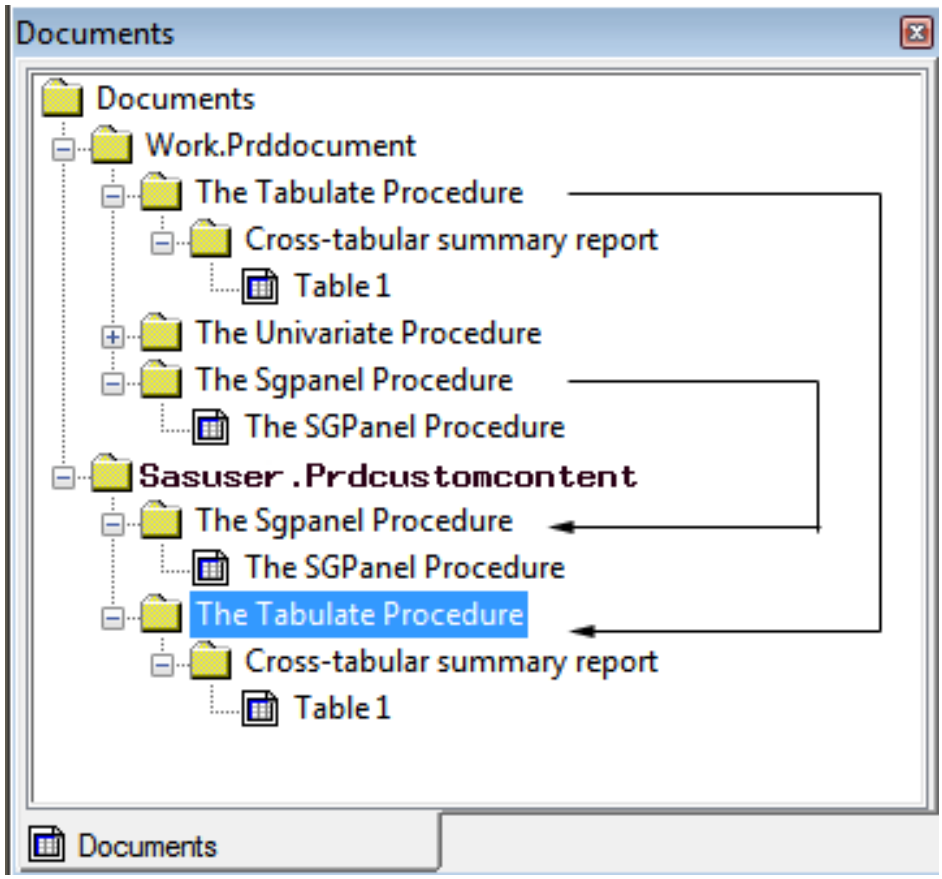
Because you selected the library Sasuser, your document is stored permanently.

Figure 4.6 *New Empty Document PrdCustomContent*



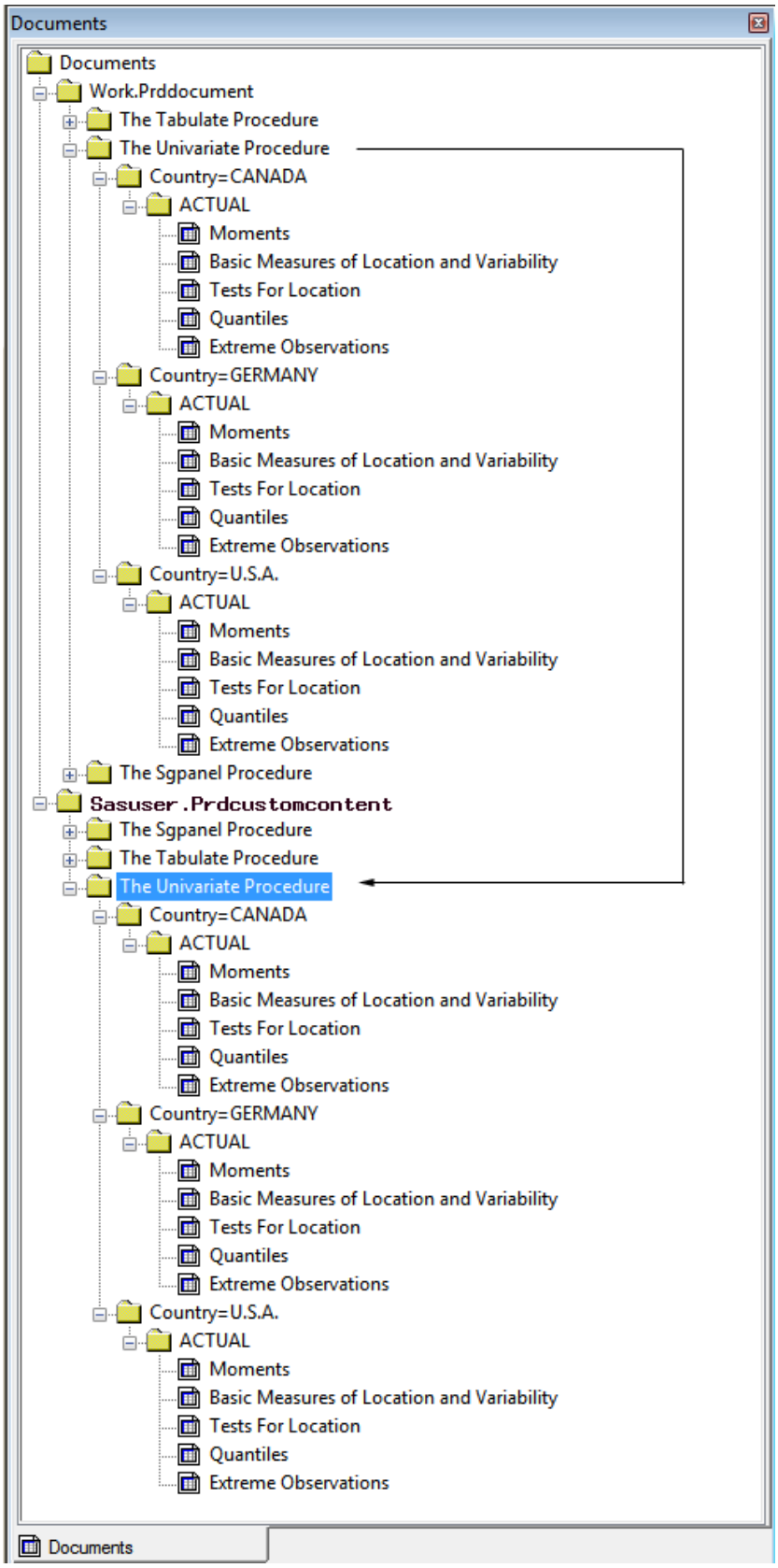
When you create a new document, it is empty. You can load output into the new document by using the Documents window in two ways. You can copy and paste output from the original document (Work.PrdDocument) into the new document (Sasuser.Prdcustomcontent). Or, you can drag and drop output into the new document. You can load individual output objects one by one, as is shown below, or you can add all of the output at once. Then, you can rearrange the output. In the output below under Sasuser.Prdcustomcontent, PROC SGPANEL output is before PROC TABULATE and PROC UNIVARIATE output. Note that you can use PROC DOCUMENT statements to accomplish all these tasks without using the Documents window. For complete documentation on the DOCUMENT procedure, see the "The DOCUMENT Procedure" in *SAS Output Delivery System: Procedures Guide*.

Figure 4.7 *Adding Output to the Sasuser.Prdcustomcontent Document*



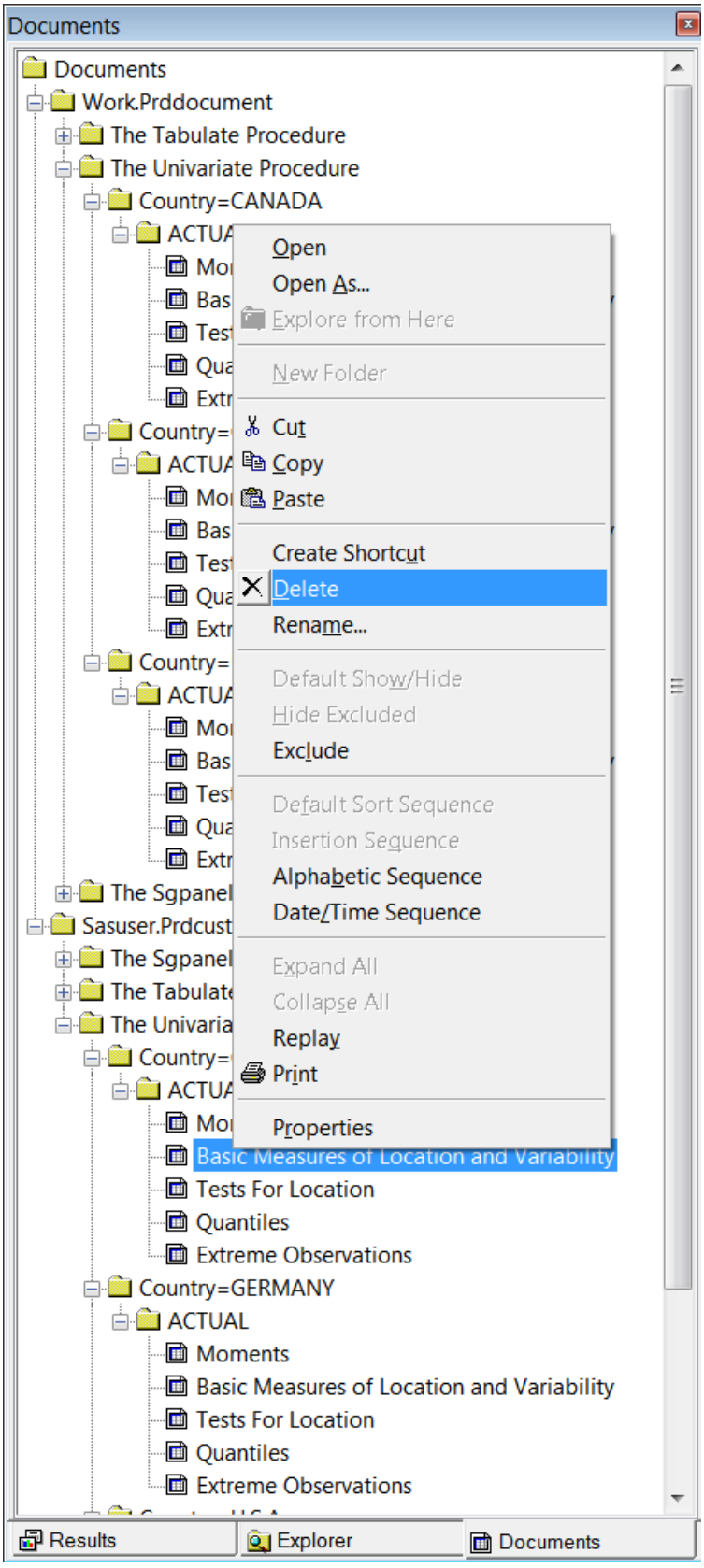
Next, drag, and drop the UNIVARIATE procedure output from Work.Prddocument into Sasuser.Prdcustomcontent. All of the output objects that PROC UNIVARIATE creates are copied into the new document.

Figure 4.8 *Adding the UNIVARIATE Procedure Output*



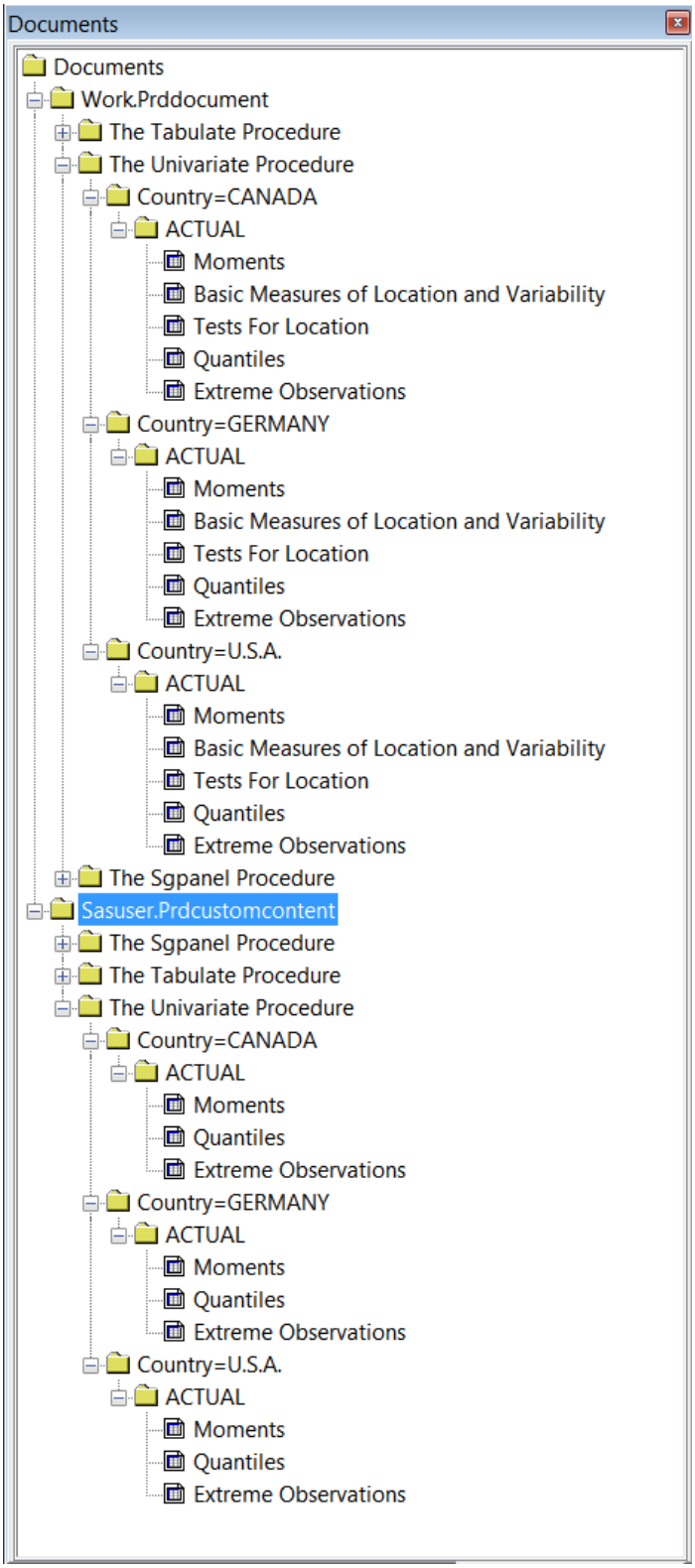
All output objects are in the new document. Delete the PROC UNIVARIATE output objects Basic Measures of Location and Variability and Tests For Location. Right-click on an object, and select **Delete**.

Figure 4.9 *Delete Output Objects*



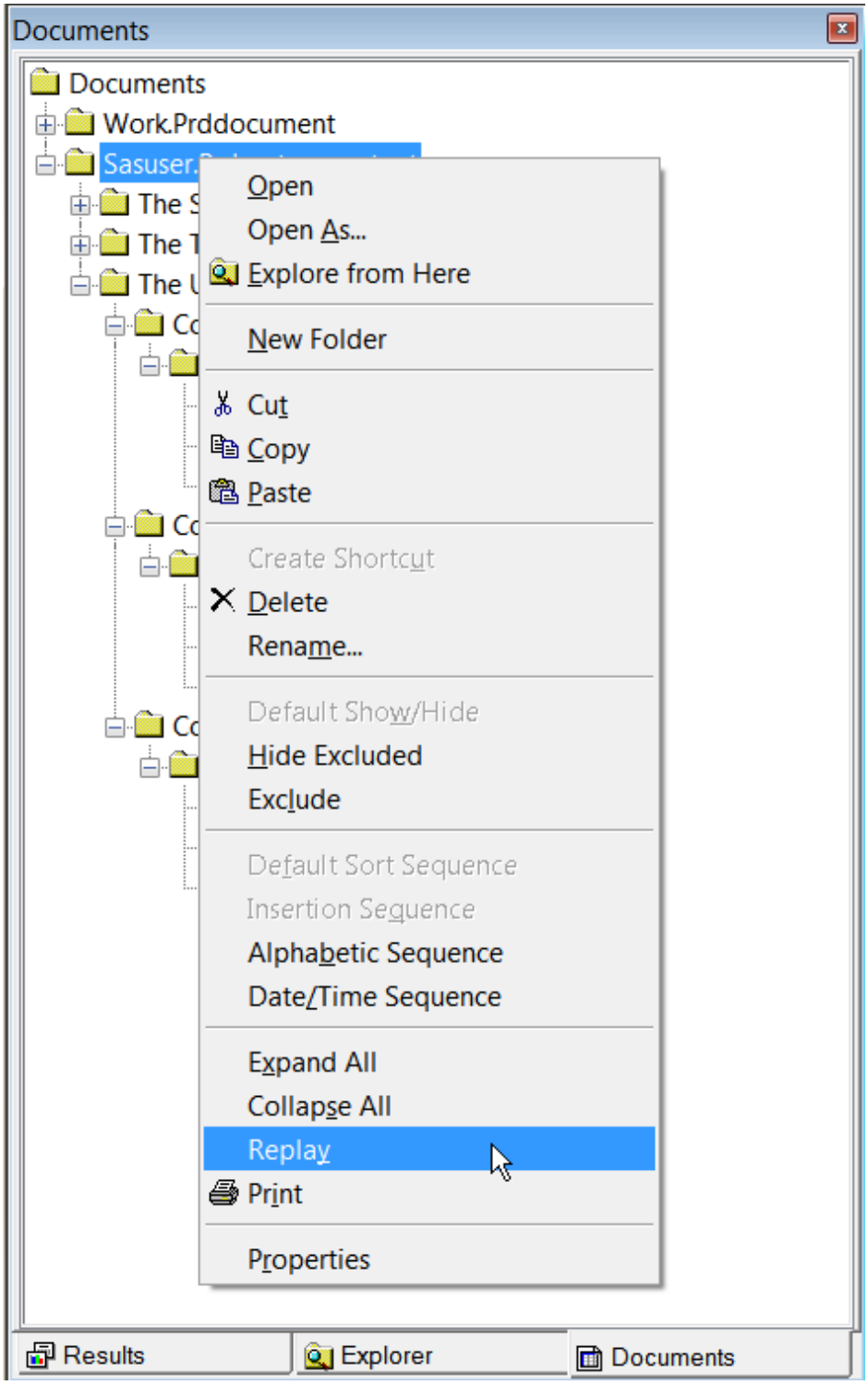
Sasuser.Prdcustomcontent now contains all of the output from PROC SGPANEL and PROC TABULATE. The UNIVARIATE output now consists of only the Moments, Quantiles, and Extreme Observations tables for Canada, Germany, and the United States.

Figure 4.10 *Completed New Document*



Now that you have the output that you want in the new document, you can re-create the output with only the data that you want shown. Right-click **Sasuser.Prdcustomcontent**, and select **Replay**. This displays your output to any open destinations without rerunning the procedures.

Figure 4.11 *Replay the New Document*



Comparing the Two Methods

If you have your output in an ODS document, you might find it easier to select and exclude output using the Documents window or PROC DOCUMENT. By using PROC DOCUMENT, you do not need to access the data or run the program again. It is easy to rearrange and rename your output objects in the Documents window.

If you do not have your output in an ODS document and you have many output objects to select or exclude, you might want to use the ODS SELECT or ODS EXCLUDE statement. In both of these ODS statements, you can create one line of code that selects or excludes all output objects that you want it to. You can place these ODS statements anywhere in the program.

For More Information

- For complete documentation on ODS statements, see "Dictionary of ODS Language Statements" in *SAS Output Delivery System: User's Guide*.
- For complete documentation on the DOCUMENT procedure, see "The DOCUMENT Procedure" in *SAS Output Delivery System: Procedures Guide*.

Chapter 5

Integrating Output with Popular Business Applications and SAS

About the Tasks That You Will Perform 41
Creating RTF Output 42
Creating an E-Book 44
Creating PowerPoint Output 45
Creating PDF Output 47
Creating Enhanced HTML Output 48
Creating Excel Output 50
Combined Program 50
For More Information 54

About the Tasks That You Will Perform

After you select the contents of your report, you can create output for many different business applications. For your report, you are going to create EPUB, PowerPoint, RTF, PDF, HTML, and Excel output. With ODS, it is easy to create output that is formatted for different business applications using ODS destination statements. The ODS statement and the SAS program that it contains form the ODS block.

An ODS block has the following form:

ODS *output-destination 1 <options(s)>*;

…

ODS *output-destination (n) <options(s)>*

<your SAS program>

ODS *destination close statement 1*;

…

ODS *destination close statement (n)*

In the ODS block, *output-destination* is the name of a valid ODS destination and *option(s)* are options that are valid for that destination. Your SAS program is inserted between the beginning ODS destination statement and the ODS CLOSE statement.

Most ODS destination statements require the FILE= or BODY= option, in which the name and path of the file that you are generating is specified. It is a good practice to

specify one of these options, but it is not always required. By default, if you have not closed and reopened the ODS HTML destination, your output is stored in your temporary directory, unless you specify a different directory with the PATH= option in the ODS HTML statement. After you have opened and closed the ODS HTML destination, your output is stored in your local directory, unless you specify a different directory with the PATH= option.

After you run your program, your HTML, PDF, and Excel output opens in the Results Viewer. RTF output opens in Microsoft Word. PowerPoint output opens in PowerPoint. The EPUB output must be opened with an e-book reader.

Creating RTF Output

The ODS RTF statement creates RTF output. Suppose you decide that you want only PROC TABULATE output and PROC SGPANEL output in RTF format. To create this output, wrap (sandwich) the ODS RTF statement and ODS RTF CLOSE statement around your program. Use the FILE= option in the ODS RTF statement to specify the name and path for your file. The RTF output opens in Microsoft Word. Because the HTML destination is open by default, it is good practice to close the HTML destination if you do not want HTML output. This saves system resources.

```
ods html close;

options nodate nonumber;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;

ods rtf file='your-file-path/RTFPrdsale.rtf';
title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale;
    class region division prodtype;
    classlev region division prodtype;
    var actual;
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];
 run;

title 'Sales Figures for First Quarter by Product';
title2;
proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;
```

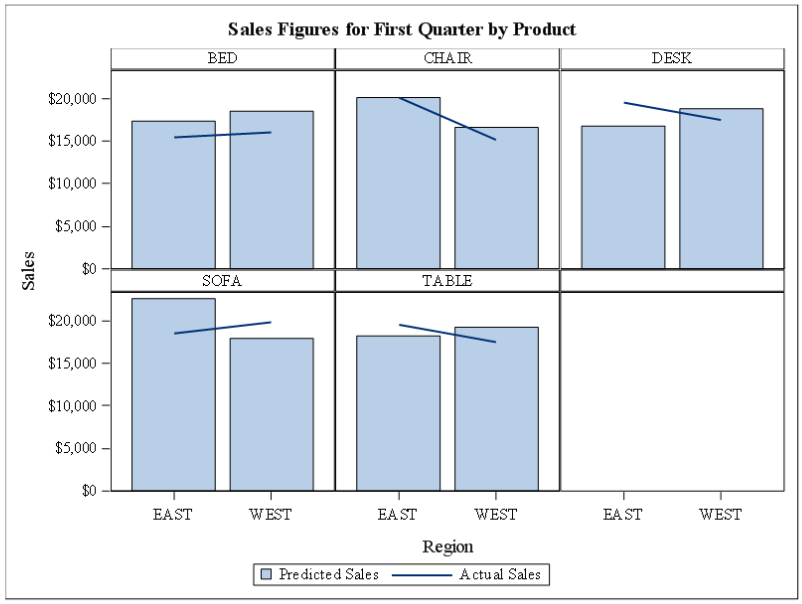
```
ods rtf close;
```

Output 5.1 *PROC TABULATE Output Viewed in Microsoft Word*

Actual Product Sales
(millions of dollars)

Region by Division and Type		Product type		Total
		FURNITURE	OFFICE	
		Actual Sales	Actual Sales	Actual Sales
		Sum	Sum	Sum
Region	Division			
EAST	CONSUMER	$72,570	$108,686	$181,256
	EDUCATION	$73,901	$115,104	$189,005
	Total	$146,471	$223,790	$370,261
WEST	Division			
	CONSUMER	$76,209	$105,020	$181,229
	EDUCATION	$67,945	$110,902	$178,847
	Total	$144,154	$215,922	$360,076
Total	Division			
	CONSUMER	$148,779	$213,706	$362,485
	EDUCATION	$141,846	$226,006	$367,852
	Total	$290,625	$439,712	$730,337

Output 5.2 *PROC SGPANEL Output Viewed in Microsoft Word*



Creating an E-Book

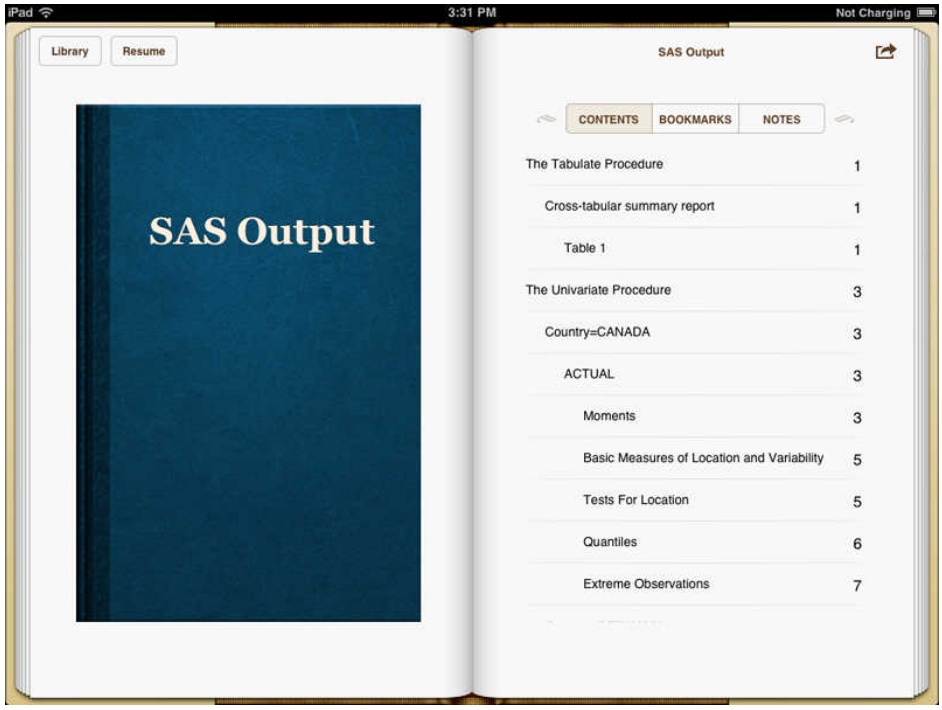
The ODS EPUB statement creates output formatted for e-book readers. The output has an .epub extension. EPUB (electronic publication) is a free and open e-book standard produced by the International Digital Publishing Forum (IDPF). EPUB output can be read by a wide variety of e-book readers, from dedicated hardware to desktop software to online-based readers.

The ODS EPUB and ODS EPUB3 destinations are the recommended destinations for creating SAS output that is accessible to the broadest audience. They create e-books that use many of the accessibility features of the EPUB specification. These features enable e-book readers such as iBooks to present e-books so that they adapt to the needs of users with disabilities. For example, when reading an e-book created by the ODS EPUB or ODS EPUB3 destination using iBooks on an iPad, users can adjust font size, color schemes, and magnification. They can also access the text using assistive technologies such as the Voiceover screen reader and refreshable Braille is displayed. Because the HTML destination is open by default, it is good practice to close the HTML destination if you do not want HTML output. This saves system resources.

```
ods html close;

ods epub3 file="your-file-path/prdsaleBookSimple.epub";
title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale;
    class region division prodtype;
    classlev region division prodtype;
    var actual;
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];
 run;

title "PROC UNIVARIATE Output";
title2;
proc univariate data=prdsale;
    by Country;
    var actual;
run;

title 'Sales Figures for First Quarter by Product';

proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
```

```
run;

title "PROC PRINT Output";
proc print data=sashelp.prdsale;
run;

ods epub3 close;
```

Output 5.3 *E-Book Viewed in iBooks*



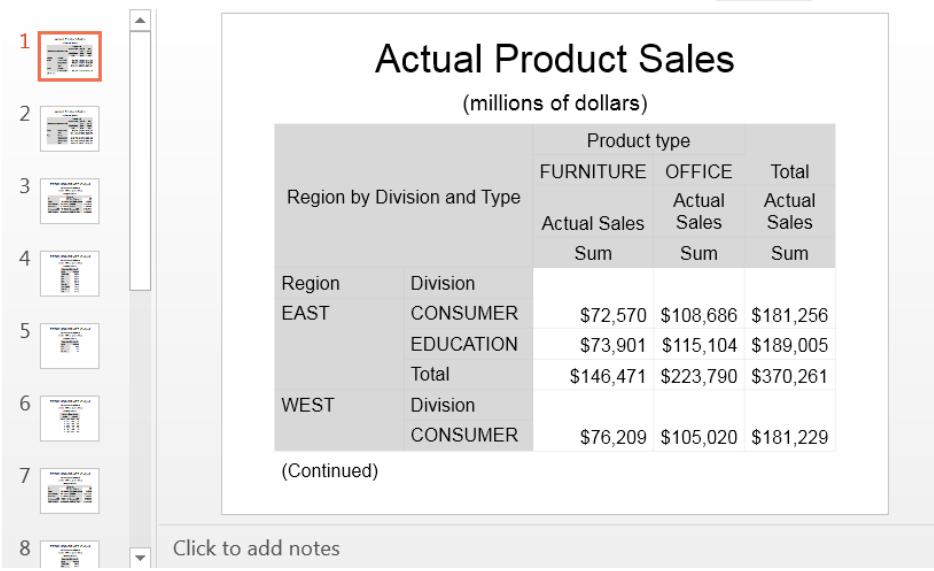
Creating PowerPoint Output

The ODS POWERPOINT statement creates output formatted for PowerPoint. Suppose you want PowerPoint slides for all of your output. To create this output, wrap (sandwich) the ODS POWERPOINT statement and ODS POWERPOINT CLOSE statement around your program. Use the FILE= option to specify the name and path for your file. The output opens in PowerPoint. Because the HTML destination is open by default, it is good practice to close the HTML destination if you do not want HTML output. This saves system resources.

```
ods html close;

options nodate nonumber;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;

ods powerpoint file='your-file-path/Prdsale.ppt';
title 'Actual Product Sales';
title2 '(millions of dollars)';
```

```
proc tabulate data=prdsale;
    class region division prodtype;
    classlev region division prodtype;
    var actual;
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];
 run;

title "PROC UNIVARIATE Output";
title2;
proc univariate data=prdsale;
    by Country;
    var actual;
run;

title 'Sales Figures for First Quarter by Product';

proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;

title "PROC PRINT Output";
proc print data=sashelp.prdsale;
run;

ods powerpoint close;
```

Output 5.4 *PROC TABULATE Output Viewed in PowerPoint*



Actual Product Sales

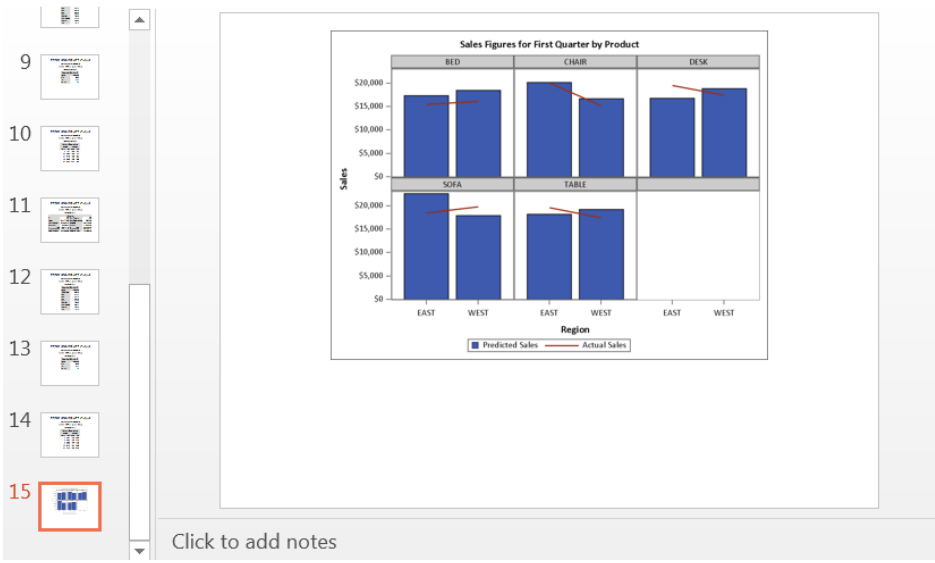
(millions of dollars)

Region by Division and Type		Product type		
		FURNITURE	OFFICE	Total
		Actual Sales	Actual Sales	Actual Sales
		Sum	Sum	Sum
Region	Division			
EAST	CONSUMER	$72,570	$108,686	$181,256
	EDUCATION	$73,901	$115,104	$189,005
	Total	$146,471	$223,790	$370,261
WEST	Division			
	CONSUMER	$76,209	$105,020	$181,229

(Continued)

Click to add notes

Output 5.5 *PROC SGPANEL Output Viewed in PowerPoint*



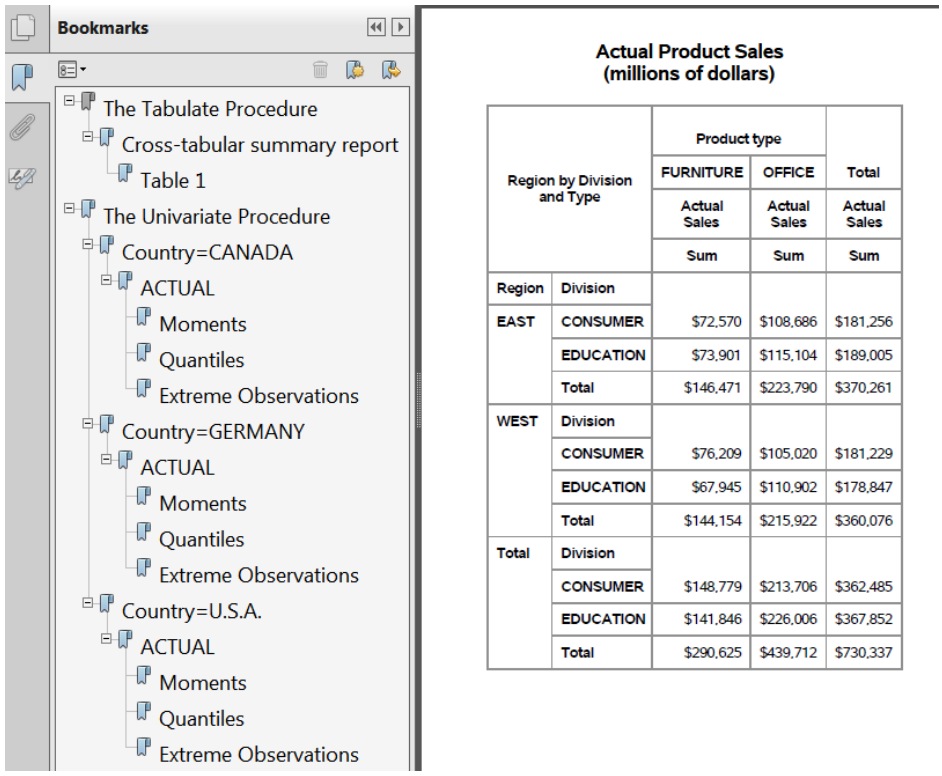
Creating PDF Output

You can generate output that is formatted for Adobe Acrobat software. To create PDF output that contains PROC TABULATE and PROC UNIVARIATE output, wrap (sandwich) the ODS PDF statement and ODS PDF CLOSE statement around your program. Use the FILE= option to specify the name and path for your file. Because the HTML destination is open by default, it is good practice to close the HTML destination if you do not want HTML output. This saves system resources.

```
ods html close;

options nodate nonumber;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;
ods pdf file='your-file-path/PDFPrdsale.pdf';
title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale;
    class region division prodtype;
    classlev region division prodtype;
    var actual;
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];
run;

ods select ExtremeObs Quantiles Moments;
title "PROC UNIVARIATE Output";
```

```
title2;

proc univariate data=prdsale;
    by Country;
    var actual;
run;
ods pdf close;
```

Output 5.6 *Default Output Viewed in Adobe Acrobat*



Actual Product Sales
(millions of dollars)

Region by Division and Type		Product type		
		FURNITURE	OFFICE	Total
		Actual Sales	Actual Sales	Actual Sales
		Sum	Sum	Sum
Region	Division			
EAST	CONSUMER	$72,570	$108,686	$181,256
	EDUCATION	$73,901	$115,104	$189,005
	Total	$146,471	$223,790	$370,261
WEST	Division			
	CONSUMER	$76,209	$105,020	$181,229
	EDUCATION	$67,945	$110,902	$178,847
	Total	$144,154	$215,922	$360,076
Total	Division			
	CONSUMER	$148,779	$213,706	$362,485
	EDUCATION	$141,846	$226,006	$367,852
	Total	$290,625	$439,712	$730,337

Creating Enhanced HTML Output

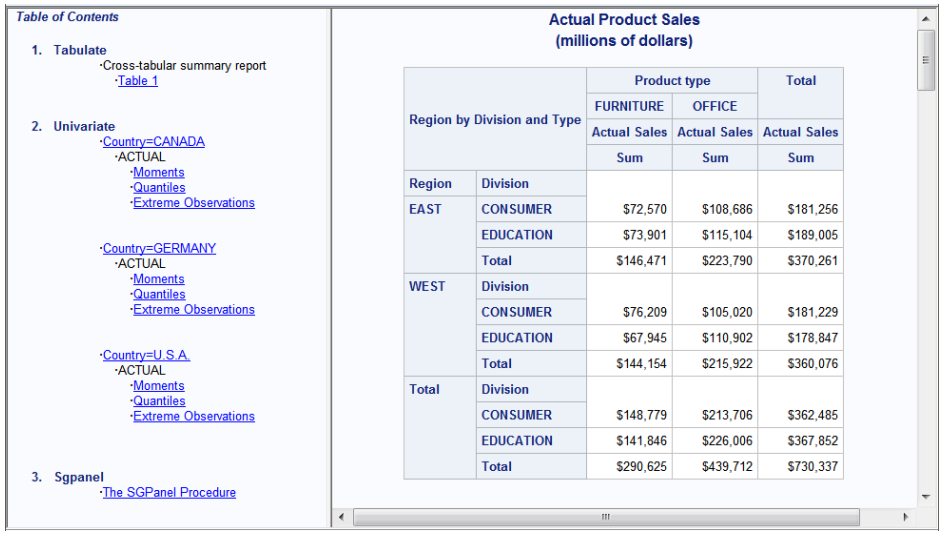
In the SAS windowing environment, by default, if you have not closed and reopened the ODS HTML destination, HTML output is created. You can specify options in the ODS HTML statement to create frame, contents, page, and body files. In the example below, when you run the program, the body file is displayed in the Results Viewer. To view the body file, contents file, and frame file together as a single HTML page, open the frame file from your local directory.

The PATH= option in the ODS HTML statement specifies the directory, folder, or partitioned data set that you are using to store your files. Once you have specified PATH='*your-directory-path*' in the ODS HTML statement, the files html-bodyPrdsale.htm, html-contentsPrdsale.htm, and html-framePrdsale.htm are automatically stored in '*your-directory-path*'.

```
options nodate nonumber;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;
```

```
ods html path='your-directory-path' body='html-bodyPrdsale.htm'
        contents='html-contentsPrdsale.htm'
        frame='html-framePrdsale.htm';
title 'Actual Product Sales';
title2 '(millions of dollars)';
proc tabulate data=prdsale;
    class region division prodtype;
    classlev region division prodtype;
    var actual;
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];
run;

title "PROC UNIVARIATE Output";
title2;

ods select ExtremeObs Quantiles Moments;
proc univariate data=prdsale;
    by Country;
    var actual;
run;

title 'Sales Figures for First Quarter by Product';

proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;
ods html close;
```

Output 5.7 *Frame File Created with the ODS HTML Statement Displayed in Internet Explorer*



Region by Division and Type		Product type		Total
		FURNITURE	OFFICE	
		Actual Sales	Actual Sales	Actual Sales
		Sum	Sum	Sum
Region	Division			
EAST	CONSUMER	$72,570	$108,686	$181,256
	EDUCATION	$73,901	$115,104	$189,005
	Total	$146,471	$223,790	$370,261
WEST	Division			
	CONSUMER	$76,209	$105,020	$181,229
	EDUCATION	$67,945	$110,902	$178,847
	Total	$144,154	$215,922	$360,076
Total	Division			
	CONSUMER	$148,779	$213,706	$362,485
	EDUCATION	$141,846	$226,006	$367,852
	Total	$290,625	$439,712	$730,337

Creating Excel Output

To view output in Excel, you can specify that your original data set be formatted for Excel. Use the ODS TAGSETS.EXCELXP statement to open the Prdsale data set in Excel.

```
ods tagsets.excelxp file='your-file-path/Prdsale.xls';
title "PROC PRINT Output";

proc print data=sashelp.prdsale;
run;
ods tagsets.excelxp close;
```

Output 5.8 *Prdsale Data Set Opened in Excel*

Obs	ACTUAL	PREDICT	COUNTRY	REGION	DIVISION	PROD
1	$925.00	$850.00	CANADA	EAST	EDUCATION	FURNIT
2	$999.00	$297.00	CANADA	EAST	EDUCATION	FURNIT
3	$608.00	$846.00	CANADA	EAST	EDUCATION	FURNIT
4	$642.00	$533.00	CANADA	EAST	EDUCATION	FURNIT
5	$656.00	$646.00	CANADA	EAST	EDUCATION	FURNIT
6	$948.00	$486.00	CANADA	EAST	EDUCATION	FURNIT
7	$612.00	$717.00	CANADA	EAST	EDUCATION	FURNIT
8	$114.00	$564.00	CANADA	EAST	EDUCATION	FURNIT
9	$685.00	$230.00	CANADA	EAST	EDUCATION	FURNIT
10	$657.00	$494.00	CANADA	EAST	EDUCATION	FURNIT

Table 1 - Data Set SASHELP.PRDS

Combined Program

Now that you have created output for the various applications that you want to use, you can combine the different programs into a single program. You can create an ODS

document to store your output objects. You can then replay your output or modify your output objects without having to resubmit any procedures.

CAUTION:

The following program contains numbered callouts that help explain specific parts of the program. When you copy and paste this program into a SAS code editor, the callouts are copied as well, which results in errors. If you want to copy and paste this program, use Example Code 5.2 on page 53.

Example Code 5.1 *Combined Program with Callouts*

```
options nodate nonumber;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;
1 ods document name=work.prddocument(write);

2 ods html path="your-directory-path"
        body="html-bodyPrdsale.htm"
        contents="html-contentsPrdsale.htm"
        frame="html-framePrdsale.htm";
3 ods pdf file="your-file-path/PDFPrdsale.pdf";
4 ods powerpoint file="your-file-path/Prdsale.ppt";
5 ods rtf file="your-file-path/RTFPrdsale.rtf";
6 ods epub file="your-file-path/prdsaleBookSimple.epub";
title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale;
    class region division prodtype;
    classlev region division prodtype;
    var actual;
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];
run;

7 ods rtf exclude all;
8 ods select ExtremeObs Quantiles Moments;

title "PROC UNIVARIATE Output";
title2;

proc univariate data=prdsale;
    by Country;
    var actual;
run;

9 ods rtf select all;
10 ods pdf select none;

title 'Sales Figures for First Quarter by Product';
proc sgpanel data=prdsale;
    where quarter=1;
```

```
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;

11 ods exclude PRINT;
12 ods tagsets.excelxp file='your-file-path.Prdsale.xls';
13 ods tagsets.excelxp select PRINT;
title "PROC PRINT Output";

proc print data=sashelp.prdsale;
run;
14 ods _all_ close;
15 ods html;
```

1 The ODS DOCUMENT statement creates the document Work.PrdDocument. Work.PrdDocument stores all of the output generated between the opening ODS DOCUMENT statement and the ODS _ALL_ CLOSE statement.

2 The ODS HTML statement specifies the names and paths for the body, contents, and frame files.

3 The ODS PDF statement with the FILE= option opens the ODS PDF destination (which is a member of the PRINTER family of destinations). It specifies the name and path for the PDF output file.

4 The ODS POWERPOINT statement with the FILE= option opens the ODS POWERPOINT destination. It specifies the name and path for the output file.

5 The ODS RTF statement with the FILE= option opens the ODS RTF destination. It specifies the name and path for the output file.

6 The ODS EPUB statement with the FILE= option opens the ODS EPUB destination. It specifies the name and path for the output file.

7 The ODS RTF statement with the EXCLUDE ALL option excludes all of the output objects from the following PROC UNIVARIATE output.

8 The ODS SELECT statement specifies that the output objects ExtremeObs, Quantiles, and Moments are sent to all open destinations that do not specifically exclude PROC UNIVARIATE output with the EXCLUDE option, such as the previous ODS RTF statement. The ODS statement with the SELECT or EXCLUDE option must be specified after the opening ODS statement.

9 The ODS RTF statement with the SELECT ALL option selects all of the output objects from the following PROC SGPANEL output. It sends the output objects to the ODS RTF destination. The ODS statement with the SELECT or EXCLUDE option must be specified after the opening ODS statement.

10 The ODS PDF statement with the SELECT NONE option selects none of the output objects from the following PROC SGPANEL output. The ODS statement with the SELECT or EXCLUDE option must be specified after the opening ODS statement.

11 The ODS EXCLUDE statement excludes the output object named Print from all open destinations that do not specifically select the Print output object with the SELECT option.

12 The ODS TAGSETS.EXCELXP statement with the FILE= option opens the TAGSETS.EXCELXP destination (which is a member of the MARKUP family of destinations). It specifies the name and path for the XLS output file. You can use

the .XML extension instead of the EXCELXP extension. Beginning in Excel 2007, using the XLS extension invokes a dialog box because of the new security feature that matches the content with the extension.

13 The ODS TAGSETS.EXCELXP statement with the SELECT option selects the output object named Print.

14 The ODS _ALL_ CLOSE statement closes all of the open destinations. This statement is useful when you have multiple destinations open at the same time.

15 Because the ODS _ALL_ CLOSE statement closes all open destinations, it is a good practice to open the ODS HTML destination again at the end of your program. If all of the destinations are closed, you get the following warning in the SAS Log: **`WARNING: No output destinations active`**.

Example Code 5.2 *Combined Program without Callouts*

```
options nodate nonumber;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;
ods document name=work.prddocument(write);

ods html path="your-directory-path"
        body="html-bodyPrdsale.htm"
        contents="html-contentsPrdsale.htm"
        frame="html-framePrdsale.htm";
ods pdf file="your-file-path/PDFPrdsale.pdf";
ods powerpoint file="your-file-path/Prdsale.ppt";
ods rtf file="your-file-path/RTFPrdsale.rtf";
ods epub file="your-file-path/prdsaleBookSimple.epub";
title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale;
    class region division prodtype;
    classlev region division prodtype;
    var actual;
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];
run;

ods rtf exclude all;
ods select ExtremeObs Quantiles Moments;

title "PROC UNIVARIATE Output";
title2;

proc univariate data=prdsale;
    by Country;
    var actual;
run;

ods rtf select all;
```

```
ods pdf select none;

title 'Sales Figures for First Quarter by Product';
proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;

ods exclude PRINT;
ods tagsets.excelxp file='your-file-path.Prdsale.xls';
ods tagsets.excelxp select PRINT;
title "PROC PRINT Output";

proc print data=sashelp.prdsale;
run;
ods _all_ close;
ods html;
```

For More Information

- For complete documentation on ODS statements, see "Dictionary of ODS Language Statements" in *SAS Output Delivery System: User's Guide*.

Chapter 6

Customizing the Presentation of a Report

About the Tasks That You Will Perform 55
Customized RTF Output 57
Customized E-Book 58
Customized PowerPoint Output 60
Customized PROC TABULATE Output 62
Customized PDF Output 64
Customized HTML Output 66
Customized Excel Output 67
Combined Program 69
For More Information 73

About the Tasks That You Will Perform

Once you have specified the ODS statements that you need to generate the appropriate output type, you can add options to those statements to customize the presentation of your output. The quickest way to make a change to all of your output at once is to specify the STYLE= option in the ODS destination statement.

You can view all styles that are provided by SAS in the Templates window. To open the Templates window, do one of the following:

- Select **Results** in the Results window, and then select **View** ⇨ **Templates** from the taskbar.
- Right-click **Results** in the Results window, and then select **Templates**.
- Issue the following command on the command line in the SAS windowing environment:

  ```
  odstemplates
  ```

To view the styles provided by SAS, select **Templates** ⇨ **Sashelp.Tmplmst** ⇨ **Styles**.

Output 6.1 *SAS Styles*

Contents of 'Styles'

Analysis	HTMLBlue	NoFontDefault	SeasidePrinter
BarrettsBlue	Journal	Normal	StatDoc
BlockPrint	Journal1a	NormalPrinter	Statistical
Daisy	Journal2	Ocean	vaDark
Default	Journal2a	Pearl	vaHighContrast
Dove	Journal3	Plateau	vaLight
Dtree	Journal3a	PowerPointDark	
EGDefault	Listing	PowerPointLight	
FancyPrinter	Meadow	Printer	
Festival	MeadowPrinter	Raven	
FestivalPrinter	Minimal	Rtf	
Gantt	MonochromePrinter	Sapphire	
GrayscalePrinter	Monospace	SasDocPrinter	
Harvest	Moonflower	SasWeb	
HighContrast	Netdraw	Seaside	

By default (for the SAS windowing environment), ODS uses the following styles for each output destination:

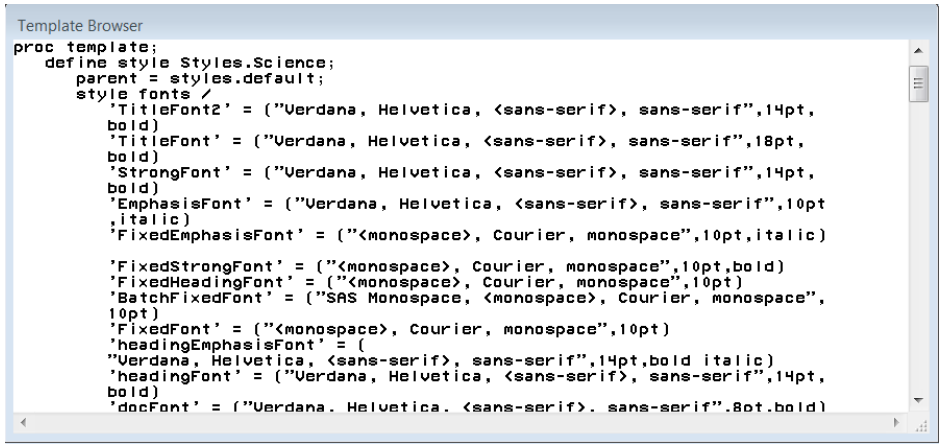
Table 6.1 *Table of Default Styles*

Output Destination	Default Style
RTF	RTF
HTML	HTMLBlue
PDF	Pearl
EPUB	Daisy
POWERPOINT	PowerPointLight
EXCEL	Default

The default style for each ODS destination can be viewed in the SAS Registry Editor. For the following examples, the Science style is used to ensure that the look of the presentation is consistent. To view the source of the Science style in the Template Browser window, do one of the following:

- Double-click the **Science** style.
- Right-click **Science**, and then select **Open**.

Figure 6.1 *Source of Science Style Viewed in the Template Browser Window*



You can customize specific areas of some procedure output by specifying the STYLE= option in specific procedure statements. With the STYLE= option, you can change the attributes of cells. Changes include aligning text, adding URLs, changing rules, and changing the size, width, and color of a font.

The STYLE= option is specified in slightly different ways for individual procedure statements. However, the general form of the STYLE= option for a procedure statement is:

STYLE=<*style-attribute-name=style-attribute-value*>

The procedures that allow direct style control are PROC PRINT, PROC REPORT, and PROC TABULATE. For documentation on these procedures, see *Base SAS Procedures Guide*. For information about using ODS and the SAS Registry Editor, see "Changing SAS Registry Settings for ODS" in *SAS Output Delivery System: User's Guide*.

Customized RTF Output

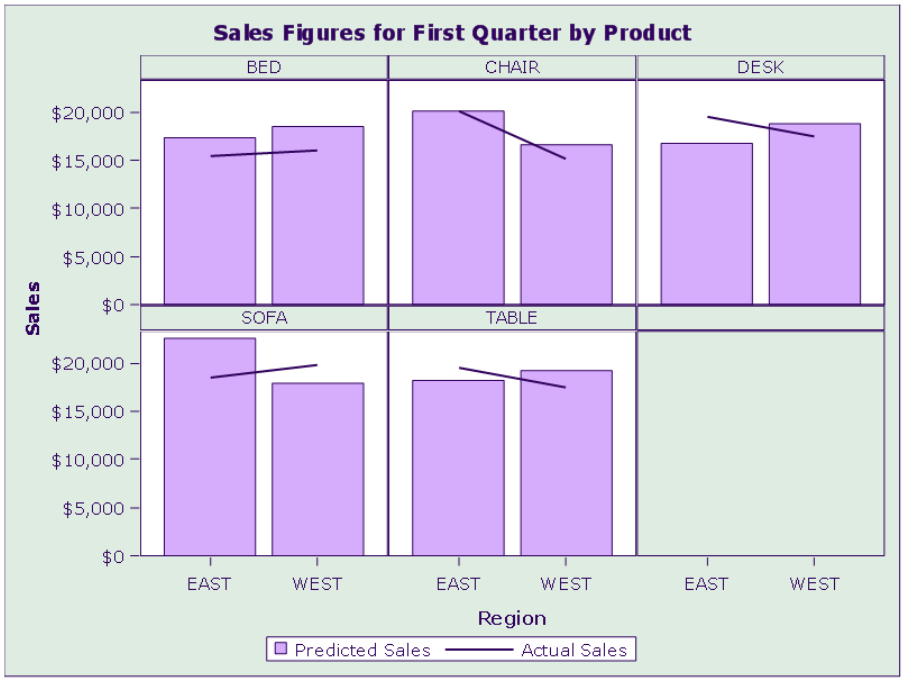
To quickly change the look of your RTF output, specify the STYLE= option in the ODS RTF statement. The STYLE= option with the Science style specified tells ODS to use the Science style instead of the default style for RTF output. Because the HTML destination is open by default, it is good practice to close the HTML destination if you do not want HTML output. This saves system resources.

```
ods html close;
options nodate nonumber;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;
ods rtf file='your-file-path/RTFPrdsaleCustom.rtf' style=Science;

title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale;
    class region division prodtype;
    classlev region division prodtype;
    var actual;
```

```
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];
 run;

title2;
title 'Sales Figures for First Quarter by Product';

proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;
ods rtf close;
```

Even though you did not specify the STYLE= option in the PROC SGPANEL statement, the **`style=Science`** option in the ODS RTF statement applies the style for PROC SGPANEL, too.

Output 6.2 *Customized RTF Output Viewed in Microsoft Word*



Customized E-Book

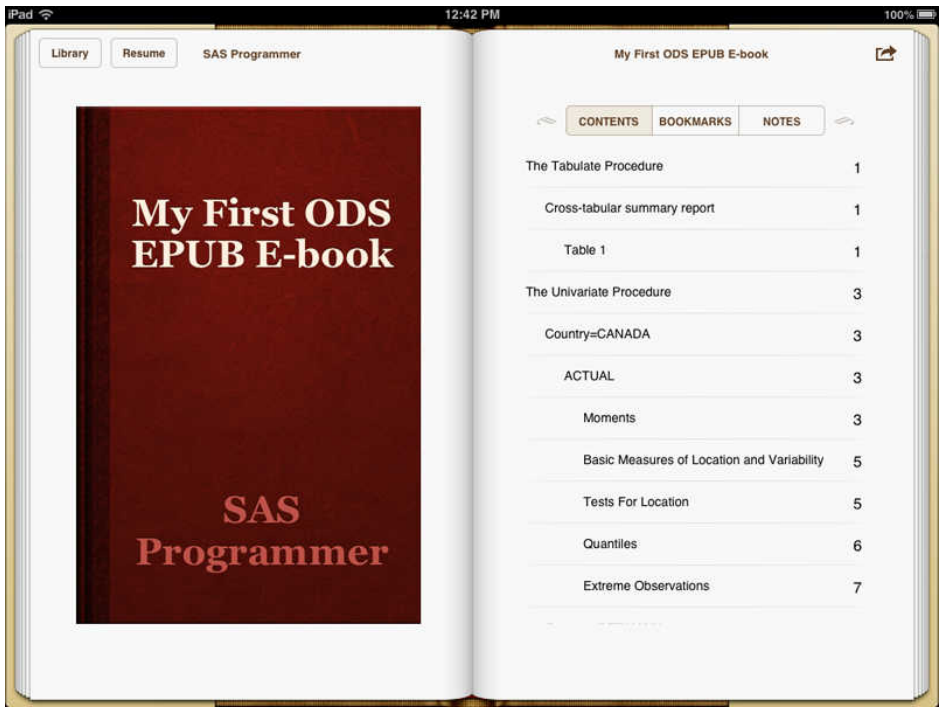
To quickly change the look of your EPUB output, specify the OPTIONS option in the ODS EPUB statement. The OPTIONS option specifies suboptions that do the following:

- specify the author with the CREATOR= suboption
- specify a description with the DESCRIPTION= suboption
- specify a subject with the SUBJECT= suboption
- specify a type with the TYPE= suboption

Because the HTML destination is open by default, it is good practice to close the HTML destination if you do not want HTML output. This saves system resources.

```
ods html close;

ods epub file="your-file-path/prdsaleBookCustom.epub"
    title="My First ODS EPUB E-book"
    options(creator="SAS Programmer"
    description="Sales Figures for First Quarter by Product"
    subject="PROC TABULATE, PROC UNIVARIATE, and PROC SGPANEL"
    type="ODS EPUB book");

title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale;
    class region division prodtype;
    classlev region division prodtype;
    var actual;
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
         box=[label='Region by Division and Type'];
 run;

title "PROC UNIVARIATE Output";
title2;
proc univariate data=prdsale;
    by Country;
    var actual;
run;

title 'Sales Figures for First Quarter by Product';

proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;

title "PROC PRINT Output";
proc print data=sashelp.prdsale;
run;

ods epub close;
```

Output 6.3 *Customized E-Book Viewed in iBooks*



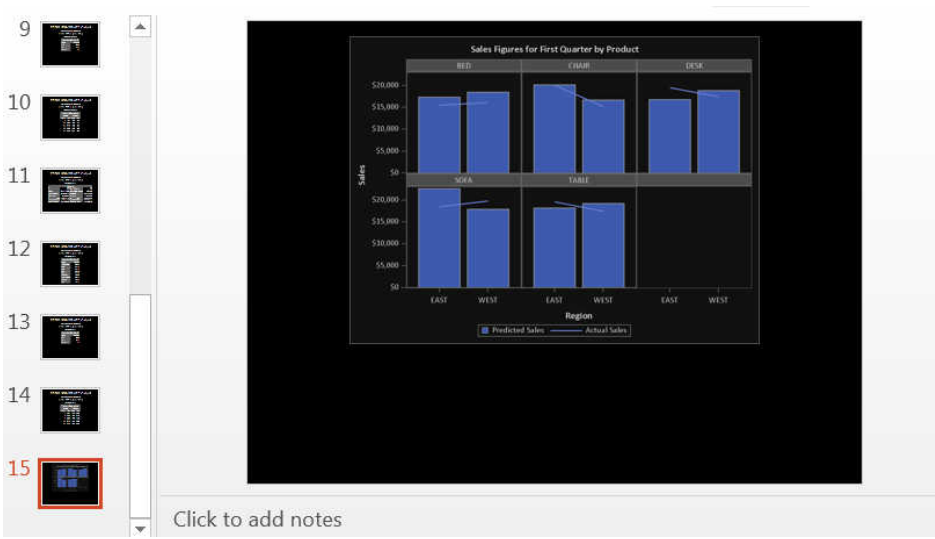
Customized PowerPoint Output

To quickly change the look of your PowerPoint output, specify the STYLE= option in the ODS POWERPOINT statement. The STYLE= option with the PowerPointDark style specified tells ODS to use the PowerPointDark style instead of the default style. The output opens in PowerPoint. Because the HTML destination is open by default, it is good practice to close the HTML destination if you do not want HTML output. This saves system resources.

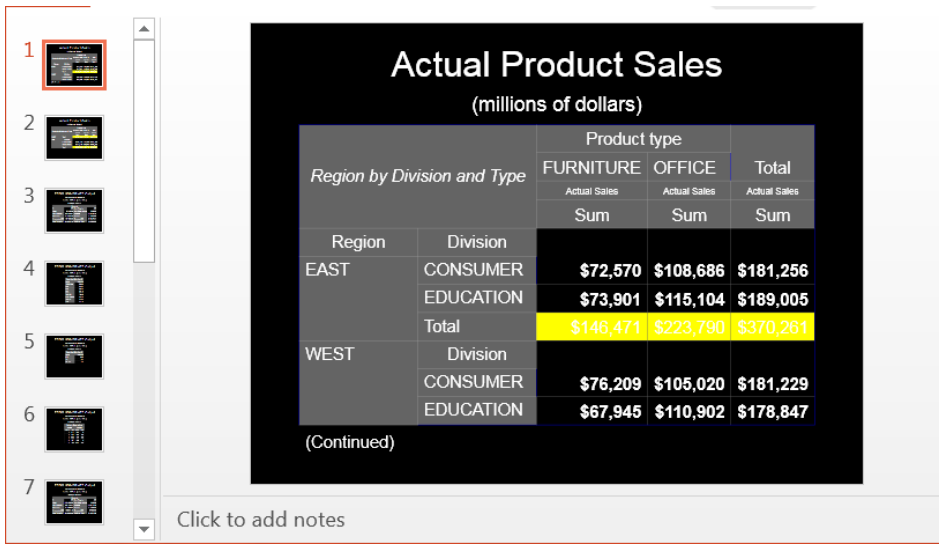
```
ods html close;

options nodate nonumber;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;

ods powerpoint file='your-file-path/PrdsaleCustom.ppt' style=PowerPointDark;
title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale;
    class region division prodtype;
    classlev region division prodtype;
    var actual;
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all),
         (prodtype all)*(actual*f=dollar10.) /
         misstext=[label='Missing']
```

```
        box=[label='Region by Division and Type'];
 run;

title "PROC UNIVARIATE Output";
title2;
proc univariate data=prdsale;
    by Country;
    var actual;
run;

title 'Sales Figures for First Quarter by Product';

proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;

title "PROC PRINT Output";
proc print data=sashelp.prdsale;
run;

ods powerpoint close;
```

Output 6.4 *PROC SGPANEL Output Viewed in PowerPoint*



Output 6.5 *PROC TABULATE Output Viewed in PowerPoint*



Customized PROC TABULATE Output

You can use ODS options in Base SAS reporting procedures, PROC PRINT, PROC REPORT, and PROC TABULATE. In each of these procedures, you can specifically specify options in individual statements. This enables you to make changes in sections of output without changing the default style of all of the output. For example, you can customize specific sections of PROC TABULATE output by specifying the STYLE= option in specific statements within the procedure. Because the HTML destination is open by default, it is good practice to close the HTML destination if you do not want HTML output. This saves system resources.

```
ods html close;

options nodate nonumber;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;
ods rtf file='your-file-path/RTFPrdsaleCustom.rtf' style=Science;

title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale 1 style=[fontweight=bold];
    class region division prodtype / 2 style=[textalign=center];
    classlev region division prodtype / 3 style=[textalign=left];
    var actual / 4 style=[fontsize=3];
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all* 5 [style=[backgroundcolor=yellow]]),
          (prodtype all)*(actual*f=dollar10.) /
          style=[bordercolor=blue] box=[label='Region by Division and Type'
          6 style=[fontstyle=italic]];
run;
```

```
title 'Sales Figures for First Quarter by Product';
title2;

proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;
ods rtf close;
```

1 The STYLE= option specified in the PROC TABULATE statement changes all of the fonts to bold for all of the data cells.

2 The STYLE= option specified in the CLASS statement centers the CLASS variable name headings.

3 The STYLE= option specified in the CLASSLEV statement left-justifies the CLASS variable level value headings.

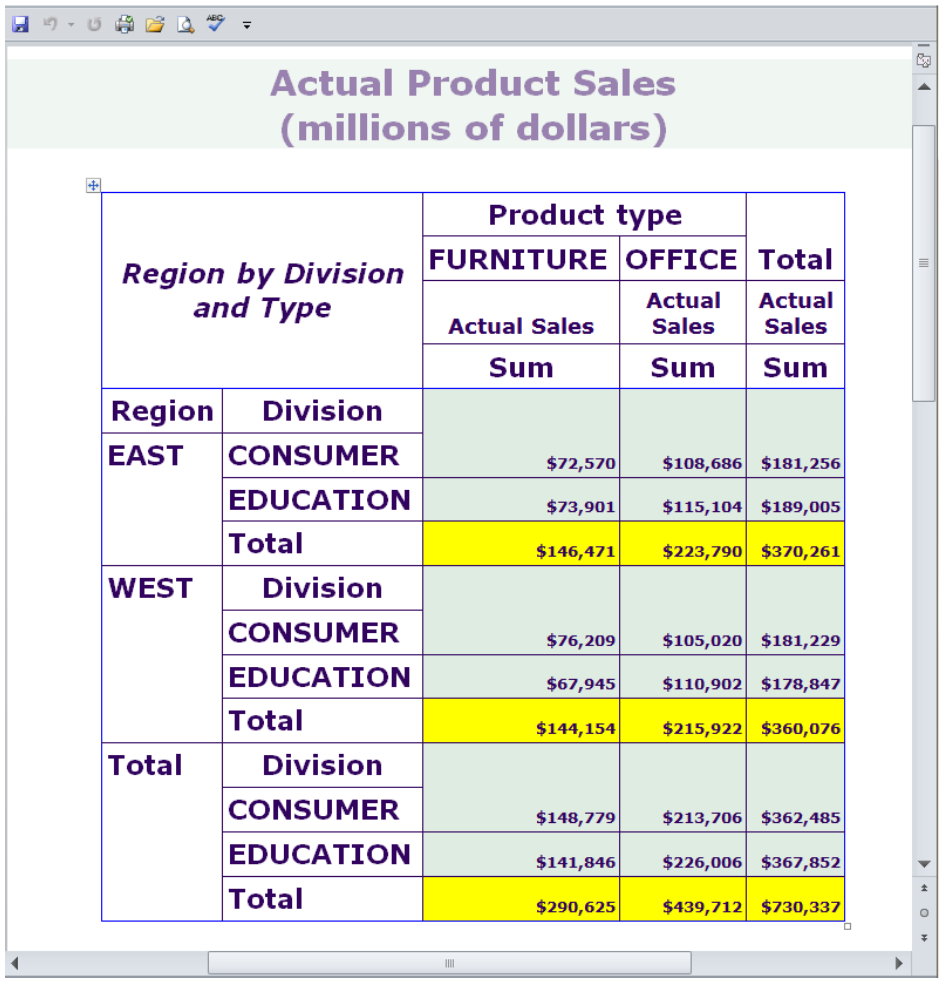
4 The STYLE= option specified in the VAR statement changes the font size of analysis variable name headings to 3.

5 The first STYLE= option specified in the TABLE statement changes the background color of the cells containing the sum totals of REGION and DIVISION to yellow.

6 The second STYLE= option specified in the TABLE statement italicizes the font of the label of the box above the row titles.

Because the STYLE= option is specified in the ODS RTF statement, PROC TABULATE output uses the Science style and the specific style overrides specified in individual statements.

Output 6.6 *Customized PROC TABULATE Output Viewed in Microsoft Word*



Actual Product Sales
(millions of dollars)

Region by Division and Type		Product type		
		FURNITURE	OFFICE	Total
		Actual Sales	Actual Sales	Actual Sales
		Sum	Sum	Sum
Region	Division			
EAST	CONSUMER	$72,570	$108,686	$181,256
	EDUCATION	$73,901	$115,104	$189,005
	Total	$146,471	$223,790	$370,261
WEST	Division			
	CONSUMER	$76,209	$105,020	$181,229
	EDUCATION	$67,945	$110,902	$178,847
	Total	$144,154	$215,922	$360,076
Total	Division			
	CONSUMER	$148,779	$213,706	$362,485
	EDUCATION	$141,846	$226,006	$367,852
	Total	$290,625	$439,712	$730,337

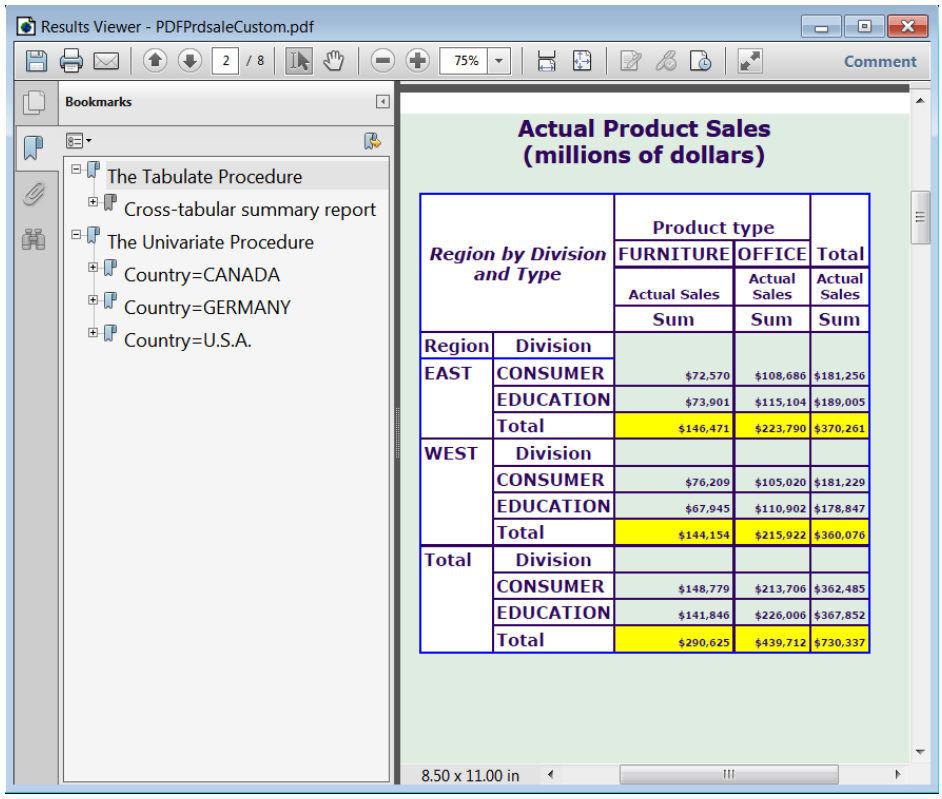
Customized PDF Output

To quickly change the look of your PDF output, specify the STYLE=, CONTENTS=, and PDFTOC= options in the ODS PDF statement. The STYLE= option with the Science style specified tells ODS to use the Science style for all PDF output. The PDFTOC= option with 2 specified expands the table of contents to two levels. Because the HTML destination is open by default, it is good practice to close the HTML destination if you do not want HTML output. This saves system resources.

```
ods html close;

options nodate nonumber orientation=portrait;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;
```

```
ods pdf file='your-file-path/PDFPrdsaleCustom.pdf' pdftoc=2 style=Science;
title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale style=[fontweight=bold];
    class region division prodtype / style=[textalign=center];
    classlev region division prodtype / style=[textalign=left];
    var actual / style=[fontsize=3];
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all*[style=[backgroundcolor=yellow]]),
          (prodtype all)*(actual*f=dollar10.) /
          style=[bordercolor=blue] style=[fontstyle=italic]];
run;

title "PROC UNIVARIATE Output";
title2;

proc univariate data=prdsale;
    by Country;
    var actual;
run;
ods pdf close;
```

Output 6.7 *Customized PDF Output Viewed in Adobe Acrobat*



Actual Product Sales
(millions of dollars)

Region by Division and Type		Product type		
		FURNITURE	OFFICE	Total
		Actual Sales	Actual Sales	Actual Sales
		Sum	Sum	Sum
Region	Division			
EAST	CONSUMER	$72,570	$108,686	$181,256
	EDUCATION	$73,901	$115,104	$189,005
	Total	$146,471	$223,790	$370,261
WEST	Division			
	CONSUMER	$76,209	$105,020	$181,229
	EDUCATION	$67,945	$110,902	$178,847
	Total	$144,154	$215,922	$360,076
Total	Division			
	CONSUMER	$148,779	$213,706	$362,485
	EDUCATION	$141,846	$226,006	$367,852
	Total	$290,625	$439,712	$730,337

Customized HTML Output

The STYLE= option with the Science style specified in the ODS HTML statement tells ODS to use the Science style for all HTML output.

```
options nodate nonumber;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;
ods html path='your-directory-path'
        body='bodyPrdsale.htm'
        contents='contentsPrdsale.htm'
        frame='framePrdsale.htm'
        style=Science;
title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale style=[fontweight=bold];
    class region division prodtype / style=[textalign=center];
    classlev region division prodtype / style=[textalign=left];
    var actual / style=[fontsize=3];
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all*[style=[backgroundcolor=yellow]]),
         (prodtype all)*(actual*f=dollar10.) /
         style=[bordercolor=blue]
         misstext=[label='Missing' style=[fontweight=light]]
         box=[label='Region by Division and Type'
         style=[fontstyle=italic]];
 run;

ods select ExtremeObs Quantiles Moments;
title "PROC UNIVARIATE Output";
title2;

proc univariate data=prdsale;
    by Country;
    var actual;
run;

title 'Sales Figures for First Quarter by Product';
proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;
ods html close;
```

Output 6.8 *Customized Frame File Created and Viewed in Internet Explorer*

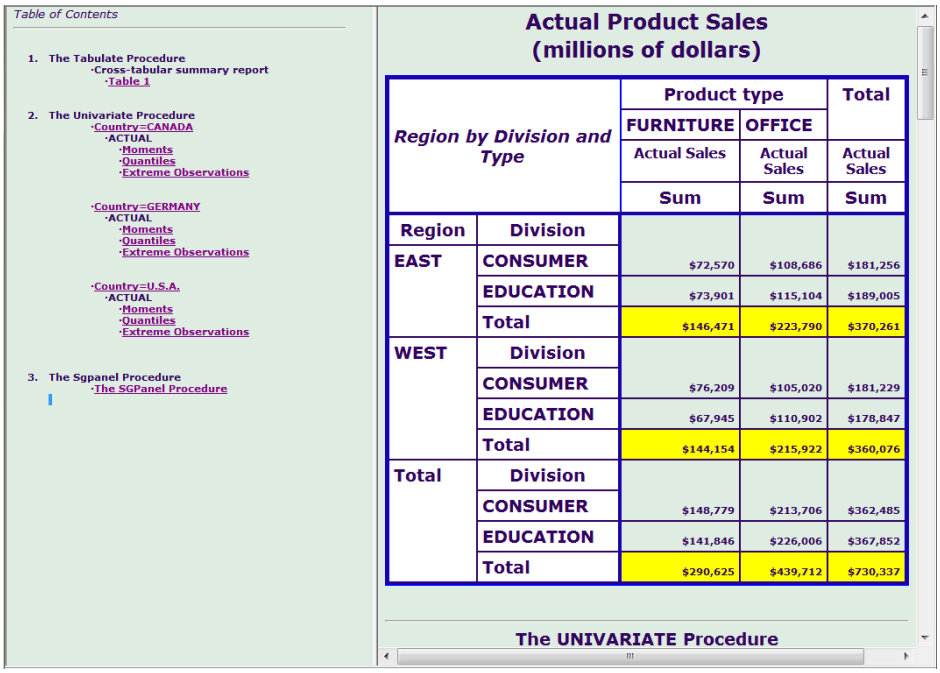


Table of Contents

1. The Tabulate Procedure
   - Cross-tabular summary report
     - Table 1
2. The Univariate Procedure
   - Country=CANADA
     - ACTUAL
       - Moments
       - Quantiles
       - Extreme Observations
   - Country=GERMANY
     - ACTUAL
       - Moments
       - Quantiles
       - Extreme Observations
   - Country=U.S.A.
     - ACTUAL
       - Moments
       - Quantiles
       - Extreme Observations
3. The Sgpanel Procedure
   - The SGPanel Procedure

Actual Product Sales
(millions of dollars)

Region by Division and Type		Product type		Total
		FURNITURE	OFFICE	
		Actual Sales	Actual Sales	Actual Sales
		Sum	Sum	Sum
Region	Division			
EAST	CONSUMER	$72,570	$108,686	$181,256
	EDUCATION	$73,901	$115,104	$189,005
	Total	$146,471	$223,790	$370,261
WEST	Division			
	CONSUMER	$76,209	$105,020	$181,229
	EDUCATION	$67,945	$110,902	$178,847
	Total	$144,154	$215,922	$360,076
Total	Division			
	CONSUMER	$148,779	$213,706	$362,485
	EDUCATION	$141,846	$226,006	$367,852
	Total	$290,625	$439,712	$730,337

The UNIVARIATE Procedure

Customized Excel Output

The STYLE= option with the Science style specified in the ODS TAGSETS.EXCELXP statement tells ODS to use the Science style for all Excel output. Specifying the OPTIONS suboption (DOC="HELP") prints Help for the ODS TAGSETS.EXCELXP statement suboptions to the SAS log file. All tagsets have instream help. For more information about the DOC= suboption, see the ODS Tagset statement in "Dictionary of ODS Language Statements" in *SAS Output Delivery System: User's Guide*.

```
ods tagsets.excelxp file='your-file-path/Prdsale.xls' style=Science options
(doc="help");
title "PROC PRINT Output";

proc print data=sashelp.prdsale;
run;
ods tagsets.excelxp close;
```

Output 6.9 *Customized Excel Output*

A1 | Obs

Prdsale.xls

	A	B	C	D	E
1	Obs	ACTUAL	PREDICT	COUNTRY	REGION
2	1	$925.00	$850.00	CANADA	EAST
3	2	$999.00	$297.00	CANADA	EAST
4	3	$608.00	$846.00	CANADA	EAST
5	4	$642.00	$533.00	CANADA	EAST
6	5	$656.00	$646.00	CANADA	EAST
7	6	$948.00	$486.00	CANADA	EAST
8	7	$612.00	$717.00	CANADA	EAST
9	8	$114.00	$564.00	CANADA	EAST
10	9	$685.00	$230.00	CANADA	EAST
11	10	$657.00	$494.00	CANADA	EAST
12	11	$608.00	$903.00	CANADA	EAST
13	12	$353.00	$266.00	CANADA	EAST
14	13	$107.00	$190.00	CANADA	EAST
15	14	$354.00	$139.00	CANADA	EAST
16	15	$101.00	$217.00	CANADA	EAST
17	16	$553.00	$560.00	CANADA	EAST
18	17	$877.00	$148.00	CANADA	EAST
19	18	$431.00	$762.00	CANADA	EAST

Figure 6.2 *Help for the EXCELXP Tagset*

Log - (Untitled)

```
50    ods tagsets.excelxp file='Prdsale.xls' style=Science options (doc="help");
NOTE: Writing TAGSETS.EXCELXP Body file: Prdsale.xls
==============================================================================
The EXCELXP Tagset Help Text.

This Tagset/Destination creates Microsoft's spreadsheetML XML.
It is used specifically for importing data into Excel.

Each table will be placed in its own worksheet within a workbook.
This destination supports ODS styles, traffic lighting, and custom formats.

Numbers, Currency and percentages are correctly detected and displayed.
Custom formats can be given by supplying a style override on the tagattr
style element.

By default, titles and footnotes are part of the spreadsheet, but are part
of the header and footer.

Also by default, printing will be in 'Portrait'.
The orientation can be changed to landscape.

The specification for this xml is here.
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnexcl2k2/html/odc_xlsmlin

See Also:
http://support.sas.com/rnd/base/topics/odsmarkup/
http://support.sas.com/rnd/papers/index.html#excelxml


Sample usage:

ods tagsets.excelxp file='test.xml' contents='index.xml' data='test.ini' options(doc='Help'

ods tagsets.excelxp options(doc='Quick');

ods tagsets.excelxp options(embedded_titles='No' Orientation='Landscape');

==============================================================================

Long descriptions of the supported options

Doc:  No default value.
     Help: Displays introductory text and available options in full detail.
     Quick: Displays introductory text and an alphabetical list of options,
            their current value, and a short description
```

Combined Program

Now that you have created output for the various applications that you want to use, you can combine the different programs into a single program. You can create an ODS document to store your output objects. You can then replay your output or modify your output objects without having to resubmit any procedures.

CAUTION:

The following program contains numbered callouts that help explain specific parts of the program. When you copy and paste this program into a SAS code editor, the callouts are copied as well, which results in errors. If you want to copy and paste this program, use Example Code 6.2 on page 71.

Example Code 6.1 *Combined Program with Callouts*

```
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;
options nodate nonumber;
1 ods document name=work.prddocument(write);

2 ods html path='your-directory-path' body='bodyPrdsale.htm'
        contents='contentsPrdsale.htm'
        frame='framePrdsale.htm'
        style=Science;

3 ods rtf file='your-file-path/RTFPrdsaleCustom.rtf' style=Science;

4 ods powerpoint file='your-file-path/PrdsaleCustom.ppt' style=PowerPointDark;
5 ods pdf file='your-file-path/PDFPrdsaleCustom.pdf'
        contents=yes pdftoc=2 style=Science;
6 ods epub file="your-file-path/prdsaleBookCustom.epub"
    title="My First ODS EPUB E-book"
    options(creator="SAS Programmer"
    description="Sales Figures for First Quarter by Product"
    subject="PROC TABULATE, PROC UNIVARIATE, and PROC SGPANEL"
    type="ODS EPUB book");


title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale 7 style=[fontweight=bold];
    class region division prodtype / 8 style=[textalign=center];
    classlev region division prodtype / 9 style=[textalign=left];
    var actual / 10 style=[fontsize=3];
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all* 11 [style=[backgroundcolor=yellow]]),
          (prodtype all)*(actual*f=dollar10.) /
          style=[bordercolor=blue] box=[label='Region by Division and Type'
          12 style=[fontstyle=italic]];
run;
```

```
13 ods rtf exclude all;
14 ods select ExtremeObs Quantiles Moments;
title "PROC UNIVARIATE Output";
title2;

proc univariate data=prdsale;
    by Country;
    var actual;
run;

15 ods rtf select all;
16 ods pdf select none;

title 'Sales Figures for First Quarter by Product';
proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
    rowaxis label='Sales';
run;

17 ods exclude PRINT;
18 ods tagsets.excelxp file='your-file-path.Prdsale.xls' style=Science;
19 ods tagsets.excelxp select PRINT;
title "PROC PRINT Output";
proc print data=sashelp.prdsale;
run;
20 ods _all_ close;
21 ods html;
```

1 The ODS DOCUMENT statement creates the document Work.PrdDocument. Work.PrdDocument stores all of the output generated between the opening ODS DOCUMENT statement and the ODS _ALL_ CLOSE statement.

2 The ODS HTML statement specifies the names and paths for the body, contents, frame, and page files. The STYLE=SCIENCE option applies the Science style to all HTML output.

3 The ODS RTF statement with the FILE= option opens the ODS RTF destination. It specifies the name and path for the RTF output file. The STYLE=SCIENCE option applies the Science style to all RTF output.

4 The ODS POWERPOINT statement with the FILE= option opens the ODS POWERPOINT destination. It specifies the name and path for the output file. The STYLE=PowerPointDark option applies the PowerPointDark style to all PowerPoint output.

5 The ODS PDF statement with the FILE= option opens the ODS PDF destination. It specifies the name and path for the PDF output file. The CONTENTS=YES option creates a table of contents for your PDF file. The PDFTOC=2 option specifies that the table of contents is expanded to two levels.

6 The ODS EPUB statement with the FILE= option opens the ODS EPUB destination. The OPTIONS option in the ODS EPUB statement specifies suboptions.

7 The STYLE= option specified in the PROC TABULATE statement changes all of the fonts to bold.

8 The STYLE= option specified in the CLASS statement centers the CLASS variable name headings.

9 The STYLE= option specified in the CLASSLEV statement left-justifies the CLASS variable level value headings.

10 The STYLE= option specified in the VAR statement changes the font size of ANALYSIS variable name headings to 3pt.

11 The first STYLE= option specified in the TABLE statement changes the background color of the cells containing the sum totals of REGION and DIVISION to yellow.

12 The second STYLE= option specified in the TABLE statement italicizes the font of the label of the box above the row titles.

13 The ODS RTF statement with the EXCLUDE ALL option excludes all of the output objects from the following PROC UNIVARIATE output.

14 The ODS SELECT statement specifies that the output objects ExtremeObs, Quantiles, and Moments be sent to all open destinations that do not specifically exclude PROC UNIVARIATE output with the EXCLUDE option, such as the previous ODS RTF statement. The ODS statement with the SELECT or EXCLUDE option must be specified after the opening ODS statement.

15 The ODS RTF statement with the SELECT ALL option selects all of the output objects from the following PROC SGPANEL output. It sends the output objects to the ODS RTF destination. The ODS statement with the SELECT or EXCLUDE option must be specified after the opening ODS statement.

16 The ODS PDF statement with the SELECT NONE option selects none of the output objects from the following PROC SGPANEL output. The ODS statement with the SELECT or EXCLUDE option must be specified after the opening ODS statement.

17 The ODS EXCLUDE statement excludes the output object named Print from all open destinations that do not specifically select the Print output object with the SELECT option.

18 The ODS TAGSETS.EXCELXP statement with the FILE= option opens the TAGSETS.EXCELXP destination (which is a member of the MARKUP family of destinations). It specifies the name and path for the XLS output file. You can use the .XML extension instead of the EXCELXP extension. Beginning in Excel 2007, using the XLS extension will invoke a dialog box because of a new security feature that matches the content with the extension. The STYLE=SCIENCE option applies the Science style to all TAGSETS.EXCELXP output.

19 The ODS TAGSETS.EXCELXP statement with the SELECT option selects the output object named Print.

20 The ODS _ALL_ CLOSE statement closes all of the open destinations. This statement is useful when you have multiple destinations open at the same time.

21 Because the ODS _ALL_ CLOSE statement closes all open destinations, it is a good practice to open the ODS HTML destination again at the end of your program. If all of the destinations are closed, you get the following warning in the SAS Log: **`WARNING: No output destinations active`**.

Example Code 6.2 *Combined Program without Callouts*

```
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;
```

```
options nodate nonumber;
ods document name=work.prddocument(write);

ods html path='your-directory-path' body='bodyPrdsale.htm'
        contents='contentsPrdsale.htm'
        frame='framePrdsale.htm'
        style=Science;

ods rtf file='your-file-path/RTFPrdsaleCustom.rtf' style=Science;
ods powerpoint file='your-file-path/PrdsaleCustom.ppt' style=PowerPointDark;
ods pdf file='your-file-path/PDFPrdsaleCustom.pdf'
        contents=yes pdftoc=2 style=Science;
ods epub file="your-file-path/prdsaleBookCustom.epub"
    title="My First ODS EPUB E-book"
    options(creator="SAS Programmer"
    description="Sales Figures for First Quarter by Product"
    subject="PROC TABULATE, PROC UNIVARIATE, and PROC SGPANEL"
    type="ODS EPUB book");


title 'Actual Product Sales';
title2 '(millions of dollars)';

proc tabulate data=prdsale style=[fontweight=bold];
    class region division prodtype / style=[textalign=center];
    classlev region division prodtype / style=[textalign=left];
    var actual / style=[fontsize=3];
    keyword all sum;
    keylabel all='Total';
    table (region all)*(division all*[style=[backgroundcolor=yellow]]),
          (prodtype all)*(actual*f=dollar10.) /
          style=[bordercolor=blue] box=[label='Region by Division and Type'
          style=[fontstyle=italic]];
run;

ods rtf exclude all;
ods select ExtremeObs Quantiles Moments;

title "PROC UNIVARIATE Output";
title2;

proc univariate data=prdsale;
    by Country;
    var actual;
run;

ods rtf select all;
ods pdf select none;

title 'Sales Figures for First Quarter by Product';
proc sgpanel data=prdsale;
    where quarter=1;
    panelby product / novarname;
    vbar region / response=predict;
    vline region / response=actual lineattrs=GraphFit;
    colaxis fitpolicy=thin;
```

```
    rowaxis label='Sales';
run;

ods exclude PRINT;
ods tagsets.excelxp file='your-file-path.Prdsale.xls' style=Science;
ods tagsets.excelxp select PRINT;
title "PROC PRINT Output";

proc print data=sashelp.prdsale;
run;
ods _all_ close;
ods html;
```

For More Information

- For information about the ODS styles provided by SAS, see "Style Templates" in *SAS Output Delivery System: User's Guide*.
- For conceptual information about ODS statements, see "Introduction to ODS Language Statements" in *SAS Output Delivery System: User's Guide*.
- For syntax and usage information about ODS statements, see "Dictionary of ODS Language Statements" in *SAS Output Delivery System: User's Guide*.
- For conceptual information about styles, see "Output Delivery System: Basic Concepts" in *SAS Output Delivery System: User's Guide*.
- For complete documentation on the Base SAS reporting procedures, see the *Base SAS Procedures Guide*.
- For information about creating and updating ODS styles, see "TEMPLATE Procedure: Creating a Style Template " in *SAS Output Delivery System: Procedures Guide*.
- For specific information about specifying the STYLE= option in PROC TABULATE statements, see "TABULATE Procedure" in *Base SAS Procedures Guide*.

Chapter 7

Next Steps: A Quick Look at Advanced Features

Working with the TEMPLATE Procedure 75
Introduction to the TEMPLATE Procedure 75
What Can You Do with the TEMPLATE Procedure? 76

Working with ODS Graphics 81

Advanced Features of the DOCUMENT Procedure 85
Overview 85
Working with the DOCUMENT Procedure 86

ODS and the DATA Step 89

Working with Cascading Style Sheets (CSS) 91

Working with the ODSTEXT Procedure 93

Working with the ODSLIST Procedure 94

Working with the ODSTABLE Procedure 96

Working with the Report Writing Interface 97

Working with ODS LAYOUT 100

Where to Go from Here 103

Working with the TEMPLATE Procedure

Introduction to the TEMPLATE Procedure

The TEMPLATE procedure enables you to customize the appearance of your SAS output. For example, you can create, extend, or modify existing templates for various types of output, such as the following:

- styles
- tables
- crosstabulation tables
- columns
- headers
- footers

- tagsets
- ODS Graphics

ODS uses these templates to produce formatted output.

You can use the TEMPLATE procedure to navigate and manage the templates stored in template stores. Here are some tasks that you can do with PROC TEMPLATE:

- Edit an existing template.
- Create links to an existing template.
- Change the location where you write new templates.
- Search for existing templates.
- View the source code of a template.

What Can You Do with the TEMPLATE Procedure?

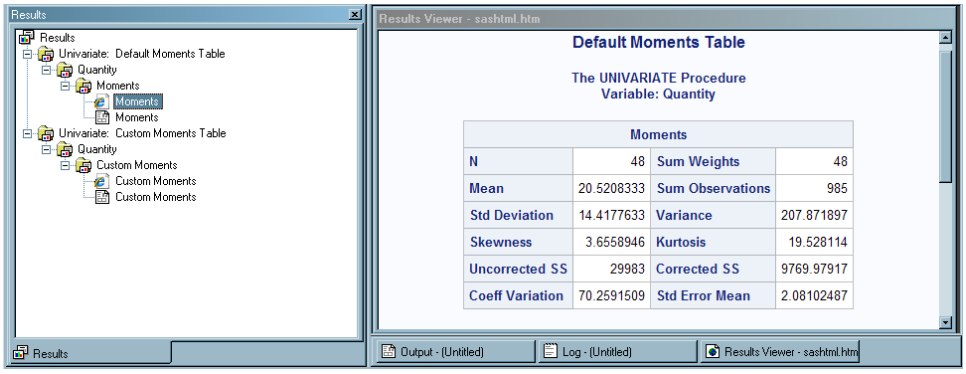
Modify a Table Template That a SAS Procedure Uses

The following output shows the use of a customized table template for the Moments output object using PROC UNIVARIATE. The program that creates the modified table template does the following:

- Creates and edits a copy of the default table template.
- Edits a header in the table template.
- Sets column attributes to enhance the appearance of the HTML output.

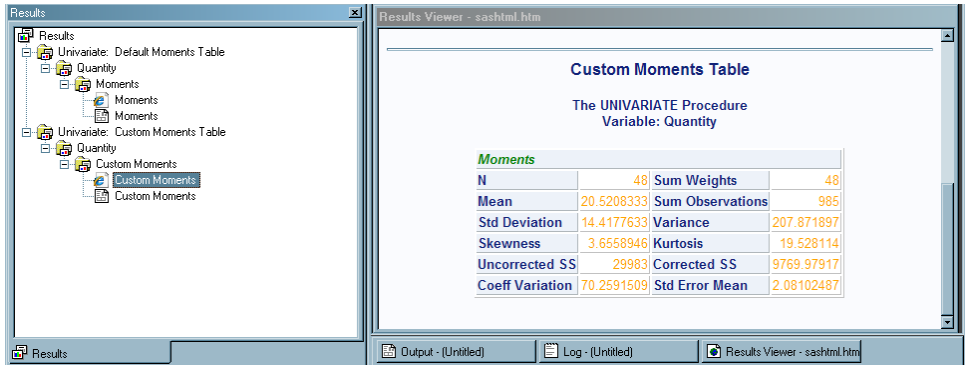
To view the code that creates the following output, see "Editing a Table Template That a SAS Procedure Uses" in *SAS Output Delivery System: Procedures Guide*.

Output 7.1 *Default Moments Table*



Moments			
N	48	Sum Weights	48
Mean	20.5208333	Sum Observations	985
Std Deviation	14.4177633	Variance	207.871897
Skewness	3.6558946	Kurtosis	19.528114
Uncorrected SS	29983	Corrected SS	9769.97917
Coeff Variation	70.2591509	Std Error Mean	2.08102487

Output 7.2 *Customized Moments Table (Customized HTML Output from PROC UNIVARIATE and Viewed with Microsoft Internet Explorer)*



Modify a Style

When you are working with styles, you are more likely to modify a style that SAS provides than create a completely new style. The following output uses the Styles.HTMLBlue template that SAS provides and includes changes made to the style to customize the output's appearance. To view the code that creates this output, see "Modifying the Default Style with the CLASS Statement" in *SAS Output Delivery System: Procedures Guide*.

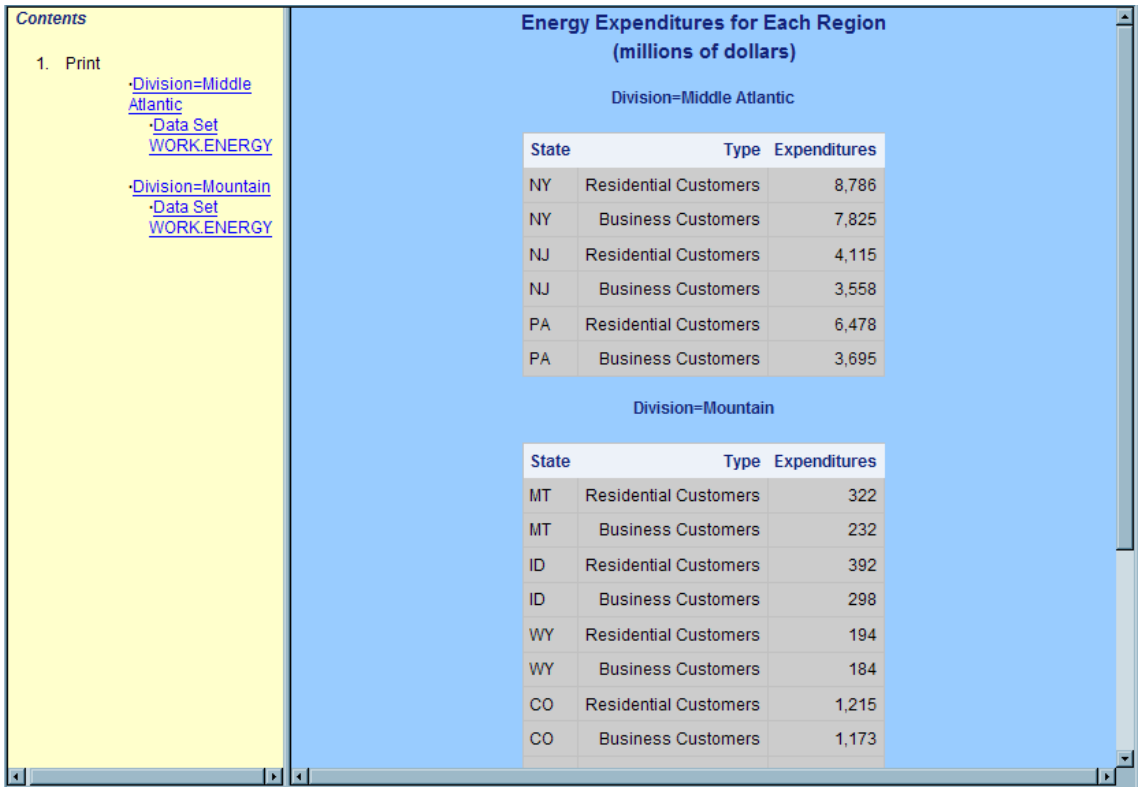
In the contents file, changes to the style are to the following:

- The text of the header and the text that identifies the procedure that produced the output.
- The colors for some of the text.
- The font size for some of the text.
- The spacing in the entries in the table of contents.

In the body file, changes to the style are to the following:

- Two of the colors in the color list. One of these colors is used as the foreground color for the table of contents, the BY line, and column headings. The other color is used as the foreground color for many parts of the body file, including SAS titles and footnotes.
- The font size for titles and footnotes.
- The font style for headers.
- The presentation of the data in the table by changing attributes like cell spacing, rules, and border width.

Figure 7.1 *HTML Output (Viewed with Microsoft Internet Explorer)*



State	Type	Expenditures
NY	Residential Customers	8,786
NY	Business Customers	7,825
NJ	Residential Customers	4,115
NJ	Business Customers	3,558
PA	Residential Customers	6,478
PA	Business Customers	3,695

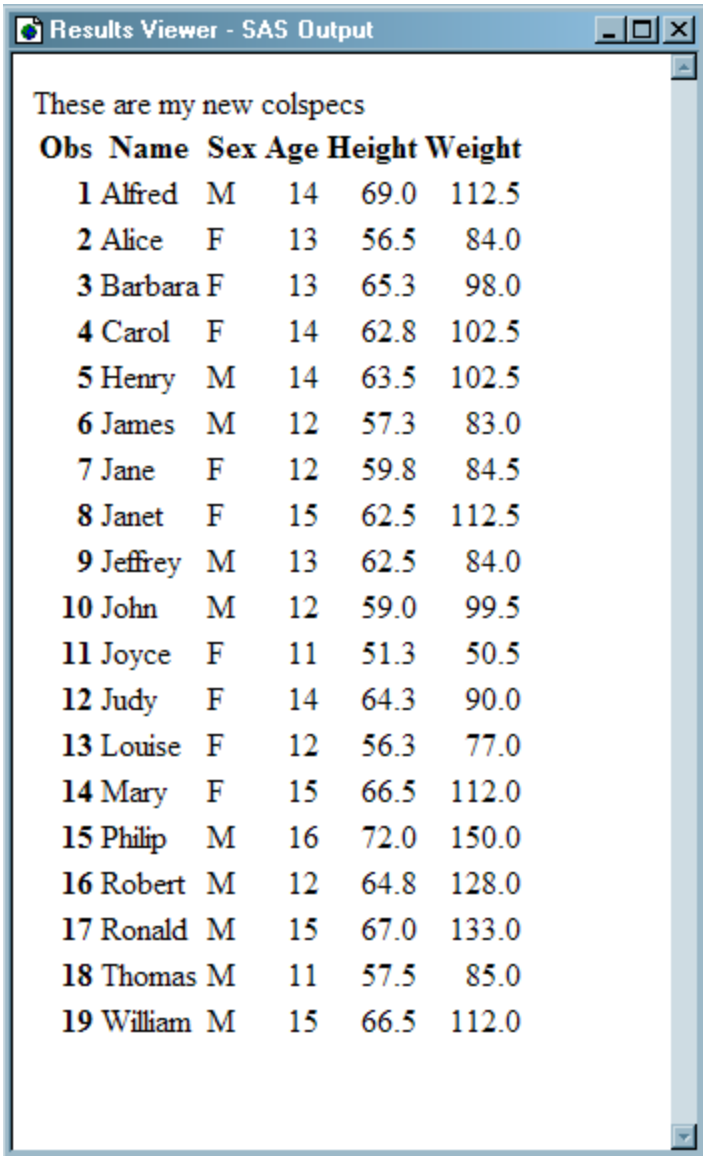
State	Type	Expenditures
MT	Residential Customers	322
MT	Business Customers	232
ID	Residential Customers	392
ID	Business Customers	298
WY	Residential Customers	194
WY	Business Customers	184
CO	Residential Customers	1,215
CO	Business Customers	1,173

Create Your Own Tagset

Tagsets are used to create custom markup output. You can create your own tagsets, extend existing tagsets, or modify a tagset that SAS provides. The following display shows the results from a new tagset named TAGSET.MYTAGS.

To see the customized MYTAGS.CHTML tagset, view the source by selecting **View** ⇨ **Source** from your web browser's toolbar.

Figure 7.2 *MYTAGS.CHTML Output (Viewed with Microsoft Internet Explorer)*



Results Viewer - SAS Output

These are my new colspecs

Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5
5	Henry	M	14	63.5	102.5
6	James	M	12	57.3	83.0
7	Jane	F	12	59.8	84.5
8	Janet	F	15	62.5	112.5
9	Jeffrey	M	13	62.5	84.0
10	John	M	12	59.0	99.5
11	Joyce	F	11	51.3	50.5
12	Judy	F	14	64.3	90.0
13	Louise	F	12	56.3	77.0
14	Mary	F	15	66.5	112.0
15	Philip	M	16	72.0	150.0
16	Robert	M	12	64.8	128.0
17	Ronald	M	15	67.0	133.0
18	Thomas	M	11	57.5	85.0
19	William	M	15	66.5	112.0

Create a Template-Based Graph

StatGraph templates create output called ODS Graphics. For complete information, see *SAS Graph Template Language: User's Guide*.

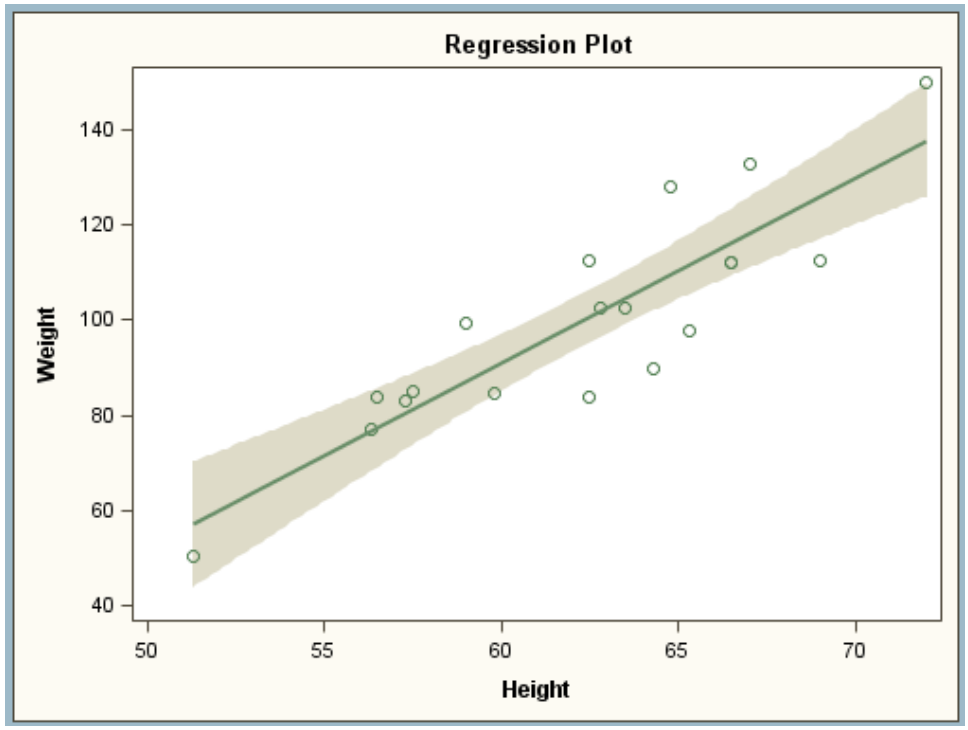
The following code creates the StatGraph template MyGraphs.Regplot, which creates the following graph:

```
proc template;
define statgraph mygraphs.regplot;
begingraph;
  entrytitle "Regression Plot";
  layout overlay;
    modelband "mean";
    scatterplot x=height y=weight;
```

```
    regressionplot x=height y=weight / clm="mean";
  endlayout;
endgraph;
end;
run;

ods listing style=analysis;
ods graphics / reset imagename="reg" width=500px;

proc sgrender data=sashelp.class template=mygraphs.regplot;
run;
```

The following display shows a scatterplot with an overlaid regression line and confidence limits for the mean of the HEIGHT and WEIGHT variables of a data set.

Figure 7.3 *Scatterplot Created with a StatGraph Template*



Modify a Crosstabulation Table

The TEMPLATE procedure enables you to customize the appearance of crosstabulation (contingency) tables that are created with the FREQ procedure. By default, crosstabulation tables are formatted based on the CrossTabFreqs template that SAS provides. However, you can create a customized CrossTabFreqs template using the TEMPLATE procedure with the DEFINE CROSSTABS statement. To view the SAS code that creates this output, see "Creating a Crosstabulation Table Template with a Customized Legend" in *SAS Output Delivery System: Procedures Guide*.

The following output shows the use of a customized CrossTabFreqs template for the CrossTabFreqs table. The program that creates the customized CrossTabFreqs template does the following:

- Modifies table regions.
- Customizes legend text.

- Modifies headers and footers.
- Modifies variable labels used in headers.
- Customizes styles for cellvalues.

Figure 7.4 *Customized CrossTabFreqs Template for the CrossTabFreqs Table*

City Government Form by Number of Meetings Scheduled

The FREQ Procedure

Frequency Percent Row Percent Column Percent	Number of Meetings Scheduled						
City Government Form	?	Not Known	100 or Less	101-200	201-300	Over 300	Total
?	0 . . .	0 . . .	0 . . .	1 . . .	0 . . .	0 . . .	. .
Not Applicable	0 . . .	10 . . .	0 . . .	0 . . .	0 . . .	0 . . .	. .
Council Manager	0 . . .	0 . . .	47 12.30 22.27 55.95	63 16.49 29.86 58.88	49 12.83 23.22 62.03	52 13.61 24.64 46.43	211 55.24
Commission	0 . . .	0 . . .	6 1.57 28.57 7.14	7 1.83 33.33 6.54	3 0.79 14.29 3.80	5 1.31 23.81 4.46	21 5.50
Mayor Council	1 . . .	0 . . .	31 8.12 20.67 36.90	37 9.69 24.67 34.58	27 7.07 18.00 34.18	55 14.40 36.67 49.11	150 39.27
Total	. .	. .	84 21.99	107 28.01	79 20.68	112 29.32	382 100.00

City Government Form by Number of Meetings Scheduled

Frequency Missing = 12

Working with ODS Graphics

Graphics are an indispensable part of statistical analysis. Graphics reveal patterns, identify differences, and provoke meaningful questions about your data. Graphics add clarity to an analytical presentation and stimulate deeper investigation.

SAS 9.2 introduced the Graph Template Language (GTL), a powerful new language for defining clear and effective statistical graphics. The GTL enables you to generate various types of plots, such as model fit plots, distribution plots, comparative plots, prediction plots, and more.

The GTL applies accepted principles of graphics design to produce plots that are clean and uncluttered. Colors, fonts, and relative sizes of graph elements are designed for optimal impact. By default, the GTL produces PNG files, which support true color (the full 24-bit RGB color model) and enable visual effects such as anti-aliasing and transparency. A PNG file retains a small file size. GTL statement options enable you to control the content and appearance of the graphic down to the smallest detail.

The GTL is designed to produce graphics with minimal syntax. The GTL uses a flexible, building-block approach to create a graph by combining statements in a StatGraph template. StatGraph templates are defined with the TEMPLATE procedure.

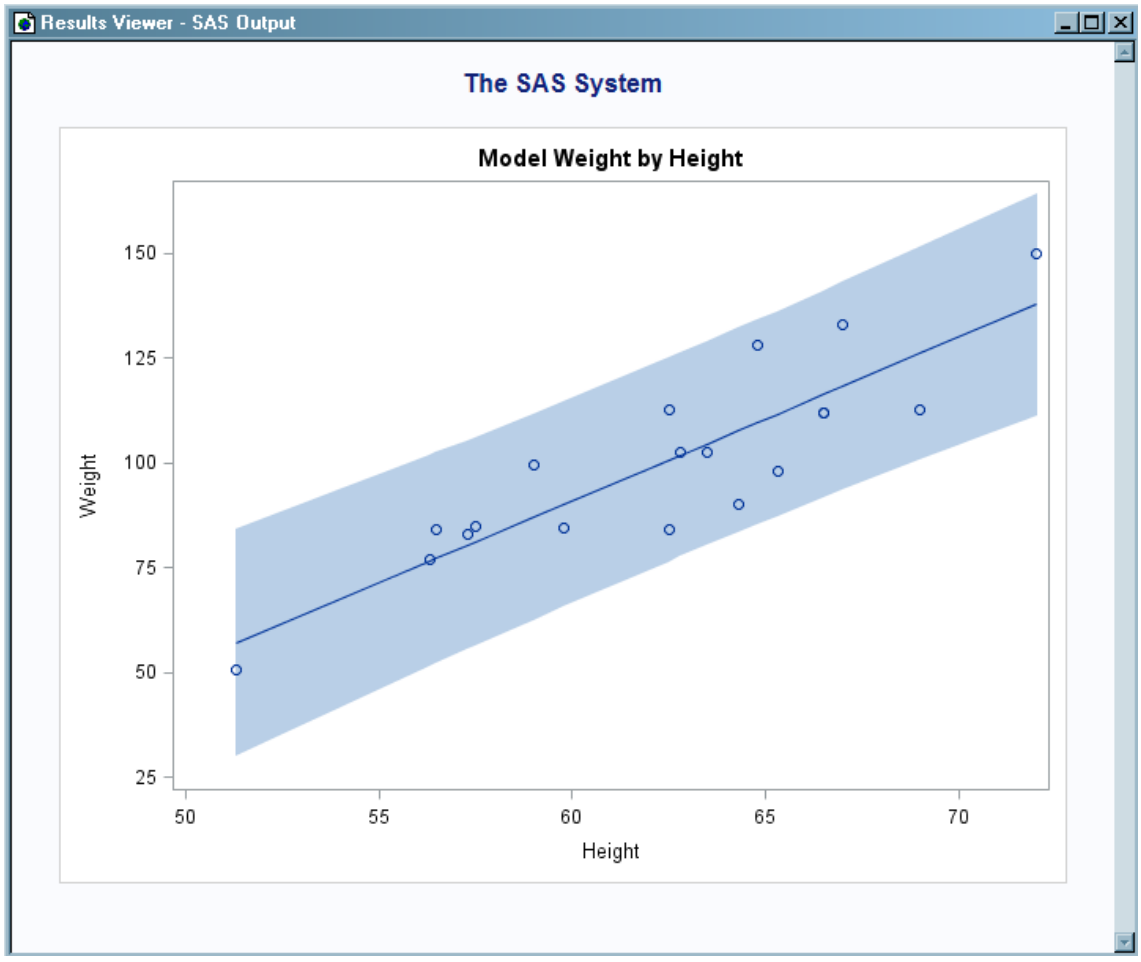
You can create custom graphs by defining your own StatGraph templates. To create a custom graph, you must perform the following steps:

1. Define a StatGraph template with the TEMPLATE procedure.
2. Use the GTL to specify the parameters of your graph.
3. Associate your data with the template using the SGRENDER procedure.

With just a few statements, you can create the graphs that you need to analyze your data. For example, you can create the following model fit plot with these statements:

```
proc template;
define statgraph mytemplate;
beginGraph;
   entrytitle "Model Weight by Height";
   layout overlay;
   bandplot x=height limitupper=upper limitlower=lower;
   scatterplot y=weight x=height;
   seriesplot y=predict x=height;
endlayout;
endGraph;
end;
run;

proc sgrender data=sashelp.classfit
              template=mytemplate;
run;
```

Figure 7.5 *Model Fit Plot Using MyTemplate and Sashelp.Classfit*



The previous example defines a StatGraph template named MyTemplate, which uses values from the data set Sashelp.Classfit. This data set contains the data variables HEIGHT and WEIGHT and precomputed values for the fitted model (**`Y=PREDICT`**) and confidence band (**`limitupper=upper limitlower=lower`**). The SGRENDER procedure uses the data in Sashelp.Classfit and the MyTemplate template to render the graph. (This example is member GTLMFIT1 in the SAS Sample Library.)

The following two graphics are more examples of what you can do with ODS graphics:

Figure 7.6 *PROC SGSCATTER (SAS) with LISTING Style*

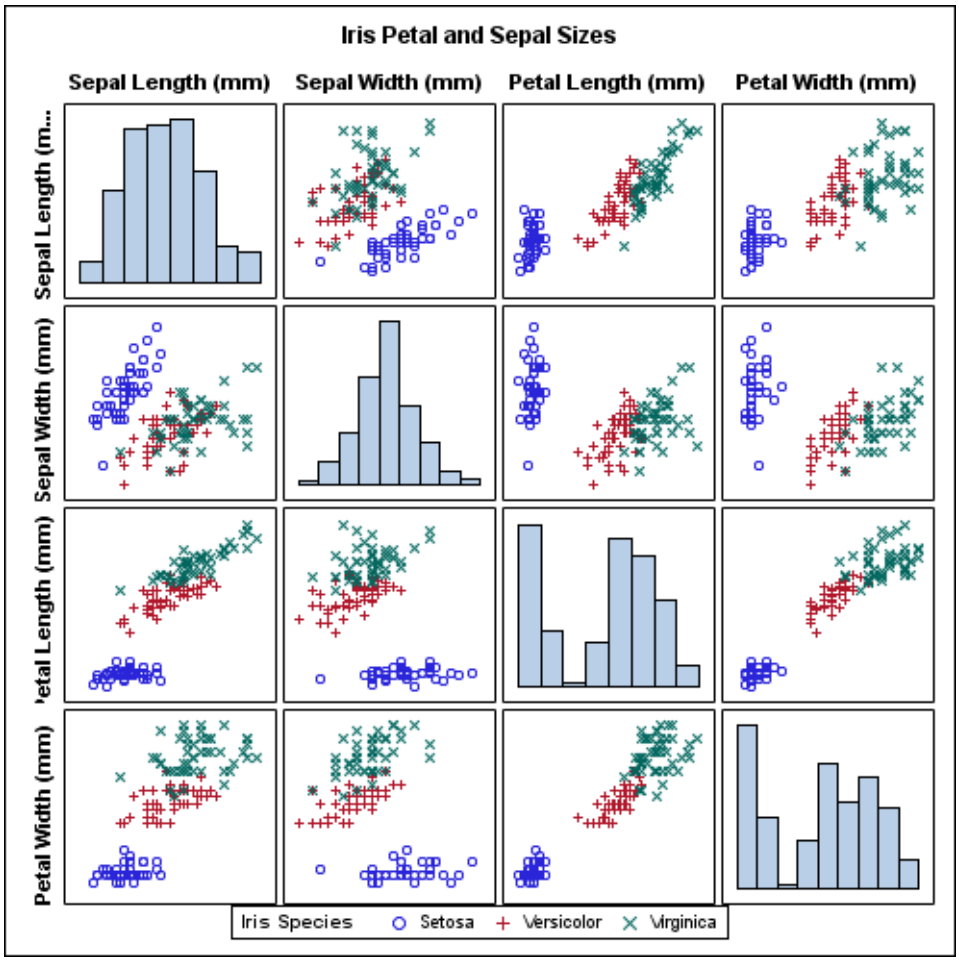
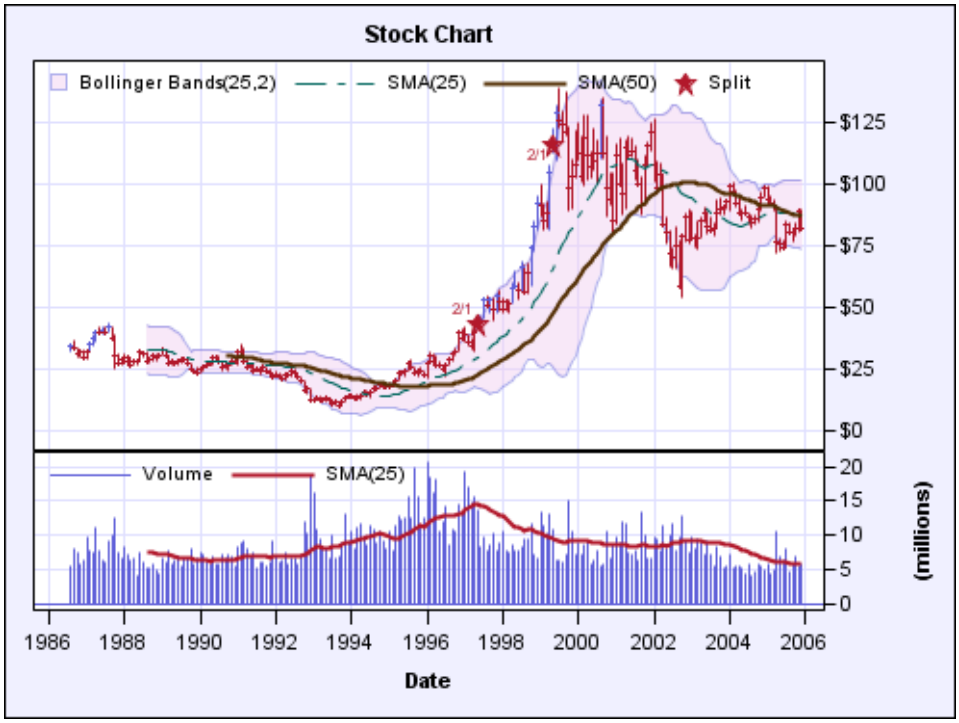


Figure 7.7 *Custom Template Rendered with PROC SGRENDER (SAS) and a Custom Style*



Advanced Features of the DOCUMENT Procedure

Overview

In Chapter 3, "Creating an ODS Document," on page 23, you learned how to manipulate output objects using the Documents window. You can use PROC DOCUMENT statements to accomplish all of the tasks that you completed using the Documents window and much more. For complete documentation on the DOCUMENT procedure, see "The DOCUMENT Procedure" in *SAS Output Delivery System: Procedures Guide*.

The combination of the ODS DOCUMENT statement and the DOCUMENT procedure enables you to store a report's individual components. You can then modify and replay the report. The ODS DOCUMENT statement stores the actual ODS objects that are created when you run a report. You can use the DOCUMENT procedure to rearrange, duplicate, or remove output from the results of a procedure or a database query without invoking the procedure or database query from the original report. You can use the DOCUMENT procedure to do the following:

- Transform a report without rerunning an analysis or repeating a database query.
- Modify the structure of the output.
- Display output to any ODS output format.
- Navigate the current directory and list entries.
- Open and list ODS documents.
- Manage output.

With the DOCUMENT procedure, you are not limited to regenerating the same report. You can change the order in which objects are rendered, the table of contents, the templates that are used, macro variables, and ODS and system options.

Working with the DOCUMENT Procedure

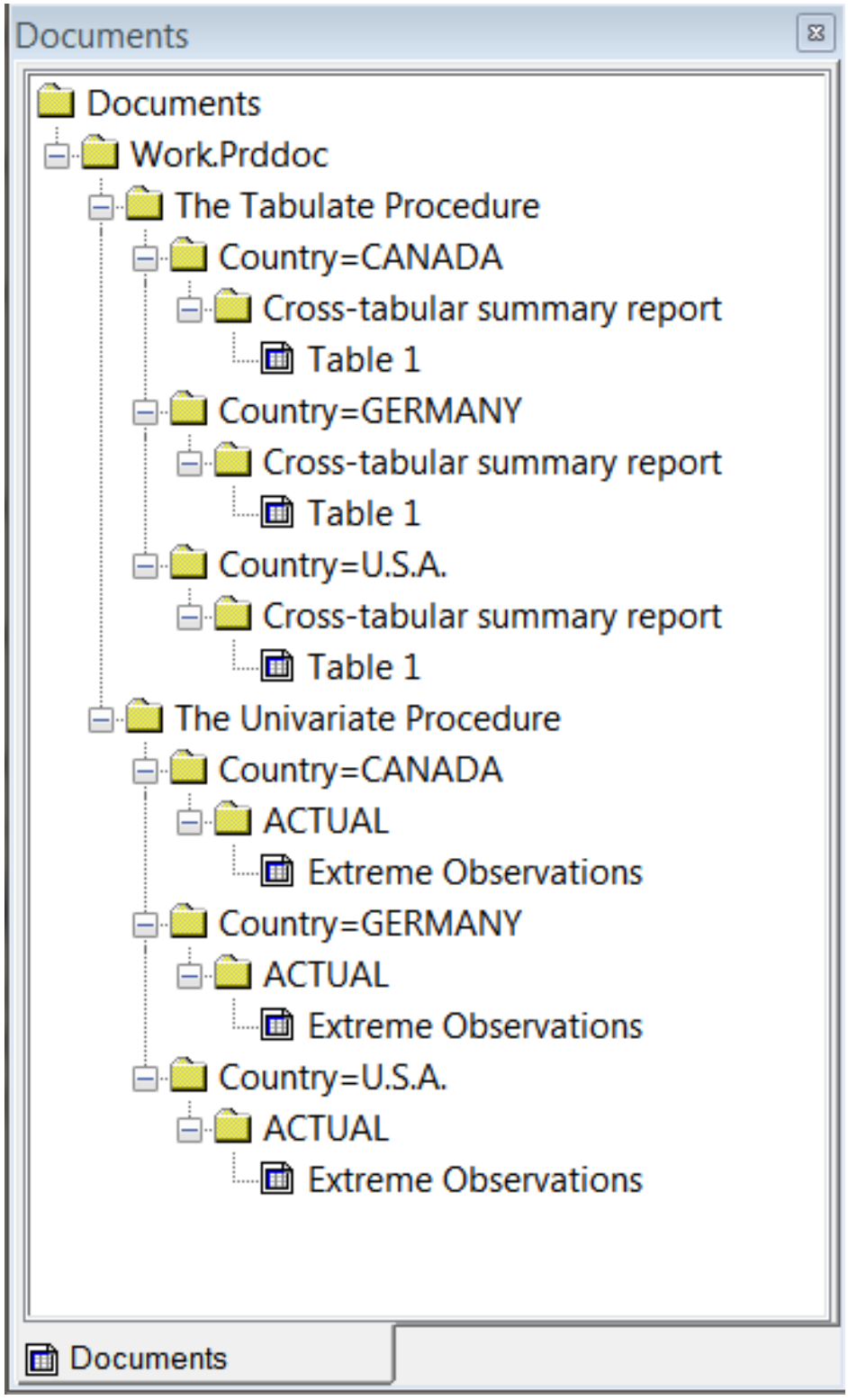
To create an ODS document, you can use the Documents window or the "ODS DOCUMENT Statement" in *SAS Output Delivery System: Procedures Guide*. The following code creates the ODS document Work.Prddoc within a document store:

```
ods listing close;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;

ods document name=work.prddoc(write);

proc tabulate data=prdsale;
    by Country;
    var predict;
    class prodtype;
    table prodtype all,
    predict*(min mean max);
run;

ods select ExtremeObs;

proc univariate data=prdsale;
    by Country;
    var actual;
run;

ods document close;
```

The following display shows the ODS document Work.Prddoc and its contents. To view the Documents window, submit the following command on the command line:

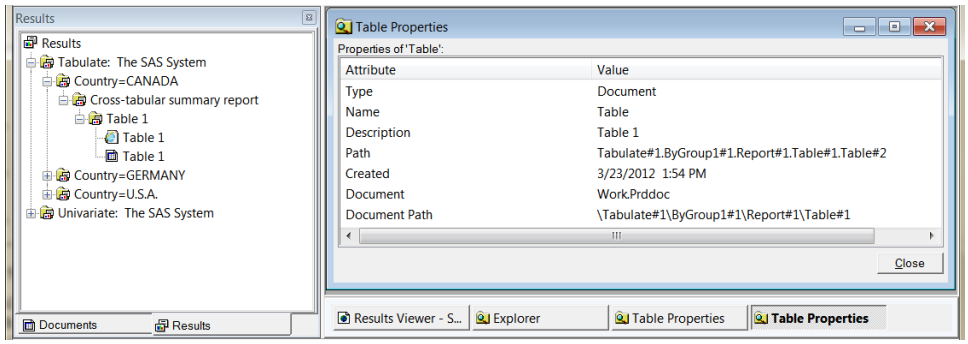
```
odsdocuments
```

Figure 7.8 *SAS Documents Window Showing Work.Prddoc Document and Documents Icon*



The following display shows the properties of Table 1. You can see the document name and the document path, as well as other information.

Figure 7.9 *Table Properties for Table 1*



An ODS document store is not a SAS data set, as you can see by the Document icon in the previous display. The Work.Prddoc document was written to the Work library. If it had been written to a permanent location (such as **`c:\temp\output`**), in Windows Explorer, the document store would have a file extension of SAS7BITM.

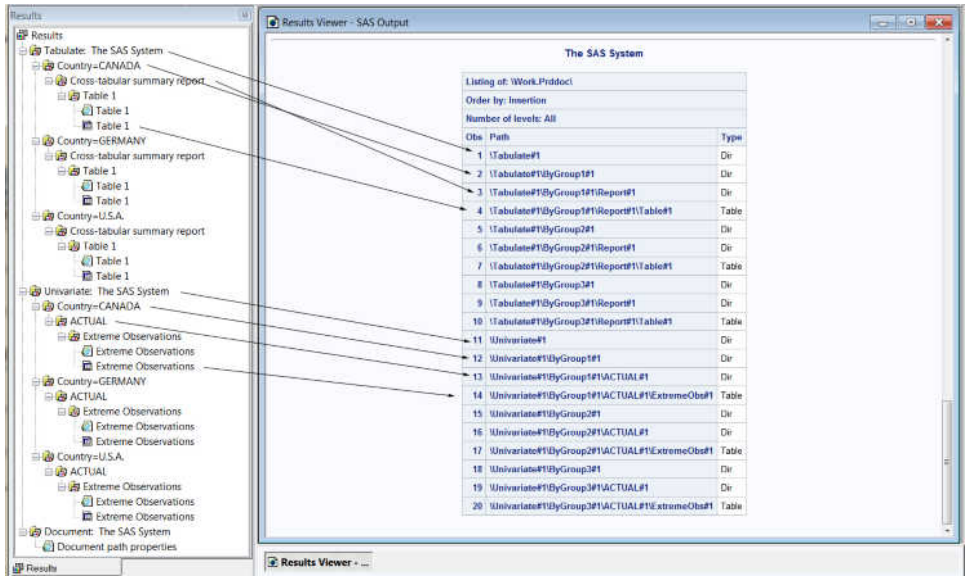
After you have created a document with the ODS DOCUMENT statement, you can use the LIST statement in a PROC DOCUMENT step to view the contents of your document. The LIST statement enables you to look at the object list and folder structure in the ODS document. The following code creates a list of all levels of the Work.Prddoc document:

```
proc document name=work.prddoc;
    list / levels=all;
run;
quit;
```

The LIST statement can list what is in an entire document or in just one of the entries. For more information about the LIST statement, see "The DOCUMENT Procedure" in *SAS Output Delivery System: Procedures Guide*.

In the following figure, every folder icon in the Results window corresponds to an item with a type of Dir in the LIST statement output. Every table created by a procedure corresponds to an item with a type of Table in the LIST statement output.

Figure 7.10 *PROC DOCUMENT List Statement Output Compared to Results Window*



ODS and the DATA Step

If you are writing DATA step reports, you are already using ODS. DATA step output is routed through ODS by default. For more than 20 years, SAS users have been able to create highly customized reports as simple LISTING output, which uses a monospace type font. With the advent of ODS, SAS users have a broad range of choices for printing customized DATA step reports.

- You can produce DATA step reports in many different formats, such as HTML, RTF, EPUB, PowerPoint, or PDF.
- You can create the report in multiple formats at the same time.
- You can produce the report in different formats at a later time without rerunning the DATA step.

To take advantage of these enhanced reporting capabilities, you can combine DATA step programming with the formatting capabilities of ODS. For example, to create PDF output, start with the DATA step tools that you are already familiar with.

- the DATA _NULL_ statement
- the FILE statement
- the PUT statement

Then, add a few simple ODS statements and options. You can choose from several ODS formatting statements to format the output in other presentation styles, such as HTML, RTF, EPUB, PowerPoint, and PDF. For more information about ODS statements, see "Introduction to ODS Language Statements" in *SAS Output Delivery System: User's Guide*.

Here are the basic steps for using ODS with the DATA step to produce reports with enhanced formatting:

Table 7.1 *Steps to Produce Enhanced ODS Output with the DATA Step*

Steps	Tools	Comments
Specify formatting for your output.	ODS formatting statements can specify formats such as listing, HTML, RTF, EPUB, PowerPoint, and PDF.	You can produce output in multiple formats at the same time by specifying more than one format. *Note:* If you want only the default output, then you do not need a destination ODS statement.
Specify structure.	The ODS option in the FILE statement lists the variables and the order in which they appear in the output.	Additional suboptions give you even more control over the structure.
Connect the data to the template.	The FILE PRINT ODS statement creates an output object by binding a data component to a table template (template).	You can specify other details by using ODS suboptions in the FILE PRINT ODS statement.
Output data.	The PUT statement writes variable values to the data component.	A simple way to output all variable values is to use PUT _ODS_.

Use ODS statements to specify how you want ODS to format your output (for example, as HTML, RTF, EPUB, PowerPoint, or PDF). Then, in the DATA step, use the FILE PRINT ODS and PUT statements with appropriate ODS-specific suboptions to produce your report.

The PUT statement writes variable values. The FILE PRINT ODS statement directs the output.[1] You can use ODS to produce the output in multiple formats and to produce output at a later time in different formats without rerunning the DATA step.

You control the formatting that is applied to your reports using ODS formatting statements. They open and close ODS destinations, which apply formatting to the output objects that you create with ODS and the DATA step.

Here is a list of topics and sources for additional information:

Table 7.2 *Where to Find More Information about How to Use ODS in the DATA Step*

Topic	Where to learn more
ODS formatting statements	"Dictionary of ODS Language Statements" in *SAS Output Delivery System: User's Guide*

1 If you do not specify a FILE statement, then the PUT statement writes to the SAS log, by default. If you use multiple PUT and FILE statements, then in addition to creating ODS-enhanced output, you can write to the SAS log, to the regular DATA step output buffer, or to another external file in the same DATA step.

Topic	Where to learn more
ODS destinations	"Understanding ODS Destinations" in *SAS Output Delivery System: User's Guide*
How ODS works	"Overview of How ODS Works" in *SAS Output Delivery System: Procedures Guide*

Working with Cascading Style Sheets (CSS)

Cascading style sheets (CSS) is a style sheet language that you can use with ODS to control the look and formatting of ODS output. A cascading style sheet is an external file that contains label-value pairs that describe the visual aspects of output, such as fonts, colors, borders, and so on. You can then apply the CSS to your ODS output. The CSS language that you can use with ODS is based on the standard CSS syntax found on the Internet at http://www.w3.org/Style/CSS/. However, with ODS, you can apply CSS to many different types of output, such as PDF, RTF, and PowerPoint. For complete documentation about using CSS with ODS, see the chapter about CSS in the *SAS Output Delivery System: Advanced Topics*.

You can use CSS as an alternative to PROC TEMPLATE styles. Unlike PROC TEMPLATE styles, CSS enables you to apply custom styles based on column names, BY group names, BY group variables, and anchor names. Using CSS with ODS gives you greater flexibility in customizing your ODS reports.

Using CSS with ODS enables you to use the power of CSS selectors with the internal data querying capabilities of PROC TEMPLATE styles. You can create custom output by defining your own external CSS file to apply to your ODS output. To use CSS with ODS, it is recommended that you perform the following steps:

1. Look at the ODS Document Object Model (ODS DOM) to determine what elements and attributes exist so that you can construct your CSS selectors to address them. You can view the ODS DOM with the **ODS TRACE DOM;** statement or with the DOM option in any ODS destination statement except ODS LISTING and ODS OUTPUT.
2. Create an external CSS file that consists of standard CSS syntax.
3. Apply the CSS file to your ODS output with the CSSSTYLE= option in any ODS destination statement except ODS LISTING, ODS OUTPUT, or ODS POWERPOINT. The style sheet is interpreted by SAS instead of by the web browser.

   In the following example, the style sheet MyCss.css is applied to PDF, RTF, and HTML output.

   ```
   ods pdf cssstyle='MyCss.css';
   ods rtf cssstyle='MyCss.css';
   ods html cssstyle='MyCss.css';
   proc print data=sashelp.class(obs=3);
   run;
   ods _all_ close;
   ```

The following CSS is an external file named ColorMapCss.css:

```
.body {
```

```
   background-color: lightblue;
}

.oo {
   background-color: pink;
}
.systemtitle {
   background-color: darkblue;
   color: white;
}
.table {
   background-color: white;
}
.header {
   background-color: gold;
}
.rowheader {
   background-color: purple;
   color: white;
}
.data {
   background-color: linen;
}
```

The following SAS code applies the CSS file ColorMapCss.css to your PDF output:

```
options nodate nonumber obs=25;
ods html close;
ods pdf nobookmarklist  nobookmarkgen file="myoutput.pdf"
        cssstyle="your-file-path\ColorMapCss.css";
title "Comparison of City and Highway Miles Per Gallon";

proc print data=sashelp.cars;
   var make model type Origin MPG_City MPG_Highway;
   run;

ods pdf close;
```

Output 7.3 *PDF Output with CSS Applied*

Comparison of City and Highway Miles Per Gallon

Obs	Make	Model	Type	Origin	MPG City	MPG Highway
1	Acura	MDX	SUV	Asia	17	23
2	Acura	RSX Type S 2dr	Sedan	Asia	24	31
3	Acura	TSX 4dr	Sedan	Asia	22	29
4	Acura	TL 4dr	Sedan	Asia	20	28
5	Acura	3.5 RL 4dr	Sedan	Asia	18	24
6	Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	18	24
7	Acura	NSX coupe 2dr manual S	Sports	Asia	17	24
8	Audi	A4 1.8T 4dr	Sedan	Europe	22	31
9	Audi	A41.8T convertible 2dr	Sedan	Europe	23	30
10	Audi	A4 3.0 4dr	Sedan	Europe	20	28
11	Audi	A4 3.0 Quattro 4dr manual	Sedan	Europe	17	26
12	Audi	A4 3.0 Quattro 4dr auto	Sedan	Europe	18	25
13	Audi	A6 3.0 4dr	Sedan	Europe	20	27
14	Audi	A6 3.0 Quattro 4dr	Sedan	Europe	18	25
15	Audi	A4 3.0 convertible 2dr	Sedan	Europe	20	27
16	Audi	A4 3.0 Quattro convertible 2dr	Sedan	Europe	18	25
17	Audi	A6 2.7 Turbo Quattro 4dr	Sedan	Europe	18	25
18	Audi	A6 4.2 Quattro 4dr	Sedan	Europe	17	24
19	Audi	A8 L Quattro 4dr	Sedan	Europe	17	24
20	Audi	S4 Quattro 4dr	Sedan	Europe	14	20
21	Audi	RS 6 4dr	Sports	Europe	15	22
22	Audi	TT 1.8 convertible 2dr (coupe)	Sports	Europe	20	28
23	Audi	TT 1.8 Quattro 2dr (convertible)	Sports	Europe	20	28
24	Audi	TT 3.2 coupe 2dr (convertible)	Sports	Europe	21	29
25	Audi	A6 3.0 Avant Quattro	Wagon	Europe	18	25

Working with the ODSTEXT Procedure

The ODSTEXT procedure is used to create text block templates. These text block templates create lists and paragraphs for your output. You can use style attributes and formats to customize your content and WHERE expressions to select your content. With PROC ODSTEXT, you can use the DATA= option to bind your data to the template without using a DATA step. PROC ODSTEXT can be used with any output destination. However, PROC ODSTEXT is essential for creating content for the ODS destination for PowerPoint and e-books. Using the ODSTEXT procedure is an advanced technique.

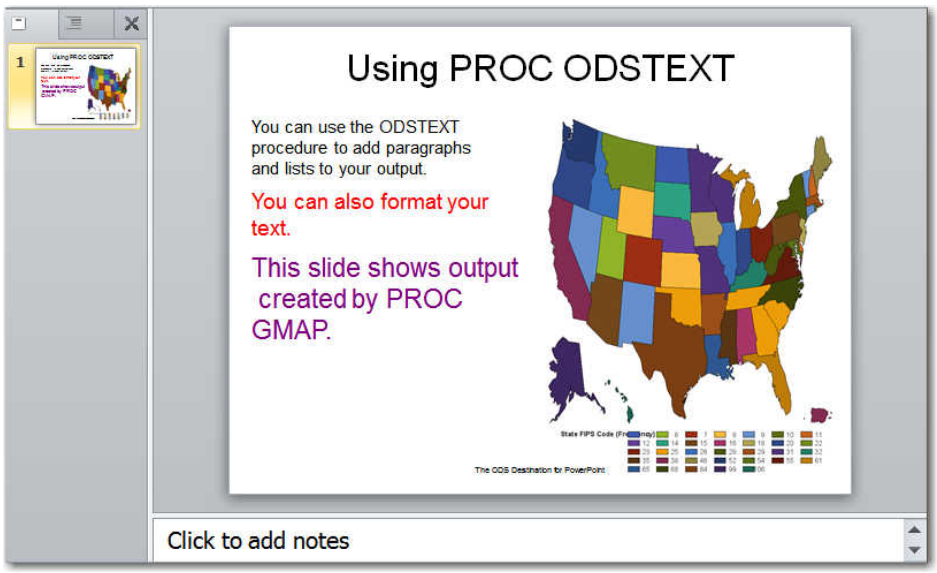
See "ODS EPUB Statement" in *SAS Output Delivery System: User's Guide* and "ODS POWERPOINT Statement" in *SAS Output Delivery System: User's Guide* for more information about creating output for the ODS destination for PowerPoint and e-book output. For complete documentation about the ODSTEXT procedure, see "The ODSTEXT Procedure" in *SAS Output Delivery System: Procedures Guide*.

The following example creates text output to add to a PowerPoint slide:

```
ods html close;
options nodate;
title 'Using PROC ODSTEXT';
footnote 'The ODS Destination for PowerPoint';

ods powerpoint file="layoutTwocontent.ppt" layout=twocontent;

proc odstext;
   p 'You can use the ODSTEXT procedure to add paragraphs
      and lists to your output.';
   p 'You can also format your text.' / style=[color=red fontsize=25pt];
   p 'This slide shows output created by PROC GMAP.'
                                    / style=[color=purple fontsize=30pt];
```

```
run;

proc gmap map=maps.us data=maps.us all;
   id state;
   choro state/statistic=frequency discrete;
run;
quit;

ods _all_ close;
```

Output 7.4 *Adding Text to Your Output*



Working with the ODSLIST Procedure

The ODSLIST procedure is used to create bulleted list templates. With PROC ODSLIST, you can create text templates for lists that can be customized and nested an infinite number of times. You can use style attributes and formats to customize your content and WHERE expressions to specify list item content. With PROC ODSLIST, you can use the DATA= option to bind your data to the template without using a DATA step.

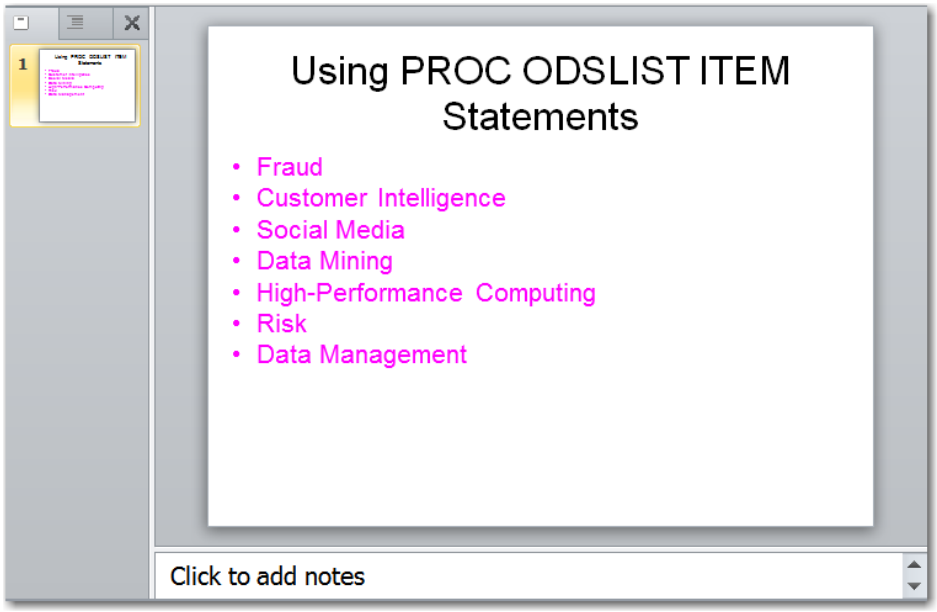
PROC ODSLIST can be used with any output destination. However, PROC ODSLIST is essential for creating content for the ODS destination for PowerPoint and e-books. See "ODS EPUB Statement" in *SAS Output Delivery System: User's Guide* and "ODS POWERPOINT Statement" in *SAS Output Delivery System: User's Guide* for more information about creating output for the ODS destination for PowerPoint and e-book output. For complete documentation about the ODSLIST procedure, see "The ODSLIST Procedure" in *SAS Output Delivery System: Procedures Guide*.

The following program creates a list for a PowerPoint slide. The CELLSTYLE AS statement formats the contents of each item.

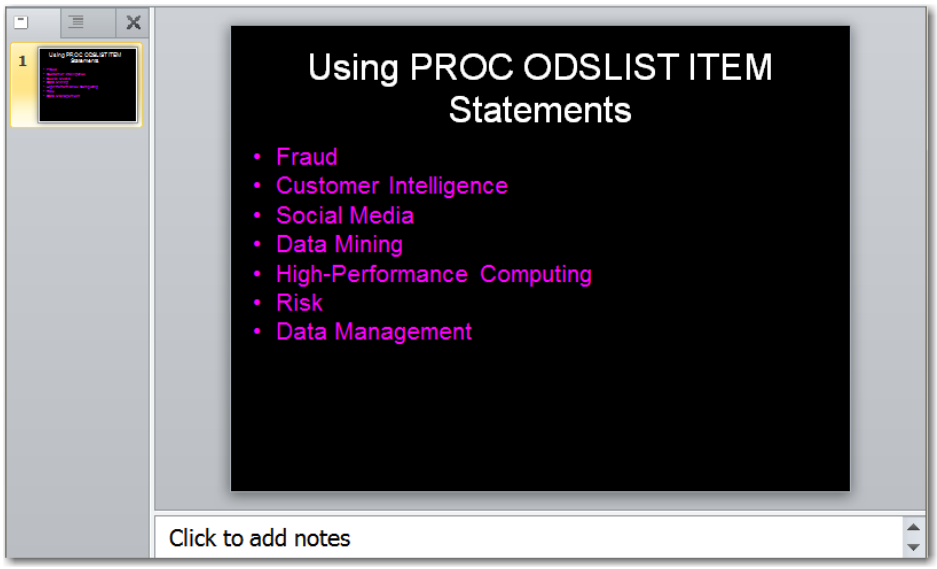
```
ods html close;
options nodate;
title 'Using PROC ODSLIST ITEM Statements';
```

```
ods powerpoint file="DefaultStyle.ppt";
ods powerpoint(2) file="PowerpointdarkStyle.ppt" style=PowerPointDark;

proc odslist name=Slides store=sasuser.Myexampleslides print;

cellstyle 1 as {fontsize=1cm color=purple fontweight=bold};
   item 'Fraud';
   item 'Customer Intelligence';
   item 'Social Media';
   item 'Data Mining';
   item 'High-Performance Computing';
   item 'Risk';
   item 'Data Management';
run;

ods _all_ close;
```

Output 7.5 *Slide with Default Style*



Output 7.6 *Slide with PowerPointDark Style*



Working with the ODSTABLE Procedure

By default, ODS output is formatted based on the various definitions or templates that the procedure or DATA step specify. However, you can create your own new tabular output templates by using the ODSTABLE procedure. ODS uses these templates to produce customized tabular output. With the ODSTABLE procedure, you can create table templates and bind them with the input data set in one statement. You can name your templates and store them in a template store.

The ODSTABLE procedure is a simpler way of producing the same output that you would expect to get from using the DEFINE TABLE statement in PROC TEMPLATE. You can use the same statements in an ODSTABLE block as you would use in a DEFINE TABLE block in PROC TEMPLATE. For complete documentation about the ODSTABLE procedure, see "The ODSTABLE Procedure" in *SAS Output Delivery System: Procedures Guide*. For complete documentation about customizing tables with the TEMPLATE procedure, see "TEMPLATE Procedure: Creating Tabular Output" in *SAS Output Delivery System: Procedures Guide*.

This example creates a table with customized columns.

```
options nodate obs=15;
title "Customizing Columns";

proc odstable data=sashelp.class;
column age sex height weight;

define age;
  style={fontsize=10pt just=l borderrightstyle=dashed background=yellow};
  header='Age of Student';
  format=3.;
end;
define sex;
  style={fontsize=10pt just=l borderrightstyle=dashed};
  header='Gender';
```

```
end;
define height;
  style={fontsize=10pt  just=l foreground=blue borderrightstyle=dashed};
  header='Height';
end;

run;
```

The following output uses the default HTMLBlue style with customized columns:

Output 7.7 *Customizing Columns*

Customizing Columns

Age of Student	Gender	Height	Weight
14	M	69	112.5
13	F	56.5	84
13	F	65.3	98
14	F	62.8	102.5
14	M	63.5	102.5
12	M	57.3	83
12	F	59.8	84.5
15	F	62.5	112.5
13	M	62.5	84
12	M	59	99.5
11	F	51.3	50.5
14	F	64.3	90
12	F	56.3	77
15	F	66.5	112
16	M	72	150

Working with the Report Writing Interface

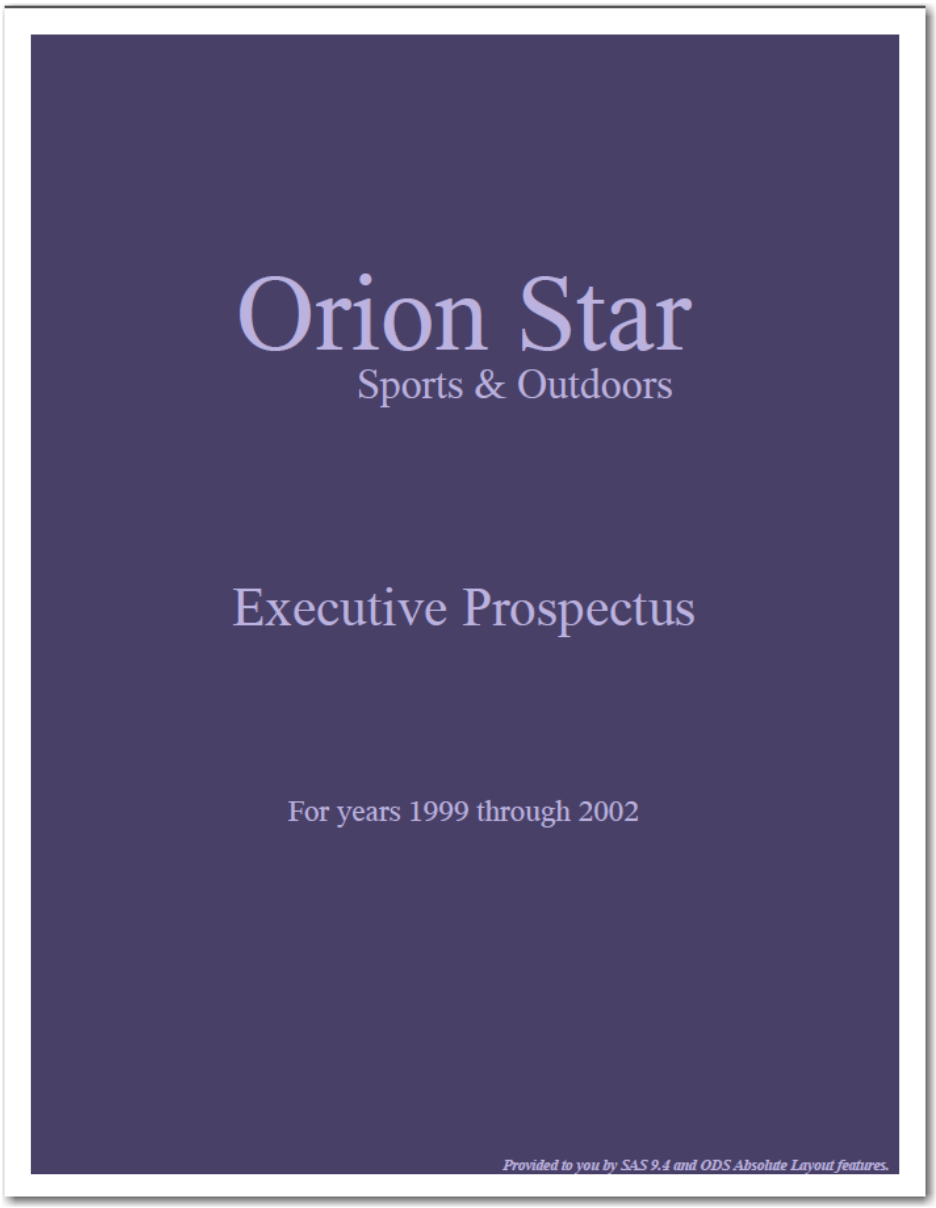
The traditional method for creating a custom report without creating a data set is called DATA _NULL_ report writing. With the ODS Report Writing Interface (RWI), you can create highly customized reports in an object-oriented language that is fully integrated with ODS. The ODS RWI does the following:

- fully embraces ODS features such as proportional fonts, trafficlighting, colors, images, and Unicode characters, while at the same time providing pixel-perfect placement capabilities
- takes advantage of the programming features that the DATA step offers, such as conditional logic, formatting capabilities, BY-group processing, and arrays
- uses an object-oriented language that provides you with flexibility and control so that even the most rigid reporting requirements can be met easily.

The following example uses the RWI to create a cover letter:

```
ods html close;
options nodate nonumber;
title;

ods escapechar="~";
footnote "~{style [font_size=10pt just=right color=cxbbb2e0]Provided to you by
           SAS 9.4 and ODS Absolute Layout features.}";

proc template;
   define style Styles.Orionbackground;
   parent=Styles.Printer;
   style body /
   background=cx494068;
   end;
run;
ods pdf file="ExecutiveSummary.pdf" style=Styles.Orionbackground notoc;

data _null_ ;
dcl odsout trt();

trt.layout_absolute();

   trt.region(y: "2in");
      trt.format_text(data: "Orion Star", just: "c",
                      style_attr: "color=cxbbb2e0 just=center font_size=72pt");

   trt.region(y: "3in", x: "3in");
      trt.format_text(data: "Sports & Outdoors",
                      style_attr:"color=cxbbb2e0 font_size=28pt");

   trt.region(y: "5in");
      trt.format_text(data: "Executive Prospectus",
                      just: "c",
                      style_attr:"font_size=36pt color=cxbbb2e0");

   trt.region(y: "7in");
      trt.format_text(data: "For years 1999 through 2002",
                      just: "c",
                      style_attr:"font_size=20pt color=cxbbb2e0");

trt.layout_end();
run;

ods pdf close;
```

Output 7.8 *Cover Page Created with the RWI*



The RWI enables you to create and manipulate predefined ODS objects in a DATA step. ODS objects are data elements that consist of attributes, methods, and operators. Attributes are the properties that specify the information that is associated with an object. Methods define the operations that an object can perform. You use the DATA step object dot syntax to access the component object's attributes and methods. You can use the RWI's object-oriented functionality to create highly specialized reports. The RWI is available for PRINTER and HTML destinations. For complete documentation about the RWI, see "Introduction to the Report Writing Interface" in *SAS Output Delivery System: Advanced Topics*.

Working with ODS LAYOUT

ODS LAYOUT statements enable you to create custom reports that easily mix SAS graphics, images, text, and tables, and then arrange them on a page where you like.

- The ODS LAYOUT GRIDDED statement enables you to format in a gridded layout. The ODS LAYOUT GRIDDED statement follows the traditional ODS statement usage in which you wrap (sandwich) your procedure code with a definitive starting and ending location. ODS layout is designed to allow nested layouts (containers) to provide endless customization.
- The ODS LAYOUT ABSOLUTE statement enables you to specify the exact location on the page to place a layout and region container. Each container needs to be explicitly placed to ensure that there is no unintended overlap. The ODS LAYOUT ABSOLUTE statement follows the traditional ODS statement usage in which you wrap (sandwich) your procedure code with a definitive starting and ending location. ODS layout is designed to allow nested layouts (containers) to provide endless customization.

For complete documentation about ODS LAYOUT statements, see "Arranging Output with ODS LAYOUT" in *SAS Output Delivery System: User's Guide*.

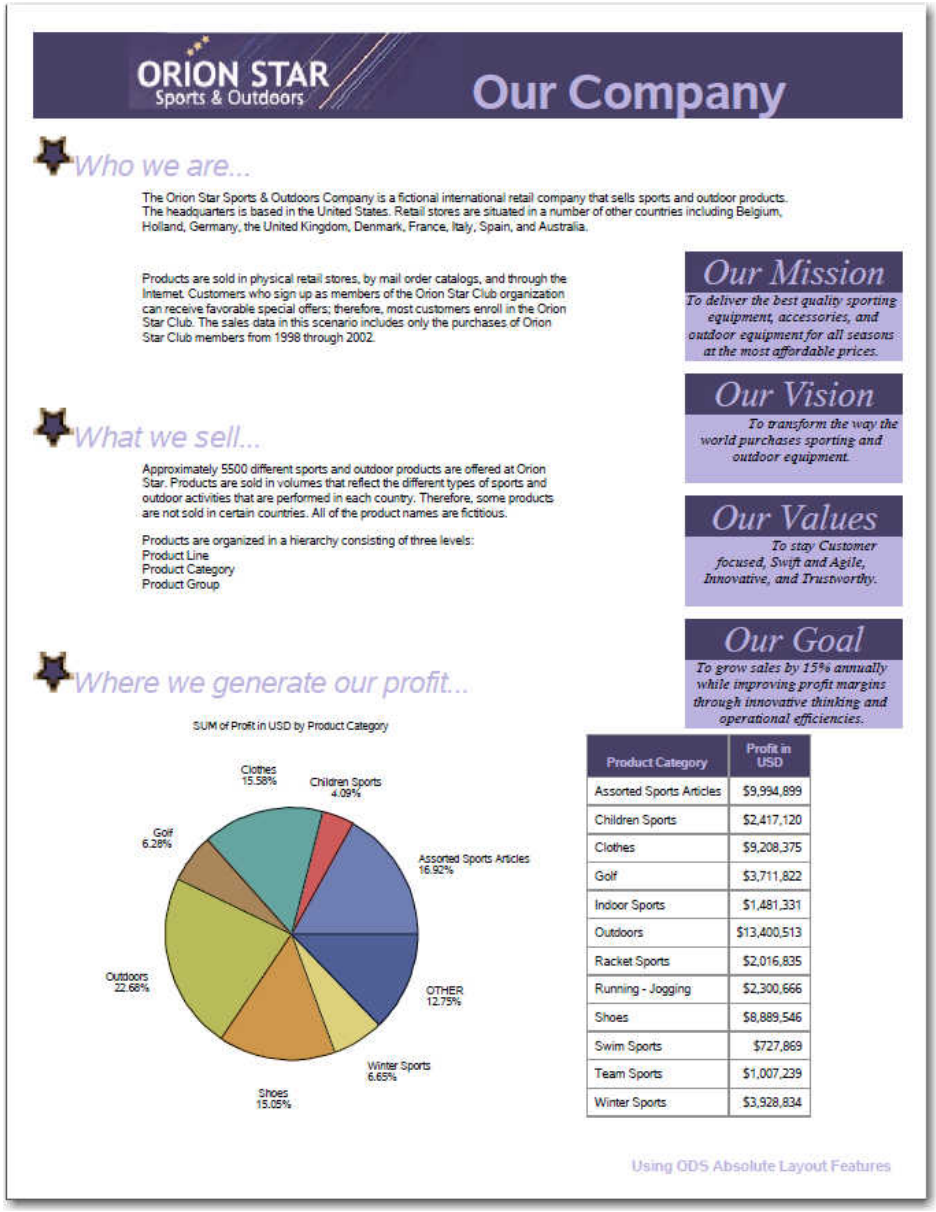
The following example uses the ODS LAYOUT ABSOLUTE statement and the ODS REGION statement for an absolute layout. This example mixes graphics, images, text, and tables.

```
options nodate nonumber;

proc template;
   define style Styles.OrionCalloutBlock;
   parent=Styles.Pearl;
   style LayoutRegion/ background=cxbbb2e0;
end;
run;

ods escapechar="~";
title "~{style [preimage='c:\Public\orionstarHeader.jpg' width=100pct
       background=cx494068 color=cxbbb2e0 font_size=32pt] Our Company }";

footnote "~{style [font_size=10pt just=right color=cxbbb2e0]
       Using ODS Absolute Layout Features.}";

ods pdf file="OrionstarCoInfo.pdf" notoc nogtitle nogfootnote;

ods layout absolute;

ods text="~{style [preimage='c:\Public\starLarge.gif'
       font_style=italic font_size=20pt color=cxbbb2e0]Who we are...}";

ods region y=0.5in x=1in width=6in;
ods text="The Orion Star Sports & Outdoors Company is a fictional
       international retail company that sells sports and outdoor products.
       The headquarters is based in the United States. Retail stores are
       situated in a number of other countries including Belgium, Holland,
       Germany, the United Kingdom, Denmark, France, Italy, Spain, and
```

```
       Australia.";

ods region y=1.25in x=1in width=4in;
ods text="Products are sold in physical retail stores, by mail order
       catalogs, and through the Internet. Customers who sign up as members
       of the Orion Star Club organization can receive favorable special
       offers; therefore, most customers enroll in the Orion Star Club. The
       sales data in this scenario includes only the purchases of Orion
       Star Club members from 1998 through 2002.";

ods region y=2.5in height=1in width=3in;
ods text="~{style [preimage='c:\Public\starLarge.gif' font_style=italic
       font_size=20pt color=cxbbb2e0]What we sell...}";

ods region y=3in x=1in width=4in height=1.75in;
ods text="Approximately 5500 different sports and outdoor products
       are offered at Orion Star. Products are sold in volumes that reflect
       the different types of sports and outdoor activities that are performed
       in each country. Therefore, some products are not sold in certain
       countries. All of the product names are fictitious.";
ods text="~{newline}Products are organized in a hierarchy
       consisting of three levels:";
ods text="Product Line";
ods text="Product Category";
ods text="Product Group";

ods region y=4.75in height=1in width=5in;
ods text="~{style [preimage='c:\Public\starLarge.gif'
       font_style=italic font_size=20pt color=cxbbb2e0]
       Where we generate our profit...}";

ods region y=5.35in width=4.75in height=3.75in;
goptions device=png htext=.1in;
proc gchart data=sashelp.orsales;
   pie product_category / sumvar=profit
                          value=none
                          percent=outside
                          slice=outside;
run;
quit;

ods region y=5.5in x=4.625in width=3in height=3.7in;
proc report nowd data=sashelp.orsales
       style(header)={background=cx494068 color=cxbbb2e0};
       columns product_category profit;
       define product_category / group;
       define profit /analysis sum format=dollar14.;
run;

ods pdf style=Styles.OrionCalloutBlock;

ods region x=6in y=1.0625in width=2in height=1in;
ods text="~{style [background=cx494068 color=cxbbb2e0
       font_size=24pt just=center font_style=italic width=100pct]
       Our Mission }";
ods text="~{style [font_style=italic vjust=center font_size=10pt
```

```
       just=center]To deliver the best quality sporting equipment,
       accessories, and outdoor equipment for all seasons at the
       most affordable prices.}";

ods region x=6in y=2.1875in width=2in height=1in;
ods text="~{style [background=cx494068 color=cxbbb2e0
       font_size=24pt just=center font_style=italic width=100pct]
       Our Vision }";
ods text="~{style [font_style=italic vjust=center font_size=10pt
       just=center]To transform the way the world purchases sporting
       and outdoor equipment.}";

ods region x=6in y=3.3125in width=2in height=1in;
ods text="~{style [background=cx494068 color=cxbbb2e0
       font_size=24pt just=center font_style=italic width=100pct]
       Our Values }";
ods text="~{style [font_style=italic vjust=center
       font_size=10pt just=center]To stay Customer focused, Swift
       and Agile, Innovative, and Trustworthy.}";

ods region x=6in y=4.4375in width=2in height=1in;
ods text="~{style [background=cx494068 color=cxbbb2e0
       font_size=24pt just=center font_style=italic width=100pct]
       Our Goal }";
ods text="~{style [font_style=italic vjust=center font_size=10pt
       just=center]To grow sales by 15% annually while improving
       profit margins through innovative thinking and operational
       efficiencies.}";

ods layout end;
ods pdf close;
```

Output 7.9 *ODS Absolute Layout – Company Information*



ORION STAR
Sports & Outdoors

Our Company

Who we are...

The Orion Star Sports & Outdoors Company is a fictional international retail company that sells sports and outdoor products. The headquarters is based in the United States. Retail stores are situated in a number of other countries including Belgium, Holland, Germany, the United Kingdom, Denmark, France, Italy, Spain, and Australia.

Products are sold in physical retail stores, by mail order catalogs, and through the Internet. Customers who sign up as members of the Orion Star Club organization can receive favorable special offers; therefore, most customers enroll in the Orion Star Club. The sales data in this scenario includes only the purchases of Orion Star Club members from 1998 through 2002.

What we sell...

Approximately 5500 different sports and outdoor products are offered at Orion Star. Products are sold in volumes that reflect the different types of sports and outdoor activities that are performed in each country. Therefore, some products are not sold in certain countries. All of the product names are fictitious.

Products are organized in a hierarchy consisting of three levels:
Product Line
Product Category
Product Group

Where we generate our profit...

Our Mission
To deliver the best quality sporting equipment, accessories, and outdoor equipment for all seasons at the most affordable prices.

Our Vision
To transform the way the world purchases sporting and outdoor equipment.

Our Values
To stay Customer focused, Swift and Agile, Innovative, and Trustworthy.

Our Goal
To grow sales by 15% annually while improving profit margins through innovative thinking and operational efficiencies.

Product Category	Profit in USD
Assorted Sports Articles	$9,994,899
Children Sports	$2,417,120
Clothes	$9,208,375
Golf	$3,711,822
Indoor Sports	$1,481,331
Outdoors	$13,400,513
Racket Sports	$2,016,835
Running - Jogging	$2,300,666
Shoes	$8,889,546
Swim Sports	$727,869
Team Sports	$1,007,239
Winter Sports	$3,928,834

Using ODS Absolute Layout Features

Where to Go from Here

Arranging output dynamically on a page and creating custom reports where you can easily mix graphics, images, text, and tables:

For more information about ODS layouts, see "Arranging Output with ODS LAYOUT" in *SAS Output Delivery System: User's Guide*.

Creating statistical graphics with ODS:

For reference information about the GTL, see *SAS Graph Template Language: Reference*.

Creating statistical graphics with PROC TEMPLATE and GTL:
For usage information about PROC TEMPLATE and the GTL, see the *SAS Graph Template Language: User's Guide*.

Managing the various templates stored in template stores:
For reference information about PROC TEMPLATE statements that help you manage and navigate ODS templates, see "TEMPLATE Procedure: Managing Template Stores" in *SAS Output Delivery System: Procedures Guide*.

Modifying an existing style or creating your own style:
For reference information about style template statements in PROC TEMPLATE, see "TEMPLATE Procedure: Creating a Style Template" in *SAS Output Delivery System: Procedures Guide*.

Creating and modifying ODS tabular output:
For reference information about tabular template statements in PROC TEMPLATE, see "TEMPLATE Procedure: Creating Tabular Output" in *SAS Output Delivery System: Procedures Guide*.

For complete documentation about the ODSTABLE procedure, see "The ODSTABLE Procedure" in *SAS Output Delivery System: Procedures Guide*.

Modifying markup language tagsets that SAS provides or creating your own tagsets:
For reference information about the markup language tagset statements in PROC TEMPLATE, see "TEMPLATE Procedure: Creating Markup Language Tagsets" in *SAS Output Delivery System: Procedures Guide*.

Using ODS with the DATA step:
For information about using ODS with the DATA step, see "Using ODS with the DATA Step" in *SAS Output Delivery System: User's Guide*.

Creating highly customized reports with the DATA step:
For information about the Report Writing Interface, see "Introduction to the Report Writing Interface" in *SAS Output Delivery System: Advanced Topics*.

Creating text and lists to use with ODS output:
For complete documentation about the ODSTEXT procedure, see the "The ODSTEXT Procedure" in *SAS Output Delivery System: Procedures Guide*.

For complete documentation about the ODSLIST procedure, see the "The ODSLIST Procedure" in *SAS Output Delivery System: Procedures Guide*.

Creating custom reports that easily mix SAS graphics, images, text, and tables, and then arrange them on a page:
For complete documentation about ODS LAYOUT statements, see "Arranging Output with ODS LAYOUT" in *SAS Output Delivery System: User's Guide*.

Creating and applying cascading style sheets (CSS) to ODS output:
For complete documentation about using CSS with ODS, see the chapter about CSS in the *SAS Output Delivery System: Advanced Topics*.

Recommended Reading

Here is the recommended reading list for this title. For a complete list of SAS publications, go to http://support.sas.com/publishing/index.html.

- *SAS Output Delivery System: User's Guide*
- *SAS Output Delivery System: Procedures Guide*
- *SAS Output Delivery System: Advanced Topics*
- *Base SAS Procedures Guide*
- *SAS Language Reference: Concepts*
- *SAS Data Set Options: Reference*
- *SAS Functions and CALL Routines: Reference*
- *SAS Statements: Reference*
- *SAS System Options: Reference*
- *Step-by-Step Programming with Base SAS*
- *SAS Graph Template Language: User's Guide*
- *SAS Graph Template Language: Reference*
- SAS 9 ODS CSS Tip Sheet
- SAS 9 ODS EPUB Tip Sheet
- SAS 9 ODS Layout Tip Sheet
- SAS 9 ODS EXCELXP Tip Sheet
- SAS 9 Report Writing Interface Tip Sheet
- SAS 9 ODS List and Text Block Tip Sheet
- SAS offers instructor-led training and self-paced e-learning courses to help you get started with the SAS Output Delivery System and learn advanced techniques for the SAS Output Delivery System. For more information about the courses available, see http://support.sas.com/learn/.

The recommended reading list from **SAS Press** includes:

- *Carpenter's Guide to Innovative SAS Techniques*
- *Carpenter's Complete Guide to the SAS REPORT Procedure*
- *Getting Started with the Graph Template Language in SAS: Examples, Tips, and Techniques for Creating Custom Graphs*

- *Learning SAS by Example: A Programmer's Guide*
- *The Little SAS Book: A Primer*
- *The Little SAS Book for Enterprise Guide*
- *ODS Techniques: Tips for Enhancing Your SAS Output*
- *Output Delivery System: The Basics and Beyond*
- *PROC DOCUMENT by Example Using SAS*
- *PROC TEMPLATE Made Easy: A Guide for SAS Users*
- *Statistical Graphics Procedures by Example: Effective Graphs Using SAS*
- *Statistical Graphics in SAS: An Introduction to the Graph Template Language and the Statistical Graphics Procedures*

For a complete list of SAS publications, go to sas.com/store/books. If you have questions about which titles you need, please contact a SAS Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-0025
Fax: 1-919-677-4444
Email: sasbook@sas.com
Web address: sas.com/store/books

Index

A

advanced topics
  Cascading Style Sheets 91
  CSS 91
  DOCUMENT procedure 85
  ODS Graphics 81
  ODS LAYOUT 100
  ODSLIST procedure 94
  ODSTABLE procedure 96
  ODSTEXT procedure 93
  PROC DOCUMENT 85
  PROC ODSLIST 94
  PROC ODSTABLE 96
  PROC ODSTEXT 93
  PROC TEMPLATE 75
  Report Writing Interface 97
  RWI 97
  TEMPLATE procedure 75

B

Base SAS reporting procedures 5
benefits of ODS 2
BODY= option
  ODS destination statements 41

C

Cascading Style Sheet
  creating, example 91
Cascading Style Sheets
  advanced topics 91
components of ODS 3
CONTENTS= option
  ODS PDF statement 64
creating output
  combined program example 50
  EPUB 44
  Excel 50
  HTML 48
  PDF 47
  PowerPoint 45
  RTF 42
creating, example
  ODS documents 86
crosstabulation tables
  modifying 80
CSS
  advanced topics 91
  creating, example 91
custom reports, example 19
customizing output 1, 55
  combined program example 69
  EPUB 58
  Excel 67
  HTML 66
  PDF 64
  POWERPOINT 60
  RTF 57
  TABULATE procedure 62

D

data component 16
DATA step
  advanced topics 89
  ODS and 89
  ODS reports with 90
default output, example 19
destinations
  See ODS destinations
DOCUMENT procedure
  advanced features 85
  advanced topics 85
  capabilities 6
  compared to Documents window 40
documents
  See ODS documents
Documents window 7, 13
  compared to DOCUMENT procedure 40
  opening 23
  selecting output objects 31
  viewing output object labels 29

E

enhanced HTML

creating output 48
EPUB output
creating 44
customizing 58
Excel output
creating 50
customizing 67

F

features of ODS 1
FILE PRINT ODS statement 90
FILE= option
ODS destination statements 41

G

global statements 7
Graph Template Language (GTL) 81
graphical output objects 3
graphics 81
GTL (Graph Template Language) 81

H

HTML output
creating 48
customizing 66

I

identifying output objects 27
integrating output with applications 41
item stores 16

N

NAME= option
ODS DOCUMENT statement 23

O

ODS (Output Delivery System) 2
ODS block 41
ODS destination statements 7
ODS destinations 16
categories of 10
ODS DOCUMENT CLOSE statement 23
ODS DOCUMENT statement 6, 23, 85
ODS document stores 88
ODS documents 16
creating, example 23, 86
loading output 35
ODS EPUB statement
creating output 44
customizing output 58
ODS EXCLUDE statement 31
ODS formatting statements 90
ODS global statements 7
ODS Graphics 81
advanced topics 81
output 79
ODS HTML statement
creating output 48
customizing output 66
ODS LAYOUT
advanced topics 100
creating, example 100
ODS LAYOUT statement
creating, example 100
ODS PDF statement
creating output 47
customizing output 64
ODS POWERPOINT statement
creating output 45
customizing output 60
ODS RTF statement
creating output 42
customizing output 57
ODS SELECT statement 30
ODS TAGSETS.EXCELXP statement
creating output 50
customizing output 67
ODS templates 17, 76
creating with TEMPLATE procedure 12
modifying style templates 77
modifying table templates 76
ODS TRACE statement 27
ODS windows 13
ODSLIST procedure
advanced topics 94
creating, example 94
ODSTABLE procedure
advanced topics 96
creating, example 96
ODSTEXT procedure
advanced topics 93
creating, example 93
OPTIONS option
ODS EPUB statement 58
output
creating default, example 19
creating EPUB 44
creating Excel 50
creating HTML 48
creating PDF 47
creating PowerPoint 45
creating RTF 42
customizing 1, 55
customizing EPUB 58
customizing Excel 67

customizing HTML 66
customizing PDF 64
customizing POWERPOINT 60
customizing RTF 57
customizing TABULATE procedure output 62
formatting for third-party software 7
integrating with applications 41
loading into ODS document 35
location 4
ODS Graphics 79
output control statements 7
output objects 3, 17
identifying 27
selecting 30
selecting in Documents window 31
templates 12
viewing in Documents window 29

P

PATH= option
ODS HTML statement 48
PDF output
creating 47
customizing 64
PDFTOC= option
ODS PDF statement 64
PNG files 81
PowerPoint output
creating 45
POWERPOINT output
customizing 60
PRINT procedure 5
PROC DOCUMENT
advanced topics 85
creating, example 86
PROC ODSLIST
advanced topics 94
creating, example 94
PROC ODSTABLE
advanced topics 96
creating, example 96
PROC ODSTEXT
advanced topics 93
creating, example 93
PROC TEMPLATE
advanced topics 75
processing for ODS 3
PUT statement 90

R

REPORT procedure 5
Report Writing Interface
advanced topics 97
creating, example 98
RTF output
creating 42
customizing 57
RWI
advanced topics 97
creating, example 98

S

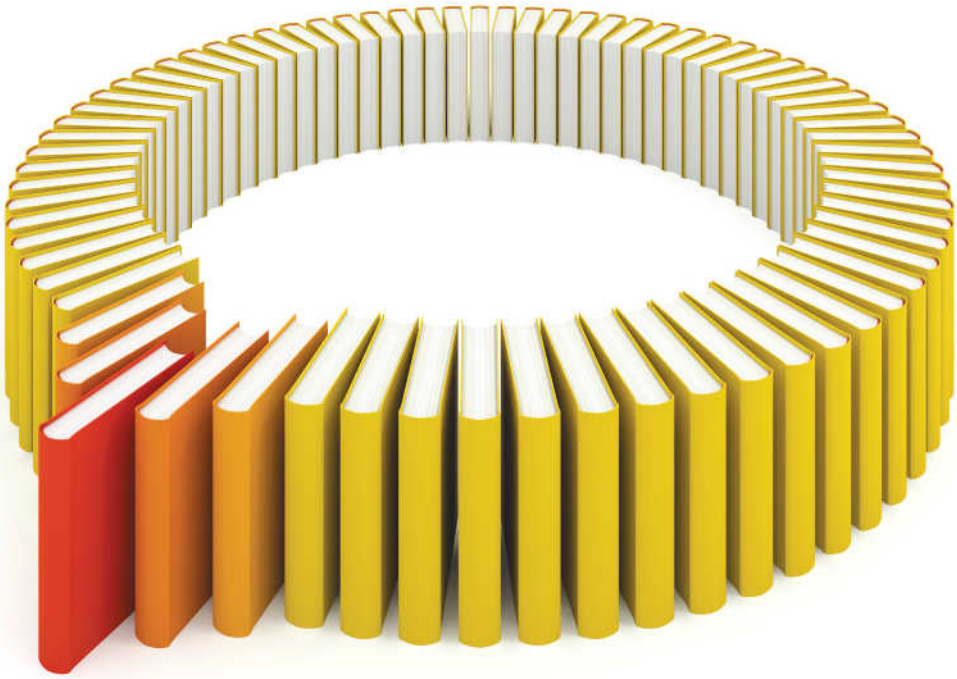
SAS formatted destinations 10
SAS styles 55
selecting output objects 30
StatGraph templates 79, 82
style templates
modifying 77
STYLE= option 57
ODS HTML statement 66
ODS PDF statement 64
ODS POWERPOINT statement 60
ODS RTF statement 57
ODS TAGSETS.EXCELXP statement 67
TABULATE procedure 62
styles provided by SAS 55
suboptions
ODS EPUB statement 58
SYTLES
ODS POWERPOINT statement 60

T

table templates 17
modifying 76
tabular output objects 3
TABULATE procedure 5
customizing output 62
tagsets
creating 78
list of 4
Template Browser window 15
TEMPLATE procedure
advanced topics 75
creating and modifying ODS templates 12
creating tagsets 78
modifying crosstabulation tables 80
modifying style templates 77
modifying table templates 76
template stores 16
templates
See ODS templates
Templates window 14
opening 55
third-party formatted destinations 10

U
usage of ODS 5

W
windows for ODS 13
WRITE option
  ODS DOCUMENT statement 23



Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore
for additional books and resources.

sas
THE POWER TO KNOW.

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2013 SAS Institute Inc. All rights reserved. S107969US.0613