Getting Started with the SAS® 9.4 Output Delivery System
# Contents

*About This Book* ................................................................. v

**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 in the SAS Windowing Environment ........ 14  
  Words to Know ................................................................ 17  

**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 ....................................................... 27  

**Chapter 4 / Selecting the Contents of Your Report** .................. 29  
  About the Tasks That You Will Perform ................................ 29  
  Identify the Output Objects ............................................... 30  
  Select the Output Objects .................................................. 33  
  For More Information ....................................................... 44  

**Chapter 5 / Integrating Output with Popular Business Applications and SAS** ......................... 45  
  About the Tasks That You Will Perform ................................ 45  
  Creating RTF Output ......................................................... 46  
  Creating an E-Book ........................................................... 48  
  Creating PowerPoint Output ............................................... 50  
  Creating PDF Output ........................................................ 52  
  Creating Enhanced HTML Output ....................................... 53  
  Creating Excel Output ....................................................... 55  
  Combined Program ........................................................... 56  
  For More Information ....................................................... 58  

**Chapter 6 / Customizing the Presentation of a Report** .............. 59  
  About the Tasks That You Will Perform ................................ 59  
  Customized RTF Output ..................................................... 61  
  Customized E-Book ......................................................... 63  
  Customized PowerPoint Output .......................................... 65  
  Customized PROC TABULATE Output .................................. 67  
  Customized PDF Output .................................................... 69  
  Customized HTML Output .................................................. 70  
  Customized Excel Output ................................................... 72  
  Combined Program ........................................................... 74  
  For More Information ....................................................... 77
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);

\(<\text{LITERAL\_ARGUMENT}>\text{argument-1} \text{<<LITERAL\_ARGUMENT}>\text{argument-2} \ldots >\)

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 (\(\ldots\)) indicates that additional literals and arguments are allowed.

The BY statement is an example of this argument:

BY <DESCENDING> \text{variable-1} <<DESCENDING> \text{variable-2} \ldots >;

\text{argument-1} <\text{options}> \text{argument-2} <\text{options}> \ldots >

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 (\(\ldots\)) 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 <(\text{format-options})>
\<value\-range\-set\-1> <(\text{picture-1\-options})>
\<value\-range\-set\-2> <(\text{picture-2\-options})> \ldots >;

\text{argument-1=} \text{value-1} \text{argument-2=} \text{value-2} \ldots >

specifies that the argument must be assigned a value and that you can specify multiple arguments. The ellipsis (\(\ldots\)) 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 \text{variable-1=} \text{label-1} \<\text{variable-2=} \text{label-2} \ldots >;

\text{argument-1}, \text{argument-2}, \ldots >

specifies that one argument is required and that you can specify multiple arguments that are separated by a comma or other punctuation. The ellipsis (\(\ldots\)) 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, \ldots >
INTO :macro\-variable\-specification-1 <, :macro\-variable\-specification-2, \ldots >

\begin{itemize}
\item \textbf{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.
\end{itemize}
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:

```sas
ERROR <message>;
```

**UPPERCASE**
identifies arguments that are literals.

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

```sas
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:

```sas
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`:

```sas
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:

```sas
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:

\[
\text{CAT} (item-1 <, item-2, \ldots>)
\]

| 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:

\[
\text{CMPMODEL=} \text{BOTH | CATALOG | XML}
\]

| 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 \textit{item} arguments are allowed, and they must be separated by a comma:

\[
\text{CAT} (item-1 <, item-2, \ldots>)
\]

| ‘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 \textit{text} is enclosed in quotation marks:

\[
\text{FOOTNOTE} <n> <ods-format-options 'text' | "text">;
\]

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

In this example, each statement ends with a semicolon:

\[
\text{data namegame;}
\text{length color name $8;}
\text{color = 'black';}
\text{name = 'jack';}
\text{game = trim(color) || name;}
\text{run;}
\]

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:

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

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
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 accessible SAS output.
  ODS enables you to make some of your output accessible by assistive technology such as a screen reader. For more information about creating accessible output with ODS, see Creating Accessible SAS® 9.4 Output Using ODS and ODS Graphics.

---

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***

<table>
<thead>
<tr>
<th>Table</th>
<th>CHTML</th>
<th>CSV</th>
<th>CSVALL</th>
<th>CSVBYLINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>CHTML</td>
<td>CSV</td>
<td>CSVALL</td>
<td>CSVBYLINE</td>
</tr>
<tr>
<td>Default</td>
<td>EXCELXP</td>
<td>HTMLPANEL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSOFFICE2K</td>
<td>PHTML</td>
<td>TAGSETS.RTF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 1.2  *Additional Diagnostic Tagsets That SAS Supports***

<table>
<thead>
<tr>
<th>Tagset</th>
<th>Style_Popup</th>
<th>Text_Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVENT_MAP</td>
<td>NAMEDHTML</td>
<td>SHORT_MAP</td>
</tr>
<tr>
<td>STYLE_POPUP</td>
<td>TEXT_MAP</td>
<td></td>
</tr>
</tbody>
</table>
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.

In SAS Studio, by default, output is generated in the HTML5, PDF, and RTF formats. The default HTML5 results are displayed in the RESULTS tab in SAS Studio. The PDF and RTF output is generated but is not displayed. If you want to save results from SAS Studio, you can download your results in the HTML5, PDF, or RTF output formats from the SAS Studio RESULTS tab.

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 7
- ODS Global Statements on page 7
- ODSLIST Procedure on page 12
- ODSTABLE Procedure on page 13
- ODSTEXT Procedure on page 13
- TEMPLATE Procedure on page 13

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:

```sas
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*

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 [W3C Cascading Style Sheets home page](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 79. 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.

In the SAS windowing environment, 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 “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 options 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.

```sas
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

Output 1.3  PDF Output with Banker Style Applied
# Chapter 1 / Introduction

**Output 1.4** Default RTF Output

![The SAS System](image1)

**Output 1.5** RTF Output with BarrettsBlue Style Applied

![The SAS System](image2)
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

<table>
<thead>
<tr>
<th>Category</th>
<th>Destinations</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formatted by SAS</td>
<td>DOCUMENT</td>
<td>ODS document</td>
</tr>
<tr>
<td></td>
<td>LISTING</td>
<td>SAS LISTING output</td>
</tr>
<tr>
<td></td>
<td>OUTPUT</td>
<td>SAS data set</td>
</tr>
<tr>
<td>Formatted by a Third Party</td>
<td>EPUB</td>
<td>Output formatted for e-book readers</td>
</tr>
<tr>
<td></td>
<td>EPUB3</td>
<td>Output formatted for e-book readers</td>
</tr>
<tr>
<td></td>
<td>Excel</td>
<td>Excel spreadsheet files compatible with Microsoft Office 2010 and later versions</td>
</tr>
<tr>
<td></td>
<td>HTML</td>
<td>HTML file for online viewing</td>
</tr>
<tr>
<td></td>
<td>MARKUP</td>
<td>Markup language tagsets</td>
</tr>
<tr>
<td></td>
<td>PowerPoint</td>
<td>PowerPoint slides</td>
</tr>
<tr>
<td></td>
<td>PRINTER</td>
<td>Printable output in one of three different formats: PCL, PDF, or PS (PostScript)</td>
</tr>
<tr>
<td></td>
<td>RTF</td>
<td>Output written in Rich Text Format for use with Microsoft Word 2000</td>
</tr>
</tbody>
</table>
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 79. 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 79. 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 79. For complete documentation about the ODSLIST procedure, see “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 79. For complete documentation about the ODSTABLE procedure, see “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 79. For complete documentation about the ODSTEXT procedure, see “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 79.

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

ODS-Specific Windows in the SAS Windowing Environment

Overview

In the SAS windowing environment, 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.
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.
**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.
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.

Note: If you run your example in SAS Studio, then RTF, HTML 5, and PDF output is created by default. The HTML output is sent to the RESULTS tab.

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

#### Actual Product Sales
(millions of dollars)

<table>
<thead>
<tr>
<th>Region by Division and Type</th>
<th>Product type</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FURNITURE</td>
<td>OFFICE</td>
</tr>
<tr>
<td></td>
<td>Actual Sales</td>
<td>Actual Sales</td>
</tr>
<tr>
<td></td>
<td>Sum</td>
<td>Sum</td>
</tr>
<tr>
<td><strong>EAST</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Division</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSUMER</td>
<td>$72,570</td>
<td>$108,696</td>
</tr>
<tr>
<td>EDUCATION</td>
<td>$73,501</td>
<td>$116,194</td>
</tr>
<tr>
<td>Total</td>
<td>$146,071</td>
<td>$224,890</td>
</tr>
<tr>
<td><strong>WEST</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Division</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSUMER</td>
<td>$76,200</td>
<td>$105,020</td>
</tr>
<tr>
<td>EDUCATION</td>
<td>$87,945</td>
<td>$110,902</td>
</tr>
<tr>
<td>Total</td>
<td>$164,145</td>
<td>$215,922</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Division</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSUMER</td>
<td>$148,779</td>
<td>$213,796</td>
</tr>
<tr>
<td>EDUCATION</td>
<td>$141,846</td>
<td>$226,096</td>
</tr>
<tr>
<td>Total</td>
<td>$290,625</td>
<td>$439,892</td>
</tr>
</tbody>
</table>
Creating an ODS Document

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 following ODS DOCUMENT statement, 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 following ODS DOCUMENT CLOSE statement closes the document so that it can be viewed in the Documents window.

In the SAS windowing environment, 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`.
In SAS studio, you must use the DOCUMENT procedure to display ODS documents. For more information, see “Navigating the Directory and Listing the Entries” in SAS Output Delivery System: Procedures Guide.

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:

Selecting the Contents of Your Report

About the Tasks That You Will Perform ................................................................. 29
Identify the Output Objects .................................................................................. 30
   Using the ODS TRACE Statement ................................................................. 30
   Using the Documents Window ....................................................................... 31
Select the Output Objects .................................................................................... 33
   Using ODS Statements .................................................................................. 33
   Using the Documents Window ....................................................................... 34
   Comparing the Two Methods ......................................................................... 44
For More Information ........................................................................................... 44

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.

Note: This section assumes that you are using the SAS windowing environment. If you are using SAS Studio, see “About SAS Studio” in SAS Output Delivery System: User’s Guide.
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.

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

**TIP** In SAS studio, you must use the DOCUMENT procedure to display and explore ODS documents. For more information, see “Navigating the Directory and Listing the Entries” in SAS Output Delivery System: Procedures Guide.
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.
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.

```plaintext
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.
Because you selected the library Sasuser, your document is stored permanently.
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 “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. The Documents window is only available in the SAS windowing environment.

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. This method works in SAS Studio, SAS Enterprise Guide, or the SAS windowing environment.

For More Information

- For complete documentation on the DOCUMENT procedure, see “DOCUMENT Procedure” in SAS Output Delivery System: Procedures Guide.
Integrating Output with Popular Business Applications and SAS

About the Tasks That You Will Perform

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.

Note: The examples assume that you are using the SAS windowing environment. If you are using SAS Studio, see “About SAS Studio” in SAS Output Delivery System: User’s Guide.

An ODS block has the following form:

```sas
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 `options` 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.

```sas
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;
```
Output 5.1  PROC TABULATE Output Viewed in Microsoft Word

<table>
<thead>
<tr>
<th>Region by Division and Type</th>
<th>Product type</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FURNITURE</td>
<td>OFFICE</td>
<td>Total</td>
</tr>
<tr>
<td></td>
<td>Actual Sales</td>
<td>Actual Sales</td>
<td>Actual Sales</td>
</tr>
<tr>
<td></td>
<td>Sum</td>
<td>Sum</td>
<td>Sum</td>
</tr>
<tr>
<td><strong>Region</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EAST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSUMER</td>
<td>$72,370</td>
<td>$108,686</td>
<td>$181,256</td>
</tr>
<tr>
<td>EDUCATION</td>
<td>$73,901</td>
<td>$115,104</td>
<td>$189,005</td>
</tr>
<tr>
<td>Total</td>
<td>$146,271</td>
<td>$223,790</td>
<td>$370,261</td>
</tr>
<tr>
<td>WEST</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSUMER</td>
<td>$76,209</td>
<td>$105,020</td>
<td>$181,229</td>
</tr>
<tr>
<td>EDUCATION</td>
<td>$67,945</td>
<td>$110,902</td>
<td>$178,847</td>
</tr>
<tr>
<td>Total</td>
<td>$144,154</td>
<td>$215,922</td>
<td>$360,076</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSUMER</td>
<td>$148,779</td>
<td>$213,706</td>
<td>$362,485</td>
</tr>
<tr>
<td>EDUCATION</td>
<td>$141,846</td>
<td>$226,006</td>
<td>$367,852</td>
</tr>
<tr>
<td>Total</td>
<td>$290,625</td>
<td>$439,712</td>
<td>$730,337</td>
</tr>
</tbody>
</table>
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.

```sas
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;
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.

```sas
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;

ods powerpoint close;

Output 5.4  PROC TABULATE Output Viewed in PowerPoint

<table>
<thead>
<tr>
<th>Region by Division and Type</th>
<th>Product type</th>
<th>FURNITURE</th>
<th>OFFICE</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>East</td>
<td>CONSUMER</td>
<td>$72,570</td>
<td>$108,686</td>
<td>$181,256</td>
</tr>
<tr>
<td></td>
<td>EDUCATION</td>
<td>$73,901</td>
<td>$115,104</td>
<td>$189,005</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>$146,471</td>
<td>$223,790</td>
<td>$370,261</td>
</tr>
<tr>
<td>West</td>
<td>CONSUMER</td>
<td>$70,269</td>
<td>$105,020</td>
<td>$185,229</td>
</tr>
</tbody>
</table>

(Continued)
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.

```sas
ods html close;
options nodate nonumber;
proc sort data=sashelp.prdsale out=prdsale;
  by Country;
run;
ods pdf file='your-file-path/PDFPrdsale.pdf' pdftoc=2;
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;

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;
Creating Excel Output

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

```plaintext
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
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.

Example Code 5.1  Combined Program with 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; /* 11 */
ods tagsets.excelxp file='your-file-path.Prdsale.xls'; /* 12 */
ods tagsets.excelxp select PRINT; /* 13 */
title "PROC PRINT Output";

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


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.
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.

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

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.

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.

For More Information

Customizing the Presentation of a Report

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.

Note: The examples assume that you are using the SAS windowing environment. If you are using SAS Studio, see “About SAS Studio” in SAS Output Delivery System: User’s Guide.

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

**TIP** To view all of the styles provided by SAS in SAS Studio, Enterprise Guide, or the SAS windowing environment, submit the following code:

```sas
proc template;
  path sashelp.tmplmst;
  list styles;
run;
```

### Output 6.1 SAS Styles

<table>
<thead>
<tr>
<th>Contents of Styles</th>
<th>Analysis</th>
<th>BarrettsBlue</th>
<th>BlockPrint</th>
<th>Daisy</th>
<th>Default</th>
<th>Dove</th>
<th>Dtree</th>
<th>EGDefault</th>
<th>FancyPrinter</th>
<th>Festival</th>
<th>FestivalPrinter</th>
<th>Gantt</th>
<th>GrayscalePrinter</th>
<th>Harvest</th>
<th>HighContrast</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoFontDefault</td>
<td>Journal</td>
<td>Journal1a</td>
<td>Journal2</td>
<td>Journal2a</td>
<td>Journal3</td>
<td>Journal3a</td>
<td>Listing</td>
<td>Meadow</td>
<td>MeadowPrinter</td>
<td>Minimal</td>
<td>MonochromePrinter</td>
<td>Monospace</td>
<td>Moonflower</td>
<td>Netdraw</td>
<td></td>
</tr>
<tr>
<td>SeaSidePrinter</td>
<td>Normal</td>
<td>NormalPrinter</td>
<td>Ocean</td>
<td>Pearl</td>
<td>Plateau</td>
<td>PowerPointDark</td>
<td>PowerPointLight</td>
<td>Printer</td>
<td>Raven</td>
<td>Rt</td>
<td>Seaside</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Output Destination</th>
<th>Default Style</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTF</td>
<td>RTF</td>
</tr>
<tr>
<td>HTML</td>
<td>HTMLBlue</td>
</tr>
<tr>
<td>PDF</td>
<td>Pearl</td>
</tr>
<tr>
<td>EPUB</td>
<td>Daisy</td>
</tr>
<tr>
<td>POWERPOINT</td>
<td>PowerPointLight</td>
</tr>
<tr>
<td>EXCEL</td>
<td>Default</td>
</tr>
</tbody>
</table>

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

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

**TIP** To view the source of the Daisy style in SAS Studio, Enterprise Guide, or the SAS windowing environment, submit the following code:

```
proc template;
  source styles.daisy;
run;
```

**Figure 6.1** Source of Daisy 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 Daisy style specified tells ODS to use the Daisy 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.

```sas
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=daisy;

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 / novarnamese;
   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=Daisy option in the ODS RTF statement applies the style for PROC SGPANEL, too.
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;
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.

```sas
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.) /
PROC UNIVARIATE Output;
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;

title 'Sales Figures for First Quarter by Product';
proc print data=sashelp.prdsale;
run;

ods powerpoint close;

Output 6.4  PROC SGPANEL 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=daisy;

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;

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 Daisy style and the specific style overrides specified in
individual statements.
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 Daisy style specified tells ODS to use the Daisy 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.

```plaintext
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=daisy;
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.) /
The STYLE= option with the Daisy style specified in the ODS HTML statement tells ODS to use the Daisy style for all HTML output.

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

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

```
ods pdf close;
```

**Output 6.7** Customized PDF Output Viewed in Adobe Acrobat
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.} / misstext=[label='Missing' style=[fontweight=light]]
  box=[label='Region by Division and Type' style=[fontstyle=italic]];
run;
ods select ExtremeObs Quantiles Moments;
proc univariate data=prdsale;
  by Country;
  var actual;
run;
ods html close;
Customized Excel Output

The STYLE= option with the Daisy style specified in the ODS TAGSETS.EXCELP statement tells ODS to use the Daisy style for all Excel output. Specifying the OPTIONS suboption (DOC="HELP") prints Help for the ODS TAGSETS.EXCELP 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.

```sas
ods tagsets.excelxp file='your-file-path/Prdsale.xls' style=daisy options (doc="help");
title "PROC PRINT Output";
proc print data=sashelp.prdsale;
run;
ods tagsets.excelxp close;
```
### Customized Excel Output

<table>
<thead>
<tr>
<th>Obs</th>
<th>ACTUAL</th>
<th>PREDICT</th>
<th>COUNTRY</th>
<th>REGION</th>
<th>DIVISION</th>
<th>PRODTYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$925.00</td>
<td>$850.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>2</td>
<td>$999.00</td>
<td>$297.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>3</td>
<td>$608.00</td>
<td>$846.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>4</td>
<td>$642.00</td>
<td>$533.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>5</td>
<td>$656.00</td>
<td>$646.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>6</td>
<td>$948.00</td>
<td>$466.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>7</td>
<td>$612.00</td>
<td>$717.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>8</td>
<td>$114.00</td>
<td>$564.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>9</td>
<td>$685.00</td>
<td>$236.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>10</td>
<td>$657.00</td>
<td>$494.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>11</td>
<td>$608.00</td>
<td>$503.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>12</td>
<td>$353.00</td>
<td>$266.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>13</td>
<td>$107.00</td>
<td>$190.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>14</td>
<td>$354.00</td>
<td>$139.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>15</td>
<td>$101.00</td>
<td>$217.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>16</td>
<td>$553.00</td>
<td>$560.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>17</td>
<td>$377.00</td>
<td>$148.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>18</td>
<td>$431.00</td>
<td>$762.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>19</td>
<td>$511.00</td>
<td>$457.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>20</td>
<td>$157.00</td>
<td>$632.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>21</td>
<td>$520.00</td>
<td>$628.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>22</td>
<td>$114.00</td>
<td>$491.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>23</td>
<td>$277.00</td>
<td>$0.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>24</td>
<td>$561.00</td>
<td>$979.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
<tr>
<td>25</td>
<td>$220.00</td>
<td>$688.00</td>
<td>CANADA</td>
<td>EAST</td>
<td>EDUCATION</td>
<td>FURNIT</td>
</tr>
</tbody>
</table>

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.

Example Code 6.1  Combined Program with Explanations

```sas
proc sort data=sashelp.prdsale out=prdsale;
   by Country;
run;
opts nodate nonumber;
ods document name=work.prddocument{write}::*1*;
ods html path='your-directory-path' body='bodyPrdsale.htm' contents='contentsPrdsale.htm' frame='framePrdsale.htm' style=daisy; /* 2 */
ods rtf file='your-file-path/RTFPrdsaleCustom.rtf' style=daisy; /* 3 */
ods powerpoint file='your-file-path/PrdsaleCustom.ppt' style=PowerPointDark; /* 4 */
```
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=daisy;
ods tagsets.excelxp select PRINT;
title "PROC PRINT Output";
proc print data=sashelp.prdsale;
run;

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

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= option applies the daisy style to all RTF output.

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.

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.

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

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

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

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

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

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.

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

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

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.

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.

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.EXCELSXP statement with the FILE= option opens the TAGSETS.EXCELSXP 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 EXCELSXP 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= option applies the daisy style to all TAGSETS.EXCELSXP output.

19 The ODS TAGSETS.EXCELSXP 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.

For More Information

- For information about the ODS styles provided by SAS, see “Style Templates” in SAS Output Delivery System: Advanced Topics.
- For conceptual information about ODS statements, see “Introduction to ODS Language Statements” 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.
Next Steps: A Quick Look at Advanced Features

Working with the TEMPLATE Procedure

Introduction to the TEMPLATE Procedure

What Can You Do with the TEMPLATE Procedure?

Working with ODS Graphics

Advanced Features of the DOCUMENT Procedure

Overview

Working with the DOCUMENT Procedure

ODS and the DATA Step

Working with Cascading Style Sheets (CSS)

Working with the ODSTEXT Procedure

Working with the ODSLIST Procedure

Working with the ODSTABLE Procedure

Working with the Report Writing Interface

Working with ODS LAYOUT

Where to Go from Here

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
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
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.
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.
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:

```sas
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;
end;
```
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 \texttt{FREQ} Procedure

<table>
<thead>
<tr>
<th>Frequency Percent</th>
<th>Row Percent</th>
<th>Column Percent</th>
<th>City Government Form</th>
<th>Number of Meetings Scheduled</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>0</td>
<td>100 or Less</td>
<td>101-200</td>
<td>201-300</td>
<td>Over 300</td>
</tr>
<tr>
<td>?</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>?</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Not Applicable</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Council Manager</td>
<td>0</td>
<td>12.30</td>
<td>16.89</td>
<td>12.83</td>
<td>13.64</td>
</tr>
<tr>
<td>Commission</td>
<td>0</td>
<td>22.27</td>
<td>29.06</td>
<td>23.22</td>
<td>24.64</td>
</tr>
<tr>
<td>Mayor Council</td>
<td>0</td>
<td>55.95</td>
<td>50.88</td>
<td>62.03</td>
<td>46.44</td>
</tr>
<tr>
<td>Total</td>
<td>0</td>
<td>155.47</td>
<td>163.63</td>
<td>149.84</td>
<td>130.64</td>
</tr>
</tbody>
</table>

---

**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:

```plaintext
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;
```
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 \((\text{limitupper}=\text{upper} \quad \text{limitlower}=\text{lower})\). 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

Iris Petal and Sepal Sizes

<table>
<thead>
<tr>
<th>Sepal Length (mm)</th>
<th>Sepal Width (mm)</th>
<th>Petal Length (mm)</th>
<th>Petal Width (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Iris Species: ○ Setosa  + Versicolor  X Virginica
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 “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.
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:

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

```sas
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 “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](image)
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 “Dictionary of 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**

<table>
<thead>
<tr>
<th>Steps</th>
<th>Tools</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify formatting</td>
<td>ODS formatting statements can specify formats such as listing, HTML,</td>
<td>You can produce output in multiple formats at the same time by specifying</td>
</tr>
<tr>
<td>for your output.</td>
<td>RTF, EPUB, PowerPoint, and PDF.</td>
<td>more than one format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> If you want only the default output, then you do not need a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>destination ODS statement.</td>
</tr>
<tr>
<td>Specify structure.</td>
<td>The ODS option in the FILE statement lists the variables and the</td>
<td>Additional suboptions give you even more control over the structure.</td>
</tr>
<tr>
<td></td>
<td>order in which they appear in the output.</td>
<td></td>
</tr>
</tbody>
</table>
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. 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**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Where to learn more</th>
</tr>
</thead>
</table>

### 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

---

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

```sas
options nodate nonumber obs=25;
ods html close;
ods pdf nobookmarklist nobookmarkgen file="myoutput.pdf"
    csstitle="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

![Output of the SAS code with CSS applied](image)

## 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.


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

```sas
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.


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

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

ods powerpoint file="DefaultStyle.ppt";
odspowerpoint(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;
```
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 “ODSTABLE Procedure” in SAS Output Delivery System: Procedures Guide. For complete documentation about customizing tables with the TEMPLATE procedure, see “TEMPLATE Procedure: Creating Tabular Templates” in SAS Output Delivery System: Procedures Guide.

This example creates a table with customized columns.

```sas
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:
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:

Output 7.7 Customizing Columns

<table>
<thead>
<tr>
<th>Age of Student</th>
<th>Gender</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>M</td>
<td>69</td>
<td>112.5</td>
</tr>
<tr>
<td>13</td>
<td>F</td>
<td>56.5</td>
<td>84</td>
</tr>
<tr>
<td>13</td>
<td>F</td>
<td>65.3</td>
<td>98</td>
</tr>
<tr>
<td>14</td>
<td>F</td>
<td>62.8</td>
<td>102.5</td>
</tr>
<tr>
<td>14</td>
<td>M</td>
<td>63.5</td>
<td>102.5</td>
</tr>
<tr>
<td>12</td>
<td>M</td>
<td>57.3</td>
<td>83</td>
</tr>
<tr>
<td>12</td>
<td>F</td>
<td>59.8</td>
<td>84.5</td>
</tr>
<tr>
<td>15</td>
<td>F</td>
<td>62.5</td>
<td>112.5</td>
</tr>
<tr>
<td>13</td>
<td>M</td>
<td>62.5</td>
<td>84</td>
</tr>
<tr>
<td>12</td>
<td>M</td>
<td>59</td>
<td>99.5</td>
</tr>
<tr>
<td>11</td>
<td>F</td>
<td>51.3</td>
<td>50.5</td>
</tr>
<tr>
<td>14</td>
<td>F</td>
<td>64.3</td>
<td>90</td>
</tr>
<tr>
<td>12</td>
<td>F</td>
<td>56.3</td>
<td>77</td>
</tr>
<tr>
<td>15</td>
<td>F</td>
<td>66.5</td>
<td>112</td>
</tr>
<tr>
<td>16</td>
<td>M</td>
<td>72</td>
<td>150</td>
</tr>
</tbody>
</table>
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;
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.

```sas
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 nogofignotofootnoto;
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.

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.

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
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.
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 complete documentation about the ODSTABLE procedure, see “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:

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

For complete documentation about the ODSLIST procedure, see the “ODS_LIST 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.