# Contents

*About This Book* ................................. vii  
*What's New in the Output Delivery System* ............................... xiii  
*Accessibility* ........................................ xxv  

## PART 1  Introduction  1  

**Chapter 1 • Getting Started with the Output Delivery System** ............................... 3  
Welcome to the Output Delivery System .............................................. 3  
Gallery of ODS Samples ............................................................... 4  
A Quick Start to Using ODS ......................................................... 13  
Creating Accessible Output with the Output Delivery System .................. 19  
Working with Output Defaults ...................................................... 21  
Where to Go from Here ............................................................... 25  

## PART 2  Concepts  27  

**Chapter 2 • Output Delivery System: Basic Concepts** ........................... 29  
Overview of How ODS Works ......................................................... 29  
Understanding Item Stores and Template Stores .................................. 32  
Understanding ODS Destinations ..................................................... 33  
How ODS Determines the Destinations for an Output Object .................. 38  
Selection and Exclusion Lists ....................................................... 39  
Customized Output for an Output Object ......................................... 39  
Customizing Titles and Footnotes .................................................. 40  
Changing SAS Registry Settings for ODS .......................................... 40  

## PART 3  Output Delivery System and the DATA Step  51  

**Chapter 3 • Using ODS with the DATA Step** ................................. 53  
Introduction ................................................................. 53  
How ODS Works with the DATA Step .............................................. 54  
Syntax for ODS Enhanced Features in a DATA Step ............................. 55  
Dictionary ................................................................. 55  
Examples ................................................................. 69  

## PART 4  SAS Studio and ODS  91  

**Chapter 4 • SAS Studio and ODS** ............................................. 93  
About SAS Studio ................................................................. 93  
The SAS Studio Output Environment .............................................. 94
Appendices

Appendix 1 • Output Object Table Names
ODS Table Names and the Base SAS Procedures That Produce Them
ODS Table Names and the Base SAS High-Performance Procedures That Produce Them
ODS Table Names and the Base SAS Statistical Procedures That Produce Them
ODS Table Names and the SAS/STAT Procedures That Produce Them
ODS Table Names and the SAS/STAT High-Performance Procedures That Produce Them
ODS Table Names and the SAS/ETS Procedures That Produce Them
ODS Table Names and the SAS/ETS High-Performance Procedures That Produce Them
ODS Table Names and the SAS/QC Procedures That Produce Them

Appendix 2 • Example Programs
Examples from the Gallery of ODS Samples
Creating the $CNTRY Format
Creating the Charity Data Set
Creating the DIVFMT. and USETYPE. Formats
Creating the DistrData Data Set
Creating the Univ ODS Document
Creating the Employee_Data Data Set
Creating the Energy Data Set
Creating the Exprev Data Set
Creating the Gov Data Set
Creating the Grain_Production Data Set
Creating the One Data Set
Creating the Iron Data Set
Creating the Model Data Set
Creating the Neuralgia Data Set
Creating the Plants Data Set
Creating the Plant_Stats Data Set
Creating the StatePop Data Set
Creating the Stats and Stats2 Data Sets
Creating the Table1 Table Template
Programs That Illustrate Inheritance

Appendix 3 • ODS and the HTML Destination
HTML Links and References Produced by the HTML Destination
Files Produced by the HTML Destination

Appendix 4 • ODS HTML Statements for Running Examples in Different Operating Environments
Using a z/OS UNIX System Services HFS Directory for HTML Output
Using a z/OS PDSE for EBCDIC HTML Output
Using a z/OS PDSE for ASCII HTML Output

Appendix 5 • PowerPoint Transition Options and the Effect Options Supported
PowerPoint Transition Options and the Effect Options Supported

Recommended Reading
About This Book

Lua License


Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Syntax Conventions for the SAS Language

Overview of Syntax Conventions for the SAS Language

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

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

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

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

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

CHAR (string, position)
CALL RANBIN (seed, n, p, x);
ALTER (alter-password)
BEST w.
REMOVE <data-set-name>

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

CALL RANBIN(seed, n, p, x)

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

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

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

DUPLEX | NODUPLEX

Some procedure statements have multiple keywords throughout the statement syntax:

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

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

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

CHAR (string, position)

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

x=char('summer', 4);

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

FIND(string, substring <, modifiers> <, startpos>
argument(s)
specifies that one argument is required and that multiple arguments are allowed.
Separate arguments with a space. Punctuation, such as a comma ( , ) is not required
between arguments.

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

MISSING character(s);

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

The BY statement is an example of this argument:

BY <DESCENDING> variable-1 <<DESCENDING> variable-2 ...>;

argument-1 <option(s)> <argument-2 <option(s)> ...>
specifies that one argument is required and that one or more options can be
associated with the argument. You can specify multiple arguments and associated
options. No punctuation is required between the argument and the option. The ellipsis (...) indicates that additional arguments with an associated option are allowed.

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

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

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

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

LABEL variable-1=label-1 <variable-2=label-2 ...>;

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

Here are examples of this form of multiple arguments:

AUTHPROVIDERDOMAIN (provider-1:domain-1 <, provider-2:domain-2, ...>
INTO:macro-variable-specification-1 <, macro-variable-specification-2, ...>

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

Style Conventions

The style conventions that are used in documenting SAS syntax include uppercase bold,
uppercase, and italic:
**UPPERCASE BOLD**
identifies SAS keywords such as the names of functions or statements. In this example, the keyword ERROR is written in uppercase bold:

```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:

  ```sas
  CAT (item-1 <, item-2, ...>)
  ```

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

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

  ```sas
  CMPMODEL=BOTH | CATALOG | XML
  ```

- `...` an ellipsis indicates that the argument can be repeated. If an argument and the ellipsis are enclosed in angle brackets, then the argument is optional. The repeated argument must contain punctuation if it appears before or after the argument.
In this example of the CAT function, multiple item arguments are allowed, and they must be separated by a comma:

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

'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 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:

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

**References to SAS Libraries and External Files**

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

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

\[
\text{infile file-specification obs = 100;}
\]
\[
\text{libname libref 'SAS-library';}
\]
What's New in the Output Delivery System

Overview

The following enhancements have been made to the Output Delivery System:

- Enhancements have been made to the DOCUMENT procedure.
- There are three new ODS procedures: PROC ODSLIST, PROC ODSTABLE, and PROC ODSTEXT.
- The Report Writing Interface is now in production.
- In the first maintenance release of SAS 9.4, you can apply cascading style sheets (CSS) to many types of ODS output.
- The default style for all Printer family destinations (PS, PDF, PCL) has changed from Styles.Printer to Styles.Pearl.
- The default HTML style for batch output has changed from Styles.Default to Styles.Htmlblue.
- Enhancements have been made to the ODS statements.
- There are several enhancements and new options for ODS Graphics.
- There are new ODS statements: ODS POWERPOINT, ODS EPUB, ODS LAYOUT, ODS HTML5, and ODS EXCEL. In the first maintenance release of SAS 9.4, new statement ODS EPUB3 was added. In the third maintenance release of SAS 9.4, the ODS EXCEL statement was added.
- In the third maintenance release of SAS 9.4, HTML5 now supports video and audio when using the Report Writing Interface.
- There are several new options for ODS HTML and HTML5.
- Enhancements have been made to the documentation.

Enhancements to the DOCUMENT Procedure

The STORE= option is new. The STORE= option enables you to specify a template store when replaying output objects from the path. If the replayed path is an output object, the template store applies only to that output object. If the path is a directory, the template store applies to any output objects contained within the directory.
New ODS Procedures

The following procedures are new of SAS 9.4.

**ODSLIST Procedure**

The ODSLIST procedure enables you to create text content rather than tables. PROC ODSLIST creates lists that can be customized.

**ODSTABLE Procedure**

The ODSTABLE procedure enables you to create table templates and bind them with the input data set in one statement. You can also name your templates and store them in a template store.

**ODSTEXT Procedure**

The ODSTEXT procedure enables you to create text content rather than the usual SAS output. PROC ODSTEXT creates lists and paragraphs for your output.

**MSCHART Procedure (Preproduction)**

The MSCHART procedure creates charts that can be opened and manipulated in Microsoft Excel.

The MSCHART procedure is a new feature in the third maintenance release of SAS 9.4.

---

The Report Writing Interface

The Report Writing Interface (RWI) enables you to create and manipulate predefined ODS objects in a DATA step to create highly customized output.

In the third maintenance release of SAS 9.4, RWI has two new methods that support the ability to embed audio and video into HTML5 output. See “AUDIO Method” in SAS Output Delivery System: Advanced Topics and “VIDEO Method” in SAS Output Delivery System: Advanced Topics for information about these RWI methods.

---

ODS and Cascading Style Sheets

In the first maintenance release of SAS 9.4, you can apply cascading style sheets to ODS output destinations HTML, PDF, and RTF.
The default style for all Printer family destinations has changed from Printer to Pearl. The following PDF output uses the new default style, Pearl:

<table>
<thead>
<tr>
<th>Region by Division and Type</th>
<th>Product type</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FURNITURE</td>
<td>OFFICE</td>
<td>Total</td>
</tr>
<tr>
<td></td>
<td>Actual Sales</td>
<td></td>
<td></td>
<td></td>
</tr>
<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>Division</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONSUMER</td>
<td>$76,209</td>
<td>$105,020</td>
<td>$181,229</td>
</tr>
<tr>
<td></td>
<td>EDUCATION</td>
<td>$67,945</td>
<td>$110,902</td>
<td>$178,847</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>$144,154</td>
<td>$215,922</td>
<td>$360,076</td>
</tr>
<tr>
<td>Total</td>
<td>Division</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONSUMER</td>
<td>$148,779</td>
<td>$213,706</td>
<td>$362,485</td>
</tr>
<tr>
<td></td>
<td>EDUCATION</td>
<td>$141,846</td>
<td>$226,006</td>
<td>$367,852</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>$290,625</td>
<td>$439,712</td>
<td>$730,337</td>
</tr>
</tbody>
</table>
New Default Style for Batch Output

The default HTML style for batch output on all hosts has changed from Default to HTMLBlue. “Working with Output Defaults” on page 21 provides detailed information for changing these defaults.

ODS Statements Enhancements

In the first maintenance release of SAS 9.4, the DATASKINMAX= option is new for the ODS GRAPHICS statement. The DATASKINMAX= option specifies the maximum number of skinned graphical elements in plots.

The new BOX_SIZING option specifies how to measure the width of cells. The BOX_SIZING option is new for the following destinations:

- EPUB
- HTML4
- HTML5
- PHTML
- PRINTER
- PCL
- PDF
- PS
- RTF
- TAGSETS.RTF

In the first maintenance release of SAS 9.4, the new DOM option specifies that the ODS document object model is written to the SAS log or an external file. The DOM option is new for the following ODS destinations:

- EPUB
- EPUB3
- MARKUP family
- PRINTER family
- RTF

In the first maintenance release of SAS 9.4, the CSSSTYLE= option was added. The CSSSTYLE= option specifies a cascading style sheet to apply to your output. The CSSSTYLE= option is new for the following ODS destinations:

- EPUB
- EPUB3
- MARKUP family
In the second maintenance release of SAS 9.4, the ODS ESCAPECHAR statement now supports the following functions that are used with the ODS EPUB and ODS EPUB3 statements:

- **BOLD** (p. 276)
  used to emphasize keywords and semantics. This function provides no voice or mood change. This function is used with Assistive Technology.

- **EMPHASIS** (p. 277)
  used for vocal stress. This function is used with Assistive Technology.

- **ITALIC** (p. 278)
  used to indicate an alternative voice or mood. This function is also used for semantics like identifying names. This function is used with Assistive Technology.

- **PDF** (p. 280)
  used to embed a PDF document in an EPUB document.

- **STRONG** (p. 281)
  used to convey importance. This function is used with Assistive Technology.

In the third maintenance release of SAS 9.4, the ODS ESCAPECHAR statement now supports the following functions that are used with the ODS EPUB3 statements:

- **AUDIO** (p. 276)
  used to embed audio.

- **IMAGE** (p. 277)
  used to embed images.

- **MATHML** (p. 278)
  supports accessibility features for MathML like large print with the ability to drill down on terms for low vision users, voiceover that speaks the math, and voiceover that displays the math on a refreshable braille display using the Nemeth braille code. This function is used with Assistive Technology.

- **NOTEREF** (p. 280)
  can be used to create pop-up footnotes.

- **VIDEO** (p. 281)
  used to embed video.

The ODS GRAPHICS statement now supports the following options:

- **ATTRPRIORITY**= (p. 331)
  specifies a priority for cycling of the group attributes.

- **BYLINE**= (p. 332)
  specifies how the BY line is displayed in graphs when an analysis is run with a BY statement. By default, no BY line is displayed.

- **DRILLTARGET**= (p. 334)
  specifies the window that the drill-down output is displayed in.
LOESSMAXOBS= (p. 337) specifies an upper limit for the number of observations that can be used with a loess plot.

The following enhancements have been made to the ODS GRAPHICS statement:

• The default for the ANTIALIASMAX= option has changed from 600 to 4000.

• The default device driver for the RTF and TAGSETS.RTF destination has changed to EMF.

• The OUTPUTFMT= option now supports PS, EPS, and EPSI for the ODS LISTING destination.

Starting with the third maintenance release of SAS 9.4, the ODS GRAPHICS statement supports the following options:

PUSH and POP (p. 338) enable you to save (PUSH) your custom settings in a stack and later restore (POP) those settings.

STACKDEPTHMAX= (p. 343) specifies the maximum size of the stack. You can also use this option to empty the stack.

SHOW (p. 342) writes the current ODS GRAPHICS settings to the SAS log.

SUBPIXEL | NOSUBPIXEL (p. 344) specifies whether subpixel rendering should be used for rendering ODS Graphics.

The following changes and enhancements were made to the ODS GRAPHICS statement in the third maintenance release of SAS 9.4:

• You can reset the IMAGENAME and OUTPUTFMT options using the RESET= option. Previously, these options were not included with the RESET= option.

• When you specify the RESET=INDEX option, you can also specify a starting value for the index counter. The new syntax is RESET=INDEX<(positive-integer)>.

• The LEGENDAREAMAX= option replaces MAXLEGENDAREA=. However, MAXLEGENDAREA= is supported as an alias. It is recommended that you use LEGENDAREAMAX=.

• The LOESSOBSMAX= option replaces LOESSMAXOBS=. However, LOESSMAXOBS= is supported as an alias. It is recommended that you use LOESSOBSMAX=.

ODS TAGSETS.RTF Statement

The ODS TAGSETS.RTF statement supports the following new options:

• The BACKGROUNDIMAGE= style attribute can be used to add a watermark image to the RTF output.

• The new OPTIONS (WATERMARK=) enables you to add watermark text to each page of the RTF output.

• The new OPTIONS (VSPACE= ) honors space requests in the document that normally occur before and after tables.
New ODS Statements

**ODS EPUB**

The **ODS EPUB statement** is used to create e-books. The ODS EPUB statement generates output with the .epub extension. E-books that use the .epub format can be read by a wide variety of e-book readers.

In the **second maintenance release** of SAS 9.4, the ODS ESCAPECHAR statement supports the following functions that are used with the ODS EPUB and ODS EPUB3 statements:

- **BOLD** (p. 276)
  - used to emphasize keywords and semantics. This function provides no voice or mood change. This function is used with Assistive Technology.

- **EMPHASIS** (p. 277)
  - used for vocal stress. This function is used with Assistive Technology.

- **ITALIC** (p. 278)
  - used to indicate an alternative voice or mood. This function is also used for semantics like identifying names. This function is used with Assistive Technology.

- **PDF** (p. 280)
  - used to embed a PDF document in an EPUB document.

- **STRONG** (p. 281)
  - used to convey importance. This function is used with Assistive Technology.

In the **third maintenance release** of SAS 9.4, encoding is specified for ODS EPUB, ODS EPUB2, and ODS EPUB3 by using the Registry commands. See “**Changing the ODS EPUB and EPUB3 Encoding Settings**” on page 49.

**ODS EPUB3**

In the first maintenance release of SAS 9.4, the ODS EPUB3 statement was added. In addition to what ODS EPUB supports, the **ODS EPUB3 statement** supports HTML5, CSS2 and CSS3, and SVG. ODS EPUB 3 supports user-specified audio and video. Any e-book reader that supports EPUB 3 supports ODS EPUB3 audio and video.

In the **second maintenance release** of SAS 9.4, the ODS ESCAPECHAR statement supports the following functions that are used with the ODS EPUB and ODS EPUB3 statements:

- **BOLD** (p. 276)
  - used to emphasize keywords and semantics. This function provides no voice or mood change. This function is used with Assistive Technology.

- **EMPHASIS** (p. 277)
  - used for vocal stress. This function is used with Assistive Technology.

- **ITALIC** (p. 278)
  - used to indicate an alternative voice or mood. This function is also used for semantics like identifying names. This function is used with Assistive Technology.
PDF (p. 280)
used to embed a PDF document in an EPUB document.

STRONG (p. 281)
used to convey importance. This function is used with Assistive Technology.

In the third maintenance release of SAS 9.4, the ODS ESCAPECHAR statement supports the following functions that are used with the ODS EPUB3 statements:

AUDIO (p. 276)
used to embed audio.

IMAGE (p. 277)
used to embed images.

MATHML (p. 278)
supports accessibility features like large print with the ability to drill down on terms for low vision users, voiceover that speaks the math, and voiceover that displays the math on a refreshable braille display using the Nemeth braille code. This function is used with Assistive Technology.

NOTEREF (p. 280)
can be used to create pop-up footnotes.

VIDEO (p. 281)
used to embed images.

In the third maintenance release of SAS 9.4, the ODS ESCAPECHAR statement supports the following functions that are used with the ODS EPUB3 statements:

AUDIO (p. 276)
used to embed audio.

IMAGE (p. 277)
used to embed images.

MATHML (p. 278)
supports accessibility features like large print with the ability to drill down on terms for low vision users, voiceover that speaks the math, and voiceover that displays the math on a refreshable braille display using the Nemeth braille code. This function is used with Assistive Technology.

NOTEREF (p. 280)
can be used to create pop-up footnotes.

VIDEO (p. 281)
used to embed images.

In the third maintenance release of SAS 9.4, the ODS EPUB3 EVENTS= option supports the following new suboptions.

BRANCH (p. 238)
creates custom TOC entries.

FIGCAPTION (p. 239)
adds text to the figure. ODS EPUB3 implements FIGCAPTION as an event. The FIGCAPTION event accepts TEXT, which is the text to be displayed above or below the content.

FIGURE (p. 239)
contains a single image. The default Daisy style element for a figure is "figure". ODS EPUB3 implements FIGURE as an event.
In the third maintenance release of SAS 9.4, the default ODS EPUB destination is ODS EPUB3.

**ODS HTML5**

The **ODS HTML5** statement is used to create HTML output using the 5.0 version of HTML.

In the second maintenance release of SAS 9.4, the ODS HTML5 statement supports the following new options:

- **SHOW_GRAPH_STYLES=** option specifies that the output should contain elements from the graph style that is specified.
- **USE_CSS_RESET=** option turns off the default CSS styles information.

In the third maintenance release of SAS 9.4, HTML5 supports the ability to embed video and audio by using the Report Writing Interface AUDIO and VIDEO methods. See “AUDIO Method” in *SAS Output Delivery System: Advanced Topics* and “VIDEO Method” in *SAS Output Delivery System: Advanced Topics* for information about the audio and video RWI methods used by ODS HTML5.

In the third maintenance release of SAS 9.4, the ODS HTML5 statement supports the following new options:

- **SGE** option turns on the ability to generates a file that can be edited only with the ODS Graphics Editor. The file created has an extension of .sge.

**ODS LAYOUT**

ODS LAYOUT statements are used to place output on page. The ODS LAYOUT statement enables you to arrange output in an absolute location (absolute) or dynamically (gridded). Here are some details about the two types of layouts:

**Absolute Layout** (p. 454)

Absolute Layout enables you to specify an exact page location using x and y coordinates. Absolute layout is limited to one page and is supported only by the Printer destinations.

**Gridded Layout** (p. 468)

Gridded layout is a mechanism for arranging output dynamically. Gridded layout enables you to arrange output in a two-dimensional grid structure. For example, gridded layout enforces automatic alignment of grid cells, allows layout to continue onto the next page, dynamically computes the size of a grid cell, and makes it easier to maintain the integrity of the report. Gridded layout is supported for HTML, POWERPOINT, and PRINTER destinations (PDF, PS, and PCL).

**ODS POWERPOINT**

The **ODS POWERPOINT** statement is used to create PowerPoint slides. Slides can contain text, graphics, and tables.

ODS POWERPOINT supports gridded layout. Three pre-defined gridded layouts templates are supported. These gridded layouts correspond to the similarly named built-in PowerPoint layouts: TitleSlide, TitleAndContent, and TwoContent.
In the third maintenance release of SAS 9.4, the ODS POWERPOINT statement enhances the LAYOUT=TwoContent to include suboption ADVANCE=. The ADVANCE= suboption specifies that the grid is explicitly populated or that the grid is dynamically populated by groups, tables, pages, and procedures. This option works the same as the ADVANCE= option on the ODS LAYOUT GRIDDED statement. See “TwoContent (ADVANCE=options)” on page 627.

In the third maintenance release of SAS 9.4, the ODS POWERPOINT statement now includes the OPTIONS option. Supported are style options like backgroundcolor, backgroundborder, and backgroundrepeat that can change the body style. There are transitions and their effects. There are also options to adjust slide duration, specify how long before advancing a slide, and specify whether to advance a slide when clicking a mouse. You can play a .wav file when transitioning between slides. See “OPTIONS (ADVANCE_AFTER= | ADVANCE_ON_MOUSE_CLICK= | BACKGROUNDCOLOR= | BACKGROUNDIMAGE= | BACKGROUNDREPEAT= | DURATION= | EFFECT_OPTION= | SOUND= | TRANSITION= )” on page 628.

### Deprecated ODS Tagsets

Starting with the second maintenance release of SAS 9.4, the following ODS tagsets have been deprecated:

- DOCBOOK
- HTMLCSS
- IMODE
- MVSHTML
- PYX
- SASREPORT family of tagsets
- TPL_STYLE_LIST
- TRPL_STYLE_MAP
- WML
- WMLOLIST
- XHTML

### Documentation Enhancements

Starting with the second maintenance release of SAS 9.4, *SAS Output Delivery System: User’s Guide* has been divided into 3 documents:

- *SAS Output Delivery System: Procedures Guide*
- *SAS Output Delivery System: Advanced Topics*
Procedures Guide

Syntax and information for ODS procedures is no longer part of the SAS Output Delivery System: User's Guide. ODS procedures that were previously documented here are now documented in SAS Output Delivery System: Procedures Guide.

Advanced Topics

More advanced topics, such as the report writing interface and cascading style sheets, are now covered in SAS Output Delivery System: Advanced Topics. They are no longer covered in SAS Output Delivery System: User's Guide.

User's Guide

In addition to the document being restructured, the section "ODS Styles Reference" has been added to SAS 9.4 Output Delivery System: User’s Guide. This section is intended to help you find ODS style-related information. The section contains the following chapters:

ODS Style Templates (p. 923)

The ODS Styles chapter contains a table of the recommended and default styles for each destination. The appendix also contains a gallery with examples of each style and a program that enables you to create a gallery of your own.

ODS Style Attributes (p. 993)

The ODS Style Attributes chapter contains a table of all available style attributes and their full descriptions.

ODS Style Elements (p. 965)

The ODS Style Attributes chapter contains a table of all available style elements.
For information about the accessibility of this product, see Accessibility Features of the Windowing Environment for SAS 9.4 at support.sas.com.
Part 1

Introduction

Chapter 1
Getting Started with the Output Delivery System ................. 3
Welcome to the Output Delivery System

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 Chapter 6, “Dictionary of ODS Language Statements,” on page 105.

Customize the report contents.

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

Customize the presentation.

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

Create more accessible SAS output.

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

---

**Gallery of ODS Samples**

*Introduction to the ODS Samples*

This section shows you samples of the different types of formatted output that you can produce with ODS. For the complete programs that create these samples, see “Examples from the Gallery of ODS Samples” on page 1068.

*EPUB Output*

With the ODS EPUB statement, you can create an e-book for use with e-book readers. iBooks on iPad is the preferred e-book reader. Results can vary using other e-book readers. See “ODS EPUB Statement” on page 211 for more information.
Excel Output

With ODS, you can produce tabular output, which can be viewed with Excel.
Graphical Output

The ODS Graph Template Language (GTL) applies accepted principles of graphics design to produce plots that are clean and uncluttered. For more information about ODS graphics, see “TEMPLATE Procedure: Creating ODS Graphics” in SAS Output Delivery System: Procedures Guide.
Figure 1.5  PROC SGSCATTER (SAS) with LISTING Style

Scatterplot Matrix for Iris Data

- Iris Species: Setosa, Versicolor, Virginica
**HTML Output**

With ODS, you can produce output in HTML (HyperText Markup Language.) In the SAS Windowing Environment, ODS is created by default. You can browse these files with Internet Explorer or any other browser that fully supports HTML 4.0. In the SAS Windowing Environment, you do not need to change your SAS programs to create HTML 4.0 output that contains embedded style sheets. SAS continues to create HTML even if you open a different destination, unless you specify the ODS HTML CLOSE statement. If you want to add more formatting to your HTML, use the ODS HTML statement.

*Note:* To create HTML 3.2 output, use the ODS HTML3 statement.
### Figure 1.7 HTML Output Viewed with Microsoft Internet Explorer

<table>
<thead>
<tr>
<th>Region by Citysize by Saletype</th>
<th>All Products</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Saletype</td>
</tr>
<tr>
<td></td>
<td>Retail</td>
</tr>
<tr>
<td></td>
<td>Quantity</td>
</tr>
<tr>
<td></td>
<td>Sum</td>
</tr>
<tr>
<td>Brazil</td>
<td>Citysize</td>
</tr>
<tr>
<td>L</td>
<td>Missing</td>
</tr>
<tr>
<td>M</td>
<td>1,968</td>
</tr>
<tr>
<td>S</td>
<td>472</td>
</tr>
<tr>
<td>Total</td>
<td>1,538</td>
</tr>
<tr>
<td>Canada</td>
<td>Citysize</td>
</tr>
<tr>
<td>L</td>
<td>2,421</td>
</tr>
<tr>
<td>M</td>
<td>1,825</td>
</tr>
<tr>
<td>S</td>
<td>623</td>
</tr>
<tr>
<td>Total</td>
<td>4,869</td>
</tr>
<tr>
<td>France</td>
<td>Citysize</td>
</tr>
<tr>
<td>L</td>
<td>2,303</td>
</tr>
<tr>
<td>M</td>
<td>2,149</td>
</tr>
<tr>
<td>S</td>
<td>1,254</td>
</tr>
<tr>
<td>Total</td>
<td>5,706</td>
</tr>
<tr>
<td>Mexico</td>
<td>Citysize</td>
</tr>
<tr>
<td>L</td>
<td>2,655</td>
</tr>
<tr>
<td>M</td>
<td>2,360</td>
</tr>
<tr>
<td>S</td>
<td>561</td>
</tr>
<tr>
<td>Total</td>
<td>5,676</td>
</tr>
<tr>
<td>Total</td>
<td>Citysize</td>
</tr>
<tr>
<td>L</td>
<td>7,378</td>
</tr>
<tr>
<td>M</td>
<td>7,400</td>
</tr>
<tr>
<td>S</td>
<td>2,910</td>
</tr>
<tr>
<td>Total</td>
<td>17,689</td>
</tr>
</tbody>
</table>

### PDF Output

With ODS, you can produce output in PDF (Portable Document Format), which can be viewed with Adobe Acrobat.
With ODS, you can produce output in PostScript format.

PostScript Output

With ODS, you can produce output in PostScript format.
**PowerPoint Output**

With the ODS POWERPOINT statement, you can create slides for Microsoft PowerPoint. See “ODS POWERPOINT Statement” on page 623 for more information.

**Figure 1.10**  PowerPoint Slide Using the Titleslide Layout

**Figure 1.11**  Slide 2 - The Types of Output Produced by the ODS Destination for PowerPoint
With ODS, you can produce RTF (Rich Text Format) output, which is used with Microsoft Word.
The Purpose of These Examples

The following examples are designed to help you begin using ODS quickly. Use them to learn how to produce output that contains more interesting formats.

With ODS, it is quick and easy to create output that is formatted for different business applications using ODS destination statements. The ODS statement and the SAS program that it contains form the ODS block.

An ODS block has the following form:

```
ODS output-destination 1 <options(s)>;
...
ODS output-destination (n) <options(s>)
<your SAS program>
ODS destination close statement 1;
...
ODS destination close statement (n)
```

In the ODS block, `output-destination` is the name of a valid ODS destination and `option(s)` are options that are valid for that destination. Your SAS program is inserted between the beginning ODS destination statement and the ODS CLOSE statement.
The FILE= or BODY= option is valid for most ODS statements. These options specify the name and path of the file that you are generating. It is a good practice to specify one of these options, but it is not required. In the SAS Windowing Environment, 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 output is opened by the Results Viewer. HTML and EXCEL output can be viewed within the Results Viewer window. PDF output opens in Adobe. RTF output opens in Microsoft Word. PowerPoint output opens in PowerPoint. The EPUB output must be opened with an e-book reading device.

You can find additional tips and tricks in the SAS press book excerpt, ODS Techniques: Tips for Enhancing Your SAS Output, by Kevin Smith. This SAS press book is a cookbook-style collection of Kevin’s top ODS tips and techniques to teach you how to bring your reports to a new level and inspire you to see ODS in a new light.

**Creating HTML Output**

Creating HTML output within an interactive SAS session is simple, and does not require an ODS statement. Run a DATA step or PROC step as usual. In the SAS Windowing environment, the HTML destination is on, and the DATA step and SAS procedures create HTML output through ODS. You can browse this output with any browser that fully supports HTML. You can choose which output to view in the Results window.

data employee_data;
  input IDNumber $ 1-4 LastName $ 9-19 FirstName $ 20-29 City $ 30-42 State $ 43-44 /
     Gender $ 1 JobCode $ 9-11 Salary 20-29 @30 Birth date9. @43 Hired date9. HomePhone $ 54-65;
  format birth hired date9.;
datalines;
1919 Adams Gerald Stamford CT
1653 Alexander Susan Bridgeport CT
1400 Apple Troy New York NY
1350 Arthur Barbara New York NY
1401 Avery Jerry Paterson NJ
1499 Barefoot Joseph Princeton NJ
1101 Baucom Walter New York NY
1333 Blair Justin Stamford CT
1402 Blalock Ralph New York NY
1479 Bostic Marie New York NY
1403 Bowden Earl Bridgeport CT
A Quick Start to Using ODS 15

proc print data=employee_data(obs=12);
    id idnumber;
    title 'Personnel Data';
run;

HTML output is the default destination. Therefore, when you request another
destination, your programs create both HTML output and output in the new destination.
To prevent HTML output from being created, use this statement:
ods html close;

Output 1.1 Default HTML Output

Creating LISTING Output

You can create traditional SAS output with monospace fonts (LISTING output) by using
the ODS LISTING statement.

The following program contains a PROC PRINT step that produces LISTING output,
but does not produce the default HTML output.

ods html close;
ods listing;

proc print data=employee_data(obs=12);
    id idnumber;
    title 'Personnel Data';
    run;

ods listing close;
ods html;

Note the two ODS statements that follow the PROC PRINT step. To be able to view your LISTING output, you must execute the ODS LISTING CLOSE statement. It is simply good practice to reset ODS to HTML output, which is the default setting.

Output 1.2 LISTING Output

<table>
<thead>
<tr>
<th>IDNumber</th>
<th>LastName</th>
<th>Name</th>
<th>City</th>
<th>State</th>
<th>Gender</th>
<th>Job Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1919</td>
<td>Adams</td>
<td>Gerald</td>
<td>Stamford</td>
<td>CT</td>
<td>M</td>
<td>TA2</td>
</tr>
<tr>
<td>1653</td>
<td>Alexander</td>
<td>Susan</td>
<td>Bridgeport</td>
<td>CT</td>
<td>F</td>
<td>ME2</td>
</tr>
<tr>
<td>1400</td>
<td>Apple</td>
<td>Troy</td>
<td>New York</td>
<td>NY</td>
<td>M</td>
<td>ME1</td>
</tr>
<tr>
<td>1350</td>
<td>Arthur</td>
<td>Barbara</td>
<td>New York</td>
<td>NY</td>
<td>F</td>
<td>PA3</td>
</tr>
<tr>
<td>1401</td>
<td>Avery</td>
<td>Jerry</td>
<td>Paterson</td>
<td>NJ</td>
<td>M</td>
<td>TA3</td>
</tr>
<tr>
<td>1499</td>
<td>Barefoot</td>
<td>Joseph</td>
<td>Princeton</td>
<td>NJ</td>
<td>M</td>
<td>ME3</td>
</tr>
<tr>
<td>1101</td>
<td>Baucom</td>
<td>Walter</td>
<td>New York</td>
<td>NY</td>
<td>M</td>
<td>SCP</td>
</tr>
<tr>
<td>1333</td>
<td>Blair</td>
<td>Justin</td>
<td>Stamford</td>
<td>CT</td>
<td>M</td>
<td>PT2</td>
</tr>
<tr>
<td>1402</td>
<td>Blalock</td>
<td>Ralph</td>
<td>New York</td>
<td>NY</td>
<td>M</td>
<td>TA2</td>
</tr>
<tr>
<td>1479</td>
<td>Bostic</td>
<td>Marie</td>
<td>New York</td>
<td>NY</td>
<td>F</td>
<td>TA3</td>
</tr>
<tr>
<td>1403</td>
<td>Bowden</td>
<td>Earl</td>
<td>Bridgeport</td>
<td>CT</td>
<td>M</td>
<td>ME1</td>
</tr>
<tr>
<td>1739</td>
<td>Boyce</td>
<td>Jonathan</td>
<td>New York</td>
<td>NY</td>
<td>M</td>
<td>PT1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IDNumber</th>
<th>Salary</th>
<th>Birth</th>
<th>Hired</th>
<th>HomePhone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1919</td>
<td>34376</td>
<td>15SEP1948</td>
<td>07JUN1975</td>
<td>203/781-1255</td>
</tr>
<tr>
<td>1653</td>
<td>35108</td>
<td>18OCT1952</td>
<td>12AUG1978</td>
<td>203/675-7715</td>
</tr>
<tr>
<td>1400</td>
<td>29769</td>
<td>08NOV1955</td>
<td>19OCT1978</td>
<td>212/586-0808</td>
</tr>
<tr>
<td>1350</td>
<td>32886</td>
<td>01SEP1953</td>
<td>01AUG1978</td>
<td>718/383-1549</td>
</tr>
<tr>
<td>1401</td>
<td>38822</td>
<td>16DEC1938</td>
<td>20NOV1973</td>
<td>201/732-8787</td>
</tr>
<tr>
<td>1499</td>
<td>43025</td>
<td>29APR1942</td>
<td>10JUN1968</td>
<td>201/812-5665</td>
</tr>
<tr>
<td>1101</td>
<td>38723</td>
<td>09JUN1950</td>
<td>04OCT1978</td>
<td>212/586-8060</td>
</tr>
<tr>
<td>1333</td>
<td>88606</td>
<td>02APR1949</td>
<td>13FEB1969</td>
<td>203/781-1777</td>
</tr>
<tr>
<td>1402</td>
<td>32615</td>
<td>20JAN1951</td>
<td>05DEC1978</td>
<td>718/384-2849</td>
</tr>
<tr>
<td>1479</td>
<td>38785</td>
<td>25DEC1956</td>
<td>08OCT1977</td>
<td>718/384-8816</td>
</tr>
<tr>
<td>1403</td>
<td>28072</td>
<td>31JAN1957</td>
<td>24DEC1979</td>
<td>203/675-3434</td>
</tr>
<tr>
<td>1739</td>
<td>66517</td>
<td>28DEC1952</td>
<td>30JAN1979</td>
<td>212/587-1247</td>
</tr>
</tbody>
</table>

Producing Output in Multiple Destinations at the Same Time

A simple way to produce output in multiple destinations at one time is to produce the default HTML output and then request an additional destination, such as PDF, RTF, or PostScript.

options nodate;
ods html file='HTML-file-pathname.html';
ods pdf file='PDF-file-pathname.pdf';
ods rtf file='RTF-file-pathname.rtf';
proc print data=sashelp.cars(obs=12);
  var model msrp invoice;
  id Make;
run;
ods _all_ close;
ods html;

Note the two ODS statements that follow the PROC statement. The first statement closes all destinations so that you can view the output files (for example, you could browse the HTML file or send the PDF file to a printer). The final statement opens the HTML destination so that ODS returns to the default settings.

The following output is formatted the default HTML, HTML4.

Figure 1.15  HTML4 Output

<table>
<thead>
<tr>
<th>Make</th>
<th>Model</th>
<th>MSRP</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acura</td>
<td>MDX</td>
<td>$36,945</td>
<td>$33,337</td>
</tr>
<tr>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>$23,820</td>
<td>$21,761</td>
</tr>
<tr>
<td>Acura</td>
<td>TSX 4dr</td>
<td>$26,990</td>
<td>$24,647</td>
</tr>
<tr>
<td>Acura</td>
<td>TL 4dr</td>
<td>$33,195</td>
<td>$30,299</td>
</tr>
<tr>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>$43,755</td>
<td>$39,014</td>
</tr>
<tr>
<td>Acura</td>
<td>3.5 RL w/Navigation 4dr</td>
<td>$46,100</td>
<td>$41,100</td>
</tr>
<tr>
<td>Acura</td>
<td>NSX coupe 2dr manual S</td>
<td>$89,765</td>
<td>$79,978</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>$25,940</td>
<td>$23,508</td>
</tr>
<tr>
<td>Audi</td>
<td>A41.8T convertible 2dr</td>
<td>$35,940</td>
<td>$32,506</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>$31,840</td>
<td>$28,846</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr manual</td>
<td>$33,430</td>
<td>$30,366</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr auto</td>
<td>$34,480</td>
<td>$31,388</td>
</tr>
</tbody>
</table>
The following output is formatted in PDF and viewed in Adobe Acrobat Reader.

**Figure 1.16  PDF Output**

### Car Pricing

<table>
<thead>
<tr>
<th>Make</th>
<th>Model</th>
<th>MSRP</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acura</td>
<td>MDX</td>
<td>$36,945</td>
<td>$33,337</td>
</tr>
<tr>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>$23,820</td>
<td>$21,761</td>
</tr>
<tr>
<td>Acura</td>
<td>TSX 4dr</td>
<td>$26,990</td>
<td>$24,647</td>
</tr>
<tr>
<td>Acura</td>
<td>TL 4dr</td>
<td>$33,195</td>
<td>$30,299</td>
</tr>
<tr>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>$43,755</td>
<td>$39,014</td>
</tr>
<tr>
<td>Acura</td>
<td>3.5 RL w/Navigation 4dr</td>
<td>$46,100</td>
<td>$41,100</td>
</tr>
<tr>
<td>Acura</td>
<td>NSX coupe 2dr manual S</td>
<td>$89,765</td>
<td>$79,978</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>$25,940</td>
<td>$23,508</td>
</tr>
<tr>
<td>Audi</td>
<td>A41.8T convertible 2dr</td>
<td>$35,940</td>
<td>$32,506</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>$31,840</td>
<td>$28,846</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr manual</td>
<td>$33,430</td>
<td>$30,366</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr auto</td>
<td>$34,480</td>
<td>$31,388</td>
</tr>
</tbody>
</table>
The following RTF output is viewed with Microsoft Word.

**Figure 1.17  RTF Output**

---

### Car Pricing

<table>
<thead>
<tr>
<th>Make</th>
<th>Model</th>
<th>MSRP</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acura</td>
<td>MDX</td>
<td>$36,945</td>
<td>$33,337</td>
</tr>
<tr>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>$23,820</td>
<td>$21,761</td>
</tr>
<tr>
<td>Acura</td>
<td>TSX 4dr</td>
<td>$26,990</td>
<td>$24,647</td>
</tr>
<tr>
<td>Acura</td>
<td>TL 4dr</td>
<td>$33,195</td>
<td>$30,299</td>
</tr>
<tr>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>$43,755</td>
<td>$39,014</td>
</tr>
<tr>
<td>Acura</td>
<td>3.5 RL w/Navigation 4dr</td>
<td>$46,100</td>
<td>$41,100</td>
</tr>
<tr>
<td>Acura</td>
<td>NSX coupe 2dr manual S</td>
<td>$89,765</td>
<td>$79,978</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>$25,940</td>
<td>$23,508</td>
</tr>
<tr>
<td>Audi</td>
<td>A41.8T convertible 2dr</td>
<td>$35,940</td>
<td>$32,506</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>$31,840</td>
<td>$28,846</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr manual</td>
<td>$33,430</td>
<td>$30,366</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr auto</td>
<td>$34,480</td>
<td>$31,388</td>
</tr>
</tbody>
</table>

---

**Creating Accessible Output with the Output Delivery System**

The following additional accessibility items are available as programming options to help you create more accessible output:
The following tagsets and ODS statements are suggested to help you create more accessible output:

**ODS EPUB and ODS EPUB3 Statements**
open, manage, or close the EPUB, EPUB2, or EPUB3 destinations, which generate EPUB e-books. The ODS EPUB and ODS EPUB3 destinations are the recommended destinations for creating SAS output that is accessible to the broadest audience. They create e-books that use many of the accessibility features of the EPUB specification. These features allow e-book readers such as iBooks to present e-books so that they adapt to the needs of users with disabilities. For example, when reading an e-book created by the ODS EPUB and ODS EPUB3 destinations using iBooks on an iPad, users can adjust font size, color schemes, and magnification. They can also access the text using assistive technologies such as the Voiceover screen reader and refreshable braille displays. For complete documentation about the ODS EPUB statement, see the “ODS EPUB Statement” on page 211. For complete documentation about the ODS EPUB3 statement, see the “ODS EPUB3 Statement” on page 233.

**ODS PHTML Statement**
opens, manages, or closes the PHTML destination, which produces simple HTML output that uses twelve style elements and no class attributes. For more information about the ODS PHTML statement, see the “ODS PHTML Statement” on page 663.

**ODS HTML Statement**
opens, manages, or closes the HTML destination, which produces HTML 4.0 output that contains embedded style sheets. For more information about the ODS HTML statement, see the “ODS HTML Statement” on page 110.

**MSOFFICE2K tagset**
produces HTML code for output generated by ODS for Microsoft Office products. For more information about the MSOFFICE2K tagset, see the “MSOFFICE2K” on page 776.

**Event Variables**
For details about the following event variables, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

- **ABBR**
  specifies an abbreviation for the event variable.

- **ACRONYM**
  specifies an acronym for an event variable.

- **ALT**
  specifies an alternate description of an event variable.

- **CAPTION**
  specifies the caption for a table.

- **LONGDESC**
  specifies the long description of an event variable.

- **SUMMARY**
  specifies a summary of a table.

**Style Template**
For information about the ODS Styles that SAS provides, see Chapter 10, “Style Templates,” on page 923.

**STYLES.HIGHCONTRAST**
creates the same output as the default output except all of the colors are black foreground on a white background.
Header Attributes
For details about the following header attributes, see “Header and Footer Attributes” in SAS Output Delivery System: Procedures Guide.

ABBR=
specifies an abbreviation for the header.

ACRONYM=
specifies an acronym for the header.

ALT=
specifies an alternate description of the header.

GENERIC
specifies whether multiple columns can use the header.

LONGDESC=
specifies the long description of the header.

In SAS 9.1 and later releases, all of the accessibility enhancements have been merged into the ODS HTML tagsets. No additional steps are required.

Working with Output Defaults

Overview
Beginning with SAS 9.3, a new set of defaults are in effect for SAS in the SAS Windowing environment in the Windows and UNIX operating systems:

• ODS Graphics is enabled at SAS start-up.
• The LISTING destination is closed and the HTML destination is open.
• The default style for the HTML destination is HTMLBlue.

ODS Graphics
Template-based graphics (frequently referred to as ODS Graphics) are created by default. ODS Graphics includes all graphical output where a compiled ODS template of type STATGRAPH is used to produce graphical output. Supplied templates are stored in Sashelp.Tmplmst. For ODS Graphics, you must use the ODS GRAPHICS statement to control the graphical environment. You do not have to specify the ODS GRAPHICS ON statement to enable ODS Graphics in the SAS Windowing environment in the Windows and UNIX operating systems.

Note: The SGSCATTER, SGRENDER, SGPLOT, and SGPANEL procedures always generate graphs, even when ODS Graphics is not enabled.

The Default Destination
By default, in the Windowing environment with the Windows and UNIX operating systems, the LISTING destination is closed and the HTML destination is open. You do not have to submit an ODS HTML statement to generate HTML output. You also do not have to use the ODS HTML CLOSE statement to view your output. However, to create LISTING output, you must either submit the ODS LISTING statement or enable the LISTING destination by other means. See “How to Restore 9.2 Behavior” on page 22.
The HTML destination does the following:

- generates HTML 4.0 and embedded style sheets
- writes output files to the Work directory
- does not require you to specify an ODS HTML CLOSE statement to view your output

These behaviors persist until you explicitly close the ODS HTML destination by specifying the ODS HTML CLOSE statement, and then reopen the HTML destination. After you have closed the HTML statement and issued a new ODS HTML statement, the HTML destination does the following:

- writes output files to the current directory
- does require you to specify an ODS HTML CLOSE statement to view your output

These behaviors persist until you close your SAS session and open a new one.

**CAUTION:**

As of SAS 9.3, HTML is the default destination and HTMLBlue is the default style for the HTML destination when using SAS in the windowing environment. As of SAS 9.4, HTMLBlue is also the default style for running SAS in batch mode and on z/OS. However, when you run SAS in batch mode or on z/OS, the LISTING destination is open and is the default and ODS Graphics is not enabled by default. Your actual defaults might be different because of your registry or configuration file settings.

### How to Restore 9.2 Behavior

**Overview**

You can change your output defaults back to 9.2 behavior in one of the following three ways:

- Use the **Results** tab in the Preferences window. This changes the behavior until you change it back.
- Use ODS statements. This change lasts only during your current SAS session.
- Use the ODSSTYLE, ODSDEST, and ODSGRAPHICS system options.

**Using the Preferences Window**

The default destination in the SAS windowing environment is HTML, and ODS Graphics is enabled by default. These new defaults have several advantages. Graphs are integrated with tables, and all output is displayed in the same HTML file using a new style. This new style, HTMLBlue, is an all-color style that is designed to integrate tables and modern statistical graphics.

**Note:** The default HTML is HTML4. Changes applied to HTML in the Preferences Window only affect the default HTML version, not HTML5 or other HTML versions.

You can view and modify the default settings by selecting **Tools** ➔ **Options** ➔ **Preferences** from the menu at the top of the main SAS window. Then open the Results tab. The settings in your Preferences window persist until you explicitly change them. The following display shows the **Results** tab with the new default settings specified.
To create LISTING output only by default, select **Create listing** and deselect **Create HTML**. To disable ODS Graphics, deselect “Use ODS Graphics”.

*Figure 1.18  Default Results Tab in the Preferences Window*

When the selections are set as shown in the Preferences window Results Tab, LISTING is the default output destination until you explicitly change it using ODS statements, the ODSDEST system option, or the Preferences window. By default, ODS Graphics are disabled until you enable ODS Graphics by specifying the ODS GRAPHICS ON
statement, use the ODSGRAPHICS system option, or change the setting in the Preferences window.

Figure 1.19  Results Tab Set to Pre-9.3 Defaults

Using ODS Statements
To change the default destination from HTML to LISTING and to disable ODS Graphics, you can use the following ODS statements:

```sas
ods graphics off;
ods html close;
ods listing;
```

These statements change the behavior of your current SAS session. When you start a new SAS session, the defaults return to SAS 9.4 behavior.

Using System Options
There are three system options that control default output.

ODSSTYLE=
specifies the default style. To change the default style to Styles.Default, specify `ODSSTYLE=styles.default`. For information about the ODSSTYLE= system option, see “ODSSTYLE= System Option” on page 909.

ODSGRAPHICS=
specifies whether ODS Graphics is enabled by default. To disable ODS Graphics by default, specify `ODSGRAPHICS=OFF`. For information about the ODSGRAPHICS= system option, see “ODSGRAPHICS= System Option” on page 908.

ODSDEST=
specifies the default output destination in the SAS windowing environment. To change the default destination to LISTING, specify `ODSDEST=LISTING`. For
Where to Go from Here

Getting Started with ODS

Getting Started with the SAS Output Delivery System

Examples of ODS output:
To see the types of output that you can create with ODS, see “Gallery of ODS Samples” on page 4.

Essential concepts in ODS:
For concepts that help you understand and use ODS to your best advantage, see Chapter 2, “Output Delivery System: Basic Concepts,” on page 29.

Creating more complex HTML pages:
With ODS, you can create HTML pages that include a frame and a table of contents. For more information, see “ODS HTML Statement” on page 110 and Appendix 3, “ODS and the HTML Destination,” on page 1101. You can see many examples of HTML output in Base SAS Procedures Guide online documentation.

ODS statements:
For reference information about the ODS statements, see Chapter 6, “Dictionary of ODS Language Statements,” on page 105. These statements control the many features of the Output Delivery System.

Using ODS with the DATA step:
With the addition of ODS-related options to the FILE and PUT statements, you can use ODS to produce enhanced DATA step reports. See Chapter 3, “Using ODS with the DATA Step,” on page 53.

Creating your own templates:
For even more control over formatting, you can create your own templates for formatting output. See “TEMPLATE Procedure: Overview” in SAS Output Delivery System: Procedures Guide.

information about the ODSDEST= system option, see “ODSDEST= System Option” on page 907.
Part 2

Concepts

Chapter 2

*Output Delivery System: Basic Concepts* ........................................ 29
Chapter 2
Output Delivery System: Basic Concepts

Overview of How ODS Works

Components of SAS Output .................................................. 29
Features of ODS ................................................................. 30
Clearing the Results Window ................................................. 31

Understanding Item Stores and Template Stores ..................... 32

Understanding ODS Destinations ........................................... 33
Overview of ODS Destination Categories ................................. 33
Definition of Destination-Independent Input ............................... 34
The SAS Formatted Destinations ............................................. 34
The Third-Party Formatted Destinations ................................... 35
Controlling the Formatting Features of Third-Party Formats ........... 37
ODS Destinations and System Resources .................................. 37

How ODS Determines the Destinations for an Output Object ......... 38
Selection and Exclusion Lists .................................................. 39

Customized Output for an Output Object ................................ 39

Customizing Titles and Footnotes ............................................ 40

Changing SAS Registry Settings for ODS ................................. 40
Overview of ODS and the SAS Registry .................................... 40
Changing the SAS Windowing Environment Output Defaults ......... 41
Changing the ODS Default HTML Style .................................... 43
Changing the Default HTML Version Setting ............................... 44
Changing ODS Destination Default Settings ............................... 45
Changing ODS Printer Destination Default Printer Value ............. 46
Changing ODS EPUB Destination Default Value ........................ 48
Changing the ODS EPUB and EPUB3 Encoding Settings ............... 49

Overview of How ODS Works

Components of SAS Output

The PROC or DATA step supplies raw data and the name of the table template that contains the formatting instructions. ODS formats the output. You can use ODS to format output from individual procedures and from the DATA step in many different forms other than HTML output.
The following figure shows how SAS produces ODS output.

**Figure 2.1  ODS Processing: What Goes in and What Comes Out**

![Diagram of ODS Processing]

**Table 2.1  * List of Tagsets That SAS Supplies and Supports**

<table>
<thead>
<tr>
<th></th>
<th>CHTML</th>
<th>CSV</th>
<th>CSVALL</th>
<th>CSVBYLINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT</td>
<td>EXCELXP</td>
<td>HTML4</td>
<td>HTMLPANEL</td>
<td></td>
</tr>
<tr>
<td>MSOFFICE2K</td>
<td>PHTML</td>
<td>TAGSETS.RTF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 2.2  * Additional Diagnostic Tagsets That SAS Supports**

<table>
<thead>
<tr>
<th></th>
<th>EVENT_MAP</th>
<th>NAMEDHTML</th>
<th>SHORT_MAP</th>
<th>STYLE_DISPLAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>STYLE_POPUP</td>
<td>TEXT_MAP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note: There are also preproduction tagsets. These tagsets can be found at [http://support.sas.com](http://support.sas.com) and are not yet supported by SAS.*

**Features of ODS**

ODS is designed to overcome the limitations of traditional SAS output and to make it easy to access and create files in different formats. ODS provides a method of delivering output in a variety of formats, and makes the formatted output easy to access.

Important features of ODS include the following:
• ODS combines raw data with one or more table templates to produce one or more output objects. These objects can be sent to any or all ODS destinations. You control the specific type of output from ODS by selecting an ODS destination. The currently available ODS destinations can produce the following types of output:
  • traditional monospace output
  • an output data set
  • an ODS document that contains a hierarchy file of the output objects
  • output that is formatted for a high-resolution printer such as PostScript and PDF
  • output that is formatted in various markup languages such as HTML
  • RTF output that is formatted for use with Microsoft Word
  • PowerPoint output that is formatted for use with Microsoft PowerPoint
  • output that is formatted for EPUB e-books

• ODS provides table templates that define the structure of the output from SAS procedures and from the DATA step. You can customize the output by modifying these templates, or by creating your own.

• ODS provides a way for you to choose individual output objects to send to ODS destinations. For example, PROC UNIVARIATE produces five output objects. You can easily create HTML output, an output data set, LISTING output, or printer output from any or all of these output objects. You can send different output objects to different destinations.

• In the SAS windowing environment, ODS stores a link to each output object in the Results folder in the Results window.

• Because formatting is now centralized in ODS, the addition of a new ODS destination does not affect any procedures or the DATA step. As future destinations are added to ODS, they will automatically become available to the DATA step and all procedures that support ODS.

• With ODS, you can produce output for numerous destinations from a single source, but you do not need to maintain separate sources for each destination. This feature saves you time and system resources by enabling you to produce multiple types of output with a single run of your procedure or data query.

**Clearing the Results Window**

The ODS HTML destination is turned on by default when you use the SAS windowing environment. Therefore, the NEWFILE=NONE option is set by default in the HTML file and all output is sent to a single HTML file. When you clear the Results window manually by clicking on a node in the Results window and selecting **Delete**, or by issuing the `ODSRESULTS; CLEAR` command, nodes are deleted from the Results window. However, the HTML file remains open and the output from subsequent procedure or DATA steps is appended to any previously created HTML output. You can remove the contents of the HTML file, close the HTML file, and reset preferences to the default state by doing one of the following:

• Using the Command Line
  1. Select the Results window or the Results Viewer window.
  2. Submit either of these commands in the command line:

     CLEAR
Using the SAS Windowing Environment

1. Select the Results window or the Results Viewer window.
2. From the Menu, select Edit ➔ Clear All.

Understanding Item Stores and Template Stores

A template store is an item store that stores items that were created by the TEMPLATE procedure. Items that SAS provides are in the item store Sashelp.Tmplmst. Compiled templates are stored physically in the Sasuser.Templat item store by default. However, you can store items that you create in any template store where you have Write access.

A template store can contain multiple levels known as directories. When you specify a template store in the ODS PATH statement, however, you specify a two-level name. The two-level name includes a libref and the name of a template store in the SAS library that the libref references.

When using a locale that is different from the default locale, or using a product that has templates stored outside of Sashelp.Tmplmst, the template store for the locale and product are automatically inserted into the ODS path when needed. Since these extra template stores are variants or extensions to Sashelp.Tmplmst, they are inserted just before Sashelp.Tmplmst in the current ODS path. For example, if the Japanese locale is...
selected, the default ODS path would be Sasuser.Templat(update) Sashelp.TMPL_JA(read) Sashelp.Tmplmst(read) rather than Sasuser.Templat(update) Sashelp.Tmplmst(read).

For products or procedures that use a template store other than Sashelp.Tmplmst for their templates, a similar operation occurs. The name of the template store is based on the template path. For example, if SAS/STAT delivered a separate template store and the Stat.GLM.Anova output object is being rendered, the ODS path would be Sasuser.Templat(update) Sashelp.TMPLPROCGLM(read) Sashelp.Tmplstat(read) Sashelp.Tmplmst(read).

When using both a non-default locale and a product that uses a template store other than Sashelp.Tmplmst, the operations are combined. If we combined the above SAS/STAT example with the Japanese locale, the ODS path would be Sasuser.Templat(update) Sashelp.Tmplprocglm_ja(read) Sashelp.Tmplprocglm(read) Sashelp.Tmplstat_en(read) Sashelp.Tmplstat(read) Sashelp.Tmplmst(read).

Because these extra template stores are inserted transiently as output objects are created, they do not show up when using the ODS PATH statement.

Understanding ODS Destinations

Overview of ODS Destination Categories

ODS enables you to produce SAS procedure and DATA step output to many different destinations. ODS destinations are organized into two categories.

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

Third-Party Formatted destinations
 produce output that enables you to apply styles or markup languages, or enables your output to be printed to physical printers using page description languages. For example, you can produce output in PostScript, HTML, XML, EPUB, or PowerPoint. You can also apply a style or use a markup language that you create with these destinations.

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

<table>
<thead>
<tr>
<th>Category</th>
<th>Destinations</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Formatted</td>
<td>DOCUMENT</td>
<td>ODS document</td>
</tr>
<tr>
<td></td>
<td>LISTING</td>
<td>SAS output listing</td>
</tr>
<tr>
<td></td>
<td>OUTPUT</td>
<td>SAS data set</td>
</tr>
<tr>
<td>Third-Party Formatted</td>
<td>EPUB</td>
<td>Output written in EPUB format that can be used with various e-book readers</td>
</tr>
</tbody>
</table>
### Definition of Destination-Independent Input

Destination-independent input means that one destination can support a feature even though another destination does not support it. In this case, the request is ignored by the destination that does not support it. Otherwise, ODS would support a small subset of features that are common to all destinations. If this were true, then it would be difficult to move your reports from one output format to another output format. ODS provides many output formatting options, so that you can use the appropriate format for the output that you want. It is best to use the destination appropriate for your purpose.

### The SAS Formatted Destinations

#### About SAS Formatted Destinations

The SAS Formatted destinations create SAS entities such as a SAS data set, a SAS output listing, or an ODS document. The statements in the ODS SAS Formatted category create the SAS entities.

#### DOCUMENT

The DOCUMENT destination enables you to restructure, navigate, and replay your data in different ways and to different destinations. You can do this without needing to rerun your analysis or repeat your data query. The DOCUMENT destination makes your entire output stream available in "raw" form and accessible to you to customize. The output is kept in the original internal representation as a data component plus a table template. When the output is in a DOCUMENT form, it is possible to rearrange, restructure, and reformat without rerunning your analysis. Unlike other ODS destinations, the DOCUMENT destination has a GUI interface. However, everything that you can do through the GUI, you can also do with batch commands using the ODS DOCUMENT statement and the DOCUMENT procedure.

---

<table>
<thead>
<tr>
<th>Category</th>
<th>Destinations</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>HTML file for online viewing</td>
<td></td>
</tr>
<tr>
<td>MARKUP</td>
<td>Markup language tagsets</td>
<td></td>
</tr>
<tr>
<td>PowerPoint</td>
<td>Output written in Microsoft PowerPoint format</td>
<td></td>
</tr>
<tr>
<td>PRINTER</td>
<td>Printable output in one of three different formats: PCL, PDF, or PS (PostScript)</td>
<td></td>
</tr>
<tr>
<td>RTF</td>
<td>Output written in Rich Text Format for use with Microsoft Word</td>
<td></td>
</tr>
</tbody>
</table>

As future destinations are added to ODS, they automatically will become available to the DATA step and to all procedures that support ODS.
LISTING
The LISTING destination enables you to produce traditional SAS output with the same look and presentation as it had in previous versions of SAS. Because most procedures share some of the same table templates, the output is more consistent. For example, if you have two different procedures producing an ANOVA table, they both produce it in the same way. This happens because each procedure uses the same template to describe the table. However, there are three procedures that do not use a default table template to produce their output: the PRINT procedure, the REPORT procedure, and the TABULATE procedure. These procedures use the structure that you specify in your program code to define their tables.

OUTPUT
The OUTPUT destination produces SAS output data sets. ODS already knows the logical structure of the data and its native form. Therefore, ODS can create a SAS data set that represents exactly the same resulting data set that the procedure worked with internally. The output data sets can be used for further analysis, or for sophisticated reports in which you want to combine similar statistics across different data sets into a single table. You can easily access and process your output data sets using all of the SAS data set features. For example, you can access your output data using variable names and perform WHERE-expression processing just as you would process data from any other SAS data set.

The Third-Party Formatted Destinations

About Third-Party Formatted Destinations
The Third-Party Formatted destinations enable you to apply styles to the output objects that are used by applications other than SAS.

EPUB (electronic publication)
EPUB (electronic publication) is a free and open e-book standard produced by the International Digital Publishing Forum (IDPF). EPUB is designed for reflowable content, meaning that an EPUB reader can optimize text for a particular display device. EPUB also supports fixed-layout content. E-books that use the .epub format can be read by a wide variety of e-book readers, from dedicated hardware to desktop software to online-based readers.

The ODS EPUB statement generates output with the .epub extension. The .epub file extension consists of XML files for reflowable digital books and publications. The .epub file is just a renamed ZIP archive containing book files, either XHTML or DTBook, a number of XML description and navigation files, and possibly image or media files.

HTML (Hypertext Markup Language)
The HTML destination produces HTML 4.0 output that contains embedded style sheets. However, you can produce HTML 3.2 output using the HTML3 statement or HTML 5.0 using the HTML5 statement. The HTML destination is the default destination that opens when you start your SAS session in DMS mode. Thus, ODS is always being used, even when you do not explicitly invoke ODS.

The HTML destination can create some or all of the following:
• an HTML file (called the body file) that contains the results from the procedure or DATA step
• a table of contents that links to the body file
• a table of pages that links to the body file
• a frame that displays the table of contents, the table of pages, and the body file

If you do not want to link to your output, then you do not have to create a table of contents or a frame and contents. However, if your output is very large, you might want to create a table of contents and a frame for easier reading and traversing through your file.

**Markup Family of Destinations**

*Tagsets* describe how to produce markup language output. You can use a tagset that SAS supplies or you can create your own tagset using the Template procedure. Similar to table templates and style attributes, tagsets enable you to modify your markup language output. For example, you can specify each variety of XML as a new tagset. SAS supplies you with a collection of XML tagsets and enables you to produce a customized variety of XML.

The important point is that you can implement either a tagset that SAS supplies or a customized tagset that you created. You do not have to wait for the next release of SAS. The additional capability to modify and create your own tagsets by using PROC TEMPLATE gives you greater flexibility in customizing your output.

For a complete list of the markup language tagsets that SAS supplies, see the section on listing tagset names in “ODS MARKUP Statement” on page 488. To learn how to define your own tagsets, see the section on methods to create your own tagsets in “TEMPLATE Procedure: Creating Markup Language Tagsets” in SAS Output Delivery System: Procedures Guide.

The MARKUP destination cannot replace ODS PRINTER or ODS RTF destinations because it cannot do text measurement. Therefore, it cannot produce output for a page description language or a hybrid language like RTF. These languages require all of the text to be measured and placed at a specific position on the page.

However, the measured markup destination, TAGSETS.RTF can determine where page breaks occur. The TAGSETS.RTF destination is the first production measured tagset. Others are planned. See “ODS TAGSETS.RTF Statement ” on page 803 for specific information.

**PowerPoint**

Microsoft PowerPoint is the name of a proprietary commercial software presentation program developed by Microsoft. It is part of the Microsoft Office suite and runs on Microsoft Windows and Apple's Mac OS X operating system. PowerPoint presentations consist of a number of individual pages or "slides". Slides can contain text, graphics, and tables. The ODS POWERPOINT statement generates output with the .ppt extension.

**Printer Family of Destinations**

The PRINTER destination produces output for the following:

• printing to physical printers such as Windows printers under Windows, PCL, and PostScript printers on other operating systems
• producing portable PostScript, PCL, and PDF files

The PRINTER destinations produce ODS output that contains page description languages: they describe precise positions where each line of text, each rule, and each graphical element are to be placed on the page. In general, you cannot edit or alter these formats. Therefore, the output from ODS PRINTER is intended to be the final form of the report.
**Rich Text Format (RTF)**

RTF produces output for Microsoft Word. Other applications can read RTF files, but the RTF output might not work successfully with them. The RTF destination enables you to view and edit the RTF output. ODS does not define the vertical measurement, which means that SAS does not determine the optimal place to position each item on the page.

Because Microsoft Word needs to know the widths of table columns and it cannot adjust tables if they are too wide for the page, ODS measures the width of the text and tables (horizontal measurement). Therefore, SAS can set all of the column widths properly and divide the table into panels if it is too wide to fit on a single page.

In short, when SAS produces RTF output for input to Microsoft Word, it determines the horizontal measurement. Microsoft Word controls the vertical measurement. Because Microsoft Word can determine how much space is on the page, your tables are displayed consistently even after you make changes to your RTF file.

However, when you use measured RTF (produced using the ODS TAGSETS.RTF statement), you can specify how and where page breaks occur. You can also specify when to place titles and footnotes into the body of a page. SAS becomes responsible for the implicit page breaks instead of Microsoft Word. See “ODS TAGSETS.RTF Statement” on page 803 for specific information.

**Controlling the Formatting Features of Third-Party Formats**

All of the formatting features that control the appearance of the third-party formatted destinations beyond what the LISTING destination can do are controlled by two mechanisms:

- ODS statement options
- ODS style attributes

The ODS statement options control three features:

1. features that are specific to a given destination, such as style sheets for HTML
2. features that are global to the document, such as AUTHOR and table of contents generation
3. features that programmers change on each document, such as the output filename

The ODS style attributes control how individual elements are created. Attributes are aspects of a given style, such as type face, weight, font size, and color. The values of the attributes collectively determine the appearance of each part of the document to which the style is applied. With style attributes, it is unnecessary to insert destination-specific code (such as raw HTML) into the document. Each output destination interprets the attributes that are necessary to generate the presentation of the document. Because not all destinations are the same, not all attributes can be interpreted by all destinations. Style attributes that are incompatible with a selected destination are ignored. For example, PostScript does not support active links, so the URL= attribute is ignored when producing PostScript output.

**ODS Destinations and System Resources**

ODS destinations can be open or closed. You open and close a destination with the appropriate ODS statement. When a destination is open, ODS sends the output objects to it. An open destination uses system resources even if you use the selection and exclusion features of ODS to select or exclude all objects from the destination. Therefore, to
conserve resources, close unnecessary destinations. See the documentation for individual
ODS statements for detailed information.

By default, only the HTML destination is open in all modes except batch mode.
Consequently, if you do nothing, your SAS programs run and produce HTML output. In
batch mode, the LISTING destination is the default.

---

**How ODS Determines the Destinations for an Output Object**

As each output object is produced, ODS uses the selection and exclusion lists to
determine which destination or destinations the output object is sent to. The following
figure illustrates this process.

For each destination, ODS first asks whether the list for that destination includes the
object. If it does not, ODS does not send the output object to that destination. If the list
for that destination does include the object, ODS reads the overall list. If the overall list
includes the object, ODS sends it to the destination. If the overall list does not include
the object, ODS does not send it to the destination.

*Figure 2.3  Directing an Output Object to a Destination*
Selection and Exclusion Lists

For each ODS destination, ODS maintains either a selection list or an exclusion list of output objects. You can use the default output objects selected or excluded for each destination or you can specify which output object you want to produce by selecting or excluding them from a list.

A selection list is a list of output objects that are sent to an ODS destination. An exclusion list is a list of output objects that are excluded from an ODS destination. ODS also maintains an overall selection or exclusion list of output objects. By checking the destination-specific lists and the overall list, ODS determines what output objects to produce. These lists can be modified by using the ODS SELECT statement and the ODS EXCLUDE statement.

**T I P** You can maintain a selection list for one destination and an exclusion list for another. However, the results are less complicated if you maintain the same types of lists for all the destinations to which you route output.

You can view the contents of the exclusion and selection lists by using the ODS SHOW statement. The current selection list is written to the SAS log.

EXCLUDE ALL is the default setting for the ODS OUTPUT destination. SELECT ALL is the default setting for all other destinations. To change the default selection and exclusion lists, use the ODS SELECT or ODS EXCLUDE statements or use the exclude and select actions that are available for some of the ODS statements. However, to set the exclusion list for the OUTPUT destination to something other than the default, use the “ODS OUTPUT Statement” on page 534. For a list of ODS output destinations and explanations of each, see “Understanding ODS Destinations” on page 33.

In order to view output objects that are selected or excluded from your program, use the ODS TRACE statement. The ODS TRACE statement prints the output objects that are selected and excluded and puts the information in a trace record that is written to the SAS log. The trace provides the path, the label, and other information about output objects that are selected and excluded. For complete documentation about viewing and selecting output objects, see the “ODS SELECT Statement” on page 758, the “ODS EXCLUDE Statement” on page 321, and the “ODS TRACE Statement” on page 854.

Customized Output for an Output Object

For a procedure, the name of the table template that is used for an output object comes from the procedure code. The DATA step uses a default table template unless you specify an alternative with the TEMPLATE= suboption in the ODS option in the FILE statement. For more information, see the section on the TEMPLATE= suboption in “FILE Statement for ODS” on page 59.

To find out which table templates a procedure or the DATA step uses for the output objects, you must look at a trace record. To produce a trace record in your SAS log, submit the following SAS statements:

```sas
ods trace on;
```
your-proc-or-DATA-step
ods trace off;

Remember that not all procedures use table templates. If you produce a trace record for
one of these procedures, no template appears in the trace record. Conversely, some
procedures use multiple table templates to produce their output. More than one template
appears in the trace record produced in the log.

For a detailed explanation of the trace record, see the “ODS TRACE Statement” on page
854.

You can use PROC TEMPLATE to modify an entire table template. When a procedure
or DATA step uses a table template, it uses the elements that are defined or referenced in
its table template. In general, you cannot directly specify a table element for your
procedure or DATA step to use without modifying the template itself.

Note: Three Base SAS procedures, PROC PRINT, PROC REPORT, and PROC
TABULATE, do provide a way for you to access table elements from the procedure
step itself. Accessing the table elements enables you to customize your report. For
more information about these procedures, see Base SAS Procedures Guide.

Customizing Titles and Footnotes

You can use the global TITLE and FOOTNOTE statements to enhance the readability of
any report. These statements have options that enable you to customize the style of the
titles and footnotes when they are used with ODS. Because these options control only
the presentation of the titles and footnotes, they have no effect on objects that go to the
LISTING or OUTPUT destination. Examples of these style options are BOLD,
COLOR=, and FONT=. For a complete list of style options, detailed information about
the style options, and example code, see the TITLE statement and the FOOTNOTE
statement in SAS Statements: Reference.

When used with SAS/GRAPH, you can choose whether to render the titles and footnotes
as part of the body of the document or as part of the graphics image. Where the titles and
footnotes are rendered determines how you control the font, size, and color of the titles
and footnotes text. For details about this ODS and SAS/GRAPH interaction, see
Controlling Titles and Footnotes in SAS/GRAPH: Reference.

For information about titles and footnotes rendered with and without using the graphics
option USEGOPT, see “ODS USEGOPT Statement” on page 860.

Changing SAS Registry Settings for ODS

Overview of ODS and the SAS Registry

The SAS registry is the central storage area for configuration data that ODS uses. This
configuration data is stored in a hierarchical form, which works in a similar manner to
how directory-based file structures work under UNIX, Windows, and the z/OS UNIX
system. Folders appear in a tree on the left side of the SAS Registry Editor. If a folder
has subfolders, you can expand or collapse folders with the + and - icons that are found
about managing the SAS Registry.
The following display shows the Registry Editor window with the ODS Printer destination properties displayed in the Contents of ‘PRINTER’ pane.

Figure 2.4  ODS Destinations in SAS Registry

Changing the SAS Windowing Environment Output Defaults

In the SAS Windowing environment for Windows and UNIX operating systems, the ODS output defaults are set as follows:

- ODS Graphics is enabled at SAS start-up.
- The LISTING destination is closed and the HTML destination is open.
- The default style for the HTML destination is HTMLBlue.

You can change the ODS output defaults using SAS Registry as shown in the following examples. For details about using other methods to change the ODS output defaults, see “Working with Output Defaults” on page 21.

To change the default ODS Graphics setting in the SAS registry:

1. Select Solutions -> Accessories -> Registry Editor or issue the command REGEDIT.
2. Expand the ODS -> DMS folders. From the DMS folder, select ODS GRAPHICS.
3. Select Default State in the Contents of ‘ODS GRAPHICS’ pane to choose what you want to modify.
4. Select Edit -> Modify or click the right mouse button and select MODIFY.
5. Turn off ODS Graphics by typing OFF in the Value Data text box in the Edit Value String or Edit Signed Integer Value dialog box.
6. Select OK.
To change the default ODS output destination in the SAS registry from the HTML destination to the LISTING destination, you need to turn off the HTML output destination and turn on the LISTING destination.

1. Select **Solutions** ⇒ **Accessories** ⇒ **Registry Editor** or issue the command `REGEDIT`.
2. Expand the **ODS** ⇒ **DMS** folders and select **DMS**.
3. Select **Default HTML State** in the **Contents of ‘DMS’** pane.
4. Select **Edit** ⇒ **Modify** or click the right mouse button and select **MODIFY**.
5. Turn off HTML output by typing **OFF** in the **Value Data** text box in the **Edit String** or **Edit Signed Integer Value** dialog box. Select **OK**.
6. Expand the **DESTINATIONS** folder and then select **MONOSPACE**.
7. Select **Default State** in the **Contents of ‘MONOSPACE’** pane.
8. Select **Edit** ⇒ **Modify** or click the right mouse button and select **MODIFY**.
9. Turn on LISTING output by typing **ON** in the **Value Data** text box in the **Edit String** Value dialog box. Select **OK**.
Changing the ODS Default HTML Style

The default style for HTML is HTMLBlue. To change this default style in the Registry:

1. Select **Solutions** ⇒ **Accessories** ⇒ **Registry Editor** or issue the command `REGEDIT`.
2. Expand the **ODS** ⇒ **DESTINATIONS** ⇒ **MARKUP** folders.
3. Select **HTML4**.

---

**Figure 2.6** Registry: Turn OFF the HTML Default Destination

**Figure 2.7** Registry: Turn ON the LISTING Destination
4. Select **Selected Style** from the **Contents of ‘HTM4’** pane.
5. Select **Edit** ⇒ **Modify** or click the right mouse button and select **MODIFY**.
6. Enter the default style in the **Value Data** text box in the Edit Value String or Edit Signed Integer Value dialog box. Select **OK**.

*Figure 2.8 Changing the HTM4 Default Style in the SAS Registry*

---

**Changing the Default HTML Version Setting**

By default, the SAS registry is configured to generate HTML4 output when you specify the ODS HTML statement. To permanently change the default HTML version, you can change the setting of the HTML version in the SAS registry.

**CAUTION:**

*If you make a mistake when you modify the SAS registry, then your system might become unstable or unusable.* You are not warned if an entry is incorrect. Incorrect entries can cause errors, and can even prevent you from bringing up a SAS session. See the section on configuring your registry in *SAS Language Reference: Concepts*.

To change the default setting of the HTML version in the SAS registry:

1. Select **Solutions** ⇒ **Accessories** ⇒ **Registry Editor** or issue the command **REGEDIT**.
2. Select **ODS**.
3. From the **Contents of ‘ODS’** pane, select **Default HTM1 Version**.
4. Select **Edit** ⇒ **Modify** or click the right mouse button and select **MODIFY**.
5. Enter the HTML version that you want in the **Value Data** text box in the Edit String Value dialog box. Select **OK**.
Changing ODS Destination Default Settings

Each ODS destination has default settings that are stored in the SAS registry. For example, to change the default values of the RTF destination:

1. Expand the ODS ➔ Destinations folders.
2. Select the RTF destination.
3. From the Contents of ‘RTF’ pane, select Columns to change the default Columns value.
4. Select Edit ➔ Modify or click the right mouse button and select MODIFY.
5. Enter the column value into the Value Data text box in the Edit String Value dialog box. Select OK.
If you are running on a Microsoft Windows platform with system option NOUNIVERSALPRINT configured, the default printer shown in the SAS Registry for ODS PRINTER is the value specified by the Windows system option SYSPRINT. Otherwise, the default printer is specified by the Default Printer value that is set in CORE PRINTING in the SAS registry. You can now set this Default Printer value.

Note: The setting in the CORE PRINTING registry key changes automatically when the default printer is changed in the Print Setup window of the SAS System.

To change the Registry Default Printer value of the ODS PRINTER destination, perform the following steps:

1. Expand the CORE PRINTING folders.
2. Select the PRINTING folder to display the default Printer settings.
3. In the Contents of ‘PRINTING’ pane, select the Default Printer value.
4. Select Edit Modify or click the right mouse button and select MODIFY.
5. Enter the value into the Value Data text box in the Edit String Value dialog box. Select OK.
The default PCL printer is determined by the Default Printer setting in the PCL key of the SAS Registry. To change the ODS PCL Default Printer value, from the Registry Editor window, perform the following steps:

1. Select ODS → DESTINATIONS → PRINTER → PCL.
2. In the Contents of ‘PCL’ pane, select the Default Printer value.
3. Select Edit → Modify or click the right mouse button and select MODIFY.
4. Enter the value into the Value Data text box in the Edit String Value dialog box. Select OK.

Note: The Default Printer value in the Registry Editor window can also be changed for the ODS PDF and ODS PS destinations.
Changing ODS EPUB Destination Default Value

The default EPUB version (EPUB2 or EPUB3) is determined by the SAS 9.4 version that you are running. In SAS 9.4, the default EPUB version was EPUB2. In the third maintenance release of SAS 9.4, the default EPUB version is EPUB3. The default versions are set in the Registry.

To change the Registry Default EPUB Version for the ODS EPUB destination, perform the following steps.

Note: Once the default EPUB version is changed, when you use the ODS EPUB statement, the version set in the Registry is used. You can also explicitly use the ODS EPUB2 statement or the ODS EPUB3 statement to ensure that you are using the version that you want.

1. Select the ODS folder to display the default ODS settings.
2. In the Contents of ODS pane, select Default EPUB Version.
3. Select Edit ⇒ Modify or click the right mouse button and select MODIFY.
4. Enter the value into the Value Data text box in the Edit String Value dialog box. Select OK.
Changing SAS Registry Settings for ODS

Figure 2.13  Changing the Default EPUB Version in the Registry Editor Window

The default PCL printer is determined by the Default Printer setting in the PCL key of the SAS Registry. To change the ODS PCL Default Printer value, from the Registry Editor window, perform the following steps:

1. Select ODS ⇔ DESTINATIONS ⇔ PRINTER ⇔ PCL.
2. In the Contents of ‘PCL’ pane, select the Default Printer value.
3. Select Edit ⇔ Modify or click the right mouse button and select MODIFY.
4. Enter the value into the Value Data text box in the Edit String Value dialog box. Select OK.

Note: The Default Printer value in the Registry Editor window can also be changed for the ODS PDF and ODS PS destinations.

Figure 2.14  Changing the Default PCL Printer Value in the Registry Editor Window

Changing the ODS EPUB and EPUB3 Encoding Settings
ODS uses the encoding that is specified in the SAS registry to specify the encoding of the e-book's contents. This value overrides the encoding for input or output processing (transcodes) of external files. Valid encodings or ODS EPUB and ODS EPUB3 are UTF-8, UTF-16, UTF-16BE, and UTF-16LE.

**Note:** If no encoding is set in the ODS registry, the encoding is determined by the locale setting described in ENCODING System Option: UNIX, Windows, and z/OS in SAS National Language Support (NLS): Reference Guide. For information about the ENCODING= option, see “ENCODING System Option: UNIX, Windows, and z/OS” in SAS National Language Support (NLS): Reference Guide.

To change the default encoding values of the ODS EPUB or EPUB3 destination:

1. Expand the **ODS ➔ Destinations ➔ SCRIPT** folders.
2. Select the **EPUB3** destination.
3. From the **Contents of ‘EPUB3’** pane, select **Encodings** to change the default Encoding value.
4. Double Click **Encoding** and the **Edit String Value** window appears.
5. Enter the encoding value to change into the **Value Data** text box in the Edit String Value dialog box. Select **OK**.

*Figure 2.15  Changing ODS EPUB3 Encoding in the SAS Registry*
Output Delivery System and the DATA Step

Chapter 3
Using ODS with the DATA Step
Chapter 3
Using ODS with the DATA Step

Introduction
If you are writing DATA step reports now, you are already using ODS. HTML output, the 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, however, you have a broad range of choices for printing your customized DATA step reports:

- You can produce DATA step reports in many different formats, such as HTML, RTF, PS (PostScript), or PDF.
- You can create the report in multiple formats at the same time.
- You can also produce the report in different formats at a later time without rerunning the DATA step.
- You can use the Report Writing Interface to create highly customized reports with the DATA step.

To take advantage of these enhanced reporting capabilities, you can combine DATA step programming with the formatting capabilities of ODS. To create PDF output, for example, start with the DATA steps 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. In addition, you can choose from several ODS formatting statements to format the output in other presentation styles, such as HTML, RTF, and PS. For more information about ODS statements, see Chapter 5, “Introduction to ODS Language Statements,” on page 103.

### How ODS Works with the DATA Step

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

<table>
<thead>
<tr>
<th>Steps</th>
<th>Tools</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify formatting for your output</td>
<td>ODS formatting statements can specify formats such as listing, MARKUP family, RTF, PS, and PDF.</td>
<td>You can also produce output in multiple formats at the same time by specifying more than one format. Note: If you want only the default output, then you do not need a destination ODS statement.</td>
</tr>
<tr>
<td>Specify structure</td>
<td>The ODS option in the FILE statement lists the variables and their order in the output.</td>
<td>Additional suboptions give you even more control over the resulting structure.</td>
</tr>
<tr>
<td>Connect the data to the template</td>
<td>The FILE PRINT ODS statement creates an output object by binding a data component to a table template.</td>
<td>You can specify other details by using various ODS suboptions in the FILE PRINT ODS statement.</td>
</tr>
<tr>
<td>Output data</td>
<td>The PUT statement writes variable values to the data component.</td>
<td>A simple way to output all variables is to use PUT <em>ODS</em>.</td>
</tr>
</tbody>
</table>

First, use ODS statements to specify how you want ODS to format your output (for example, as HTML, RTF 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, and the FILE PRINT ODS statement directs the output.\(^1\) You can use ODS to produce the same output in multiple formats. You can also produce output at a later time in a different format without rerunning the DATA step.

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

Here is a list of topics, with sources for additional information.

---
\(^1\) 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 log, to the regular DATA step output buffer, or to another external file in the same DATA step.
Syntax for ODS Enhanced Features in a DATA Step

To use the DATA step and ODS to produce output that contains more enhanced formatting features than the default output, you must use both the FILE PRINT ODS statement and the PUT statement.

Dictionary

PUT Statement for ODS

Writes data values to a special buffer from which they can be written to the data component and then formatted by ODS.

Valid in: DATA step
Category: File-Handling
Type: Executable
Requirement: If you use the _ODS_ option in the PUT statement, then you must use the FILE PRINT ODS statement.
Note: This syntax shows only the ODS form of the PUT statement when you are binding to a template. For the complete syntax, see the “PUT Statement” in SAS Statements: Reference.

Syntax

PUT <specification> <_ODS_> <@|@@> ;

Optional Arguments

specification

specifies one or more variables to write and where to write them. specification has the following form:
ods-pointer-control variable-1 <...<ods-pointer-control-n> variable-n>

ods-pointer-control
  moves the pointer in the buffer to a specified line or column.

See  “When the Pointer Moves Past the End of a Line ” on page 58

variable
  identifies the variable to write.

Example  “Example 4: Creating and Using a User-Defined Table Template” on page 85

_ODS_
  specifies that the PUT statement writes values to the data component for each of the
variables that were defined as columns with the FILE PRINT ODS COLUMNS= statement.

Default  The order of these columns is determined by the order that is
  specified by the COLUMNS= suboption in the FILE PRINT ODS
  statement. If you omit the COLUMNS= suboption, then the order of
  the variables in the program data vector determines their order in the
  output object.

Requirement  If you specify the _ODS_ option, then you must use the FILE PRINT
  ODS statement and the FILE PRINT ODS statement must precede
  the PUT _ODS_ statement.

Interaction  You can use _ODS_ in a PUT statement that specifies the placement
  of individual variables. _ODS_ writes to a particular row and column
  only if another PUT statement has not already written a variable to
  that same row and column. The position of _ODS_ in the PUT
  statement does not affect the outcome in the data component.

Tip  By default, the order of the columns in the data component matches
  the order of the columns in the buffer. However, if you have specified
  a table template, it might override this order. For more information,
  see the discussion of ORDER_DATA.

See  For more information, see ODS<=(ODS-suboptions)> on page 59.

@ | @@
  holds an output line for the execution of the next PUT statement across iterations of
  the DATA step. The line-hold specifiers are called trailing @ and double trailing @.

Default  If you do not use @ or @@, then each PUT statement in a DATA step
  writes a new line to the buffer.

See  “When the Pointer Moves Past the End of a Line ” on page 58

Details

**ODS Column Pointer Controls**

ODS column pointer controls differ slightly from column pointer controls in a PUT
statement that does not use ODS. An ODS column refers not to a single character space
but to a column that contains an entire variable value. Therefore, an ODS column pointer
control moves from one entire value to the next, not from one character space to another.
Column 1 contains values for the first variable in the output; column 2 contains values for the second variable, and so on.

ODS column pointer controls have the following general forms:

@ods-column

moves the pointer to the specified ODS column. ods-column is a number, a numeric variable, or an expression that identifies the column to write to.

Default
If ods-column exceeds the number of columns in the data component, then ODS writes the current line, moves the pointer to the first ODS column on the next line, and continues to process the PUT statement.

Requirement
If ods-column is a number, then it must be a positive integer. If ods-column is a numeric variable or an expression, then SAS treats it as follows. If ods-column is not an integer, then SAS truncates the decimal portion and uses only the integer value. If ods-column is 0 or negative, then SAS moves the pointer to column 1.

Tip
You can alter the default behavior with options in the FILE PRINT ODS statement. For more information, see the discussion of overflow control on page 59.

Example
“Example 4: Creating and Using a User-Defined Table Template” on page 85

+ods-column

moves the pointer by the specified number of ODS columns. ods-column is a number, a numeric variable, or an expression that specifies the number of columns to move the pointer.

Requirement
If ods-column is a number, then it must be an integer. If ods-column is a numeric variable or an expression, then it does not have to be an integer. If it is not an integer, then SAS truncates the decimal portion and uses only the integer value. If ods-column is a positive integer, SAS moves the pointer to the right. If ods-column is a negative integer, SAS moves the pointer to the left. If ods-column is 0, SAS does not move the pointer.

Tip
If the current column position becomes less than 1, then the pointer moves to column 1. If the current column position exceeds the number of columns in the data component, then ODS writes the current line, moves the pointer to the first ODS column on the next line, and continues to process the PUT statement.

Example
“Example 4: Creating and Using a User-Defined Table Template” on page 85

@ 'column-name'

moves the pointer to the ODS column identified by 'column-name'. The column name is a data component variable name.

Requirement
column-name must be enclosed in quotation marks.
**ODS Line Pointer Controls**
Line pointer controls in a DATA step that uses ODS are the same as line pointer controls in a DATA step that does not use ODS. However, you can use only those listed below with ODS. Line pointer controls have the following general forms:

### `#line`
- moves the pointer to the specified line. `line` is a number, a numeric variable, or an expression that identifies the line to write to.

**Requirement** If `line` is a number, then it must be an integer. If `line` is a numeric variable or an expression, it does not have to be an integer. If it is not an integer, then SAS truncates the decimal portion and uses only the integer value.

### `/`
- moves the pointer to the first column of the next line.

**Example** “Example 4: Creating and Using a User-Defined Table Template” on page 85

**Note:** If you use a line pointer control to skip lines in ODS output, then all columns that are not referenced on the current line, or skipped lines, are set to a missing value. Columns that contain numeric values display a period for the missing value. If you prefer not to include these periods in your ODS output, you can display missing numeric values as a blank by using the `MISSING` statement (or the `MISSING=` system option). For more information about the statement, see “MISSING Statement” in *SAS Statements: Reference*. For more information about the system option, see “MISSING= System Option” in *SAS System Options: Reference*.

### When the Pointer Moves Past the End of a Line
In a DATA step that uses ODS, the number of columns in the buffer and in the data component are determined in one of three ways:

- By default, the number of variables in the program data vector determines the number of ODS columns.
- You can override the default by defining ODS columns with the `COLUMNS=` suboption in the FILE PRINT ODS statement.
- If you associate a template with the data component, then the specifications in the template take precedence. As a result, the number of columns that actually appear in the output object could change.

When using pointer controls and the `@` or `@@`, you might inadvertently position the pointer beyond the last ODS column. You can control how SAS handles this situation with options in the FILE PRINT ODS statement. For more information, see the discussion of overflow control on page 59.

**See Also**
- Output Delivery System and the DATA Step on page 53
- Examples on page 69

**Statement**
- “FILE Statement for ODS” on page 59
FILE Statement for ODS

Creates an ODS output object by binding the data component to the table template. As an option, the FILE Statement lists the variables to include in the ODS output, and it specifies options that control how the variables are formatted.

Valid in: DATA step
Category: File-Handling
Type: Executable
Default: ODS sends the output object to all open ODS destinations.
Note: This syntax shows only the ODS form of the FILE statement. For the complete syntax, see “FILE Statement” in SAS Statements: Reference.

Syntax

FILE PRINT ODS <=(ODS-suboption(s))> <options> ;

Required Arguments

PRINT
is a reserved fileref that you must use when you direct output to ODS.

Requirement You must use PRINT in a FILE statement that uses the ODS option.

See “Example 1: Creating a Report with the DATA Step and the Default Table Definition” on page 69

ODS<= (ODS-suboptions)>
defines the structure of the data component and binds the data component to a table template. The result is an ODS output object. ODS sends this object to all open ODS destinations.

See “ODS Suboptions” on page 60 for information about the ODS suboptions

Optional Arguments

N=number
specifies the number of lines that are available to the output pointer in the current iteration of the DATA step.

overflow-control
determines the PUT statement behavior when the output pointer attempts to move past the last ODS column in the buffer.

overflow-control is one of the following:

DROPOVER
discards items when a PUT statement attempts to write beyond the last ODS column in the buffer. A message in the log at the end of the DATA step informs you if data was not written to the buffer.
FLOWOVER
moves the output pointer to a new line if a PUT statement attempts to write an
data item beyond the last ODS column in the buffer. The PUT statement writes the
next item in the first ODS column of the new line.

STOPOVER
stops processing the DATA step immediately if a PUT statement attempts to
write beyond the last ODS column in the buffer. SAS discards the data item,
writes the portion of the buffer that was built before the error occurred, and
issues an error message.

Default: FLOWOVER

**ODS Suboptions**

**Table 3.3  ODS Suboptions**

<table>
<thead>
<tr>
<th>Task</th>
<th>Suboption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify one or more columns for the data component</td>
<td>COLUMNS= or VARIABLES= on page 60</td>
</tr>
<tr>
<td>Specify default values for dynamic-attribute values</td>
<td>DYNAMIC= on page 62</td>
</tr>
<tr>
<td>Specify whether all column definitions in the table template can be used by more than one variable</td>
<td>“GENERIC=ON</td>
</tr>
<tr>
<td>Specify a column heading to use for any column that does not have a column specified in the COLUMNS= or VARIABLES= suboption</td>
<td>LABEL= on page 63</td>
</tr>
<tr>
<td>Specify a name for the output object that the DATA step produces</td>
<td>OBJECT= on page 63</td>
</tr>
<tr>
<td>Specify a label for the output object that the DATA step produces</td>
<td>OBJECTLABEL= on page 64</td>
</tr>
<tr>
<td>Specify the table template to use with the data component to produce the output object</td>
<td>TEMPLATE= on page 64</td>
</tr>
</tbody>
</table>

**COLUMNS=(column-specification(s))**
specifies one or more columns for the data component and determines their order in
the data component.

Each column-specification associates a DATA step variable with a column that is
defined in the table template. column-specification has this general form:

\[
(column-name-1<=variable-name-1<(attribute-suboptions)>)> \\
<… column-name-n<=variable-name-n<(attribute-suboptions)> > >
\]

*column-name*
is the name of a column. This name must match the name that is defined in the
table template that you use.

**Restriction**  column-name must conform to the rules for SAS variable names.

**Requirement** You must enclose a column-name in parentheses.
**Tip**
You can use list notation (for example, `score1-score5`) to specify multiple column names.

**Example**
“Example 4: Creating and Using a User-Defined Table Template” on page 85

variable-name

specifies a variable in the DATA step to place in the specified column.

**Default**
If you omit variable-name, then ODS looks for a DATA step variable named column-name to place in the specified column. If no such variable exists, then ODS returns an error.

**Tip**
You can use list notation (for example, `score1-score5`) to specify a range of variable names.

**Example**
“Example 4: Creating and Using a User-Defined Table Template” on page 85

(attribute-suboptions)

assigns a characteristic, such as a label or a format, to a particular column in the data component. These individual specifications override any attributes that are set by the DATA step.

The following table lists the attribute suboptions that are available for the COLUMNS= suboption. For a complete description, see “Attribute Suboptions” on page 66.

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute Suboption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a value for the variable defined by the DYNAMIC statement in a table template</td>
<td>DYNAMIC= on page 66</td>
</tr>
<tr>
<td>Specify a format for the current column</td>
<td>FORMAT= on page 66</td>
</tr>
<tr>
<td>Specify whether the DATA step uses this column definition for multiple variables</td>
<td>GENERIC= on page 67</td>
</tr>
<tr>
<td>Specify a label for a particular column</td>
<td>LABEL= on page 67</td>
</tr>
</tbody>
</table>

**Requirement**
You must enclose attribute-suboptions in parentheses.

**Restrictions**
You can use only one COLUMNS= suboption in a FILE PRINT ODS statement.

You can use either the COLUMNS= suboption or the VARIABLES= suboption, but not both, in a single FILE PRINT ODS statement.

**Requirement**
You must enclose a column-specification in parentheses.

**Tips**
The order of the columns in the output object is determined by their order in the table template, not by their order in the data component.

To override the default order, use the ORDER_DATA= table attribute in the PROC TEMPLATE step that creates the definition. The default
DATA step table template uses this attribute. For more information, see the discussion of `ORDER_DATA=`.

If you do not specify `COLUMNS=` or `VARIABLES=`, then the order of columns in the data component matches the order of the corresponding variables in the program data vector.

**DYNAMIC=(dynamic-specification(s))**

specifies default values for dynamic-attribute values.

A dynamic-attribute value is defined in the table template. Its name serves as a placeholder for the value that is supplied to the data component with the `DYNAMIC=` suboption. When ODS creates the output object from the table template and the data component, it substitutes the appropriate value from the data component for the value's name in the table template.

Each *dynamic-specification* has the following form:

`dynamic-value-name <= variable-name | constant>`

- `dynamic-value-name` is the name that the table template gives to a dynamic-attribute value.
- `variable-name` specifies a variable whose value is assigned to `dynamic-value-name` and passed to ODS to substitute for the placeholder in the table template when it creates the output object.
- `constant` specifies a constant to assign to `dynamic-value-name` and pass to ODS to substitute for the placeholder in the table template when it creates the output object.

**Default**

By default, the `DYNAMIC=` suboption applies to all columns in the data component.

**Interaction**

Columns that do not contain their own `DYNAMIC=` suboption specifications use these `dynamic-specifications`.

**Tip**

You can override the default specification for an individual column by specifying the `DYNAMIC=` suboption as an attribute for that column in the `COLUMNS=` or the `VARIABLES=` suboption.

**See**


**GENERIC=ON | OFF**

indicates whether the DATA step uses all column definitions for multiple variables.

- **ON** indicates that the DATA step uses all column definitions for multiple variables.
- **OFF** indicates that the DATA step uses no column definitions for multiple variables.

**Defaults**

OFF

By default, the `GENERIC=` suboption applies to all columns in the data component.
Restriction

ODS does not recognize the column names as a match unless you specify the (COLUMNS=(GENERIC=ON)) suboption.

Interaction

If you do not specify a table template, the GENERIC= suboption is set to ON.

Tips

To override the default specification for an individual column, specify the GENERIC= suboption as an attribute for that column in the COLUMNS= or the VARIABLES= suboption.

The GENERIC= option in the DATA step is used in conjunction with the GENERIC= column attribute in the table template. See the GENERIC= column attribute in “Column Attributes” in *SAS Output Delivery System: Procedures Guide*.

<table>
<thead>
<tr>
<th><strong>LABEL=column-label</strong></th>
<th>specifies a label for any column that does not have a label specified in the COLUMNS= or VARIABLES= suboption.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default</strong></td>
<td>If you use the LABEL= suboption, ODS uses the first of these labels that it finds:</td>
</tr>
<tr>
<td></td>
<td>1. a label that is specified with the HEADER= attribute for a particular column in the table template (see HEADER=)</td>
</tr>
<tr>
<td></td>
<td>2. a label that is specified for a particular column with the LABEL= suboption in the COLUMNS= or VARIABLES= suboption</td>
</tr>
<tr>
<td></td>
<td>3. a label that is specified with the LABEL= suboption in the ODS= option</td>
</tr>
<tr>
<td></td>
<td>4. a label that is assigned with the LABEL statement in the DATA step</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>If you omit the LABEL= suboption, the contents of the table template determines whether the column heading contains the variable name or is blank.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>“Example 3: Assigning Attributes to Columns in ODS Output” on page 78</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>OBJECT= object-name</strong></th>
<th>specifies a name for the output object.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The Results window and the HTML contents file both contain a description of, and a link to, each output object. The description contains the first of the following items that ODS finds:</td>
</tr>
<tr>
<td></td>
<td>- the object's label</td>
</tr>
<tr>
<td></td>
<td>- the current title if it is not the default title, &quot;The SAS System&quot;</td>
</tr>
<tr>
<td></td>
<td>- the object's name</td>
</tr>
</tbody>
</table>
• the string **FilePrint##**, in which # increases by 1 for each DATA step that you run in the current SAS process without specifying an object name or an object label

**Restriction** *object-name* must conform to the rules for SAS variable names. For information about these rules, see “Rules for Words and Names in the SAS Language” in *SAS Language Reference: Concepts*.

**OBJECTLABEL=’object-label’**
specifies a label for the output object.

The Results window and the HTML contents file both contain a description of, and a link to, each output object. The description contains the first of the following items that ODS finds:

• the object's label
• the current title if it is not the default title, "The SAS System"
• the object's name (see OBJECT= on page 63)
• the string **FilePrint##**, in which # increases by 1 for each DATA step that you run in the current SAS process without specifying an object name or an object label

**Requirement** You must enclose an *object-label* in quotation marks.

**Example**

“Example 3: Assigning Attributes to Columns in ODS Output” on page 78

**TEMPLATE= ’table-definition-name’**
specifies the table definition to use with the data component to produce the output object.

*table-definition-name*

is the path to the table template. SAS stores a table template as an item in an item store.

**Defaults**

If you do not specify the TEMPLATE= option, ODS uses BASE.DATASETP.TABLE, the default table template.

If you do specify the TEMPLATE= suboption, ODS first looks for *table-definition-name* in Sasuser.Templat, and then it looks in Sashelp.Tmplmst.

**Requirement** You must enclose a *table-definition-name* in quotation marks.

**Interaction**

When you use the default table template, the GENERIC= suboption is set to ON for all columns in the data component. For more information, see GENERIC= on page 62.

**Tips**

When you use the BASE.DATASETP.TABLE template, character values are left-justified. If you want character values to be right-justified, specify the BASE.DATASETP.TABLENOJUST template.

You can change the locations in which ODS searches for the *table-definition-name* by using the ODS PATH on page 563 statement.
Example

“Example 4: Creating and Using a User-Defined Table Template” on page 85

**VARIABLES**=(variable-specification(s))

specifies one or more columns for the data component of the output object. Each variable-specification associates a DATA step variable with a column that is defined in the table template. The variable-specification value has this general form:

(variable-name-1<=column-name-1<(attribute-suboptions)> >
  <… variable-name-n<=column-name-n<(attribute-suboptions)> >>)

variable-name specifies a variable in the DATA step to use as a column in the data component.

**Tip**

You can use list notation (for example, _score1-score5_) to specify a range of variable names.

**Examples**

“Example 2: Producing ODS Output That Contains Selected Variables” on page 73

“Example 3: Assigning Attributes to Columns in ODS Output” on page 78

column-name is the name of a column. This name must match a name that is defined in the table template.

**Defaults**

If you are using the default table template and you omit column-name, then ODS uses the variable label to name the column. If the variable has no label, then ODS uses the variable name.

If you use a table template other than the default table template and you omit column-name, ODS looks in the table template for a column that is named variable-name and places the variable in that column. ODS returns an error if no such column exists.

**Restriction**

column-name must match a column name in the table template that you are using. It must also conform to the rules for SAS variable names. For information about these rules, see “Rules for Words and Names in the SAS Language” in *SAS Language Reference: Concepts*.

**Tip**

You can use list notation (for example, _score1-score5_) to specify a range of column names.

(attribute-suboptions)

assigns a characteristic, such as a label or a format, to a particular column in the data component. These individual specifications override any attributes that are set in the DATA step for the entire data component.

The following table lists the attribute suboptions available for the VARIABLES= suboption. For a complete description, see “Attribute Suboptions” on page 66.

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute Suboption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a value for the variable defined by the DYNAMIC statement in a table template</td>
<td>DYNAMIC= on page 66</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute Suboption</td>
</tr>
<tr>
<td>------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Specify a format for the current column</td>
<td>FORMAT= on page 66</td>
</tr>
<tr>
<td>Specify whether the DATA step uses this column definition for multiple variables</td>
<td>GENERIC= on page 67</td>
</tr>
<tr>
<td>Specify a label for a particular column</td>
<td>LABEL= on page 67</td>
</tr>
</tbody>
</table>

**Default**

If you specify the VARIABLES= suboption, the order of the columns in the output object is determined by their order in the table template, not by their order in the data component. If you do not specify COLUMNS= or VARIABLES= suboptions, the order of columns in the data component matches the order of the corresponding variables in the program data vector.

**Restrictions**

You can use only one VARIABLES= suboption in a FILE PRINT ODS statement.

You can use either the COLUMNS= suboption or the VARIABLES= suboption to associate variables with columns, but you cannot use both suboptions in the same FILE PRINT ODS statement.

**Tips**

To override the default order, use the ORDER_DATA table attribute in the PROC TEMPLATE step that creates the definition. The default DATA step table template uses this attribute. For more information, see the ORDER_DATA=.

The VARIABLES= suboption is for use primarily with the default DATA step table template. When you use the default definition, the DATA step can map variables to the appropriate column in the definition so that you do not need to specify a column name.

**Examples**

“Example 2: Producing ODS Output That Contains Selected Variables” on page 73

“Example 3: Assigning Attributes to Columns in ODS Output” on page 78

**Attribute Suboptions**

**DYNAMIC=dynamic-specification(s)**

specifies a value for the variable defined by the DYNAMIC statement in a table template.

See DYNAMIC= suboption on page 62

**Example**

“Example 4: Creating and Using a User-Defined Table Template” on page 85

**FORMAT=format-name**

specifies a format for the current column.
Default ODS uses the first of these formats for the variable that it finds:

1. for nongeneric columns, a format that is specified in the column definition
2. a format that is specified in the FORMAT= column attribute
3. a format that is specified in a FORMAT statement
4. the default format ($w. for character variables; BEST12. for numeric variables)

Note Formats for generic columns that are specified in the table template are ignored by the DATA step interface to ODS.

Example “Example 4: Creating and Using a User-Defined Table Template” on page 85

**GENERIC=ON | OFF**
specifies whether the DATA step uses this column definition for multiple variables.

Default OFF

Tip The GENERIC= option in the DATA step is used in conjunction with the GENERIC= column attribute in the table template. See the GENERIC= column attribute in “Column Attributes” in *SAS Output Delivery System: Procedures Guide*.

See GENERIC= suboption on page 62

Example “Example 4: Creating and Using a User-Defined Table Template” on page 85

**LABEL=column-label**
specifies a label for the specified column.

See LABEL= suboption on page 63

“Example 3: Assigning Attributes to Columns in ODS Output” on page 78

**Details**

**Restrictions When Using the FILE Statement with ODS**
The following restrictions apply to the FILE statement when you use it with ODS:

- These arguments affect only listing output:
  - FOOTNOTES and NOFOOTNOTES
  - LINESIZE
  - PAGESIZE
  - TITLE and NOTITLES
- Do not use these arguments:
  - DELIMITER=

Using Options and Suboptions

Options apply to all columns and suboptions apply to specific columns.

For example, both of the following DATA steps produce the same output. This DATA step specifies the suboption GENERIC=ON for every column.

Example Code 3.1 DATA Step Using the GENERIC=ON Suboption

data _null_;  
set top3list;  
file print ods = (  
template='means.topn'  
columns={  
  class=school(generic=on)  
  class=year(generic=on)  
  sum=moneyRaised_sum(generic=on)  
  mean=moneyRaised_mean(generic=on)  
  raised=moneyRaised_1(generic=on)  
  raised=moneyRaised_2(generic=on)  
  raised=moneyRaised_3(generic=on)  
  name=name_1(generic=on)  
  name=name_2(generic=on)  
  name=name_3(generic=on)  
  school=school_1(generic=on)  
  school=school_2(generic=on)  
  school=school_3(generic=on)  
  year=year_1(generic=on)  
  year=year_2(generic=on)  
  year=year_3(generic=on)  
}  
);  
put _ods_;  
run;  

This DATA step uses the GENERIC=ON option, which has to be specified only once.

Example Code 3.2 DATA Step Using the GENERIC=ON Option

data _null_;  
set top3list;  
file print ods = (  
template='means.topn'  
generic=on  
columns={  
  class=school  
  class=year  
  sum=moneyRaised_sum  
);
Without ODS Suboptions

If you do not specify any ODS suboptions, the DATA step uses a default table template (BASE.DATASET.TABLE) that is stored in the Sashelp.Tmplmst template store. This definition defines two generic columns: one for character variables and one for numeric variables. ODS associates each variable in the DATA step with one of these columns and displays the variables in the order in which they are defined in the DATA step.

If there are no suboptions, the default table template uses the variable's label as its column heading. If no label exists, the definition uses the variable's name as the column heading.

See Also

• Chapter 3, “Using ODS with the DATA Step,” on page 53
• Examples on page 69

Statement

• “PUT Statement for ODS ” on page 55

Examples

Example 1: Creating a Report with the DATA Step and the Default Table Definition

Features:

<table>
<thead>
<tr>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE PRINT ODS statement</td>
</tr>
<tr>
<td>PUT <em>ODS</em> statement</td>
</tr>
</tbody>
</table>

ODS destination: HTML
Details

This example uses the DATA step and ODS to create an HTML report. It uses the default table template for the DATA step and writes an output object to the HTML destination (the default).

Program

```sas
options obs=15;
title 'Leading Grain Producers';
proc format;
  value $cntry 'BRZ'='Brazil'
    'CHN'='China'
    'IND'='India'
    'INS'='Indonesia'
    'USA'='United States';
run;
data _null_;
  length Country $ 3 Type $ 5;
  format country $cntry.;
  label type='Grain';
  input Year country $ type $ Kilotons;
    file print ods;
    put _ods_;
    datalines;
1995 BRZ Wheat    1516
1995 BRZ Rice     11236
1995 BRZ Corn     36276
1995 CHN Wheat    102207
1995 CHN Rice     185226
1995 CHN Corn     112331
1995 IND Wheat    63007
1995 IND Rice     122372
1995 IND Corn     9800
1995 INS Wheat    .
1995 INS Rice     49860
1995 INS Corn     8223
1995 USA Wheat    59494
1995 USA Rice     7888
1995 USA Corn     187300
1996 BRZ Wheat    3302
1996 BRZ Rice     10035
1996 BRZ Corn     31975
1996 CHN Wheat    109000
1996 CHN Rice     190100
1996 CHN Corn     119350
1996 IND Wheat    62620
1996 IND Rice     120012
1996 IND Corn     8660
1996 INS Wheat    .
1996 INS Rice     51165
```

Chapter 3 • Using ODS with the DATA Step
Program Description

Set the SAS system options. The OBS= option specifies the number of observations to be printed.

    options obs=15;

Specify a title. The TITLE statement specifies a title for the output.

    title 'Leading Grain Producers';

Create a user-defined format. PROC FORMAT creates the format $CNTRY. for the variable COUNTRY.

    proc format;
    value $cntry 'BRZ'='Brazil'
       'CHN'='China'
       'IND'='India'
       'INS'='Indonesia'
       'USA'='United States';
    run;

Begin a DATA step that does not create an output data set. Using _NULL_ saves computer resources because it prevents the DATA step from creating an output data set.

    data _null_;

Define variables, assign lengths and formats, read a record, and assign values to four variables. The LENGTH statement defines a length that is shorter than the default to two character variables. The FORMAT statement assigns a user-defined format to the variable COUNTRY. The LABEL statement assigns a label to the variable TYPE. The INPUT statement reads a record from the data lines and assigns a value to four variables.

    length Country $ 3 Type $ 5;
    format country $cntry.;
    label type='Grain';
    input Year country $ type $ Kilotons;

Use the default table template to create HTML output. The combination of the fileref PRINT and the ODS option in the FILE statement routes the DATA step output to ODS. The only open ODS destination is the HTML destination, which is open by default when you begin your SAS session. Because no suboptions are specified, ODS uses the default DATA step table template. This FILE PRINT ODS statement creates an output object and binds it to the default template.

    file print ods;

Write the variables to the data component. The _ODS_ option in the PUT statement writes every variable to the buffer that the PUT statement writes to the data component.
Because no formats or labels are specified for individual columns, ODS uses the defaults.

```plaintext
put _ods_;
```

**The data provides information about the amounts of wheat, rice, and corn that five leading grain-producing nations produced during 1995 and 1996.**

```plaintext
datalines;
1995 BRZ Wheat 1516
1995 BRZ Rice 11236
1995 BRZ Corn 36276
1995 CHN Wheat 102207
1995 CHN Rice 185226
1995 CHN Corn 112331
1995 IND Wheat 63007
1995 IND Rice 122372
1995 IND Corn 9800
1995 INS Wheat .
1995 INS Rice 49860
1995 INS Corn 8223
1995 USA Wheat 59494
1995 USA Rice 7888
1995 USA Corn 187300
1996 BRZ Wheat 3302
1996 BRZ Rice 10035
1996 BRZ Corn 31975
1996 CHN Wheat 109000
1996 CHN Rice 190100
1996 CHN Corn 119350
1996 IND Wheat 62620
1996 IND Rice 120012
1996 IND Corn 8660
1996 INS Wheat .
1996 INS Rice 51165
1996 INS Corn 8925
1996 USA Wheat 62099
1996 USA Rice 7771
1996 USA Corn 236064
;
run;
```

**HTML Output**

The default table template produces a column for each variable in the DATA step. The order of the columns is determined by their order in the program data vector. Because no attributes are specified for individual columns, ODS uses the default column headings and formats.
Example 2: Producing ODS Output That Contains Selected Variables

Features:
FILE PRINT ODS statement:
   VARIABLES= suboption

ODS PDF statement:
   FILE= option

PUT _ODS_ statement

Format: $CNTRY.

ODS destinations: HTML, PRINTER (PDF)

Details
This example selects variables to include in the output. The resulting output is produced in two formats, PDF and HTML. The HTML output is produced by default, and the PDF output is requested by the ODS PDF statement. This example uses filenames that might not be valid in all operating environments. To successfully run the example in your operating environment, you might need to change the file specifications. See Appendix 4, “ODS HTML Statements for Running Examples in Different Operating Environments,” on page 1113.
Program

options nodate pageno=1;
ods pdf file='your-pdf-file.pdf';
title 'Leading Grain Producers';
title2 'for 1996';
data _null_;

length Country $ 3 Type $ 5;
format country $cntry.;
label type='Grain';

input Year country $ type $ Kilotons;
if year=1996;
file print ods=(variables=(country
    type
    kilotons));
put _ods_;

datalines;
1995 BRZ  Wheat  1516
1995 BRZ  Rice   11236
1995 BRZ  Corn   36276
1995 CHN  Wheat  102207
1995 CHN  Rice   185226
1995 CHN  Corn   112331
1995 IND  Wheat  63007
1995 IND  Rice   122372
1995 IND  Corn   9800
1995 INS  Wheat  .
1995 INS  Rice   49860
1995 INS  Corn   8223
1995 USA  Wheat  59494
1995 USA  Rice   7888
1995 USA  Corn   187300
1996 BRZ  Wheat  3302
1996 BRZ  Rice   10035
1996 BRZ  Corn   31975
1996 CHN  Wheat  109000
1996 CHN  Rice   190100
1996 CHN  Corn   119350
1996 IND  Wheat  62620
1996 IND  Rice   120012
1996 IND  Corn   8660
1996 INS  Wheat  .
1996 INS  Rice   51165
1996 INS  Corn   8925
1996 USA  Wheat  62099
1996 USA  Rice   7771
1996 USA  Corn   236064
;
run;
ods pdf close;
**Program Description**

**Set the SAS system options.** The NODATE option suppresses the display of the date and time in the output. The PAGENO= option specifies the starting page number. None of these options affects the HTML output.

```
options nodate pageno=1;
```

**Specify that you want ODS to create PDF output and store it in the specified file.** The ODS PDF statement opens the PDF destination. Any procedure or DATA step output that is created is routed to this destination (and any others that are open) and is, therefore, formatted as PDF. The FILE= option sends all output objects to the PDF file that you specify.

```
ods pdf
  file='your-pdf-file.pdf';
```

**Specify the titles.** The TITLE statements provide titles for the output.

```
title 'Leading Grain Producers';
title2 'for 1996';
```

**Begin a DATA step that does not create an output data set.** Using _NULL_ saves computer resources because it prevents the DATA step from creating an output data set.

```
data _null_;
```

**Assign lengths other than the default to two character variables. Also assign a user-defined format to one variable and a label to another.** The FORMAT statement assigns a format to the variable COUNTRY. The LABEL statement assigns a label to the variable TYPE.

```
length Country $ 3 Type $ 5;
format country $cntry.;
label type='Grain';
```

**Read a record from the input data, assign values to four variables. Continue to process only observations that match the criterion.** The INPUT statement reads a single record and assigns values to four variables. The subsetting IF statement causes the DATA step to continue to process only those observations that contain the value 1996 for YEAR.

```
input Year country $ type $ Kilotons;
if year=1996;
```

**Send the DATA step output to whatever ODS destinations are open. Specify the variables and their order in the data component that is created.** The combination of the fileref PRINT and the ODS option in the FILE statement sends the results of the DATA step to ODS. Two ODS destinations, the PDF and the HTML destinations, are open. Because no table template is specified, ODS uses the default DATA step definition. The VARIABLES= suboption specifies that the resulting data component contains three columns in the order that is listed.

```
file print ods=(variables=(country type kilotons));
```
Write values for all variables that are specified with the VARIABLES= suboption in the FILE statement. The _ODS_ option in the PUT statement writes variable values to the data component. It writes only those variables that were specified with the VARIABLES= suboption in the FILE statement. Because no formats or labels are specified for these ODS columns, ODS uses the defaults.

```put _ods_;```

The data provides information about the amounts of wheat, rice, and corn that were produced by the five leading grain-producing nations during 1995 and 1996.

data datalines;
1995 BRZ Wheat 1516
1995 BRZ Rice 11236
1995 BRZ Corn 36276
1995 CHN Wheat 102207
1995 CHN Rice 185226
1995 CHN Corn 112331
1995 IND Wheat 63007
1995 IND Rice 122372
1995 IND Corn 9800
1995 INS Wheat .
1995 INS Rice 49860
1995 INS Corn 8223
1995 USA Wheat 59494
1995 USA Rice 7888
1995 USA Corn 187300
1996 BRZ Wheat 3302
1996 BRZ Rice 10035
1996 BRZ Corn 31975
1996 CHN Wheat 109000
1996 CHN Rice 190100
1996 CHN Corn 119350
1996 IND Wheat 62620
1996 IND Rice 120012
1996 IND Corn 8660
1996 INS Wheat .
1996 INS Rice 51165
1996 INS Corn 8925
1996 USA Wheat 62099
1996 USA Rice 7771
1996 USA Corn 236064
; run;
```

Close the PDF destination so that you can view the output. The ODS PDF statement closes the PDF destination and all the files that are associated with it. You must close the destination before you can view the output. Also, closing the destination prevents all subsequent ODS jobs from automatically producing PDF output.

```ods pdf close;```
Output 3.2  HTML Body File Produced by ODS

<table>
<thead>
<tr>
<th>Country</th>
<th>Grain</th>
<th>Kilotons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil</td>
<td>Wheat</td>
<td>3302</td>
</tr>
<tr>
<td>Brazil</td>
<td>Rice</td>
<td>10635</td>
</tr>
<tr>
<td>Brazil</td>
<td>Corn</td>
<td>31975</td>
</tr>
<tr>
<td>China</td>
<td>Wheat</td>
<td>169000</td>
</tr>
<tr>
<td>China</td>
<td>Rice</td>
<td>190100</td>
</tr>
<tr>
<td>China</td>
<td>Corn</td>
<td>119350</td>
</tr>
<tr>
<td>India</td>
<td>Wheat</td>
<td>62020</td>
</tr>
<tr>
<td>India</td>
<td>Rice</td>
<td>120012</td>
</tr>
<tr>
<td>India</td>
<td>Corn</td>
<td>86660</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Wheat</td>
<td>.</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Rice</td>
<td>51165</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Corn</td>
<td>8925</td>
</tr>
<tr>
<td>United States</td>
<td>Wheat</td>
<td>62099</td>
</tr>
<tr>
<td>United States</td>
<td>Rice</td>
<td>7771</td>
</tr>
<tr>
<td>United States</td>
<td>Corn</td>
<td>200054</td>
</tr>
</tbody>
</table>
Example 3: Assigning Attributes to Columns in ODS Output

**Features:**
- FILE PRINT ODS statement:
  - OBJECTLABEL= suboption
  - VARIABLES= suboption
  - LABEL= suboption
FORMAT= suboption

PUT _ODS_ statement

**Format:** $\text{CNTRY.}

**ODS destinations:** HTML, RTF, PRINTER (PDF)

**Details**

This example assigns a label to the output object that it creates. It also specifies a label and a format for individual columns. This example uses filenames that might not be valid in all operating environments. To successfully run the example in your operating environment, you might need to change the file specifications. See Appendix 4, “ODS HTML Statements for Running Examples in Different Operating Environments,” on page 1113.

**Program**

```plaintext
options nodate pageno=1;
ods html body='your_body_file.html'
contents='your_contents_file.html'
frame='your_frame_file.html';
ods printer
file='your_pdf_file.pdf';
title 'Leading Grain Producers';
title2 'for 1996';
data _null_;

length Country $ 3 Type $ 5;
format country $cntry.;
label type='Grain';
input Year country $ type $ Kilotons;
if year=1996;
file print ods= (objectlabel='1996 Grain Production'
variables=(country
    type(label='Type of Grain')
    kilotons(format=comma12.)));
put _ods_;
datalines;
1995 BRZ Wheat 1516
1995 BRZ Rice 11236
1995 BRZ Corn 36276
1995 CHN Wheat 102207
1995 CHN Rice 185226
1995 CHN Corn 112331
1995 IND Wheat 63007
1995 IND Rice 122372
1995 IND Corn 9800
```

Example 3: Assigning Attributes to Columns in ODS Output 79
options nodate pageno=1;

Program Description

Set the SAS system options. The NODATE option suppresses the display of the date and time in the output. The PAGENO= option specifies the starting page number. These options affect the PDF output, but none of them affects the HTML output.

```
options nodate pageno=1;
```

Specify that you want to create HTML output. Also specify where to store the HTML output: the body file, the contents file, and the frame file. The ODS HTML statement opens the HTML destination and creates HTML output. The BODY= option identifies the file that contains the HTML output. The CONTENTS= option identifies the file that contains a table of contents to the HTML output. The contents file links to the body file. The FRAME= option identifies the file that integrates the table of contents, the page contents, and the body file. If you open the frame file, you see a table of contents, a table of pages, or both, as well as the body file.

```
ods html body='your_body_file.html'
contents='your_contents_file.html'
frame='your_frame_file.html';
```

Specify that you want PDF output. Also specify where to store the PDF output. The ODS PDF statement opens the PRINTER destination and creates PDF output. The FILE= option sends all output objects to the external file in the current directory.

```
ods printer
file='your_pdf_file.pdf';
```
Specify the titles. The TITLE statements provide titles for the output.

```plaintext
title 'Leading Grain Producers';
title2 'for 1996';
```

Begin a DATA step that does not create an output data set. Using _NULL_ saves computer resources because it prevents the DATA step from creating an output data set.

```plaintext
data _null_;
```

Assign lengths other than the default to two character variables. Also assign a user-defined format to one variable and a label to another. The LENGTH statement assigns lengths to COUNTRY and TYPE. The FORMAT statement assigns a format to the variable COUNTRY. The LABEL statement assigns a label to the variable TYPE.

```plaintext
length Country $ 3 Type $ 5;
format country $cntry.;
label type='Grain';
```

Read a record from the input data, assign values to four variables. Continue to process only observations that match the criterion. The INPUT statement reads a single record and assigns values to four variables. The subsetting IF statement causes the DATA step to continue to process only those observations that contain the value 1996 for YEAR.

```plaintext
input Year country $ type $ Kilotons;
if year=1996;
```

Send the DATA step output to the open destinations, specify a label for the output object, and specify the variables to write to the data component and the order in which to write them. The combination of the fileref PRINT and the ODS option in the FILE statement sends the results of the DATA step to ODS. The LISTING, the HTML, and the PRINTER destinations are open. Because no table template is specified, ODS uses the default DATA step definition. The OBJECTLABEL= suboption specifies the label '1996 Grain Production' to the output object. This label appears in the Results folder and in the HTML contents file. The VARIABLES= suboption specifies the variables to write to the data component and the order in which to write them. The LABEL= suboption specifies a label for the variable TYPE. The label specified here takes precedence over the LABEL statement assignment that was made previously in the DATA step, so it is used as the column heading for TYPE. The FORMAT= suboption assigns a format for the variable KILOTONS.

```plaintext
file print ods= (objectlabel='1996 Grain Production'
               variables=(country
type(label='Type of Grain')
kilotons(format=comma12.))
               );
```

Write the variables to the buffer. The _ODS_ option in the PUT statement writes all of the variables that are defined to ODS (in the FILE PRINT ODS statement) to a special buffer. It uses default attributes for COUNTRY, and it uses any attributes specified in the VARIABLES= suboption for the other variables. For attributes that might be specified elsewhere in the DATA step but are not specified in VARIABLES=, it uses the defaults.

```plaintext
put _ods_;
The data provides information about the amounts of wheat, rice, and corn that five leading grain-producing nations produced during 1995 and 1996.

data;
1995 BRZ Wheat 1516
1995 BRZ Rice 11236
1995 BRZ Corn 36276
1995 CHN Wheat 102207
1995 CHN Rice 185226
1995 CHN Corn 112331
1995 IND Wheat 63007
1995 IND Rice 122372
1995 IND Corn 9800
1995 INS Wheat.
1995 INS Rice 49860
1995 INS Corn 8223
1995 USA Wheat 59494
1995 USA Rice 7888
1995 USA Corn 187300
1996 BRZ Wheat 3302
1996 BRZ Rice 10035
1996 BRZ Corn 31975
1996 CHN Wheat 109000
1996 CHN Rice 190100
1996 CHN Corn 119350
1996 IND Wheat 62620
1996 IND Rice 120012
1996 IND Corn 8660
1996 INS Wheat.
1996 INS Rice 51165
1996 INS Corn 8925
1996 USA Wheat 62099
1996 USA Rice 7771
1996 USA Corn 236064
;
run;

To view the HTML output and the PDF output, close both the HTML and PRINTER destinations. This statement closes the LISTING, HTML, and PRINTER destinations and all the files that are associated with them. You must close the HTML destination before you can view the output with a browser. You must close the PRINTER destination before you can print the output on a physical printer. If you do not close these destinations, then output created in subsequent sessions is routed to them, and you might inadvertently continue to generate both HTML and PDF output.

ods _all_ close;

Output

In this HTML frame file, the object's label, '1996 Grain Production' was supplied by the OBJECTLABEL= suboption. It appears in the table of contents as the link to the output object. In the body file, the label 'Type of Grain' that was supplied by the LABEL=
Example 3: Assigning Attributes to Columns in ODS Output

The suboption for the variable TYPE becomes its column heading. The format for KILOTONS was supplied by the FORMAT= suboption in the FILE statement.

Output 3.4  HTML Frame File Produced by ODS

Leading Grain Producers for 1996

<table>
<thead>
<tr>
<th>Country</th>
<th>Type of Grain</th>
<th>Kilotons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil</td>
<td>Wheat</td>
<td>3,302</td>
</tr>
<tr>
<td>Brazil</td>
<td>Rice</td>
<td>10,035</td>
</tr>
<tr>
<td>Brazil</td>
<td>Corn</td>
<td>31,975</td>
</tr>
<tr>
<td>China</td>
<td>Wheat</td>
<td>109,000</td>
</tr>
<tr>
<td>China</td>
<td>Rice</td>
<td>190,100</td>
</tr>
<tr>
<td>China</td>
<td>Corn</td>
<td>119,350</td>
</tr>
<tr>
<td>India</td>
<td>Wheat</td>
<td>62,620</td>
</tr>
<tr>
<td>India</td>
<td>Rice</td>
<td>120,012</td>
</tr>
<tr>
<td>India</td>
<td>Corn</td>
<td>8,660</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Wheat</td>
<td></td>
</tr>
<tr>
<td>Indonesia</td>
<td>Rice</td>
<td>51,165</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Corn</td>
<td>8,925</td>
</tr>
<tr>
<td>United States</td>
<td>Wheat</td>
<td>62,099</td>
</tr>
<tr>
<td>United States</td>
<td>Rice</td>
<td>7,771</td>
</tr>
<tr>
<td>United States</td>
<td>Corn</td>
<td>236,064</td>
</tr>
</tbody>
</table>

Just as in the HTML body file and in the LISTING output, the PDF output displays the label 'Type of Grain' that was supplied by the LABEL= suboption for the variable TYPE.
as its column heading. The format for KILOTONS was supplied by the FORMAT= suboption in the FILE statement.

Output 3.5  PDF Output

<table>
<thead>
<tr>
<th>Country</th>
<th>Type of Grain</th>
<th>Kilotons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil</td>
<td>Wheat</td>
<td>3,302</td>
</tr>
<tr>
<td>Brazil</td>
<td>Rice</td>
<td>10,035</td>
</tr>
<tr>
<td>Brazil</td>
<td>Corn</td>
<td>31,975</td>
</tr>
<tr>
<td>China</td>
<td>Wheat</td>
<td>109,000</td>
</tr>
<tr>
<td>China</td>
<td>Rice</td>
<td>190,100</td>
</tr>
<tr>
<td>China</td>
<td>Corn</td>
<td>119,350</td>
</tr>
<tr>
<td>India</td>
<td>Wheat</td>
<td>62,620</td>
</tr>
<tr>
<td>India</td>
<td>Rice</td>
<td>120,012</td>
</tr>
<tr>
<td>India</td>
<td>Corn</td>
<td>8,660</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Wheat</td>
<td></td>
</tr>
<tr>
<td>Indonesia</td>
<td>Rice</td>
<td>51,165</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Corn</td>
<td>8,925</td>
</tr>
<tr>
<td>United States</td>
<td>Wheat</td>
<td>62,099</td>
</tr>
<tr>
<td>United States</td>
<td>Rice</td>
<td>7,771</td>
</tr>
<tr>
<td>United States</td>
<td>Corn</td>
<td>236,064</td>
</tr>
</tbody>
</table>

Just as in the HTML body file and the PDF output, the LISTING output displays the label 'Type of Grain' that was supplied by the LABEL= suboption for the variable TYPE.
The format for KILOTONS was supplied by the FORMAT= suboption in the FILE statement.

Output 3.6  RTF Output

Leading Grain Producers for 1996

<table>
<thead>
<tr>
<th>Country</th>
<th>Type of Grain</th>
<th>Kilotons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil</td>
<td>Wheat</td>
<td>3,302</td>
</tr>
<tr>
<td>Brazil</td>
<td>Rice</td>
<td>10,035</td>
</tr>
<tr>
<td>Brazil</td>
<td>Corn</td>
<td>31,975</td>
</tr>
<tr>
<td>China</td>
<td>Wheat</td>
<td>109,000</td>
</tr>
<tr>
<td>China</td>
<td>Rice</td>
<td>190,100</td>
</tr>
<tr>
<td>China</td>
<td>Corn</td>
<td>119,350</td>
</tr>
<tr>
<td>India</td>
<td>Wheat</td>
<td>62,620</td>
</tr>
<tr>
<td>India</td>
<td>Rice</td>
<td>120,012</td>
</tr>
<tr>
<td>India</td>
<td>Corn</td>
<td>8,660</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Wheat</td>
<td></td>
</tr>
<tr>
<td>Indonesia</td>
<td>Rice</td>
<td>51,165</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Corn</td>
<td>8,923</td>
</tr>
<tr>
<td>United States</td>
<td>Wheat</td>
<td>62,099</td>
</tr>
<tr>
<td>United States</td>
<td>Rice</td>
<td>7,771</td>
</tr>
<tr>
<td>United States</td>
<td>Corn</td>
<td>236,064</td>
</tr>
</tbody>
</table>

Example 4: Creating and Using a User-Defined Table Template

Features: PROC TEMPLATE
FILE PRINT ODS statement:
  COLUMNS= suboption:
  FORMAT= suboption
  DYNAMIC= suboption
  GENERIC= suboption
  TEMPLATE=
PUT _ODS_ statement:
column pointer controls
line pointer controls

ODS destination: RTF
Details

This example shows how to do the following:

• create a simple user-defined template (table definition) with PROC TEMPLATE
• use a simple user-defined template in the DATA step
• use pointer controls in the PUT _ODS_ statement

Program: Creating the User-Defined Table Template

```sas
proc template;
define table phonelist;
  column name phone;
  dynamic colheader;
  define name;
    generic=on;
    header=colheader;

    style=data{fontstyle=italic fontsize=5};
  end;

  define phone;
    header='Telephone';
    style=datafixed;
  end;
end;
end;
run;
ods html close;
ods rtf body='your_rtf_file.rtf';
title 'New Subscriber Telephone List';
proc format;
  picture phonenum .='Not available'
    other='0000)000-0000' (prefix='(');
run;
data phones;
length first_name $20 last_name $25;
input first_name $ last_name $ business_phone home_phone;
datalines;
Jerome Johnson 9193191677 9198462198
Romeo Montague 8008992164 3609736201
Imani Rashid 5088522146 5083669821
Palinor Kent . 9197823199
Ruby Archuleta . .
Takei Ito 7042982145 .
Tom Joad 2099632764 2096684741;
proc sort data=phones;
  by last_name;
run;
data _null_;
  set phones;
```

file print
ods=(template='phonelist'
        columns=
            (name=last_name
             (generic=on
              dynamic=(colheader='Last Name'))
            name=first_name
             (generic=on
              dynamic=(colheader='First Name'))
            phone=business_phone
             (format=phonenum.)
        )
    );

if (missing(business_phone)) then
  put _ods_ @3 home_phone;
else if (missing(home_phone)) then
  put _ods_;
else
  put _ods_ / @3 home_phone;
run;
ods RTF close;

Program Description

Define the table template PHONELIST. This PROC TEMPLATE step defines a table template named PHONELIST. The template defines two columns: NAME and PHONE. The GENERIC=ON attribute defines the column for NAME as one that the DATA step can use for multiple variables. The column definition uses dynamic headers. That is, a variable that uses this column definition takes the value of the header at run time from the DATA step that uses this template. Thus, each variable can have a different column heading. The STYLE= attribute specifies that the style element DATA be used as the basis for generating the data in this column. The font face and font size that DATA normally uses are replaced by the ones that are specified in the STYLE= attribute. The header for PHONE is hardcoded as Telephone. The STYLE= attribute specifies a style element to use for the data in this column.

proc template;
  define table phonelist;
    column name phone;
    dynamic colheader;
  define name;
    generic=on;
    header=colheader;

    style=data{fontstyle=italic fontsize=5};
  end;

  define phone;
    header='Telephone';
    style=datafixed;
  end;
end;
run;
Specify that you do not want to produce the default HTML output. The ODS HTML CLOSE statement closes the HTML destination to conserve resources. The HTML destination is open by default when you open your SAS session.

    ods html close;

Specify that you want the output formatted in RTF. The ODS RTF statement opens the RTF destination and creates RTF output for use by Microsoft Word. Subsequent output objects are sent to the body file.

    ods rtf body='your_rtf_file.rtf';

Specify a title. The TITLE statement provides a title for the output.

    title 'New Subscriber Telephone List';

Create a format for telephone numbers. PROC FORMAT creates a user-defined format for telephone numbers.

    proc format;
      picture phonenum .='Not available'
        other='0000)000-0000' (prefix='(');
    run;

Create the PHONES data set. The data set PHONES contains names and their corresponding phone numbers. Some observations contain missing values for the business or home phone numbers.

    data phones;
      length first_name $20 last_name $25;
      input first_name $ last_name $ business_phone home_phone;
    datalines;
    Jerome Johnson 9193191677 9198462198
    Romeo Montague 8008992164 3609736201
    Imani Rashid 5088522146 5083669821
    Palinor Kent . 9197823199
    Ruby Archuleta . .
    Takei Ito 7042982145 .
    Tom Joad 2099632764 2096684741
    ;

Sort the PHONES data set by last name. PROC SORT sorts the data set PHONES by LAST_NAME and replaces the original data set with the sorted data set.

    proc sort data=phones;
      by last_name;
    run;

Begin a DATA step that does not create an output data set. Read an observation from the PHONES data set. Using _NULL_ saves computer resources because it prevents the DATA step from creating an output data set.

    data
      _null_
    set phones;

Request that ODS output be created and use the template named PHONELIST. The combination of the fileref PRINT and the ODS option in the FILE statement sends the
results of the DATA step to ODS. ODS creates an output object and binds it to the PHONELIST template. Only RTF output is created because only the RTF destination is open. The TEMPLATE= suboption tells ODS to use the template PHONELIST, which was created previously in the PROC TEMPLATE step.

```
file print
ods=(template='phonelist'

Place variable values in columns. The COLUMNS= suboption places values of variables into columns that are defined in the template. Values for both the LAST_NAME and FIRST_NAME variables are written to columns that are defined as NAME in the template. The GENERIC=ON suboption must be set in both the template and the ODS= option in order for you to use a column definition for more than one column. The value of the variable BUSINESS_PHONE is placed in a column that is defined as PHONE. The DYNAMIC= suboption assigns a value to the variable COLHEADER. This value is passed to the template when the output object is created, and the template uses it for the column heading. Thus, even though the variables use the same column definition from the template, the columns in the output object have different column headings. The FORMAT= suboption assigns the format PHONENUM. to the column named PHONE.

```
columns=
(name=last_name
  (generic=on
   dynamic=(colheader='Last Name'))
  name=first_name
  (generic=on
   dynamic=(colheader='First Name'))
  phone=business_phone
  (format=phonenum.)
);```

The following IF/THEN-ELSE statements execute a different PUT _ODS_ statement based on the specified conditions. If BUSINESS_PHONE contains missing values, then the PUT statement writes values for LAST_NAME, FIRST_NAME, and BUSINESS_PHONE (the columns that are defined in the ODS= option) into the output buffer. The PUT statement then writes the value for HOME_PHONE in column 3, overwriting the missing value of BUSINESS_PHONE. If HOME_PHONE contains a missing value, then the PUT statement simply writes values for LAST_NAME, FIRST_NAME, and BUSINESS_PHONE to the buffer. Finally, if both phone numbers have values, then the PUT statement writes values for LAST_NAME, FIRST_NAME, and BUSINESS_PHONE to the buffer in the first line. SAS then goes to the next line (as directed by the line pointer control / ) and writes the value of HOME_PHONE in the third column of the next line.

```
if
  (missing(business_phone)) then
    put _ods_ @3 home_phone;
else if (missing(home_phone)) then
    put _ods_;
else
    put _ods_ / @3 home_phone;
run;
```

Close the RTF destination so that you can view the output. The ODS RTF statement closes the RTF destination and all the files that are associated with it. You must close the
destination before you can view the output in Microsoft Word. Also, closing the output prevents all subsequent ODS jobs from automatically producing RTF output.

```ods RTF close;```

**RTF Output**

**Output 3.7**  RTF Output Viewed with Microsoft Word

---

**New Subscriber Telephone List**

<table>
<thead>
<tr>
<th>Last Name</th>
<th>First Name</th>
<th>Telephone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archuleta</td>
<td>Ruby</td>
<td>Not available</td>
</tr>
<tr>
<td>Ito</td>
<td>Takei</td>
<td>(704)298-2145</td>
</tr>
<tr>
<td>Joad</td>
<td>Tom</td>
<td>(209)953-2754</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(209)658-4741</td>
</tr>
<tr>
<td>Johnson</td>
<td>Jerome</td>
<td>(319)319-1677</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(319)846-2198</td>
</tr>
<tr>
<td>Kent</td>
<td>Palinor</td>
<td>(319)782-3199</td>
</tr>
<tr>
<td>Montague</td>
<td>Romeo</td>
<td>(300)699-2154</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(360)973-6201</td>
</tr>
<tr>
<td>Rashid</td>
<td>Imani</td>
<td>(508)852-2146</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(508)366-9821</td>
</tr>
</tbody>
</table>
Part 4

SAS Studio and ODS

Chapter 4
SAS Studio and ODS ........................................ 93
Chapter 4

SAS Studio and ODS

About SAS Studio

SAS Studio is a web interface to the SAS system. Using SAS Studio, you can access your SAS resources such as data, libraries, and existing programs, and you can write new programs. You can use SAS Studio to access SAS installed on your local machine, on a remote server in your local environment, or on a hosted server in a cloud environment. The following editions of SAS Studio are currently available:

- SAS Studio Single-User edition, which is delivered with Base SAS
- SAS Studio Basic edition, which is delivered with Base SAS and SAS University Edition
- SAS Studio Mid-Tier edition, which is delivered with SAS Integration Technologies

To see which edition of SAS Studio you are using, select About SAS Studio in the SAS Studio Help menu.

Note: SAS/GRAPH is not included in the SAS University Edition.

For more information about SAS Studio, see SAS Studio: User's Guide.
The SAS Studio Output Environment

SAS Studio provides a default output environment for your SAS programs. Using the default output environment, you can do the following:

- Generate HTML5, PDF, and RTF output by default, and view the HTML5 output in the RESULTS tab.
- Download your generated output using the download buttons. There is a download button for each of the three default output types.
- Change the default output style for each destination using the Preferences window.

Default SAS Studio Output

Viewing Default Results

In SAS Studio, by default, output is generated in the HTML5, PDF, and RTF formats. If you want to change the default output, in the Preferences window in SAS Studio, specify only the output that you want. You can also change the default style for your output to any of the ODS styles that are delivered. See “Changing the Default Styles” on page 95. For more information, see SAS Studio: User's Guide

The default HTML5 results are the only results displayed in the RESULTS tab in SAS Studio.
The PDF and RTF output is generated but is not displayed. See “Downloading Default HTML5, PDF, and RTF Results from SAS Studio ” on page 95.

**Downloading Default HTML5, PDF, and RTF Results from SAS Studio**

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. The **RESULTS** tab toolbar provides a separate download button (†) for HTML5 (†), PDF (†), and RTF (†). To download your results, click the icon for the desired format, and then specify a location for the output file.

**Changing the Default Styles**

The Preferences window enables you to change the styles for the HTML5, PDF, and RTF output.

To access the options to change the styles, do the following:

1. Click the More application options button and select **Preferences**.
2. In the Preferences window, click **Results** as shown in the following figure.

![Preferences Window](image)

**Note:** If you want to use a custom style, you must customize the SAS Studio output environment. See “Customized Output Environment ” on page 96.

**Display a Data Table**

To display a data table in SAS Studio, double-click the table name in the Libraries section of the navigation pane. In the following example, the CLASS table in the SASHELP library was selected.
You must customize the SAS Studio output environment when you want to do any of the following:

- generate output for other output destinations
- send results to another location
- use a custom style for your output
- use an image format other than the default
- create a drill-down graph
- create an animated GIF or SVG image

To customize the SAS Studio output environment, you should first disable the default output environment in order to conserve system resources. Next, establish your own output environment, and then execute the SAS statements that are required to generate your output. Use ODS statements, ODS Procedures, or ODS options in your SAS program to define the environment that you need.

As a best practice, if your SAS program requires a customized output environment in SAS Studio, your program should always perform these steps:
1. Create a file reference for your ODS output. You can use the &_SASWS_ macro variable that is defined in SAS Studio to reference your home directory as shown in the following statement:

   filename odsout "&_SASWS_/charts";

   If you want to store your image files in a separate directory, create a second file reference for your image files as shown in the following statement:

   filename odsiout "&_SASWS_/charts/images";

   **Note:** The directories that you specify must already exist, and you must have Write access to the directories.

2. To conserve system resources, disable the default output environment using the following statement:

   ods _all_ close;

3. Open the desired ODS destination. Use the PATH= option to specify the file reference that you created for your ODS output. If you created a separate file reference for your image files, use the GPATH= option to specify the image output file reference. Here is an example.

   ods html path=odsout gpath=odsiout file="saleschart.html";

4. Execute the SAS statements that are required to generate your output.

5. Close your ODS destination.

When you disable the default SAS Studio output environment, results are no longer displayed in the RESULTS tab for the duration of your program. The results are generated only by the ODS destination that you open.

**Generate Output for Other Output Destinations**

If you need to generate output other than the default HTML5, PDF, or RTF output, you must open your own ODS destination. Examples of output destinations include HTML, PowerPoint, and LISTING. After you disable the default output environment, use an ODS statement to open your own output destination. Here is an example:

   filename odsout "&_SASWS_/charts";
   ods _all_ close;
   ods powerpoint path=odsout file="filename";


**Send Your Results to Another Location**

When you execute a program in SAS Studio, you can download the output from the RESULTS tab to your local machine. If you want to send your output directly to another location, you must open your own ODS destination. By default, output files that are generated by the ODS destinations that you open are written to your home directory.

In SAS Studio Single-User deployments, the output is written to your home directory on your local machine.

In SAS Studio Basic and in SAS Studio Mid-Tier deployments, the output is written to your home directory on the remote SAS server. Your home directory appears in the navigation pane under Files and Folders or Server Files and Folders.
If you want to send the results to a specific location, use a FILENAME statement to define a file reference to the desired location. You can use the &_SASWS_ macro variable to reference your home directory. After you create the file reference, use the PATH=FILEREFERENCE option in your ODS statement. Here is an example.

```sas
filename odsout "&_SASWS_/charts";
ods _all_ close;
ods html path=odsout file="sales.htm";
```

In this case, file sales.htm and any image files that are generated are written to subdirectory charts in your home directory.

**Use a Custom Style for Your Output**

When you need to use a custom ODS style such as a corporate style for your results in SAS Studio, you must open your own ODS destination. You cannot specify a custom style for the default results. Use the STYLE= option in your ODS statement to specify your custom style. Here is an example.

```sas
filename odsout "&_SASWS_/charts";
ods _all_ close;
ods html path=odsout file="filename.htm" style="style-name";
```

To create a custom style, use the ODS Template procedure or CSSStyles. See Chapter 9, “Overview,” on page 913 for information about ODS styles.

**Use an Image Format Other Than the Default**

When you need to use an image format other than the default, you must specify the desired output format, and then open your own ODS destination. To specify the image format, do one of the following:

- If you are using SAS/GRAPH to create your graphs, specify the DEVICE= option in an OPTIONS or GOPTIONS statement. See SAS/GRAPH: Reference.
- If you are using ODS Graphics to create your graphs, specify the OUTPUTFMT= option in an ODS GRAPHICS statement. See “ODS GRAPHICS Statement”.

**Create a Drill-down Graph**

When you need to create a drill-down graph in SAS Studio, you must open your own ODS destination. Drill-down graphs provide a convenient means for users to explore complex data. In a drill-down graph, certain elements of the graph contain active links. When a user clicks a linked element, the linked resource appears in a new browser window by default.

For more information, see the following documents:

- If you are using SAS/GRAPH to create the graph, see SAS/GRAPH: Reference.
- If you are using the Graph Template Language to create the graph, see SAS Graph Template Language: User's Guide.

**Create an Animated GIF or SVG Image**

When you need to create an animated graph in SAS Studio, you must open your own ODS destination. An animated graph displays a series of charts automatically when the
graph is viewed in a web browser or other viewer that supports animation. The animation plays as a sequence of graphs in a slide-show fashion with a delay between each graph. The sequence can play only one time, loop a fixed number of times and then stop, or loop indefinitely.

For more information, see the following documents:

- If you are using SAS/GRAPH to create the graph, see SAS/GRAPH: Reference.
- If you are using the Graph Template Language to create the graph, see SAS Graph Template Language: User's Guide.
Part 5

ODS Statements

Chapter 5
  Introduction to ODS Language Statements 103

Chapter 6
  Dictionary of ODS Language Statements 105
Definition of ODS Statements

ODS statements provide greater flexibility in generating, storing, and reproducing SAS procedure and DATA step output. You can use the ODS statements to control different features of the Output Delivery System. ODS statements can be used anywhere in your SAS program. Some ODS statements remain in effect until you explicitly change them. Others are automatically cleared at particular times. (See the documentation for individual statements.)

Types of ODS Statements

**DATA Step Statements**

DATA step statements are either executable or declarative statements that appear in the DATA step. The ODS statements that are used in the DATA step are executable statements. Executable statements result in some action during individual iterations of the DATA step. For more information, see “Executable and Declarative Statements” in *SAS Statements: Reference*.

**Global Statements**

Global statements perform the following actions:

- provide information to SAS
- request information or data
- move between different modes of execution
• set values for system options

The global ODS statements deliver or store output in a variety of formats. You can use global statements anywhere in a SAS program. Global statements are not executable; they take effect as soon as SAS compiles program statements.

Global ODS statements are organized into three categories:

ODS: Output Control

are statements that provide descriptive information about the specified output objects and indicate whether the style template or table template is supplied by SAS. The Output Control statements can do the following:

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

ODS: SAS Formatted

are statements that enable you to produce output that is specific to SAS, such as a SAS data set, SAS output listing, or an ODS document. The statements in the ODS SAS Formatted category create the SAS output. For more information, see “Understanding ODS Destinations” on page 33.

ODS: Third-Party Formatted

are statements that enable you to apply styles and markup languages, or produce output to physical printers using page description languages. For more information, see “The Third-Party Formatted Destinations” on page 35.

Procedure Statements

Chapter 6

Dictionary of ODS Language Statements

ODS Statement Category Descriptions .................................................. 106
ODS Statements by Category .............................................................. 106

Dictionary ................................................................. 110
ODS HTML Statement .......................................................... 110
FILE Statement for ODS ...................................................... 155
LIBNAME Statement, SASEDOC ............................................ 165
ODS_ALL_CLOSE Statement .................................................. 169
ODS_CHTML Statement .................................................... 170
ODS_CSHELL Statement ..................................................... 195
ODS_DECIMAL_ALIGN Statement ........................................... 207
ODS_DOCUMENT Statement .................................................... 208
ODS_EPUB Statement ......................................................... 211
ODS_EPUB2 Statement ....................................................... 213
ODS_EPUB3 Statement ....................................................... 233
ODS_ESCAPECHAR Statement .............................................. 269
ODS_EXCEL Statement ....................................................... 292
ODS_EXCLUDE Statement ................................................... 321
ODS_GRAPHICS Statement .................................................. 328
ODS_HTML3 Statement ....................................................... 352
ODS_HTML Statement ........................................................ 379
ODS_HTML5 Statement ....................................................... 425
ODS_LAYOUT_ABSOLUTE Statement ........................................ 454
ODS_REGION Statement, Absolute ........................................... 465
ODS_LAYOUT_GRIDDED Statement ........................................... 468
ODS_REGION Statement, Gridded ............................................ 479
ODS_LAYOUT_END Statement ................................................ 484
ODS_LISTING Statement ..................................................... 484
ODS_MARKUP Statement ........................................................ 488
ODS_NO_DECIMAL_ALIGN Statement ....................................... 533
ODS_OUTPUT Statement ........................................................ 534
ODS_PACKAGE Statement ..................................................... 554
ODS_PATH Statement .......................................................... 563
ODS_PCL Statement ............................................................ 565
ODS_PDF Statement ............................................................. 575
ODSPOWERPOINT Statement ................................................. 623
ODS_PREFERENCES Statement ................................................ 663
ODS_PHTML Statement ........................................................ 663
ODS_PRINTER Statement ........................................................ 690
ODS_PROCLABEL Statement ................................................... 716
ODS_PROCTITLE Statement .................................................... 716
ODS Statement Category Descriptions

The following table lists and describes the categories of ODS global statements:

Table 6.1 Global Statements by Category

<table>
<thead>
<tr>
<th>Statement Category</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODS: Output Control</td>
<td>Provide descriptive information about the specified output objects and their locations.</td>
</tr>
<tr>
<td>ODS: SAS Formatted</td>
<td>Produce LISTING output, a SAS output data set, or a hierarchy file.</td>
</tr>
<tr>
<td>ODS: Third-Party Formatted</td>
<td>Produce files that are formatted in the proper destination format.</td>
</tr>
</tbody>
</table>

ODS Statements by Category

<table>
<thead>
<tr>
<th>Category</th>
<th>Language Elements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Access</td>
<td>ODS PACKAGE Statement (p. 554)</td>
<td>Opens, adds to, publishes, or closes one SAS Output Delivery System (ODS) package object.</td>
</tr>
<tr>
<td>File-Handling</td>
<td>FILE Statement for ODS (p. 155)</td>
<td>Creates an ODS output object by binding the data component to the table template. As an option, the FILE Statement lists the variables to include in the ODS output, and it specifies options that control how the variables are formatted.</td>
</tr>
<tr>
<td></td>
<td>PUT Statement for ODS (p. 864)</td>
<td>Writes data values to a special buffer from which they can be written to the data component and then formatted by ODS.</td>
</tr>
<tr>
<td>ODS: Output Control</td>
<td>LIBNAME Statement, SASEDOC (p. 165)</td>
<td>Uses the SASEDOC engine to associate a SAS libref (library reference) with one or more ODS output objects that are stored in an ODS document.</td>
</tr>
<tr>
<td>Category</td>
<td>Language Elements</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ODS <em>ALL</em> CLOSE Statement (p. 169)</td>
<td>Closes all open ODS output destinations.</td>
<td></td>
</tr>
<tr>
<td>ODS DOCUMENT Statement (p. 208)</td>
<td>Opens, manages, or closes the DOCUMENT destination, which produces a hierarchy of output objects that enables you to produce multiple ODS output formats without rerunning a PROC or DATA step.</td>
<td></td>
</tr>
<tr>
<td>ODS ESCAPECHAR Statement (p. 269)</td>
<td>Defines a representative character to be used in output strings.</td>
<td></td>
</tr>
<tr>
<td>ODS EXCLUDE Statement (p. 321)</td>
<td>Specifies output objects to exclude from ODS destinations.</td>
<td></td>
</tr>
<tr>
<td>ODS GRAPHICS Statement (p. 328)</td>
<td>Enables or disables ODS Graphics processing and sets graphics environment options. This statement affects ODS template-based (ODS Graphics) graphics only. The ODS GRAPHICS statement does not affect device-based graphics (SAS/GRAPH).</td>
<td></td>
</tr>
<tr>
<td>ODS LAYOUT ABSOLUTE Statement (p. 454)</td>
<td>Enables you to specify an exact page location using x and y coordinates. Absolute layout is limited to one page and is supported only by the PRINTER destinations.</td>
<td></td>
</tr>
<tr>
<td>ODS REGION Statement, Absolute (p. 465)</td>
<td>Creates a region container for absolute layouts. A region can contain output (such as text, tables, images, graphs) or nested layout containers. This ODS REGION statement is sandwiched between the ODS LAYOUT ABSOLUTE statement and the ODS LAYOUT END statement. The ODS LAYOUT ABSOLUTE statement manages the ODS LAYOUT destination when producing one page of output for PRINTER destinations.</td>
<td></td>
</tr>
<tr>
<td>ODS LAYOUT GRIDDED Statement (p. 468)</td>
<td>Enables you to arrange output dynamically in a two-dimensional grid structure. For example, gridded layout enforces automatic alignment of grid cells, allows layout to continue onto the next page, dynamically computes the size of a grid cell, and makes it easier to maintain the integrity of the report. Gridded layout is supported for HTML, POWERPOINT, and PRINTER destinations (PDF, PS, and PCL).</td>
<td></td>
</tr>
<tr>
<td>ODS REGION Statement, Gridded (p. 479)</td>
<td>Creates a region container for gridded layouts. A region can contain output (such as text, tables, images, graphs) or nested layout containers. This ODS REGION statement is sandwiched between the ODS LAYOUT GRIDDED statement and the ODS LAYOUT END statement. The ODS LAYOUT GRIDDED statement manages the ODS LAYOUT destination.</td>
<td></td>
</tr>
<tr>
<td>ODS LAYOUT END Statement (p. 484)</td>
<td>Ends a gridded or absolute layout statement block.</td>
<td></td>
</tr>
<tr>
<td>ODS PATH Statement (p. 563)</td>
<td>Specifies locations to write to or read from when creating or using PROC TEMPLATE definitions and the order in which to search for them.</td>
<td></td>
</tr>
<tr>
<td>ODS PREFERENCES Statement (p. 663)</td>
<td>Reverts the ODS settings back to start-up defaults.</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Language Elements</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>ODS PROCLABEL Statement (p. 716)</td>
<td>Enables you to change a procedure label.</td>
</tr>
<tr>
<td></td>
<td>ODS PROCTITLE Statement (p. 716)</td>
<td>Determines whether to write the title that identifies the procedure that produces the results in the output.</td>
</tr>
<tr>
<td></td>
<td>ODS RESULTS Statement (p. 731)</td>
<td>Tracks ODS output in the Results window.</td>
</tr>
<tr>
<td></td>
<td>ODS SELECT Statement (p. 758)</td>
<td>Specifies output objects for ODS destinations.</td>
</tr>
<tr>
<td></td>
<td>ODS SHOW Statement (p. 771)</td>
<td>Writes the specified selection or exclusion list to the SAS log.</td>
</tr>
<tr>
<td></td>
<td>ODS TEXT= Statement (p. 847)</td>
<td>Inserts text into your ODS output.</td>
</tr>
<tr>
<td></td>
<td>ODS TRACE Statement (p. 854)</td>
<td>Writes to the SAS log a record of each output object that is created, or suppresses the writing of this record.</td>
</tr>
<tr>
<td></td>
<td>ODS USEGOPT Statement (p. 860)</td>
<td>Determines whether ODS uses traditional SAS/GRAPH option settings.</td>
</tr>
<tr>
<td></td>
<td>ODS VERIFY Statement (p. 863)</td>
<td>Prints or suppresses a message indicating that a style template or a table template being used is not supplied by SAS.</td>
</tr>
<tr>
<td>ODS: SAS Formatted</td>
<td>ODS DECIMAL_ALIGN Statement (p. 207)</td>
<td>Controls the justification of numeric columns when no justification is specified.</td>
</tr>
<tr>
<td></td>
<td>ODS LISTING Statement (p. 484)</td>
<td>Opens, manages, or closes the LISTING destination.</td>
</tr>
<tr>
<td></td>
<td>ODS NO_DECIMAL_ALIGN Statement (p. 533)</td>
<td>Right-justifies numeric columns when no justification is specified.</td>
</tr>
<tr>
<td></td>
<td>ODS OUTPUT Statement (p. 534)</td>
<td>Produces a SAS data set from an output object and manages the selection and exclusion lists for the OUTPUT destination.</td>
</tr>
<tr>
<td>ODS: Third-Party Formatted</td>
<td>ODS HTML Statement (p. 110)</td>
<td>Opens, manages, or closes the HTML destination, which produces HTML 4.0 output that can contain embedded style sheets.</td>
</tr>
<tr>
<td></td>
<td>ODS CHTML Statement (p. 170)</td>
<td>Opens, manages, or closes the CHTML destination, which produces a compact, minimal HTML that does not use style information.</td>
</tr>
<tr>
<td></td>
<td>ODS CSVALL Statement (p. 195)</td>
<td>Opens, manages, or closes the CSVALL destination, which produces CSVALL output containing columns of data values that are separated by commas, and produces tabular output with titles, notes, and BY lines.</td>
</tr>
<tr>
<td></td>
<td>ODS EPUB Statement (p. 211)</td>
<td>Opens, manages, or closes the ODS EPUB destination, which generates EPUB e-books.</td>
</tr>
<tr>
<td>Category</td>
<td>Language Elements</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ODS EPUB2 Statement (p. 213)</td>
<td></td>
<td>Opens, manages, or closes the EPUB2 destination, which generates EPUB e-books using the EPUB2 standard.</td>
</tr>
<tr>
<td>ODS EPUB3 Statement (p. 233)</td>
<td></td>
<td>Opens, manages, or closes the EPUB3 destination, which generates EPUB e-books.</td>
</tr>
<tr>
<td>ODS EXCEL Statement (p. 292)</td>
<td></td>
<td>Opens, manages, or closes the ODS destination for Excel, which produces Excel spreadsheet files compatible with Microsoft Office 2010 and later versions.</td>
</tr>
<tr>
<td>ODS HTML3 Statement (p. 352)</td>
<td></td>
<td>Opens, manages, or closes the HTML3 destination, which produces HTML 3.2 formatted output.</td>
</tr>
<tr>
<td>ODS HTML Statement (p. 379)</td>
<td></td>
<td>Opens, manages, or closes the HTML destination, which produces HTML 4.0 output that can contain embedded style sheets.</td>
</tr>
<tr>
<td>ODS HTML5 Statement (p. 425)</td>
<td></td>
<td>Opens, manages, or closes the HTML5 destination, which produces HTML 5.0 output that contains embedded style sheets.</td>
</tr>
<tr>
<td>ODS MARKUP Statement (p. 488)</td>
<td></td>
<td>Opens, manages, or closes the MARKUP destination, which produces SAS output that is formatted using one of many different markup languages.</td>
</tr>
<tr>
<td>ODS PCL Statement (p. 565)</td>
<td></td>
<td>Opens, manages, or closes the PCL destination, which produces printable output for PCL (HP LaserJet) files.</td>
</tr>
<tr>
<td>ODS PDF Statement (p. 575)</td>
<td></td>
<td>Opens, manages, or closes the PDF destination, which produces PDF output, a form of output that is read by Adobe Acrobat and other applications.</td>
</tr>
<tr>
<td>ODS POWERPOINT Statement (p. 623)</td>
<td></td>
<td>Opens, manages, or closes the ODS destination for PowerPoint, which produces PowerPoint output.</td>
</tr>
<tr>
<td>ODS PHTML Statement (p. 663)</td>
<td></td>
<td>Opens, manages, or closes the PHTML destination, which produces simple HTML output that uses twelve style elements and no class attributes for the presentation. Class attributes are used only for the justification.</td>
</tr>
<tr>
<td>ODS PRINTER Statement (p. 690)</td>
<td></td>
<td>Opens, manages, or closes the PRINTER destination, which produces printable output.</td>
</tr>
<tr>
<td>ODS PS Statement (p. 717)</td>
<td></td>
<td>Opens, manages, or closes the PS destination, which produces PostScript (PS) output.</td>
</tr>
<tr>
<td>ODS RTF Statement (p. 732)</td>
<td></td>
<td>Opens, manages, or closes the RTF destination, which produces output written in Rich Text Format for use with Microsoft Word 2002.</td>
</tr>
<tr>
<td>ODS Tagset Statement (p. 772)</td>
<td></td>
<td>Opens, manages, or closes the specified tagset destination.</td>
</tr>
<tr>
<td>ODS TAGSETS.RTF Statement (p. 803)</td>
<td></td>
<td>Opens, manages, or closes the RTF destination, which produces measured output that is written in Rich Text Format.</td>
</tr>
</tbody>
</table>
Dictionary

ODS HTML Statement

Opens, manages, or closes the HTML destination, which produces HTML 4.0 output that can contain embedded style sheets.

Valid in: Anywhere
Category: ODS: Third-Party Formatted
Default: The default style for Markup family destinations is HTMLBlue.
Interaction: By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC PLOT or PROC CHART), ODS formats the output in SAS Monospace font. If you are creating output that will be viewed in an operating environment where SAS software is not installed, this output is not displayed correctly. This is because without SAS, the SAS Monospace font is not recognized. To make your document display correctly, include the following statement before your SAS program:

```sas
OPTIONS FORMCHAR="|----|+|-----|-----|-----|/\<>*";
```

z/OS specifics: If you use graphics that are created with either the ACTXIMG or JAVAIMG device drivers in the z/OS operating environment, then specify either the GPATH= option or the PATH= option in the ODS HTML statement.


Syntax

```sas
ODS HTML (<ID=> identifier> < action> ;
ODS HTML (<ID=> identifier> <option(s)> ;

Summary of Optional Arguments

(ID= identifier)
Open multiple instances of the same destination at the same time

ANCHOR= 'anchor-name'
Specify a unique base name for the anchor tag that identifies each output object in the current body file

ARCHIVE='string'
Specify which applet to use to view ODS HTML output

ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)
Specify attributes to write between the tags that generate dynamic graphics output
BASE= ’base-text’
Specify text to use as the first part of all links and references that ODS
creates in output files

BODY= ’file-specification’ (suboption(s))
Open a markup family destination and specify the file that contains the
primary output that is created by the ODS statement

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
Specify how to measure the width of cells. Use to override the default value
of BOX_SIZING for a destination

CHARSET= character-set
Specify the character set to be generated in the META declaration for the
HTML output

CLOSE
Close the destination and the file that is associated with it

CODE= ’file-specification’ <<(suboption(s))>>
Open the HTML destination and specify the file that contains relevant style
information

CODEBASE=’string’
Create a file path that can be used by the GOPTIONS devices

CONTENTS= ’file-specification’ <(suboption(s))>
Open the HTML destination and specify the file that contains a table of
contents for the output

CSSSTYLE= ’file-specification’<(<media-type-1<…media-type-10>)>
Specify a cascading style sheet to apply to your output

DEVICE= device-driver
Specify a device for the output destination

DOM=’external-file’
Specify that the ODS document object model is written to the SAS log or to
an external file.

ENCODING= local-character-set-encoding
Override the encoding for input or output processing (transcodes) of external
files

EVENT=event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | 
TARGET= | TEXT= | URL= )
Specify an event and the value for event variables that is associated with the
event

EXCLUDE exclusion(s) | ALL | NONE
Exclude output objects from the destination

FRAME= ’file-specification’ <(suboption(s))>
Specify the file that integrates the table of contents, the page contents, and the
body file

GFOOTNOTE | NOGFOOTNOTE
Control the location where footnotes are printed in the graphics output

GPATH= ’aggregate-file-storage-specification’ | fileref | libref.catalog (URL= 
’Uniform-Resource-Locator’ | NONE)
Specify the location for all graphics output that is generated while the
destination is open

GTITLE | NOGTITLE
Control the location where titles are printed in the graphics output

HEADTEXT= ’markup-document-head’
Specify HTML tags to place between the <HEAD> and </HEAD> tags in all of the output files.

**IMAGE DPI**
Specify the image resolution for graphical output

**METATEXT**='metatext-for-document-head'
Specify HTML code to use as the <META> tag between the <HEAD> and </HEAD> tags in all of the HTML output files.

**NEWFILE**= starting-point
Create a new body file at the specified starting point

**OPTIONS** (DOC= | <suboption(s)>)
Specify tagset-specific suboptions and a named value

**PACKAGE** <package-name>
Specify that the output from the destination be added to an ODS package

**PAGE**='file-specification' <(suboption(s))>
Open the HTML destination and specify the file that contains a description of each page of the body file and contains links to the body file

**PARAMETERS**=(parameter-pair-1 ... parameter-pair-n)
Write the specified parameters between the tags that generate dynamic graphics output

**PATH**='aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)
Specify the location of an aggregate storage location or a SAS catalog for all markup files

**RECORD_SEPARATOR**='alternative-separator' | NONE
Specify an alternative character or string to separate lines in the output files

**SELECT** selection(s) | ALL | NONE
Select output objects for the destination

**SGE**=ON | YES | OFF | NO
Generate a file that can be edited with the ODS Graphics Editor

**SHOW**
Write to the SAS log the current selection or exclusion list for the destination

**STYLE**= style-template
Specify a style template to use in writing output files

**STYLESHEET**='file-specification' <(suboption(s))>
Open the HTML destination and place style information for output into an external file, or read style sheet information from an existing file

**TEXT**=text-string
Insert text into your document

**TRANTAB**= 'translation-table'
Specify a translation table to use when transcoding a file for output

### Without Arguments
If you use the ODS HTML statement without an action or options, then it opens the HTML destination and creates HTML output.

### Actions
The following actions are available for the ODS HTML statement.

**CLOSE**
closes the destination and any files that are associated with it.
Tip  When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

**EXCLUDE exclusion(s) | ALL | NONE**
excludes one or more output objects from the destination.

- **Default**: NONE
- **Restriction**: A destination must be open for this action to take effect.
- **See**: “ODS EXCLUDE Statement” on page 321

**SELECT selection(s) | ALL | NONE**
selects output objects for the specified destination.

- **Default**: ALL
- **Restriction**: A destination must be open for this action to take effect.
- **See**: “ODS SELECT Statement” on page 758

**SHOW**
writes the current selection list or exclusion list for the destination to the SAS log.

- **Restriction**: The destination must be open for this action to take effect.
- **Tip**: If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.
- **See**: “ODS SHOW Statement” on page 771

---

**Optional Arguments**

**ANCHOR= 'anchor-name'**
specifies a unique base name for the anchor tag that identifies each output object in the current body file.

Each output object has an anchor tag for the contents, page, and frame files to reference. The links and references are automatically created by ODS. The links and references point to the name of an anchor. Therefore, each anchor name in a file must be unique.

- **anchor-name** is the base name for the anchor tag that identifies each output object in the current body file.

  ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR= 'TABULATE', then ODS names the first anchor `tabulate`. The second anchor is named `tabulate1`; the third is named `tabulate2`, and so on.

- **Restrictions**: Each anchor name in a file must be unique.

  Only alphanumeric values, the special characters "$ - _ . + ! * ' () , " and reserved characters used for their reserved purposes can be used unencoded within a URL.
### ARCHIVE='string'

specifies which applet to use to view the ODS HTML output. The ARCHIVE= option is valid only for the GOPTIONS Java device.

The string must be one that the browser can interpret. For example, if the archive file is local to the computer that you are running SAS on, you can use the FILE protocol to identify the file. If you want to point to an archive file that is on a web server, use the HTTP protocol.

**Default**

If you do not specify ARCHIVE= and you are using the JAVA device driver, ODS uses the value of the SAS system option APPLETOC=. There is no default if you are using the ACTIVEX device driver.

**Requirements**

You must enclose string in quotation marks.

**Interaction**

Use ARCHIVE= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA or DEVICE=ACTIVEX option in the GOPTIONS statement.

**Tips**

Typically, this option should not be used, because the SAS server automatically determines the correct SAS/GRAPH applets to view the ODS HTML output. However, if you have renamed the JAR files, or have other applets with which to view the ODS HTML output, this option enables you to access these applets.

Use the CODEBASE= option to specify the file path. It is recommended that you do not put a file path in your ARCHIVE= option.
The value of APPLETOC= points to the location of the Java archive files that are shipped with the SAS system. To find out what the value of this option is, you can either look in the Options window in the Files folder under Environment Control, or you can submit the following procedure step:

```
proc options option=appletloc;
run;
```

**ATTRIBUTES=** *(attribute-pair-1 .. attribute-pair-n)*

writes the specified attributes between the tags that generate dynamic graphics output.

- **attribute-pair** specifies the name and value of each attribute. *attribute-pair* has the following form:
  
  `'attribute-name'='attribute-value'`

- **attribute-name** is the name of the attribute.

- **attribute-value** is the value of the attribute.

**Requirement** You must enclose *attribute-name* and *attribute-value* in quotation marks.

**Interaction** Use the ATTRIBUTES= option in conjunction with SAS/GRAPH procedures and with the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

**See** SAS/GRAPH: Reference for valid attributes for the Graph Applet, the Map Applet, the Contour Applet, and the MetaView Applet.

**BASE=** *(base-text)*

specifies the text to use as the first part of all links and references that ODS creates in the output files.

- **base-text** is the text that ODS uses as the first part of all links and references that ODS creates in the file.

Consider this specification:

```
BASE= 'http://www.your-company.com/local-url/'
```

In this case, ODS creates links that begin with the string http://www.your-company.com/local-url/. The appropriate anchor-name completes the link.

**Requirement** You must enclose *base-text* in quotation marks.

**BODY=** *(file-specification) (suboption(s))*

opens a markup family destination and specifies the file that contains the primary output that is created by the ODS statement. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.
file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

Restriction The BODY=fileref option cannot be used in conjunction with the NEWFILE= option.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

entry.markup
specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

(suboption(s))
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 136.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 136.

(NO_TOP_MATTER)
See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 137.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 137.

(URL='Uniform-Resource-Locator')
See For complete documentation about the URL= suboption, see “(URL='Uniform-Resource-Locator’)” on page 138.
## Alias

FILE=

## Interaction

Using the BODY= option in an ODS markup family statement that refers to an open ODS markup destination forces ODS to close the destination and all associated files. ODS then opens a new instance of the destination. For more information see “Opening and Closing the MARKUP Destination” on page 517.

### Note

For some values of TAGSET=, this output is an HTML file. For other TAGSET= values, the output is an XML file, and so on.

**BOX_SIZING=(CONTENT_BOX | BORDER_BOX)**

specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry. For information about using the SAS Registry, see “Changing SAS Registry Settings for ODS” on page 40.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at [http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing](http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing).

**CHARSET= character-set**

specifies the character set to be generated in the META declaration for the HTML output.

See For information about the CHARSET= option, see “CHARSET= Option” in [SAS National Language Support (NLS): Reference Guide](index.html).

**CODE= ’file-specification’ <(suboption(s))>**

opens a markup family destination and specifies the file that contains accompanying programming code, such as JavaScript or XSL (Extensible Stylesheet Language). These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification

specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

- **external-file**
  
is the name of an external output file.

  **Requirement** You must enclose external-file in quotation marks.

- **fileref**
  
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

  See See “FILENAME Statement” in [SAS Statements: Reference](index.html).

- **entry.markup**
  
specifies an entry in a SAS catalog to write to.

  Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.
suboption(s)

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)

enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 136.

(URL= 'Uniform-Resource-Locator' )

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator' )” on page 138.

CODEBASE=’string’

specifies the location of the executable Java applet or the ActiveX control file. string is specified as a pathname or as a URL. The CODEBASE file path option has two definitions, depending on the GOPTIONS device used.

When you generate web presentations with the JAVA and ActiveX device drivers, SAS generates HTML pages that automatically look for the JAVA archive files or the ActiveX control file in the default installation location.

For the ActiveX device:

If you use the ActiveX device driver with ODS to generate output containing an ActiveX control, then specify the CODEBASE= option in the ODS statement. The value of the CODEBASE= option should include the location and the version of the EXE file.

Tip You do not need to specify the CODEBASE= option with the DEVICE=ACTIVEX option unless the users that view your output do not have the ActiveX control installed on their machine. When users that do not have the ActiveX control installed view your output, they are prompted to download the control.

See SAS/GRAPH: Reference for information about specifying the location of control and applet files using the CODEBASE= and ARCHIVE= options.

For the Java device:

If you use the Java device driver with ODS to generate output containing a SAS/GRAPH applet, specify the path to the JAR file with the CODEBASE= option in the ODS statement.

When you specify DEVICE=JAVA, the users that view your output must have access to the appropriate Java applet. By default, SAS sets the value of CODEBASE= to refer to the executable file for the applet that is automatically installed with SAS. The default location of the SAS Java archive files is specified by the APPLETLOC= system option. You do not need to specify the CODEBASE= option if both of the following conditions are true.

• The default location is accessible by users who are viewing your web presentation.
• The SAS Java archive is installed at that location.
Tip Specify only the directory of the JAR file. The CODEBASE= location can be specified as a pathname or as a URL.

See SAS/GRAPH: Reference for information about specifying the location of control and applet files using the CODEBASE= and ARCHIVE= options.

CONTENTS= 'file-specification' <(suboption(s))>
opens a markup family destination and specifies the file that contains a table of contents for the output. These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

entry.markup
specifies an entry in a SAS catalog to write to.

interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC) enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 136.

(NO_BOTTOM_MATTER) specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 136.
(NO_TOP_MATTER)
  specifies that no beginning markup language source code be added to the top
  of the output file. For HTML 4.0, the NO_TOP_MATTER option removes
  the style sheet.

  See  For complete documentation about the NO_TOP_MATTER suboption,
       see “(NO_TOP_MATTER)” on page 137.

(TITLE="title-text")
  inserts into the metadata of a file the text string that you specify as the text to
  appear in the browser window title bar.

    title-text
    is the text in the metadata of a file that indicates the title.

  See  For complete documentation about the TITLE= suboption, see
       “(TITLE="title-text")” on page 137.

(URL= 'Uniform-Resource-Locator')
  specifies a URL for the file-specification. ODS uses this URL (instead of the
  filename) in all the links and references that it creates and that point to the
  file.

  See  For complete documentation about the URL= suboption, see “(URL=

CSSSTYLE= 'file-specification'<(media-type-1<.. media-type-10>)>
  specifies a cascading style sheet to apply to your output.

    file-specification
    specifies a file, fileref, or URL that contains CSS code.

    file-specification is one of the following:

      "external-file"
      is the name of the external file.

      Requirement  You must enclose external-file in quotation marks.

      fileref
      is a file reference that has been assigned to an external file. Use the
      FILENAME statement to assign a fileref.

      See  For information about the FILENAME statement, see “FILENAME
            Statement” in SAS Statements: Reference.

      "URL"
      is a URL to an external file.

      Requirement  You must enclose URL in quotation marks.

(media-type-1<.. media-type-10>)
  specifies one or more media blocks that correspond to the type of media that your
  output is rendered on. CSS uses media type blocks to specify how a document is
  to be presented on different media: on the screen, on paper, with a speech
  synthesizer, with a braille device, and so on.

  The media block is added to your output in addition to the CSS code that is not
  contained in any media blocks. By using the media-type suboption, in addition to
the general CSS code, you can import the section of a CSS file intended only for a specific media type.

**Default**
If no `media-type` is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

**Range**
You can specify up to ten different media types.

**Requirements**
You must enclose `media-type` in parentheses.

**Tip**
You must specify `media-type` next to the `file-specification` specified by the CSSSTYLE= option.

If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

**Restriction**
The CSSSTYLE= option does not affect SAS/GRAPH output.

**Requirement**
CSS files must be written in the same type of CSS produced by the ODS HTML statement. To view the CSS code that ODS creates, you can do one of the following:
- specify the ODS TRACE DOM statement
- specify the DOM option

**Interaction**
If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

**See**
For complete documentation about ODS and Cascading Style Sheets, see *SAS Output Delivery System: Advanced Topics*

**Example**
“Example 6: Applying a CSS File to ODS Output” on page 527

**DEVICE= device-driver**
specifies the name of a device driver. ODS automatically selects an optimal default device for each open output destination.

The following table lists the default devices for the most common ODS output destinations. These default devices are used when graphics are created using SAS/GRAPH or ODS Graphics. In the third maintenance release of SAS 9.4, EPUB3 is the default EPUB destination. EPUB2 was the default EPUB version in prior releases of SAS 9.4. This default is set in the Registry. For a complete list of supported devices and file types, see “Supported File Types for Output Destinations” on page 349.

**Table 6.2 Default Devices for ODS Output Destinations**

<table>
<thead>
<tr>
<th>Output Destination</th>
<th>Default Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPUB2</td>
<td>PNG</td>
</tr>
<tr>
<td>EPUB3 (EPUB)</td>
<td>SVG</td>
</tr>
<tr>
<td>Output Destination</td>
<td>Default Device</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>HTML</td>
<td>PNG</td>
</tr>
<tr>
<td>HTML5</td>
<td>SVG</td>
</tr>
<tr>
<td>LISTING</td>
<td>PNG</td>
</tr>
<tr>
<td>Measured RTF</td>
<td>EMF</td>
</tr>
<tr>
<td>RTF</td>
<td>EMF</td>
</tr>
<tr>
<td>Markup Tagsets</td>
<td>PNG</td>
</tr>
</tbody>
</table>

Tips  Specifying a device on the ODS DEVICE= option takes precedence over the SAS global option and the graphics option.

For information about using the SAS Registry to change defaults, see “Changing SAS Registry Settings for ODS” on page 40.


DOM="external-file”>
specifies that the ODS document object model is written to the SAS log or an external file.

*external-file*
is the name of an external output file.

Requirement You must enclose *external-file* in quotation marks.

See  For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in SAS Output Delivery System: Advanced Topics.

ENCODING= local-character-set-encoding
overrides the encoding for input or output processing (transcodes) of external files.

See  For information about the ENCODING= option, see SAS National Language Support (NLS): Reference Guide.

EVENT=event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )
specifies an event and the value for event variables that are associated with the event.

(FILE= BODY | CODE | CONTENTS | DATA | FRAME | PAGES | STYLESHEET);
triggers one of the known types of output files that correspond to the BODY=, CODE=, CONTENTS=, FRAME=, PAGES=, and STYLESHEET= options.

(FINISH)
triggers the finish section of an event.
See For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(LABEL=’variable-value’) specifies the value for the LABEL event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the LABEL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(NAME=’variable-value’) specifies the value for the NAME event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the NAME event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(START) triggers the start section of an event.

See For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(STYLE=style-element) specifies a style element.

See For information about style elements, see “Style Attributes Overview” in SAS Output Delivery System: Procedures Guide.

(TARGET=’variable-value’) specifies the value for the TARGET event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the TARGET event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(TEXT=’variable-value’) specifies the value for the TEXT event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the TEXT event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(URL=’variable-value’) specifies the value for the URL event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the URL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

Default (FILE=’BODY’)
Requirement  The EVENT= option's suboptions must be enclosed in parentheses.

**FRAME= 'file-specification' <(suboption(s))>

opens a markup family destination and, for HTML output, specifies the file that integrates the table of contents, the page contents, and the body file. If you open the frame file, then you see a table of contents, a table of pages, or both, as well as the body file. For XML output, FRAME= specifies the file that contains the DTD. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

**file-specification**

specifies the file, fileref, or SAS catalog to write to.

**file-specification** is one of the following:

**external-file**

is the name of an external output file.

**Requirement**  You must enclose external-file in quotation marks.

**fileref**

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

**See**  For information about the FILENAME statement, see “FILENAME Statement” in *SAS Statements: Reference*.

**entry.markup**

specifies an entry in a SAS catalog to write to.

**Interaction**  If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

**suboption(s)**

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

**(DYNAMIC)**

enables you to send output directly to a web server instead of writing it to a file.

**See**  For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 136.

**(NO_BOTTOM_MATTER)**

specifies that no ending markup language source code be added to the output file.

**See**  For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 136.
(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 137.

(TITLE="title-text")
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

  title-text
  is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE="title-text")” on page 137.

(URL= 'Uniform-Resource-Locator' )
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator' )” on page 138.

Restriction If you specify the FRAME= option, then you must also specify the CONTENTS= option, the PAGE= option, or both.

Example “Example 2: Creating an XML File and a DTD” on page 520

GFOOTNOTE | NOGFOOTNOTE
controls the location where footnotes are printed in the graphics output.

GFOOTNOTE
writes footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

NOGFOOTNOTE
writes footnotes that are created by ODS, which appear outside the graph borders.

Default GFOOTNOTE

Restrictions Footnotes that are displayed by a markup language statement support all SAS/GRAPH FOOTNOTE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH FOOTNOTE statement, see “FOOTNOTE Statement” in SAS/GRAPH: Reference.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.
GPATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)

specifies the location for all graphics output that is generated while the destination is open. Use this option when you want to write graphics output files to a location different that specified by the PATH= option for markup files. If you specify an invalid filename, the ActiveX and Java devices send output to the default filename. Other devices create the file as a directory and write output to that directory using the default filename. For information about how ODS names catalog entries and external files, see SAS/GRAPH: Reference.

'aggregate-file-storage-location'

specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement You must enclose aggregate-file-storage-location in quotation marks.

fileref

is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction If you specify a fileref in the GPATH= option, then ODS does not use information from the GPATH= option when it constructs links.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

libref.catalog

specifies a SAS catalog to write to.

URL= 'Uniform-Resource-Locator' | NONE

specifies a URL for file-specification.

Uniform-Resource-Locator

is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

Requirement You must enclose Uniform-Resource-Locator in quotation marks.

NONE

specifies that no information from the GPATH= option appears in the links or references.

Tip This option is useful for building output files that can be moved from one location to another. If the links from the contents and page files are constructed with a simple URL (one name), then they resolve, as long as the contents, page, and body files are all in the same location.

Default If you omit the GPATH= option, then ODS stores graphics in the location that is specified by the PATH= option. If you do not specify the PATH= option, then ODS stores the graphics in the current directory. For more information, see the PATH= option.

GTITLE | NOGTITLE

controls the location where titles are printed in the graphics output.
GTITLE
writes the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.

NOGTITLE
writes the title that is created by ODS, which appears outside of the graph borders.

Default
GTITLE

Restrictions
Titles that are displayed by any markup language statement support most SAS/GRAPH TITLE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH TITLE statement, see TITLE statement.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

HEADTEXT= 'markup-document-head'
specifies markup tags to place between the <HEAD> and </HEAD> tags in all of the output files.

mark-up-document-head
specifies the markup tags to place between the <HEAD> and </HEAD> tags.

Restriction
HEADTEXT= cannot exceed 256 characters.

Requirement
You must enclose markup-document-head in quotation marks.

Tips
ODS cannot parse the markup that you supply. It should be well-formed markup that is correct in the context of the <HEAD> and </HEAD> tags.

Use the HEADTEXT= option to define programs (such as JavaScript) that you can use later in the file.

(ID= identifier)
enables you to run multiple instances of the same destination at the same time. Each instance can have different options.

identifier
specifies another instance of the destination that is already open. identifier is numeric or a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numeric characters.

Restriction
If identifier is numeric, it must be a positive integer.

Requirement
You must specify the ID= option immediately after the destination name.

Tip
You can omit the ID= option and instead use a name or a number to identify the instance.

Example
“Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594
**IMAGE_DPI=**
specifies the image resolution for graphical output.

<table>
<thead>
<tr>
<th>Alias</th>
<th>DPI=</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>96</td>
</tr>
<tr>
<td>Restriction</td>
<td>This option only applies when the ODS GRAPHICS statement IMAGEMAP = OFF</td>
</tr>
<tr>
<td>CAUTION</td>
<td>Caution: When using high DPI= or DPI_IMAGE values (values over 1000), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500M or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.</td>
</tr>
</tbody>
</table>

**METATEXT= 'metatext-for-document-head'**
specifies HTML code to use as the <META> tag between the <HEAD> and </HEAD> tags of all of the HTML output files.

'metatext-for-document-head'
specifies the HTML code that provides the browser with information about the document that it is loading. For example, this attribute could specify the content type and the character set to use.

| Requirement | You must enclose metatext-for-document-head in quotation marks. |
| Default     | If you do not specify METATEXT=, then ODS writes a simple <META> tag, which includes the content-type of the document and the character set to use, to all the HTML files that it creates. |
| Restriction | METATEXT= cannot exceed 256 characters. |
| Tip         | ODS cannot parse the HTML code that you supply. It should be well-formed HTML code that is correct in the context of the <HEAD> tags. If you are using METATEXT= as it is intended, then your META tag should look like this: <META your-metatext-is-here> |

**NEWFILE= starting-point**
creates a new body file at the specified starting-point.

*starting-point*
is the location in the output where you want to create a new body file.

ODS automatically names new files by incrementing the name of the body file. In the following example, ODS names the first body file REPORT.XML. Additional body files are named REPORT1.XML, REPORT2.XML, and so on.

Example:

```
BODY= 'REPORT.XML'
```

*starting-point* is one of the following:

**BYGROUP**
starts a new file for the results of each BY group.
NONE
writes all output to the body file that is currently open.

OUTPUT
starts a new body file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

Alias   TABLE

PAGE
starts a new body file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.

PROC
starts a new body file each time you start a new procedure.

Default     NONE
Restriction The NEWFILE= option cannot be used in conjunction with the BODY= fileref option.
Tips If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first body file MAY5.XML. Additional body files are named MAY6.XML, MAY7.XML, and so on.

Example:
BODY= 'MAY5.XML'

OPTIONS ( DOC= | <suboption(s)> )
specifies tagset-specific suboptions and a named value.

(DOC= 'HELP' | 'QUICK' | 'SETTINGS' | "CHANGELOG")
provides information about the specified tagset.

HELP
provides generic help and information with a quick reference.

QUICK
describes the options available for this tagset.

SETTINGS
provides the current option settings.

CHANGELOG
lists a history of changes made to the tagset. This suboption is supported only on the RTF tagset.

Requirement All values must be enclosed in quotation marks.

suboption(s)
specifies one or more suboptions that are valid for the specified tagset. Suboptions have the following format:

keyword=value
Specify one of the following options when opening an ODS tagset statement, or at any time after the destination has been opened, to get information about suboptions for the tagset.
• options(doc='help');
• options(doc='quick');
• options(doc='settings');

Requirement  suboption(s) must be enclosed in parentheses.

Example   “Example: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information ” on page 802

PACKAGE <package-name>
specifies that the output from the destination be added to a package.

package-name
specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

See “ODS PACKAGE Statement ” on page 554

Example  “Example 1: Creating an ODS Package” on page 558

PAGE= ’file-specification’ <(suboption(s))>
opens a markup family destination and specifies the file that contains a description of each page of the body file, and contains links to the body file. ODS produces a new page of output whenever a procedure requests a new page. These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement  You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

entry.markup
specifies an entry in a SAS catalog to write to.

Interaction  If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:
(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 136.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 136.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 137.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

title-text
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 137.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator’)” on page 138.

Interaction The SAS system option PAGESIZE= has no effect on pages in HTML output except when you are creating batch output. For information about the PAGESIZE= option, see “PAGESIZE= System Option” in SAS System Options: Reference.

PARAMETERS= (parameter-pair-1 ... parameter-pair-n)
writes the specified parameters between the tags that generate dynamic graphics output.

parameter-pair
specifies the name and value of each parameter. parameter-pair has the following form:

'parameter-name'='parameter-value'

parameter-name
is the name of the parameter.
parameter-value

is the value of the parameter.

Requirement

You must enclose parameter-name and parameter-value in quotation marks.

Interaction

Use PARAMETERS= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

See

SAS/GRAPH: Reference for valid parameters for the Graph Applet, Map Applet, Contour Applet, and the MetaView Applet.

PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)

specifies the location of an aggregate storage location or a SAS catalog for all markup files. If the GPATH= option is not specified, all graphics output files are written to the "aggregate-file-storage-specification" or libref.

'aggregate-file-storage-location'

specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement

You must enclose aggregate-file-storage-location in quotation marks.

fileref

is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction

If you use a fileref in the PATH= option, then ODS does not use information from PATH= when it constructs links.

See

For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

libref.catalog

specifies a SAS catalog to write to.

See

For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Statements: Reference.

URL= 'Uniform-Resource-Locator' | NONE

specifies a URL for the file-specification.

Uniform-Resource-Locator

is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

NONE

specifies that no information from the PATH= option appears in the links or references.

Tip

This option is useful for building output files that can be moved from one location to another. The links from the contents and page files must be constructed with a single-name URL, and the contents, page, and body files must be in the same location.
Interaction

If you use the BODY= or FILE= external file option in conjunction with the PATH= option, the external file specification should not include path information.

RECORD_SEPARATOR= 'alternative-separator' | NONE

specifies an alternative character or string that separates lines in the output files.

Different operating environments use different separator characters. If you do not specify a record separator, then the files are formatted for the environment where you run the SAS program. However, if you are generating files for viewing in a different operating environment that uses a different separator character, then you can specify a record separator that is appropriate for the target environment.

alternative-separator

represents one or more characters in hexadecimal or ASCII format. For example, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= '0D0A'x

Operating Environment Information

In a mainframe environment, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= '0D25'x

Requirement

You must enclose alternative-separator in quotation marks.

NONE

produces the markup language that is appropriate for the environment where you run the SAS job.

Windows Specifics

In a mainframe environment, by default, ODS produces a binary file that contains embedded record separator characters. This binary file is not restricted by the line-length restrictions on ASCII files. However, if you view the binary files in a text editor, then the lines run together. If you want to format the files so that you can read them with a text editor, then use RECORD_SEPARATOR= NONE. In this case, ODS writes one line of markup language at a time to the file. When you use a value of NONE, the logical record length of the file that you are writing to must be at least as long as the longest line that ODS produces. If the logical record length of the file is not long enough, then the markup language might wrap to another line at an inappropriate place.

Aliases

RECSEP=
RS=

SGE=ON | YES | OFF | NO

generates a file that can be edited only with the ODS Graphics Editor. The file created has an extension of .sge.

See

For details about using the ODS Graphics Editor to create SGE files, see SAS ODS Graphics Editor: User's Guide.

Example

ods html sge=on;
proc sgplot data=sashelp.class;
STYLE= style-template
specifies the style template to use in writing the output files.

style-template
describes how to display the presentation aspects (color, font face, font size, and
so on) of your SAS output. A style template determines the overall appearance of
the documents that use it. Each style template consists of style elements.

Interaction The STYLE= option is not valid when you are creating XML
output.

Note If you are using SAS Studio, you do not need to specify the
STYLE= option. You can go to Preferences ⇒ Results and change
the style from the drop-down list for your selected destination.

See For a complete discussion of style templates, see “TEMPLATE
Procedure: Creating a Style Template” in SAS Output Delivery
System: Procedures Guide.

Default If you do not specify a style template, then ODS uses the file that is
specified in the SAS registry subkey ODS ⇒ DESTINATIONS ⇒
MARKUP. By default, this value specifies Default.

Interaction If you specify the STYLE= option in an ODS HTML4 statement and
subsequently need PROC PRINT output to use new style templates in
another ODS HTML4 statement, close the first statement before
specifying the second statement.

STYLESHEET= 'file-specification' <(suboption(s))>
opens a markup family destination and places the style information for markup
output into an external file, or reads style sheet information from an existing file.
These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE
statement or ODS _ALL_ CLOSE statement.

• open the same destination with a second markup family statement. This closes
the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the
FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME
Statement” in SAS Statements: Reference.
entry.markup
specifies an entry in a SAS catalog to write to.

Interaction
If you specify an entry name, you must also specify a library
and catalog. See the discussion of the PATH= option.

suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for
writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a
file.

See For complete documentation about the DYNAMIC suboption, see
“(DYNAMIC)” on page 136.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output
file.

See For complete documentation about the NO_BOTTOM_MATTER
suboption, see “(NO_BOTTOM_MATTER)” on page 136.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top
of the output file. For HTML 4.0, the NO_TOP_MATTER option removes
the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption,
see “(NO_TOP_MATTER)” on page 137.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to
appear in the browser window title bar.

title-text
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see
“(TITLE='title-text')” on page 137.

(URL='Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the
filename) in all the links and references that it creates and that point to the
file.

See For complete documentation about the URL= suboption, see “(URL=

Note By default, if you do not specifically send the information to a separate
file, then the style sheet information is included in the specified HTML
file.

Example “Example 5: Including Multiple Cascading Style Sheets in One HTML
Document” on page 525
**TEXT=text-string**  
inserts text into your document by triggering the paragraph event and specifying a text string to be assigned to the VALUE event variable.

**Default**  
By default the TEXT= option is used in a paragraph event.

**Tip**  
You can specify a text-string for a specific event by using the TEXT= option with the EVENT= option by using the following syntax:  
\[ \text{EVENT=event-name (TEXT=text-string)} \]

**See**  
For information about events and event variables, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in *SAS Output Delivery System: Procedures Guide*.

**Example**  
“Example: Conditionally Excluding Output Objects and Sending Them to Different Output Destinations” on page 325

**TRANTAB='translation-table'**  
specifies the translation table to use when transcoding a file for output.

**See**  

### Suboptions

**DYNAMIC**

enables you to send output directly to a web server instead of writing it to a file. This option sets the value of the CONTENTTYPE= style attribute. For more information, see CONTENTTYPE= on page 1015 in PROC TEMPLATE.

**Default**  
If you do not specify DYNAMIC, then ODS sets the value of HTMLCONTENTTYPE= for writing to a file.

**Restriction**  
If you specify the DYNAMIC suboption with one of the following options in the ODS HTML statement, then you must specify it for all of these options in that statement.
- BODY=
- CONTENTS=
- PAGE=
- FRAME=
- STYLESHEET=
- TAGSET=

**Requirements**  
You must enclose DYNAMIC in parentheses.

**NO_BOTTOM_MATTER**

specifies that no ending markup language source code be added to the output file.

**Alias**  
NOBOT

**Requirements**  
You must enclose NO_BOTTOM_MATTER in parentheses.
You must specify NO_BOTTOM_MATTER next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

Interactions

The NO_BOTTOM_MATTER suboption, in conjunction with the NO_TOP_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

Tip

If you want to leave a body file in a state that you can append to with ODS, then use NO_BOTTOM_MATTER with the file-specification BODY= option in any markup language statement.

See

The NO_TOP_MATTER suboption

(NO_TOP_MATTER)

specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

Alias

NOTOP

Requirements

You must enclose NO_TOP_MATTER in parentheses.

You must specify NO_TOP_MATTER next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

Interactions

The NO_TOP_MATTER suboption, in conjunction with the NO_BOTTOM_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

See

The NO_BOTTOM_MATTER suboption and the ANCHOR=

(TITLE=’title-text’)

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

*title-text*

is the text in the metadata of a file that indicates the title.
### Requirements
You must enclose TITLE= in parentheses.

You must enclose title-text in quotation marks.

### Tip
If you are creating a web page that uses frames, then it is the TITLE= specification for the frame file that appears in the browser window title bar.

### Example
“Example 3: Creating Multiple Markup Output” on page 522

(URL=''Uniform-Resource-Locator''

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

### Requirements
You must enclose URL=''Uniform-Resource-Locator'' in parentheses.

You must enclose Uniform-Resource-Locator in quotation marks.

You must specify URL=''Uniform-Resource-Locator'' next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

### Tips
This option is useful for building HTML files that can be moved from one location to another. The links from the contents and page files must be constructed with a single name URL, and the contents, page, and body files must all be in the same location.

You never need to specify this suboption with the FRAME= option because ODS files do not reference the frame file.

### Example
“Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

### Details
The ODS HTML statement is part of the ODS markup family of statements. ODS statements in the markup family produce output that is formatted using one of many different markup languages, such as HTML (Hypertext Markup Language) and XML (Extensible Markup Language). You can specify a markup language that SAS supplies, or create one of your own and store it as a user-defined markup language.

Beginning with SAS 9.3, by default, in the Windowing environment with the Windows and UNIX operating systems, the LISTING destination is closed and the HTML destination is open. You do not have to submit an ODS HTML statement to generate HTML output, and you do not have to use the ODS HTML CLOSE statement to be able to view your output. However, to create LISTING output, you must either submit the ODS LISTING statement or enable the LISTING destination by other means. For more details, see “Working with Output Defaults” on page 21.

Starting in SAS 9.3, the default style for HTML is HTMLBlue.

The HTML destination supports Scalable Vector Graphics (SVG). For more information about creating SVG files in SAS/GRAPH, see “Using SVG Graphics” in SAS/GRAPH: Reference. For information about SVG and Universal Printing, see “Creating SVG (Scalable Vector Graphics) Files Using Universal Printing” in SAS Language Reference: Concepts. In order to view SVG graphics that were created using the ODS HTML
Examples

Example 1: Using the DOC Suboption to Get ODS HTML Information

Features:
- ODS HTML statement action:
  - CLOSE
- ODS HTML statement options:
  - OPTIONS (DOC="HELP")

Other features:
- PROC PRINT

Details
The following example prints a list of OPTIONS suboptions and a description of each suboption that is available for the HTML tagset. The information is printed to the SAS log.

Program

```plaintext
ods html options (doc="help");
proc print data=Sashelp.Class;
run;
```

Program Description

Print information about the OPTIONS suboptions to the SAS log file.

```plaintext
ods html options (doc="help");
```

Print the data set Sashelp.Class. The PROC PRINT statement prints the Sashelp.Class data set.

```plaintext
proc print data=Sashelp.Class;
run;
```

Output
Specify the DOC='HELP' suboption to print all of the OPTIONS suboptions and information about each of the suboptions to the SAS log.
Example 2: Using the Option Suboption PAGEBREAK=

Features:
- ODS HTML statement options:
  OPTIONS (PAGEBREAK="NO")

Other features:
- PROC PRINT

Details
The following example shows how to use the PAGEBREAK= suboption to control whether a page break is allowed or not. The default is to provide a page break after each print statement. In HTML, a page break is rendered by separating output with a horizontal rule. With PAGEBREAK="NO", the horizontal rule is not produced.
Program

```sas
ods html file="test.html" options(pagebreak='no');
options obs=2;
proc print data=Sashelp.Class;
runc;
proc print data=Sashelp.Class;
runc;
```

Program Description

**Specify the PAGEBREAK="NO" suboption.** The two data sets are rendered without a separating horizontal rule. The output is printed to the test.html file.

```sas
ods html file="test.html" options(pagebreak='no');
```

**Print only two observations of the data set.**

```sas
options obs=2;
```

**Print the data set Sashelp.Class.** The PROC PRINT statement prints the Sashelp.Class data set.

```sas
proc print data=Sashelp.Class;
runc;
```

**Print the data set Sashelp.Class.** Print the Sashelp.Class data set again. Because PAGEBREAK="NO" is specified, there is not a page break between the two data sets. By default, these two data sets would be written to two different pages.

```sas
proc print data=Sashelp.Class;
runc;
```

Output

Specify the PAGEBREAK="NO" suboption if you want fewer pages of output.

**Output 6.2  PAGEBREAK= Suboption Set to NO in HTML**

<table>
<thead>
<tr>
<th>The SAS System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obs</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>The SAS System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obs</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>
Example 3: Creating a Separate Body File for Each Page of Output

Features:
ODS HTML statement action:
   CLOSE

ODS HTML statement options:
   BASE=
   CONTENTS=
   BODY=
   FRAME=
   NEWFILE=
   PAGE=

Other features:
   #BYVAL parameter in titles
   NOBYLINE|BYLINE system option
   OPTIONS statement
   PROC FORMAT
   PROC SORT
   PROC REPORT
   PROC TABULATE
   TITLE statement

Data set:
   Grain_Production

Format:
   $CNTRY.

Details
The following example creates a separate HTML file for each page of procedure output, as well as a table of contents, a table of pages, and a frame file. The table of contents and table of pages appear and behave the same as those that would be created if all the output was in a single file. Because the output is in separate files, you cannot scroll from one page of output to the next. However, you can select individual HTML files to include in a report.

Program
proc sort data=grain_production;
   by year country type;
run;

ods html body='grain-body.htm'
   contents='grain-contents.htm'
   frame='grain-frame.htm'
   pages='grain-page.htm'
   newfile=page;
ods html;
options nobyline;
title 'Leading Grain-Producing Countries';
title2 'for #byval(year)';
proc report data=grain_production nowindows;
   by year;
   column country type kilotons;
define country / group width=14 format=$cntry.;
define type     / group 'Type of Grain';
define kilotons / format=comma12.;
footnote 'Measurements are in metric tons.';
run;

options byline;
title2;

proc tabulate data=grain_production format=comma12.;
class year country type;
var kilotons;
table year,
country*type,
kilotons*sum=' ' / box=_page_ misstext='No data';
format country $cntry.;
footnote 'Measurements are in metric tons.';
run;
ods html close;

Program Description

Sort the data set Grain_Production. PROC SORT sorts the data, first by values of the variable Year, then by values of the variable Country, and finally by values of the variable Type.

proc sort data=grain_production;
   by year country type;
run;

Create HTML output. The ODS HTML statement opens the HTML destination and creates HTML output. The FRAME=, CONTENTS=, and PAGE= options create a frame that includes a table of contents and a table of pages that link to the contents of the body file. The body file also appears in the frame. BASE= specifies a string to use as the first part of all links and references to the HTML files. Because no URL is specified for individual files, the final part of the link matches the filename. The string that the BASE= option specifies must be a valid path to your HTML files.

ods html body='grain-body.htm'
  contents='grain-contents.htm'
  frame='grain-frame.htm'
  page='grain-page.htm'

Specify that SAS create a new body file for each page of output. The NEWFILE=PAGE option opens and creates a new body file for each page of output.

newfile=page;

 Suppress the default BY line and specify a new value into the BY line. The NOBYLINE option suppresses the default BY line variable. The #BYVAL parameter specification inserts the current value of the BY variable Year into the title.

options nobyline;
title 'Leading Grain-Producing Countries';
title2 'for #byval(year)';
Produce a report. This PROC REPORT step produces a report on grain production. Each BY group produces a page of output, so ODS creates a new body file for each BY group. The NOWINDOWS option specifies that PROC REPORT runs without the REPORT window and sends its output to any open output destinations.

```sas
proc report data=grain_production nowindows;
  by year;
  column country type kilotons;
  define country / group width=14 format=$cntry.;
  define type     / group 'Type of Grain';
  define kilotons / format=comma12.;
  footnote 'Measurements are in metric tons.';
run;
```

Restore the default BY line and clear the second TITLE statement. The BYLINE option restores the default BY line. The TITLE2 statement clears the second TITLE statement.

```sas
options byline;
title2;
```

Produce a report. The TABLE statement in this PROC TABULATE step specifies the variable Year. Therefore, PROC TABULATE explicitly produces one page of output for 1995 and one for 1996. ODS starts a new body file for each page.

```sas
proc tabulate data=grain_production format=comma12.;
  class year country type;
  var kilotons;
  table year,
    country*type,
    kilotons*sum=' ' / box=_page_ misstext='No data';
  format country $cntry.;
  footnote 'Measurements are in metric tons.';
run;
```

Close the HTML destination. The ODS HTML CLOSE statement closes the HTML destination and all the files that are associated with it. If you do not close the destination, then you cannot view the HTML file specified by the FRAME attribute until you close your SAS session.

```sas
ods html close;
```

HTML Output

This frame file shows the first body file. Links in the table of contents and the table of pages point to the other body files. The frame file is not rendered in the results viewer after running this example. To open it, locate the file in your SAS output location.
Output 6.3  HTML Frame File

Table of Contents

1. Report
   • Year=1995
     • Detailed and/or summarized report
       • Table 1
   • Year=1996
     • Detailed and/or summarized report
       • Table 1

2. Tabulate
   • Cross-tabular summary report
     • Table 1
     • Year 1995
     • Year 1996

Table of Pages

1. Report
   1. Page 1
   2. Page 2

2. Tabulate
   1. Page 3
   2. Page 4

Links That Are Created in the HTML Output

These HREF= attributes from the links in the contents file point to the HTML tables that ODS creates from the PROC REPORT and PROC TABULATE steps.

href="grain-body.htm#IDX"
href="grain-body1.htm#IDX1"
href="grain-body2.htm#IDX2"
href="grain-body3.htm#IDX3"

Notice how these HREF attributes are constructed:

• The value of the BODY= option, grain-body, provides the basis for the next part of the HREF. However, because the NEWFILE= option creates a new file for each output object, ODS increments this base value each time it creates a file. The resulting filenames become part of the HREF. They are Grain-Body.htm, Grain-Body1.htm, Grain-Body2.htm, and Grain-Body3.htm.

• The value of the ANCHOR= option provides the basis for the last part of the HREF, which follows the number sign (#). Because the ANCHOR= option is not used in this example, ODS uses the default value of IDX. With each use, ODS increments the value of the anchor.
Example 4: Appending to HTML Files

Features:
ODS HTML statement options:
- ANCHOR=
- BODY= with a fileref
- BODY= using the NO_BOTTOM_MATTER suboption
- BODY= using the NO_TOP_MATTER suboption
- STYLE=

Other features:
- FILENAME statement
- PROC PRINT
- PROC REPORT
- DATA _NULL_ statement

Data set:
- Grain_Production

Format:
- $CNTRY.

Details
The following example creates HTML output from PROC PRINT and PROC REPORT. It also uses the DATA step to write customized HTML code to the file that contains the HTML output. The DATA step executes between procedure steps.

Program

```sas
options obs=10;
filename reports 'GrainReport.html';
ods html body=reports (no_bottom_matter)
  style=Blockprint;
proc print data=grain_production;
  var country type kilotons;
  format country $cntry. kilotons comma12.;
  where year=1996;
  title 'Leading Grain-Producing Countries';
  footnote 'Measurements are in metric tons.';
run;
ods html close;
filename reports '../ods/grain-reports-body.htm' mod;
filename reports 'GrainReport.html' mod;
data _null_;
  file reports;
  put "<h2>The preceding output is from PROC PRINT.</h2>";
  put "I am going to try a variety of procedures.";
  put "Let me know which procedure you prefer.";
  put "This report uses the Blockprint style.</h2>";
run;
ods html body=reports (no_top_matter no_bottom_matter)
```

146 Chapter 6 • Dictionary of ODS Language Statements
Program Description

Set system options. This OBS option limits processing of observations in the data set to 10.

```r
options obs=10;
```

Assign a fileref to the file GrainReport.html. The FILENAME statement assigns the fileref REPORTS to the file GrainReport.html that contains the HTML output.

```r
filename reports 'GrainReport.html';
```

Create HTML output and suppress the writing of the default HTML code that would be written at the end of the file. The ODS HTML statement opens the HTML destination and creates HTML output. The NO_BOTTOM_MATTER option suppresses the writing of the default HTML code that, by default, ODS writes at the end of a file.

```r
ods html body=reports (no_bottom_matter)
```

Specify the style for formatting the HTML output. The STYLE= option specifies that the style Blockprint be used.

```r
style=Blockprint;
```

Create a report that contains only the data from 1996. Select and format the variables that you want to include, specify a title, and specify a footnote. This PROC PRINT step prints the observations in the data set Grain_Production that have a value of 1996 for the variable Year. The VAR statement selects Country, Type, and Kilotons as the variables that you want to be displayed in the output. The TITLE and FOOTNOTE statements specify the title and footnote.

```r
proc print data=grain_production;
var country type kilotons;
format country $cntry. kilotons comma12.;
where year=1996;
title 'Leading Grain-Producing Countries';
footnote 'Measurements are in metric tons.';
run;
```
Close the HTML destination. The ODS HTML CLOSE statement closes the HTML destination and all the files that are associated with it.

    ods html close;
    filename reports '../ods/grain-reports-body.htm' mod;

Assign the fileref REPORTS to the file 'GrainReport.html'. This FILENAME statement assigns a fileref to the file to be updated, GrainReport.html. The MOD option opens the file in Update mode. The MOD option might not be valid in all operating environments. See your operating environment documentation for more information.

    filename reports 'GrainReport.html' mod;

Append text to the HTML file REPORTS. This DATA step writes to the file that is referenced by REPORTS. The PUT statements create an H2 header in the HTML file.

    data _null_;  
        file reports;  
        put "<h2>The preceding output is from PROC PRINT.";  
        put "I am going to try a variety of procedures.";  
        put "Let me know which procedure you prefer.";  
        put "This report uses the Blockprint style.</h2>";  
    run;

Create HTML output. This ODS HTML statement opens the HTML destination and creates HTML output. The NO_TOP_MATTER and NO_BOTTOM_MATTER suboptions suppress the default HTML code that ODS writes to the top and the bottom of a file.

    ods html body=reports (no_top_matter no_bottom_matter)

Specify the root name for the HTML anchor tags. The ANCHOR= option specifies report as the root name for the HTML anchor tags. When you use ODS to append to an HTML file that ODS created, you must specify a new anchor name each time you open the file from ODS so that you do not write the same anchors to the file again. (ODS cannot recognize anchors that are already in the file when it opens it, and by default it uses IDX as the base for anchor names).

    anchor='report';

Create a report that contains only the 1996 data. The PROC REPORT step prints the data set. ODS adds HTML output to the body file. The NOWINDOWS option specifies that PROC REPORT runs without the REPORT window and sends its output to the open output destination(s).

    proc report data=grain_production nowindows;  
        where year=1996;  
        column country type kilotons;  
        define country / group width=14 format=$cntry.;;  
        define type / group 'Type of Grain';  
        define kilotons / format=comma12.;  
        run;

Close the HTML destination. The ODS HTML CLOSE statement closes the HTML destination and all the files that are associated with it.

    ods html close;
Append text to the HTML file REPORTS. This DATA step writes to the file that is referenced by REPORTS. The PUT statements create an H2 header in the HTML file.

```sql
data _null_
  file reports;
  put "<h2>The preceding output is from PROC REPORT."
  put "It does not repeat the name of the country on every line."
  put "This report uses the default style.";
run;
```

Create HTML output to write the bottom matter to the file, repress the printing of the top matter, and provide a new root name for the anchor tags. In order to write the bottom matter to the HTML file so that it contains valid HTML code, you must open the HTML destination one more time. NO_TOP_MATTER ensures that the top matter is not placed in the file again. ANCHOR= provides a new root name for the anchors in the bottom matter.

```sql
ods html body=reports(no_top_matter)anchor='end';
```

**HTML Output**

This output is created by appending HTML output to an existing HTML file.
Leading Grain-Producing Countries

<table>
<thead>
<tr>
<th>Obs</th>
<th>Country</th>
<th>Type</th>
<th>Kilotons</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Brazil</td>
<td>Wheat</td>
<td>3,302</td>
</tr>
<tr>
<td>17</td>
<td>Brazil</td>
<td>Rice</td>
<td>10,035</td>
</tr>
<tr>
<td>18</td>
<td>Brazil</td>
<td>Corn</td>
<td>31,975</td>
</tr>
<tr>
<td>19</td>
<td>China</td>
<td>Wheat</td>
<td>109,000</td>
</tr>
<tr>
<td>20</td>
<td>China</td>
<td>Rice</td>
<td>190,100</td>
</tr>
<tr>
<td>21</td>
<td>China</td>
<td>Corn</td>
<td>119,350</td>
</tr>
<tr>
<td>22</td>
<td>India</td>
<td>Wheat</td>
<td>62,620</td>
</tr>
<tr>
<td>23</td>
<td>India</td>
<td>Rice</td>
<td>120,012</td>
</tr>
<tr>
<td>24</td>
<td>India</td>
<td>Corn</td>
<td>8,660</td>
</tr>
<tr>
<td>25</td>
<td>Indonesia</td>
<td>Wheat</td>
<td></td>
</tr>
</tbody>
</table>

Measurements are in metric tons.

The preceding output is from PROC PRINT. I am going to try a variety of procedures. Let me know which procedure you prefer. This report uses the Banker style.

Leading Grain-Producing Countries

<table>
<thead>
<tr>
<th>Country</th>
<th>Type of Grain</th>
<th>Kilotons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil</td>
<td>Corn</td>
<td>31,975</td>
</tr>
<tr>
<td></td>
<td>Rice</td>
<td>10,035</td>
</tr>
</tbody>
</table>

**Example 5: Removing the Horizontal Rule between Procedures**

**Features:**

- OPTIONS option:
  - PAGEBREAK=NO

**Other features:**

- GPLOT procedure
- PRINT procedure
- GOPTIONS statement

**Details**

HTML documents are usually formed as one continuous page of information without any page breaks. When the document is printing, objects falling on the margins of the printed page might be split across two pages. To prevent this, the ODS HTML destination inserts page breaks between each output object by inserting a paragraph tag that automatically includes a page break command. When printed, each output object
appears on a separate page. Sometimes, with smaller documents, it might be desirable to remove these hardcoded page breaks. This example shows you how to remove the page breaks.

**Program**

```sas
options nodate obs=10;
goptions xpixels=500 ypixels=400;

ods html options(pagebreak='no');
title "Student Correlation";
symbol1 font="albnay amt" value='O' height=15pt color=pink;
symbol2 font="albnay amt" value='X' height=15pt color=lib;
proc gplot data=sashelp.class;
plot height*weight=sex / des="" name="name";
run;

title;
proc print data=sashelp.class;
run;
quit;
odc html close;
```

**Program Description**

**Set the options and goptions.** The OPTIONS statement sets the global options. The GOPTIONS statement set the graphical options.

```sas
options nodate obs=10;
goptions xpixels=500 ypixels=400;
```

**Specify that no page break is created.** By default, there is a page break that resolves as a horizontal rule between the graph and the table when the two procedures are run. The PAGEBREAK= NO suboption specifies that no page break is drawn between the two output objects.

For information about the OPTIONS option, see “Example 7: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information ” on page 532 and “OPTIONS ( DOC= | <suboption(s)> )” on page 129.

```sas
ods html options(pagebreak='no');
```

**Create the graph.** The SYMBOL statements and the GPLOT procedure create the graph.

```sas
title "Student Correlation";
symbol1 font="albnay amt" value='O' height=15pt color=pink;
symbol2 font="albnay amt" value='X' height=15pt color=lib;
proc gplot data=sashelp.class;
plot height*weight=sex / des="" name="name";
run;
```

**Print the output.** The PRINT procedure prints the data set.
title;
proc print data=sashelp.class;
run;
quit;
ods html close;
Example 5: Removing the Horizontal Rule between Procedures

Output

Output 6.5  Default Output with Page Break

---

Student Correlation

---

<table>
<thead>
<tr>
<th>Obs</th>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69.0</td>
<td>112.5</td>
</tr>
<tr>
<td>2</td>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84.0</td>
</tr>
<tr>
<td>3</td>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98.0</td>
</tr>
<tr>
<td>4</td>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
<td>102.5</td>
</tr>
<tr>
<td>5</td>
<td>Henry</td>
<td>M</td>
<td>14</td>
<td>63.5</td>
<td>102.5</td>
</tr>
<tr>
<td>6</td>
<td>James</td>
<td>M</td>
<td>12</td>
<td>57.3</td>
<td>83.0</td>
</tr>
<tr>
<td>7</td>
<td>Jane</td>
<td>F</td>
<td>12</td>
<td>59.8</td>
<td>84.5</td>
</tr>
<tr>
<td>8</td>
<td>Janet</td>
<td>F</td>
<td>15</td>
<td>62.5</td>
<td>112.5</td>
</tr>
<tr>
<td>9</td>
<td>Jeffrey</td>
<td>M</td>
<td>13</td>
<td>62.5</td>
<td>84.0</td>
</tr>
<tr>
<td>10</td>
<td>John</td>
<td>M</td>
<td>12</td>
<td>59.0</td>
<td>99.5</td>
</tr>
</tbody>
</table>
Output 6.6  Output with No Page Break

See Also

- Appendix 3, “ODS and the HTML Destination,” on page 1101

Statements

- “ODS MARKUP Statement” on page 488
- “ODS Tagset Statement” on page 772
FILE Statement for ODS

Creates an ODS output object by binding the data component to the table template. As an option, the FILE Statement lists the variables to include in the ODS output, and it specifies options that control how the variables are formatted.

Valid in: DATA step
Category: File-Handling
Type: Executable
Default: ODS sends the output object to all open ODS destinations.
Note: This syntax shows only the ODS form of the FILE statement. For the complete syntax, see “FILE Statement” in SAS Statements: Reference.

Syntax

FILE PRINT ODS <=(ODS-suboption(s))> <options> ;

Required Arguments

PRINT

is a reserved fileref that you must use when you direct output to ODS.

Requirement You must use PRINT in a FILE statement that uses the ODS option.

See “Example 1: Creating a Report with the DATA Step and the Default Table Definition” on page 69

ODS<= (ODS-suboptions)>

defines the structure of the data component and binds the data component to a table template. The result is an ODS output object. ODS sends this object to all open ODS destinations.

See “ODS Suboptions” on page 156 for information about the ODS suboptions

Optional Arguments

N=number

specifies the number of lines that are available to the output pointer in the current iteration of the DATA step.

overflow-control
determines the PUT statement behavior when the output pointer attempts to move past the last ODS column in the buffer.

overflow-control is one of the following:

DROPOVER
discards items when a PUT statement attempts to write beyond the last ODS column in the buffer. A message in the log at the end of the DATA step informs you if data was not written to the buffer.
FLOWOVER
moves the output pointer to a new line if a PUT statement attempts to write an
item beyond the last ODS column in the buffer. The PUT statement writes the
next item in the first ODS column of the new line.

STOPOVER
stops processing the DATA step immediately if a PUT statement attempts to
write beyond the last ODS column in the buffer. SAS discards the data item,
writes the portion of the buffer that was built before the error occurred, and
issues an error message.

Default FLOWOVER

**ODS Suboptions**

**Table 6.3 ODS Suboptions**

<table>
<thead>
<tr>
<th>Task</th>
<th>Suboption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify one or more columns for the data component</td>
<td>COLUMNS= or VARIABLES= on page 156</td>
</tr>
<tr>
<td>Specify default values for dynamic-attribute values</td>
<td>DYNAMIC= on page 158</td>
</tr>
<tr>
<td>Specify whether all column definitions in the table template can be used by more than one variable</td>
<td>“GENERIC=ON</td>
</tr>
<tr>
<td>Specify a column heading to use for any column that does not have a column specified in the COLUMNS= or VARIABLES= suboption</td>
<td>LABEL= on page 159</td>
</tr>
<tr>
<td>Specify a name for the output object that the DATA step produces</td>
<td>OBJECT= on page 159</td>
</tr>
<tr>
<td>Specify a label for the output object that the DATA step produces</td>
<td>OBJECTLABEL= on page 160</td>
</tr>
<tr>
<td>Specify the table template to use with the data component to produce the output object</td>
<td>TEMPLATE= on page 160</td>
</tr>
</tbody>
</table>

**COLUMNS=(column-specification(s))**
specifies one or more columns for the data component and determines their order in
the data component.

Each **column-specification** associates a DATA step variable with a column that is defined in the table template. **column-specification** has this general form:

\[(\text{column-name-1}=\text{variable-name-1}(\text{attribute-suboptions})> >\]
\[<\ldots \text{column-name-n}=\text{variable-name-n}(\text{attribute-suboptions})> > ) \]

**column-name**
is the name of a column. This name must match the name that is defined in the table template that you use.

**Restriction** **column-name** must conform to the rules for SAS variable names.

**Requirement** You must enclose a **column-name** in parentheses.
Tip

You can use list notation (for example, score1-score5) to specify multiple column names.

Example

“Example 4: Creating and Using a User-Defined Table Template” on page 85

variable-name

specifies a variable in the DATA step to place in the specified column.

Default

If you omit variable-name, then ODS looks for a DATA step variable named column-name to place in the specified column. If no such variable exists, then ODS returns an error.

Tip

You can use list notation (for example, score1-score5) to specify a range of variable names.

Example

“Example 4: Creating and Using a User-Defined Table Template” on page 85

(attribute-suboptions)

assigns a characteristic, such as a label or a format, to a particular column in the data component. These individual specifications override any attributes that are set by the DATA step.

The following table lists the attribute suboptions that are available for the COLUMNS= suboption. For a complete description, see “Attribute Suboptions” on page 162.

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute Suboption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a value for the variable defined by the DYNAMIC statement in a table template</td>
<td>DYNAMIC= on page 162</td>
</tr>
<tr>
<td>Specify a format for the current column</td>
<td>FORMAT= on page 162</td>
</tr>
<tr>
<td>Specify whether the DATA step uses this column definition for multiple variables</td>
<td>GENERIC= on page 163</td>
</tr>
<tr>
<td>Specify a label for a particular column</td>
<td>LABEL= on page 163</td>
</tr>
</tbody>
</table>

Requirement

You must enclose attribute-suboptions in parentheses.

Restrictions

You can use only one COLUMNS= suboption in a FILE PRINT ODS statement.

You can use either the COLUMNS= suboption or the VARIABLES= suboption, but not both, in a single FILE PRINT ODS statement.

Requirement

You must enclose a column-specification in parentheses.

Tips

The order of the columns in the output object is determined by their order in the table template, not by their order in the data component.

To override the default order, use the ORDER_DATA= table attribute in the PROC TEMPLATE step that creates the definition. The default
DATA step table template uses this attribute. For more information, see the discussion of ORDER_DATA=.

If you do not specify COLUMNS= or VARIABLES=, then the order of columns in the data component matches the order of the corresponding variables in the program data vector.

**DYNAMIC=(dynamic-specification(s))**

specifies default values for dynamic-attribute values.

A dynamic-attribute value is defined in the table template. Its name serves as a placeholder for the value that is supplied to the data component with the DYNAMIC= suboption. When ODS creates the output object from the table template and the data component, it substitutes the appropriate value from the data component for the value's name in the table template.

Each dynamic-specification has the following form:

dynamic-value-name<=variable-name | constant>

**dynamic-value-name**

is the name that the table template gives to a dynamic-attribute value.

**variable-name**

specifies a variable whose value is assigned to dynamic-value-name and passed to ODS to substitute for the placeholder in the table template when it creates the output object.

**constant**

specifies a constant to assign to dynamic-value-name and pass to ODS to substitute for the placeholder in the table template when it creates the output object.

**Default**

By default, the DYNAMIC= suboption applies to all columns in the data component.

**Interaction**

Columns that do not contain their own DYNAMIC= suboption specifications use these dynamic-specifications.

**Tip**

You can override the default specification for an individual column by specifying the DYNAMIC= suboption as an attribute for that column in the COLUMNS= or the VARIABLES= suboption.

**See**

“DYNAMIC Statement” in *SAS Output Delivery System: Procedures Guide*

**GENERIC=ON | OFF**

indicates whether the DATA step uses all column definitions for multiple variables.

**ON**

indicates that the DATA step uses all column definitions for multiple variables.

**OFF**

indicates that the DATA step uses no column definitions for multiple variables.

**Defaults**

OFF

By default, the GENERIC= suboption applies to all columns in the data component.
Restriction
ODS does not recognize the column names as a match unless you specify the (COLUMNS=(GENERIC=ON)) suboption.

Interaction
If you do not specify a table template, the GENERIC= suboption is set to ON.

Tips
To override the default specification for an individual column, specify the GENERIC= suboption as an attribute for that column in the COLUMNS= or the VARIABLES= suboption.

The GENERIC= option in the DATA step is used in conjunction with the GENERIC= column attribute in the table template. See the GENERIC= column attribute in “Column Attributes” in SAS Output Delivery System: Procedures Guide.

\textbf{LABEL='\textit{column-label}'}

specifies a label for any column that does not have a label specified in the COLUMNS= or VARIABLES= suboption.

Default
If you use the LABEL= suboption, ODS uses the first of these labels that it finds:

1. a label that is specified with the HEADER= attribute for a particular column in the table template (see HEADER=)
2. a label that is specified for a particular column with the LABEL= suboption in the COLUMNS= or VARIABLES= suboption
3. a label that is specified with the LABEL= suboption in the ODS= option
4. a label that is assigned with the LABEL statement in the DATA step

Tip
If you omit the LABEL= suboption, the contents of the table template determines whether the column heading contains the variable name or is blank.

Example
“Example 3: Assigning Attributes to Columns in ODS Output” on page 78

\textbf{OBJECT= \textit{object-name}}

specifies a name for the output object.

The Results window and the HTML contents file both contain a description of, and a link to, each output object. The description contains the first of the following items that ODS finds:

- the object's label
- the current title if it is not the default title, "The SAS System"
- the object's name
• the string FilePrint#, in which # increases by 1 for each DATA step that you run in the current SAS process without specifying an object name or an object label

Restriction object-name must conform to the rules for SAS variable names. For information about these rules, see “Rules for Words and Names in the SAS Language” in SAS Language Reference: Concepts.

OBJECTLABEL=’object-label’
specifies a label for the output object.

The Results window and the HTML contents file both contain a description of, and a link to, each output object. The description contains the first of the following items that ODS finds:
• the object's label
• the current title if it is not the default title, "The SAS System"
• the object's name (see OBJECT= on page 159)
• the string FilePrint#, in which # increases by 1 for each DATA step that you run in the current SAS process without specifying an object name or an object label

Requirement You must enclose an object-label in quotation marks.

Example “Example 3: Assigning Attributes to Columns in ODS Output” on page 78

TEMPLATE= 'table-definition-name'
specifies the table definition to use with the data component to produce the output object.

table-definition-name
is the path to the table template. SAS stores a table template as an item in an item store.

Defaults If you do not specify the TEMPLATE= option, ODS uses BASE.DATASTEP.TABLE, the default table template.

If you do specify the TEMPLATE= suboption, ODS first looks for table-definition-name in Sasuser.Templat, and then it looks in Sashelp.Tmplmst.

Requirement You must enclose a table-definition-name in quotation marks.

Interaction When you use the default table template, the GENERIC= suboption is set to ON for all columns in the data component. For more information, see GENERIC= on page 158.

Tips When you use the BASE.DATASTEP.TABLE template, character values are left-justified. If you want character values to be right-justified, specify the BASE.DATASTEP.TABLENOJUST template.

You can change the locations in which ODS searches for the table-definition-name by using the ODS PATH on page 563 statement.
Example 4: Creating and Using a User-Defined Table Template

**VARIABLES=(variable-specification(s))**

specifies one or more columns for the data component of the output object. Each variable-specification associates a DATA step variable with a column that is defined in the table template. The variable-specification value has this general form:

```
(variable-name-1<=column-name-1<attribute-suboptions>>
  <… variable-name-n<=column-name-n<attribute-suboptions>>>)
```

**variable-name**

specifies a variable in the DATA step to use as a column in the data component.

**Tip**

You can use list notation (for example, `score1-score5`) to specify a range of variable names.

**Examples**

“Example 2: Producing ODS Output That Contains Selected Variables” on page 73

“Example 3: Assigning Attributes to Columns in ODS Output” on page 78

**column-name**

is the name of a column. This name must match a name that is defined in the table template.

**Defaults**

If you are using the default table template and you omit column-name, then ODS uses the variable label to name the column. If the variable has no label, then ODS uses the variable name.

If you use a table template other than the default table template and you omit column-name, ODS looks in the table template for a column that is named variable-name and places the variable in that column. ODS returns an error if no such column exists.

**Restriction**

column-name must match a column name in the table template that you are using. It must also conform to the rules for SAS variable names. For information about these rules, see “Rules for Words and Names in the SAS Language” in *SAS Language Reference: Concepts*.

**Tip**

You can use list notation (for example, `score1-score5`) to specify a range of column names.

**attribute-suboptions**

assigns a characteristic, such as a label or a format, to a particular column in the data component. These individual specifications override any attributes that are set in the DATA step for the entire data component.

The following table lists the attribute suboptions available for the VARIABLES= suboption. For a complete description, see “Attribute Suboptions” on page 162.

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute Suboption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify a value for the variable defined by the DYNAMIC statement in a table template</td>
<td>DYNAMIC= on page 162</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute Suboption</td>
</tr>
<tr>
<td>------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Specify a format for the current column</td>
<td>FORMAT= on page 162</td>
</tr>
<tr>
<td>Specify whether the DATA step uses this column definition for multiple variables</td>
<td>GENERIC= on page 163</td>
</tr>
<tr>
<td>Specify a label for a particular column</td>
<td>LABEL= on page 163</td>
</tr>
</tbody>
</table>

**Default**

If you specify the VARIABLES= suboption, the order of the columns in the output object is determined by their order in the table template, not by their order in the data component. If you do not specify COLUMNS= or VARIABLES= suboptions, the order of columns in the data component matches the order of the corresponding variables in the program data vector.

**Restrictions**

You can use only one VARIABLES= suboption in a FILE PRINT ODS statement.

You can use either the COLUMNS= suboption or the VARIABLES= suboption to associate variables with columns, but you cannot use both suboptions in the same FILE PRINT ODS statement.

**Tips**

To override the default order, use the ORDER_DATA table attribute in the PROC TEMPLATE step that creates the definition. The default DATA step table template uses this attribute. For more information, see the ORDER_DATA=.

The VARIABLES= suboption is for use primarily with the default DATA step table template. When you use the default definition, the DATA step can map variables to the appropriate column in the definition so that you do not need to specify a column name.

**Examples**

“Example 2: Producing ODS Output That Contains Selected Variables” on page 73

“Example 3: Assigning Attributes to Columns in ODS Output” on page 78

**Attribute Suboptions**

**DYNAMIC=** dynamic-specification(s)

specifies a value for the variable defined by the DYNAMIC statement in a table template.

**See**

DYNAMIC= suboption on page 158

DYNAMIC Statement

**Example**

“Example 4: Creating and Using a User-Defined Table Template” on page 85

**FORMAT=** format-name

specifies a format for the current column.
ODS uses the first of these formats for the variable that it finds:

1. for nongeneric columns, a format that is specified in the column definition
2. a format that is specified in the FORMAT= column attribute
3. a format that is specified in a FORMAT statement
4. the default format ($w. for character variables; BEST12. for numeric variables)

Note Formats for generic columns that are specified in the table template are ignored by the DATA step interface to ODS.

Example “Example 4: Creating and Using a User-Defined Table Template” on page 85

**GENERIC=ON | OFF**
specifies whether the DATA step uses this column definition for multiple variables.

Default OFF

Tip The GENERIC= option in the DATA step is used in conjunction with the GENERIC= column attribute in the table template. See the GENERIC= column attribute in “Column Attributes” in *SAS Output Delivery System: Procedures Guide*.

See GENERIC= suboption on page 158

Example “Example 4: Creating and Using a User-Defined Table Template” on page 85

**LABEL='column-label'**
specifies a label for the specified column.

See LABEL= suboption on page 159

“Example 3: Assigning Attributes to Columns in ODS Output” on page 78

**Details**

**Restrictions When Using the FILE Statement with ODS**
The following restrictions apply to the FILE statement when you use it with ODS:

- These arguments affect only listing output:
  - FOOTNOTES and NOFOOTNOTES
  - LINESIZE
  - PAGESIZE
  - TITLE and NOTITLES
- Do not use these arguments:
  - DELIMITER=
Using Options and Suboptions

Options apply to all columns and suboptions apply to specific columns.

For example, both of the following DATA steps produce the same output. This DATA step specifies the suboption GENERIC=ON for every column.

**Example Code 6.1  DATA Step Using the GENERIC=ON Suboption**

```plaintext
data _null_
  set top3list;
    file print ods = (
      template='means.topn'
      columns={
        class=school(generic=on)
        class=year(generic=on)
        sum=moneyRaised_sum(generic=on)
        mean=moneyRaised_mean(generic=on)
        raised=moneyRaised_1(generic=on)
        raised=moneyRaised_2(generic=on)
        raised=moneyRaised_3(generic=on)
        name=name_1(generic=on)
        name=name_2(generic=on)
        name=name_3(generic=on)
        school=school_1(generic=on)
        school=school_2(generic=on)
        school=school_3(generic=on)
        year=year_1(generic=on)
        year=year_2(generic=on)
        year=year_3(generic=on)
      }
    );
    put _ods_;
run;
```

This DATA step uses the GENERIC=ON option, which has to be specified only once.

**Example Code 6.2  DATA Step Using the GENERIC=ON Option**

```plaintext
data _null_
  set top3list;
    file print ods = (
      template='means.topn'
      generic=on
      columns={
        class=school
        class=year
        sum=moneyRaised_sum
```
Without ODS Suboptions
If you do not specify any ODS suboptions, the DATA step uses a default table template (BASE.DATASTEP.TABLE) that is stored in the Sashelp.Tmplmst template store. This definition defines two generic columns: one for character variables and one for numeric variables. ODS associates each variable in the DATA step with one of these columns and displays the variables in the order in which they are defined in the DATA step.

If there are no suboptions, the default table template uses the variable's label as its column heading. If no label exists, the definition uses the variable's name as the column heading.

See Also
• Chapter 3, “Using ODS with the DATA Step,” on page 53
• Examples on page 69

Statement
• “PUT Statement for ODS ” on page 864

LIBNAME Statement, SASEDOC
Uses the SASEDOC engine to associate a SAS libref (library reference) with one or more ODS output objects that are stored in an ODS document.

Valid in: Anywhere
Category: ODS: Output Control
Restriction: The LIBNAME statement that is used with the SASEDOC engine provides Read access to an output object. You cannot write an output object to a library with the SASEDOC engine, but you can delete or rename a data set.

Syntax
LIBNAME libref SASEDOC 'path' <sasedoc-engine-option> <options> ;
**Required Arguments**

**libref**

is a shortcut name or a nickname for the aggregate storage location where your SAS files are stored. It is any SAS name that you choose for assigning a new libref. When you are disassociating a libref from a SAS library, or when you are listing attributes, specify a libref that was previously assigned or else use the CLEAR argument.

**Tip**  
The association between a libref and a SAS library lasts only for the duration of the SAS session or until you change it or discontinue it with another LIBNAME statement for the same libref.

**SASEDOC**

is the name of the engine that associates a SAS libref (library reference) with one or more ODS output objects that are stored in an ODS document.

**path**

is the fully specified location of an ODS document directory.

**SASEDOC Engine Options**

**DOC_SEQNO=sequence-number**

permits you to specify the sequence number of the output object to be accessed. This is necessary when multiple output objects that are in the same directory have the same name. By default, the SASEDOC LIBNAME engine can access only the most recently created output object, which might not be the one that you want to access. Specify DOC_SEQNO to override the default.

**sequence-number**

is a number which, when combined with a pathname, uniquely identifies the entry in the directory.


**Additional LIBNAME Statement Arguments and Options**

For additional arguments and options that are valid for the LIBNAME statement, see the LIBNAME statement in *SAS Statements: Reference*.

**Details**

The SASEDOC LIBNAME engine permits you to access output objects that are stored in an ODS document. A data set that is accessed by using the SASEDOC LIBNAME engine might differ structurally from one created by replaying the ODS document output object to the ODS OUTPUT destination. This is because the ODS OUTPUT destination recognizes the output object's template, but the SASEDOC LIBNAME engine does not.

**Example: Assigning a LIBNAME to an ODS DOCUMENT**

**Features:**

- ODS DOCUMENT statement: NAME= option

**Other features:**

- LIBNAME statement: DOC_SEQNO option
- PROC DATASETS
- PROC GLM
- PROC PRINT
**Data sets:**
- Plants
- Plant_Stat

**Details**
This example assigns a libref to an ODS document directory that contains four output objects created by PROC GLM. The four output objects are tables:

- Overall ANOVA
- Fit statistics
- Type I model ANOVA
- Type III model ANOVA

**Program**
```sas
ods document name=sasuser.odsglm(write);
proc glm data=plant_stats;
  class month;
  model age age2 age3=month / nouni;
  manova h=month /print;
run;

proc glm data=plants order=data;
  class type block;
  model stemleng=type block;
  means type;
  contrast 'compost vs others' type -1 -1 -1 6 -1 -1;
  contrast 'river soils vs non' type -1 -1 -1 0 5 -1,
    type -1 4 -1 -1 0 0 -1;
  contrast 'glacial vs drift' type -1 0 1 1 0 0 -1;
  contrast 'clarion vs webster' type -1 0 0 0 0 0 1;
  contrast 'knox vs oneill' type 0 0 1 -1 0 0 0;
quit;

ods document close;
libname mylib sasedoc '\sasuser.odsglm\glm\anova#1\stemleng';

proc datasets lib=mylib;
run;
quit;

proc print data=mylib.modelanova;
run;
proc print data=mylib.modelanova(doc_seqno=1);
run;
```

**Program Description**

Create the ODS document and open the DOCUMENT destination. The ODS DOCUMENT statement opens the DOCUMENT destination. The NAME= option assigns the name `sasuser.odsglm` to the ODS document that contains the output from the
PROC GLM program. The access-option WRITE provides Write access to the document. Note that odsglm is created in the Ssuser library.

```
ods document name=sasuser.odsglm(write);
```

Create the output objects. The GLM procedure creates the output objects. The Plant Stats data set contains the statistical information that PROC GLM uses to create the output objects.

For information about viewing a record of each output object that is created, see the “ODS TRACE Statement” on page 854.

```
proc glm data=plant_stats;
  class month;
  model age age2 age3=month / nouni;
  manova h=month /print;
run;
```

Close the DOCUMENT destination. If you do not close the DOCUMENT destination, you cannot see DOCUMENT procedure output.

```
ods document close;
```

Associate the libref mylib with the directory stemleng. The LIBNAME statement uses the SASEDOC engine to associate the SAS libref mylib with the directory stemleng that is stored in the ODS document sasuser.odsglm. Notice that the path includes anova#1 and not just anova. This is because there are two anova directories, and this code is specifying the first directory. If the sequence number was omitted, then ODS would associate the libref with the second directory.

```
libname mylib sasedoc '\sasuser.odsglm\glm\anova#1\stemleng';
```

The LIBRARY= option specifies mylib as the procedure input library. The QUIT statement stops the DATASETS procedure.

```
proc datasets lib=mylib;
run;
quit;
```
Print the data sets. Since two output objects have the same name (ModelANOVA), the SASEDOC LIBNAME engine recognizes only the second table, because it was created more recently than the first table. The DOC_SEQNO= data set option specifies a sequence number of 1 in order to access the first table.

```sas
proc print data=mylib.modelanova;
run;
proc print data=mylib.modelanova(doc_seqno=1);
run;
```

Output

The first display shows the Explorer window that contains the SAS library Mylib which is associated with the directory stemleng. The stemleng directory is stored in the ODS document sasuser.odsglm. The second display shows the Explorer window that contains the contents of the SAS library Mylib. The three output objects are actually stored in an ODS document.

Output 6.7 Explorer Window

![Explorer Window](image1)

Output 6.8 The Contents of Mylib

![The Contents of Mylib](image2)

See Also

Procedures


Statements

- “ODS DOCUMENT Statement” on page 208
- “ODS TRACE Statement” on page 854

ODS _ALL_ CLOSE Statement

Closes all open ODS output destinations.
Valid in: Anywhere
Category: ODS: Output Control

Syntax

```ods_all_close;
```

Details

The ODS _ALL_ CLOSE statement closes all open ODS output destinations.

*Note:* Be sure to open one or more ODS destinations before you execute your next program so that you can view or print your output within the same SAS session.

---

**ODS CHTML Statement**

Opens, manages, or closes the CHTML destination, which produces a compact, minimal HTML that does not use style information.

Valid in: Anywhere
Category: ODS: Third-Party Formatted
Default: The default style for Markup family destinations is HTMLBlue.

Syntax

```ods_chtml <(id=identifier)>< action> ;
ods_chtml <(id=identifier)><option(s)> ;
```

Summary of Optional Arguments

- `(ID=identifier)`
  - Open multiple instances of the same destination at the same time
- `ANCHOR='anchor-name'`
  - Specify a unique base name for the anchor tag that identifies each output object in the current body file
- `ARCHIVE='string'`
  - Specify which applet to use to view ODS HTML output
- `ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)`
  - Specify attributes to write between the tags that generate dynamic graphics output
- `BASE='base-text'`
  - Specify text to use as the first part of all links and references that ODS creates in output files
- `BODY= 'file-specification' (suboption(s))`
  - Open a markup family destination and specify the file that contains the primary output that is created by the ODS statement
- `CHARSET= character-set`
  - Specify the character set to be generated in the META declaration for the HTML output
CLOSE
    Close the destination and the file that is associated with it

CODE='file-specification' <(suboption(s))>
    Open the HTML destination and specify the file that contains relevant style
    information

CODEBASE='string'
    Create a file path that can be used by the GOPTIONS devices

CONTENTS='file-specification' <(suboption(s))>
    Open the HTML destination and specify the file that contains a table of
    contents for the output

DOM="external-file">
    Specify that the ODS document object model is written to the SAS log or to
    an external file.

ENCODING= local-character-set-encoding
    Override the encoding for input or output processing (transcodes) of external
    files

EVENT=event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )
    Specify an event and the value for event variables that is associated with the
    event

EXCLUDE exclusion(s) | ALL | NONE
    Exclude output objects from the destination

FRAME='file-specification' <(suboption(s))>
    Specify the file that integrates the table of contents, the page contents, and the
    body file

GFOOTNOTE | NOGFOOTNOTE
    Control the location where footnotes are printed in the graphics output

GPATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL=
    'Uniform-Resource-Locator' | NONE)
    Specify the location for all graphics output that is generated while the
    destination is open

GTITLE | NOGTITLE
    Control the location where titles are printed in the graphics output

HEADTEXT='markup-document-head'
    Specify HTML tags to place between the < HEAD> and </HEAD> tags in
    all of the output files.

METATEXT='metatext-for-document-head'
    Specify HTML code to use as the <META> tag between the <HEAD> and </HEAD>
    tags in all of the HTML output files.

NEWFILE= starting-point
    Create a new body file at the specified starting point

OPTIONS ( DOC= | <suboption(s)> )
    Specify tagset-specific suboptions and a named value

PACKAGE <package-name>
    Specify that the output from the destination be added to an ODS package

PAGE='file-specification' <(suboption(s))>
    Open the HTML destination and specify the file that contains a description of
    each page of the body file and contains links to the body file

PARAMETERS= (parameter-pair-1 ... parameter-pair-n)
    Write the specified parameters between the tags that generate dynamic
    graphics output
PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)
    Specify the location of an aggregate storage location or a SAS catalog for all markup files
RECORD_SEPARATOR= 'alternative-separator' | NONE
    Specify an alternative character or string to separate lines in the output files
SELECT selection(s) | ALL | NONE
    Select output objects for the destination
SHOW
    Write to the SAS log the current selection or exclusion list for the destination
STYLE= style-template
    Specify a style template to use in writing output files
STYLESHEET='file-specification' <(suboption(s))>
    Open the HTML destination and place style information for output into an external file, or read style sheet information from an existing file
TEXT= text-string
    Insert text into your document
TRANTAB= 'translation-table'
    Specify a translation table to use when transcoding a file for output

Without Arguments
If you use the ODS CHTML statement without an action or options, then it opens the CHTML destination and creates CHTML output.

Actions
The following actions are available for the ODS CHTML statement:

CLOSE
    closes the destination and any files that are associated with it.
    Tip  When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.
EXCLUDE exclusion(s) | ALL | NONE
    excludes one or more output objects from the destination.
    Default  NONE
    Restriction  A destination must be open for this action to take effect.
    See  “ODS EXCLUDE Statement” on page 321
SELECT selection(s) | ALL | NONE
    selects output objects for the specified destination.
    Default  ALL
    Restriction  A destination must be open for this action to take effect.
    See  “ODS SELECT Statement ” on page 758
SHOW
    writes the current selection list or exclusion list for the destination to the SAS log.
    Restriction  The destination must be open for this action to take effect.
Tip  If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

See  “ODS SHOW Statement” on page 771

Optional Arguments
The following options are available for the ODS CHTML statement, which is part of the markup family of statements:

ANCHOR= 'anchor-name'
specifies a unique base name for the anchor tag that identifies each output object in the current body file.

Each output object has an anchor tag for the contents, page, and frame files to reference. The links and references are automatically created by ODS. The links and references point to the name of an anchor. Therefore, each anchor name in a file must be unique.

anchor-name
is the base name for the anchor tag that identifies each output object in the current body file.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR= 'TABULATE', then ODS names the first anchor tabulate. The second anchor is named tabulate1; the third is named tabulate2, and so on.

Restrictions  Each anchor name in a file must be unique.

Only alphanumeric values, the special characters "$ - . + ! * ' ( ) , " and reserved characters used for their reserved purposes can be used unencoded within a URL.

Requirement  You must enclose anchor-name in quotation marks.

Interaction  If you open a file to append to it, be sure to specify a new anchor name to prevent writing the same anchors to the file again. ODS does not recognize anchors that are already in a file when it opens the file.

Tips  You can change anchor names as often as you want by specifying the ANCHOR= option in a markup family statement anywhere in your program. After you have specified an anchor name, it remains in effect until you specify a new one.

Specifying new anchor names at various points in your program is useful when you want other web pages to link to specific parts of your markup language output. Because you can control where the anchor name changes, you know in advance what the anchor name is at those points.

Anchor-name must begin with a letter ([A-Za-z]) and can be followed by any number of letters, digits ([0-9]), hyphens (-), underscores (_), colons (:), and periods (.)

An anchor-name must begin with a letter ([A-Za-z]) and can be followed by any number of letters, digits ([0-9]), hyphens (-), underscores (_), colons (:), and periods (.)
ARCHIVE='string'

specifies which applet to use to view the ODS HTML output. The ARCHIVE= option is valid only for the GOPTIONS Java device.

The string must be one that the browser can interpret. For example, if the archive file is local to the computer that you are running SAS on, you can use the FILE protocol to identify the file. If you want to point to an archive file that is on a web server, use the HTTP protocol.

Default

If you do not specify ARCHIVE= and you are using the JAVA device driver, ODS uses the value of the SAS system option APPLETOC=. There is no default if you are using the ACTIVEX device driver.

Requirements

You must enclose string in quotation marks.

The ARCHIVE attribute is a feature of Java 1.1. Therefore, if you are using the Java device driver, your browser must support this version of Java. Both Internet Explorer 4.01 and Netscape 4.05 support Java 1.1.

Interaction

Use ARCHIVE= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA or DEVICE=ACTIVEX option in the GOPTIONS statement.

Tips

Typically, this option should not be used, because the SAS server automatically determines the correct SAS/GRAPH applets to view the ODS HTML output. However, if you have renamed the JAR files, or have other applets with which to view the ODS HTML output, this option enables you to access these applets.

Use the CODEBASE= option to specify the file path. It is recommended that you do not put a file path in your ARCHIVE= option.

The value of APPLETOC= points to the location of the Java archive files that are shipped with the SAS system. To find out what the value of this option is, you can either look in the Options window in the Files folder under Environment Control, or you can submit the following procedure step:

```sas
proc options option=appletloc;
run;
```

ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)

writes the specified attributes between the tags that generate dynamic graphics output.

attribute-pair

specifies the name and value of each attribute. attribute-pair has the following form:

'attribute-name'='attribute-value'

attribute-name

is the name of the attribute.

attribute-value

is the value of the attribute.
Requirement: You must enclose attribute-name and attribute-value in quotation marks.

Interaction: Use the ATTRIBUTES= option in conjunction with SAS/GRAph procedures and with the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

See: SAS/GRAph: Reference for valid attributes for the Graph Applet, the Map Applet, the Contour Applet, and the MetaView Applet.

**BASE= ’base-text’**

specifies the text to use as the first part of all links and references that ODS creates in the output files.

*base-text*

is the text that ODS uses as the first part of all links and references that ODS creates in the file.

Consider this specification:

```
BASE= ’http://www.your-company.com/local-url/
```

In this case, ODS creates links that begin with the string http://www.your-company.com/local-url/. The appropriate *anchor-name* completes the link.

Requirement: You must enclose *base-text* in quotation marks.

**BODY= ’file-specification’ (suboption(s))**

opens a markup family destination and specifies the file that contains the primary output that is created by the ODS statement. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

*file-specification*

specifies the file, fileref, or SAS catalog to write to.

*file-specification* is one of the following:

- **external-file**
  
is the name of an external output file.

  Requirement: You must enclose *external-file* in quotation marks.

- **fileref**
  
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

  Restriction: The BODY=fileref option cannot be used in conjunction with the NEWFILE= option.

See: For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

**entry.markup**

specifies an entry in a SAS catalog to write to.
Interaction
If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

(suboption(s))
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 193.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 193.

(NO_TOP_MATTER)
See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 194.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 194.

(URL= 'Uniform-Resource-Locator' )
See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator' )” on page 195.

Alias FILE=
Interaction Using the BODY= option in an ODS markup family statement that refers to an open ODS markup destination forces ODS to close the destination and all associated files. ODS then opens a new instance of the destination. For more information see “Opening and Closing the MARKUP Destination ” on page 517.

Note For some values of TAGSET=, this output is an HTML file. For other TAGSET= values, the output is an XML file, and so on.

CHARSET= character-set
specifies the character set to be generated in the META declaration for the HTML output.

See For information about the CHARSET= option, see “CHARSET= Option” in SAS National Language Support (NLS): Reference Guide.
CODE= 'file-specification' <(suboption(s))>
opens a markup family destination and specifies the file that contains accompanying programming code, such as JavaScript or XSL (Extensible Stylesheet Language). These files remain open until you do one of the following:

• close the destination with either an ODS \textit{markup-family-destination}\ CLOSE statement or ODS _ALL_\ CLOSE statement.
• open the same destination with a second markup family statement. This closes the first file and opens the second file.

\textit{file-specification} specifies the file, fileref, or SAS catalog to write to.

\textit{file-specification} is one of the following:

\textit{external-file}
is the name of an external output file.

\textbf{Requirement} You must enclose \textit{external-file} in quotation marks.

\textit{fileref}
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

\textbf{See} See “FILENAME Statement” in \textit{SAS Statements: Reference}.

\textit{entry.markup}
specifies an entry in a SAS catalog to write to.

\textbf{Interaction} If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

\textit{suboption(s)} specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

\textbf{See} For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 193.

(URL= \textit{Uniform-Resource-Locator})
specifies a URL for the \textit{file-specification}. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

\textbf{See} For complete documentation about the URL= suboption, see “(URL= \textit{Uniform-Resource-Locator})” on page 195.

\textbf{CODEBASE=’string’}
specifies the location of the executable Java applet or the ActiveX control file. \textit{string} is specified as a pathname or as a URL. The CODEBASE file path option has two definitions, depending on the GOPTIONS device used.

When you generate web presentations with the JAVA and ActiveX device drivers, SAS generates HTML pages that automatically look for the JAVA archive files or the ActiveX control file in the default installation location.
For the ActiveX device:

If you use the ActiveX device driver with ODS to generate output containing an ActiveX control, then specify the CODEBASE= option in the ODS statement. The value of the CODEBASE= option should include the location and the version of the EXE file.

Tip You do not need to specify the CODEBASE= option with the DEVICE=ACTIVEX option unless the users that view your output do not have the ActiveX control installed on their machine. When users that do not have the ActiveX control installed view your output, they are prompted to download the control.

See SAS/GRAPH: Reference for information about specifying the location of control and applet files using the CODEBASE= and ARCHIVE= options.

For the Java device:

If you use the Java device driver with ODS to generate output containing a SAS/GRAPH applet, specify the path to the JAR file with the CODEBASE= option in the ODS statement.

When you specify DEVICE=JAVA, the users that view your output must have access to the appropriate Java applet. By default, SAS sets the value of CODEBASE= to refer to the executable file for the applet that is automatically installed with SAS. The default location of the SAS Java archive files is specified by the APPLETLOC= system option. You do not need to specify the CODEBASE= option if both of the following conditions are true.

• The default location is accessible by users who are viewing your web presentation.
• The SAS Java archive is installed at that location.

Tip Specify only the directory of the JAR file. The CODEBASE= location can be specified as a pathname or as a URL.

See SAS/GRAPH: Reference for information about specifying the location of control and applet files using the CODEBASE= and ARCHIVE= options.

CONTENTS='file-specification' <<(suboption(s))>

opens a markup family destination and specifies the file that contains a table of contents for the output. These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file

is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

fileref

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.
entry.markup
specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

**(DYNAMIC)**
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 193.

**(NO_BOTTOM_MATTER)**
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 193.

**(NO_TOP_MATTER)**
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 194.

**(TITLE='title-text')**
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

\textit{title-text} is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 194.

**(URL= 'Uniform-Resource-Locator')**
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 195.

DOM<="external-file">
specifies that the ODS document object model is written to the SAS log or an external file.
external-file

is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

See For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in SAS Output Delivery System: Advanced Topics.

ENCODING= local-character-set-encoding

overrides the encoding for input or output processing (transcodes) of external files.

See For information about the ENCODING= option, see SAS National Language Support (NLS): Reference Guide.

EVENT=event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )

specifies an event and the value for event variables that are associated with the event.

(FILE= BODY | CODE | CONTENTS | DATA | FRAME | PAGES | STYLESHEET);

triggers one of the known types of output files that correspond to the BODY=, CODE=, CONTENTS=, FRAME=, PAGES=, and STYLESHEET= options.

(FINISH)

triggers the finish section of an event.

See For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(LABEL=' variable-value ')

specifies the value for the LABEL event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the LABEL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(NAME=' variable-value ')

specifies the value for the NAME event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the NAME event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(START)

triggers the start section of an event.

See For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(STYLE=style-element)

specifies a style element.

See For information about style elements, see “Style Attributes Overview” in SAS Output Delivery System: Procedures Guide.
(TARGET='variable-value')
specifies the value for the TARGET event variable.

Requirement  
variable-value must be enclosed in quotation marks.

See  
For information about the TARGET event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(TEXT='variable-value')
specifies the value for the TEXT event variable.

Requirement  
variable-value must be enclosed in quotation marks.

See  
For information about the TEXT event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(URL='variable-value')
specifies the value for the URL event variable.

Requirement  
variable-value must be enclosed in quotation marks.

See  
For information about the URL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

Default  
(FILE='BODY')

Requirement  
The EVENT= option's suboptions must be enclosed in parentheses.

FRAME= 'file-specification' <(suboption(s))>
opens a markup family destination and, for HTML output, specifies the file that integrates the table of contents, the page contents, and the body file. If you open the frame file, then you see a table of contents, a table of pages, or both, as well as the body file. For XML output, FRAME= specifies the file that contains the DTD. These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.

• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement  
You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See  
For information about the FILENAME Statement, see “FILENAME Statement” in SAS Statements: Reference.
entry markup
specifies an entry in a SAS catalog to write to.

Interaction
If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 193.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 193.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 194.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

title-text
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 194.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 195.

Restriction
If you specify the FRAME= option, then you must also specify the CONTENTS= option, the PAGE= option, or both.

Example “Example 2: Creating an XML File and a DTD” on page 520

GFOOTNOTE | NOGFOOTNOTE
controls the location where footnotes are printed in the graphics output.
GFOOTNOTE
writes footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

NOGFOOTNOTE
writes footnotes that are created by ODS, which appear outside the graph borders.

 Default  GFOOTNOTE

Restrictions
Footnotes that are displayed by a markup language statement support all SAS/GRAPH FOOTNOTE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH FOOTNOTE statement, see “FOOTNOTE Statement” in SAS/GRAPH: Reference.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

GPATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)
specifies the location for all graphics output that is generated while the destination is open. Use this option when you want to write graphics output files to a location different that specified by the PATH= option for markup files. If you specify an invalid filename, the ActiveX and Java devices send output to the default filename. Other devices create the file as a directory and write output to that directory using the default filename. For information about how ODS names catalog entries and external files, see SAS/GRAPH: Reference.

'aggregate-file-storage-location'
specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement  You must enclose aggregate-file-storage-location in quotation marks.

fileref
is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction  If you specify a fileref in the GPATH= option, then ODS does not use information from the GPATH= option when it constructs links.

See  For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

libref.catalog
specifies a SAS catalog to write to.

URL= 'Uniform-Resource-Locator' | NONE
specifies a URL for file-specification.

Uniform-Resource-Locator
is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.
Requirement
You must enclose Uniform-Resource-Locator in quotation marks.

NONE
specifies that no information from the GPATH= option appears in the links or references.

Tip
This option is useful for building output files that can be moved from one location to another. If the links from the contents and page files are constructed with a simple URL (one name), then they resolve, as long as the contents, page, and body files are all in the same location.

Default
If you omit the GPATH= option, then ODS stores graphics in the location that is specified by the PATH= option. If you do not specify the PATH= option, then ODS stores the graphics in the current directory. For more information, see the PATH= option.

GTITLE | NOGTITLE
controls the location where titles are printed in the graphics output.

GTITLE
writes the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.

NOGTITLE
writes the title that is created by ODS, which appears outside of the graph borders.

Default
GTITLE

Restrictions
Titles that are displayed by any markup language statement support most SAS/GRAPH TITLE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH TITLE statement, see TITLE statement.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

HEADTEXT= 'markup-document-head'
specifies markup tags to place between the <HEAD> and </HEAD> tags in all of the output files.

markup-document-head
specifies the markup tags to place between the <HEAD> and </HEAD> tags.

Restriction
HEADTEXT= cannot exceed 256 characters.

Requirement
You must enclose markup-document-head in quotation marks.

Tips
ODS cannot parse the markup that you supply. It should be well-formed markup that is correct in the context of the <HEAD> and </HEAD> tags.

Use the HEADTEXT= option to define programs (such as JavaScript) that you can use later in the file.
(ID= identifier)
enables you to run multiple instances of the same destination at the same time. Each
instance can have different options.

identifier specifies another instance of the destination that is already open. identifier is
numeric or a series of characters that begin with a letter or an underscore.
Subsequent characters can include letters, underscores, and numeric characters.

Restriction If identifier is numeric, it must be a positive integer.

Requirement You must specify the ID= option immediately after the destination
name.

Tip You can omit the ID= option and instead use a name or a number to
identify the instance.

Example “Example 1: Opening Multiple Instances of the Same Destination at
the Same Time” on page 594

METATEXT= 'metatext-for-document-head'
specifies HTML code to use as the <META> tag between the <HEAD> and </
HEAD> tags of all of the HTML output files.

'metatext-for-document-head'
specifies the HTML code that provides the browser with information about the
document that it is loading. For example, this attribute could specify the content
type and the character set to use.

Requirement You must enclose metatext-for-document-head in quotation
marks.

Default If you do not specify METATEXT=, then ODS writes a simple
<META> tag, which includes the content-type of the document and the
character set to use, to all the HTML files that it creates.

Restriction METATEXT= cannot exceed 256 characters.

Tip ODS cannot parse the HTML code that you supply. It should be well-
formed HTML code that is correct in the context of the <HEAD> tags.
If you are using METATEXT= as it is intended, then your META tag
should look like this:
<META your-metatext-is-here>

NEWFILE= starting-point
creates a new body file at the specified starting-point.

starting-point is the location in the output where you want to create a new body file.

ODS automatically names new files by incrementing the name of the body file.
In the following example, ODS names the first body file REPORT.XML.
Additional body files are named REPORT1.XML, REPORT2.XML, and so on.

Example:

BODY= 'REPORT.XML'

starting-point is one of the following:
BYGROUP
starts a new file for the results of each BY group.

NONE
writes all output to the body file that is currently open.

OUTPUT
starts a new body file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

Alias  TABLE

PAGE
starts a new body file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.

PROC
starts a new body file each time you start a new procedure.

Default  NONE

Restriction  The NEWFILE= option cannot be used in conjunction with the BODY= fileref option.

Tips  If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first body file MAY5.XML. Additional body files are named MAY6.XML, MAY7.XML, and so on.

Example:
BODY= ‘MAY5.XML’

OPTIONS ( DOC= | <suboption(s)> )
specifies tagset-specific suboptions and a named value.

(DOC= 'HELP' | 'QUICK' | 'SETTINGS' | 'CHANGELOG')
provides information about the specified tagset.

HELP
provides generic help and information with a quick reference.

QUICK
describes the options available for this tagset.

SETTINGS
provides the current option settings.

CHANGELOG
lists a history of changes made to the tagset. This suboption is supported only on the RTF tagset.

Requirement  All values must be enclosed in quotation marks.

suboption(s)
specifies one or more suboptions that are valid for the specified tagset. Suboptions have the following format:

keyword= 'value'
Specify one of the following options when opening an ODS tagset statement, or at any time after the destination has been opened, to get information about suboptions for the tagset.

- `options(doc='help');`
- `options(doc='quick');`
- `options(doc='settings');`

Requirement: Suboption(s) must be enclosed in parentheses.

Example: "Example: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information" on page 802

/package-name/
specifies that the output from the destination be added to a package.

package-name
specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

See "ODS PACKAGE Statement" on page 554

Example: "Example 1: Creating an ODS Package" on page 558

/page= 'file-specification' <(suboption(s))>
opens a markup family destination and specifies the file that contains a description of each page of the body file, and contains links to the body file. ODS produces a new page of output whenever a procedure requests a new page. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS ALL CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement: You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see "FILENAME Statement" in SAS Statements: Reference.

entry.markup
specifies an entry in a SAS catalog to write to.

Interaction: If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.
suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for
writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a
file.

See For complete documentation about the DYNAMIC suboption, see
“(DYNAMIC)” on page 193.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output
file.

See For complete documentation about the NO_BOTTOM_MATTER
suboption, see “(NO_BOTTOM_MATTER)” on page 193.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top
of the output file. For HTML 4.0, the NO_TOP_MATTER option removes
the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption,
see “(NO_TOP_MATTER)” on page 194.

(TITLE=’title-text’)
inserts into the metadata of a file the text string that you specify as the text to
appear in the browser window title bar.

title-text
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see
“(TITLE=’title-text’)” on page 194.

(URL= ’Uniform-Resource-Locator’)
specifies a URL for the file-specification. ODS uses this URL (instead of the
filename) in all the links and references that it creates and that point to the
file.

See For complete documentation about the URL= suboption, see “(URL=
’Uniform-Resource-Locator’)” on page 195.

Interaction The SAS system option PAGESIZE= has no effect on pages in HTML
output except when you are creating batch output. For information
about the PAGESIZE= option, see “PAGESIZE= System Option” in
SAS System Options: Reference.

PARAMETERS= (parameter-pair-1 ... parameter-pair-n)
writes the specified parameters between the tags that generate dynamic graphics
output.

parameter-pair
specifies the name and value of each parameter. parameter-pair has the following
form:

’parameter-name’= ’parameter-value’
**parameter-name**

is the name of the parameter.

**parameter-value**

is the value of the parameter.

**Requirement**

You must enclose **parameter-name** and **parameter-value** in quotation marks.

**Interaction**

Use PARAMETERS= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

**See**

*SAS/GRAPH: Reference* for valid parameters for the Graph Applet, Map Applet, Contour Applet, and the MetaView Applet.

**PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)**

specifies the location of an aggregate storage location or a SAS catalog for all markup files. If the GPATH= option is not specified, all graphics output files are written to the "aggregate-file-storage-specification" or libref.

'*aggregate-file-storage-location'*

specifies an aggregate storage location such as directory, folder, or partitioned data set.

**Requirement**

You must enclose **aggregate-file-storage-location** in quotation marks.

**fileref**

is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

**Interaction**

If you use a fileref in the PATH= option, then ODS does not use information from PATH= when it constructs links.

**See**

For information about the FILENAME statement, see “FILENAME Statement” in *SAS Statements: Reference*.

**libref.catalo**

g specifies a SAS catalog to write to.

**See**

For information about the LIBNAME statement, see “LIBNAME Statement” in *SAS Statements: Reference*.

**URL= 'Uniform-Resource-Locator' | NONE**

specifies a URL for the file-specification.

**Uniform-Resource-Locator**

is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

**NONE**

specifies that no information from the PATH= option appears in the links or references.

**Tip**

This option is useful for building output files that can be moved from one location to another. The links from the contents and page files must be
constructed with a single-name URL, and the contents, page, and body files must be in the same location.

Interaction  
If you use the BODY= or FILE= external file option in conjunction with the PATH= option, the external file specification should not include path information.

RECORD_SEPARATOR= 'alternative-separator' | NONE  
specifies an alternative character or string that separates lines in the output files.

Different operating environments use different separator characters. If you do not specify a record separator, then the files are formatted for the environment where you run the SAS program. However, if you are generating files for viewing in a different operating environment that uses a different separator character, then you can specify a record separator that is appropriate for the target environment.

alternative-separator  
represents one or more characters in hexadecimal or ASCII format. For example, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= '0D0A'x  

Operating Environment Information  
In a mainframe environment, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= '0D25'x  

Requirement  You must enclose alternative-separator in quotation marks.

NONE  
produces the markup language that is appropriate for the environment where you run the SAS job.

Windows Specifics  
In a mainframe environment, by default, ODS produces a binary file that contains embedded record separator characters. This binary file is not restricted by the line-length restrictions on ASCII files. However, if you view the binary files in a text editor, then the lines run together. If you want to format the files so that you can read them with a text editor, then use RECORD_SEPARATOR= NONE. In this case, ODS writes one line of markup language at a time to the file. When you use a value of NONE, the logical record length of the file that you are writing to must be at least as long as the longest line that ODS produces. If the logical record length of the file is not long enough, then the markup language might wrap to another line at an inappropriate place.

Aliases  RECSEP=
          RS=

STYLE= style-template  
specifies the style template to use in writing the output files.

style-template  
describes how to display the presentation aspects (color, font face, font size, and so on) of your SAS output. A style template determines the overall appearance of the documents that use it. Each style template consists of style elements.
Interaction  The STYLE= option is not valid when you are creating XML output.

Note  If you are using SAS Studio, you do not need to specify the STYLE= option. You can go to Preferences ⇒ Results and change the style from the drop-down list for your selected destination.

See  For a complete discussion of style templates, see “TEMPLATE Procedure: Creating a Style Template” in SAS Output Delivery System: Procedures Guide.

Default  If you do not specify a style template, then ODS uses the file that is specified in the SAS registry subkey ODS ⇒ DESTINATIONS ⇒ MARKUP. By default, this value specifies Default.

Interaction  If you specify the STYLE= option in an ODS HTML4 statement and subsequently need PROC PRINT output to use new style templates in another ODS HTML4 statement, close the first statement before specifying the second statement.

STYLESHEET= ’file-specification’ <(suboption(s))>

opens a markup family destination and places the style information for markup output into an external file, or reads style sheet information from an existing file. These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification  specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file  is the name of an external output file.

Requirement  You must enclose external-file in quotation marks.

fileref  is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See  For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

entry.markup  specifies an entry in a SAS catalog to write to.

Interaction  If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)  specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 193.

(specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 193.

(specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 194.

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE=’title-text’)” on page 194.

specifies a URL for the file specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= ’Uniform-Resource-Locator’ )” on page 195.

Note By default, if you do not specifically send the information to a separate file, then the style sheet information is included in the specified HTML file.

Example “Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

inserts text into your document by triggering the paragraph event and specifying a text string to be assigned to the VALUE event variable.

By default the TEXT= option is used in a paragraph event.

You can specify a text-string for a specific event by using the TEXT= option with the EVENT= option by using the following syntax:

EVENT= event-name (TEXT= text-string)
See For information about events and event variables, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in SAS Output Delivery System: Procedures Guide.

Example “Example: Conditionally Excluding Output Objects and Sending Them to Different Output Destinations” on page 325

TRANTAB= 'translation-table'
specifies the translation table to use when transcoding a file for output.


Suboptions

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file. This option sets the value of the CONTENTTYPE= style attribute. For more information, see CONTENTTYPE= on page 1015 in PROC TEMPLATE.

Default If you do not specify DYNAMIC, then ODS sets the value of HTMLCONTENTTYPE= for writing to a file.

Restriction If you specify the DYNAMIC suboption with one of the following options in the ODS HTML statement, then you must specify it for all of these options in that statement.
- BODY=
- CONTENTS=
- PAGE=
- FRAME=
- STYLESHEET=
- TAGSET=

Requirements You must enclose DYNAMIC in parentheses.

You must specify DYNAMIC next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

Alias NOBOT

Requirements You must enclose NO_BOTTOM_MATTER in parentheses.

You must specify NO_BOTTOM_MATTER next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.
Interactions
The NO_BOTTOM_MATTER suboption, in conjunction with the NO_TOP_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

Tip
If you want to leave a body file in a state that you can append to with ODS, then use NO_BOTTOM_MATTER with the file-specification BODY= option in any markup language statement.

See
The NO_TOP_MATTER suboption

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

Alias
NOTOP

Requirements
You must enclose NO_TOP_MATTER in parentheses.

You must specify NO_TOP_MATTER next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

Interactions
The NO_TOP_MATTER suboption, in conjunction with the NO_BOTTOM_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

See
The NO_BOTTOM_MATTER suboption and the ANCHOR= option

(TITLE="title-text")
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

Title
is the text in the metadata of a file that indicates the title.

Requirements
You must enclose TITLE= in parentheses.

You must enclose title-text in quotation marks.

Tip
If you are creating a web page that uses frames, then it is the TITLE= specification for the frame file that appears in the browser window title bar.
Example 3: Creating Multiple Markup Output on page 522

(URL= ‘Uniform-Resource-Locator’)

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

Requirements

You must enclose URL= ‘Uniform-Resource-Locator’ in parentheses.

You must enclose Uniform-Resource-Locator in quotation marks.

You must specify URL= ‘Uniform-Resource-Locator’ next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

Tips

This option is useful for building HTML files that can be moved from one location to another. The links from the contents and page files must be constructed with a single name URL, and the contents, page, and body files must all be in the same location.

You never need to specify this suboption with the FRAME= option because ODS files do not reference the frame file.

Example

“Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

Details

The ODS CHTML statement is part of the ODS markup family of statements. ODS statements in the markup family produce output that is formatted using one of many different markup languages such as HTML (Hypertext Markup Language) and XML (Extensible Markup Language). You can specify a markup language that SAS supplies, or create one of your own and store it as a user-defined markup language.

ODS CSVALL Statement

Opens, manages, or closes the CSVALL destination, which produces CSVALL output containing columns of data values that are separated by commas, and produces tabular output with titles, notes, and BY lines.

Valid in: Anywhere
Category: ODS: Third-Party Formatted

Syntax

ODS CSVALL (<ID= identifier>) <action> ;

Summary of Optional Arguments

(ID= identifier)
Open multiple instances of the same destination at the same time
BODY= "file-specification" (suboption(s))
Open a markup family destination and specify the file that contains the primary output that is created by the ODS statement

**CLOSE**

Close the destination and the file that is associated with it

**DOM=** "external-file">

Specify that the ODS document object model is written to the SAS log or to an external file.

**ENCODING=** local-character-set-encoding

Override the encoding for input or output processing (transcodes) of external files

**EVENT=** event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )

Specify an event and the value for event variables that is associated with the event

**EXCLUDE** exclusion(s) | ALL | NONE

Exclude output objects from the destination

**GFOOTNOTE | NOGFOOTNOTE**

Control the location where footnotes are printed in the graphics output

**GTITLE | NOGTITLE**

Control the location where titles are printed in the graphics output

**NEWFILE=** starting-point

Create a new body file at the specified starting point

**OPTIONS ( DOC= | <suboption(s)> )**

Specify tagset-specific suboptions and a named value

**PACKAGE <(package-name)>**

Specify that the output from the destination be added to an ODS package

**PATH=** 'aggregate-file-storage-specification' | fileref | libref.catalog

Specify the location of an aggregate storage location or a SAS catalog for all markup files

**RECORD_SEPARATOR=** 'alternative-separator' | NONE

Specify an alternative character or string to separate lines in the output files

**SELECT** selection(s) | ALL | NONE

Select output objects for the destination

**SHOW**

Write to the SAS log the current selection or exclusion list for the destination

**TRANTAB=** 'translation-table'

Specify a translation table to use when transcoding a file for output

---

**Without Arguments**

If you use the ODS CSVALL statement without an action or options, then it opens the CSVALL destination and creates CSVALL output.

**Actions**

The following actions are available for the ODS CSVALL statement:

**CLOSE**

closes the destination and any files that are associated with it.

**Tip** When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.
EXCLUDE exclusion(s) | ALL | NONE

excludes one or more output objects from the destination.

Default     NONE
Restriction A destination must be open for this action to take effect.
See         “ODS EXCLUDE Statement” on page 321

SELECT selection(s) | ALL | NONE

selects output objects for the specified destination.

Default     ALL
Restriction A destination must be open for this action to take effect.
See         “ODS SELECT Statement” on page 758

SHOW

writes the current selection list or exclusion list for the destination to the SAS log.

Restriction The destination must be open for this action to take effect.
Tip         If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.
See         “ODS SHOW Statement” on page 771

Optional Arguments
The following options are available for the ODS CSVALL statement.

BODY= "file-specification" (suboption(s))

opens a markup family destination and specifies the file that contains the primary output that is created by the ODS statement. These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.
Restriction The BODY=fileref option cannot be used in conjunction with the NEWFILE= option.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

entry.markup specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

Alias FILE=

Interaction Using the BODY= option in an ODS markup family statement that refers to an open ODS markup destination forces ODS to close the destination and all associated files. ODS then opens a new instance of the destination. For more information see “Opening and Closing the MARKUP Destination” on page 517.

Note For some values of TAGSET=, this output is an HTML file. For other TAGSET= values, the output is an XML file, and so on.

DOM="external-file">

specifies that the ODS document object model is written to the SAS log or an external file.

external-file is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

See For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in SAS Output Delivery System: Advanced Topics.

ENCODING= local-character-set-encoding overrides the encoding for input or output processing (transcodes) of external files.

See For information about the ENCODING= option, see SAS National Language Support (NLS): Reference Guide.

EVENT=event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )

specifies an event and the value for event variables that are associated with the event.

(FILE= BODY | CODE | CONTENTS | DATA | FRAME | PAGES | STYLESHEET);

triggers one of the known types of output files that correspond to the BODY=, CODE=, CONTENTS=, FRAME=, PAGES=, and STYLESHEET= options.

(FINISH)

triggers the finish section of an event.

See For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.
(LABEL='variable-value')
specifies the value for the LABEL event variable.

Requirement  

variable-value must be enclosed in quotation marks.

See  

For information about the LABEL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(NAME='variable-value')
specifies the value for the NAME event variable.

Requirement  

variable-value must be enclosed in quotation marks.

See  

For information about the NAME event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(START)
triggers the start section of an event.

See  

For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(STYLE=style-element)
specifies a style element.

See  

For information about style elements, see “Style Attributes Overview” in SAS Output Delivery System: Procedures Guide.

(TARGET='variable-value')
specifies the value for the TARGET event variable.

Requirement  

variable-value must be enclosed in quotation marks.

See  

For information about the TARGET event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(TEXT='variable-value')
specifies the value for the TEXT event variable.

Requirement  

variable-value must be enclosed in quotation marks.

See  

For information about the TEXT event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(URL='variable-value')
specifies the value for the URL event variable.

Requirement  

variable-value must be enclosed in quotation marks.

See  

For information about the URL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

Default  

(FILE='BODY')

 Requirement  

The EVENT= option's suboptions must be enclosed in parentheses.

GFOOTNOTE | NOGFOOTNOTE
controls the location where footnotes are printed in the graphics output.
GFOOTNOTE
writes footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

NOGFOOTNOTE
writes footnotes that are created by ODS, which appear outside the graph borders.

Default GFOOTNOTE

Restrictions Footnotes that are displayed by a markup language statement support all SAS/GRAPH FOOTNOTE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH FOOTNOTE statement, see “FOOTNOTE Statement” in SAS/GRAPH: Reference.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

GTITLE | NOGTITLE
controls the location where titles are printed in the graphics output.

GTITLE
writes the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.

NOGTITLE
writes the title that is created by ODS, which appears outside of the graph borders.

Default GTITLE

Restrictions Titles that are displayed by any markup language statement support most SAS/GRAPH TITLE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH TITLE statement, see TITLE statement.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

(ID= identifier)
enables you to run multiple instances of the same destination at the same time. Each instance can have different options.

identifier specifies another instance of the destination that is already open. identifier is numeric or a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numeric characters.

Restriction If identifier is numeric, it must be a positive integer.

Requirement You must specify the ID= option immediately after the destination name.
Tip  You can omit the ID= option and instead use a name or a number to identify the instance.

Example  “Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594

**NEWFILE= starting-point**  
creates a new body file at the specified *starting-point*.

*starting-point*  
is the location in the output where you want to create a new body file.

ODS automatically names new files by incrementing the name of the body file.  
In the following example, ODS names the first body file **REPORT.XML**.  
Additional body files are named **REPORT1.XML**, **REPORT2.XML**, and so on.

Example:

```plaintext
BODY= 'REPORT.XML'
```

*starting-point* is one of the following:

- **BYGROUP**  
  starts a new file for the results of each BY group.

- **NONE**  
  writes all output to the body file that is currently open.

- **OUTPUT**  
  starts a new body file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

  **Alias**  
  **TABLE**

- **PAGE**  
  starts a new body file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.

- **PROC**  
  starts a new body file each time you start a new procedure.

**Default**  
**NONE**

**Restriction**  
The NEWFILE= option cannot be used in conjunction with the BODY=fileref option.

**Tips**  
If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first body file **MAY5.XML**. Additional body files are named **MAY6.XML**, **MAY7.XML**, and so on.

Example:

```plaintext
BODY= 'MAY5.XML'
```

**OPTIONS ( DOC= | <suboption(s)> )**  
specifies tagset-specific suboptions and a named value.

- **(DOC= 'HELP' | 'QUICK' | 'SETTINGS' | 'CHANGELOG')**  
  provides information about the specified tagset.
HELP
provides generic help and information with a quick reference.

QUICK
describes the options available for this tagset.

SETTINGS
provides the current option settings.

CHANGELOG
lists a history of changes made to the tagset. This suboption is supported only on the RTF tagset.

Requirement
All values must be enclosed in quotation marks.

suboption(s)
specifies one or more suboptions that are valid for the specified tagset. Suboptions have the following format:

keyword='value'
Specify one of the following options when opening an ODS tagset statement, or at any time after the destination has been opened, to get information about suboptions for the tagset.

• options(doc='help');
• options(doc='quick');
• options(doc='settings');

Requirement
suboption(s) must be enclosed in parentheses.

Example
“Example: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information ” on page 802

PACKAGE <(package-name)>
specifies that the output from the destination be added to a package.

ods csvall package;

package-name
specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

ods csvall package(a);

See
“ODS PACKAGE Statement ” on page 554

Example
“Example 1: Creating an ODS Package” on page 558

PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog
specifies the location of an aggregate storage location or a SAS catalog for all markup files. If the GPATH= option is not specified, all graphics output files are written to the "aggregate-file-storage-specification" or libref.

'aggregate-file-storage-location'
specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement
You must enclose aggregate-file-storage-location in quotation marks.
fileref
is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction If you use a fileref in the PATH= option, then ODS does not use information from PATH= when it constructs links.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

libref.catalog
specifies a SAS catalog to write to.

See For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Statements: Reference.

Interaction If you use the BODY= or FILE= external file option in conjunction with the PATH= option, the external file specification should not include path information.

RECORD_SEPARATOR= 'alternative-separator' | NONE
specifies an alternative character or string that separates lines in the output files.

Different operating environments use different separator characters. If you do not specify a record separator, then the files are formatted for the environment where you run the SAS program. However, if you are generating files for viewing in a different operating environment that uses a different separator character, then you can specify a record separator that is appropriate for the target environment.

alternative-separator
represents one or more characters in hexadecimal or ASCII format. For example, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= '0D0A'x

Operating Environment Information
In a mainframe environment, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= '0D25'x

Requirement You must enclose alternative-separator in quotation marks.

NONE
produces the markup language that is appropriate for the environment where you run the SAS job.

Windows Specifics
In a mainframe environment, by default, ODS produces a binary file that contains embedded record separator characters. This binary file is not restricted by the line-length restrictions on ASCII files. However, if you view the binary files in a text editor, then the lines run together. If you want to format the files so that you can read them with a text editor, then use RECORD_SEPARATOR= NONE. In this case, ODS writes one line of markup language at a time to the file. When you use a value of NONE, the logical record length of the file that you are writing to must be at least as long as the longest line that ODS produces. If the logical record length of the file
is not long enough, then the markup language might wrap to another line at an inappropriate place.

<table>
<thead>
<tr>
<th>Aliases</th>
<th>RECESSP=</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RS=</td>
</tr>
</tbody>
</table>

**TRANTAB= 'translation-table'**
specifies the translation table to use when transcoding a file for output.

**See** For information about the TRANTAB= option, see “TRANTAB= System Option” in SAS National Language Support (NLS): Reference Guide.

### Details

The ODS CSVALL statement is part of the ODS markup family of statements. ODS statements in the markup family open the markup destination and produce output that is formatted using one of many different markup languages. Among the supported markup languages are HTML (Hypertext Markup Language) and XML (Extensible Markup Language). You can specify a markup language that SAS supplies, or create one of your own and store it as a user-defined markup language.

### Example: Creating Multiple Markup Output

#### Features:
- ODS LISTING statement action:
  - CLOSE
- ODS CSVALL statement option:
  - BODY=
- ODS MARKUP statement options:
  - BODY=
  - TAGSET=
  - TITLE=

#### Other features:
- OPTIONS statement
- PROC PRINT
- TITLE statement

#### Data set:
- Grain_Production

#### Details

The following ODS example creates two different types of markup output from the same procedure output. To create two markup outputs requires two ODS destinations. Because ODS MARKUP is considered one destination, you cannot specify two tagsets without the use of the ID= option. However, you can specify one output using ODS MARKUP. You can then specify the other output using ODS syntax in which the tagset is the destination.
Program

ods html close;
options obs=15;
ods csvall body='procprintcsvall.csv';
ods markup tagset=chtml body='procprintchtml.html'
   {title='This Text Identifies Your Content.'};
title 'Leading Grain-Producing Countries';
proc print data=grain_production;
run;
ods csvall close;
ods markup tagset=chtml close;

Program Description

Close the HTML destination so that no HTML output is produced. The HTML destination is open by default. The ODS HTML statement closes the HTML destination to conserve resources. The OPTIONS statement specifies that only fifteen observations be used.

ods html close;
options obs=15;

Create tabular output. The ODS CSV ALL statement produces tabular output with titles that contain columns of data values that are separated by commas.

ods csvall body='procprintcsvall.csv';

Create CHTML output. The ODS MARKUP TAGSET=CHTML statement produces compact, minimal HTML output that does not use style information, and a hierarchical table of contents. The TITLE= option specifies the text that appears in the browser window title bar.

ods markup tagset=chtml body='procprintchtml.html'
   {title='This Text Identifies Your Content.'};

Print the data set. The PRINT procedure prints the data set Grain_Production. The TITLE statement specifies the title.

title 'Leading Grain-Producing Countries';
proc print data=grain_production;
run;

Close the open destinations so that you can view or print the output. The ODS CSV ALL CLOSE statement closes the CSV ALL destination and all of the files that are associated with it. The ODS MARKUP TAGSET=CHTML CLOSE statement closes the MARKUP destination and all of the files that are associated with it. You must close the destinations before you can view the output with a browser or before you can send the output to a physical printer.

ods csvall close;
ods markup tagset=chtml close;
Output

The following output was created by specifying the MARKUP TAGSET=CHTML statement. The text "This Text Identifies Your Content." was specified by the TITLE= option.

Output 6.9  CHTML Output

Leading Grain-Producing Countries

<table>
<thead>
<tr>
<th>Obs</th>
<th>Country</th>
<th>Type</th>
<th>Year</th>
<th>Kilotons</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BRZ</td>
<td>Wheat</td>
<td>1995</td>
<td>1516</td>
</tr>
<tr>
<td>2</td>
<td>BRZ</td>
<td>Rice</td>
<td>1995</td>
<td>11236</td>
</tr>
<tr>
<td>3</td>
<td>BRZ</td>
<td>Corn</td>
<td>1995</td>
<td>36276</td>
</tr>
<tr>
<td>4</td>
<td>CHN</td>
<td>Wheat</td>
<td>1995</td>
<td>102207</td>
</tr>
<tr>
<td>5</td>
<td>CHN</td>
<td>Rice</td>
<td>1995</td>
<td>185226</td>
</tr>
<tr>
<td>6</td>
<td>CHN</td>
<td>Corn</td>
<td>1995</td>
<td>112331</td>
</tr>
<tr>
<td>7</td>
<td>IND</td>
<td>Wheat</td>
<td>1995</td>
<td>63007</td>
</tr>
<tr>
<td>8</td>
<td>IND</td>
<td>Rice</td>
<td>1995</td>
<td>122372</td>
</tr>
<tr>
<td>9</td>
<td>IND</td>
<td>Corn</td>
<td>1995</td>
<td>9800</td>
</tr>
<tr>
<td>10</td>
<td>LNS</td>
<td>Wheat</td>
<td>1995</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>LNS</td>
<td>Rice</td>
<td>1995</td>
<td>49860</td>
</tr>
<tr>
<td>12</td>
<td>LNS</td>
<td>Corn</td>
<td>1995</td>
<td>8223</td>
</tr>
<tr>
<td>13</td>
<td>USA</td>
<td>Wheat</td>
<td>1995</td>
<td>59494</td>
</tr>
<tr>
<td>14</td>
<td>USA</td>
<td>Rice</td>
<td>1995</td>
<td>7588</td>
</tr>
<tr>
<td>15</td>
<td>USA</td>
<td>Corn</td>
<td>1995</td>
<td>187500</td>
</tr>
</tbody>
</table>
The following output was created by specifying the ODS CSVALL statement.

**Output 6.10** CSVALL Output Viewed in Microsoft Excel

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>----</td>
</tr>
<tr>
<td>1</td>
<td>Leading Grain-Producing Countries</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Obs</td>
<td>Country</td>
<td>Type</td>
<td>Year</td>
<td>Kilotons</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>BRZ</td>
<td>Wheat</td>
<td>1995</td>
<td>1516</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>BRZ</td>
<td>Rice</td>
<td>1995</td>
<td>11236</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>BRZ</td>
<td>Corn</td>
<td>1995</td>
<td>36278</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>CHN</td>
<td>Wheat</td>
<td>1995</td>
<td>102207</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>CHN</td>
<td>Rice</td>
<td>1995</td>
<td>185226</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>CHN</td>
<td>Corn</td>
<td>1995</td>
<td>112331</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>IND</td>
<td>Wheat</td>
<td>1995</td>
<td>63007</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>IND</td>
<td>Rice</td>
<td>1995</td>
<td>122372</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>IND</td>
<td>Corn</td>
<td>1995</td>
<td>9800</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>INS</td>
<td>Wheat</td>
<td>1995</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>INS</td>
<td>Rice</td>
<td>1995</td>
<td>492860</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>12</td>
<td>INS</td>
<td>Corn</td>
<td>1995</td>
<td>8223</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>13</td>
<td>USA</td>
<td>Wheat</td>
<td>1995</td>
<td>59494</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>14</td>
<td>USA</td>
<td>Rice</td>
<td>1995</td>
<td>7888</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>15</td>
<td>USA</td>
<td>Corn</td>
<td>1995</td>
<td>187300</td>
<td></td>
</tr>
</tbody>
</table>

### ODS DECIMAL_ALIGN Statement

Controls the justification of numeric columns when no justification is specified.

- **Valid in:** Anywhere
- **Category:** ODS: SAS Formatted
- **Default:** ODS NO_DECIMAL_ALIGN
- **Interaction:** The ODS DECIMAL_ALIGN statement only affects the RTF destination and the printer family of destinations.

**See:** “Values in Table Columns and How They Are Justified” in SAS Output Delivery System: Procedures Guide

### Syntax

```
ODS DECIMAL_ALIGN;
ODS NO_DECIMAL_ALIGN;
```

### Required Arguments

**ODS DECIMAL_ALIGN**

aligns values by the decimal point in numeric columns when no justification is specified.

**Alias**

`ODS DECIMAL_ALIGN=YES`
ODS NO_DECIMAL_ALIGN
right-justifies numeric columns when no justification is specified.

Alias ODS DECIMAL_ALIGN=NO

Details
The ODS DECIMAL_ALIGN statement has no effect on any column that is assigned a justification from a procedure or column definition.

ODS DOCUMENT Statement
Opens, manages, or closes the DOCUMENT destination, which produces a hierarchy of output objects that enables you to produce multiple ODS output formats without rerunning a PROC or DATA step.

Valid in: Anywhere
Category: ODS: Output Control
Interaction: 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 running a report. You can then use the DOCUMENT procedure to rearrange, duplicate, or remove output from the results of a procedure or a database query without invoking the procedures from the original report. For complete documentation about the DOCUMENT procedure, see “The DOCUMENT Procedure” in SAS Output Delivery System: Procedures Guide.

Syntax
ODS DOCUMENT action;

Actions
The following actions are available for the ODS DOCUMENT statement:

CLOSE
  closes the destination and any files that are associated with it.

Tip When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination frees some system resources.

EXCLUDE exclusion(s)| ALL | NONE
  excludes one or more output objects from the DOCUMENT destination.

Default NONE

Restriction The DOCUMENT destination must be open for this action to take effect.
SELECT selection(s) | ALL | NONE
selects one or more output objects for the DOCUMENT destination.

Default ALL

Restriction The DOCUMENT destination must be open for this action to take effect.

See “ODS SELECT Statement ” on page 758

SHOW
writes the current selection or exclusion list for the destination to the SAS log.

Restriction The destination must be open for this action to take effect.

Tip If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list.

See “ODS SHOW Statement” on page 771

Optional Arguments

CATALOG=permanent-catalog | _NULL_

CAUTION: If you do not specify a value (other than _NULL_) for this option, then you can replay temporary GRSEGs only during the session in which they are created, not in subsequent sessions.

permanent-catalog copies any temporary GRSEG to the specified permanent catalog and keeps a reference to the permanent GRSEG in the document. This value persists until the ODS DOCUMENT statement is closed, or until you delete it by specifying CATALOG=_NULL_.

The permanent catalog has the following form:

<libref.> < member-name> ;

_NULL_ deletes the catalog name that was previously specified for the CATALOG= option. Thereafter, temporary GRSEGs are not copied into the permanent catalog and thus are unavailable in subsequent sessions.

Alias CAT=

Default By default, no value is assigned to CATALOG=, which means that temporary GRSEGs are not copied to a permanent catalog.

DIR=(<PATH=path<(access-option)>>()<LABEL='label'>);

specifies the directory path and/or label for ODS output.

LABEL=label
assigns a label to a path.

Requirement The label that you assign must be enclosed in quotation marks.
If LABEL= is used with the PATH= option, then the label applies to the path. If LABEL= is used without the PATH= option, then the label applies to the entire document.

**PATH=**<path><(access-option)> is specified as a sequence of entries that are delimited by backslashes.

*path* can have the following form:

*path*<#sequence-number>

*path* is the name of the path.

*#sequence-number* is a number which, when combined with a pathname, uniquely identifies the entry in the directory that contains it.

**Default** The default path is "\" (root).

**Tip** You can specify a directory that contains entries that do not exist in the document.

*access-option* specifies the access mode for the ODS document.

**WRITE** opens a document and provides Write access as well as Read access.

**Interaction** If a label is specified with the LABEL= option, then it overrides any existing label assigned to the document.

**Tip** If the ODS document does not exist, then it is created.

**CAUTION** If the ODS document already exists, then it is overwritten.

**UPDATE** opens an ODS document and appends new content to the document. UPDATE provides Update access as well as Read access.

**Interaction** If a label is specified with the LABEL= option, then it is assigned to the document.

**Tip** If the ODS document does not exist, then the document is created.

**CAUTION** If the document already exists, then its contents are not changed.

**Default** UPDATE

**Note** Procedure output or data queries are added at the end of the directory.

**NAME=**<libref.> member-name<(access-option)>

*libref* specifies the SAS library where the document is stored.
If no library name is specified, the Work library is used.

**member-name**

specifies the document name.

**Defaults**

If no NAME= is specified, the specified options apply to the currently open document.

If you do not specify an `access-option` with NAME=, then your directories will open in UPDATE mode.

**access-option**

specifies the access mode for the ODS document.

**WRITE**

opens a document and provides Write access as well as Read access.

**Interaction**

If a label is specified with the LABEL= option, then it overrides any existing label assigned to the document.

**Tip**

If the ODS document does not exist, then it is created.

**CAUTION**

If the ODS document already exists, then it is overwritten.

**UPDATE**

opens an ODS document and appends new content to the document. UPDATE provides Update access as well as Read access.

**Interaction**

If a label has been specified with the LABEL= option, then it is assigned to the document.

**Tip**

If the ODS document does not exist, then the document is created.

**CAUTION**

If the document already exists, then its contents is not changed.

**Default**

UPDATE

**Interaction**

When a DOCUMENT destination is open, using the NAME= option in an ODS DOCUMENT statement forces ODS to close the destination and all files associated with it, and to open a new instance of the destination.

---

**ODS EPUB Statement**

Opens, manages, or closes the ODS EPUB destination, which generates EPUB e-books.

**Valid in:** Anywhere

**Category:** ODS: Third-Party Formatted

**Defaults:** Through the second maintenance release of SAS 9.4, the default EPUB version is EPUB2. In the third maintenance release of SAS 9.4, ODS EPUB3 is the default EPUB destination. These defaults are set in the SAS Registry. To change your
default EPUB version in the Registry, see “Changing ODS EPUB Destination Default Value” on page 48.

The default style for EPUB destinations is Daisy.

The default name for EPUB files is “sasepub.epub”.

Interaction: By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC CONTENTS), ODS formats the output in SAS Monospace font. If you are creating output that is viewed in an operating environment where SAS software is not installed, this output is not displayed correctly. This is because without SAS, the SAS Monospace font is not recognized. To make your document display correctly, include the following statement before your SAS program:

```sas
OPTIONS FORMCHAR="|----|+|---+=|-/<>*";
```

ODS destination: The version of EPUB generated by the ODS EPUB destination is specified in the SAS registry.

z/OS specifics: On z/OS, the ODS destination for EPUB works only with the Hierarchical File System (HFS). You must use the FILESYSTEM=HFS option. The external file specified by the FILE= option must be an HFS file. If the WORK= option is used, the directory must be an HFS directory.

Notes: Encoding is set in the SAS Registry for ODS EPUB destinations. See “Changing the ODS EPUB and EPUB3 Encoding Settings” on page 49.

If you want to customize your markup output, continue to use ODS MARKUP for that purpose. The EPUB destination does not permit user customization beyond sending surfaced EVENT= events.

iBooks on iPad is the preferred e-book reader. Results can vary using other e-book readers.

Examples:

“Example 1: Create an ODS EPUB2 Book with a Title and Creator” on page 225

“Example 2: Create an ODS EPUB E-book with Inline and Nonlinear Tables” on page 229

“Example 3: ODS EPUB Book with a Customized TOC” on page 231

Syntax

```sas
ODS EPUB (<ID=> identifier>) < action > ;
ODS EPUB (<ID=> identifier>) < option(s)> ;
```

Without Arguments

The ODS EPUB statement is used in a generic sense to refer to either the EPUB2 version or the EPUB3 version of EPUB. The version that is used is determined by the version that is set in the Registry. Therefore, you can specify the ODS EPUB statement where the EPUB version that is used is specified in the Registry or you can explicitly specify the ODS EPUB2 statement or the ODS EPUB3 statement.

To change the default EPUB version in the Registry. See “Changing ODS EPUB Destination Default Value” on page 48.

Depending on which EPUB version you are using, refer to the ODS EPUB2 statement or the ODS EPUB3 statement for the list of options that are available for each destination. See “ODS EPUB2 Statement” on page 213 or “ODS EPUB3 Statement” on page 233.
Details

Overview of ODS EPUB
The ODS EPUB statement generates output that has the .epub extension. EPUB (electronic publication) is a free and open e-book standard produced by the International Digital Publishing Forum (IDPF). EPUB is designed for reflowable content, meaning that an EPUB reader can optimize text for a particular display device. E-books that use the .epub extension can be read by a wide variety of e-book readers, from dedicated hardware to desktop software to online based readers.

The .epub file extension consists of XML files for reflowable digital books and publications. The .epub file is a ZIP archive containing the books files, either XHTML or DTBook, a number of XML description and navigation files, and possibly image or media files.

The version of EPUB generated by the ODS EPUB destination is specified in the SAS registry. In the SAS Registry, click on ODS to find the default EPUB version. See “Changing ODS EPUB Destination Default Value” on page 48.

Accessibility of EPUB Output
The ODS EPUB destinations (ODS EPUB2 and ODS EPUB3) are the recommended destinations for creating SAS output that is accessible to the broadest audience. They create e-books that use many of the accessibility features of the EPUB specification. These features allow e-book readers such as iBooks to present e-books so that they adapt to the needs of users with disabilities. For example, when reading an e-book created by ODS EPUB destinations using iBooks on an iPad, users can adjust font size, color schemes, and magnification. They can also access the text using assistive technologies such as the Voiceover screen reader and refreshable braille displays.

ODS EPUB2 Statement
Opens, manages, or closes the EPUB2 destination, which generates EPUB e-books using the EPUB2 standard.

Valid in: Anywhere
Category: ODS: Third-Party Formatted
Defaults: Through the second maintenance release of SAS 9.4, the default EPUB version is EPUB2. In the third maintenance release of SAS 9.4, ODS EPUB3 is the default EPUB destination and is set in the Registry. To change your default EPUB version in the Registry, see “Changing ODS EPUB Destination Default Value” on page 48.

The default style for EPUB destinations is Daisy.
The default name for EPUB files is “sasepub.epub”.

Interaction: By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC CONTENTS), ODS formats the output in SAS Monospace font. If you are creating output that is viewed in an operating environment where SAS software is not installed, this output is not displayed correctly. This is because without SAS, the SAS Monospace font is not recognized. To make your document display correctly, include the following statement before your SAS program:

```
OPTIONS FORMCHAR="|----|+|---+=|-/>\<>**;`
```

ODS destination: The version of EPUB generated by the ODS EPUB destination is specified in the SAS registry. The version of EPUB generated by the ODS EPUB2 destination is EPUB2.
z/OS specifics: On z/OS, the ODS destination for EPUB works only with the Hierarchical File System (HFS). You must use the FILESYSTEM=HFS option. The external file specified by the FILE= option must be an HFS file. If the WORK= option is used, the directory must be an HFS directory.

Notes: Encoding is set in the SAS Registry for ODS EPUB destinations. See “Changing the ODS EPUB and EPUB3 Encoding Settings” on page 49.

If you want to customize your markup output, continue to use ODS MARKUP for that purpose. The EPUB destination does not permit user customization beyond sending surfaced EVENT= events.

iBooks on iPad is the preferred e-book reader. Results can vary using other e-book readers.

Examples: “Example 1: Create an ODS EPUB2 Book with a Title and Creator” on page 225
“Example 2: Create an ODS EPUB E-book with Inline and Nonlinear Tables” on page 229
“Example 3: ODS EPUB Book with a Customized TOC” on page 231

Syntax

ODS EPUB2 <(<ID=> identifier)> < action> ;
ODS EPUB2 <(<ID=> identifier)> <option(s)> ;

Summary of Optional Arguments

(ID= identifier)
Open multiple instances of the same destination at the same time

ANCHOR= 'anchor-name'
Specify a unique base name for the anchor tag that identifies each output object in the current chapter

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
Specify how to measure the width of cells. Use to override the default value of BOX_SIZING for a destination

CLOSE
Close the destination and the file that is associated with it

CSSSTYLE= 'file-specification'(media-type-1<…media-type-10>)
Specify a cascading style sheet to apply to your output

DEVICE= device-driver
Specify the name of a device driver.

DOM=<"external-file">
Specify that the ODS document object model is written to the SAS log or to an external file.

EVENT= event-name (FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )
Specify an event and the value for event variables that is associated with the event

EXCLUDE exclusion(s) | ALL | NONE
Exclude output objects from the destination

FILE='file-specification'
Specify the file that contains the e-book created by the destination

GFOOTNOTE | NOGFOOTNOTE
Control the location where footnotes are printed in the graphics output
**GTITLE | NOGTITLE**
Control the location where titles are printed in the graphics output

**IMAGE DPI**
Specify the image resolution for the graphical output

**NEWCHAPTER=starting-point**
Create a new chapter at the specified starting point

**OPTIONS (<suboption(s)>)**
Specify destination-specific suboptions and a space-delimited named value for how e-books are handled

**SELECT selection(s) | ALL | NONE**
Select output objects for the destination

**SHOW**
Write to the SAS log the current selection or exclusion list for the destination

**STYLE= style-override(s)**
Specifies one or more style-override(s) to use when writing output files

**STYLESHEET= ( URL= ‘external-file(s)’ )**
Specify one or more style sheet files to apply to the output

**TITLE=’text-string’**
Insert the title into the metadata of the e-book

**WORK=’directory-name’**
Specify a work directory for constructing the e-book

---

**Without Arguments**
If you use the ODS EPUB statement without an action or options, then it opens the EPUB destination and creates EPUB output based on the version (EPUB2 or EPUB3) that is specified in the Registry.

**Actions**
The following actions are available for the ODS EPUB2 statement.

**CLOSE**
closes the destination and any files that are associated with it.

**Tip** When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

**EXCLUDE exclusion(s) | ALL | NONE**
excludes one or more output objects from the destination.

<table>
<thead>
<tr>
<th>Default</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Restriction</strong></td>
<td>A destination must be open for this action to take effect.</td>
</tr>
<tr>
<td><strong>See</strong></td>
<td>“ODS EXCLUDE Statement” on page 321</td>
</tr>
</tbody>
</table>

**SELECT selection(s) | ALL | NONE**
selects output objects for the specified destination.

<table>
<thead>
<tr>
<th>Default</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Restriction</strong></td>
<td>A destination must be open for this action to take effect.</td>
</tr>
<tr>
<td><strong>See</strong></td>
<td>“ODS SELECT Statement ” on page 758</td>
</tr>
</tbody>
</table>
SHOW
writes the current selection list or exclusion list for the destination to the SAS log.

Restriction
The destination must be open for this action to take effect.

Tip
If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

See
“ODS SHOW Statement” on page 771

Optional Arguments

ANCHOR= 'anchor-name'
specifies a unique base name for the anchor tag that identifies each output object in the current body file.

Each output object has an anchor tag. The links and references are automatically created by ODS. These links and references point to the name of an anchor. Therefore, each anchor name in a file must be unique.

anchor-name
is the base name for the anchor tag that identifies each output object in the current body file.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR= 'TABULATE', then ODS names the first anchor tabulate. The second anchor is named tabulate1; the third is named tabulate2, and so on.

Restrictions
Each anchor name in a file must be unique.

Requirement
You must enclose anchor-name in quotation marks.

Tip
You can change anchor names as often as you want by specifying the ANCHOR= option in an ODS EPUB statement anywhere in your program. After you have specified an anchor name, it remains in effect until you specify a new one.

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry. In the SAS Registry, expand the ODS \ DESTINATION \ SCRIPT folder to locate the box_sizing= default for ODS EPUB destinations. For information about using the SAS Registry, see “Changing SAS Registry Settings for ODS” on page 40.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing.

CSSSTYLE= 'file-specification'</(media-type-1<…media-type-10>)/>
specifies a cascading style sheet to apply to your output.
file-specification
specifies a file, fileref, or URL that contains CSS code.

file-specification is one of the following:

"external-file"
is the name of the external file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

"URL"
is a URL to an external file.

Requirement You must enclose URL in quotation marks.

(media-type-1<.. media-type-10>)
specifies one or more media blocks that correspond to the type of media that your output is rendered on. CSS uses media type blocks to specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not contained in any media blocks. By using the media-type suboption, in addition to the general CSS code, you can import the section of a CSS file intended only for a specific media type.

Default If no media-type is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

Range You can specify up to ten different media types.

Requirements You must enclose media-type in parentheses.

Tip If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

Restriction The CSSSTYLE= option does not affect SAS/GRAPH output.

Requirement CSS files must be written in the same type of CSS produced by the ODS HTML statement. Only class names are supported, with no IDs and no context-based selectors. To view the CSS code that ODS creates, you can do one of the following:

• specify the ODS TRACE DOM statement
• specify the DOM option
Interaction

If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

See

For complete documentation about ODS and Cascading Style Sheets, see *SAS Output Delivery System: Advanced Topics*.

Example

“Example 6: Applying a CSS File to ODS Output” on page 527

**DEVICE= device-driver**

specifies the name of a device driver. ODS automatically selects an optimal default device for each open output destination.

The default device value can be found in the SAS registry. In the SAS Registry, expand the `ODS DESTINATION SCRIPT` folder to locate the DEVICE= default for ODS EPUB destinations. For information about using the SAS Registry, see “Changing SAS Registry Settings for ODS” on page 40.

**Range**

ODS uses the device driver that is specified in the SAS registry.

**Restriction**

Valid devices are GIF, JPEG, PNG, and SVG.

**Tip**

Specifying a device with the DEVICE= option takes precedence over the SAS global option and the graphics option.

**See**


**DOM<="external-file">**

specifies that the ODS document object model is written to the SAS log or an external file.

**external-file**

is the name of an external output file.

**Requirement**

You must enclose `external-file` in quotation marks.

**See**

For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in *SAS Output Delivery System: Advanced Topics*.

**EVENT= event-name (FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )**

specifies an event and the value for event variables that are associated with the event.

(FINISH)

triggers the finish section of an event.

See

For information about events, see “Understanding Events” in *SAS Output Delivery System: Procedures Guide*.

(LABEL='variable-value')

specifies the value for the LABEL event variable.

**Requirement**

`variable-value` must be enclosed in quotation marks.
(NAME='variable-value')
specifies the value for the NAME event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the NAME event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(START)
triggers the start section of an event.

See For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(STYLE=style-element)
specifies a style element.

See For information about style elements, see “Style Attributes Overview” in SAS Output Delivery System: Procedures Guide.

(TARGET='variable-value')
specifies the value for the TARGET event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the TARGET event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(TEXT='variable-value')
specifies the value for the TEXT event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the TEXT event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(URL='variable-value')
specifies the value for the URL event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the URL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

Requirement The EVENT= option's suboptions must be enclosed in parentheses.

FILE='file-specification'
specifies the file that contains the e-book created by the destination.

'file-specification'
specifies the file and fileref to receive output.

file-specification is one of the following:

external-file
is the name of an external file to receive output.
Requirement  You must enclose external-file in quotation marks.

fileref

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

Default  ODS uses the filename that is specified in the SAS registry. The default name for EPUB files is sasepub.epub. For information on using the SAS Registry to change defaults, see “Changing SAS Registry Settings for ODS” on page 40.

GFOOTNOTE | NOGFOOTNOTE

controls the location where footnotes are printed in the graphics output.

GFOOTNOTE

prints footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

NOGFOOTNOTE

prints footnotes that are created by ODS, which appear outside the graph borders.

Default  GFOOTNOTE

Restrictions  Footnotes that are displayed support most SAS/GRAPH FOOTNOTE options. The font must be valid for the e-book reader. Options that ODS cannot handle, such as text angle specifications, are ignored.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

GTITLE | NOGTITLE

controls the location where titles are printed in the graphics output.

GTITLE

prints the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.

NOGTITLE

prints the title that is created by ODS, which appears outside of the graph borders.

Default  GTITLE

Restrictions  Titles that are displayed support most SAS/GRAPH TITLE options. The font must be valid for the e-book reader. Options that ODS cannot handle, such as text angle specifications, are ignored.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

(ID= identifier)

enables you to run multiple instances of the same destination at the same time. Each instance can have different options.
**identifier**
specifies another instance of the destination that is already open. *identifier* is numeric or a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numeric characters.

**Restriction**
If *identifier* is numeric, it must be a positive integer.

**Requirement**
You must specify the ID= option immediately after the ODS EPUB statement keywords.

**Tip**
You can omit the ID= option and instead use a name or a number to identify the instance.

**IMAGE_DPI**
specifies the image resolution for graphical output.

**Alias**
DPI=

**Default**
ODS uses the resolution specified in the SAS registry.

**NEWCHAPTER=starting-point**
creates a new chapter at the specified starting point.

**starting-point**
is the location in the e-book where you want to create a new chapter.

**starting-point** is one of the following:

- **BYGROUP**
  starts a new chapter for the results of each BY group.

- **NONE**
  writes all output to the current chapter.

- **NOW**
  starts a new chapter and writes all output to that new chapter.

- **OUTPUT**
  starts a new chapter for each output object.

  **Alias**
  TABLE

- **PAGE**
  starts a new chapter for each page of output. A page break occurs when you start a new procedure, or when a procedure explicitly starts a new page.

  **Default**
  NONE

- **PROC**
  starts a new chapter for each procedure.

**OPTIONS (<suboption(s)>)**
specifies destination-specific suboptions with space-delimited name='value' pairs.

**suboption(s)** are the following:

- **(CONTENTS = 'YES' | 'NO' | 'OFF' | 'ON')**
  generates a table of contents.

  **OFF**
  does not generate a table of contents.
**Alias NO**

---

**ON**

generates a table of contents.

**Alias YES**

---

**Default ON | YES**

**(CONTRIBUTOR='text-string')**

specifies one or more contributor’s names responsible for making contributions to the content of the e-book. Delimit multiple contributors with commas.

**(COVER_IMAGE='external-file')**

specifies a cover image for the e-book.

**Requirement** You must enclose `external-file` in quotation marks.

**(COVERAGE='text-string')**

specifies the extent or scope of the content of the e-book.

**(CREATOR='text-string')**

specifies one or more primary creators or authors of the e-book.

**Note** Delimit multiple creators with commas.

**(DEFEAT_IBOOKS_CACHING= 'YES' | 'NO' | 'OFF' | 'ON')**

defeats iBooks e-book reader’s caching of e-books that have the same title.

**OFF**

 caches iBooks of the same title.

**Alias NO**

---

**ON**

defeats iBooks caching.

**Alias YES**

---

**Default ON | YES**

**(DESCRIPTION='text-string')**

specifies a description of the content of the e-book.

**(EMBEDDED_FONTS='fonts')**

specifies a list of fonts to be embedded in the e-book.

**See** For information about using fonts, see “Using Fonts with Universal Printers and SAS/GRAPH Devices” in SAS Language Reference: Concepts.

**(ISBN='isbn')**


**Requirement** Every EPUB e-book must be uniquely identifiable.
specifies output to be written to the non-linear section of the e-book.

NONE
occurs within the linear flow of the e-book.

CHAPTER
causes the entire chapter to be non-linear.

BATCH
causes batch (preformatted) output to be non-linear.

TABLE
causes tables to be non-linear.

ALL
causes the entire chapter to be non-linear. This has the same effect as the CHAPTER value.

Default NONE

OPTIMIZE_FOR_IBOOKS= 'YES' | 'NO' | 'OFF' | 'ON')

OFF
overrides user-specified fonts.

Alias NO

ON
honors user-specified fonts.

Alias YES

Default ON | YES

Note iBooks on iPad is the preferred e-book reader. Results can vary using other e-book readers.

PAGEBREAK= 'YES' | 'NO' | 'OFF' | 'ON')
honors explicit page breaks. The default (AUTO) allows ODS and procedures to determine when page breaks occur.

OFF
does not honor explicit page breaks.

Alias NO

ON
honors explicit page breaks.

Alias YES

AUTO
allows ODS and procedures to determine when page breaks occur. Honors explicit page breaks.

Default AUTO
(PUBLISHER=’text-string’)  
specifies the publisher of the e-book.

(RELATION=’text-string’)  
specifies a reference to a related resource.

(RIGHTS=’text-string’)  
specifies information about rights held in and over the e-book.

(SOURCE=’text-string’)  
specifies information about a prior resource from which the e-book was derived.

(START=’chapter-number’)  
specifies the component (chapter) at which an e-book first opens.

Default 1

(SUBJECT=’text-string’)  
specifies one or more topics about the content of the e-book. Delimit multiple subjects with commas.

(TYPE=’text-string’)  
specifies the nature or genre of the content of the e-book.

Requirement All values must be enclosed in quotation marks.

Tip Refer to the Dublin Core Metadata Element Set for more information about metadata as supported by the relevant OPTIONS suboptions.

STYLE= style-override(s)  
specifies one or more style-override(s) to use when writing output files. Daisy is the default style for EPUB documents.

You can specify a style override in two ways:

- Specify a style element. A style element is a collection of style attributes that apply to a particular part of the output for a SAS program.
- Specify a style attribute. A style attribute is a name-value pair that describes a single behavioral or visual aspect of a piece of output. This is the most specific method of changing the appearance of your output.

style-override(s) has the following form:

style-element-name | [style-attribute-name-1=style-attribute-value-1  
<style-attribute-name-2=style-attribute-value-2 …>]

Default Daisy is the default style. For information on using the SAS Registry to change defaults, see “Changing SAS Registry Settings for ODS” on page 40.

See For a complete discussion about styles, see Chapter 9, “Overview,” on page 913.

STYLESHEET= ( URL= ’external-file(s)’ )  
specifies one or more style sheet files to apply to the output.

(URL= ’external-file(s)’)  
specifies one or more space-delimited external style sheet filenames.

Requirement external-file(s) must be enclosed in parentheses.
TITLE="text-string"
inserts the title into the metadata of the e-book. The title also appears as an identifier in the e-book reader library.

Default 'SAS Output'

Requirement Every EPUB e-book must have a title.

Note This title is the title of the book and not a SAS title.

WORK='directory-name'
specifies a work directory for constructing the e-book.

directory-name
is the name of the directory.

Details

Overview of ODS EPUB2
The ODS EPUB2 statement generates output that has the .epub extension. EPUB (electronic publication) is a free and open e-book standard produced by the International Digital Publishing Forum (IDPF). EPUB is designed for reflowable content, meaning that an EPUB reader can optimize text for a particular display device. E-books that use the .epub extension can be read by a wide variety of e-book readers, from dedicated hardware to desktop software to online based readers.

The .epub file extension consists of XML files for reflowable digital books and publications. The .epub file is a ZIP archive containing the books files, either XHTML or DTBook, a number of XML description and navigation files, and possibly image or media files.

The version of EPUB generated by the ODS EPUB destination is specified in the SAS registry. In the SAS Registry, click on ODS to find the default EPUB version. See “Changing ODS EPUB Destination Default Value” on page 48.

Accessibility of EPUB Output
The ODS EPUB destinations (ODS EPUB2 and ODS EPUB3) are the recommended destinations for creating SAS output that is accessible to the broadest audience. They create e-books that use many of the accessibility features of the EPUB specification. These features allow e-book readers such as iBooks to present e-books so that they adapt to the needs of users with disabilities. For example, when reading an e-book created by ODS EPUB destinations using iBooks on an iPad, users can adjust font size, color schemes, and magnification. They can also access the text using assistive technologies such as the Voiceover screen reader and refreshable braille displays.

Examples

Example 1: Create an ODS EPUB2 Book with a Title and Creator

Details
Every EPUB e-book must have a title. If the user does not assign one, the ODS EPUB2 destination assigns the title 'SAS Output'.

Users assign an author with the CREATOR suboption of the OPTIONS option. An author is not required EPUB e-book metadata.
Note: The following program assumes that the default EPUB version that is set in the Registry is EPUB2. See “Changing ODS EPUB Destination Default Value” on page 48 for additional information.

Program

data drugtest;
    input Drug $ PreTreatment PostTreatment @@;
datalines;
A 11 6 A 8 0 A 5 2 A 14 8 A 19 11
A 6 4 A 10 13 A 6 1 A 11 8 A 3 0
D 6 0 D 6 2 D 7 3 D 8 1 D 18 18
D 8 4 D 19 14 D 8 9 D 5 1 D 15 9
F 16 13 F 13 10 F 11 18 F 9 5 F 21 23
F 16 12 F 12 5 F 12 16 F 7 1 F 12 20
;
ods epub file="glm.epub" title="My First ODS EPUB E-book"
options(creator="SAS Programmer" description="My First ODS EPUB Book"
subject="PROC GLM" type="ODS EPUB book");
ods graphics on;
proc glm data=DrugTest;
    class Drug;
    model PostTreatment = Drug|PreTreatment;
run;
quit;
ods epub close;

Program Description

Create the drugtest data set.

data drugtest;
    input Drug $ PreTreatment PostTreatment @@;
datalines;
A 11 6 A 8 0 A 5 2 A 14 8 A 19 11
A 6 4 A 10 13 A 6 1 A 11 8 A 3 0
D 6 0 D 6 2 D 7 3 D 8 1 D 18 18
D 8 4 D 19 14 D 8 9 D 5 1 D 15 9
F 16 13 F 13 10 F 11 18 F 9 5 F 21 23
F 16 12 F 12 5 F 12 16 F 7 1 F 12 20
;

Create ODS EPUB output. Every EPUB e-book must have a title. If the user does not assign one, the ODS EPUB destination assigns the title 'SAS Output'. Users can insert information about the e-book into the metadata using the OPTIONS options.

ods epub file="glm.epub" title="My First ODS EPUB E-book"
options(creator="SAS Programmer" description="My First ODS EPUB Book"
subject="PROC GLM" type="ODS EPUB book");

Create PROC GLM output.

ods graphics on;
proc glm data=DrugTest;
    class Drug;
model PostTreatment = Drug|PreTreatment;
run;
quit;

Close the EPUB destination.
ods epub close;

Output
This program generates a simple EPUB book that is read on the iBooks reader.

Output 6.11  EPUB Book Title Page

The ODS EPUB statement produces output in the form of an EPUB book.
Example 2: Create an ODS EPUB E-book with Inline and Nonlinear Tables

Details

By default, tables are placed inline, which means that they appear in the linear flow of reporting elements. If the user specifies that the OPTIONS suboption NONLINEAR='table', the ODS EPUB destination places links to the generated tables instead of placing them inline. In other words, the generated tables are placed outside the linear flow of the chapter.

For Apple iBooks users, nonlinear tables can access an iBooks feature that permits panning and zooming of nonlinear tables. This iBooks feature is especially useful when a table is so wide or so tall that it does not fit in the device display in linear mode.

Note: The following program assumes that the default EPUB version that is set in the Registry is EPUB2. See “Changing ODS EPUB Destination Default Value” on page 48 for additional information.

Program

```sas
ods epub file="nonLinearTablesEx2.epub"
   title="Inline and Non-Linear Tables"
   options(creator="SAS Programmer");
ods proclabel='Inline Table';
title 'Inline Table';
proc print data=sashelp.prdsal2(obs=15);
   run;
ods proclabel='Non-linear Table';
title 'Non-linear Table';
ods epub options(nonlinear="table");
```
proc print data=sashelp.prdsal2(obs=15);
    run;
ods epub close;
title;

Program Description

Open the ODS EPUB destination. Open the ODS EPUB destination, provide a file for the EPUB output, and give the EPUB document a title. Also add information about the creator of the e-book to the metadata.

ods epub file="nonLinearTablesEx2.epub"
    title="Inline and Non-Linear Tables"
    options(creator="SAS Programmer");

Create a label for the inline table. Create a label for the inline table. Print the data set as an inline table.

ods proclabel='Inline Table';
title 'Inline Table';
proc print data=sashelp.prdsal2(obs=15);
    run;

Create a label for the non-linear table. Create a label for the non-linear table and print the data set as a nonlinear table.

ods proclabel='Non-linear Table';
title 'Non-linear Table';
ods epub options(nonlinear="table");


proc print data=sashelp.prdsal2(obs=15);
    run;

Close the EPUB destination.

ods epub close;
title;

Output

An EPUB book is generated showing inline and nonlinear tables.
**Example 3: ODS EPUB Book with a Customized TOC**

**Details**

In this example program, a customized table of contents is created in the ODS EPUB e-book by specifying the EVENT=BRANCH option. The BRANCH event customizes the table of contents using the LABEL= suboption to specify the text of the link, and URL=.
specifies the e-book location. The OPTIONS CONTENTS suboption controls whether the table of contents is created. In this example, the TOC is turned off after the
BRANCH events complete in order to suppress the standard PROC PRINT table of contents entries.

Note: The following program assumes that the default EPUB version that is set in the Registry is EPUB2. See “Changing ODS EPUB Destination Default Value” on page 48 for additional information.

Program

title;
ods epub file="customtoc.epub" title="E-book with Custom TOC";
ods epub event=branch(start label="My SASHELP.CLASS Data Set Link"
url="chapter1.html";
ods epub event=branch(finish);
ods epub options(contents="off");
proc print data=sashelp.class;
  run;
ods epub close;

Program Description

Turn off the system titles.

title;

Open the ODS EPUB destination. Open the ODS EPUB destination, provide a unique filename for the EPUB output, and give the EPUB document a title.

ods epub file="customtoc.epub" title="E-book with Custom TOC";

The BRANCH event customizes the table of contents. The BRANCH event customizes the table of contents. The LABEL= suboption specifies the text of the link (My SASHELP.CLASS Data Set Link) that shows up in the TOC. The URL= suboption specifies the e-book location, chapter1.html. By default, an ODS EPUB e-book has a single chapter. When you tap the TOC, you are taken to chapter one of the book. Note that ODS EPUB e-book chapters are always enumerated in the form chapter1.html, chapter2.html, and so on.

ods epub event=branch(start label="My SASHELP.CLASS Data Set Link"
url="chapter1.html";

Turn off the BRANCH event.

ods epub event=branch(finish);

Do not generate the standard PROC PRINT table of contents. The CONTENTS= suboption turns off the creation of the standard PROC PRINT table of contents entries. If contents are not turned off, you get both the customized entry in the TOC and the standard PROC PRINT entry in the TOC.

ods epub options(contents="off");
Print the data set Sashelp.class. The PROC PRINT statement prints the Sashelp.class data.

```sas
proc print data=sashelp.class;
  run;
```

Close the EPUB destination.

```sas
ods epub close;
```

Output

The following output shows the cover page and the customized TOC for the book.

**Output 6.17  ODS EPUB E-book with a Customized TOC**

---

**ODS EPUB3 Statement**

Opens, manages, or closes the EPUB3 destination, which generates EPUB e-books.

- **Valid in:** Anywhere
- **Category:** ODS: Third-Party Formatted
- **Defaults:** In the third maintenance release of SAS 9.4, ODS EPUB3 is the default EPUB destination and is set in the Registry. To change your default EPUB version in the Registry, see “Changing ODS EPUB Destination Default Value” on page 48.
  
The default style for EPUB destinations is Daisy.
  
The default name for EPUB files is “sasepub.epub”.

- **Interaction:** By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC CONTENTS), ODS formats the output in SAS Monospace font. If you are creating output that is viewed in an operating environment where SAS
software is not installed, this output will not be displayed correctly. This is because without SAS, the SAS Monospace font is not recognized. To make your document display correctly, include the following statement before your SAS program:

```
OPTIONS FORMCHAR="|----|+|---+=-|\<>**";
```

**ODS destination:** The version of EPUB generated by the ODS EPUB destination is specified in the SAS registry. The version of EPUB generated by the ODS EPUB3 destination is EPUB3.

**z/OS specifics:** On z/OS, the ODS destination for EPUB works only with the Hierarchical File System (HFS) file system. You must use the FILESYSTEM=HFS option. The external file specified by the FILE= option must be an HFS file. If the WORK= option is used, the directory must be an HFS directory.

**Notes:** Encoding is set in the SAS Registry for ODS EPUB destinations. See “Changing the ODS EPUB and EPUB3 Encoding Settings” on page 49. If you want to customize your markup output, continue to use ODS MARKUP for that purpose. The EPUB destination does not permit user customization beyond sending surfaced EVENT= events.

iBooks on iPad is the preferred e-book reader. Results can vary using other e-book readers.

**Syntax**

```
ODS EPUB3 (<ID=> identifier) < action > ;
ODS EPUB3 (<ID=> identifier) <option(s)> ;
```

**Summary of Optional Arguments**

- **(ID= identifier)**
  - Open multiple instances of the same destination at the same time

- **ANCHOR= 'anchor-name'**
  - Specify a unique base name for the anchor tag that identifies each output object in the current chapter

- **BOX_SIZING=(CONTENT_BOX | BORDER_BOX)**
  - Specify how to measure the width of cells. Use to override the default value of BOX_SIZING for a destination

- **CLOSE**
  - Close the destination and the file that is associated with it

- **CSSSTYLE= 'file-specification' <(media-type-1<…media-type-10>)>**
  - Specify a cascading style sheet to apply to your output

- **DEVICE= device-driver**
  - Specify the name of a device driver.

- **DOM=<"external-file">**
  - Specify that the ODS document object model is written to the SAS log or to an external file.

- **EVENT=event-name(FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL=)**
  - Specify an event and the value for event variables that are associated with the event

- **EXCLUDE exclusion(s) | ALL | NONE**
  - Exclude output objects from the destination
FILE='file-specification'
   Specify the file that contains the e-book created by the destination

GFOOTNOTE | NOGFOOTNOTE
   Control the location where footnotes are printed in the graphics output

GTITLE | NOGTITLE
   Control the location where titles are printed in the graphics output

IMAGE_DPI
   Specify the image resolution for the graphical output

NEWCHAPTER=starting-point
   Create a new chapter at the specified starting point

OPTIONS (<suboption(s)>)
   Specify destination-specific suboptions and a space-delimited named value for how e-books are handled

SELECT selection(s) | ALL | NONE
   Select output objects for the destination

SHOW
   Write to the SAS log the current selection or exclusion list for the destination

STYLE= style-override(s)
   Specifies one or more style-overrides to use when writing output files

STYLESHEET=( URL= 'external-file(s)' )
   Specify one or more style sheet files to apply to the output

TITLE='text-string'
   Insert the text string of a title that you specify into the metadata of the e-book

WORK=directory-name'
   Specify a work directory for constructing the e-book

Without Arguments
If you use the ODS EPUB3 statement without an action or options, then it opens the EPUB3 destination and creates EPUB output.

Actions
The following actions are available for the ODS EPUB3 statement.

CLOSE
   closes the destination and any files that are associated with it.

   Tip   When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

EXCLUDE exclusion(s) | ALL | NONE
   excludes one or more output objects from the destination.

   Default     NONE

   Restriction A destination must be open for this action to take effect.

   See        “ODS EXCLUDE Statement” on page 321

SELECT selection(s) | ALL | NONE
   selects output objects for the specified destination.

   Default     ALL
Restriction  A destination must be open for this action to take effect.

See  “ODS SELECT Statement ” on page 758

SHOW
writes the current selection list or exclusion list for the destination to the SAS log.

Restriction  The destination must be open for this action to take effect.

Tip  If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

See  “ODS SHOW Statement” on page 771

Optional Arguments

ANCHOR= 'anchor-name'
specifies a unique base name for the anchor tag that identifies each output object in the current body file.

Each output object has an anchor tag. The links and references are automatically created by ODS. These links and references point to the name of an anchor. Therefore, each anchor name in a file must be unique.

anchor-name
is the base name for the anchor tag that identifies each output object in the current body file.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR= 'TABULATE', then ODS names the first anchor tabulate. The second anchor is named tabulate1; the third is named tabulate2, and so on.

Restrictions  Each anchor name in a file must be unique.

Requirement  You must enclose anchor-name in quotation marks.

Tip  You can change anchor names as often as you want by specifying the ANCHOR= option in an ODS EPUB statement anywhere in your program. After you have specified an anchor name, it remains in effect until you specify a new one.

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry. In the SAS Registry, expand the ODS ⇨ DESTINATION ⇨ SCRIPT folder to locate the box_sizing= default for ODS EPUB destinations. For information about using the SAS Registry, see “Changing SAS Registry Settings for ODS” on page 40.
BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing.

CSSSTYLE= 'file-specification'<(media-type-1<...media-type-10>)> specifies a cascading style sheet to apply to your output.

file-specification
specifies a file, fileref, or URL that contains CSS code.

"external-file"
is the name of the external file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

"URL"
is a URL to an external file.

Requirement You must enclose URL in quotation marks.

(media-type-1<.. media-type-10>) specifies one or more media blocks that correspond to the type of media that your output is rendered on. CSS uses media type blocks to specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not contained in any media blocks. By using the media-type suboption, in addition to the general CSS code, you can import the section of a CSS file intended only for a specific media type.

Default If no media-type is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

Range You can specify up to ten different media types.

Requirements You must enclose media-type in parentheses.

You must specify media-type next to the file-specification specified by the CSSSTYLE= option.

Tip If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

Restriction The CSSSTYLE= option does not affect SAS/GRAPH output.

Requirement CSS files must be written in the same type of CSS produced by the ODS HTML statement. Only class names are supported, with no IDs
and no context-based selectors. To view the CSS code that ODS creates, you can do one of the following:

- specify the ODS TRACE DOM statement
- specify the DOM option

**Interaction**

If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

**See**

For complete documentation about ODS and Cascading Style Sheets, see *SAS Output Delivery System: Advanced Topics*

**Example**

“Example 6: Applying a CSS File to ODS Output” on page 527

**DEVICE= device-driver**

specifies the name of a device driver. ODS automatically selects an optimal default device for each open output destination.

The default device value can be found in the SAS registry. In the SAS Registry, expand the **ODS ➔ DESTINATION ➔ SCRIPT** folder to locate the DEVICE= default for ODS EPUB destinations. For information about using the SAS Registry, see “Changing SAS Registry Settings for ODS” on page 40.

**Range**

ODS uses the device driver that is specified in the SAS registry.

**Restriction**

Valid devices are GIF, JPEG, PNG, and SVG.

**Tip**

Specifying a device with the DEVICE= option takes precedence over the SAS global option and the graphics option.

**See**


**DOM<="external-file">**

specifies that the ODS document object model is written to the SAS log or an external file.

**external-file**

is the name of an external output file.

**Requirement**

You must enclose *external-file* in quotation marks.

**See**

For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in *SAS Output Delivery System: Advanced Topics*.

**EVENT=EVENT=**

**LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL=>**

specifies an event and the value for event variables that are associated with the event.

*Event-name* accepts the following arguments: AUDIO, BRANCH, FIGURE, FIGCAPTION, IMAGE, KEEP, MATHML, PDF, VIDEO.

**BRANCH= (START | FINISH)**

specifies that an entry for the chapter is to be created in the table of contents for the EPUB3 book. ODS EPUB3 does not automatically create table of contents entries for nonlinear output. BRANCH events can be nested to build nested table of contents entries. Match up the START and FINISH options to the desired depth.
For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

Example:
```sas
ods epub3 event=branch(start label="Hello World!" url="chapter1.html");
```

**FIGURE=(START | FINISH)**

adds figures that are embedded in the e-book book. This block contains a single image. ODS EPUB3 implements FIGURE as an event.

This EVENT is the recommended markup for captioned content that is self-contained and can be separated from the main flow of the document.

*Note:* The ability to distinguish between primary and secondary content is an important consideration for accessibility.

**START**

starts the code block that embeds the figure.

**FINISH**

ends the code block that embeds the figure.

**Default**
The default Daisy style element for a figure is "figure".

---

For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

**Example**:
```sas
%macro figure(image, caption);
ods epub3 event=figure(start)
    event=image(url=&image )
    event=figcaption(text=&caption)
    event=figure(finish);
%mend;
```

**FIGCAPTION=(TEXT=text-string)**

adds text to the figure. This option provides markup for captioned content that is self-contained and can be separated from the main flow of the document. A figure can contain one caption and it must appear first or last in the caption content. ODS EPUB3 implements FIGCAPTION as an event. The FIGCAPTION event accepts TEXT, which is the text to be displayed above or below the content.

iBooks, ADE, AZARDI, Calibre, and Readium support FIGCAPTION.

*Note:* The ability to distinguish between primary and secondary content is an important consideration for accessibility.

**Default**
The default Daisy style element for a caption is "figurecaption".

---

For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

**Example**:
```sas
%macro figure(image, caption);
ods epub3 event=figure(start)
    event=image(url=&image )
    event=figcaption(text=&caption)
    event=figure(finish);
%mend;
```

**(FINISH)**

triggers the finish section of an event.
See For information about events, see “Understanding Events” in *SAS Output Delivery System: Procedures Guide*.

**IMAGE (URL= *variable-value*)**
specifies a URL for the image.

See For information about events, see “Understanding Events” in *SAS Output Delivery System: Procedures Guide*.

**Example**

```sas
%macro figure(image, caption);
  ods epub3 event=figure(start)
    event= image(url=&image )
    event=figcaption(text=&caption)
  event=figure(finish);
%mend;
```

**(LABEL=’*variable-value’)**
specifies the value for the LABEL event variable.

Requirement *variable-value* must be enclosed in quotation marks.

See For information about the LABEL event variable, see “Event Variables” in *SAS Output Delivery System: Procedures Guide*.

**(NAME=’*variable-value’)**
specifies the value for the NAME event variable.

Requirement *variable-value* must be enclosed in quotation marks.

See For information about the NAME event variable, see “Event Variables” in *SAS Output Delivery System: Procedures Guide*.

**(START)**
triggers the start of an event.

See For information about events, see “Understanding Events” in *SAS Output Delivery System: Procedures Guide*.

**(STYLE= *style-element*)**
specifies a style element.

See For information about style elements, see “Style Attributes Overview” in *SAS Output Delivery System: Procedures Guide*.

**(TARGET=’*variable-value’)**
specifies the value for the TARGET event variable.

Requirement *variable-value* must be enclosed in quotation marks.

See For information about the TARGET event variable, see “Event Variables” in *SAS Output Delivery System: Procedures Guide*.

**(TEXT=’*variable-value’)**
specifies the value for the TEXT event variable.

Requirement *variable-value* must be enclosed in quotation marks.
See For information about the TEXT event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(URL='variable-value')
specifies the value for the URL event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the URL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

Example “Example 3: ODS EPUB3 - Adding Figures and Figure Captions Using Events” on page 255

Requirement The EVENT= event-name suboptions must be enclosed in parentheses.

FILE='file-specification'
specifies the file that contains the e-book created by the destination.

'file-specification'
specifies the file and fileref to receive output.

file-specification is one of the following:

external-file
is the name of an external file to receive output.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

Default ODS uses the filename that is specified in the SAS registry. The default name for EPUB files is sasepub.epub. For information on using the SAS Registry to change defaults, see “Changing SAS Registry Settings for ODS” on page 40.

GFOOTNOTE | NOGFOOTNOTE
controls the location where footnotes are printed in the graphics output.

GFOOTNOTE
prints footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

NOGFOOTNOTE
prints footnotes that are created by ODS, which appear outside the graph borders.

Default GFOOTNOTE

Restrictions Footnotes that are displayed support most SAS/GRAPH FOOTNOTE options. The font must be valid for the e-book reader. Options that ODS cannot handle, such as text angle specifications, are ignored.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.
GTITLE | NOGTITLE
controls the location where titles are printed in the graphics output.

GTITLE
prints the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.

NOGTITLE
prints the title that is created by ODS, which appears outside of the graph borders.

Default GTITLE

Restrictions
Titles that are displayed support most SAS/GRAPH TITLE options. The font must be valid for the e-book reader. Options that ODS cannot handle, such as text angle specifications, are ignored.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

(ID= identifier)
enables you to run multiple instances of the same destination at the same time. Each instance can have different options.

identifier
specifies another instance of the destination that is already open. identifier is numeric or a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numeric characters.

Restriction
If identifier is numeric, it must be a positive integer.

Requirement
You must specify the ID= option immediately after the ODS EPUB3 statement keywords.

Tip
You can omit the ID= option and instead use a name or a number to identify the instance.

IMAGE_DPI
specifies the image resolution for graphical output.

Alias DPI=

Default ODS uses the resolution specified in the SAS registry.

NEWCHAPTER=starting-point
creates a new chapter at the specified starting point.

starting-point
is the location in the e-book where you want to create a new chapter.

starting-point is one of the following:

BYGROUP
starts a new chapter for the results of each BY group.

NONE
writes all output to the current chapter.
NOW  
starts a new chapter and writes all output to that new chapter.

OUTPUT  
starts a new chapter for each output object.

Alias  TABLE  

PAGE  
starts a new chapter for each page of output. A page break occurs when you start a new procedure, or when a procedure explicitly starts a new page.

PROC  
starts a new chapter for each procedure.

Default  NONE  

OPTIONS (< suboption(s)>)
specifies destination-specific suboptions with space-delimited name='value' pairs.

suboption(s) are the following:

(CHAPTER_TYPE='chapter-types')  
Specifies the type of the current chapter. The type is one or more space-delimited terms defined by EPUB 3 Structural Semantics Vocabulary. For more information, see http://www.idpf.org/epub/vocab/structure/.  

Default  Default is 'bodymatter chapter'.  

(CONTENTS= 'YES' | 'NO' | 'OFF' | 'ON')  
generates a table of contents.

OFF  
does not generate a table of contents.

Alias  NO  

ON  
generates a table of contents.

Alias  YES  

Default  ON | YES  

(CONTRIBUTOR='text-string')  
specifies one or more contributor’s names responsible for making contributions to the content of the e-book. Delimit multiple contributors with commas.

(COVER_IMAGE='external-file')  
specifies a cover image for the e-book.

Requirement  You must enclose external-file in quotation marks.

(COVERAGE='text-string')  
specifies the extent or scope of the content of the e-book.

(CREATOR='text-string')  
specifies one or more primary creators or authors of the e-book.
Requirement | All values must be enclosed in quotation marks.
---|---
Note | Delimit multiple creators with commas.

(\texttt{DEFEAT_IBOOKS_CACHING= 'YES' | 'NO' | 'OFF' | 'ON'})
defeats iBooks e-book reader's caching of e-books that have the same title.

\begin{tabular}{|l|}
\hline
\textbf{OFF} & caches iBooks of the same title. \\
\hline
\textbf{Alias} & NO \\
\hline
\textbf{ON} & defeats iBooks caching. \\
\hline
\textbf{Alias} & YES \\
\hline
\textbf{Default} & ON | YES \\
\hline
\end{tabular}

(\texttt{DESCRIPTION='text-string'})
specifies a description of the content of the e-book.

(\texttt{EMBEDDED_FONTS='fonts'})
specifies a list of fonts to be embedded in the e-book.

\textbf{See} For information about using fonts, see “Using Fonts with Universal Printers and SAS/GRAPH Devices” in \textit{SAS Language Reference: Concepts}.

(\texttt{HIDE_CONTENTS= 'YES' | 'NO' | 'OFF' | 'ON'})
hides entries in a linear table of contents.

\begin{tabular}{|l|}
\hline
\textbf{OFF} & does not hide the entries in a linear table of contents. \\
\hline
\textbf{Alias} & NO \\
\hline
\textbf{ON} & hides the entries in a linear table of contents. \\
\hline
\textbf{Alias} & YES \\
\hline
\textbf{Default} & OFF | NO \\
\hline
\end{tabular}

(\texttt{HIDE_LANDMARKS= 'YES' | 'NO' | 'OFF' | 'ON'})
hides entries in a linear table of landmarks.

\begin{tabular}{|l|}
\hline
\textbf{YES} & hides entries in a linear table of landmarks. \\
\hline
\textbf{NO} & does not hide the entries in a linear table of landmarks. \\
\hline
\textbf{OFF} & does not hide the entries in a linear table of landmarks. \\
\hline
\textbf{Alias} & NO \\
\hline
\end{tabular}
ON

  hides the entries in a linear table of landmarks.

Alias  YES

Default  OFF | NO

(ISBN='isbn')

Requirement  Every EPUB e-book must be uniquely identifiable.

(LANDMARK='landmark' | 'AUTO')
specifies the landmark for the current chapter. If ’AUTO’ is specified, the landmark will be derived from the chapter type.

See  See “(CHAPTER_TYPE='chapter-types’)” on page 243.

(LINEAR_NAV= 'NONE' | 'CONTENTS' | 'LANDMARKS' | 'ALL')
specifies navigational aids to embed in the e-book.

NONE  no navigational aids are embedded in the e-book.

CONTENTS  causes the table of contents to be embedded.

LANDMARKS  causes table of landmarks to be embedded.

ALL  causes the table of contents and the table of landmarks to be embedded.

Default  NONE

(NONLINEAR= 'NONE' | 'CHAPTER' | 'BATCH' | 'TABLE' | 'ALL')
specifies output to be written to the non-linear section of the e-book.

NONE  occurs within the linear flow of the e-book.

CHAPTER  causes the entire chapter to be non-linear.

BATCH  causes batch (preformatted) output to be non-linear.

TABLE  causes tables to be non-linear.

ALL  causes the entire chapter to be non-linear. This has the same effect as the CHAPTER value.

Default  NONE
(OPTIMIZE_FOR_IBOOKS= 'YES' | 'NO' | 'OFF' | 'ON') optimizes for iBooks e-book reader.

OFF overrides user-specified fonts.

Alias NO

ON honors user-specified fonts.

Alias YES

Default ON | YES

Note iBooks on iPad is the preferred e-book reader. Results can vary using other e-book readers.

(PAGEBREAK= 'YES' | 'NO' | 'OFF' | 'ON') honors explicit page breaks.

OFF does not honor explicit page breaks.

Alias NO

ON honors explicit page breaks.

Alias YES

AUTO allows ODS and procedures to determine when page breaks occur. Honors explicit page breaks.

Default AUTO

(PUBLISHER='text-string') specifies the publisher of the e-book.

(RELATION='text-string') specifies a reference to a related resource.

(RIGHTS='text-string') specifies information about rights held in and over the e-book.

(SOURCE='text-string') specifies information about a prior resource from which the e-book was derived.

(START='chapter-number') specifies the component (chapter) at which an e-book first opens.

Default 1

(SUBJECT='text-string') specifies one or more topics about the content of the e-book. Delimit multiple subjects with commas.

(TYPE='text-string') specifies the nature or genre of the content of the e-book.
Requirement: All values must be enclosed in quotation marks.

Tip: For more information about metadata as supported by the relevant OPTIONS suboptions, refer to the Dublin Core Metadata Element Set at http://dublincore.org/documents/dces/.

**STYLE= style-override(s)**

specifies one or more style-overrides to use when writing output files.

You can specify a style override in two ways:

- Specify a style element. A style element is a collection of style attributes that apply to a particular part of the output for a SAS program.
- Specify a style attribute. A style attribute is a name-value pair that describes a single behavioral or visual aspect of a piece of output. This is the most specific method of changing the appearance of your output.

style-override(s) has the following form:

```plaintext
style-element-name | [style-attribute-name-1=style-attribute-value-1 <style-attribute-name-2=style-attribute-value-2 …>]
```

Default: If you do not specify a style-override, then ODS uses the style specified in the SAS registry. The default style for EPUB is Daisy. For information on using the SAS Registry to change defaults, see “Changing SAS Registry Settings for ODS” on page 40.

See: For a complete discussion about styles, see Chapter 9, “Overview,” on page 913.

**STYLESHEET= ( URL= ‘external-file(s)’ )**

specifies one or more style sheet files to apply to the output.

(URL= ’external-file(s)’)

specifies one or more space-delimited external style sheet filenames.

Requirement: external-file(s) must be enclosed in parentheses.

**TITLE=’text-string’**

inserts the title that you specify into the metadata of the e-book. The title also appears as an identifier in the e-book reader library.

Default: ‘SAS Output’

Requirement: Every EPUB e-book must have a title.

Note: This title is the title of the book and not a SAS title.

**WORK=directory-name’**

specifies a work directory for constructing the e-book.

directory-name

is the name of the directory.
Details

Overview of EPUB and EPUB 3
The ODS EPUB statements generate output that has the .epub extension. EPUB (electronic publication) is a free and open e-book standard produced by the International Digital Publishing Forum (IDPF). EPUB is designed for reflowable content, meaning that an EPUB reader can optimize text for a particular display device. E-books that use the .epub extension can be read by a wide variety of e-book readers, from dedicated hardware to desktop software to online based readers.

ODS EPUB3 supports HTML5, CSS2 and CSS3, and SVG. ODS EPUB3 supports user-specified audio and video. Any e-book reader that supports EPUB3 supports ODS EPUB3 audio and video.

ODS EPUB3 also provides the following features:

- ODS EPUB3 supports web open font format (WOFF) fonts. For more information, see http://en.wikipedia.org/wiki/Web_Open_Font_Format.
- ODS EPUB3 provides the ability for the customer to define pop-up footnotes for their e-books.
- ODS EPUB3 also provides more accessible e-book documentation.
- iBooks implements much of the 2011 EPUB 3 specification, which is based in part on HTML 5 and CSS 3. Those technology standards allow for audio and video, as well as advanced presentation features such as multicolumn layout. iBooks supports those EPUB 3 features and more.

The version of EPUB generated by the ODS EPUB destination is specified in the SAS registry. In the SAS Registry, click on ODS to find the default EPUB version. For information about changing defaults in the SAS Registry, see “Changing SAS Registry Settings for ODS” on page 40.

ODS EPUB3 Support of Multimedia

Overview
Multimedia describes the combination of content types in an e-book or other electronic publication. Content types include text, still images, animated images, audio, and video. ODS EPUB3 supports images, audio, and video by using destination-specific inline formatting functions AUDIO, VIDEO, IMAGE, by using the Report Writing Interface (RWI) methods AUDIO, VIDEO, IMAGE, and by using the EVENT= IMAGE option in ODS EPUB3.

Generally, audio and video are embedded in EPUB e-books so that they can be accessed offline. Typically, e-books embed just one video or audio file. This helps keep the file size manageable.

Because ODS EPUB and ODS EPUB3 target Apple iBooks, the supported video and audio types are the ones that iBooks supports.

When using the RWI Audio and Video methods, if you specify multiple audio files or videos, ODS EPUB3 accepts only the first audio or the first video of a supported type. The other types are not accepted. ODS EPUB3 issues a warning for each unaccepted audio or video. The warning will be one of two kinds:

1. The audio or video is not of a supported type.
2. A previous audio or video was selected, and therefore this one is being ignored.
Note: Apple prefers m4a format, but supports the mp3 format. W3C has not standardized a video format.

Using ODS EPUB EVENTS to Add Image Files
There are various ways to add images to your e-book. One of them is by using an event that is specified in the ODS EPUB3 statement using the OPTIONS option. Here is an example.

%macro figure(image, caption);
ods epub3 event=figure(start)
  event=image(url=&image )
  event=figcaption(text=&caption)
  event=figure(finish);
%mend;

See “Example 3: ODS EPUB3 - Adding Figures and Figure Captions Using Events” on page 255 for a complete example.

Using Destination-Specific Inline Formatting Functions AUDIO, VIDEO, IMAGE
Multimedia combines text with images, audio, and video. A way to add images, video, and audio to your e-book is by using destination-specific inline formatting functions. In order to use this technique, you need to specify them as part of SAS language text statements like the ODSTEXT statement.

You can see examples of using ODS ESCAPECHAR functions in the following examples and documents.

• Advanced Mobile Reporting with the ODS EPUB3 Destination
• See “Example 4: ODS EPUB3 - Adding Audio Using Inline Functions” on page 259.
• See “Example 5: ODS EPUB3 - Adding Video Using Inline Functions” on page 261.

Using the Report Writing Interface (RWI) Methods AUDIO, VIDEO, and IMAGE
Multimedia combines text with images, audio, and video. A way to add images, video, and audio to your e-book is by using the Report Writing Interface (RWI).

You can see examples of using the RWI multimedia methods in the following examples and documents.

• See “Example 6: ODS EPUB3 - Adding Audio Using RWI” on page 263.
• See “Example 7: ODS EPUB3 - Adding Video Using RWI” on page 265.
• See “IMAGE Method” in SAS Output Delivery System: Advanced Topics.

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

Some of the more accessible features of ODS EPUB 3 are as follows:
• ODS EPUB3 supports the ability to embed the table of contents in the e-book. This can be useful when an e-book application does not work as expected for the table of contents or is not accessible to visually impaired readers.

• ODS EPUB3 supports the ability to define landmarks. Landmarks are a supplemental navigational aid that can help readers access specific sections of the e-book quickly.

• ODS EPUB3 can define table of contents and landmark entries. The landmark entries can be hidden for sighted readers but can also be accessible to visually impaired readers. The visually impaired readers benefit from the additional context.

• ODS EPUB3 supports accessibility features for MathML. The ODS ESCAPECHAR MATHML function implements this feature. The accessible features for MathML are large print with the ability to drill down on terms for low vision users, voiceover that speaks the math, and voiceover that displays the math on a refreshable braille display using the Nemeth braille code.

MathML is supported in both iBooks and Safari in iOS 7 and later versions. This support presents MathML in multiple modes simultaneously.

Note: Math must be encoded using "presentation MathML" to be rendered using the features above.

Additional reading about EPUB content, EPUB3 Accessibility, and the support of MathML in EPUB3, see the following documents.

• EPUB Content Documents 3.0
• EPUB 3 Accessibility Guidelines
• iBooks is built with WebKit. WebKit's support for MathML is documented in WebKit's MathML Support Documentation

Examples

Example 1: ODS EPUB E-book with TOC and Landmark TOC
Features:
  ODS EPUB3 Statement
    FILE= option
    TITLE= option
    OPTIONS linear navigation suboption

Other features:
  Data Set
  ARIMA procedure

Details
The OPTIONS LINEAR_NAV suboption can be used to embed a table of contents and landmarks into the e-book. Some e-book applications do not provide the ability to display landmarks unless they are embedded. The LINEAR_NAV suboption enables you to add landmarks to the TOC. In the following example, you can see the iBooks generated TOC plus the landmark TOC included. Note that for visually impaired readers, the ability to embed the landmark tables in the TOC can be helpful.

Program

title;

ods epub3 file="tocwithlandmarktoc.epub"
title="PROC ARIMA TOC and Landmark TOC in EPUB3 e-book"
   options(linear_nav="all");
   data a;
         input x @@;
   cards;
      01 02 03 02 03 04 05 04 06 07 06 08 09 10 10 10
      09 11 11 13 17 13 14 15 15 16 18 19 20 21 23 24
   ;
   proc arima data=a;
         identify nlag=6 var=x(1) outcov=cov1;
   run;
         estimate p=1 outest=est1 outmodel=mod1 outstat=stat1;
   run;
         forecast lead=2 out=out1 alpha=0.10;
   run;
   quit;
ods epub3 close;

Program Description

Turn off the system titles. Turn off the system title.
   title;

Open the ODS EPUB3 destination. Open the ODS EPUB3 destination, provide a file
for the EPUB output, and give the EPUB document a title. Use (LINEAR_NAV="ALL")
to provide all of the navigational aids available.
   ods epub3 file="tocwithlandmarktoc.epub"
   title="PROC ARIMA TOC and Landmark TOC in EPUB3 e-book"
   options(linear_nav="all");

Read in the data set.
   data a;
         input x @@;
   cards;
      01 02 03 02 03 04 05 04 06 07 06 08 09 10 10 10
      09 11 11 13 17 13 14 15 15 16 18 19 20 21 23 24
   ;

Run PROC ARIMA
   proc arima data=a;
         identify nlag=6 var=x(1) outcov=cov1;
   run;
         estimate p=1 outest=est1 outmodel=mod1 outstat=stat1;
   run;
         forecast lead=2 out=out1 alpha=0.10;
   run;
   quit;

Close the EPUB 3 destination.
ods epub3 close;

Output
The following display shows the ODS EPUB3 e-book with a TOC and a landmark TOC.

Output 6.18 ODS EPUB3 E-book with TOC and Landmark TOC

Example 2: ODS EPUB3 - Adding Landmark TOC Entries
Features:
ODS EPUB3 Statement
FILE= option
TITLE= option
NEWCHAPTER=
OPTIONS linear navigation, creator, contents, chapter_type, landmark, start, landmark suboptions

ODSTEXT Procedure

ODS ESCAPECHAR statement

STYLE function

Other features:

Data Set
ARIMA Procedure
FREQ Procedure

Details

Landmarks are a collection of references to well-known and or recurring components within the publication. Landmarks are commonly occurring navigational aids that can help readers access specific sections of the e-book quickly. In the following example, we create a landmark TOC and create links to start the book at chapter 2.

Program

ods escapechar='^';

ods epub3 file="landmarkTOC.epub" title="Navigate Using the Landmarks TOC"
options(creator='SAS R&D Staff'
linear_nav="landmarks"
contents="off"
chapter_type="frontmatter titlepage"
landmark="auto"
start="2");

title;

proc odstext;
   p "Navigate Using the Landmarks TOC"/style=systemtitle;
   p "by SAS R&D Staff"/style={just=c};
run;

ods epub3 newchapter=now options(landmark="Start Reading");

proc freq data=sashelp.class;
   tables sex*age/plots=freq;
run;

ods epub3 newchapter=now options(chapter_type="backmatter appendix" landmark="auto");

title "Appendix";

proc odstext;
   p "For more information,
visit the ^{style [url='http://support.sas.com'] SAS Support site}.";
run;

title;
ods epub3 close;
Program Description

Set the ODS inline formatting escape character

    ods escapechar='^';

Open the ODS EPUB3 destination. Open the ODS EPUB3 destination, provide a file for the EPUB output, and give the EPUB document a title. Use the OPTIONS suboptions to add the creator of the e-book and create a table of landmarks that are embedded in the e-book. Instead of using the regular TOC to navigate the e-book (CONTENTS="OFF"), use the landmark TOC instead. Create a chapter that looks like a title page (CHAPTER_TYPE=), and start in chapter 2.

    ods epub3 file="landmarkTOC.epub" title="Navigate Using the Landmarks TOC" options(creator='SAS R&D Staff' linear_nav="landmarks" contents="off" chapter_type="#frontmatter titlepage" landmark="auto" start="2");

Turn off system titles

    title;

Run PROC ODSTEXT to specify the chapter text.

    proc odstext;
      p "Navigate Using the Landmarks TOC"/style=systemtitle;
      p "by SAS R&D Staff"/style={just=c};
    run;

Define a second landmark. Define a landmark for the new chapter. Name the landmark TOC entry, “Start Reading”. Accept the default chapter type.

    ods epub3 newchapter=now options(landmark="Start Reading");

Run PROC FREQ.

    proc freq data=sashelp.class;
      tables sex*age/plots=freq;
    run;

Define a third landmark Define a third landmark for an appendix and determine the landmark from the chapter type.

    ods epub3 newchapter=now options(chapter_type="backmatter appendix" landmark="auto");

Supply a title for the Appendix TOC entry.

    title "Appendix";

Run PROC ODSTEXT Add text and a url to the appendix.

    proc odstext;
      p "For more information, visit the ^{style [url='http://support.sas.com'] SAS Support site}.";
    run;
Close the EPUB 3 destination and clear the title.

    title;
    ods epub3 close;

**Output**

The following shows the ODS EPUB3 statement used to add landmark TOC entries.

**Output 6.19**  ODS EPUB3 - Adding Landmark TOC Entries

---

**Example 3: ODS EPUB3 - Adding Figures and Figure Captions Using Events**

**Features:**
- ODS EPUB3 Statement
  - OPTIONS EVENT= with FIGURE, FIGCAPTION, IMAGE
  - TITLE= option
  - OPTIONS nonlinear suboption
  - STYLE= option
  - FILE=
- ODS TEXT Procedure
- TEMPLATE Procedure
- ODS ESCAPECHAR Statement
  - NEWLINE Function

**Other features:**
- TITLE Statement
- Macro Variables
Details

You can add figures and captions to your e-books.

Program

```sas
proc template;
    define style styles.mystyle;
        parent = styles.daisy;

        class figure /
            backgroundcolor = cxffff0f5
        ;
        class figurecaption /
            color = cx0000ff
        ;
    end;
run;

%macro figure1(image, caption);
    ods epub3 event=figure(start);
    ods epub3 event=figcaption(text=&caption);
    ods epub3 event=image(url=&image);
    ods epub3 event=figure(finish);
%mend;

/*-- macro to insert an image + caption into a chapter --*/

%macro figure2(image, caption);
    ods epub3 event=figure(start);
    ods epub3 event=image(url=&image);
    ods epub3 event=figcaption(text=&caption);
    ods epub3 event=figure(finish);
%mend;

%let TEXTURELOC = %sysfunc(getoption(textureloc));

%figure1(&TEXTURELOC\Table.png?alt_desc=Nonlinear table icon, "Figure 1. Nonlinear table icon")

proc odstext contents="
    p "Here is a figure with a top caption."/style=systemtitle{just=c};
run;
%figure1("&TEXTURELOC\Table.png?alt_desc=Nonlinear table icon", "Figure 1. Nonlinear table icon")

proc odstext contents="
    p **(newline)Here is a figure with a bottom caption.*
    /style=systemtitle{just=c};
run;
%figure2("&TEXTURELOC\Table2x.png?alt_desc=Nonlinear table icon", "Figure 2. Nonlinear table icon")
```
Program Description

Customize your styles Use PROC TEMPLATE to customize your output. Here we are specifying the daisy style for the EPUB output. Next the background color for the figure is specified (a pink color) and then the figure caption color (blue) is specified.

```sas
proc template;
define style styles.mystyle;
parent = styles.daisy;

class figure /
  backgroundcolor = cxfff0f5
  ;
class figurecaption /
  color = cx0000ff
  ;
end;
run;
```

Use macros to set up the figure and figure caption events We are using macros that allow us to pass in arguments that will create the figure, figure caption, and image events for our e-books.

```sas
%macro figure1(image, caption);
  ods epub3 event=figure(start);
  ods epub3 event=figcaption(text=&caption);
  ods epub3 event=image(url=&image);
  ods epub3 event=figure(finish);
%mend;

/*-- macro to insert an image + caption into a chapter --*/

%macro figure2(image, caption);
  ods epub3 event=figure(start);
  ods epub3 event=image(url=&image);
  ods epub3 event=figcaption(text=&caption);
  ods epub3 event=figure(finish);
%mend;
```

Use the TEXTURELOC system option to locate existing images and styles. Use the TEXTURELOC system option to specify the location of textures and images that are used by ODS styles. See “TEXTURELOC= System Option” in SAS System Options: Reference.

```sas
%let TEXTURELOC = %sysfunc(getoption(textureloc));
```
Set the ODS inline formatting escape character and close the system title.

title;
ods escapechar='^';

Open the ODS EPUB3 destination. Open the ODS EPUB3 destination, provide a file for the EPUB output, give the EPUB document a title, and specify a style for the book.

ods epub3 file="figure.epub"
   title="Figure and Caption Test"
   style=mystyle;

Add a figure and a top caption. Use the ODSTEXT statement to state what is happening when we add the figure with a figure caption. We are using the %FIGURE1 macro to embed the Table image and add a caption to the figure.

proc odstext contents="";
   p "Here is a figure with a top caption."/style=systemtitle{just=c};
run;
%figure1("&TEXTURELOC\Table.png?
   alt_desc=Nonlinear table icon",
   "Figure 1. Nonlinear table icon")

Add a figure and a bottom caption. Use the ODSTEXT statement to state what is happening when we add the figure with a bottom caption. We are using the %FIGURE2 macro to embed the Table2x image and add a caption to the bottom of the figure.

proc odstext contents="";
   p "\{newline\}Here is a figure with a bottom caption."
      /style=systemtitle{just=c};
run;
%figure2("&TEXTURELOC\Table2x.png?
   alt_desc=High resolution (2x) nonlinear table icon",
   "Figure 2. High resolution (2x) nonlinear table icon")

Close the EPUB 3 destination and clear the title.

   title;
   ods epub3 close;

Reset the styles template

   proc template;
      delete styles.mystyle;
   run;

Output

The following shows the ODS EPUB3 statement where figure captions are added to the figure.
Here is a figure with a top caption.

Figure 1. Nonlinear table icon

Here is a figure with a bottom caption.

Figure 2. High resolution (2x) nonlinear table icon

**Example 4: ODS EPUB3 - Adding Audio Using Inline Functions**

**Features:**
- ODS EPUB3 Statement
  - FILE= option
  - NEWCHAPTER= option
  - OPTIONS nonlinear suboption
  - EVENT= option
- ODSTEXT Procedure
- ODS ESCAPECHAR Statement
  - STYLE Function
  - IMAGE Function
  - AUDIO Function

**Other features:**
- TITLE Statement

**Details**

Audio can be added to an e-book using ODS EPUB and ODS EPUB3 statements by using either the ODS ESCAPECHAR AUDIO function within the ODSTEXT procedure or by using the ODS Report Writing Interface. Here is an example of using the ODS ESCAPECHAR statement to add audio to your e-book.

**Program**

```plaintext
ods epub3 file="AudioEscapechar.epub"
  newchapter=now
  options(nonlinear="chapter");
ods escapechar='^';
title2 "You can embed Audio into your eBook";
title3 "Use the Inline Formatting AUDIO Function";
```
Program Description

**Open the ODS EPUB3 destination.** Open the ODS EPUB3 destination, provide a file for the EPUB output, and create a new chapter.

```sas
ods epub3 file="AudioEscapechar.epub"
newchapter=now
options(nonlinear="chapter");
```

**Set the ODS inline formatting escape character.**

```sas
ods escapechar='^';
```

**Add a title when the procedure runs that describes what is being added to the e-book**

```sas
title2 "You can embed Audio into your eBook";
title3 "Use the Inline Formatting AUDIO Function";
```

**Add audio and an image to the e-book.** Use the ODS ESCAPECHAR AUDIO function with PROC ODSTEXT to add audio to your e-book.

```sas
proc odstext contents="";
p "(audio SAS02_Orchestral30.mp3?controls=controls)"/style={just=c};
run;
```

**Label the audio file and add it to the table of contents.** Label the audio file and add it to the table of contents.

```sas
ods epub3 options(nonlinear="none");
ods epub3 event=branch(start label="Embedded Audio File (Audio)"
url="chapter2.html");
ods epub3 event=branch(finish);
```

**Close the EPUB3 destination and clear the title.**

```sas
ods epub3 close;
```

**Output**

The following output shows an image of the e-book created that has embedded audio.
Example 5: ODS EPUB3 - Adding Video Using Inline Functions

Features:
- ODS EPUB3 Statement
  - FILE= option
  - NEWCHAPTER= option
  - TITLE= option
- ODS PROCLABEL statement
- ODS TEXT Procedure
- ODS ESCAPECHAR Statement
  - STYLE Function
  - NOTEREF Function
  - VIDEO Function

Other features:
- TITLE Statement

Details
Video can be added to an e-book using ODS EPUB and ODS EPUB3 statements by using either the ODS ESCAPECHAR AUDIO function within the ODSTEXT procedure or by using the ODS Report Writing Interface. Here is an example of using the ODS ESCAPECHAR statement to add video to your e-book.

Program
```
title;
ods epub3 file="VideoEscapechar.epub"
title="Marmot Sighting eBook"
    newchapter=now;
ods escapechar='^';
ods proclabel = "Marmot Sighting (Video)";
proc odstext contents="";
p "^\{video multimedia/marmot.m4v?controls=controls;
    poster=multimedia/
```
No vacation is complete without home video.
If you enjoy wildlife, check out the marmot
^{noteref large mountain-dwelling ground squirrel}
we saw while driving the
"^[style [url='http://en.wikipedia.org/wiki/Trail_Ridge_Road']
Trail Ridge Road].";
run;
ods epub3 close;
Output

The following output shows an image of the e-book created that has embedded video. The first page shows the table of contents and links to the video. The second page shows the video.

Output 6.22  ODS EPUB3 - TOC of the E-book Created By Adding Audio Using ODS ESCAPECHAR Function

Output 6.23  ODS EPUB3 - Embedded Video Using the ODS ESCAPECHAR Function

Example 6: ODS EPUB3 - Adding Audio Using RWI
Features:
- ODS EPUB3 Statement
FILE= option
TITLE= option

RWI Audio Method

Details
Audio can be added to an e-book using ODS EPUB and ODS EPUB3 statements by using either the ODS ESCAPEHCAR AUDIO function within the ODS\TEXT procedure or by using the ODS Report Writing Interface. Here is an example of using the Report Writing Interface AUDIO method to add audio to your e-book.

Program
ods epub3 file="c:\your-directory\AudioRWITest.epub"
   title="RWI Audio eBook" newchapter=now;
   title "Adding Audio to EPUB Output";

data _null_;
dcl odsout obj();

   obj.format_text(data: "You can embed Audio into your eBook. ",
                   data: " Use the RWI Audio Method.",
                   style_elem: "SystemTitle");

   filename AudFile url "c:\your-directory\SAS02_Orchestral30.mp3";
   obj.audio(file:"fileref:AudFile",
              type: "mp3",
              preload: "auto",
              autoplay: "off",
              loop: "no"
   );
run;
ods epub3 close;

Program Description

Open the ODS EPUB3 destination. Open the ODS EPUB3 destination, provide a filename for the EPUB output, create a title for the e-book, and add a title to provide information about the purpose of this e-book.

ods epub3 file="c:\your-directory\AudioRWITest.epub"
   title="RWI Audio eBook" newchapter=now;
   title "Adding Audio to EPUB Output";

Add information to the e-book. Use the format_text method to add title information to the e-book.

data _null_; 
dcl odsout obj();

   obj.format_text(data: "You can embed Audio into your eBook. ",
                   data: " Use the RWI Audio Method.",
                   style_elem: "SystemTitle");
Use the RWI AUDIO method to embed audio into the e-book. We are embedding an orchestral audio clip that is an mp3 format.

```sas
filename AudFile url "c:\your-directory\SAS02_Orchestral30.mp3";
obj.audio(file:"fileref:AudFile",
  type: "mp3",
  preload: "auto",
  autoplay: "off",
  loop: "no"
); run;
```

Close the EPUB3 destination.

```sas
ods epub3 close;
```

Output

The following output shows an image of the e-book created that has embedded audio. There is simply an audio file embedded by using the RWI AUDIO method.

**Output 6.24**  
**ODS EPUB3 - Adding Audio Using RWI AUDIO Method**

Adding Audio to EPUB Output

You can embed Audio into your eBook. Use the RWI Audio Method.

---

**Example 7: ODS EPUB3 - Adding Video Using RWI**

Features:
- ODS EPUB3 Statement
  - FILE= option
  - TITLE= option
- RWI Audio Method
Details

Audio can be added to an e-book using ODS EPUB and ODS EPUB3 statements by using either the ODS ESCAPEHCAR AUDIO function within the ODSTEXT procedure or by using the ODS Report Writing Interface. Here is an example of using the Report Writing Interface AUDIO method statement to add audio to your e-book.

Program

```ods epub3 file="CatVideoRWI.epub" title="RWI Video eBooK";
title;
data _null_;  
dcl odsout obj();
    obj.video(file:"http://www.elizabethcastro.com/epub/examples/catbox.mp4",  
        type:"mp4",  
        poster:"poster.jpg",  
        width:"382px",  
        height:"287px"  
    );
run;
ods epub3 close;
```

Program Description

**Open the ODS EPUB3 destination.** Open the ODS EPUB3 destination, provide a filename for the EPUB output, create a title for the e-book. Suppress the system titles

```ods epub3 file="CatVideoRWI.epub" title="RWI Video eBooK";
title;
```

**Use the RWI VIDEO method to embed audio into the e-book.** We are embedding a video clip of two kitties playing.

```data _null_;  
dcl odsout obj();
    obj.video(file:"http://www.elizabethcastro.com/epub/examples/catbox.mp4",  
        type:"mp4",  
        poster:"poster.jpg",  
        width:"382px",  
        height:"287px"  
    );
run;
```

**Close the EPUB3 destination.**

```ods epub3 close;
```

Output

The following output shows an image of the e-book created that has embedded video. This output was created using the RWI VIDEO method.
Example 8: ODS EPUB3 - Adding Mathematical Equations Using ODS ESCAPECHAR

Features:
- ODS EPUB3 Statement
  - FILE= option
  - TITLE= option
- ODSTEXT Procedure
- ODS ESCAPECHAR Statement
  - MATHML Function

Other features:
- TITLE Statement

Details

MATHML can be added to ODS EPUB3 statements by using the MATHML function within the ODSTEXT procedure. Here is an example of using the ODS ESCAPECHAR Inline function, MATHML, to add MathML to your e-book.

Program

```sas
title;
ods epub3 file="mathml.epub" title="MathML Test";
ods escapechar='^';
proc odstext contents="";
p 'Summation:';
p '{mathml <math xmlns="http://www.w3.org/1998/Math/MathML"> <mrow>
```
Program Description

Open the ODS EPUB3 destination. Open the ODS EPUB3 destination, provide a file for the EPUB output, create a title for the e-book, and suppress the system title.

```ods epub3 file="mathml.epub" title="MathML Test";```

Set the ODS inline formatting escape character.

```ods escapechar='^';```

Add mathematical equations to the e-book. Use the ODS ESCAPECHAR MATHML function with PROC ODSTEXT to add mathematical equations to your e-book. In this case, we added a summarization math equation.

```proc odstext contents="";
p 'Summation:';
p '^
<math xmlns="http://www.w3.org/1998/Math/MathML">
<mrow>
  <munderover>
    <mo>&#x02211;</mo>
    <mrow>
      <mi>n</mi>=
      <mn>1</mn>
    </mrow>
    <mrow>
      <mn>5</mn>
    </mrow>
  </munderover>
  <msup>
    <mrow>
      <mi>n</mi>
    </mrow>
    <mrow>
      <mn>2</mn>
    </mrow>
  </msup>
</math>'';
run;
```

ods epub3 close;
Close the EPUB3 destination.

ods epub3 close;

Output

The following output shows a summary equation generated by using the ODS ESCAPECHAR function MATHML.

Output 6.26  ODS EPUB3 - Adding Equations Using ODS ESCAPECHAR MATHML Function

Summation:

$$\sum_{n=1}^{5} n^2$$

ODS ESCAPECHAR Statement

Defines a representative character to be used in output strings.

Valid in: Anywhere

Category: ODS: Output Control
Restrictions: Affects all open destinations except for the LISTING destination.
SAS/GRAPH does not support inline formatting.

Examples: “Example 1: Basic Inline Formatting Functions” on page 285
“Using the Page Method” in SAS Output Delivery System: Advanced Topics
“Creating a Flyer with Text Methods” in SAS Output Delivery System: Advanced Topics
“Example 4: ODS EPUB3 - Adding Audio Using Inline Functions” on page 259
“Example 5: ODS EPUB3 - Adding Video Using Inline Functions” on page 261
“Example 8: ODS EPUB3 - Adding Mathematical Equations Using ODS ESCAPECHAR” on page 267

Syntax

ODS ESCAPECHAR='escape-character';

Required Argument

escape-character

specifies the special character that identifies the inline formatting symbol. The escape-character should be one of the following rarely used characters: @, ^, or \

With the ODS ESCAPECHAR statement, you can define an escape character for use with the inline formatting functions. These functions provide the ability to enhance and interpret text strings that are used by statements and variables. You can use these functions to modify text strings within table cells and title and footnotes.

For a complete list of the inline formatting functions and a detailed description for each, see “Using the ODS ESCAPECHAR Functions” on page 274.

Notes

For RTF output, the -, *, or # can also be used. The \ is a special RTF character. Therefore, it is recommended that you use an escape character other than \ for RTF output.

There is no default value for the escape character, but you can use the special escape sequence (*ESC*) in the same way as an escape character. The special escape sequence (*ESC*) cannot be used in nested functions.

Details

Basic Inline Formatting

Inline formatting functions enable you to change styles in titles and footnotes, text strings, and table cells. For example, you can perform the following formatting functions.

• insert subscript or superscript into SAS output
• underline text
• justify text
• insert page X of Y numbers into output
• insert line feeds into long text strings
• insert destination-specific, raw text into HTML or RTF output
Existing style elements and style attributes can be used with the STYLE functions. See the explanation of using styles in “Inline Style Attributes and Nesting” on page 271.

**Inline Style Attributes and Nesting**

You can use the ODS ESCAPECHAR statement and inline formatting syntax to justify text or change the color of titles, footnotes, and text. Other styles or attributes that are useful and that can add emphasis are font size, underlining, overlining, and strikethrough.

You can also change the styles. For example, ODS PRINTER now supports two style settings for underlining. ODS PRINTER recognizes the SAS/GRAPH syntax UNDERLIN=1,2,3 for underlining text. However, this option does not change the thickness of the line. Other style attributes that are useful and that can add emphasis are font size, underlining, overlining, and strikethrough. See the section on style attributes and style attribute values in “Detailed Information for All Style Attributes” in *SAS Output Delivery System: Procedures Guide* for more details about these style attributes. See the global title style options in “TEMPLATE Procedure: Creating a Style Template” in *SAS Output Delivery System: Procedures Guide*. See the global footnote style options in “FOOTNOTE Statement” in *SAS Statements: Reference*.

Nested inline formatting is also supported. This feature enables you to set several style attributes for a string without resetting the previously used style. You can start a string with one set of style attributes and add to them later in the string. Nested inline formatting has the following syntax:

```latex
^{style <style-element-name><[style-attribute-specification(s)]> formatted text}
```

The syntax begins with the function name STYLE. Next you can add a style element such as Headerfixed or SystemTitle, as needed. Then you can add new attributes such as FONTSTYLE= or COLOR=, within brackets. The syntax ends with the text that you want to format. The following code is an example of nested formatting for RTF output:

```sas
title "test of ^{super ^{style [color=red] red ^{style [color=green] green} and ^{style [color=blue] blue }formatting }} and such" ;
```

In the example code, the `^{super<text>` is invoked to start using the superscript function. Then the style function is used to add another style attribute, `^{style [color=red]<text>` for your text.

To understand this nesting, think of a stack in which the first item that is placed on the stack is the last item to come off of the stack (FILO). The superscript function is pushed onto the stack first. Then the style function is used to push a red color onto the stack, resulting in a text string that is superscripted and red in color. Next, the green color is pushed onto the stack. Because the new style attribute is a color, the text changes to the new color. Continuing the string processing, this style attribute is then closed with a close bracket. After the green color is closed or popped off the stack, the red color becomes the active style attribute for the text. Next the blue color is pushed onto the stack, and the text string uses that color. After the blue color is closed, the red and superscript are closed. Now the stack is empty, so ODS uses the default style attributes to finish processing the text string.

**Note:** Each output destination has limitations. When you use the PRINTER destination, you can only nest styles. The SUB and SUPER functions cannot be nested with the STYLE function. However, the HTML and RTF destinations can nest the STYLE function with the SUB and SUPER functions.

**Note:** For information of style elements see Chapter 11, “Style Elements,” on page 965.
Using Unicode Symbols

ODS now supports the ability to incorporate Unicode symbols such as Greek symbols into your output. The new inline formatting function is UNICODE, and the syntax is \(^\text{unicode < value>}\). The syntax is similar to other inline formatting functions in that you can use a four-digit Unicode value that is predefined in a list. Another way to use the UNICODE function is to redefine a list that is stored as a tagset. See “Concepts: Markup Languages and the TEMPLATE Procedure” in SAS Output Delivery System: Procedures Guide for information about tagsets.

There is a new tagset that contains a predefined list of common Greek symbols and their Unicode values. You can update this template as needed. The template increases the flexibility of the new inline style function. You can still use existing inline style functions, \(^\text{DAGGER}\) and \(^\text{SIGMA}\).

To find out what symbols are available to you for Windows XP, you can select Start ⇒ Programs ⇒ Accessories ⇒ System Tools ⇒ Character Map. The window that appears shows you the font and all of the symbols available for a given font. For the font displayed, you can highlight a symbol and see the Unicode value for that symbol. The Unicode value is displayed at the bottom of the Character Map window. You can use that Unicode value in the argument to the UNICODE function. For example, Unicode value 216b displays a Roman Numeral 12. The following code illustrates the use of this value:

```sas
title 'Roman Numeral twelve is ^{unicode 216b}';
```

The Base.Template.Tagsets tagset contains a table of Unicode values and their mnemonics. To add or change a mnemonic, you must SOURCE the tagset, make the desired changes, and then run the modified tagset. The following code creates a file called core.tpl in the current directory that contains the Base.Template.Tagsets tagset also:

```sas
proc template;
  source base.template.tagset / file="core.tpl";
run;
```

If you open the core.tpl file, you notice some text that appears similar to the following text:

```sas
set $unicodeMap["ALPHA"] "03B1";
set $unicodeMap["BETA"] "03B2";
set $unicodeMap["DAGGER"] "2020";
```

To update the file with a new mnemonic and corresponding Unicode value, use the following syntax and add it to the file:

```sas
set $unicodeMap["<new function name>" ] "<unicode value>";
```

Save the file and compile the modified tagset using PROC TEMPLATE as follows:

```sas
proc template;
  %inc "core.tpl";
run;
```

The modified tagset is stored in the first writable template store in your ODS path. See “Concepts: Markup Languages and the TEMPLATE Procedure” in SAS Output Delivery System: Procedures Guide for more information about PROC TEMPLATE and using tagsets.

A new registry setting holds the Unicode font value. You can change this Unicode font value to any valid font that is installed on your computer and recognized by SAS. For specific details about how to change your font, see “Changing SAS Registry Settings for ODS” on page 40. For information about all the new True Type fonts available in SAS 9.2, see “Using Fonts with Universal Printers and SAS/GRAPH Devices” in SAS.
Language Reference: Concepts. This chapter contains information about how to install the TrueType fonts on your computer, too.

**Inline Formatting with the PUT Statement**

Inline formatting information that is used in the PUT statement is counted as printed space that is needed for the ODS LISTING output destination. Therefore, the LINESIZE= system option might need to be set to prevent wrapping of the output. For example, the inline formatting information shown in the following code defines the font size, font face, and font weight for the 'AAA' and 'BBB' values.

```plaintext
ods escapechar="^";
ods html file='file.html';
ods pdf file='file.pdf';
ods rtf file='file.rtf';
data _null_;
  file print ls=200;
  put @1 '^{style [fontsize=8pt font_face=courier font_weight=bold] AAA}';
  put +5 '^{style [fontsize=8pt font_face=courier font_weight=bold] BBB}';
run;
ods _all_ close;
```

The line size needed for printed output is three characters. However, the inline formatting information is also counted as part of the line size, even though that count only affects the appearance of the output. This increased line size can cause the text to wrap if it exceeds the current value of the LINESIZE= system option.

To prevent wrapping in the ODS LISTING output, increase the value of the LINESIZE= system option or decrease the font size.

**Inline Formatting with ODS Statistical Graphics**

ODS Statistical Graphics include template-based procedures (SGPLOT, SGPANEL, and SGSCATTER) and some statements that support ODS ESCAPECHAR in conjunction with the UNICODE, SUB, and SUP inline formatting functions. For information about how to use these functions with ODS Statistical Graphics, see *SAS ODS Graphics: Procedures Guide* and *SAS Graph Template Language: User's Guide*.

**Interpreting Inline Formatting Output Strings**

ODS ESCAPECHAR controls the interpretation of output strings by ODS, except for the LISTING, the OUTPUT, and the DOCUMENT destinations. Whenever ODS destinations encounter the specified character in their output, regardless of the source of the output, ODS interprets the character as a special "escape" character that enables special formatting options. For example, the following program produces output in italics.

```plaintext
data italic;
  x='This font is ^{style[fontstyle=italic]italic}.';
  output;
  run;
ods pdf file="italicFont.pdf";
ods escapechar="^";
proc print data=italic;
  run;
ods _all_ close;
```
Using the ODS ESCAPECHAR Functions

This section describes the use of the defined ODS ESCAPECHAR character value to perform inline formatting. After you define the ODS ESCAPECHAR character value, you can use the available inline formatting functions to change the style within cells, text, titles, and footnotes. You can use these functions to insert page numbers, line feeds, destination-specific raw data, additional spaces and formatting into your output. You can use the functions to add multimedia (audio, video, images) to EPUB3 e-books.

Note: For traditional RTF output, you must use Print Preview to view your output. Some of the formatting will not show up in the SAS results viewer window.

The inline functions available are shown in Table 6.4 on page 274. Here is the syntax for the ODS ESCAPECHAR functions:

escape-character {function-name <arg-1 <arg-2 <arg-n>>> }

CAUTION:
A space between the escape character and left bracket of the inline formatting style function can produce undesired results. Here is an example of how the code should look:

^{style [color=green] title green};

escape-character
is the character defined using the ODS ESCAPECHAR statement.

{ }
enclose the inline formatting grouping characters.

arg-1, arg-n
are arguments that are given to the function. The number of arguments depends on the function. Some functions have no arguments.

function-name
is the name of the inline formatting function.

Table 6.4 Valid Functions That Can Be Used with ODS ESCAPECHAR

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Argument</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUDIO on page 275</td>
<td>Filename</td>
</tr>
<tr>
<td></td>
<td>Fileref</td>
</tr>
<tr>
<td></td>
<td>Web Address</td>
</tr>
<tr>
<td>BOLD on page 276</td>
<td>Text</td>
</tr>
<tr>
<td>DAGGER on page 276</td>
<td>None</td>
</tr>
<tr>
<td>DATE on page 276</td>
<td>None</td>
</tr>
<tr>
<td>DEST on page 277</td>
<td>OUTPUT destination</td>
</tr>
<tr>
<td>EMPHASIS on page 277</td>
<td>Text</td>
</tr>
<tr>
<td>IMAGE on page 277</td>
<td>Filename</td>
</tr>
<tr>
<td></td>
<td>Fileref</td>
</tr>
<tr>
<td>ITALIC on page 278</td>
<td>Text</td>
</tr>
</tbody>
</table>
**AUDIO Function**

\[^{\text{AUDIO}} \text{filename} | \text{fileref} | \text{web-address}; <\text{options-list}> \]^{\text{}}

includes an audio in an e-book created by using the ODS EPUB3 destination. Use a filename, a fileref, or a web address. ODS EPUB3 supports the MP3, MP4, M4A, and WAV audio formats. Apple uses the M4A format in its products and services. The M4A audio format in iBooks provides an excellent user experience.

*filename*

is the name and optional path of an external audio file.
fileref

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

web-address

is the web address of an external audio file.

<options-list>

The options specified in the options-list are appended to the AUDIO function using a ‘?’ character followed by the options. Separate the options with a ‘;’ or ‘&’ character.

The following are supported options that can be applied to the AUDIO function:

- AUTOPLAY: The audio automatically starts playing without stopping.
- CONTROLS: The audio controls should be displayed.
- LOOP: The audio starts over again, every time it is finished.
- MUTED: The audio output should be muted.
- PRELOAD: Specifies if and how the audio should be loaded.

See W3C HTML Audio/Video Properties for more options that you can use with in the options-list.

Restriction

This function is use with the ODS EPUB and EPUB3 destinations.

Requirement

You must enclose filename and fileref in quotation marks.

Tip

You can use this function only with ODS EPUB3 destination.

Example

Example 4: ODS EPUB3 - Adding Audio Using Inline Functions

BOLD Function

^\{BOLD text\}

used to emphasize keywords and semantics for Assistive Technologies. This function provides no voice or mood change.

text

is text that you want bold in your e-book. An example is:

^\{bold 'Your text string'};

Tip

You can use this function only with ODS EPUB destinations.

DAGGER Function

^\{DAGGER\}

produces the Greek dagger sign.

Tip

It is preferable to use the UNICODE function to generate a dagger sign.

DATE Function

^\{DATE\}
inserts the RTF to express the date.

Tip You can use this function only with the TAGSETS.RTF destination.

**DEST Function**

```
^{DEST <[output-destination]> text
```

*output-destination*

is one of the ODS output destinations, RTF, printer family, HTML, HTML5, or EPUB. This is the destination that is used by the inline formatting function.

Tip You can specify more than one *output-destination*.

*text*

is text that you want to write to the output. An example is:

```
^{dest [rtf html] ^(raw rawtext string) }
```

**EMPHASIS Function**

```
^{EMPHASIS 'text'}
```

used for vocal stress. This function is used for Assistive Technologies.

*text*

is text that you want emphasized in your e-book. An example is:

```
^{emphasis 'Your text string'}
```

Tip You can use this function only with ODS EPUB destinations.

**IMAGE Function**

```
^{IMAGE filename | fileref; <options-list>}
```

includes an image in an e-book created by using the ODS EPUB3 destination. ODS EPUB3 supports PNG, GIF, JPEG, and SVG image types. Use JPEG for photographs and SVG for small-footprint, scalable performance.

*Note:* Putting too many SVG images in your e-book slows down iBooks noticeably, because it has to draw each image based on its vector graphics definition.

*filename*

is the name and optional path of an external image file.

*fileref*

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

*<options-list>*

The IMAGE function accepts the options WIDTH, HEIGHT, and ALT, DESC, as well as CSS properties such as FLOAT. The function scripts an HTML IMG tag in the internal HTML5 file. See W3C HTML `<img>` tag Documentation for more options that you can use.

The options specified in the *options-list* are appended to the IMAGE function using a ‘?’ character followed by the options. Separate the options with a ‘;’ or ‘&’ character.

Use percentages when supplying the image width or height. This helps ensure a consistent user experience across e-book reader software and hardware platforms.
Always provide informative alternate text using the ALT_DESC option. The image might not be accessible to all of your readers, so additional context is required.

**Note:** iBooks zooms an image if you double-tap it. To prevent pixilation, raster images should be at least 1.5 times the intended viewing size up to a maximum of 3.2 million pixels.

**Tip** You can use this function only with the ODS EPUB3 destination.

**Example**
```
ods epub3 file="ImageeBook.epub"
   title="My Trip to Bhutan in eBook";
ods escapechar='^';
proc odstext contents="";
p "Visit Bhutan"/style={just=c};
p "^[image Bhutan.jpg?width=75%;alt_desc=Bhutan]" /style={just=c};
ods epub3 close;
```

**ITALIC Function**

`^[ITALIC 'text']`

used to indicate an alternative voice or mood. This function is also used for semantics such as identifying names. This function is used for Assistive Technologies.

`text` is text that you want italicized in your output. For example:

`^[italic 'Your text string']`;

**Tip** You can use this function only with ODS EPUB destinations.

**LASTPAGE Function**

`^[LASTPAGE]`

inserts the total number of pages.

**Restrictions**

Do not use the LASTPAGE function with images for PDF output.

Do not use the LASTPAGE function with the table of contents option (CONTENTS=YES) for ODS PDF.

**Tips**

This function works with the RTF and TAGSETS.RTF destinations.

You must use Print Preview to view the resolved LASTPAGE function output that is generated by the TAGSETS.RTF destination.

**Example**

“Example 2: Customizing Titles and Page Numbers” on page 289

**LEADERS Function**

`^[LEADERS <string>]`
string

is the string that is repeated to fill the space between the leading text and the
following text. This function is often used when generating a table of
contents. Example code is:

PostText = "^{leaders . }^{tocentrypage}

Tip You can use this function only with the PRINTER destination.

MATHML Function

^{MATHML <MathML-presentation-markup>}

MathML-presentation-markup

supports accessibility features for MathML such as large print with the ability
to drill down on terms for low vision users, voiceover that speaks the math,
and voiceover that displays the math on a refreshable braille display using the
Nemeth braille code. This function is used with Assistive Technology.
iBooks is built with WebKit. WebKit's support for MathML is documented at
WebKit's MathML Support Documentation.

Restriction This function is used with the EPUB3 destination.

Notes The MATHML function accepts the MathML itself, not a file
containing the MathML.

Math must be encoded using "presentation MathML" to be
rendered using the features above.

Example

```
p \^{mathml<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <your math goes here>
  </mrow>
</math>};
```

Example “Example 8: ODS EPUB3 - Adding Mathematical Equations
Using ODS ESCAPECHAR” on page 267

NBSPACE Function

^{NBSPACE <number>}

number

is the number of non-breaking spaces that you want to insert. A single space
is inserted if you do not specify a number argument.

Default The NBSPACE value defaults to 1. A single space is inserted if a
number is not specified.

See “Example 1: Basic Inline Formatting Functions ” on page 285

NEWLINE Function

^{NEWLINE <number>}

number

is the number of line breaking characters that you want to insert.

Default The newline value defaults to 1. A single line break is inserted if
you do not specify a number.
See “Example 1: Basic Inline Formatting Functions ” on page 285

NOTEREF Function

declares text that iBooks displays in a pop-up footnote. By default the footnote marker is “1”, “2”, and so on. You can customize the marker with the TEXT option.

\{NOTEREF "<text>"\}

text

is the text that you want to see when you tap the number that has been applied to the note.

See “Example 1: Basic Inline Formatting Functions ” on page 285

Example

proc odstext contents="";
  p "  ^{\{video multimedia/marmot.m4v?controls=controls; poster=multimedia/marmot.jpg;\}
    Check out the marmot
    ^{noteref large mountain-dwelling ground squirrel}
    we saw while driving the
    ^{style [url='http://en.wikipedia.org/wiki/Trail_Ridge_Road']
      Trail Ridge Road}."
  run;

Example “Example 5: ODS EPUB3 - Adding Video Using Inline Functions” on page 261

PAGEOF Function

\{PAGEOF\}

inserts RTF syntax to express all the controls for Page X of Y.

Tips

You can use the PAGEOF function in the TITLE and FOOTNOTE statements. However, if the BODYTITLE option is also specified with the ODS RTF statement, the titles and footnotes are removed from the header and footer sections of the RTF file. As a result, the "page of" information is not written as expected. If the desired location of the page numbering is in the title or the footnote, remove the BODYTITLE option.

When the \ character is specified as the ODS ESCAPECHAR character, the PAGEOF function is not interpreted properly for the TAGSETS.RTF destination. Instead, specify a different escape character.

You can use the PAGEOF function only with the RTF and TAGSETS.RTF destinations. You must use Print Preview to view the resolved PAGEOF function output that is generated by the TAGSETS.RTF destination.

PDF Function

\{PDF filename | fileref\}

used to embed a PDF file into an e-book. iBooks supports embedded PDF files.

You must specify an image file in the syntax of the PDF inline formatting function. The image appears in the linear order of the e-book. The image links to the PDF, which is not in the linear order of the e-book.

filename

is the name and optional path of an external PDF file.
fileref

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

**Tip**

You can use this function only with ODS EPUB destinations.

**Example**

```
proc odsотовtext;
  p "Embed a PDF file:
    ^[pdf class.pdf?image=selectric.jpg; width=50%]*;
run;
```

**RAW Function**

```
^[RAW <string>]
```

**string**

is inserted directly without translation. This function enables you to insert control characters. This function works for markup destinations such as HTML and RTF.

**Restriction**

The RAW function does not work with PDF or Windows drivers.

**Tips**

After this function has been turned on in a session, you cannot turn it off for that session.

The \ is a special RTF character. The / and / are special function characters. When you use these special characters with the RAW function, ODS might generate unexpected output.

**See**

“Example 1: Basic Inline Formatting Functions” on page 285

**SIGMA Function**

```
^[SIGMA]
```

generates the Greek SIGMA sign σ.

**Tip**

The preferable way to produce a SIGMA sign is to use the UNICODE function.

**See**

“Example 1: Basic Inline Formatting Functions” on page 285

**STRONG Function**

```
^[STRONG 'text']
```

used to convey importance. This function is used for Assistive Technologies.

**text**

is text that you want to be bold in your e-book. For example:

```
^[strong 'Your text string'];
```

**Tip**

You can use this function only with ODS EPUB destinations.

**STYLE Function**

```
^[STYLE <style-element-name>[<style attribute-specification>] formatted text}
```

**style-element-name**

specifies the style element. For the STYLE function, you can use the same format that is available for the STYLE= option in all of the templates. For example:
^\{style rowheader [color=red] my text\};

or

^\{style rowheader my text\};

See Chapter 11, “Style Elements,” on page 965

style-attribute-specification

specifies the style attribute. For the STYLE function, you can use the same format that is available for the STYLE= option in all the templates. For example:

^\{style [color=red] my text\};

Tip

When specifying border style or border color attributes, you might also need to specify a border width attribute to override the style in the ODS destination. For example, when you specify BORDERRIGHTSTYLE= or BORDERRIGHTCOLOR=, also specify BORDERRIGHTWIDTH=.

See For a list of style attributes and their values, see “Style Attributes Tables” on page 993.

Example “Example 2: Customizing Titles and Page Numbers” on page 289

formatted-text

specifies the text to which to apply the styles.

Notes

The style attributes or elements remain in effect until they are overridden by another style. A default style can also reset the style. The following code illustrates that the bolded text style is reset by the sub functions default style:

```ods pdf text='^\{style [fontweight=bold] BOLDED\} ^\{sub a\} NOT bolded\}'```

You can nest inline styles. The following example illustrates nesting inline styles. For an explanation of nesting styles, see “Inline Style Attributes and Nesting” on page 271.

SUB Function

^\{SUB <subscript-character>\}

`script-value`

can be a numeric, alphanumeric, or a character value. This value is written below and immediately to one side of another character.

Restrictions

Microsoft Word honors only one level of subscript for RTF and TAGSETS.RTF.

The PRINTER destination does not recognize nesting of the SUB function. The `script-value` must immediately follow the SUB function.

See “Example 1: Basic Inline Formatting Functions ” on page 285

SUPER Function

^\{SUPER <superscript-value>\}
superscript-value can be a numeric, alphanumeric, or a character value. This value is written above and immediately to one side of another character.

Restrictions Microsoft Word only honors one level of subscript for RTF and TAGSETS.RTF.

Nesting of the SUPER function is not recognized by the PRINTER destination. The subscript-value must immediately follow the SUPER function.

See “Example 1: Basic Inline Formatting Functions” on page 285

THISPAGE Function
\[^{\text{THISPAGE}}\]
inserts the current page number.

Tips This function can be used only with the PRINTER, RTF, and TAGSETS.RTF destinations.

You must use Print Preview to view the resolved THISPAGE function output that is generated by the TAGSETS.RTF destination.

Example “Example 2: Customizing Titles and Page Numbers” on page 289

TOCENTRYINDENT Function
\[^{\text{TOCENTRYINDENT} \langle \text{len} \rangle}\]

len
is the amount to be indented per level. Example code is:

PreText = "\[^{\text{tocentryindent 2em}}\]"

Tip This function can be used only with the PRINTER destination.

TOCENTRYPAGE Function
\[^{\text{TOCENTRYPAGE}}\]
is the page number of the current TOC entry. Example code is:

PostText = "^{leaders . } ^{tocentrypage}\"

Tip This function can be used only with the PRINTER destination.

UNICODE Function < | >
\[^{\text{UNICODE} \langle \text{unicode-value} | \text{unicode-value}'X' \rangle}\]

unicode-value
can be an actual four-place hexadecimal Unicode value or one of the names listed in the Base.Template.Tagsets template. For example, 03B2 is the Unicode value for the Alpha symbol. For details about Unicode values, see “Using Unicode Symbols” on page 272.

Default Times New Roman Unicode Uni is the default font used in the inline style Unicode function for the PDF destination.

Restriction SAS Unicode fonts Arial Unicode MS and Times New Roman Uni do not support some supplementary characters when used with ODS PDF.
'unicode-value'X

is syntax used with ODS Statistical Graphics. A hexadecimal value is enclosed in single or double quotation marks followed by an X. The X specifies that the value in quotation marks is a hexadecimal value. This quoted value must be an actual four-place hexadecimal Unicode value or one of the names listed in the Base.Template.Tagsets template. For example, 03B2 is the Unicode value for the Alpha symbol. For details about Unicode values, see “Using Unicode Symbols” on page 272.

Tips

Thorndale Duospace WT J is the default font used in the inline style Unicode function for the PDF destination.

The unicode-value can be enclosed with single or double quotation marks.

See

For information about how to use this function with ODS Statistical Graphics, see SAS ODS Graphics: Procedures Guide and SAS Graph Template Language: User's Guide

VIDEO Function

`\{VIDEO filename | fileref ; <options-list>\}

includes a video in an e-book created by using the ODS EPUB3 destination.

filename | fileref

ODS EPUB3 supports the M4V, MP4, and QuickTime video formats.

filename

is the name and optional path of an external video file.

fileref

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

web-address

is the web address of an external video file.

<options-list>

The options specified in the options-list are appended to the VIDEO function using a ‘?’ character followed by the options. Separate the options with a ‘;’ or ‘&’ character.

The following are supported options that can be applied to the VIDEO function:

- **AUTOPLAY**: The video automatically starts playing without stopping.
- **CONTROLS**: The video controls should be displayed.
- **LOOP**: The video starts over again, every time it is finished.
- **MUTED**: The video output should be muted.
- **POSTER**: The poster specifies a poster image that is displayed by the video control prior to playback. iBooks requires a poster image. Commonly the poster image is created from a frame from the associated video.
- **PRELOAD**: Specifies if and how the video should be loaded.
- **CSS Attributes**: Applies the CSS values like FLOAT, MARGINTOP, MARGINRIGHT attributes.
Example attributes: Applies to the POSTER image. It is probably best to specify Height and Width attributes as pixels for best use in iBooks.

See W3C HTML Audio/Video Properties for more options that you can use with in the options-list. Also refer to W3C HTML <img> tag Documentation for information related to the images used for POSTER.

Restriction 
This function is use with the ODS EPUB3 destination.

Example 
title;
ods epub3 file="VideoeBook.epub"
    title="Marmot Sighting eBook"
    newchapter=now;
ods escapechar='^';
ods proclabel = "Marmot Sighting (Video)";
proc odstext contents="";
p ""{video multimedia/marmot.m4v?controls=controls;
    poster=multimedia/marmot.jpg;
    float=left;
    margin=0;
    marginright=.5em;
    width=240px;
    height=135px}
Check out the marmot
"{noteref large mountain-dwelling ground squirrel}
we saw while driving the
"{style
    [url='http://en.wikipedia.org/wiki/Trail_Ridge_Road']
    Trail Ridge Road}.
run;
ods epub3 close;

Examples

Example 1: Basic Inline Formatting Functions

Features:
- ODS RTF statement:
  - Actions: CLOSE
  - Options: FILE=
- Other features:
  - OPTIONS statement
  - PROC PRINT
  - TITLE statement

Details

The following example highlights inline formatting functions that are supported for all destinations. It also shows how to nest inline formatting functions. In this example, the RTF and PDF destinations are used.

Note: To see all of the styles and colors displayed properly, use Print Preview to view the output.
Program

options nodate nonumber;

ods html close;

ods escapechar="^";

ods rtf file="rtfInlinPuncs.rtf";
ods pdf file="pdfInlinPuncs.pdf";

title "Examples of Inline Formatting Functions";

title2 'Example of ^{nbspace 3} Non-Breaking Spaces Function';

title3 'Example of ^{newline 2} Newline Function';

title4 'Example of ^{raw \cf12 RAW} RAW function';

title5 'Example of UNICODE ^{unicode 03B1} function';

title6 'Example ^{style [foreground=red] of Super, Alpha ^{super ^{unicode ALPHA} ^{style [foreground=green] Nested}}} Formatting} and Scoping";

proc print data=sashelp.class(obs=4);
run;

ods _all_ close;

Program Description

Turn off the Date and Page number. The NODATE option turns off the output of the date and time. The NONUMBER option tells SAS not to print the page number on the first title line of each page of output.

options nodate nonumber;
Close the HTML destination so that no HTML output is produced. The HTML destination is open by default. The ODS HTML statement closes the HTML destination to conserve resources.

    ods html close;

Set the escape character for inline formatting.

    ods escapechar="^";

Create RTF and PDF output. The ODS RTF statement opens the RTF destination and creates RTF output. The ODS PDF statement opens the PDF destination and creates PDF output.

    ods rtf file="rtfInlinFuncs.rtf";
    ods pdf file="pdfInlinFuncs.pdf";

Set the TITLE statement. This TITLE statement provides the topic title for the RTF output.

    title "Examples of Inline Formatting Functions";

Show the NBSPACE function. The non-breaking spaces function, NBSPACE, puts the number of spaces that you specified in the output of the title.

    title2 'Example of ^{nbspace 3} Non-Breaking Spaces Function';

Show the NEWLINE function. The NEWLINE function puts the specified number of additional line feeds in the output of the title.

    title3 'Example of ^{newline 2} Newline Function';

Show the RAW function. The RAW function puts the escaped text that you specified into the file exactly as it is shown. Each ODS destination has special instructions that are recognized. In the following code, \texttt{\textbackslash cf12} is an instruction that the RTF destination recognizes and can display. The PDF destination does not recognize this instruction.

    title4 'Example of ^{raw \textbackslash cf12 RAW} RAW function';

Show the UNICODE function. This TITLE statement shows how the UNICODE function works.

    title5 'Example of UNICODE ^{unicode 03B1} function';

Show the STYLE function and the nesting of functions. This TITLE statement shows the STYLE function using style attribute FOREGROUND=. This example also shows the nesting of the STYLE, SUPER, and UNICODE functions.
Show the SUPER function and the nesting of functions. This TITLE statement shows the STYLE function using style attribute FOREGROUND=. This example also shows the nesting of the STYLE, SUB, and SIGMA functions.

Print the data set.

proc print data=sashelp.class(obs=4);
run;

Close the ODS destinations. The ODS _ALL_ CLOSE statement closes the RTF and PDF destinations and all of the files that are associated with it. If you do not close the destination, you cannot view the files in a browser window.

RTF Output

This output shows the basic inline formatting functions and how you can use them with the TITLE statement, starting with the non-breaking line function (NBSPACE). The other functions used in the code and shown in the following output are NEWLINE, RAW, UNICODE, ALPHA, STYLE, SUPER, SUB, and SIGMA. Nesting functions are also demonstrated in the RTF output. Note that only one level of nesting occurs with the SUB and SUPER functions in the RTF output.

### Examples of Inline Formatting Functions

#### Example of Non-Breaking Spaces Function

<table>
<thead>
<tr>
<th>Obs</th>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69.0</td>
<td>112.5</td>
</tr>
<tr>
<td>2</td>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84.0</td>
</tr>
<tr>
<td>3</td>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98.0</td>
</tr>
<tr>
<td>4</td>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
<td>102.5</td>
</tr>
</tbody>
</table>
PDF Output

This output shows the basic inline formatting functions and how you can use them with the TITLE statement, starting with the non-breaking line function (NBSPACE). The other functions used in the code and shown in the following output are NEWLINE, RAW, UNICODE, STYLE, SUPER, SUB, and SIGMA. Nesting functions are also demonstrated in this PDF output. Note that the SUB and SUPER functions are not honored when nested in the PDF destination. Notice that the SUPER function is not recognized in title6 because of where it is nested. The PDF destination does not recognize the SUB function properly because the subscript-value does not immediately follow the SUB function.

Also note that in title4, the PDF destination cannot display the special instruction provided in the RAW function. The \cf12 instruction is an RTF instruction.

Example 2: Customizing Titles and Page Numbers

Features:
- ODS ESCAPECHAR statement
- STYLE function
- ODS PDF statement
- STARTPAGE=NO option
- NOGTITLE option

Other features:
- OPTIONS statement
- PROC PRINT
- PROC SORT
- PROC TABULATE
- TITLE statement

Details

More granular control over the style of title and footnote text is available by using style overrides. Using the ODS ESCAPECHAR statement with the STYLE format enables you to change the style of individual pieces of text inside a title or footnote. This
example uses the PDF destination, but the technique can be applied to the HTML and
RTF destinations as well. This example shows the nesting of style attributes that is
available with inline style overrides.

This example also shows how page numbers can be customized in PDF output by using
the escape character with the \{THISPAGE\} function and \{LASTPAGE\} function.

Program

ods html close;
ods pdf file="your-file-path/PDFOverrides.pdf" startpage=no nogtitle;
ods escapechar=^;
title "^\{style [fontstyle=italic] Predicted
    ^\{style [fontstyle=roman] and
    ^\{style [foreground=italic] Actual
    ^\{style [fontstyle=roman] Sales\}\}";
title2 "^\{style [color=orange] For Furniture By Region\}";
footnote "Page ^\{thispage\} of ^\{lastpage\}";
options obs=50000;
proc sort data=sashelp.prdsale out=prdsale;
    by Country;
run;
proc tabulate data=sashelp.prdsale;
    by country;
    var predict actual;
    class region division prodtype year;
    table year=\{label='\'},
        region*(division*prodtype all=\{label='division total'\})
        all=\{label='grand total'\},
        predict=\{label='total predicted sales'\}
        *f=dollar10.*sum=\{label='\'}
        actual=\{label='total actual sales'\}
        *f=dollar10.*sum=\{label='\'} / box=_page_;
run;
quit;

options obs=8;
proc print data=prdsale;
    var product region actual predict;
run;
ods pdf close;
ods html;

Program Description

Close the HTML destination so that no HTML output is produced. The HTML
destination is open by default. The ODS HTML CLOSE statement closes the HTML
OGN the PDF destination and specify the ODS PDF statement options. The STARTPAGE=NO option specifies that no new pages are inserted at the beginning of each procedure, or within certain procedures, even if new pages are requested by the procedure code.

```
ods pdf file="your-file-path/PDFOverrides.pdf" startpage=no nogtitle;
```

Set the escape character for inline formatting and specify the titles. The ODS ESCAPECHAR statement specifies that ^ is the representative character to use with inline styles. The STYLE function specifies the style attributes to use to customize the appearance of the titles.

```
ods escapechar="^";

title "'^{style [fontstyle=italic] Predicted
    ^{style [fontstyle=roman] and }
    ^{style [foreground=italic] Actual }
    ^{style [fontstyle=roman] Sales});";

title2 "'^{style [color=orange ] For Furniture By Region}";
```

Customize page numbers. The {THISPAGE} function and {LASTPAGE} function together with the escape character add the text “Page x of y”, where x and y are the appropriate page numbers.

```
footnote "Page ^{thispage} of ^{lastpage};"
```

Create the procedure output.

```
options obs=50000;

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

proc tabulate data=sashelp.prdsale;
  by country;
  var predict actual;
  class region division prodtype year;
  table year=[label=' '],
    region*(division*prodtype all=[label='division total'])
      all=[label='grand total'],
    predict=[label='total predicted sales']
      *f=dollar10.*sum=[label='']
    actual=[label='total actual sales']
      *f=dollar10.*sum=[label=''] /
      box=_page_;
run;
quit;
```

```
options obs=8;
proc print data=prdsale;
  var product region actual predict;
run;
```
Close the PDF destination and open the HTML destination. The ODS PDF CLOSE statement closes the PDF destination and all of the files that are associated with it. You must close the destinations before you can view the output with a browser or before you can send the output to a physical printer. The ODS HTML statement opens the HTML destination and returns SAS to the default ODS destination.

```sas
ods pdf close;
ods html;
```

PDF Output

**Output 6.27  PDF Output with Customized Titles and Page Numbers**

**Predicted and Actual Sales**

*For Furniture By Region*

<table>
<thead>
<tr>
<th>Region</th>
<th>Division</th>
<th>Product type</th>
<th>1993</th>
<th>total predicted sales</th>
<th>total actual sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>$11,081</td>
<td>$12,483</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$21,939</td>
<td>$16,991</td>
<td></td>
</tr>
<tr>
<td>EDUCATION</td>
<td>FURNITURE</td>
<td></td>
<td>$12,972</td>
<td>$14,467</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$16,434</td>
<td>$20,189</td>
<td></td>
</tr>
<tr>
<td></td>
<td>division total</td>
<td></td>
<td>$62,426</td>
<td>$64,130</td>
<td></td>
</tr>
<tr>
<td>WEST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>$10,286</td>
<td>$10,380</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$16,042</td>
<td>$16,371</td>
<td></td>
</tr>
<tr>
<td>EDUCATION</td>
<td>FURNITURE</td>
<td></td>
<td>$12,816</td>
<td>$11,234</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$17,759</td>
<td>$18,905</td>
<td></td>
</tr>
<tr>
<td></td>
<td>division total</td>
<td></td>
<td>$56,903</td>
<td>$56,880</td>
<td></td>
</tr>
<tr>
<td></td>
<td>grand total</td>
<td></td>
<td>$119,329</td>
<td>$121,020</td>
<td></td>
</tr>
</tbody>
</table>

Page 1 of 7

ODS EXCEL Statement

Opens, manages, or closes the ODS destination for Excel, which produces Excel spreadsheet files compatible with Microsoft Office 2010 and later versions.

**Valid in:** Anywhere

**Category:** ODS: Third-Party Formatted
Defaults: The default style is Excel.
PNG is the default device driver for the ODS destination for Excel.
ODS uses the filename that is specified in the SAS registry. The default filename for the ODS destination for Excel is "sasexcl.xlsx".

Interaction: By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC PLOT or PROC CHART), ODS formats the output in SAS Monospace font. If you are creating output that is viewed in an operating environment where SAS software is not installed, this output will not be displayed correctly. This is because without SAS, the SAS Monospace font is not recognized. To make your document display correctly, include the following statement before your SAS program:

```sas
OPTIONS FORMCHAR="|----|+|---+=|-/<>*";
```

z/OS specifics: On z/OS, the ODS destination for Excel only works with the HFS file system. You must use the FILESYSTEM=HFS option. The external file specified by the FILE= option must be an HFS file. If the WORK= option is used, the directory must be an HFS directory.

Syntax

```sas
ODS EXCEL <(<ID>= identifier)> < action> ;
ODS EXCEL <(<ID>= identifier)> <option(s)> ;
```

Summary of Optional Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ID= identifier)</td>
<td>Open multiple instances of the same destination at the same time</td>
</tr>
<tr>
<td>ANCHOR='anchor-name'</td>
<td>Specify the root name for the anchor tag that identifies each output object in the current file</td>
</tr>
<tr>
<td>AUTHOR='text-string'</td>
<td>Specify the author of the Excel document</td>
</tr>
<tr>
<td>BOX_SIZING=(CONTENT_BOX</td>
<td>BORDER_BOX)</td>
</tr>
<tr>
<td>CATEGORY='text-string'</td>
<td>Specify the category of the Excel document</td>
</tr>
<tr>
<td>CLOSE</td>
<td>Close the destination and the file that is associated with it</td>
</tr>
<tr>
<td>COMMENTS='text-string'</td>
<td>Add comments to the properties of the Excel document</td>
</tr>
<tr>
<td>CSSSTYLE='file-specification'=(media-type1&lt;...media-type-10&gt;)</td>
<td>Specify a cascading style sheet to apply to your output</td>
</tr>
<tr>
<td>DOM='external-file'&gt;</td>
<td>Specify that the ODS document object model is written to the SAS log or to an external file.</td>
</tr>
<tr>
<td>DPI='number'</td>
<td>Specify the image resolution for the graphical output</td>
</tr>
<tr>
<td>EXCLUDE exclusion(s)</td>
<td>ALL</td>
</tr>
</tbody>
</table>
FILE='file-specification'
Specify the file that contains the Excel created by the destination

GFOOTNOTE | NOGFOOTNOTE
Control the location where footnotes are printed in the graphics output

GTITLE | NOGTITLE
Control the location where titles are printed in the graphics output

IMAGE_DPI='number'
Specify the image resolution for the graphical output

KEYWORDS='text-string'
Add keywords to the Excel document properties

OPTIONS (<suboption(s)>)
Specify destination-specific suboptions

SASDATE
Insert the standard SAS date in the document in place of the default Excel
date and time field

SELECT selection(s) | ALL | NONE
Select output objects for the destination

SHOW
Write to the SAS log the current selection or exclusion list for the destination

STATUS='text-string'
Insert the status of the Excel document

STYLE= style-override(s)
Specifies one or more style-overides to use when writing output files

TEXT='text-string'
Insert text into your document

TITLE='text-string'
Specify a title for the Excel document

WORK='fileref' | 'directory-name'
Specify an alternate directory for the temporary files

Without Arguments
If you use the ODS EXCEL statement without an action or options, then it opens the
ODS destination for Excel and creates Excel output.

Actions
The following actions are available for the ODS EXCEL statement.

CLOSE
closes the destination and any files that are associated with it.

Tip  When an ODS destination is closed, ODS does not send output to that
destination. Closing an unneeded destination conserves system resources.

EXCLUDE exclusion(s) | ALL | NONE
excludes one or more output objects from the destination.

Default NONE

Restriction A destination must be open for this action to take effect.

See “ODS EXCLUDE Statement” on page 321
**SELECT** selection(s) | ALL | NONE
selects output objects for the specified destination.

<table>
<thead>
<tr>
<th>Default</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restriction</td>
<td>A destination must be open for this action to take effect.</td>
</tr>
<tr>
<td>See</td>
<td>“ODS SELECT Statement” on page 758</td>
</tr>
</tbody>
</table>

**SHOW**
writes the current selection list or exclusion list for the destination to the SAS log.

<table>
<thead>
<tr>
<th>Restriction</th>
<th>The destination must be open for this action to take effect.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tip</td>
<td>If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.</td>
</tr>
<tr>
<td>See</td>
<td>“ODS SHOW Statement” on page 771</td>
</tr>
</tbody>
</table>

**Optional Arguments**

**ANCHOR='anchor-name'**
specifies the root name for the anchor tag that identifies each output object in the current file.

The ANCHOR= option in ODS EXCEL acts like IDs do in CSS. The ANCHOR= option allows you to change the value of the ID= attribute. ID=attributes can be seen when you use the DOM option. The DOM option is used when adding the #ID selector for style output when using CSS with the ODS Excel destination.

Each output object must have an anchor tag for the bookmarks to reference. The references are automatically created by ODS. These references, point to the name of an anchor. Therefore, each anchor name in a file must be unique.

*anchor-name*
is the root name for the anchor tag that identifies each output object in the current file. Each output object must have an anchor tag for the bookmarks to reference. The references are automatically created by ODS. These references, point to the name of an anchor. Therefore, each anchor name in a file must be unique. By default IDX is the default name for the first object.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR="TABULATE", then ODS names the first anchor tabulate. The second anchor is named tabulate1; the third is named tabulate2, and so on. ODS Excel uses anchors to name ID selectors when using CSS to style worksheets.

| Requirement | You must enclose anchor-name in quotation marks. |

**Example**

```ods excel anchor="Robin";```

**AUTHOR='text-string'**
specifies the author of the Excel document. This information can be seen in the document properties.
BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry. For information about using the SAS Registry, see “Changing SAS Registry Settings for ODS” on page 40.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing.

CATEGORY='text-string'
specifies the category of the Excel document. This information can be seen in the document properties.

COMMENTS='text-string'
adds comments to the properties of the Excel document. This information can be seen in the document properties.

CSSSTYLE='file-specification'(media-type1<..media-type-10>)
specifies a cascading style sheet to apply to your output.

file-specification
specifies a file, fileref, or URL that contains CSS code.

file-specification is one of the following:

"external-file"
 is the name of the external file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see SAS Statements: Reference.

"URL"
 is a URL to an external file.

Requirement You must enclose URL in quotation marks.

(media-type-1<.. media-type-10>)
specifies one or more media blocks that correspond to the type of media that your output is rendered on. CSS uses media type blocks to specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not contained in any media blocks. By using the media-type suboption, in addition to the general CSS code, you can import the section of a CSS file intended only for a specific media type.

Default If no media-type is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

Range You can specify up to ten different media types.

Requirements You must enclose media-type in parentheses.
You must specify `media-type` next to the `file-specification` specified by the `CSSSTYLE=` option.

**Tip**

If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

**Interaction**

If both the `STYLE=` option and the `CSSSTYLE=` option are specified in an ODS statement, the option specified last is the option that is used.

**See**

For an example of a valid for ODS CSS file, see “Example 6: Applying a CSS File to ODS Output” on page 527.

**DOM<"external-file">**

specifies that the ODS document object model is written to the SAS log or an external file.

`external-file`

is the name of an external output file.

**Requirement**

You must enclose `external-file` in quotation marks.

**See**

For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in *SAS Output Delivery System: Advanced Topics*.

**DPI='number'**

specifies the image resolution for graphical output.

**Alias**

`Image_DPI=

**Default**

150

**CAUTION**

Caution: When using high DPI= or DPI_IMAGE values (values over 1000), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500M or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.

**FILE='file-specification'**

specifies the file that contains the Excel created by the destination.

`'file-specification'`

specifies the file or fileref to receive output.

`file-specification` is one of the following:

**external-file**

is the name of an external file to receive output.

**Requirement**

You must enclose `external-file` in quotation marks.

**fileref**

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.
ODS uses the filename that is specified in the SAS registry. The default filename for the ODS destination for Excel is “sasexcl.xlsx”.

**GFOOTNOTE | NOGFOOTNOTE**
controls the location where footnotes are printed in the graphics output.

**GFOOTNOTE**
prints footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

**NOGFOOTNOTE**
prints footnotes that are created by ODS, which appear outside the graph borders.

Default **GFOOTNOTE**

Restriction This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

See For details about the SAS/GRAPH FOOTNOTE statement, see “FOOTNOTE Statement” in SAS/GRAPH: Reference.

**GTITLE | NOGTITLE**
controls the location where titles are printed in the graphics output.

**GTITLE**
prints the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.

**NOGTITLE**
prints the title that is created by ODS, which appears outside of the graph borders.

Default **GTITLE**

Restriction This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

See For details about the SAS /GRAPH TITLE statement, see “TITLE, FOOTNOTE, and NOTE Statements” in SAS/GRAPH: Reference.

**IMAGE_DPI='number'**
specifies the image resolution for graphical output.

Alias **DPI=**

Default 150

**CAUTION** Caution: When using high DPI= or DPI_IMAGE values (values over 1000), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500M or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.
KEYWORDS='text-string'
provides keywords in the Excel document. This information can be seen in the document properties.

Note: The KEYWORDS values are listed next to “Tags” in the properties pane.

(ID= identifier)
enables you to run multiple instances of the same destination at the same time. Each instance can have different options.

identifier
specifies another instance of the destination that is already open. identifier is numeric or a series of characters that begin with a letter or an underscore.
Subsequent characters can include letters, underscores, and numeric characters.

Restriction
If identifier is numeric, it must be a positive integer.

Requirement
You must specify the ID= option immediately after the destination name.

Tip
You can omit the ID= option and instead use a name or a number to identify the instance.

Example
“Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594

OPTIONS (< suboption(s)>)
specifies destination-specific suboptions with space-delimited name='value' pairs.

suboption(s) are the following:

(Absolute_COLUMN_WIDTH='number-list' | 'NONE')
Specifies the column widths. Lists widths to use for columns instead of allowing SAS to determine the column width (measured widths). The number-list is a comma separated list of numbers. You can use 'NONE' to reset to the default.

Default
None. SAS determines the width.

Example
ods excel file='footer.xlsx'
   options(absolute_column_width=’16’);
proc print data=sashelp.class(obs=5);
run;
ods excel close;

(Absolute_ROW_HEIGHT='number_list')
Specifies the row heights. Lists heights to use for each row instead of allowing SAS to determine the column height (measured height).

Default
Default is nil.

Tip
Delimit multiple row values with commas.

Example
ods excel options(Absolute_ROW_HEIGHT=’20’);

(AUTOFILTER = 'ALL' | 'NONE' | 'range')
Turns on filtering for specified columns in the worksheet.

ALL
an autofilter is applied to all columns.
NONE
no autofiltering is applied.

range
filtering is applied to the range of columns specified. For example, if a range
of '3-5' is specified, the auto filter is applied to that range of columns.

Default NONE.

Example
ods excel file='footer.xlsx'
options(autofilter='all');
proc print
data=sashelp.class(obs=5);
run;
ods excel close;

(BLACKANDWHITE='OFF' | 'ON')
enables printing of the worksheet in black and white.

ON
prints the worksheet in black and white.

Alias YES

OFF
does not print the worksheet in black and white.

Alias NO

Default OFF

Example ods excel options(blackandwhite='on');

(BLANK_SHEET='string')
creates a blank worksheet with the specified name. The string is a string with a
length greater than zero. This name is used in combination with a worksheet
counter to create a unique name.

Default NONE

Range Worksheet names can be up to 28 characters long.

Example ods excel options(BLANK_SHEET='SAS Sheet 1');

(CENTER_HORIZONTAL= 'OFF' | 'ON')
centers the worksheet horizontally when printing.

ON
centers the worksheet horizontally when printing.

Alias YES

OFF
does not center the worksheet horizontally when printing.

Alias NO

Default OFF
Example  ods excel options(center_horizontal='yes');

(CENTER_VERTICAL= 'OFF' | 'ON')
specifies if the worksheet is to be centered vertically when printing.

ON
centers the worksheet vertically when printing.

Alias  YES

OFF
does not center the worksheet vertically when printing.

Alias  NO

Default  OFF

Example  ods excel options(center_vertical='yes');

(COLUMN_REPEAT="number " | "number-range" | "HEADER")
controls how column headings are repeated across pages.

HEADER
repeat any of the columns containing headers.

number
specifies that the header of the column specified is repeated on each page.

Example  The following example specifies that the header for column 1 is repeated across the printed page.
  ods excel options(column_repeat='1');

number-range
specifies that the headers of the columns within the range specified are repeated on each page.

Example  The following example specifies that the headers for columns 1, 2, and 3 are repeated on each page.
  ods excel options(column_repeat='1-3');

Default  nil (no repeating)

(CONTENTS= 'OFF' | 'ON')
creates a worksheet that contains the table of contents.

ON
creates a worksheet that contains the table of contents.

Alias  YES

OFF
creates a worksheet that does not contain a table of contents.

Alias  NO

Default  OFF

Example  ods excel options(contents='yes');
(DPI='number')
specifies the dots per inch for print resolution. Numbers allowed are 300, 600, and 1200.

Default: 300 DPI

Example: ods excel options(dpi='600');

(DRAFTQUALITY='OFF'|'ON')
specifies if draft quality should be used for printing

ON
specifies that draft quality should be used for printing.

Alias: YES

OFF
specifies that draft quality should not be used for printing.

Alias: NO

Default: OFF

Note: Graphs will not be printed if DRAFTQUALITY = YES.

Example: ods excel options(draftquality='on');

(EMBEDDED_FOOTNOTES='OFF'|'ON')
specifies whether footnotes should appear in the worksheet.

ON
embed footnotes in the worksheet.

Alias: YES

OFF
do not embed footnotes in the worksheet.

Alias: NO

Default: OFF

Example: ods excel options(embedded_footnotes='yes');

(EMBED_FOOTNOTES_ONCE='OFF'|'ON')
specifies whether embedded footnotes should appear only at the bottom of the worksheet.

ON
embedded footnotes appear only once at the bottom of the worksheet.

Alias: YES

OFF
embedded footnotes appear at the bottom of the worksheet.

Alias: NO
Alias        EMBED_FOOTERS_ONCE =
Default       OFF
Example       ods excel options(embed_footnotes_once='yes');

(EMBEDDED_TITLES= 'OFF' | 'ON')
specifies whether titles should appear in the worksheet.

ON
embed titles in the worksheet.

Alias        YES

OFF
do not embed titles in the worksheet.

Alias        NO
Default       NO
Example       ods excel file='myxml.xlsx' options(embedded_titles='on');

Example       “Example 1: Customizing Your Excel Output” on page 312

(EMBEDDED_TITLES_ONCE= 'OFF' | 'ON')
specifies whether embedded titles should appear at the top of the worksheet only once.

ON
embedded titles appear only once at the top of the worksheet.

Alias        YES

OFF
titles appear as they normally appear.

Alias        NO
Default       OFF
Example       ods excel options(embed_titles_once='on');

Example       “Example 1: Customizing Your Excel Output” on page 312

FITTOPAGE= 'OFF' | 'ON')
specifies that the worksheet should fit on a page when printing.

ON
fits the worksheet on the page when printing.

Alias        YES

OFF
does not try to fit the worksheet on the page when printing.

Alias        NO
(FORMULAS= 'OFF' | 'ON')
specifies if data values that begin with an "=" become formulas or cell values.

ON
data values that begin with an "=" become formulas.

Alias YES

OFF
data values that begin with an "=" become cell values.

Alias NO

Default ON

Example ods excel options(formulas='off');

(FROZEN_HEADERS= 'OFF' | 'ON' | number)
specifies that headers can scroll or not scroll with the scroll bar.

ON
headers do scroll with the scroll bar.

Alias YES

TRUE
headers do scroll with the scroll bar.

OFF
headers do not scroll with the scroll bar.

Alias NO

FALSE
headers do not scroll with the scroll bar.

number
the number of the header row that does not scroll with the scroll bar.

Default OFF

Example ods excel options(frozen_headers='on');

(FROZEN_ROWHEADERS= 'OFF' | 'ON' | number)
specifies if the row headers on the left scroll when the table data scrolls.

ON
the header rows on the left scroll when the table data scrolls.

Alias YES

OFF
no headers are frozen.

Alias NO
number
freeze the number of columns specified.

Default OFF
Example ods excel options(frozen_rowheaders='yes');

(GRIDLINES='OFF' | 'ON')
specifies if grid lines are printed.

ON
grid lines are printed.

Alias YES

OFF
grid lines are not printed.

Alias NO

Default OFF
Example ods excel options(gridlines='on');

(HIDDEN_COLUMNS = 'number_list_range')
Specifies the columns to hide. The columns identified are hidden. You can specify a number to hide a specific column, a list of numbers to hide a bunch of columns, or a range of numbers to hide consecutive columns.

Note: Each value in the number_list_range is separated by commas.

Default None. All columns are shown.
Example ods excel options(HIDDEN_COLUMNS='1, 2, 5, 6, 8-10');

(HIDDEN_ROWS = 'number_list_range')
Specifies the rows to hide. You can specify a list of rows to hide or a range of rows to hide.

Default All rows are shown.
Example ods excel options(HIDDEN_ROWS='1,2,5,6,8-10');

(INDEX= 'OFF' | 'ON')
creates a worksheet that contains an index of all worksheets.

ON
creates a worksheet that contains an index of all worksheets.

Alias YES

OFF
does not create a worksheet that contains an index of all worksheets.

Alias NO

Default OFF
Example ods excel options(index='on');

(MSG_LEVEL='string')
suppresses messages from the Excel.

Default  No.

(ORIENTATION= 'PORTRAIT'| 'LANDSCAPE')
orient the printed page as either portrait or landscape.

PORTRAIT
prints a portrait-oriented page.

LANDSCAPE
prints a landscape-oriented page.

Default  PORTRAIT

Example  ods excel options(orientation='landscape');

(PAGE_ORDER_ACROSS= 'OFF'| 'ON')
Specifies that the information across the page is printed first followed by the information that is down the page.

ON
print all of the information across the page first, followed by the information down the page.

Alias  YES

OFF
print all of the information down the page first, followed by the information across the page.

Alias  NO

Default  OFF

Example  ods excel options(page_order_across='on');

(PAGES_FITHEIGHT='number')
specifies the number of pages down to fit the worksheet when printing.

Example  ods excel options(PAGES_FITHEIGHT='3');

(PAGES_FITWIDTH='number')
specifies the number of pages to fit the worksheet across when printing.

Example  ods excel options(PAGES_FITWIDTH='3');

(PRINT_AREA= 'item')
describes the printed area in terms of the column and row to start and end with. You can use column and row numbers and letters. For example,

print_areas='a,2,g,20'

indicates top left corner and bottom right corner.

Default  NONE
Tip: Separate each PRINT_AREA item with a comma.

(PRINT_FOOTER='text-string')
specifies the text that is placed in the footer when printing. If a footnote is specified, that footnote is used. Otherwise, this text is placed in the footer.

Example: ods excel options(PRINT_FOOTER="Draft Copy for Review");

(PRINT_FOOTER_MARGIN='number')
specifies the footer margin that is set in the page setup window when printing. This margin is measured in inches.

Default: 0.5 inches.

Example: ods excel options(PRINT_FOOTER_MARGIN="2");

(PRINT_HEADER='text-string')
specifies the text that is placed in the header when printing. If no title is specified, this text is used by Excel on the printed page. If a title has been specified with the TITLE statement, that title is used.

Example: The following example specifies “My custom header” as the text for headers.
          ods excel file='footer.xlsx' options(print_header='My custom header');

(PRINT_HEADER_MARGIN='number')
specifies the header margin that is set in the page setup dialog window when printing. This margin is measured in inches.

Default: 0.5 inches

Example: ods excel options(print_header_margin="1");

(ROWBREAKS_COUNT='number')
specifies that for every number data rows, insert a print page for printing.

Example: ods excel options(rowbreaks_count="20");

(ROWBREAKS_INTERVAL='OUTPUT' | 'PROC' | 'NONE')
controls the placement of page breaks. This option places a page break after each output object or after each procedure.

OUTPUT
inserts a page break between output objects.

PROC
inserts a page break between each procedure’s output.

NONE
does not insert custom page breaks.

Default: NONE

Example: ods excel options(rowbreaks_interval='proc');

(ROWCOLHEADINGS='OFF' | 'ON')
specifies if row and column headings should be printed.

ON
prints row and column headings.
Alias YES
——
OFF does not print row and column headings.

Alias NO
——
Default OFF. Does not print row and column headings.

Example ods excel options(rowcolheadings='on');

(ROW_HEIGHTS ='number_list') specifies the height of the row. The measurement is in points and is the positional array of row height values. This value overrides the default height.

The parameters of this option are positional, but not all values must be specified. A value of 0 means that the height should be taken from the style. The first value is the height for table header rows. The next is the height for the table body rows. The next value is the row height for BY lines. The fourth is for titles, the fifth is for footers, the sixth is the page break height, and the last value is the height for paragraph skip.

Default By default, the measurement is taken from the font size. The table row height is defined by the font size in the header style

Tip Delimit multiple row values with commas.

Example ods excel options(ROW_HEIGHTS='20,50,100,20,50,100,50');

(ROW_REPEAT='NONE' | 'HEADER' | 'number' | number-range) controls how row headings are repeated across pages. For example, row_repeat="header" repeats all the row headers, row_repeat="1" repeats only one row, and row_repeat="1-3" repeats rows 1, 2, and 3.

NONE specifies that no rows are repeated on each page.

HEADER specifies that all row headings are repeated on each page.

number specifies that the header of the row specified is repeated on each page.

Example The following example specifies that the header for row 1 is repeated across the printed page.
ods excel options(row_repeat='1');

number-range specifies that the headers of the rows within the range specified are repeated on each page.

Example The following example specifies that the headers for rows 1, 2, and 3 are repeated across the printed page.
ods excel options(row_repeat='1-3');

Default NONE
(SCALE='number')
specifies the scale level for printing.

Default 100

Example ods excel options(scale="10");

(SHEET_INTERVAL='BYGROUP' | 'PAGE' | 'PROC' | 'NONE' | 'TABLE')
specifies the criteria for when a new worksheet is created.

BYGROUP creates a new worksheet after each BYGROUP.

Alias BYGROUPS

NONE creates one worksheet with all of the data.

PAGE creates a worksheet for each page of procedure output.

PROC creates a worksheet of all of the procedure output.

TABLE creates a worksheet for each table.

Alias OUTPUT

Default TABLE

Example ods excel file='myExcel.xlsx' options(sheet_interval='proc');

Example “Example 1: Customizing Your Excel Output” on page 312

(SHEET_LABEL='text-string' | 'NONE')
used as the first part of the name in the worksheet label instead of the predefined string. This option is used in combination with the various worksheet naming options like SHEET_INTERVAL.

Note: The sheet label prepends to the sheet name.

NONE creates worksheets named by default.

text-string names the first part of the label of a worksheet with the specified string.

Default NONE

Example ods excel options(sheet_label="country");

Example “Example 1: Customizing Your Excel Output” on page 312

(SHEET_NAME='text-string')
specifies the name for the next worksheet. This name is used along with the worksheet counter to create a unique name.

Range Worksheet names can be up to 28 characters long.
Example:
```
ods excel options (sheet_name="PROC REPORT Stats");
```

**START_AT='string'**

Specifies a starting cell for the report. The default is to start at column 1 and row 1.

*Default* 1,1

**Tip** This option cannot be changed in the middle of a sheet.

**Example**
```
ods excel options (start_at="2,2");
```

**SUPPRESS_BYLINES='OFF' | 'ON'**

Specifies whether to suppress BY lines in the worksheet.

- **ON**
  - Suppress BY lines in the worksheet.
  - *Alias* YES

- **OFF**
  - BY lines appear in the worksheet.
  - *Alias* NO

*Default* OFF

**Example**
```
ods excel options (suppress_bylines='on');
```

**Example**
```
“Example 1: Customizing Your Excel Output” on page 312
```

**TAB_COLOR='string'**

Specifies the color for the next worksheet.

**Example**
```
ods excel options (tab_color='red');
ods excel options (tab_color='#ff0000');
ods excel options (tab_color='rgba(0,100%,0,0.5)');
```

**TITLE_FOOTNOTE_NOBREAK='number'**

Specifies that titles and footnotes do not wrap across lines.

- **ON**
  - Titles and footnotes wrap across lines.
  - *Alias* YES

- **OFF**
  - Titles and footnotes do not wrap across lines.
  - *Alias* NO

*Default* NO.

**Example**
```
ods excel file=a options (title_footnote_nobreak='yes'
embedded_titles='yes');
```
(TITLE_FOOTNOTE_WIDTH="number")
specifies the number of columns that titles and footnotes should span. If zero,
titles and footnotes are merged across the number of columns currently in use.

Default 0. Titles and footnotes merge across the number of columns currently in use.

Example ods excel options(TITLE_FOOTNOTE_WIDTH="10");

(ZOOM="number")
indicates the initial zoom level on the worksheet.

Default 100

Example ods excel options(zoom="75");

SASDATE
inserts the standard SAS date in the document in place of the default Excel date and
time field. Excel formats this date field using the format specified by the Excel
Header and Footer dialog. The date field is updated whenever you open the
presentation. When the SASDATE option is used, instead of a date field, the ODS
destination for Excel inserts the date and time that you started your SAS session.
Excel does not update the date and time.

STATUS="text-string"
specifies the status of the Excel document. This information can be seen in the
document properties.

STYLE= style-override(s)
Specify one or more style-overides to use when writing output files.

You can specify a style override in two ways:

• Specify a style element. A style element is a collection of style attributes that
apply to a particular part of the output for a SAS program.

• Specify a style attribute. A style attribute is a name-value pair that describes a
single behavioral or visual aspect of a piece of output. This is the most specific
method of changing the appearance of your output.

style-override has the following form:

style-element-name [ [style-attribute-name-1=style-attribute-value-1
<style-attribute-name-2=style-attribute-value-2 ...>]]

Default Excel is the default style.

See For a complete discussion of style templates, see “TEMPLATE
Procedure: Creating a Style Template” in SAS Output Delivery System:
Procedures Guide.

Example Custom formats can be specified with a style override on the TAGATTR=
style attribute.

style={tagattr='format:$#,##0_);[Red]\{$#,##0\}
formula:RC[-1]-RC[-2]'};

TEXT="text-string"
inserts text into your document by triggering the paragraph event and specifying a
text string to be assigned to the VALUE event variable.
By default the TEXT= option is used in a paragraph event.

See For information about events and event variables, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in SAS Output Delivery System: Procedures Guide.

TITLE='text-string'
specifies a title for the Excel document. This information can be seen in the document properties.

WORK='fileref' | 'directory-name'
specifies an alternate directory for the temporary files. By default, the ODS destination for Excel uses the SAS Work library to hold temporary files. The WORK= option specifies an alternate directory for the temporary files.

fileref is a file reference that has been assigned to a directory. Use the FILENAME statement to assign a fileref.

directory-name is the name of the directory.

Details
The ODS destination for Excel uses Microsoft Open Office XML Format for Office 2010 and later. This statement produces Extensible Markup Language (XML) and represents a way to define and format data for easy exchange.

'Portrait' is the default printing orientation. The orientation can be changed to landscape.

The ODS destination for Excel creates Microsoft spreadsheetML XML. Each table is placed in its own worksheet within a workbook. This destination supports ODS styles, trafficlighting, and custom formats. Numbers, Currency, and percentages are correctly detected and displayed. Style override, TAGATTR= style attribute, can be used to create custom formats for the data. By default, titles and footnotes are included in the worksheet, but they are part of the header and footer of the worksheet.

SAS supports Microsoft Excel 2010 and later. For more information, see

• Microsoft Excel Standards Support
• Base SAS Focus Area
• Using ODS EXCEL and PROC EXPORT to bundle Excel-based reports

Examples

Example 1: Customizing Your Excel Output
Features:
  ODS EXCEL Statement options
    FILE=
    OPTIONS (SHEET_INTERVAL=, SUPPRESS_BYLINES=, SHEET_LABEL=, EMBEDDED_TITLES=, EMBED_TITLES_ONCE = );

Other features:
  PROC TABULATE
Details

The following example shows you how to create a customized Excel workbook that contains PROC TABULATE output. The ODS EXCEL statement is used to make the following customizations:

- a new sheet is created for each BY group
- the BY lines are suppress
- the title created by the TITLE statement is embedded in the output
- the worksheet labels are customized

Program

```plaintext
ods excel file="multitablefinal.xlsx"
   options(sheet_interval="bygroup"
            suppress_bylines="yes"
            sheet_label="country"
            embedded_titles="yes"
            embed_titles_once="yes" );

   title "Historical Sales Data";
   proc tabulate data=sashelp.prdsale;
      by country;
      var predict actual;
      class region division prodtype year;
      table years=[label=' '],
                region*(division*prodtype all=[label='division total'])
                all=[label='grand total'],
                predict=[label='total predicted sales']*f=dollar10.*sum=[label='']
                actual=[label='total actual sales']*f=dollar10.*sum=[label=''] /
                box=_page_; run;
   ods excel close;
```

Program Description

**Open the ODS destination for Excel and specify the options.** Open the ODS destination for Excel and provide a filename for the workbook. The SHEET_INTERVAL= option specifies that a new sheet is created after each by group. The SUPPRESS_BYLINES= option is used to suppress the printing of the BY lines text. The SHEET_LABEL= option specifies the text “Country” as the first part of the worksheet label. The EMBEDDED_TITLES= option specifies that the title created by the TITLE statement is embedded in the Excel worksheet. The EMBED_TITLES_ONCE option specifies that the title is embedded once, at the top of each sheet.

```plaintext
ods excel file="multitablefinal.xlsx"
   options(sheet_interval="bygroup"
            suppress_bylines="yes"
            sheet_label="country"
            embedded_titles="yes"
            embed_titles_once="yes" );
```

**Specify a title for the workbook**
title "Historical Sales Data";

**Produce the procedure output.**

```sas
proc tabulate data=sashelp.prdsale;
  by country;
  var predict actual;
  class region division prodtype year;
  table year=[label=' '],
         region*(division*prodtype all=[label='division total'])
         all=[label='grand total'],
         predict=[label='total predicted sales']*f=dollar10.*sum=[label='']
         actual=[label='total actual sales']*f=dollar10.*sum=[label=''] /
         box=_page_;    
run;
```

**Close the ODS destination for Excel.**

```sas
ods excel close;
```

**Excel Output**

*Output 6.28  Customized Excel Output*
Example 2: Customize ODS Excel Output Using TAGATTR Style Attribute

Features:
- ODS EXCEL Statement

Other features:
- PROC SORT
  - PROC PRINT procedure
    - VAR with STYLE=TAGATTR
    - SUM with STYLE=TAGATTR
    - LABEL

Details
In the following example, the TAGATTR= style attribute allows the user to customize the styles and format of the data in the Excel workbook. Negative values are noted by red font and parenthesis. These are created using the TAGATTR= style attribute.

Program
ods html close;

data prdsale;
  set sashelp.prdsale;
  Difference = actual - predict;
run;

proc sort data=prdsale;
  by country region division year;
run;

ods excel file='tagattr.xlsx';
proc print data=prdsale(obs=15) noobs label split='*';
  id country region division;
run;

var prodttype product quarter month year;
var predict actual /
  style={tagattr='format:$#,##0_\);[Red]\($#,##0\)'};

var difference /
  style={tagattr='format:$#,##0_\);[Red]\($#,##0\)formula:RC[-1]-RC[-2]'};

sum predict actual difference /
  style={tagattr='format:$#,##0_\);[Red]\($#,##0\)'};

label prodttype = 'Product*Type'
predict = 'Predicted*Sales*For Area'
actual = 'Actual*Sales*Amount';
run;

ods excel close;
ods html;

Program Description

Close the HTML destination.
Create the PRDSALE data set. The DATA step creates the PRDSALE data set that contains the calculated difference between the Actual Sales and the Predicted Sales values.

```sas
data prdsale;
  set sashelp.prdsale;
  Difference = actual-predict;
run;
```

Sort the data set.

```sas
proc sort data=prdsale;
  by country region division year;
run;
```

Open the ODS destination for Excel and name the output file.

```sas
ods excel file='tagattr.xlsx';
```

Begin the PRINT procedure step.

```sas
proc print data=prdsale(obs=15) noobs label split='*';
  id country region division;
run;
```

Customize the output format for Predicted Sales and Actual Sales The TAGATTR= style attribute on the first VAR statement specifies the format for the positive values. The negative numbers are printed in red and are surrounded by parenthesis.

```sas
var prodtype product quarter month year;
var predict actual /
  style={tagattr='format:$#,##0_);[Red]\($#,##0\)'};
```

Customize the output format for the difference between the Predicted Sales and Actual Sales The TAGATTR= style attribute on the second VAR statement specifies the format for negative values. The negative numbers are printed in red and are surrounded by parenthesis. The difference is calculated using formula: RC[-1] - RC[-2].

```sas
var difference /
  style={tagattr='format:$#,##0_);[Red]\($#,##0\)formula:RC[-1]-RC[-2]'};
```

Customize the output format for the summary. The TAGATTR= style attribute in the SUM statement specifies the format that is used to show the sums for the Predicted Sales, the Actual Sales, and the Difference. The first TAGATTR= specifies the format for the positive values and the second format is for negative values. The negative values are printed in red and are surrounded by parenthesis.

```sas
sum predict actual difference /
  style={tagattr='format:$#,##0_);[Red]\($#,##0\)'};
```

Specify the labels for the column headings.

```sas
label prodtype = 'Product*Type'
predict  = 'Predicted*Sales*For Area'
actual   = 'Actual*Sales*Amount';
run;
```
Close the ODS destination for Excel. Close the ODS destination for Excel and open the HTML destination.

```
ods excel close;
ods html;
```

**Excel Output**

This output is created using the ODS destination for Excel. It uses the style attribute TAGATTR= to customize the data in the Excel workbook.

**Output 6.29  Customize the Data in the Excel Workbook Using the TAGATTR= Style Attribute**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>SOFA</td>
<td>1</td>
<td>Jan</td>
<td>1993</td>
<td>$425</td>
<td>$5</td>
</tr>
<tr>
<td>3</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>SOFA</td>
<td>1</td>
<td>Feb</td>
<td>1993</td>
<td>$215</td>
<td>$164</td>
</tr>
<tr>
<td>4</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>SOFA</td>
<td>1</td>
<td>Mar</td>
<td>1993</td>
<td>$948</td>
<td>$422</td>
</tr>
<tr>
<td>5</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>SOFA</td>
<td>1</td>
<td>Apr</td>
<td>1993</td>
<td>$544</td>
<td>$424</td>
</tr>
<tr>
<td>6</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>SOFA</td>
<td>2</td>
<td>May</td>
<td>1993</td>
<td>$764</td>
<td>$854</td>
</tr>
<tr>
<td>7</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>SOFA</td>
<td>2</td>
<td>Jun</td>
<td>1993</td>
<td>$446</td>
<td>$168</td>
</tr>
<tr>
<td>8</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>SOFA</td>
<td>2</td>
<td>Jul</td>
<td>1993</td>
<td>$957</td>
<td>$8</td>
</tr>
<tr>
<td>9</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>SOFA</td>
<td>2</td>
<td>Aug</td>
<td>1993</td>
<td>$967</td>
<td>$748</td>
</tr>
<tr>
<td>10</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>SOFA</td>
<td>3</td>
<td>Sep</td>
<td>1993</td>
<td>$11</td>
<td>$662</td>
</tr>
<tr>
<td>11</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>SOFA</td>
<td>3</td>
<td>Oct</td>
<td>1993</td>
<td>$110</td>
<td>$300</td>
</tr>
<tr>
<td>12</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>SOFA</td>
<td>4</td>
<td>Nov</td>
<td>1993</td>
<td>$283</td>
<td>$672</td>
</tr>
<tr>
<td>13</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>SOFA</td>
<td>4</td>
<td>Dec</td>
<td>1993</td>
<td>$215</td>
<td>$894</td>
</tr>
<tr>
<td>14</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>BED</td>
<td>1</td>
<td>Jan</td>
<td>1993</td>
<td>$414</td>
<td>$284</td>
</tr>
<tr>
<td>15</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>BED</td>
<td>1</td>
<td>Feb</td>
<td>1993</td>
<td>$770</td>
<td>$705</td>
</tr>
<tr>
<td>16</td>
<td>CANADA</td>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>BED</td>
<td>1</td>
<td>Mar</td>
<td>1993</td>
<td>$679</td>
<td>$737</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Example 3: Applying a Style Sheet to Excel Output**

**Features:**
- ODS EXCEL Statement
  - ANCHOR option
- CSS Style Sheet

**Other features:**
- PROC PRINT Statement

**Details**

The following program applies a style sheet created in a CSS file to the Excel output. In the StyleSheet, we specify the value for the anchor (the value used with CSS as the ID). The default ID is #IDX, and by renaming the anchor, we can choose the name of the ID so that it does not use the default.

The following example adds the worksheets on the same sheet using option SHEET_INTERVAL= . The CSS styles create different background colors for the headers in the first table and the headers in the second table. ID #expense is specified for
the first table and the ID #Reports is specified for the second table using the ANCHOR=
option.

The following code is an example of the external CSS file StyleSheet.css. Copy and
paste this code into a text editor and save it as StyleSheet.css.

```css
#Expense .header {
    background-color:green
}

#Reports .header {
    background-color:red
}
```

Program

```odds
ods html close;
ods excel file="c:\excelAnchorCss.xlsx"
cssstyle="file-path\Stylesheet.css"
options(sheet_interval="none");
ods excel anchor="expense";
proc print data=sashelp.class;
run;

ods excel anchor="reports" cssstyle="file-path\Stylesheet.css";
proc print data=sashelp.class;
run;

ods excel close;
ods html;
```

Program Description

Close the HTML destination.

```odds
ods html close;
```

Open the ODS destination for Excel, provide a filename, and specify the style
sheet. Open the destination for Excel, specify the filename for the output. The
SHEET_INTERVAL="NONE" option specifies that all output appears on the same sheet.
The CSSSTYLE= option specifies the file StyleSheet.css to provide different
background colors to the output..

```odds
ods excel file="c:\excelAnchorCss.xlsx"
cssstyle="file-path\Stylesheet.css"
options(sheet_interval="none");
```

Apply the CSS styles to the Expense Header. Create a different background color for
the header in the first table by adding the ID #Expense for the first table using the
ANCHOR= option. The file StyleSheet.css contains a matching ID. By default, the IDs
are #IDX and #IDX1 without specifying an anchor ID.

```odds
ods excel anchor="expense";
```
Create the procedure output.

    proc print data=sashelp.class;
    run;

Apply the CSS styles to the Reports Header. Create a different background colors for the header in the second table by adding the ID #Reports using the ANCHOR= option. The file StyleSheet.css contains a matching ID. By default, the ID for the second table is #IDX1 without specifying an anchor ID. However, we have specified a unique ID of #Reports for the second table.

    ods excel anchor="reports" cssstyle="file-path\StyleSheet.css";

Create the procedure output.

    proc print data=sashelp.class;
    run;

Close the ODS destination for Excel. Close the ODS destination for Excel and open the HTML destination.

    ods excel close;
    ods html;
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obs</td>
<td>Name</td>
<td>Sex</td>
<td>Age</td>
<td>Height</td>
<td>Weight</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69.0</td>
<td>112.5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84.0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98.0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
<td>102.5</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>Henry</td>
<td>M</td>
<td>14</td>
<td>63.5</td>
<td>102.5</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>James</td>
<td>M</td>
<td>12</td>
<td>57.3</td>
<td>83.0</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>Jane</td>
<td>F</td>
<td>12</td>
<td>59.8</td>
<td>84.5</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>Janet</td>
<td>F</td>
<td>15</td>
<td>62.5</td>
<td>112.5</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>Jeffrey</td>
<td>M</td>
<td>13</td>
<td>62.5</td>
<td>84.0</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>John</td>
<td>M</td>
<td>12</td>
<td>59.0</td>
<td>99.5</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>Joyce</td>
<td>F</td>
<td>11</td>
<td>51.3</td>
<td>50.5</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>Judy</td>
<td>F</td>
<td>14</td>
<td>64.3</td>
<td>90.0</td>
</tr>
<tr>
<td>14</td>
<td>13</td>
<td>Louise</td>
<td>F</td>
<td>12</td>
<td>56.3</td>
<td>77.0</td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>Mary</td>
<td>F</td>
<td>15</td>
<td>66.5</td>
<td>112.0</td>
</tr>
<tr>
<td>16</td>
<td>15</td>
<td>Philip</td>
<td>M</td>
<td>16</td>
<td>72.0</td>
<td>150.0</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>Robert</td>
<td>M</td>
<td>12</td>
<td>64.8</td>
<td>128.0</td>
</tr>
<tr>
<td>18</td>
<td>17</td>
<td>Ronald</td>
<td>M</td>
<td>15</td>
<td>67.0</td>
<td>133.0</td>
</tr>
<tr>
<td>19</td>
<td>18</td>
<td>Thomas</td>
<td>M</td>
<td>11</td>
<td>57.5</td>
<td>85.0</td>
</tr>
<tr>
<td>20</td>
<td>19</td>
<td>William</td>
<td>M</td>
<td>15</td>
<td>66.5</td>
<td>112.0</td>
</tr>
</tbody>
</table>

**Output 6.30** Customized ODS Excel Output By Using a Style Sheet
**ODS EXCLUDE Statement**

Specifies output objects to exclude from ODS destinations.

- **Valid in:** Anywhere
- **Category:** ODS: Output Control

### Syntax

```ods <ods-destination> exclude exclusion(s) | all | none;```

### Required Arguments

- **exclusion(s)**
  
  Specifies one or more output objects to add to an exclusion list.

By default, ODS automatically modifies exclusion lists at the end of a DATA step that uses ODS, or at the end of a procedure step. For information about modifying these lists, see “Selection and Exclusion Lists” on page 39.

Each exclusion has the following form:

- **output-object** `<(Persist)>`

  Specifies one or more output objects to exclude. To specify an output object, you need to know which output objects your SAS program produces. The ODS TRACE statement writes to the SAS log a trace record that includes the path, the label, and other information about each output object that is produced. You can specify an output object in any of the following ways:

  - a full path. For example, the following is the full path of the output object:
    ```univariate.City_Pop_90.TestsForLocation```

  - a partial path. A partial path consists of any part of the full path that begins immediately after a period (.) and continues to the end of the full path. For example, suppose the full path is the following:
    ```univariate.City_Pop_90.TestsForLocation```

    Then the partial paths are as follows:
    ```city_pop_90.testsforlocation
testsforlocation```

  - a label that is enclosed in quotation marks.

    For example:
    ```"the univariate procedure"```

  - a label path. For example, the following is the label path for the output object:
    ```"the univariate procedure"."citypop_90"."tests for location"```

  *Note:* The trace record shows the label path only if you specify the LABEL option in the ODS TRACE statement.
• a partial label path. A partial label path consists of any part of the label that begins immediately after a period (.) and continues to the end of the label. For example, suppose the label path is the following:

"The UNIVARIATE Procedure"."CityPop_90"."Tests For Location"

Then the partial label paths are as follows:

"CityPop_90"."Tests For Location"
"Tests For Location"

• a mixture of labels and paths.

• any of the partial path specifications, followed by a pound sign (#) and a number. For example, TestsForLocation#3 refers to the third output object that is named TestsForLocation.

See “ODS TRACE Statement” on page 854.

(PERSIST) keeps the output-object that precedes the PERSIST option in the exclusion list until you explicitly modify the list with any of the following ODS statements:

• any ODS SELECT statement
• ODS EXCLUDE NONE
• ODS EXCLUDE ALL
• an ODS EXCLUDE statement that applies to the same output object but does not specify PERSIST

This action is true even if the DATA or procedure step ends.

Requirement You must enclose PERSIST in parentheses.

ALL specifies that ODS does not send any output objects to the open destination.

Alias ODS EXCLUDE DEFAULT

Interaction If you specify ALL without specifying a destination, ODS sets the overall list to EXCLUDE ALL and sets all other lists to their defaults.

Tips Using ODS EXCLUDE ALL is different from closing a destination. The destination remains open, but no output objects are sent to it.

To temporarily suspend a destination, use ODS SELECT NONE. Use ODS SELECT ALL when you want to resume sending output to the suspended destination.

NONE specifies that ODS send all of the output objects to the open destination.

Interaction If you specify the NONE argument without specifying a destination, ODS sets the overall list to EXCLUDE NONE and sets all other lists to their defaults.

Tips ODS EXCLUDE NONE has the same effect as ODS SELECT ALL.
To temporarily suspend a destination, use ODS SELECT NONE. Use ODS SELECT ALL when you want to resume sending output to the suspended destination.

**Optional Arguments**

**NOWARN**

suppresses the warning that an output object was requested but not created.

**Requirement**
The NOWARN option must be enclosed in parentheses.

**Interaction**
The NOWARN option cannot be used with the ALL option or the NONE option.

**Example**
The ODS EXCLUDE statement in the following example specifies that no warning is created if the output object Summary is requested but not created.

```sas
ods exclude summary (nowarn);
```

```sas
proc contents data=sashelp.class;
run;
```

**ODS-destination**

specifies to which ODS destination's exclusion list to write, where `ODS-destination` can be any valid ODS destination. For a discussion of ODS destinations, see “Understanding ODS Destinations” on page 33.

**Default**
If you omit `ODS-destination`, ODS writes to the overall exclusion list.

**Tip**
To set the exclusion list for the output destination to something other than the default, use the “ODS OUTPUT Statement” on page 534.

**WHERE=where-expression**

excludes output objects that meet a particular condition. For example, the following statement excludes only output objects with the word "Histogram" in their name:

```sas
ods exclude where=(_name_ ? 'Histogram');
```

**where-expression**
is an arithmetic or logical expression that consists of a sequence of operators and operands. `where-expression` has this form:

```
(subsetting-variable <comparison-operator where-expression-n>)
```

**subsetting-variable**
is a special type of WHERE expression operand used by SAS to help you find common values in items. For example, this EXCLUDE statement excludes only output objects with the path `City_Pop_90.TestsForLocation`:

```sas
ods exclude / where=(_path_ = 'City_Pop_90.TestsForLocation' );
```

**subsetting-variable** is one of the following:

- `_LABEL_`
  - is the label of the output object.
- `_LABELPATH_`
  - is the label path of the output object.
operator

compares a variable with a value or with another variable. *operator* can be AND, OR NOT, OR, AND NOT, or a comparison operator.

The following table lists some comparison operators:

**Table 6.5  Examples of Comparison Operators**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Mnemonic Equivalent</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>EQ</td>
<td>Equal to</td>
</tr>
<tr>
<td>^= or ^= or ^= or &lt;&gt;</td>
<td>NE</td>
<td>Not equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>GT</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>LT</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>GE</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt;=</td>
<td>LE</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>IN</td>
<td></td>
<td>Equal to one from a list of values</td>
</tr>
</tbody>
</table>

**Details**

For each ODS destination, ODS maintains either a selection list or an exclusion list of output objects. You can use the default output objects selected or excluded for each destination or you can specify which output object you want to produce by selecting or excluding them from a list.

A selection list is a list of output objects that are sent to an ODS destination. An exclusion list is a list of output objects that are excluded from an ODS destination. ODS also maintains an overall selection or exclusion list of output objects. By checking the destination-specific lists and the overall list, ODS determines what output objects to produce. These lists can be modified by using the ODS SELECT statement and the ODS EXCLUDE statement.

**TIP** You can maintain a selection list for one destination and an exclusion list for another. However, the results are less complicated if you maintain the same types of lists for all the destinations to which you route output.

You can view the contents of the exclusion and selection lists by using the ODS SHOW statement. The current selection list is written to the SAS log.

EXCLUDE ALL is the default setting for the ODS OUTPUT destination. SELECT ALL is the default setting for all other destinations. To change the default selection and exclusion lists, use the ODS SELECT or ODS EXCLUDE statements or use the exclude and select actions that are available for some of the ODS statements. However, to set the exclusion list for the OUTPUT destination to something other than the default, use the “ODS OUTPUT Statement” on page 534. For a list of ODS output destinations and explanations of each, see “Understanding ODS Destinations” on page 33.
In order to view output objects that are selected or excluded from your program, use the ODS TRACE statement. The ODS TRACE statement prints the output objects that are selected and excluded and puts the information in a trace record that is written to the SAS log. The trace provides the path, the label, and other information about output objects that are selected and excluded. For complete documentation about viewing and selecting output objects, see the “ODS SELECT Statement” on page 758, the “ODS EXCLUDE Statement” on page 321, and the “ODS TRACE Statement” on page 854.

Example: Conditionally Excluding Output Objects and Sending Them to Different Output Destinations

Features:
- ODS EXCLUDE statement:
  - Options: ODS-Destination, WHERE=
- ODS HTML statement options:
  - CONTENTS=
  - FRAME=
  - PAGE=
  - TEXT=
- ODS PDF statement options:
  - TEXT=
  - STARTPAGE=

Other features:
- PROC UNIVARIATE

Program

```sas
options nodate;

data BPressure;
  length PatientID $2;
  input PatientID $ Systolic Diastolic @@;
  datalines;
  CK 120 50  SS 96  60 FR 100 70 
  CP 120 75  BL 140 90 ES 120 70 
  CP 165 110 JI 110 40 MC 119 66 
  FC 125 76  RW 133 60 KD 108 54 
  DS 110 50  JW 130 80 BH 120 65 
  JW 134 80  SB 118 76 NS 122 78 
  GS 122 70  AB 122 78 EC 112 62 
  HH 122 82
;
run;

ods html text='Systolic Blood Pressure' file='Systolic-body.html' frame='Systolic-frame.htm'
contents='Systolic-contents.htm' page='Systolic-page.htm';

ods pdf file='Diastolic.pdf' text='Diastolic Blood Pressure' startpage=no;
```
ods html exclude where=(_path_ ? "Diastolic") ;
ods pdf exclude where=(_path_ ? "Systolic") ;

proc univariate data=BPressure;
  var Systolic Diastolic;
  run;
ods html close;

ods pdf close;

Program Description

Create the BPressure data set.

options nodate;
  data BPressure;
    length PatientID $2;
    input PatientID $ Systolic Diastolic @@;
    datalines;
    CK 120 50  SS 96  60 FR 100 70
    CP 120 75  BL 140 90 ES 120 70
    CP 165 110 JI 110 40 MC 119 66
    FC 125 76  RW 133 60 KD 108 54
    DS 110 50  JW 130 80 BH 120 65
    JW 134 80  SB 118 76 NS 122 78
    GS 122 70  AB 122 78 EC 112 62
    HH 122 82
    ;
    run;

Create HTML output and add text.

ods html text='Systolic Blood Pressure' file='Systolic-body.html'
  frame='Systolic-frame.htm'
  contents='Systolic-contents.htm'
  page='Systolic-page.htm';

Create PDF output and add text.

ods pdf file='Diastolic.pdf' text='Diastolic Blood Pressure' startpage=no;

Exclude output objects from different output destinations. The first ODS EXCLUDE statement excludes from the HTML destination output objects that have 'Diastolic' in the pathname. The second ODS EXCLUDE statement excludes from the PDF destination output objects that have 'Systolic' in the pathname.

ods html exclude where=(_path_ ? "Diastolic") ;
ods pdf exclude where=(_path_ ? "Systolic") ;
**Create the output objects.** As PROC UNIVARIATE sends each output object to the Output Delivery System, ODS does not send the output objects from PROC UNIVARIATE that match the items in the exclusion list to the open destinations.

```
proc univariate data=BPressure;
  var Systolic Diastolic;
run;
```

**Close the HTML destination.** The ODS HTML CLOSE statement closes the HTML destination and all the files that are associated with it. If you do not close the destination, then you cannot view the HTML file specified by the FRAME attribute until you close your SAS session.

```
ods html close;
```

**Close the PDF destination.** This ODS PDF statement closes the PDF destination and all the files that are associated with it.

```
ods pdf close;
```

**Output**

*Output 6.31  Partial HTML Output with Systolic Output Objects*
**ODS GRAPHICS Statement**

Enables or disables ODS Graphics processing and sets graphics environment options. This statement affects ODS template-based (ODS Graphics) graphics only. The ODS GRAPHICS statement does not affect device-based graphics (SAS/GRAPH).

- **Valid in:** Anywhere
- **Category:** ODS: Output Control
- **Default:** ON. Beginning in SAS 9.4, ODS Graphics is enabled by default on all platforms except z/OS. When running SAS in batch mode, the default is OFF.
- **Interaction:** SAS/GRAPH device-based global statements such as GOPTIONS, SYMBOL, PATTERN, AXIS, and LEGEND do not affect template-based graphics. The ODS GRAPHICS statement does not affect device-based graphics.
- **See:** For information about common tasks for managing ODS Graphics output, see SAS Graph Template Language: User’s Guide.

Syntax

ODS GRAPHICS <OFF | ON> </option(s)> ;

Summary of Optional Arguments

- **ANTIALIAS | NOANTIALIAS | ANTIALIAS= ON | OFF**
  specifies whether anti-aliasing is applied to the rendering of the line and markers in any graph.

- **ANTIALIASMAX= n**
  specifies the maximum number of graphics elements before anti-aliasing is disabled.

- **ATTRPRIORITIY=COLOR | NONE**
  specifies a priority for cycling of the group attributes.

- **BORDER | NOBORDER | BORDER=ON | OFF**
  specifies whether to draw a border around each graph.

- **BYLINE=NOBYLINE | TITLE | FOOTNOTE**
  specifies how the BY line is displayed in graphs.

- **DATASKINMAX=n**
  specifies the maximum number of skinned graphical elements allowed per plot.

- **DISCRETEMAX=n**
  specifies the maximum number of discrete values to be shown in any graph.

- **DRILLTARGET=":_blank" | ":_self" | ":_parent" | ":_top" | "frame-name"**
  specifies the window that displays the drill-down output.

- **GROUPMAX=n**
  specifies the maximum number of group values to be shown in any graph.

- **HEIGHT=dimension**
  specifies the height of a graph.

- **IMAGEMAP | NOIMAGEMAP | IMAGEMAP=ON | OFF**
  specifies whether data tips are generated.

- **IMAGENAME=":filename"**
  specifies the base image filename.

- **LABELMAX= n**
  specifies the maximum number of labeled areas before labeling is disabled.

- **LABELPLACEMENT= GREEDY | SA**
  specifies the label-placement algorithm to use for positioning labels in the graphs.

- **LEGENDAREAMAX= n**
  specifies an integer that is interpreted as the maximum percentage of the overall graphics area that a legend can occupy.

- **LOESSOBSMAX= n**
  specifies an upper limit for the number of observations that can be used with a loess plot.

- **OUTPUTFMT= file–type | STATIC**
  specifies the output format used to generate image or vector graphic files.

- **PANELCELLMAX=n**
  specifies the maximum number of cells in a graph panel where the number of cells is determined dynamically by classification variables.

- **PUSH | POP**
  pushes and pops ODS GRAPHICS settings in a stack.
RESET | RESET= option
Reset one or more ODS GRAPHICS options to its default.

SCALE | NOSCALE | SCALE=ON | OFF
specifies whether the content of any graph is scaled proportionally.

SCALEMARKERS | NOSCALEMARKERS | SCALEMARKERS=ON | OFF
specifies whether the plot markers are to be scaled with the graph size.

SHOW
writes the current ODS GRAPHICS settings to the SAS log.

STACKDEPTHMAX=n
specifies the maximum stack depth for PUSH and POP requests.

SUBPIXEL | NOSUBPIXEL | SUBPIXEL=ON | OFF
specifies whether subpixel rendering should be used for rendering ODS Graphics.

TIPMAX=n
specifies the maximum number of distinct mouse-over areas allowed before data tips are disabled.

WIDTH=dimension
specifies the width of any graph.

Without Arguments
If the ODS automatic graphic capabilities are currently disabled, then specifying the ODS GRAPHICS statement without options enables them. If the ODS automatic graphic capabilities are currently enabled, then specifying the ODS GRAPHICS statement leaves them enabled.

Required Arguments

ON
enables ODS Graphics processing. This is the default if no argument is used.

Note: Beginning in SAS 9.4, ODS Graphics is enabled by default on all platforms except z/OS.

Alias YES

OFF
disables ODS Graphics processing.

Alias NO

Optional Arguments

ANTIALIAS | NOANTIALIAS | ANTIALIAS= ON | OFF
specifies whether anti-aliasing is applied to the rendering of the line and markers in any graph. Anti-aliasing smooths the appearance of lines and some markers. Text displayed in the graph is always anti-aliased. For graphical displays that plot large numbers of points it is recommended that ANTIALIAS=OFF be specified for performance considerations.

ANTIALIAS
smoothes jagged edges of all components in the graph.

NOANTIALIAS
does not smooth jagged edges of components other than text in the graph.
ANTIALIAS=ON | OFF
specifies whether anti-aliasing is applied to the rendering of the line and markers in the graph.

ON
smoothes jagged edges of all components in the graph.

Alias YES

OFF
does not smooth jagged edges of components other than text in the graph.

Alias NO

Default ANTIALIAS or ANTIALIAS=ON | YES

Restriction If the number of markers or lines in the plot exceeds the number specified by the ANTIALIASMAX= option, then the ANTIALIAS option is turned off. This is true even if you specify the option ANTIALIAS=ON or ANTIALIAS.

ANTIALIASMAX= \( n \)
specifies the maximum number of graphics elements before anti-aliasing is disabled. For example, if there are more than 400 scatter point markers to be anti-aliased and ANTIALIASMAX=400, then no markers are anti-aliased. The default value is 4000.

Note: Prior to the third maintenance release of SAS 9.4, the ANTIALIASMAX= option specifies the maximum number of observations in the graph data to be anti-aliased before anti-aliasing is disabled. The default is 4000. When the graph data contains more than 4000 observations, anti-aliasing is disabled for the entire graph. Starting with the third maintenance release of SAS 9.4, the ANTIALIASMAX= option specifies the maximum number of graphics elements to be anti-aliased in each plot on a per-plot basis. The default remains at 4000. If any plot in a graph contains more than 4000 elements, anti-aliasing is disabled for that plot. Anti-aliasing is enabled for the rest of the graph in that case.

\( n \)
specifies a positive integer.

Default 4000

ATTRPRIORITY=COLOR | NONE
specifies a priority for cycling of the group attributes.

COLOR
assigns priority to the color attribute rotation by cycling through the list of colors while holding the marker symbol and line pattern constant. When all of the colors are exhausted, the marker symbol and line style attributes increment to the next element, and then the colors in the list are repeated. This pattern repeats as needed.

NONE
does not use an attribute priority in the rotation pattern, even if one is set in the active style’s AttrPriority attribute. The rotation pattern cycles progressively through the attribute lists.

Default The AttrPriority attribute of the graph style element, or NONE if the current style does not define the AttrPriority style attribute.
Interaction

The default lists of data colors, contrast colors, marker symbols, and line patterns are set in the active style’s GraphData1–GraphDataN elements.

**BORDER | NOBORDER | BORDER=ON | OFF**

specifies whether to draw a border around each graph.

**BORDER**

specifies whether to draw a border around the graph.

**NOBORDER**

specifies not to draw a border around any graph.

**BORDER=ON | OFF**

specifies whether to draw the graph with a border on the outermost layout.

**ON**

specifies to draw a border around the graph.

Alias **YES**

**OFF**

specifies not to draw a border around the graph.

Alias **NO**

Default **BORDER or BORDER=ON | YES**

**BYLINE=NOBYLINE | TITLE | FOOTNOTE**

specifies how the BY line is displayed in graphs when an analysis is run with a BY statement. By default, no BY line is displayed.

The following code is an example of how the placement of the BY line is controlled in each graph template:

```sas
if (_BYTITLE_)
   entrytitle _BYLINE_ / textattrs=GraphValueText;
else
   if (_BYFOOTNOTE_)
      entryfootnote halign=left _BYLINE_;
   endif;
endif;
```

You can modify the graph template if you want to change how the BY line is displayed. Because most graphs have titles and few graphs have footnotes, the BY line looks better when it is displayed as a footnote. For complete documentation about the Graph Template Language, see *SAS Graph Template Language: User's Guide*.

When the **BYLINE=** option is specified, and there are BY groups, ODS creates a BY line and sets the appropriate special dynamic variables. The following table lists the special dynamic variables for BY lines. For complete documentation about special dynamic variables, see “Special Dynamic Variables” in *SAS Graph Template Language: User's Guide*. 
Table 6.6  Special Dynamic Variables for BY Lines

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>BYFOOTNOTE</em></td>
<td>This variable is set to 1 when you specify a BY statement and the ODS GRAPHICS BYLINE= option is set to FOOTNOTE. Otherwise, the variable is set to 0 or is NULL.</td>
</tr>
<tr>
<td><em>BYTITLE</em></td>
<td>This variable is set to 1 when you specify a BY statement and the ODS GRAPHICS BYLINE= option is set to TITLE. Otherwise, the variable is set to 0 or is NULL.</td>
</tr>
</tbody>
</table>

The variables in the table are set automatically only for analytical procedures that support ODS GRAPHICS. For a list of these procedures, see “Automatic Graphics from SAS Analytical Procedures” in SAS Graph Template Language: User’s Guide. For all other procedures, the variables are not set automatically (NULL).

NOBYLINE
specifies that no BY line is displayed. NOBYLINE is the default.

FOOTNOTE
specifies that the BY line is displayed as a left-justified graph footnote. This is the recommended setting.

TITLE
specifies that the BY line is displayed as a centered graph title. Specifying TITLE is not recommended because graphs are not designed to have additional title lines.

Default  NOBYLINE

DATASKINMAX=\textit{n}
specifies the maximum number of skinned graphical elements allowed per plot.

Note: This feature applies to the first maintenance release of SAS 9.4 and to later releases.

\textit{n} specifies a positive integer.

Default  200

DISCRETEMAX=\textit{n}
specifies the maximum number of discrete values to be shown in any graph. Bar charts and box plots are examples of affected plot types. Scatter plots and other plot types can be affected if the data to be plotted is discrete or the axis is discrete.

\textit{n} specifies a positive integer.

Default  1000

Tips
Some plot layers might be unaffected by the DISCRETEMAX= option, and those layers are rendered. If all layers are affected, a blank graph is rendered.

If the value specified by the DISCRETEMAX= option is exceeded by any plot layer in the graph, that layer is not drawn and a warning message is issued.
DRILLTARGET="_blank" | "_self" | "_parent" | "_top" | "frame-name"
specifies the window that displays the drill-down output.

*Note:* This option is supported only for HTML.

"_blank"
opens a new browser window to display the drilldown output.

**Default**
_blank is the default.

**Requirements**
- You must enclose _blank in quotation marks.
- You must specify _blank in lowercase.

"_self"
opens the drill-down output in the same window.

**Requirements**
- You must enclose _self in quotation marks.
- You must specify _self in lowercase.

"_parent"
opens the drill-down output in the parent frame.

**Requirements**
- You must enclose _parent in quotation marks.
- You must specify _parent in lowercase.

"_top"
opens the drill-down output in the full body of the window.

**Requirements**
- You must enclose _top in quotation marks.
- You must specify _top in lowercase.

"frame-name"
opens the drill down output in the named frame in the current window. If the name does not exist, the output is opened in a new window.

**Requirement**
- You must enclose frame-name in quotation marks.

**GROUPMAX=n**
specifies the maximum number of group values to be shown in any graph. Any graph that supports the GROUP= option is affected.

**n**
specifies a positive integer.

**Default**
1000

**Tip**
If the value specified by the GROUPMAX= option is exceeded by any plot layer in the graph, that layer is rendered. The system ignores the GROUP= option and issues a warning message.

**HEIGHT=dimension**
specifies the height of a graph.

**dimension**
is a nonnegative number followed by one of these units of measure:
Table 6.7  Units of Measure for Dimension

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>Centimeters</td>
</tr>
<tr>
<td>in</td>
<td>Inches</td>
</tr>
<tr>
<td>mm</td>
<td>Millimeters</td>
</tr>
<tr>
<td>pct or %</td>
<td>Percentage</td>
</tr>
<tr>
<td>pt</td>
<td>Point size (72 points = 1 inch)</td>
</tr>
<tr>
<td>px</td>
<td>Pixels</td>
</tr>
</tbody>
</table>

Defaults  The value of the SAS registry entry "ODS > ODS GRAPHICS > Design Height" or the value of the DesignHeight= option in a STATGRAPH template. Typically, the value is 480px.

For the PRINTER destination, units of 1/150 of an inch

Tip  If only the HEIGHT= option is specified, then the default aspect of the graph is maintained.

IMAGEMAP | NOIMAGEMAP | IMAGEMAP=ON | OFF
controls data tips and drill down generation. Data tips are pieces of explanatory text that appear when you hold the mouse pointer over the data portions of a graph contained in an HTML page.

IMAGEMAP  specifies to generate data tips.

NOIMAGEMAP  specifies not to generate data tips.

IMAGEMAP= ON | OFF  controls data tips generation.

ON  specifies to generate data tips.

Alias  YES

OFF  specifies not to generate data tips.

Alias  NO

Default  NOIMAGEMAP or IMAGEMAP=OFF | NO

Restrictions  This option applies only when the ODS HTML destination is used.

An image map is not generated using SVG with ODS Graphics. The image map data that is used to produce tooltips and links is written directly in the SVG and is not part of the HTML. Using HTML5 with
When IMAGEMAP | IMAGEMAP=ON is specified and the ODS HTML destination is used, the IMAGE_DPI option in the ODS HTML destination is ignored, if specified, and the default image resolution of 96 DPI is used.

**IMAGENAME= "filename"**

specifies the base image filename. If more than one image is generated, each is assigned filename as a base name followed by a number in order to create unique names. This numbering can be reset with the RESET=INDEX option. Path information (if needed) can be set with the GPATH= option on the ODS destination statement. The default path is the current output directory. A file extension for filename is automatically generated based on the OUTPUTFMT= option.

**Default**
The name of the output object.

**Restriction**
filename must be a single name. It must not include any path specification or image-format name extension.

**Requirement**
You must enclose filename in quotation marks.

**See**
“Specifying and Resetting the Image Name” on page 346

**LABELMAX= n**
specifies the maximum number of labeled areas before labeling is disabled. For example, if there are more than 50 points to be labeled and LABELMAX=50, then no points are labeled.

\[ n \]
specifies a positive integer.

**Default**
200

**Restriction**
Data label collision avoidance is turned off under the following conditions:
- The number of observations with nonmissing labels exceeds the value specified by LABELMAX=.
- The number of observations exceeds five times the value specified by LABELMAX=.
A message is then sent to the SAS log.

**Tip**
To turn off collision avoidance specify LABELMAX=0.

**LABELPLACEMENT= GREEDY | SA**
specifies the label-placement algorithm to use for positioning labels in the graphs. The following labels are affected:
- data labels for needle plots, scatter plots, series plots, step plots, and vector plots
- vertex labels for line charts
- curve labels when the curve label is positioned at the start or end of the curve

**GREEDY**
specifies the Greedy method for managing label collision. The Greedy method tries different placement combinations in order to find an optimal approximation that avoids collisions. Label placement using this method is often less optimal
than label placement using the Simulated Annealing (SA) method. However, depending on the number of data points and the potential for label collisions, the Greedy process can be significantly faster.

**SA**

specifies the Simulated Annealing method for managing label collision. The SA method attempts to determine the global minimization-of-cost function, which is based on a simulated annealing algorithm. The resulting label placement is usually better than placement using the Greedy method. However, depending on the number of data points and the potential for label collisions, the SA method can be significantly slower.

**Restriction**

For BANDPLOT and LINECHART, the SA method has no effect on the curve labels when the CURVELABELPOSITION= option specifies START or END.

**Default**

**GREEDY**

**LEGENDAREAMAX= n**

specifies an integer that is interpreted as the maximum percentage of the overall graphics area that a legend can occupy.

*Note:* Starting with the third maintenance release for SAS 9.4, LEGENDAREAMAX= replaces MAXLEGENDAREA=. However, MAXLEGENDAREA= is supported as an alias. It is recommended that you use LEGENDAREAMAX=.

**n**

specifies a positive integer.

**Default**

20

**Range**

0–100

**Tip**

To turn off the legend, specify LEGENDAREAMAX=0. No warning is issued when the legend is turned off in this way.

**LOESSOBSMAX= n**

specifies an upper limit for the number of observations that can be used with a loess plot.

*Note:* Starting with the third maintenance release for SAS 9.4, LOESSOBSMAX= replaces LOESSMAXOBS=. However, LOESSMAXOBS= is supported as an alias. It is recommended that you use LOESSOBSMAX=.

If the number of observations of the loess plot exceeds the specified limit, the loess plot is not drawn.

For example, the following specifies that the most observations a loess plot can have is 1000.

**LOESSOBSMAX=1000**

**Alias**

LOESSMAXOBS=

**Default**

5000
OUTPUTFMT= file-type | STATIC
specifies the output format used to generate image or vector graphic files. If the image or vector graphic format is not valid for the active output destination, the format is automatically changed to the default format for that destination.

file-type
is the image or vector graphic format to be generated. See “Supported File Types for Output Destinations” on page 349.

STATIC
uses the best quality static image format for the active output destination. This is the default output format.

TIP The STATIC keyword can be used to reset the output format to its default state.

Default STATIC

See “Specifying the Image Format” on page 347

PANELCELLMAX=n
specifies the maximum number of cells in a graph panel where the number of cells is determined dynamically by classification variables. If the number of cells in the panel exceeds the specified limit, the panel is not drawn.

n
specifies a positive integer.

Default 10000

Note Graphs with DataPanel or DataLattice templates layouts are affected. In the ODS Graphics Procedures, this option affects graphs that are created with the SGPANEL procedure. If the value specified by the PANELCELLMAX= option is exceeded by any of these layouts, an empty graph is rendered and a warning message is issued.

PUSH | POP
pushes and pops ODS GRAPHICS settings in a stack. This feature enables you to temporarily save your custom settings in a stack and later restore those settings.

Note: This feature applies to the third maintenance release of SAS 9.4 and to later releases.

PUSH
pushes the current ODS GRAPHICS settings to a stack.

POPCo
restores the most recently pushed settings from the stack. For each PUSH action, you can specify a POP request. ODS issues a warning if you specify POP without a corresponding PUSH. In that case, nothing is popped because nothing has been pushed.

The pushed settings remain in the stack in the current SAS session until they are popped or the stack is emptied.

Interaction You can specify PUSH as many times as you like up to the limit that is defined by the STACKDEPTHMAX= option. You can also use STACKDEPTHMAX= to empty the stack. For more information, see “Managing the Stack Depth” on page 352.
Note  Order of specification is important when using the PUSH and POP options. For more information, see “About PUSH and POP” on page 351.

Tip  Use the SHOW option to show the current ODS GRAPHICS settings.

See  “Temporarily Saving and Restoring ODS GRAPHICS Settings” on page 351

**RESET | RESET= option**
Reset one or more ODS GRAPHICS options to its default.

**RESET**
 resets all options to their defaults.

**RESET=**
 resets one of the following to its default:

- **ALL**
  resets all reset-options to their defaults.

- **ANTIALIAS**
  resets the ANTIALIAS= option to its default.

  See  **ANTIALIAS=** on page 330

- **ANTIALIASMAX**
  resets the ANTIALIASMAX= option to its default.

  See  **ANTIALIASMAX=** on page 331

- **ATTRPRIORITY**
  resets the ATTRPRIORITY= option to its default.

  See  **ATTRPRIORITY=** on page 331

- **BORDER**
  resets the BORDER= option to its default.

  See  **BORDER=** on page 332

- **BYLINE**
  resets the BYLINE= option to its default.

  See  **BYLINE=** on page 332

- **DATASKINMAX**
  resets the DATASKINMAX= option to its default.

  See  **DATASKINMAX=** on page 333

- **DISCRETEMAX**
  resets the DISCRETEMAX= option to its default.

  See  **DISCRETEMAX=** on page 333

- **DRILLTARGET**
  resets the DRILLTARGET= option to its default.
GROUPMAX

resets the GROUPMAX= option to its default.

See GROUPMAX= on page 334

HEIGHT

resets the HEIGHT= option to its default.

See HEIGHT= on page 334

IMAGEMAP

resets the IMAGEMAP= option to its default.

Note: Not all output destinations support this feature.

See IMAGEMAP= on page 335

IMAGENAME

resets the IMAGENAME= option to its default.

Note: This feature applies to the third maintenance release of SAS 9.4 and to later releases.

See IMAGENAME= on page 336

INDEX <(+positive-integer)> reset the index counter that is appended to static image files.

When specifying this option, you can also specify the value for the index counter. The number that you specify must be enclosed in parentheses. positive-integer determines the suffix for the next subsequent image, and increments with each new image. This feature applies to the third maintenance release of SAS 9.4 and to later releases.

See “Resetting the Image Name” on page 346

LABELMAX

resets the LABELMAX= option to its default.

See LABELMAX= on page 336

LABELPLACEMENT

specifies the label-placement algorithm to use for positioning labels in the graphs.

See LABELPLACEMENT= on page 336

LEGENDAREAMAX

resets the LEGENDAREAMAX= option to its default.

See LEGENDAREAMAX= on page 337

LOESSOBSMAX

resets the LOESSOBSMAX= option to its default.

See LOESSOBSMAX= on page 337
OUTPUTFMT
resets the OUTPUTFMT= option to its default.

Note: This feature applies to the third maintenance release of SAS 9.4 and to later releases.

See OUTPUTFMT= on page 338

PANELCELLMAX
resets the PANELCELLMAX= option to its default.

See PANELCELLMAX= on page 338

SCALE
resets the SCALE= option to its default.

See SCALE= on page 341

SCALEMARKERS
resets the SCALEMARKERS= option to its default.

See SCALEMARKERS= on page 342

STACKDEPTHMAX
resets the STACKDEPTHMAX= option to its default.

Note: This feature applies to the third maintenance release of SAS 9.4 and to later releases.

See STACKDEPTHMAX= on page 343

SUBPIXEL
resets the SUBPIXEL option to its default.

Note: This feature applies to the third maintenance release of SAS 9.4 and to later releases.

See SUBPIXEL on page 344

TIPMAX
resets the TIPMAX= option to its default.

See TIPMAX = on page 345

WIDTH
resets the WIDTH= option to its default.

See WIDTH= on page 345

SCALE | NOSCALE | SCALE=ON | OFF
specifies whether the content of any graph is scaled proportionally.

SCALE
scales the components of graph proportionally.

NOSCALE
does not scale the components of graph proportionally.

SCALE=ON | OFF
specifies whether the content of the graph is scaled proportionally.
ON
scales the components of graph proportionally.

Alias  YES

OFF
does not scale the components of graph proportionally.

Aliases  NOSCALE
NO

Default  SCALE or SCALE=ON | YES

SCALEMARKERS | NOSCALEMARKERS | SCALEMARKERS=ON | OFF
specifies whether the plot markers are to be scaled with the graph size. The scaling factor is based on the height of the graph cells and the height of the graph.

SCALEMARKERS
scales the markers with the graph size.

NOSCALE
does not scale the markers with the graph size.

SCALEMARKERS=ON | OFF
specifies whether the plot markers are to be scaled with the graph size.

ON
scales the markers with the graph size.

Alias  YES

OFF
does not scale the markers with the graph size.

Aliases  NOSCALE
NO

Default  SCALEMARKERS or SCALEMARKERS=ON | YES

Restriction  Scaling is done only if the graph contains multiple cells or single nested cells.

SHOW
writes the current ODS GRAPHICS settings to the SAS log. This option enables you to verify which settings are in effect. The option is especially useful when you use the PUSH and POP options to restore settings. For more information, see “Temporarily Saving and Restoring ODS GRAPHICS Settings” on page 351.

Note:  This feature applies to the third maintenance release of SAS 9.4 and to later releases.

If no options have been specified, then SHOW lists those options for which ODS currently knows the default values.

The following statement resets all settings and shows the default values.

ods graphics / reset=all show;
Here are the default values displayed in the SAS log:

Here are the default values plus the image width, as displayed in the SAS log:

Tip If you have specified the settings for some options but want to see the default values without losing your specified settings, issue the following two statements. The first statement pushes your specified settings, resets all settings, and then lists options for which ODS currently knows the default values. The second statement restores your previous settings.

STACKDEPTHMAX=n

specifies the maximum stack depth for PUSH and POP requests. The stack is used to temporarily store ODS GRAPHICS settings when you issue PUSH requests. PUSH saves the current settings to the stack and increments the stack depth. POP restores the most recently saved settings from the stack and decrements the stack depth.

Note: This feature applies to the third maintenance release of SAS 9.4 and to later releases.

n specifies a positive integer.

If n is less than the current stack depth, then the stack is popped until its depth equals n. Popping the stack does not affect other option settings.

Defaults 1024 is the default maximum depth
0 is the default depth

Tips
To empty the stack and then reset it to the default maximum depth, issue the following statement:

```plaintext
ods graphics / stackdepthmax=0 reset=stackdepthmax;
```

You can use any of the following commands to reset the stack to its default maximum depth:

```plaintext
reset=stackdepthmax
reset=all
reset
stackdepthmax=1024
```

See “Managing the Stack Depth” on page 352

**SUBPIXEL | NOSUBPIXEL | SUBPIXEL=ON | OFF**
specifies whether subpixel rendering should be used for rendering ODS Graphics. Subpixel rendering produces smoother curves and more precise bar spacing.

*Note:* This feature applies to the third maintenance release of SAS 9.4 and to later releases.

**SUBPIXEL**
always uses subpixel rendering, when applicable, for rendering lines and bars.

**NOSUBPIXEL**
ever uses subpixel rendering.

**SUBPIXEL=ON | OFF**
specifies whether subpixel rendering should be used.

**ON**
always uses subpixel rendering, when applicable, for rendering lines and bars.

Alias **YES**

**OFF**
ever uses subpixel rendering.

Alias **NO**

Default Subpixel rendering is always enabled for vector-graphics output. It is enabled by default for image output, unless the graph contains a scatter plot or a scatter-plot matrix. In those cases, subpixel rendering is disabled by default.

Requirement Antialiasing must be enabled for this option to have any effect. Antialiasing is enabled by default. To re-enable antialiasing, use the ANTIALIAS=ON option in the ODS GRAPHICS statement.

Tip For a large amount of data, antialiasing is disabled when the number of observations exceeds the default maximum of 4000 observations. In that case, subpixel rendering is also disabled. To increase the maximum, use the ANTIALIASMAX= option in the ODS GRAPHICS statement.
TIPMAX=n
specifies the maximum number of distinct mouse-over areas allowed before data tips are disabled. For example, if there are more than 400 points in a scatterplot, and TIPMAX=400, then no data tips appear. The default maximum value is 500.

Note: Prior to the third maintenance release of SAS 9.4, the TIPMAX= option specifies the maximum number of observations in the graph data to be allowed before data tips are disabled. The default is 500. When the graph data contains more than 500 observations, data tips are disabled for the entire graph. Starting with the third maintenance release of SAS 9.4, the TIPMAX= option specifies the maximum number of mouse-over areas allowed before data tips are disabled. This threshold is applied separately for each plot. The default remains at 500. If any plot in a graph contains more than 500 mouse-over areas, data tips are disabled for that plot. Data tips are enabled for the remaining plots in the graph.

n
specifies a positive integer.

Default 500

WIDTH=dimension
specifies the width of any graph.

dimension
is a nonnegative number followed by one of these units of measure:

Table 6.8 Units of Measure for Dimension

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>Centimeters</td>
</tr>
<tr>
<td>in</td>
<td>Inches</td>
</tr>
<tr>
<td>mm</td>
<td>Millimeters</td>
</tr>
<tr>
<td>pct or %</td>
<td>Percentage</td>
</tr>
<tr>
<td>pt</td>
<td>Point size (72 points = 1 inch)</td>
</tr>
<tr>
<td>px</td>
<td>Pixels</td>
</tr>
</tbody>
</table>

Defaults The value of the SAS registry entry "ODS > ODS GRAPHICS > Design Width" or the value of the DesignWidth= option in a STATGRAPH template. Typically, this value is 640px.

For the PRINTER destination, units of 1/150 of an inch

Tip If only the WIDTH= option is specified, then the default aspect of the graph is maintained.
Details

Using the ODS GRAPHICS Statement
You can enable ODS Graphics by using one of the following equivalent statements:

ods graphics on;
ods graphics;

When you specify one of these statements before your procedure invocation, Base, SAS/STAT, SAS/ETS, and SAS/QC procedures support ODS Graphics, either by default, or when you specify procedure options for requesting particular graphs.

To disable ODS Graphics, specify the following statement:

ods graphics off;

Note: ODS Graphics is ON by default for procedures SGPLOT, SGPANEL, SGSCATTER, SGDESIGN, and SGRENDER. For other products, the initial state of ODS Graphics is determined by a SAS Registry setting.

Using the ODS GRAPHICS Statement for Batch Jobs
To generate device-based graphics output in UNIX batch jobs, you must set the DISPLAY system option before creating the output. To set the display, enter the following command:

export DISPLAY=<ip_address>:0

The ip_address is the TCP/IP address, or the name of a UNIX terminal. Usually, the IP address of the UNIX system where SAS is running would be used. If you do not set the DISPLAY variable, then you get an error message in the SAS log.

Specifying and Resetting the Image Name

Specifying the Image Name
For ODS Graphics output, by default, the ODS object name is used as the “root” name for the image output file. The following example creates a GIF image named REGPLOT:

ods graphics / imagename="regplot" outputfmt=gif;

The assigned name REGPLOT is treated as a "root" name and the first output created is named REGPLOT. Subsequent graphs are named REGPLOT1, REGPLOT2, and so on, with an increasing index counter. This numbering can be reset with the RESET=INDEX option.

Resetting the Image Name
The RESET=INDEX option enables you to reset the filename numbering sequence. For example, if you are developing a template and it takes several submissions to get the desired output, you can use the RESET or RESET=INDEX option to force each output to replace itself:

ods graphics / reset=index ... ;

This specification causes all subsequent images to be created with the default or current image name.

When specifying this option, you can also specify the value for the index counter. The value that you specify determines the suffix for the next subsequent image. For example:

ods graphics / reset=index(100) imagename="MyName";
The next graph that you produce is named MYNAME100.

This feature is useful for creating animated graphics. For example, for a sequence of 100 images, you might begin with the following statement:

```R
ods graphics / reset=index(1) imagename="MyName";
```

In the example, your program produces 100 images named MYNAME1, MYNAME2, ..., MYNAME100. If you later add more images to the animation, you might submit the following:

```R
ods graphics / reset=index(101) imagename="MyName";
```

The next generated image is named MYNAME101.

*Note:* The ability to specify the value for the index counter applies to the third maintenance release in SAS 9.4 and later releases.

### Specifying the Image Format

Each ODS destination uses a default format for its output. You can use the OUTPUTFMT= option in the ODS GRAPHICS statement to change the output format.

*Note:* Unless you have a special requirement for changing the image format, we recommend that you not change it. The default PNG or vector graphic format is far superior to other formats, such as GIF, in support for transparency and a large number of colors. Also, PNG and vector graphics images require much less disk storage space than JPEG or TIFF formats.

If you want to generate vector graphics images, you can use the following OUTPUTFMT= values for each destination:

**Table 6.9 Generating Vector Graphics Output with ODS**

<table>
<thead>
<tr>
<th>ODS Destination</th>
<th>OUTPUTFMT=value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODS EPUB</td>
<td>OUTPUTFMT=SVG</td>
</tr>
<tr>
<td>ODS destination for Excel</td>
<td>OUTPUTFMT=EMF</td>
</tr>
<tr>
<td>ODS HTML</td>
<td>OUTPUTFMT=SVG</td>
</tr>
<tr>
<td>ODS LISTING</td>
<td>OUTPUTFMT=EMF</td>
</tr>
<tr>
<td></td>
<td>OUTPUTFMT=PDF</td>
</tr>
<tr>
<td></td>
<td>OUTPUTFMT=PS</td>
</tr>
<tr>
<td></td>
<td>OUTPUTFMT=SVG</td>
</tr>
<tr>
<td></td>
<td>OUTPUTFMT=PCL</td>
</tr>
<tr>
<td>ODS PDF</td>
<td>Vector graphics images are generated by default</td>
</tr>
<tr>
<td>ODS PCL</td>
<td>OUTPUTFMT=PCL (for PCL output)</td>
</tr>
<tr>
<td>ODS PS</td>
<td>OUTPUTFMT=PS (for PostScript output)</td>
</tr>
<tr>
<td>ODS destination for PowerPoint</td>
<td>OUTPUTFMT=EMF</td>
</tr>
</tbody>
</table>
When a vector graphics image cannot be generated for the format that you specify, a PNG image is generated instead and is embedded in the specified output file. The output file format and extension are not changed in that case. In the following cases, a vector graphics image cannot be generated:

- surface plots
- bivariate histograms
- graphs that use smooth gradient contours
- graphs that include continuous legends
- graphs that use data skins
- graphs that use transparency (EMF and PS ODS destinations only)
- graphs that contain one or more rotated images
- graphs that have a broken axis
- graphs that contain outline marker characters

Starting with the second maintenance release of SAS 9.4, additional cases for which vector graphics output cannot be generated for graphs are as follows:

- graphs that use gradient fill for bars in a bar chart, histogram, or waterfall chart
- graphs that use the back-light effect on text
- graphs that include a text plot that displays text with an outlined bounding box or text with a filled bounding-box background
- graphs that include images (PostScript output only)

Starting with the third maintenance release of SAS 9.4, vector graphics output can be generated in the EMF, PDF, and SVG output formats for the following cases:

- graphs that use data skins

  Note: For the EMF, PDF, and SVG formats, vector graphics output is not supported for graphs that use transparency and data skins. An image is generated in that case.

- graphs that include one or more rotated images
- graphs that use gradient fills (except PDF)
- graphs that use a continuous legend
Note: For the PDF output format, vector graphics output is not supported for graphs that use a continuous legend and data transparency. An image is generated in that case.

**Supported File Types for Output Destinations**
The following table lists all of the supported file types for some ODS output destinations.

**Table 6.10  Supported File Types for Output Destinations**

<table>
<thead>
<tr>
<th>Output Destination</th>
<th>Supported File Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPUB, EPUB2</td>
<td>PNG (default), GIF, JPG, SVG</td>
</tr>
<tr>
<td>EPUB3</td>
<td>SVG (default), PNG, GIF, JPG</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> EPUB3 was added in the first maintenance release of SAS 9.4.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Starting with the third maintenance release of SAS 9.4, EPUB3 is an alias for EPUB, and the EPUB3 supported file types supersede the EPUB supported file types.</td>
</tr>
<tr>
<td>ODS destination for Excel</td>
<td>PNG (default), JPEG, JPG, EMF</td>
</tr>
<tr>
<td>HTML</td>
<td>PNG (default), GIF, JPEG, JPG, SVG</td>
</tr>
<tr>
<td>HTML5</td>
<td>SVG (default), PNG, GIF, JPEG, JPG</td>
</tr>
<tr>
<td>LISTING</td>
<td>PNG (default), BMP, EMF, EPS, GIF, JFIF, JPEG, JPG, PDF, PS, SASEMF, STATIC, TIFF, WMF, PSL, SVG</td>
</tr>
<tr>
<td>PDF</td>
<td>Native PDF (default), JPEG, JPG, GIF, PNG</td>
</tr>
<tr>
<td>ODS destination for PowerPoint</td>
<td>PNG (default), JPEG, JPG, GIF, EMF, TIFF, BMP</td>
</tr>
<tr>
<td>PS</td>
<td>PNG (default), JPEG, JPG, GIF, EPS, PDF, PCL, PS</td>
</tr>
<tr>
<td>RTF and Measured RTF</td>
<td>EMF (default), PNG, JPEG, JPG, JFIF</td>
</tr>
</tbody>
</table>

**Description of Supported File Types**
The following table provides descriptions of the supported file types for ODS output destinations.

**Table 6.11  Description of Supported File Types**

<table>
<thead>
<tr>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP (Microsoft Windows Device Independent Bitmap)</td>
<td>Supports color-mapped and true color images that are stored as uncompressed or run-length encoded data. BMP was developed by Microsoft Corporation.</td>
</tr>
<tr>
<td>File Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>CGM (Computer Graphics Metafile)</td>
<td>A free and open international standard file format for 2-D vector graphics, raster graphics, and text. This format is defined by ISO/IEC 8632.</td>
</tr>
<tr>
<td>DIB (Microsoft Windows Device Independent Bitmap)</td>
<td>See the description of BMP.</td>
</tr>
<tr>
<td>EMF Plus (Enhanced Metafile Format Plus Extensions)</td>
<td>Supports Enhanced Metafile Plus Extensions that provides additional functionality, such as support of RGBA colors.</td>
</tr>
<tr>
<td>EMF Dual (Enhanced Metafile Format and Enhanced Metafile Format Plus Extensions)</td>
<td>Produces both EMF and EMF Plus formats simultaneously in the same output.</td>
</tr>
<tr>
<td>EPS</td>
<td>Encapsulated PostScript</td>
</tr>
<tr>
<td>EPSI (Microsoft NT Enhanced Metafile)</td>
<td>An extended version of the standard PostScript (PS) format. Files that use this format can be printed on PostScript printers and can also be imported into other applications. Notice that EPSI files can be read, but PS files cannot be read.</td>
</tr>
<tr>
<td>GIF (Graphics Interchange Format)</td>
<td>Supports only color-mapped images. GIF is owned by CompuServe, Inc.</td>
</tr>
<tr>
<td>JIF (JPEG File Interchange Format)</td>
<td>Supports JPEG image compression. JIF software is developed by the Independent JPEG Group.</td>
</tr>
<tr>
<td>JPEG or JPG (Joint Photographic Experts Group)</td>
<td>A file format that is used for storing noninteractive images.</td>
</tr>
<tr>
<td>PBM (Portable Bitmap Utilities)</td>
<td>Supports gray-scale, color, RGB, and bitmap files. The Portable Bitmap Utilities are a set of free utility programs that were developed primarily by Jef Poskanzer.</td>
</tr>
<tr>
<td>PCL</td>
<td>Printer Control Language</td>
</tr>
<tr>
<td>PNG (Portable Network Graphic)</td>
<td>Supports true color, gray-scale, and 8-bit images.</td>
</tr>
<tr>
<td>PS (PostScript Image File Format)</td>
<td>The Image classes use only PostScript image operators. A level II PS printer is required for color images. PostScript was developed by Adobe Systems, Inc.</td>
</tr>
</tbody>
</table>
File Type | Description
--- | ---
PSL (PostScript) | PostScript
STATIC | Chooses the best image format for the current ODS destination.
SVG (Scalable Vector Graphics) | Is an XML language for describing two-dimensional vector graphics.
TIFF (Tagged Image File Format) | Internally supports a number of compression types and image types, including bitmapped, color-mapped, gray-scaled, and true color. TIFF was developed by Aldus Corporation and Microsoft Corporation and is used by a wide variety of applications (available if licensed).
XBM | X Window Bitmap
XPM | X Window Pixmap

**Temporarily Saving and Restoring ODS GRAPHICS Settings**

This feature applies to the third maintenance release of SAS 9.4 and to later releases.

**About PUSH and POP**

Although you can use the RESET option to restore the default ODS GRAPHICS settings, there might be times when you want to save your current custom settings and later restore them. ODS enables you to temporarily store your custom settings in a stack created for this purpose, perform some other task with different settings, and then restore the previous settings.

The PUSH option saves the current ODS GRAPHICS settings to the stack and increments the stack depth. The POP option restores the most recently stored settings from the stack and decrements the stack depth.

This feature is useful when you run macros. Within a macro you can PUSH at the start of the macro and POP at the end. This enables your macro to have custom ODS GRAPHICS behaviors without affecting the calling environment.

You can specify PUSH as many times as you like up to the limit that is defined by the STACKDEPTHMAX= option. The pushed settings remain in the stack in the current SAS session until they are popped or the stack is emptied. For more information, see “Managing the Stack Depth” on page 352. For each PUSH request, you can specify a POP request. ODS issues a warning if you specify POP without a corresponding PUSH. In that case, nothing is popped because nothing has been pushed to the stack.

Order of specification is important when using the PUSH option. For example, the following statement pushes the NOBORDER option to the stack along with any other custom settings that are in effect.

```ods graphics / noborder push;```

A subsequent POP request restores the pushed settings including NOBORDER.

However, the following statement pushes the current custom settings and then sets the NOBORDER option.
Here, the subsequent POP request restores whatever border setting was in effect when the PUSH request was made.

**TIP** Use the SHOW option to show the ODS GRAPHICS settings that are currently in effect.

**Settings That Can Be Pushed**
The PUSH and POP commands apply to all ODS GRAPHICS options except the following: PUSH, POP, RESET=INDEX, and SHOW.

**How Code Errors Affect the PUSH Operation**
If the ODS GRAPHICS statement contains a syntax error, then the PUSH request is ignored.

For example, the PUSH request is ignored in the following statement:

```ods graphics / antialias=bogus push;```

A syntax error (BOGUS) in ANTIALIAS causes the parser to ignore the remaining options. However, a simple semantics error does not prevent the remaining options from being handled. In the following statement, the PUSH request is honored.

```ods graphics / antialiasmax=-1 push;```

In this statement, ANTIALIASMAX= –1 is invalid. The option expects a zero or a positive integer. In this case, a warning is issued to the log, but the PUSH occurs.

*Note:* Syntax errors in your code can have other unexpected results that are not described here.

**Managing the Stack Depth**
By default, the stack supports up to 1024 pushes. You can change the default by using the STACKDEPTHMAX= option.

If the specified STACKDEPTHMAX= value is less than the current stack depth, then the stack is popped until its depth equals the specified value. Popping the stack does not affect other option settings.

If you want to empty the stack, issue the following statement:

```ods graphics / stackdepthmax=0 reset=stackdepthmax;```

This statement first empties the stack of all PUSH requests and then restores the stack size to 1024.

---

**ODS HTML3 Statement**
Opens, manages, or closes the HTML3 destination, which produces HTML 3.2 formatted output.

**Valid in:** Anywhere  
**Category:** ODS: Third-Party Formatted  
**Default:** The default style for Markup family destinations is HTMLBlue.

**Syntax**

```ods html3<((id=>identifier))<action>;```
ODS HTML3 \(<(\text{ID=} \text{identifier})> \text{option(s)}> ;

**Summary of Optional Arguments**

- **(ID= identifier)**
  Open multiple instances of the same destination at the same time

- **ANCHOR= 'anchor-name'**
  Specify a unique base name for the anchor tag that identifies each output object in the current body file

- **ARCHIVE='string'**
  Specify which applet to use to view ODS HTML output

- **ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)**
  Specify attributes to write between the tags that generate dynamic graphics output

- **BASE= 'base-text'**
  Specify text to use as the first part of all links and references that ODS creates in output files

- **BODY= 'file-specification' (suboption(s))**
  Open a markup family destination and specify the file that contains the primary output that is created by the ODS statement

- **CHARSET= character-set**
  Specify the character set to be generated in the META declaration for the HTML output

- **CLOSE**
  Close the destination and the file that is associated with it

- **CODE= 'file-specification' (suboption(s))**
  Open the HTML destination and specify the file that contains relevant style information

- **CODEBASE='string'**
  Create a file path that can be used by the GOPTIONS devices

- **CONTENTS= 'file-specification' (suboption(s))**
  Open the HTML destination and specify the file that contains a table of contents for the output

- **CSSSTYLE= 'file-specification'<(media-type-1<…media-type-10>)>**
  Specify a cascading style sheet to apply to your output

- **DOM=<"external-file">**
  Specify that the ODS document object model is written to the SAS log or to an external file.

- **ENCODING= local-character-set-encoding**
  Override the encoding for input or output processing (transcodes) of external files

- **EVENT=\textbf{event-name} (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )**
  Specify an event and the value for event variables that is associated with the event

- **EXCLUDE exclusion(s) | ALL | NONE**
  Exclude output objects from the destination

- **FRAME= 'file-specification' (suboption(s))**
  Specify the file that integrates the table of contents, the page contents, and the body file

- **GFOOTNOTE | NOGFOOTNOTE**

Control the location where footnotes are printed in the graphics output

\texttt{GPATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)}

Specify the location for all graphics output that is generated while the destination is open

\texttt{GTITLE | NOGTITLE}

Control the location where titles are printed in the graphics output

\texttt{HEADTEXT= 'markup-document-head'}

Specify HTML tags to place between the < HEAD> and </HEAD> tags in all of the output files.

\texttt{METATEXT= 'metatext-for-document-head'}

Specify HTML code to use as the <META> tag between the <HEAD> and </HEAD> tags in all of the HTML output files.

\texttt{NEWFILE= starting-point}

Create a new body file at the specified starting point

\texttt{OPTIONS ( DOC= | <suboption(s)> )}

Specify tagset-specific suboptions and a named value

\texttt{PACKAGE <package-name>}

Specify that the output from the destination be added to an ODS package

\texttt{PAGE= 'file-specification' <(suboption(s))>}

Open the HTML destination and specify the file that contains a description of each page of the body file and contains links to the body file

\texttt{PARAMETERS= (parameter-pair-1 ... parameter-pair-n)}

Write the specified parameters between the tags that generate dynamic graphics output

\texttt{PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)}

Specify the location of an aggregate storage location or a SAS catalog for all markup files

\texttt{RECORD_SEPARATOR= 'alternative-separator' | NONE}

Specify an alternative character or string to separate lines in the output files

\texttt{SELECT selection(s) | ALL | NONE}

Select output objects for the destination

\texttt{SHOW}

Write to the SAS log the current selection or exclusion list for the destination

\texttt{STYLE= style-template}

Specify a style template to use in writing output files

\texttt{STYLESHEET= 'file-specification' <(suboption(s))>}

Open the HTML destination and place style information for output into an external file, or read style sheet information from an existing file

\texttt{TEXT= text-string}

Insert text into your document

\texttt{TRANTAB= 'translation-table'}

Specify a translation table to use when transcoding a file for output

**Without Arguments**

If you use the ODS HTML3 statement without an action or options, then it opens the HTML3 destination and creates HTML3 output.
**Actions**
The following actions are available for the ODS HTML3 statement:

**CLOSE**
closes the destination and any files that are associated with it.

**Tip** When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

**EXCLUDE exclusion(s) | ALL | NONE**
excludes one or more output objects from the destination.

**Default** NONE

**Restriction** A destination must be open for this action to take effect.

**See** “ODS EXCLUDE Statement” on page 321

**SELECT selection(s) | ALL | NONE**
selects output objects for the specified destination.

**Default** ALL

**Restriction** A destination must be open for this action to take effect.

**See** “ODS SELECT Statement” on page 758

**SHOW**
writes the current selection list or exclusion list for the destination to the SAS log.

**Restriction** The destination must be open for this action to take effect.

**Tip** If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

**See** “ODS SHOW Statement” on page 771

**Optional Arguments**
The following options are available for the ODS HTML3 statement, which is part of the markup family of statements:

**ANCHOR='anchor-name’**
specifies a unique base name for the anchor tag that identifies each output object in the current body file.

Each output object has an anchor tag for the contents, page, and frame files to reference. The links and references are automatically created by ODS. The links and references point to the name of an anchor. Therefore, each anchor name in a file must be unique.

*anchor-name* is the base name for the anchor tag that identifies each output object in the current body file.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify **ANCHOR='TABULATE’**, then ODS names the first
anchor **tabulate**. The second anchor is named **tabulate1**; the third is named **tabulate2**, and so on.

**Restrictions**
Each anchor name in a file must be unique.

Only alphanumeric values, the special characters "$ - _ . + ! "( ) , " and reserved characters used for their reserved purposes can be used unencoded within a URL.

**Requirement**
You must enclose *anchor-name* in quotation marks.

**Interaction**
If you open a file to append to it, be sure to specify a new anchor name to prevent writing the same anchors to the file again. ODS does not recognize anchors that are already in a file when it opens the file.

**Tips**
You can change anchor names as often as you want by specifying the ANCHOR= option in a markup family statement anywhere in your program. After you have specified an anchor name, it remains in effect until you specify a new one.

Specifying new anchor names at various points in your program is useful when you want other web pages to link to specific parts of your markup language output. Because you can control where the anchor name changes, you know in advance what the anchor name is at those points.

An *anchor-name* must begin with a letter ([A-Za-z]) and can be followed by any number of letters, digits ([0-9]), hyphens (-), underscores (_), colons (:), and periods (.).

**ARCHIVE=’string’**
specifies which applet to use to view the ODS HTML output. The ARCHIVE= option is valid only for the GOPTIONS Java device.

The string must be one that the browser can interpret. For example, if the archive file is local to the computer that you are running SAS on, you can use the FILE protocol to identify the file. If you want to point to an archive file that is on a web server, use the HTTP protocol.

**Default**
If you do not specify ARCHIVE= and you are using the JAVA device driver, ODS uses the value of the SAS system option APPLETOC=. There is no default if you are using the ACTIVEX device driver.

**Requirements**
You must enclose *string* in quotation marks.

The ARCHIVE attribute is a feature of Java 1.1. Therefore, if you are using the Java device driver, your browser must support this version of Java. Both Internet Explorer 4.01 and Netscape 4.05 support Java 1.1.

**Interaction**
Use ARCHIVE= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA or DEVICE=ACTIVEX option in the GOPTIONS statement.

**Tips**
Typically, this option should not be used, because the SAS server automatically determines the correct SAS/GRAPH applets to view.
the ODS HTML output. However, if you have renamed the JAR files, or have other applets with which to view the ODS HTML output, this option enables you to access these applets.

Use the CODEBASE= option to specify the file path. It is recommended that you do not put a file path in your ARCHIVE= option.

The value of APPLETOC= points to the location of the Java archive files that are shipped with the SAS system. To find out what the value of this option is, you can either look in the Options window in the Files folder under Environment Control, or you can submit the following procedure step:

```sas
proc options option=appletloc;
run;
```

**ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)**

writes the specified attributes between the tags that generate dynamic graphics output.

*attribute-pair*  
specifies the name and value of each attribute. *attribute-pair* has the following form:

'attribute-name'='attribute-value'

*attribute-name*  
is the name of the attribute.

*attribute-value*  
is the value of the attribute.

**Requirement**  
You must enclose *attribute-name* and *attribute-value* in quotation marks.

**Interaction**  
Use the ATTRIBUTES= option in conjunction with SAS/GRAPH procedures and with the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

**See**  
*SAS/GRAPH: Reference* for valid attributes for the Graph Applet, the Map Applet, the Contour Applet, and the MetaView Applet.

**BASE= 'base-text'**

specifies the text to use as the first part of all links and references that ODS creates in the output files.

*base-text*  
is the text that ODS uses as the first part of all links and references that ODS creates in the file.

Consider this specification:

```
BASE= 'http://www.your-company.com/local-url/
```

In this case, ODS creates links that begin with the string http://www.your-company.com/local-url/. The appropriate *anchor-name* completes the link.

**Requirement**  
You must enclose *base-text* in quotation marks.
BODY= 'file-specification' (suboption(s))
opens a markup family destination and specifies the file that contains the primary output that is created by the ODS statement. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

- external-file
  is the name of an external output file.
  Requirement You must enclose external-file in quotation marks.

- fileref
  is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.
  Restriction The BODY=fileref option cannot be used in conjunction with the NEWFILE= option.
  See For information about the FILENAME statement, see "FILENAME Statement" in SAS Statements: Reference.

- entry.markup
  specifies an entry in a SAS catalog to write to.
  Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

(suboption(s))
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

- (DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.
  See For complete documentation about the DYNAMIC suboption, see "(DYNAMIC)" on page 377.

- (NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.
  See For complete documentation about the NO_BOTTOM_MATTER suboption, see "(NO_BOTTOM_MATTER)" on page 377.

- (NO_TOP_MATTER)
  See For complete documentation about the NO_TOP_MATTER suboption, see "(NO_TOP_MATTER)" on page 378.
(TITLE=’title-text’)
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

See For complete documentation about the TITLE= suboption, see “(TITLE=’title-text’)” on page 378.

(URL= ’Uniform-Resource-Locator’)
See For complete documentation about the URL= suboption, see “(URL= ’Uniform-Resource-Locator’)” on page 378.

Alias FILE=

Interaction Using the BODY= option in an ODS markup family statement that refers to an open ODS markup destination forces ODS to close the destination and all associated files. ODS then opens a new instance of the destination. For more information see “Opening and Closing the MARKUP Destination” on page 517.

Note For some values of TAGSET=, this output is an HTML file. For other TAGSET= values, the output is an XML file, and so on.

CHARSET= character-set
specifies the character set to be generated in the META declaration for the HTML output.

See For information about the CHARSET= option, see “CHARSET=Option” in SAS National Language Support (NLS): Reference Guide.

CODE= ’file-specification’ <(suboption(s))>
opens a markup family destination and specifies the file that contains accompanying programming code, such as JavaScript or XSL (Extensible Stylesheet Language). These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.


entry.markup
specifies an entry in a SAS catalog to write to.
Interaction: If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

**suboption(s)**

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

**(DYNAMIC)**

enables you to send output directly to a web server instead of writing it to a file.

See: For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 377.

**(URL= 'Uniform-Resource-Locator' )**

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See: For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator' )” on page 378.

**CODEBASE=’string’**

specifies the location of the executable Java applet or the ActiveX control file. *string* is specified as a pathname or as a URL. The CODEBASE file path option has two definitions, depending on the GOPTIONS device used.

When you generate web presentations with the JAVA and ActiveX device drivers, SAS generates HTML pages that automatically look for the JAVA archive files or the ActiveX control file in the default installation location.

**For the ActiveX device:**

If you use the ActiveX device driver with ODS to generate output containing an ActiveX control, then specify the CODEBASE= option in the ODS statement. The value of the CODEBASE= option should include the location and the version of the EXE file.

**Tip**

You do not need to specify the CODEBASE= option with the DEVICE=ACTIVEX option unless the users that view your output do not have the ActiveX control installed on their machine. When users that do not have the ActiveX control installed view your output, they are prompted to download the control.

See: SAS/GRAPH: Reference for information about specifying the location of control and applet files using the CODEBASE= and ARCHIVE= options.

**For the Java device:**

If you use the Java device driver with ODS to generate output containing a SAS/GRAPH applet, specify the path to the JAR file with the CODEBASE= option in the ODS statement.

When you specify DEVICE=JAVA, the users that view your output must have access to the appropriate Java applet. By default, SAS sets the value of CODEBASE= to refer to the executable file for the applet that is automatically installed with SAS. The default location of the SAS Java archive files is specified by the APPLETLOC= system option. You do not need to specify the CODEBASE= option if both of the following conditions are true.
The default location is accessible by users who are viewing your web presentation.

The SAS Java archive is installed at that location.

Tip Specify only the directory of the JAR file. The CODEBASE= location can be specified as a pathname or as a URL.

See SAS/GRAPH: Reference for information about specifying the location of control and applet files using the CODEBASE= and ARCHIVE= options.

`CONTENTS= 'file-specification' <(suboption(s))>`

opens a markup family destination and specifies the file that contains a table of contents for the output. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

`file-specification` specifies the file, fileref, or SAS catalog to write to.

`file-specification` is one of the following:

- `external-file` is the name of an external output file.
  
  Requirement You must enclose `external-file` in quotation marks.

- `fileref` is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.
  
  See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

- `entry.markup` specifies an entry in a SAS catalog to write to.
  
  Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

`suboption(s)` specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

- `(DYNAMIC)` enables you to send output directly to a web server instead of writing it to a file.
  
  See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 377.

- `(NO_BOTTOM_MATTER)` specifies that no ending markup language source code be added to the output file.
See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 377.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 378.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

title-text
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 378.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 378.

CSSSTYLE= 'file-specification'<(media-type-1<…media-type-10>)>
specifies a cascading style sheet to apply to your output.

file-specification
specifies a file, fileref, or URL that contains CSS code.

file-specification is one of the following:

"external-file"
is the name of the external file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

"URL"
is a URL to an external file.

Requirement You must enclose URL in quotation marks.

(media-type-1<…, media-type-10>)specifies one or more media blocks that correspond to the type of media that your output is rendered on. CSS uses media type blocks to specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, and so on.
The media block is added to your output in addition to the CSS code that is not contained in any media blocks. By using the `media-type` suboption, in addition to the general CSS code, you can import the section of a CSS file intended only for a specific media type.

**Default**
If no `media-type` is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

**Range**
You can specify up to ten different media types.

**Requirements**
You must enclose `media-type` in parentheses.

You must specify `media-type` next to the `file-specification` specified by the CSSSTYLE= option.

**Tip**
If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

**Restriction**
The CSSSTYLE= option does not affect SAS/GRAPH output.

**Requirement**
CSS files must be written in the same type of CSS produced by the ODS HTML statement. To view the CSS code that ODS creates, you can do one of the following:
- specify the ODS TRACE DOM statement
- specify the DOM option

**Interaction**
If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

**See**
For complete documentation about ODS and Cascading Style Sheets, see *SAS Output Delivery System: Advanced Topics*

**Example**
“Example 6: Applying a CSS File to ODS Output” on page 527

**DOM<="external-file">**
specifies that the ODS document object model is written to the SAS log or an external file.

**external-file**
is the name of an external output file.

**Requirement**
You must enclose `external-file` in quotation marks.

**See**
For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in *SAS Output Delivery System: Advanced Topics*.

**ENCODING= local-character-set-encoding**
overrides the encoding for input or output processing (transcodes) of external files.

**See**
For information about the ENCODING= option, see *SAS National Language Support (NLS): Reference Guide*. 
EVENT=event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )
specifies an event and the value for event variables that are associated with the event.

(FILE= BODY | CODE | CONTENTS | DATA | FRAME | PAGES | STYLESHEET);
triggers one of the known types of output files that correspond to the BODY=, CODE=, CONTENTS=, FRAME=, PAGES=, and STYLESHEET= options.

(FINISH)
triggers the finish section of an event.

See For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(LABEL='variable-value')
specifies the value for the LABEL event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the LABEL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(NAME='variable-value')
specifies the value for the NAME event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the NAME event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(START)
triggers the start section of an event.

See For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(STYLE=style-element)
specifies a style element.

See For information about style elements, see “Style Attributes Overview” in SAS Output Delivery System: Procedures Guide.

(TARGET='variable-value')
specifies the value for the TARGET event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the TARGET event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(TEXT='variable-value')
specifies the value for the TEXT event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the TEXT event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.
(URL='variable-value')  
specifies the value for the URL event variable.

**Requirement**  
`variable-value` must be enclosed in quotation marks.

**See**  
For information about the URL event variable, see “Event Variables” in *SAS Output Delivery System: Procedures Guide*.

---

**Default**  
(FILE='BODY')

**Requirement**  
The EVENT= option's suboptions must be enclosed in parentheses.

**FRAME= 'file-specification' <(suboption(s))>**  
opens a markup family destination and, for HTML output, specifies the file that integrates the table of contents, the page contents, and the body file. If you open the frame file, then you see a table of contents, a table of pages, or both, as well as the body file. For XML output, FRAME= specifies the file that contains the DTD. These files remain open until you do one of the following:

- close the destination with either an ODS `markup-family-destination` CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

**file-specification**  
specifies the file, fileref, or SAS catalog to write to.

**file-specification** is one of the following:

**external-file**  
is the name of an external output file.

**Requirement**  
You must enclose `external-file` in quotation marks.

**fileref**  
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

**See**  
For information about the FILENAME statement, see “FILENAME Statement” in *SAS Statements: Reference*.

**entry.markup**  
specifies an entry in a SAS catalog to write to.

**Interaction**  
If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

**suboption(s)**  
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

**(DYNAMIC)**  
enables you to send output directly to a web server instead of writing it to a file.

**See**  
For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 377.
(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 377.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 378.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

title-text
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 378.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 378.

Restriction If you specify the FRAME= option, then you must also specify the CONTENTS= option, the PAGE= option, or both.

Example “Example 2: Creating an XML File and a DTD” on page 520

GFOOTNOTE | NOGFOOTNOTE
controls the location where footnotes are printed in the graphics output.

GFOOTNOTE
writes footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

NOGFOOTNOTE
writes footnotes that are created by ODS, which appear outside the graph borders.

Default GFOOTNOTE

Restrictions Footnotes that are displayed by a markup language statement support all SAS/GRAPH FOOTNOTE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH FOOTNOTE statement, see “FOOTNOTE Statement” in SAS/GRAPH: Reference.
This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

\[ \text{GPATH=} \ 'aggregate-file-storage-specification' \ | \ \text{fileref} \ | \ \text{libref.catalog} \ (\text{URL=} \ 'Uniform-Resource-Locator' \ | \ \text{NONE}) \]

specifies the location for all graphics output that is generated while the destination is open. Use this option when you want to write graphics output files to a location different that specified by the PATH= option for markup files. If you specify an invalid filename, the ActiveX and Java devices send output to the default filename. Other devices create the file as a directory and write output to that directory using the default filename. For information about how ODS names catalog entries and external files, see SAS/GRAPH: Reference.

\[ 'aggregate-file-storage-location' \]

specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement You must enclose \textit{aggregate-file-storage-location} in quotation marks.

\[ \text{fileref} \]

is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction If you specify a fileref in the GPATH= option, then ODS does not use information from the GPATH= option when it constructs links.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

\[ \text{libref.catalog} \]

specifies a SAS catalog to write to.

\[ \text{URL=} \ 'Uniform-Resource-Locator' \ | \ \text{NONE} \]

specifies a URL for file-specification.

\[ \text{Uniform-Resource-Locator} \]

is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

Requirement You must enclose \textit{Uniform-Resource-Locator} in quotation marks.

\[ \text{NONE} \]

specifies that no information from the GPATH= option appears in the links or references.

Tip This option is useful for building output files that can be moved from one location to another. If the links from the contents and page files are constructed with a simple URL (one name), then they resolve, as long as the contents, page, and body files are all in the same location.

Default If you omit the GPATH= option, then ODS stores graphics in the location that is specified by the PATH= option. If you do not specify the PATH= option, then ODS stores the graphics in the current directory. For more information, see the PATH= option.
GTITLE | NOGTITLE
controls the location where titles are printed in the graphics output.

GTITLE
writes the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.

NOGTITLE
writes the title that is created by ODS, which appears outside of the graph borders.

Default: GTITLE

Restrictions
Titles that are displayed by any markup language statement support most SAS/GRAPH TITLE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH TITLE statement, see TITLE statement.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

HEADTEXT= 'markup-document-head'
specifies markup tags to place between the <HEAD> and </HEAD> tags in all of the output files.

markup-document-head
specifies the markup tags to place between the <HEAD> and </HEAD> tags.

Restriction
HEADTEXT= cannot exceed 256 characters.

Requirement
You must enclose markup-document-head in quotation marks.

Tips
ODS cannot parse the markup that you supply. It should be well-formed markup that is correct in the context of the <HEAD> and </HEAD> tags.

Use the HEADTEXT= option to define programs (such as JavaScript) that you can use later in the file.

(ID= identifier)
enable you to run multiple instances of the same destination at the same time. Each instance can have different options.

identifier
specifies another instance of the destination that is already open. identifier is numeric or a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numeric characters.

Restriction
If identifier is numeric, it must be a positive integer.

Requirement
You must specify the ID= option immediately after the destination name.

Tip
You can omit the ID= option and instead use a name or a number to identify the instance.
Example

“Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594

METATEXT= 'metatext-for-document-head'
specifies HTML code to use as the <META> tag between the <HEAD> and </HEAD> tags of all of the HTML output files.

'metatext-for-document-head'
specifies the HTML code that provides the browser with information about the document that it is loading. For example, this attribute could specify the content type and the character set to use.

Requirement You must enclose metatext-for-document-head in quotation marks.

Default If you do not specify METATEXT=, then ODS writes a simple <META> tag, which includes the content-type of the document and the character set to use, to all the HTML files that it creates.

Restriction METATEXT= cannot exceed 256 characters.

Tip ODS cannot parse the HTML code that you supply. It should be well-formed HTML code that is correct in the context of the <HEAD> tags. If you are using METATEXT= as it is intended, then your META tag should look like this:

<META your-metatext-is-here>

NEWFILE= starting-point creates a new body file at the specified starting-point.

starting-point is the location in the output where you want to create a new body file.

ODS automatically names new files by incrementing the name of the body file. In the following example, ODS names the first body file REPORT.XML. Additional body files are named REPORT1.XML, REPORT2.XML, and so on.

Example:

BODY= 'REPORT.XML'

starting-point is one of the following:

BYGROUP starts a new file for the results of each BY group.

NONE writes all output to the body file that is currently open.

OUTPUT starts a new body file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

Alias TABLE

PAGE starts a new body file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.
PROC
starts a new body file each time you start a new procedure.

Default  NONE

Restriction The NEWFILE= option cannot be used in conjunction with the BODY=fileref option.

Tips If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first body file MAY5.XML. Additional body files are named MAY6.XML, MAY7.XML, and so on.

Example:
BODY= 'MAY5.XML'

OPTIONS ( DOC= | <suboption(s)> )
specifies tagset-specific suboptions and a named value.

(DOC= 'HELP' | 'QUICK' | 'SETTINGS' | 'CHANGELOG')
provides information about the specified tagset.

HELP
provides generic help and information with a quick reference.

QUICK
describes the options available for this tagset.

SETTINGS
provides the current option settings.

CHANGELOG
lists a history of changes made to the tagset. This suboption is supported only on the RTF tagset.

Requirement All values must be enclosed in quotation marks.

suboption(s)
specifies one or more suboptions that are valid for the specified tagset.

Suboptions have the following format:

keyword= 'value'
Specify one of the following options when opening an ODS tagset statement, or at any time after the destination has been opened, to get information about suboptions for the tagset.

• options(doc= 'help');
• options(doc= 'quick');
• options(doc= 'settings');

Requirement  suboption(s) must be enclosed in parentheses.

Example  “Example: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information ” on page 802

PACKAGE <package-name>
specifies that the output from the destination be added to a package.
package-name

specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

See  “ODS PACKAGE Statement” on page 554

Example  “Example 1: Creating an ODS Package” on page 558

PAGE='file-specification' <(suboption(s))>

opens a markup family destination and specifies the file that contains a description of each page of the body file, and contains links to the body file. ODS produces a new page of output whenever a procedure requests a new page. These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification

specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file

is the name of an external output file.

Requirement  You must enclose external-file in quotation marks.

fileref

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See  For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

entry.markup

specifies an entry in a SAS catalog to write to.

Interaction  If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)

enables you to send output directly to a web server instead of writing it to a file.

See  For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 377.

(NO_BOTTOM_MATTER)

specifies that no ending markup language source code be added to the output file.
See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 377.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 378.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

title-text
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 378.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL='Uniform-Resource-Locator')” on page 378.

Interaction The SAS system option PAGESIZE= has no effect on pages in HTML output except when you are creating batch output. For information about the PAGESIZE= option, see “PAGESIZE= System Option” in SAS System Options: Reference.

PARAMETERS= (parameter-pair-1 ... parameter-pair-n)
writes the specified parameters between the tags that generate dynamic graphics output.

parameter-pair
specifies the name and value of each parameter. parameter-pair has the following form:

'parameter-name'= 'parameter-value'

parameter-name
is the name of the parameter.

parameter-value
is the value of the parameter.

Requirement You must enclose parameter-name and parameter-value in quotation marks.

Interaction Use PARAMETERS= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.
See SAS/GRAPH: Reference for valid parameters for the Graph Applet, Map Applet, Contour Applet, and the MetaView Applet.

**PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)**

specifies the location of an aggregate storage location or a SAS catalog for all markup files. If the GPATH= option is not specified, all graphics output files are written to the "aggregate-file-storage-specification" or libref.

'request-file-storage-location'

specifies an aggregate storage location such as directory, folder, or partitioned data set.

**Requirement**

You must enclose *aggregate-file-storage-location* in quotation marks.

**fileref**

is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

**Interaction**

If you use a fileref in the PATH= option, then ODS does not use information from PATH= when it constructs links.

**See**

For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

**libref.catalog**

specifies a SAS catalog to write to.

**See**

For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Statements: Reference.

**URL= 'Uniform-Resource-Locator' | NONE**

specifies a URL for the file-specification.

**Uniform-Resource-Locator**

is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

**NONE**

specifies that no information from the PATH= option appears in the links or references.

**Tip**

This option is useful for building output files that can be moved from one location to another. The links from the contents and page files must be constructed with a single-name URL, and the contents, page, and body files must be in the same location.

**Interaction**

If you use the BODY= or FILE= external file option in conjunction with the PATH= option, the external file specification should not include path information.

**RECORD_SEPARATOR= 'alternative-separator' | NONE**

specifies an alternative character or string that separates lines in the output files.

Different operating environments use different separator characters. If you do not specify a record separator, then the files are formatted for the environment where you run the SAS program. However, if you are generating files for viewing in a different
operating environment that uses a different separator character, then you can specify a record separator that is appropriate for the target environment.

**alternative-separator**
represents one or more characters in hexadecimal or ASCII format. For example, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

```
RECORD_SEPARATOR= '0D0A'x
```

**Operating Environment Information**
In a mainframe environment, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

```
RECORD_SEPARATOR= '0D25'x
```

**Requirement**
You must enclose `alternative-separator` in quotation marks.

**NONE**
produces the markup language that is appropriate for the environment where you run the SAS job.

**Windows Specifics**
In a mainframe environment, by default, ODS produces a binary file that contains embedded record separator characters. This binary file is not restricted by the line-length restrictions on ASCII files. However, if you view the binary files in a text editor, then the lines run together. If you want to format the files so that you can read them with a text editor, then use `RECORD_SEPARATOR= NONE`. In this case, ODS writes one line of markup language at a time to the file. When you use a value of NONE, the logical record length of the file that you are writing to must be at least as long as the longest line that ODS produces. If the logical record length of the file is not long enough, then the markup language might wrap to another line at an inappropriate place.

**Aliases**

```
RECSEP=
RS=
```

**STYLE= style-template**
specifies the style template to use in writing the output files.

**style-template**
describes how to display the presentation aspects (color, font face, font size, and so on) of your SAS output. A style template determines the overall appearance of the documents that use it. Each style template consists of style elements.

**Interaction**
The `STYLE=` option is not valid when you are creating XML output.

**Note**
If you are using SAS Studio, you do not need to specify the `STYLE=` option. You can go to Preferences ⇒ Results and change the style from the drop-down list for your selected destination.

**See**
For a complete discussion of style templates, see “TEMPLATE Procedure: Creating a Style Template” in SAS Output Delivery System: Procedures Guide.
If you do not specify a style template, then ODS uses the file that is specified in the SAS registry subkey `ODS \ DESTINATIONS \ MARKUP`. By default, this value specifies `Default`.

**Interaction**

If you specify the `STYLE=` option in an ODS HTML4 statement and subsequently need PROC PRINT output to use new style templates in another ODS HTML4 statement, close the first statement before specifying the second statement.

**STYLESHEET= 'file-specification' <(suboption(s))>**

opens a markup family destination and places the style information for markup output into an external file, or reads style sheet information from an existing file. These files remain open until you do one of the following:

- close the destination with either an ODS `markup-family-destination CLOSE` statement or ODS _ALL_ `CLOSE` statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

**file-specification**

specifies the file, fileref, or SAS catalog to write to.

**external-file**

is the name of an external output file.

**Requirement**

You must enclose `external-file` in quotation marks.

**fileref**

is a file reference that has been assigned to an external file. Use the `FILENAME` statement to assign a fileref.

**See**

For information about the `FILENAME Statement`, see “FILENAME Statement” in SAS Statements: Reference.

**entry.markup**

specifies an entry in a SAS catalog to write to.

**Interaction**

If you specify an entry name, you must also specify a library and catalog. See the discussion of the `PATH=` option.

**suboption(s)**

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

**(DYNAMIC)**

enables you to send output directly to a web server instead of writing it to a file.

**See**

For complete documentation about the `DYNAMIC` suboption, see “(DYNAMIC)” on page 377.

**(NO_BOTTOM_MATTER)**

specifies that no ending markup language source code be added to the output file.
See, For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 377.

(NO_TOP_MATTER)

specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See, For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 378.

(TITLE="title-text")

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

\textit{title-text}

is the text in the metadata of a file that indicates the title.

See, For complete documentation about the TITLE= suboption, see “(TITLE="title-text")” on page 378.

(URL= 'Uniform-Resource-Locator'

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See, For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 378.

Note

By default, if you do not specifically send the information to a separate file, then the style sheet information is included in the specified HTML file.

Example

“Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

\textbf{TEXT}=text-string

inserts text into your document by triggering the paragraph event and specifying a text string to be assigned to the VALUE event variable.

\textbf{Default}

By default the TEXT= option is used in a paragraph event.

\textbf{Tip}

You can specify a \textit{text-string} for a specific event by using the TEXT= option with the EVENT= option by using the following syntax:

\texttt{EVENT=event-name (TEXT=text-string)}

\textbf{See}

For information about events and event variables, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in \textit{SAS Output Delivery System: Procedures Guide}.

\textbf{Example}

“Example: Conditionally Excluding Output Objects and Sending Them to Different Output Destinations” on page 325

\textbf{TRANTAB}= 'translation-table'

specifies the translation table to use when transcoding a file for output.

Suboptions

(DYNAMIC)

enables you to send output directly to a web server instead of writing it to a file. This option sets the value of the CONTENTTYPE= style attribute. For more information, see CONTENTTYPE= on page 1015 in PROC TEMPLATE.

Default If you do not specify DYNAMIC, then ODS sets the value of HTMLCONTENTTYPE= for writing to a file.

Restriction If you specify the DYNAMIC suboption with one of the following options in the ODS HTML statement, then you must specify it for all of these options in that statement.

• BODY=
• CONTENTS=
• PAGE=
• FRAME=
• STYLESHEET=
• TAGSET=

Requirements You must enclose DYNAMIC in parentheses.

You must specify DYNAMIC next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

(NO BOTTOM_MATTER)

specifies that no ending markup language source code be added to the output file.

Alias NOBOT

Requirements You must enclose NO_BOTTOM_MATTER in parentheses.

You must specify NO_BOTTOM_MATTER next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

Interactions The NO_BOTTOM_MATTER suboption, in conjunction with the NO_TOP_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.
Tip If you want to leave a body file in a state that you can append to with ODS, then use NO_BOTTOM_MATTER with the file-specification BODY= option in any markup language statement.

See The NO_TOP_MATTER suboption

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

Alias NOTOP

Requirements You must enclose NO_TOP_MATTER in parentheses.

You must specify NO_TOP_MATTER next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

Interactions The NO_TOP_MATTER suboption, in conjunction with the NO_BOTTOM_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

See The NO_BOTTOM_MATTER suboption and the ANCHOR= option

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

title-text
is the text in the metadata of a file that indicates the title.

Requirements You must enclose TITLE= in parentheses.

You must enclose title-text in quotation marks.

Tip If you are creating a web page that uses frames, then it is the TITLE= specification for the frame file that appears in the browser window title bar.

Example “Example 3: Creating Multiple Markup Output” on page 522

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

Requirements You must enclose URL= 'Uniform-Resource-Locator' in parentheses.
You must enclose Uniform-Resource-Locator in quotation marks.

You must specify URL='Uniform-Resource-Locator' next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

**Tips**

This option is useful for building HTML files that can be moved from one location to another. The links from the contents and page files must be constructed with a single name URL, and the contents, page, and body files must all be in the same location.

You never need to specify this suboption with the FRAME= option because ODS files do not reference the frame file.

**Example**

“Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

**Details**

The ODS HTML3 statement is part of the ODS markup family of statements. ODS statements in the markup family produce output that is formatted using one of many different markup languages such as HTML (Hypertext Markup Language) and XML (Extensible Markup Language). You can specify a markup language that SAS supplies, or create one of your own and store it as a user-defined markup language.

By default, the SAS registry is configured to generate HTML 4 output when you specify the ODS HTML statement. To permanently change the default HTML version to 3.2, you can change the setting of the HTML version in the SAS registry. The ODS HTML statement produces HTML 3.2 output. For information about how to change your default HTML version, see “Changing the Default HTML Version Setting” on page 44.

**See Also**

- Appendix 3, “ODS and the HTML Destination,” on page 1101
- “Changing SAS Registry Settings for ODS” on page 40

**Statements**

- “ODS MARKUP Statement” on page 488
- “ODS HTML Statement” on page 379

---

**ODS HTML Statement**

Opens, manages, or closes the HTML destination, which produces HTML 4.0 output that can contain embedded style sheets.

- **Valid in:** Anywhere
- **Category:** ODS: Third-Party Formatted
- **Default:** The default style for Markup family destinations is HTMLBlue.
- **Interaction:** By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC PLOT or PROC CHART), ODS formats the output in SAS Monospace font. If you are creating output that will be viewed in an operating
environment where SAS software is not installed, this output is not displayed correctly. This is because without SAS, the SAS Monospace font is not recognized. To make your document display correctly, include the following statement before your SAS program:

```sas
OPTIONS FORMCHAR="|----|+|---+=|-/<>*";
```

**z/OS specifics:** If you use graphics that are created with either the ACTXIMG or JAVAIMG device drivers in the z/OS operating environment, then specify either the GPATH= option or the PATH= option in the ODS HTML statement.


### Syntax

```sas
ODS HTML (<ID=> identifier>) <action> ;
ODS HTML (<ID=> identifier>) <option(s)> ;
```

### Summary of Optional Arguments

- **(ID= identifier)**
  Open multiple instances of the same destination at the same time
- **ANCHOR='anchor-name'**
  Specify a unique base name for the anchor tag that identifies each output object in the current body file
- **ARCHIVE='string'**
  Specify which applet to use to view ODS HTML output
- **ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)**
  Specify attributes to write between the tags that generate dynamic graphics output
- **BASE= 'base-text'**
  Specify text to use as the first part of all links and references that ODS creates in output files
- **BODY= 'file-specification' (suboption(s))**
  Open a markup family destination and specify the file that contains the primary output that is created by the ODS statement
- **BOX_SIZING=(CONTENT_BOX | BORDER_BOX)**
  Specify how to measure the width of cells. Use to override the default value of BOX_SIZING for a destination
- **CHARSET= character-set**
  Specify the character set to be generated in the META declaration for the HTML output
- **CLOSE**
  Close the destination and the file that is associated with it
- **CODE= 'file-specification' <(suboption(s))>**
  Open the HTML destination and specify the file that contains relevant style information
- **CODEBASE='string'**
Create a file path that can be used by the GOPTIONS devices

**CONTENTS= 'file-specification' <(suboption(s))>**
Open the HTML destination and specify the file that contains a table of contents for the output

**CSSSTYLE= 'file-specification'<(media-type-1<…media-type-10>)>**
Specify a cascading style sheet to apply to your output

**DEVICE= device-driver**
Specify a device for the output destination

**DOM<="external-file">**
Specify that the ODS document object model is written to the SAS log or to an external file.

**ENCODING= local-character-set-encoding**
Override the encoding for input or output processing (transcodes) of external files

**EVENT=event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )**
Specify an event and the value for event variables that is associated with the event

**EXCLUDE exclusion(s) | ALL | NONE**
Exclude output objects from the destination

**FRAME= 'file-specification' <(suboption(s))>**
Specify the file that integrates the table of contents, the page contents, and the body file

**GFOOTNOTE | NOGFOOTNOTE**
Control the location where footnotes are printed in the graphics output

**GPATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)**
Specify the location for all graphics output that is generated while the destination is open

**GTITLE | NOGTITLE**
Control the location where titles are printed in the graphics output

**HEADTEXT= 'markup-document-head'**
Specify HTML tags to place between the <HEAD> and </HEAD> tags in all of the output files.

**IMAGE_DPI=**
Specify the image resolution for graphical output

**METATEXT= 'metatext-for-document-head'**
Specify HTML code to use as the <META> tag between the <HEAD> and </HEAD> tags in all of the HTML output files.

**NEWFILE= starting-point**
Create a new body file at the specified starting point

**OPTIONS ( DOC= | <suboption(s)> )**
Specify tagset-specific suboptions and a named value

**PACKAGE <package-name>**
Specify that the output from the destination be added to an ODS package

**PAGE= 'file-specification' <(suboption(s))>**
Open the HTML destination and specify the file that contains a description of each page of the body file and contains links to the body file

**PARAMETERS= (parameter-pair-1 ... parameter-pair-n)**
Write the specified parameters between the tags that generate dynamic graphics output
PATH = 'aggregate-file-storage-specification' | fileref | libref.catalog (URL = 'Uniform-Resource-Locator' | NONE)
    Specify the location of an aggregate storage location or a SAS catalog for all markup files
RECORD_SEPARATOR = 'alternative-separator' | NONE
    Specify an alternative character or string to separate lines in the output files
SELECT selection(s) | ALL | NONE
    Select output objects for the destination
SGE = ON | YES | OFF | NO
    Generate a file that can be edited with the ODS Graphics Editor
SHOW
    Write to the SAS log the current selection or exclusion list for the destination
STYLE = style-template
    Specify a style template to use in writing output files
STYLESHEET = 'file-specification' <(suboption(s))>
    Open the HTML destination and place style information for output into an external file, or read style sheet information from an existing file
TEXT = text-string
    Insert text into your document
TRANTAB = 'translation-table'
    Specify a translation table to use when transcoding a file for output

Without Arguments
If you use the ODS HTML statement without an action or options, then it opens the HTML destination and creates HTML output.

Actions
The following actions are available for the ODS HTML statement.

CLOSE
    closes the destination and any files that are associated with it.
    Tip When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

EXCLUDE exclusion(s) | ALL | NONE
    excludes one or more output objects from the destination.
    Default NONE
    Restriction A destination must be open for this action to take effect.
    See “ODS EXCLUDE Statement” on page 321

SELECT selection(s) | ALL | NONE
    selects output objects for the specified destination.
    Default ALL
    Restriction A destination must be open for this action to take effect.
    See “ODS SELECT Statement ” on page 758

SHOW
    writes the current selection list or exclusion list for the destination to the SAS log.
Restriction  The destination must be open for this action to take effect.

Tip  If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

See  “ODS SHOW Statement” on page 771

**Optional Arguments**

ANCHOR= 'anchor-name'

specifies a unique base name for the anchor that identifies each output object in the current body file.

Each output object has an anchor tag for the contents, page, and frame files to reference. The links and references are automatically created by ODS. The links and references point to the name of an anchor. Therefore, each anchor name in a file must be unique.

anchor-name

is the base name for the anchor tag that identifies each output object in the current body file.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR= 'TABULATE', then ODS names the first anchor tabulate. The second anchor is named tabulate1; the third is named tabulate2, and so on.

Restrictions  Each anchor name in a file must be unique.

Only alphanumeric values, the special characters "$ - _ . + ! * ' () , " and reserved characters used for their reserved purposes can be used unencoded within a URL.

Requirement  You must enclose anchor-name in quotation marks.

Interaction  If you open a file to append to it, be sure to specify a new anchor name to prevent writing the same anchors to the file again. ODS does not recognize anchors that are already in a file when it opens the file.

Tips  You can change anchor names as often as you want by specifying the ANCHOR= option in a markup family statement anywhere in your program. After you have specified an anchor name, it remains in effect until you specify a new one.

Specifying new anchor names at various points in your program is useful when you want other web pages to link to specific parts of your markup language output. Because you can control where the anchor name changes, you know in advance what the anchor name is at those points.

An anchor-name must begin with a letter ([A-Za-z]) and can be followed by any number of letters, digits ([0-9]), hyphens (-), underscores (_), colons (:), and periods (.)

Anchors are prefixed with the anchor name to form the anchor tag in each output object.
ARCHIVE='string'
specifies which applet to use to view the ODS HTML output. The ARCHIVE= option is valid only for the GOPTIONS Java device.

The string must be one that the browser can interpret. For example, if the archive file is local to the computer that you are running SAS on, you can use the FILE protocol to identify the file. If you want to point to an archive file that is on a web server, use the HTTP protocol.

Default
If you do not specify ARCHIVE= and you are using the JAVA device driver, ODS uses the value of the SAS system option APPLETOC=. There is no default if you are using the ACTIVEX device driver.

Requirements
You must enclose string in quotation marks.

The ARCHIVE attribute is a feature of Java 1.1. Therefore, if you are using the Java device driver, your browser must support this version of Java. Both Internet Explorer 4.01 and Netscape 4.05 support Java 1.1.

Interaction
Use ARCHIVE= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA or DEVICE=ACTIVEX option in the GOPTIONS statement.

Tips
Typically, this option should not be used, because the SAS server automatically determines the correct SAS/GRAPH applets to view the ODS HTML output. However, if you have renamed the JAR files, or have other applets with which to view the ODS HTML output, this option enables you to access these applets.

Use the CODEBASE= option to specify the file path. It is recommended that you do not put a file path in your ARCHIVE= option.

The value of APPLETOC= points to the location of the Java archive files that are shipped with the SAS system. To find out what the value of this option is, you can either look in the Options window in the Files folder under Environment Control, or you can submit the following procedure step:

```sas
proc options option=appletloc;
run;
```

ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)
writes the specified attributes between the tags that generate dynamic graphics output.

attribute-pair
specifies the name and value of each attribute. attribute-pair has the following form:

'attribute-name'='attribute-value'

attribute-name
is the name of the attribute.

attribute-value
is the value of the attribute.
Requirement  You must enclose *attribute-name* and *attribute-value* in quotation marks.

Interaction  Use the ATTRIBUTES= option in conjunction with SAS/GRAF procedures and with the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

See  SAS/GRAF: Reference for valid attributes for the Graph Applet, the Map Applet, the Contour Applet, and the MetaView Applet.

**BASE= 'base-text'**

specifies the text to use as the first part of all links and references that ODS creates in the output files.

*base-text*  
is the text that ODS uses as the first part of all links and references that ODS creates in the file.

Consider this specification:

```
BASE= 'http://www.your-company.com/local-url/'
```

In this case, ODS creates links that begin with the string http://www.your-company.com/local-url/. The appropriate *anchor-name* completes the link.

Requirement  You must enclose *base-text* in quotation marks.

**BODY= 'file-specification' (suboption(s))**

opens a markup family destination and specifies the file that contains the primary output that is created by the ODS statement. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

*file-specification*  
specifies the file, fileref, or SAS catalog to write to.

*file-specification* is one of the following:

- **external-file**  
is the name of an external output file.

  Requirement  You must enclose *external-file* in quotation marks.

- **fileref**  
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

  Restriction  The BODY=fileref option cannot be used in conjunction with the NEWFILE= option.

  See  For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

- **entry.markup**  
specifies an entry in a SAS catalog to write to.
If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

(suboption(s))
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see "(DYNAMIC)" on page 406.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see "(NO_BOTTOM_MATTER)" on page 406.

(NO_TOP_MATTER)
See For complete documentation about the NO_TOP_MATTER suboption, see "(NO_TOP_MATTER)" on page 407.

(TITLE=('title-text'))
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

See For complete documentation about the TITLE= suboption, see "(TITLE='title-text')" on page 407.

(URL= 'Uniform-Resource-Locator')
See For complete documentation about the URL= suboption, see "(URL= 'Uniform-Resource-Locator')" on page 407.

FILE=

Using the BODY= option in an ODS markup family statement that refers to an open ODS markup destination forces ODS to close the destination and all associated files. ODS then opens a new instance of the destination. For more information see “Opening and Closing the MARKUP Destination” on page 517.

For some values of TAGSET=, this output is an HTML file. For other TAGSET= values, the output is an XML file, and so on.

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry. For information about using the SAS Registry, see “Changing SAS Registry Settings for ODS” on page 40.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing.
CHARSET= character-set
specifies the character set to be generated in the META declaration for the HTML output.

See For information about the CHARSET= option, see “CHARSET= Option” in SAS National Language Support (NLS): Reference Guide.

CODE= 'file-specification'<(suboption(s))>
opens a markup family destination and specifies the file that contains accompanying programming code, such as JavaScript or XSL (Extensible Stylesheet Language). These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.

• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.


entry.markup
specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 406.

(URL= 'Uniform-Resource-Locator' )
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator' )” on page 407.
CODEBASE='string'
specifies the location of the executable Java applet or the ActiveX control file. *string*
is specified as a pathname or as a URL. The CODEBASE file path option has two
definitions, depending on the GOPTIONS device used.

When you generate web presentations with the JAVA and ActiveX device drivers,
SAS generates HTML pages that automatically look for the JAVA archive files or the
ActiveX control file in the default installation location.

For the ActiveX device:
If you use the ActiveX device driver with ODS to generate output containing an
ActiveX control, then specify the CODEBASE= option in the ODS statement.
The value of the CODEBASE= option should include the location and the
version of the EXE file.

Tip You do not need to specify the CODEBASE= option with the
DEVICE=ACTIVEX option unless the users that view your output do not
have the ActiveX control installed on their machine. When users that do
not have the ActiveX control installed view your output, they are prompted
to download the control.

See *SAS/GRAPH: Reference* for information about specifying the location of
control and applet files using the CODEBASE= and ARCHIVE= options.

For the Java device:
If you use the Java device driver with ODS to generate output containing a
SAS/GRAPH applet, specify the path to the JAR file with the CODEBASE=
option in the ODS statement.

When you specify DEVICE=JAVA, the users that view your output must have
access to the appropriate Java applet. By default, SAS sets the value of
CODEBASE= to refer to the executable file for the applet that is automatically
installed with SAS. The default location of the SAS Java archive files is specified
by the APPLETLOC= system option. You do not need to specify the
CODEBASE= option if both of the following conditions are true.

• The default location is accessible by users who are viewing your web
  presentation.

• The SAS Java archive is installed at that location.

Tip Specify only the directory of the JAR file. The CODEBASE= location can
be specified as a pathname or as a URL.

See *SAS/GRAPH: Reference* for information about specifying the location of
control and applet files using the CODEBASE= and ARCHIVE= options.

CONTENTS='file-specification' <(suboption(s))>
opens a markup family destination and specifies the file that contains a table of
contents for the output. These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE
  statement or ODS _ALL_ CLOSE statement.

• open the same destination with a second markup family statement. This closes
  the first file and opens the second file.

*file-specification* specifies the file, fileref, or SAS catalog to write to.

*file-specification* is one of the following:
external-file

is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

fileref

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

entry.markup

specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)

enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 406.

(NO_BOTTOM_MATTER)

specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 406.

(NO_TOP_MATTER)

specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 407.

(TITLE='title-text')

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

title-text

is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 407.

(URL= 'Uniform-Resource-Locator')

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.
CSSSTYLE= 'file-specification'<(media-type-1<...media-type-10>)>
specifies a cascading style sheet to apply to your output.

file-specification
specifies a file, fileref, or URL that contains CSS code.

"external-file"
is the name of the external file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

"URL"
is a URL to an external file.

Requirement You must enclose URL in quotation marks.

(media-type-1<.. media-type-10>)
specifies one or more media blocks that correspond to the type of media that your output is rendered on. CSS uses media type blocks to specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not contained in any media blocks. By using the media-type suboption, in addition to the general CSS code, you can import the section of a CSS file intended only for a specific media type.

Default If no media-type is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

Range You can specify up to ten different media types.

Requirements You must enclose media-type in parentheses.

You must specify media-type next to the file-specification specified by the CSSSTYLE= option.

Tip If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

Restriction The CSSSTYLE= option does not affect SAS/GRAFH output.
**Requirement**  
CSS files must be written in the same type of CSS produced by the ODS HTML statement. To view the CSS code that ODS creates, you can do one of the following:  
  - specify the ODS TRACE DOM statement  
  - specify the DOM option

**Interaction**  
If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

**See**  
For complete documentation about ODS and Cascading Style Sheets, see SAS Output Delivery System: Advanced Topics

**Example**  
“Example 6: Applying a CSS File to ODS Output” on page 527

**DEVICE= device-driver**  
specifies the name of a device driver. ODS automatically selects an optimal default device for each open output destination.

The following table lists the default devices for the most common ODS output destinations. These default devices are used when graphics are created using SAS/GRAPH or ODS Graphics. In the third maintenance release of SAS 9.4, EPUB3 is the default EPUB destination. EPUB2 was the default EPUB version in prior releases of SAS 9.4. This default is set in the Registry. For a complete list of supported devices and file types, see “Supported File Types for Output Destinations” on page 349.

**Table 6.12  Default Devices for ODS Output Destinations**

<table>
<thead>
<tr>
<th>Output Destination</th>
<th>Default Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPUB2</td>
<td>PNG</td>
</tr>
<tr>
<td>EPUB3 (EPUB)</td>
<td>SVG</td>
</tr>
<tr>
<td>HTML</td>
<td>PNG</td>
</tr>
<tr>
<td>HTML5</td>
<td>SVG</td>
</tr>
<tr>
<td>LISTING</td>
<td>PNG</td>
</tr>
<tr>
<td>Measured RTF</td>
<td>EMF</td>
</tr>
<tr>
<td>RTF</td>
<td>EMF</td>
</tr>
<tr>
<td>Markup Tagsets</td>
<td>PNG</td>
</tr>
</tbody>
</table>

**Tips**  
Specifying a device on the ODS DEVICE= option takes precedence over the SAS global option and the graphics option.

For information about using the SAS Registry to change defaults, see “Changing SAS Registry Settings for ODS” on page 40.

DOM="external-file">
specifies that the ODS document object model is written to the SAS log or an external file.

external-file
  is the name of an external output file.

Requirement  You must enclose external-file in quotation marks.

See  For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in SAS Output Delivery System: Advanced Topics.

ENCODING= local-character-set-encoding
overrides the encoding for input or output processing (transcodes) of external files.

See  For information about the ENCODING= option, see SAS National Language Support (NLS): Reference Guide.

EVENT=event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )
specifies an event and the value for event variables that are associated with the event.

(FILE= BODY | CODE | CONTENTS | DATA | FRAME | PAGES | STYLESHEET);
  triggers one of the known types of output files that correspond to the BODY=, CODE=, CONTENTS=, FRAME=, PAGES=, and STYLESHEET= options.

(FINISH)
  triggers the finish section of an event.

See  For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(LABEL='variable-value')
specifies the value for the LABEL event variable.

Requirement  variable-value must be enclosed in quotation marks.

See  For information about the LABEL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(NAME='variable-value')
specifies the value for the NAME event variable.

Requirement  variable-value must be enclosed in quotation marks.

See  For information about the NAME event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(START)
  triggers the start section of an event.
(STYLE=\texttt{style-element})
specifies a style element.

See For information about style elements, see “Style Attributes Overview” in SAS Output Delivery System: Procedures Guide.

(TARGET='\texttt{variable-value}')
specifies the value for the TARGET event variable.

Requirement \texttt{variable-value} must be enclosed in quotation marks.

See For information about the TARGET event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(TEXT='\texttt{variable-value}')
specifies the value for the TEXT event variable.

Requirement \texttt{variable-value} must be enclosed in quotation marks.

See For information about the TEXT event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(URL='\texttt{variable-value}')
specifies the value for the URL event variable.

Requirement \texttt{variable-value} must be enclosed in quotation marks.

See For information about the URL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

FRAME= \texttt{\texttt{file-specification}'<\texttt{suboption(s)}>}
opens a markup family destination and, for HTML output, specifies the file that integrates the table of contents, the page contents, and the body file. If you open the frame file, then you see a table of contents, a table of pages, or both, as well as the body file. For XML output, FRAME= specifies the file that contains the DTD. These files remain open until you do one of the following:

- close the destination with either an ODS \texttt{markup-family-destination} CLOSE statement or ODS \texttt{_ALL_} CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

\texttt{file-specification}
specifies the file, fileref, or SAS catalog to write to.

\texttt{file-specification} is one of the following:

\texttt{external-file}
is the name of an external output file.
Requirement You must enclose external-file in quotation marks.

**fileref**

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

**entry.markup**

specifies an entry in a SAS catalog to write to.

**Interaction**

If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

**suboption(s)**

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)

enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 406.

(NO_BOTTOM_MATTER)

specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 406.

(NO_TOP_MATTER)

specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 407.

(TITLE='title-text')

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

**title-text**

is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 407.

(URL='Uniform-Resource-Locator')

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL='Uniform-Resource-Locator’)” on page 407.
Restriction  If you specify the FRAME= option, then you must also specify the CONTENTS= option, the PAGE= option, or both.

Example  “Example 2: Creating an XML File and a DTD” on page 520

GFOOTNOTE | NOGFOOTNOTE
controls the location where footnotes are printed in the graphics output.

GFOOTNOTE
 writes footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

NOGFOOTNOTE
 writes footnotes that are created by ODS, which appear outside the graph borders.

Default  GFOOTNOTE

Restrictions  Footnotes that are displayed by a markup language statement support all SAS/GRAPH FOOTNOTE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH FOOTNOTE statement, see “FOOTNOTE Statement” in SAS/GRAPH: Reference.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

GPATH= 'aggregate-file-storage-specification' | fileref | libref.catalg (URL= 'Uniform-Resource-Locator' | NONE)
specifies the location for all graphics output that is generated while the destination is open. Use this option when you want to write graphics output files to a location different that specified by the PATH= option for markup files. If you specify an invalid filename, the ActiveX and Java devices send output to the default filename. Other devices create the file as a directory and write output to that directory using the default filename. For information about how ODS names catalog entries and external files, see SAS/GRAPH: Reference.

'aggregate-file-storage-location'
specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement  You must enclose aggregate-file-storage-location in quotation marks.

fileref
is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction  If you specify a fileref in the GPATH= option, then ODS does not use information from the GPATH= option when it constructs links.

See  For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.
libref.catalog
specifies a SAS catalog to write to.

URL= 'Uniform-Resource-Locator' | NONE
specifies a URL for file-specification.
Uniform-Resource-Locator
is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

Requirement
You must enclose Uniform-Resource-Locator in quotation marks.

NONE
specifies that no information from the GPATH= option appears in the links or references.

Tip
This option is useful for building output files that can be moved from one location to another. If the links from the contents and page files are constructed with a simple URL (one name), then they resolve, as long as the contents, page, and body files are all in the same location.

Default
If you omit the GPATH= option, then ODS stores graphics in the location that is specified by the PATH= option. If you do not specify the PATH= option, then ODS stores the graphics in the current directory. For more information, see the PATH= option.

GTITLE | NOGTITLE
controls the location where titles are printed in the graphics output.
GTITLE
writes the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.
NOGTITLE
writes the title that is created by ODS, which appears outside of the graph borders.

Default
GTITLE

Restrictions
Titles that are displayed by any markup language statement support most SAS/GRAPH TITLE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH TITLE statement, see TITLE statement.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

HEADTEXT= 'markup-document-head'
specifies markup tags to place between the < HEAD> and < /HEAD> tags in all of the output files.
markup-document-head
specifies the markup tags to place between the < HEAD> and < /HEAD> tags.

Restriction
HEADTEXT= cannot exceed 256 characters.
Requirement You must enclose `markup-document-head` in quotation marks.

Tips ODS cannot parse the markup that you supply. It should be well-formed markup that is correct in the context of the `<HEAD>` and `</HEAD>` tags.

Use the HEADTEXT= option to define programs (such as JavaScript) that you can use later in the file.

**ID= `identifier`**

enables you to run multiple instances of the same destination at the same time. Each instance can have different options.

`identifier` specifies another instance of the destination that is already open. `identifier` is numeric or a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numeric characters.

**Restriction** If `identifier` is numeric, it must be a positive integer.

**Requirement** You must specify the ID= option immediately after the destination name.

**Tip** You can omit the ID= option and instead use a name or a number to identify the instance.

**Example** “Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594

**IMAGE_DPI=**

specifies the image resolution for graphical output.

**Alias** DPI=

**Default** 96

**Restriction** This option only applies when the ODS GRAPHICS statement IMAGEMAP = OFF | NOIMAGEMAP.

**CAUTION** Caution: When using high DPI= or DPI_IMAGE values (values over 1000), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500M or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.

**METATEXT= `metatext-for-document-head`**

specifies HTML code to use as the `<META>` tag between the `<HEAD>` and `</HEAD>` tags of all of the HTML output files.

`metatext-for-document-head` specifies the HTML code that provides the browser with information about the document that it is loading. For example, this attribute could specify the content type and the character set to use.

**Requirement** You must enclose `metatext-for-document-head` in quotation marks.
If you do not specify METATEXT=, then ODS writes a simple <META> tag, which includes the content-type of the document and the character set to use, to all the HTML files that it creates.

METATEXT= cannot exceed 256 characters.

ODS cannot parse the HTML code that you supply. It should be well-formed HTML code that is correct in the context of the <HEAD> tags. If you are using METATEXT= as it is intended, then your META tag should look like this:

```
<META your-metatext-is-here>
```

NEWFILE= starting-point

creates a new body file at the specified starting-point.

starting-point is the location in the output where you want to create a new body file.

ODS automatically names new files by incrementing the name of the body file. In the following example, ODS names the first body file REPORT.XML. Additional body files are named REPORT1.XML, REPORT2.XML, and so on.

Example:

```
BODY= 'REPORT.XML'
```

starting-point is one of the following:

**BYGROUP**

starts a new file for the results of each BY group.

**NONE**

writes all output to the body file that is currently open.

**OUTPUT**

starts a new body file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

**PAGE**

starts a new body file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.

**PROC**

starts a new body file each time you start a new procedure.

The NEWFILE= option cannot be used in conjunction with the BODY=fileref option.

If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first body file MAY5.XML. Additional body files are named MAY6.XML, MAY7.XML, and so on.

Example:
**OPTIONS** (DOC= | <suboption(s)> )
specifies tagset-specific suboptions and a named value.

(DOC= 'HELP' | 'QUICK' | 'SETTINGS' | 'CHANGELOG')
provides information about the specified tagset.

**HELP**
provides generic help and information with a quick reference.

**QUICK**
describes the options available for this tagset.

**SETTINGS**
provides the current option settings.

**CHANGELOG**
lists a history of changes made to the tagset. This suboption is supported only on the RTF tagset.

*Requirement*
All values must be enclosed in quotation marks.

**suboption(s)**
specifies one or more suboptions that are valid for the specified tagset.
Suboptions have the following format:

keyword='value'
Specify one of the following options when opening an ODS tagset statement, or at any time after the destination has been opened, to get information about suboptions for the tagset.

• options(doc='help');
• options(doc='quick');
• options(doc='settings');

*Requirement*
suboption(s) must be enclosed in parentheses.

*Example*
“Example: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information ” on page 802

**PACKAGE <package-name>**
specifies that the output from the destination be added to a package.

**package-name**
specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

*See*
“ODS PACKAGE Statement ” on page 554

*Example*
“Example 1: Creating an ODS Package” on page 558

**PAGE= 'file-specification' <(suboption(s))>**
opens a markup family destination and specifies the file that contains a description of each page of the body file, and contains links to the body file. ODS produces a new page of output whenever a procedure requests a new page. These files remain open until you do one of the following:
• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.

• open the same destination with a second markup family statement. This closes the first file and opens the second file.

**file-specification**

specifies the file, fileref, or SAS catalog to write to.

**file-specification** is one of the following:

- **external-file**
  is the name of an external output file.

  **Requirement** You must enclose *external-file* in quotation marks.

- **fileref**
  is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

  **See** For information about the FILENAME statement, see “FILENAME Statement” in *SAS Statements: Reference*.

- **entry.markup**
  specifies an entry in a SAS catalog to write to.

  **Interaction** If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

**suboption(s)**

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

- **(DYNAMIC)**
  enables you to send output directly to a web server instead of writing it to a file.

  **See** For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 406.

- **(NO_BOTTOM_MATTER)**
  specifies that no ending markup language source code be added to the output file.

  **See** For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 406.

- **(NO_TOP_MATTER)**
  specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

  **See** For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 407.

- **(TITLE='title-text')**
  inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.
title-text

is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 407.

(URL= 'Uniform-Resource-Locator')

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL='Uniform-Resource-Locator')” on page 407.

Interaction The SAS system option PAGESIZE= has no effect on pages in HTML output except when you are creating batch output. For information about the PAGESIZE= option, see “PAGESIZE= System Option” in SAS System Options: Reference.

PARAMETERS= (parameter-pair=1 ... parameter-pair=n)

writes the specified parameters between the tags that generate dynamic graphics output.

parameter-pair

specifies the name and value of each parameter. parameter-pair has the following form:

'order-name'='order-value'

order-name

is the name of the parameter.

order-value

is the value of the parameter.

Requirement You must enclose order-name and order-value in quotation marks.

Interaction Use PARAMETERS= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

See SAS/GRAPH: Reference for valid parameters for the Graph Applet, Map Applet, Contour Applet, and the MetaView Applet.

PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)

specifies the location of an aggregate storage location or a SAS catalog for all markup files. If the GPATH= option is not specified, all graphics output files are written to the "aggregate-file-storage-specification" or libref.

'aggregate-file-storage-location'

specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement You must enclose aggregate-file-storage-location in quotation marks.
fileref

is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction

If you use a fileref in the PATH= option, then ODS does not use information from PATH= when it constructs links.

See

For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

libref.catalog

specifies a SAS catalog to write to.

See

For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Statements: Reference.

URL= 'Uniform-Resource-Locator' | NONE

specifies a URL for the file-specification.

Uniform-Resource-Locator

is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

NONE

specifies that no information from the PATH= option appears in the links or references.

Tip

This option is useful for building output files that can be moved from one location to another. The links from the contents and page files must be constructed with a single-name URL, and the contents, page, and body files must be in the same location.

Interaction

If you use the BODY= or FILE= external file option in conjunction with the PATH= option, the external file specification should not include path information.

RECORD_SEPARATOR= 'alternative-separator' | NONE

specifies an alternative character or string that separates lines in the output files.

Different operating environments use different separator characters. If you do not specify a record separator, then the files are formatted for the environment where you run the SAS program. However, if you are generating files for viewing in a different operating environment that uses a different separator character, then you can specify a record separator that is appropriate for the target environment.

alternative-separator

represents one or more characters in hexadecimal or ASCII format. For example, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= '0D0A'x

Operating Environment Information

In a mainframe environment, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= '0D25'x
Requirement  You must enclose alternative-separator in quotation marks.

NONE
produces the markup language that is appropriate for the environment where you run the SAS job.

Windows Specifics
In a mainframe environment, by default, ODS produces a binary file that contains embedded record separator characters. This binary file is not restricted by the line-length restrictions on ASCII files. However, if you view the binary files in a text editor, then the lines run together. If you want to format the files so that you can read them with a text editor, then use RECORD_SEPARATOR= NONE. In this case, ODS writes one line of markup language at a time to the file. When you use a value of NONE, the logical record length of the file that you are writing to must be at least as long as the longest line that ODS produces. If the logical record length of the file is not long enough, then the markup language might wrap to another line at an inappropriate place.

Aliases  RECSEP=
        RS=

SGE=ON | YES | OFF | NO
generates a file that can be edited only with the ODS Graphics Editor. The file created has an extension of .sge.

See  For details about using the ODS Graphics Editor to create SGE files, see SAS ODS Graphics Editor: User's Guide.

Example  ods html sge=on;
          proc sgplot data=sashelp.class;
          scatter x=weight y=height;
          run;

STYLE= style-template
specifies the style template to use in writing the output files.

style-template
describes how to display the presentation aspects (color, font face, font size, and so on) of your SAS output. A style template determines the overall appearance of the documents that use it. Each style template consists of style elements.

Interaction  The STYLE= option is not valid when you are creating XML output.

Note  If you are using SAS Studio, you do not need to specify the STYLE= option. You can go to Preferences ⇒ Results and change the style from the drop-down list for your selected destination.

See  For a complete discussion of style templates, see “TEMPLATE Procedure: Creating a Style Template” in SAS Output Delivery System: Procedures Guide.

Default  If you do not specify a style template, then ODS uses the file that is specified in the SAS registry subkey ODS ⇒ DESTINATIONS ⇒ MARKUP. By default, this value specifies Default.
Interaction  If you specify the STYLE= option in an ODS HTML4 statement and subsequently need PROC PRINT output to use new style templates in another ODS HTML4 statement, close the first statement before specifying the second statement.

STYLESHEET= 'file-specification' <(suboption(s))> 
opens a markup family destination and places the style information for markup output into an external file, or reads style sheet information from an existing file. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification 
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file 
is the name of an external output file.

Requirement  You must enclose external-file in quotation marks.

fileref 
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See  For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

entry.markup 
specifies an entry in a SAS catalog to write to.

Interaction  If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s) 
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC) 
enables you to send output directly to a web server instead of writing it to a file.

See  For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 406.

(NO_BOTTOM_MATTER) 
specifies that no ending markup language source code be added to the output file.

See  For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 406.
(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top
of the output file. For HTML 4.0, the NO_TOP_MATTER option removes
the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption,
see “(NO_TOP_MATTER)” on page 407.

(TITLE="title-text")
inserts into the metadata of a file the text string that you specify as the text to
appear in the browser window title bar.

     title-text
     is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see
“(TITLE="title-text")” on page 407.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the
filename) in all the links and references that it creates and that point to the
file.

See For complete documentation about the URL= suboption, see “(URL=
'Uniform-Resource-Locator')” on page 407.

Note By default, if you do not specifically send the information to a separate
file, then the style sheet information is included in the specified HTML file.

Example “Example 5: Including Multiple Cascading Style Sheets in One HTML
Document” on page 525

TEXT=text-string
inserts text into your document by triggering the paragraph event and specifying a
text string to be assigned to the VALUE event variable.

Default By default the TEXT= option is used in a paragraph event.

Tip You can specify a text-string for a specific event by using the TEXT=
option with the EVENT= option by using the following syntax:

EVENT=event-name (TEXT=text-string)

See For information about events and event variables, see “TEMPLATE
Procedure: Creating Markup Language Tagsets” in SAS Output Delivery
System: Procedures Guide.

Example “Example: Conditionally Excluding Output Objects and Sending Them to
Different Output Destinations” on page 325

TRANTAB= 'translation-table'
specifies the translation table to use when transcoding a file for output.

See For information about the TRANTAB= option, see “TRANTAB= System
**Suboptions**

**(DYNAMIC)**

enables you to send output directly to a web server instead of writing it to a file. This option sets the value of the CONTENTTYPE= style attribute. For more information, see CONTENTTYPE= on page 1015 in PROC TEMPLATE.

**Default**

If you do not specify DYNAMIC, then ODS sets the value of HTMLCONTENTTYPE= for writing to a file.

**Restriction**

If you specify the DYNAMIC suboption with one of the following options in the ODS HTML statement, then you must specify it for all of these options in that statement.

- BODY=
- CONTENTS=
- PAGE=
- FRAME=
- STYLESHEET=
- TAGSET=

**Requirements**

You must enclose DYNAMIC in parentheses.

You must specify DYNAMIC next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

**(NO_BOTTOM_MATTER)**

specifies that no ending markup language source code be added to the output file.

**Alias**

NOBOT

**Requirements**

You must enclose NO_BOTTOM_MATTER in parentheses.

You must specify NO_BOTTOM_MATTER next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

**Interactions**

The NO_BOTTOM_MATTER suboption, in conjunction with the NO_TOP_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

**Tip**

If you want to leave a body file in a state that you can append to with ODS, then use NO_BOTTOM_MATTER with the file-specification BODY= option in any markup language statement.

**See**

The NO_TOP_MATTER suboption
(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

**Alias**
NOTOP

**Requirements**
You must enclose NO_TOP_MATTER in parentheses.

You must specify NO_TOP_MATTER next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

**Interactions**
The NO_TOP_MATTER suboption, in conjunction with the NO_BOTTOM_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

**See**
The NO_BOTTOM_MATTER suboption and the ANCHOR= option

(TITLE="title-text")
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

**title-text**
is the text in the metadata of a file that indicates the title.

**Requirements**
You must enclose TITLE= in parentheses.

You must enclose title-text in quotation marks.

**Tip**
If you are creating a web page that uses frames, then it is the TITLE= specification for the frame file that appears in the browser window title bar.

**Example**
“Example 3: Creating Multiple Markup Output” on page 522

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

**Requirements**
You must enclose URL= 'Uniform-Resource-Locator' in parentheses.

You must enclose Uniform-Resource-Locator in quotation marks.

You must specify URL= 'Uniform-Resource-Locator' next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.
**Tips**

This option is useful for building HTML files that can be moved from one location to another. The links from the contents and page files must be constructed with a single name URL, and the contents, page, and body files must all be in the same location.

You never need to specify this suboption with the FRAME= option because ODS files do not reference the frame file.

**Example**

“Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

**Details**

The ODS HTML statement is part of the ODS markup family of statements. ODS statements in the markup family produce output that is formatted using one of many different markup languages, such as HTML (Hypertext Markup Language) and XML (Extensible Markup Language). You can specify a markup language that SAS supplies, or create one of your own and store it as a user-defined markup language.

Beginning with SAS 9.3, by default, in the Windowing environment with the Windows and UNIX operating systems, the LISTING destination is closed and the HTML destination is open. You do not have to submit an ODS HTML statement to generate HTML output, and you do not have to use the ODS HTML CLOSE statement to be able to view your output. However, to create LISTING output, you must either submit the ODS LISTING statement or enable the LISTING destination by other means. For more details, see “Working with Output Defaults” on page 21.

Starting in SAS 9.3, the default style for HTML is HTMLBlue.

The HTML destination supports Scalable Vector Graphics (SVG). For more information about creating SVG files in SAS/GRAPH, see “Using SVG Graphics” in SAS/GRAPH: Reference. For information about SVG and Universal Printing, see “Creating SVG (Scalable Vector Graphics) Files Using Universal Printing” in SAS Language Reference: Concepts. In order to view SVG graphics that were created using the ODS HTML destination, you need a viewer or browser that supports Scalable Vector Graphics. For more information, see “Browser Support for Viewing SVG Documents” in SAS Language Reference: Concepts.

**Examples**

**Example 1: Using the DOC Suboption to Get ODS HTML Information**

**Features:**
- ODS HTML statement action:
  - CLOSE
- ODS HTML statement options:
  - OPTIONS (DOC="HELP")

**Other features:**
- PROC PRINT

**Details**

The following example prints a list of OPTIONS suboptions and a description of each suboption that is available for the HTML tagset. The information is printed to the SAS log.
Program

ods html options (doc="help");
proc print data=Sashelp.Class;
run;

Program Description

Print information about the OPTIONS suboptions to the SAS log file.

ods html options (doc="help");

Print the data set Sashelp.Class. The PROC PRINT statement prints the Sashelp.Class data set.

proc print data=Sashelp.Class;
run;

Output

Specify the DOC='HELP' suboption to print all of the OPTIONS suboptions and information about each of the suboptions to the SAS log.


Example 2: Using the Option Suboption PAGEBREAK=

Features:
ODS HTML statement options:
   OPTIONS (PAGEBREAK="NO")

Other features:
   PROC PRINT

Details
The following example shows how to use the PAGEBREAK= suboption to control whether a page break is allowed or not. The default is to provide a page break after each print statement. In HTML, a page break is rendered by separating output with a horizontal rule. With PAGEBREAK="NO", the horizontal rule is not produced.
Program

```sas
ods html file="test.html" options(pagebreak='no');
options obs=2;
proc print data=Sashelp.Class;
run;
proc print data=Sashelp.Class;
run;
```

Program Description

**Specify the PAGEBREAK="NO" suboption.** The two data sets are rendered without a separating horizontal rule. The output is printed to the test.html file.

```sas
ods html file="test.html" options(pagebreak='no');
```

**Print only two observations of the data set.**

```sas
options obs=2;
```

**Print the data set Sashelp.Class.** The PROC PRINT statement prints the Sashelp.Class data set.

```sas
proc print data=Sashelp.Class;
run;
```

**Print the data set Sashelp.Class.** Print the Sashelp.Class data set again. Because PAGEBREAK="NO" is specified, there is not a page break between the two data sets. By default, these two data sets would be written to two different pages.

```sas
proc print data=Sashelp.Class;
run;
```

Output

Specify the PAGEBREAK="NO" suboption if you want fewer pages of output.

**Output 6.34 PAGEBREAK= Suboption Set to NO in HTML**
Example 3: Creating a Separate Body File for Each Page of Output

Features:

ODS HTML statement action:
    CLOSE

ODS HTML statement options:
    BASE=
    CONTENTS=
    BODY=
    FRAME=
    NEWFILE=
    PAGE=

Other features:

#BYVAL parameter in titles
NOBYLINE|BYLINE system option
OPTIONS statement
PROC FORMAT
PROC SORT
PROC REPORT
PROC TABULATE
TITLE statement

Data set:
    Grain_Production

Format:
    $CNTRY.

Details

The following example creates a separate HTML file for each page of procedure output, as well as a table of contents, a table of pages, and a frame file. The table of contents and table of pages appear and behave the same as those that would be created if all the output was in a single file. Because the output is in separate files, you cannot scroll from one page of output to the next. However, you can select individual HTML files to include in a report.

Program

    proc sort data=grain_production;
        by year country type;
    run;

    ods html body='grain-body.htm'
        contents='grain-contents.htm'
        frame='grain-frame.htm'
        pages='grain-page.htm'
        newfile=page;
    options nobyline;
    title 'Leading Grain-Producing Countries';
    title2 'for #byval(year)';

    proc report data=grain_production nowindows;
        by year;
        column country type kilotons;
Program Description

Sort the data set Grain_Production. PROC SORT sorts the data, first by values of the variable Year, then by values of the variable Country, and finally by values of the variable Type.

```sas
proc sort data=grain_production;
   by year country type;
run;
```

Create HTML output. The ODS HTML statement opens the HTML destination and creates HTML output. The FRAME=, CONTENTS=, and PAGE= options create a frame that includes a table of contents and a table of pages that link to the contents of the body file. The body file also appears in the frame. BASE= specifies a string to use as the first part of all links and references to the HTML files. Because no URL is specified for individual files, the final part of the link matches the filename. The string that the BASE= option specifies must be a valid path to your HTML files.

```sas
ods html body='grain-body.htm'
   contents='grain-contents.htm'
   frame='grain-frame.htm'
   page='grain-page.htm'
```

Specify that SAS create a new body file for each page of output. The NEWFILE=PAGE option opens and creates a new body file for each page of output.

```sas
newfile=page;
```

Suppress the default BY line and specify a new value into the BY line. The NOBYLINE option suppresses the default BY line variable. The #BYVAL parameter specification inserts the current value of the BY variable Year into the title.

```sas
options nobyline;
   title 'Leading Grain-Producing Countries';
   title2 'for #byval(year)';
```
Produce a report. This PROC REPORT step produces a report on grain production. Each BY group produces a page of output, so ODS creates a new body file for each BY group. The NOWINDOWS option specifies that PROC REPORT runs without the REPORT window and sends its output to any open output destinations.

```
proc report data=grain_production nowindows;
   by year;
   column country type kilotons;
   define country  / group width=14 format=$cntry.;
   define type     / group 'Type of Grain';
   define kilotons / format=comma12.;
   footnote 'Measurements are in metric tons.';
run;
```

Restore the default BY line and clear the second TITLE statement. The BYLINE option restores the default BY line. The TITLE2 statement clears the second TITLE statement.

```
options byline;
   title2;
```

Produce a report. The TABLE statement in this PROC TABULATE step specifies the variable Year. Therefore, PROC TABULATE explicitly produces one page of output for 1995 and one for 1996. ODS starts a new body file for each page.

```
proc tabulate data=grain_production format=comma12.;
   class year country type;
   var kilotons;
   table year,
         country*type,
         kilotons*sum='' / box=_page_ misstext='No data';
   format country $cntry.;
   footnote 'Measurements are in metric tons.';
run;
```

Close the HTML destination. The ODS HTML CLOSE statement closes the HTML destination and all the files that are associated with it. If you do not close the destination, then you cannot view the HTML file specified by the FRAME attribute until you close your SAS session.

```
ods html close;
```

HTML Output

This frame file shows the first body file. Links in the table of contents and the table of pages point to the other body files. The frame file is not rendered in the results viewer after running this example. To open it, locate the file in your SAS output location.
Output 6.35  HTML Frame File

Table of Contents

1. Report
   • Year=1995
     • Detailed and/or summarized report
       • Table 1
   • Year=1996
     • Detailed and/or summarized report
       • Table 1

2. Tabulate
   • Cross-tabular summary report
     • Table 1
     • Year 1995
     • Year 1996

Table of Pages

1. Report
   • Page 1
   • Page 2

2. Tabulate
   • Page 3
   • Page 4

Leading Grain-Producing Countries for 1995

<table>
<thead>
<tr>
<th>Country</th>
<th>Type of Grain</th>
<th>Kilotons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil</td>
<td>Corn</td>
<td>30,270</td>
</tr>
<tr>
<td></td>
<td>Rice</td>
<td>11,230</td>
</tr>
<tr>
<td></td>
<td>Wheat</td>
<td>1,510</td>
</tr>
<tr>
<td>China</td>
<td>Corn</td>
<td>112,331</td>
</tr>
<tr>
<td></td>
<td>Rice</td>
<td>185,226</td>
</tr>
<tr>
<td></td>
<td>Wheat</td>
<td>102,207</td>
</tr>
<tr>
<td>India</td>
<td>Corn</td>
<td>9,800</td>
</tr>
<tr>
<td></td>
<td>Rice</td>
<td>122,372</td>
</tr>
<tr>
<td></td>
<td>Wheat</td>
<td>63,007</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Corn</td>
<td>8,223</td>
</tr>
<tr>
<td></td>
<td>Rice</td>
<td>40,860</td>
</tr>
<tr>
<td></td>
<td>Wheat</td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>Corn</td>
<td>187,300</td>
</tr>
<tr>
<td></td>
<td>Rice</td>
<td>7,888</td>
</tr>
<tr>
<td></td>
<td>Wheat</td>
<td>59,494</td>
</tr>
</tbody>
</table>

Measurements are in metric tons.

Links That Are Created in the HTML Output

These HREF= attributes from the links in the contents file point to the HTML tables that ODS creates from the PROC REPORT and PROC TABULATE steps.

href="grain-body.htm#IDX"
href="grain-body1.htm#IDX1"
href="grain-body2.htm#IDX2"
href="grain-body3.htm#IDX3"

Notice how these HREF attributes are constructed:

• The value of the BODY= option, grain-body, provides the basis for the next part of the HREF. However, because the NEWFILE= option creates a new file for each output object, ODS increments this base value each time it creates a file. The resulting filenames become part of the HREF. They are Grain-Body.htm, Grain-Body1.htm, Grain-Body2.htm, and Grain-Body3.htm.

• The value of the ANCHOR= option provides the basis for the last part of the HREF, which follows the number sign (#). Because the ANCHOR= option is not used in this example, ODS uses the default value of IDX. With each use, ODS increments the value of the anchor.
Example 4: Appending to HTML Files

Features:
- ODS HTML statement options:
  - ANCHOR=
  - BODY= with a fileref
  - BODY= using the NO_BOTTOM_MATTER suboption
  - BODY= using the NO_TOP_MATTER suboption
  - STYLE=
- Other features:
  - FILENAME statement
  - PROC PRINT
  - PROC REPORT
  - DATA _NULL_ statement

Data set:
- Grain_Production

Format:
- $CNTRY.

Details
The following example creates HTML output from PROC PRINT and PROC REPORT. It also uses the DATA step to write customized HTML code to the file that contains the HTML output. The DATA step executes between procedure steps.

Program
```plaintext
options obs=10;
filename reports 'GrainReport.html';
ods html body=reports (no_bottom_matter)
   style=Blockprint;
proc print data=grain_production;
   var country type kilotons;
   format country $cntry. kilotons comma12.;
   where year=1996;
   title 'Leading Grain-Producing Countries';
   footnote 'Measurements are in metric tons.';
run;
ods html close;
filename reports '../ods/grain-reports-body.htm' mod;
filename reports 'GrainReport.html' mod;
data _null_;    
   file reports;
   put "<h2>The preceding output is from PROC PRINT.";
   put "I am going to try a variety of procedures.";
   put "Let me know which procedure you prefer.";
   put "This report uses the Blockprint style.</h2>";
run;
ods html body=reports (no_top_matter no_bottom_matter)
```
anchor='report';

proc report data=grain_production nowindows;
   where year=1996;
   column country type kilotons;
   define country / group width=14 format=$cntry.;
   define type / group 'Type of Grain';
   define kilotons / format=comma12.;
run;
ods html close;
data _null_;  
   file reports;
   put "<h2>The preceding output is from PROC REPORT.";
   put "It does not repeat the name of the country on every line.";
   put "This report uses the default style.</h2>";
run;
ods html body=reports(no_top_matter)anchor='end';

Program Description

Set system options. This OBS option limits processing of observations in the data set to 10.

options obs=10;

Assign a fileref to the file GrainReport.html. The FILENAME statement assigns the fileref REPORTS to the file GrainReport.html that contains the HTML output.

filename reports 'GrainReport.html';

Create HTML output and suppress the writing of the default HTML code that would be written at the end of the file. The ODS HTML statement opens the HTML destination and creates HTML output. The NO_BOTTOM_MATTER option suppresses the writing of the default HTML code that, by default, ODS writes at the end of a file.

ods html body=reports (no_bottom_matter)
style=Blockprint;

Specify the style for formatting the HTML output. The STYLE= option specifies that the style Blockprint be used.

Create a report that contains only the data from 1996. Select and format the variables that you want to include, specify a title, and specify a footnote. This PROC PRINT step prints the observations in the data set Grain_Report that have a value of 1996 for the variable Year. The VAR statement selects Country, Type, and Kilotons as the variables that you want to be displayed in the output. The TITLE and FOOTNOTE statements specify the title and footnote.

proc print data=grain_production;
   var country type kilotons;
   format country $cntry. kilotons comma12.;
   where year=1996;
   title 'Leading Grain-Producing Countries';
   footnote 'Measurements are in metric tons.';
run;
Close the HTML destination. The ODS HTML CLOSE statement closes the HTML
destination and all the files that are associated with it.

```sql
ods html close;
filename reports '../ods/grain-reports-body.htm' mod;
```

Assign the fileref REPORTS to the file 'GrainReport.html'. This FILENAME
statement assigns a fileref to the file to be updated, GrainReport.html. The MOD option
opens the file in Update mode. The MOD option might not be valid in all operating
environments. See your operating environment documentation for more information.

```sql
filename reports 'GrainReport.html' mod;
```

Append text to the HTML file REPORTS. This DATA step writes to the file that is
referenced by REPORTS. The PUT statements create an H2 header in the HTML file.

```sql
data _null_;  
file reports;  
put "<h2>The preceding output is from PROC PRINT.";  
put "I am going to try a variety of procedures.";  
put "Let me know which procedure you prefer.";  
put "This report uses the Blockprint style.</h2>";  
run;
```

Create HTML output. This ODS HTML statement opens the HTML destination and
creates HTML output. The NO_TOP_MATTER and NO_BOTTOM_MATTER
suboptions suppress the default HTML code that ODS writes to the top and the bottom
of a file.

```sql
ods html body=reports (no_top_matter no_bottom_matter)
```

Specify the root name for the HTML anchor tags. The ANCHOR= option specifies
report as the root name for the HTML anchor tags. When you use ODS to append to an
HTML file that ODS created, you must specify a new anchor name each time you open
the file from ODS so that you do not write the same anchors to the file again. (ODS
cannot recognize anchors that are already in the file when it opens it, and by default it
uses IDX as the base for anchor names).

```sql
anchor='report';
```

Create a report that contains only the 1996 data. The PROC REPORT step prints the
data set. ODS adds HTML output to the body file. The NOWINDOWS option specifies
that PROC REPORT runs without the REPORT window and sends its output to the open
output destination(s).

```sql
proc report data=grain_production nowindows;  
where year=1996;  
column country type kilotons;  
define country / group width=14 format=$cntry.;  
define type / group 'Type of Grain';  
define kilotons / format=comma12.;  
run;
```

Close the HTML destination. The ODS HTML CLOSE statement closes the HTML
destination and all the files that are associated with it.

```sql
ods html close;
```
Append text to the HTML file REPORTS. This DATA step writes to the file that is referenced by REPORTS. The PUT statements create an H2 header in the HTML file.

```plaintext
data _null_;  
  file reports;  
  put "<h2>The preceding output is from PROC REPORT.";  
  put "It does not repeat the name of the country on every line.";  
  put "This report uses the default style.</h2>";  
run;
```

Create HTML output to write the bottom matter to the file, repress the printing of the top matter, and provide a new root name for the anchor tags. In order to write the bottom matter to the HTML file so that it contains valid HTML code, you must open the HTML destination one more time. NO_TOP_MATTER ensures that the top matter is not placed in the file again. ANCHOR= provides a new root name for the anchors in the bottom matter.

```plaintext
ods html body=reports(no_top_matter)anchor='end';
```

**HTML Output**

This output is created by appending HTML output to an existing HTML file.
Output 6.36  HTML Output with Appended HTML

Example 5: Removing the Horizontal Rule between Procedures

Features:
- **OPTIONS option:**
  - PAGEBREAK=NO

Other features:
- GPLOT procedure
- PRINT procedure
- GOPTIONS statement

Details

HTML documents are usually formed as one continuous page of information without any page breaks. When the document is printing, objects falling on the margins of the printed page might be split across two pages. To prevent this, the ODS HTML destination inserts page breaks between each output object by inserting a paragraph tag that automatically includes a page break command. When printed, each output object
appears on a separate page. Sometimes, with smaller documents, it might be desirable to remove these hardcoded page breaks. This example shows you how to remove the page breaks.

**Program**

```plaintext
options nodate obs=10;
goptions xpixels=500 ypixels=400;
ods html options(pagebreak='no');
title "Student Correlation";
symbol1 font="albnay amt" value='O' height=15pt color=pink;
symbol2 font="albnay amt" value='X' height=15pt color=lib;
proc gplot data=sashelp.class;
plot height*weight=sex / des="" name="name";
run;

title;
proc print data=sashelp.class;
run;
quit;
ods html close;
```

**Program Description**

**Set the options and goptions.** The OPTIONS statement sets the global options. The GOPTIONS statement set the graphical options.

```plaintext
options nodate obs=10;
goptions xpixels=500 ypixels=400;
```

**Specify that no page break is created.** By default, there is a page break that resolves as a horizontal rule between the graph and the table when the two procedures are run. The PAGEBREAK= NO suboption specifies that no page break is drawn between the two output objects.

For information about the OPTIONS option, see “Example 7: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information” on page 532 and “OPTIONS ( DOC= | <suboption(s)>)” on page 399.

```plaintext
ods html options(pagebreak='no');
```

**Create the graph.** The SYMBOL statements and the GPLOT procedure create the graph.

```plaintext
title "Student Correlation";
symbol1 font="albnay amt" value='O' height=15pt color=pink;
symbol2 font="albnay amt" value='X' height=15pt color=lib;
proc gplot data=sashelp.class;
plot height*weight=sex / des="" name="name";
run;
```

**Print the output.** The PRINT procedure prints the data set.
title;
proc print data=sashelp.class;
run;
quit;
ods html close;
Output

Output 6.37  Default Output with Page Break

### Student Correlation

![Graph showing student correlation between Height and Weight.](image)

<table>
<thead>
<tr>
<th>Obs</th>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69.0</td>
<td>112.5</td>
</tr>
<tr>
<td>2</td>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84.0</td>
</tr>
<tr>
<td>3</td>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98.0</td>
</tr>
<tr>
<td>4</td>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
<td>102.5</td>
</tr>
<tr>
<td>5</td>
<td>Henry</td>
<td>M</td>
<td>14</td>
<td>63.5</td>
<td>102.5</td>
</tr>
<tr>
<td>6</td>
<td>James</td>
<td>M</td>
<td>12</td>
<td>57.3</td>
<td>83.0</td>
</tr>
<tr>
<td>7</td>
<td>Jane</td>
<td>F</td>
<td>12</td>
<td>59.8</td>
<td>84.5</td>
</tr>
<tr>
<td>8</td>
<td>Janet</td>
<td>F</td>
<td>15</td>
<td>62.5</td>
<td>112.5</td>
</tr>
<tr>
<td>9</td>
<td>Jeffrey</td>
<td>M</td>
<td>13</td>
<td>62.5</td>
<td>84.0</td>
</tr>
<tr>
<td>10</td>
<td>John</td>
<td>M</td>
<td>12</td>
<td>59.0</td>
<td>99.5</td>
</tr>
</tbody>
</table>
Output 6.38   Output with No Page Break

See Also

- Appendix 3, “ODS and the HTML Destination,” on page 1101

Statements

- “ODS MARKUP Statement” on page 488
- “ODS Tagset Statement” on page 772
ODS HTML5 Statement

Opens, manages, or closes the HTML5 destination, which produces HTML 5.0 output that contains embedded style sheets.

Valid in: Anywhere
Category: ODS: Third-Party Formatted
Defaults: The default style is HTMLBlue. SVG is the default Universal Printer and device driver for the ODS HTML5 destination.

Interaction: By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC PLOT or PROC CHART), ODS formats the output in SAS Monospace font. If you are creating output that is viewed in an operating environment where SAS software is not installed, this output is not displayed correctly. This is because without SAS, the SAS Monospace font is not recognized. To make your document display correctly, include the following statement before your SAS program:

```
OPTIONS FORMCHAR="|----|+|---+=|-/<>*";
```

z/OS specifics: If you use graphics that are created with either the ACTXIMG or JAVAIMG device drivers in the z/OS operating environment, then specify either the GPATH= option or the PATH= option in the ODS HTML5 statement.

Note: If you want to customize your markup output, continue to use ODS MARKUP for that purpose. The HTML5 destination does not permit user customization.

See: For details about viewing SVG graphics, see “Browser Support for Viewing SVG Files” in SAS/GRAPH: Reference.

Examples: “Example: ODS Graphics SVG Graph in an HTML5 File” on page 452
“Adding a Video to HTML5 Output” in SAS Output Delivery System: Advanced Topics
“Adding Audio to HTML5 Output” in SAS Output Delivery System: Advanced Topics

Syntax

```
ODS HTML5 (<ID=> identifier) < action> ;
ODS HTML5 (<ID=> identifier) <option(s)> ;
```

Summary of Optional Arguments

- **(ID= identifier)**
  Open multiple instances of the same destination at the same time
- **ANCHOR= 'anchor-name'**
  Specify a unique base name for the anchor tag that identifies each output object in the current body file
- **BASE= 'base-text'**
  Specify text to use as the first part of all links and references that ODS creates in output files
- **BODY= 'file-specification' (suboption(s))**
Open a markup family destination and specify the file that contains the primary output that is created by the ODS statement

\textbf{BOX\_SIZING=(CONTENT\_BOX | BORDER\_BOX)}

Specify how to measure the width of cells. Use to override the default value of BOX\_SIZING for a destination

\textbf{CHARSET= character-set}

Specify the character set to be generated in the META declaration for the HTML output

\textbf{CLOSE}

Close the destination and the file that is associated with it

\textbf{CODE= 'file-specification' <(suboption(s))>}

Open the HTML destination and specify the file that contains relevant style information

\textbf{CONTENTS= 'file-specification' <(suboption(s))>}

Open the HTML5 destination and specify the file that contains a table of contents for the output

\textbf{CSSSTYLE='file-specification'<(media-type1<...media-type-10<>)>}

Specify a cascading style sheet to apply to your output

\textbf{DEVICE= device-driver}

Specify a device for the output destination

\textbf{DOM="external-file"}>

Specify that the ODS document object model is written to the SAS log or to an external file.

\textbf{ENCODING= local-character-set-encoding}

Override the encoding for input or output processing (transcodes) of external files

\textbf{EXCLUDE exclusion(s) | ALL | NONE}

Exclude output objects from the destination

\textbf{FRAME= 'file-specification' <(suboption(s))>}

Specify the file that integrates the table of contents, the page contents, and the body file

\textbf{GFOOTNOTE | NOGFOOTNOTE}

Control the location where footnotes are printed in the graphics output

\textbf{GPATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)}

Specify the location for all graphics output that is generated while the destination is open

\textbf{GTITLE | NOGTITLE}

Control the location where titles are printed in the graphics output

\textbf{HEADTEXT= 'markup-document-head'}

Specify HTML tags to place between the <HEAD> and </HEAD> tags in all of the output files.

\textbf{IMAGE\_DPI=}

Specify the image resolution for graphical output

\textbf{METATEXT= 'metatext-for-document-head'}

Specify HTML code to use as the <META> tag between the <HEAD> and </HEAD> tags in all of the HTML output files.

\textbf{NEWFILE= starting-point}

Create a new body file at the specified starting point
OPTIONS ( BITMAP_MODE= | SVG_MODE= | GRAPH_BITMAP_MODE= | STYLE_BITMAP_MODE= | GRAPH_SVG_MODE= | STYLE_SVG_MODE= | SHOW_GRAPH_STYLES= | USE_CSS_RESET)
Specify suboptions and a named value for how images are handled using the HTML5 destination

PACKAGE <package-name>
Specify that the output from the destination be added to an ODS package

PAGE= 'file-specification' <(suboption(s))>
Open the HTML destination and specify the file that contains a description of each page of the body file and contains links to the body file

PARAMETERS= (parameter-pair-1 ... parameter-pair-n)
Write the specified parameters between the tags that generate dynamic graphics output

PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)
Specify the location of an aggregate storage location or a SAS catalog for all markup files

RECORD_SEPARATOR= 'alternative-separator' | NONE
Specify an alternative character or string to separate lines in the output files

SELECT selection(s) | ALL | NONE
Select output objects for the destination

SGE=ON | YES | OFF | NO
Generate a file that can be edited with the ODS Graphics Editor

SHOW
Write to the SAS log the current selection or exclusion list for the destination

STYLE= style-template
Specify a style template to use in writing output files

STYLESHEET= 'file-specification' <(suboption(s))>
Open the HTML destination and place style information for output into an external file, or read style sheet information from an existing file

TEXT= text-string
Insert text into your document

TRANTAB= 'translation-table'
Specify a translation table to use when transcoding a file for output

---

**Without Arguments**
If you use the ODS HTML5 statement without an action or options, then it opens the HTML5 destination and creates HTML5 output.

**Actions**
The following actions are available for the ODS HTML5 statement.

**CLOSE**
closes the destination and any files that are associated with it.

**Tip** When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

**EXCLUDE exclusion(s) | ALL | NONE**
excludes one or more output objects from the destination.

**Default** NONE
**Restriction**
A destination must be open for this action to take effect.

**See**
“ODS EXCLUDE Statement” on page 321

**SELECT**

**selection(s) | ALL | NONE**
selects output objects for the specified destination.

**Default**
ALL

**Restriction**
A destination must be open for this action to take effect.

**See**
“ODS SELECT Statement” on page 758

**SHOW**
writes the current selection list or exclusion list for the destination to the SAS log.

**Restriction**
The destination must be open for this action to take effect.

**Tip**
If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

**See**
“ODS SHOW Statement” on page 771

**Optional Arguments**

**ANCHOR= 'anchor-name'**
specifies a unique base name for the anchor tag that identifies each output object in the current body file.

Each output object has an anchor tag for the contents, page, and frame files to reference. The links and references are automatically created by ODS. The links and references point to the name of an anchor. Therefore, each anchor name in a file must be unique.

*anchor-name*
is the base name for the anchor tag that identifies each output object in the current body file.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR= 'TABULATE', then ODS names the first anchor `tabulate`. The second anchor is named `tabulate1`; the third is named `tabulate2`, and so on.

**Restrictions**
Each anchor name in a file must be unique.

Only alphanumeric values, the special characters "$ - _ + ! * ' () , " and reserved characters used for their reserved purposes can be used unencoded within a URL.

**Requirement**
You must enclose *anchor-name* in quotation marks.

**Interaction**
If you open a file to append to it, be sure to specify a new anchor name to prevent writing the same anchors to the file again. ODS does not recognize anchors that are already in a file when it opens the file.
Tips

You can change anchor names as often as you want by specifying the ANCHOR= option in a markup family statement anywhere in your program. After you have specified an anchor name, it remains in effect until you specify a new one.

Specifying new anchor names at various points in your program is useful when you want other web pages to link to specific parts of your markup language output. Because you can control where the anchor name changes, you know in advance what the anchor name is at those points.

An anchor-name must begin with a letter ([A-Za-z]) and can be followed by any number of letters, digits ([0-9]), hyphens (-), underscores (_), colons (:), and periods (.).

**BASE= 'base-text'**

specifies the text to use as the first part of all links and references that ODS creates in the output files.

*base-text*

is the text that ODS uses as the first part of all links and references that ODS creates in the file.

Consider this specification:

```plaintext
BASE= 'http://www.your-company.com/local-url/'
```

In this case, ODS creates links that begin with the string http://www.your-company.com/local-url/. The appropriate anchor-name completes the link.

**Requirement**

You must enclose *base-text* in quotation marks.

**BODY= 'file-specification' (suboption(s))**

opens a markup family destination and specifies the file that contains the primary output that is created by the ODS statement. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

*file-specification*

specifies the file, fileref, or SAS catalog to write to.

*file-specification* is one of the following:

**external-file**

is the name of an external output file.

**Requirement**

You must enclose *external-file* in quotation marks.

**fileref**

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

**Restriction**

The BODY=fileref option cannot be used in conjunction with the NEWFILE= option.
entry.markup

specifies an entry in a SAS catalog to write to.

Interaction

If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

(suboption(s))

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)

enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 449.

(NO_BOTTOM_MATTER)

specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 449.

(NO_TOP_MATTER)

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 450.

(TITLE=’title-text’)

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

See For complete documentation about the TITLE= suboption, see “(TITLE=’title-text’)” on page 450.

(URL= ’Uniform-Resource-Locator’)

See For complete documentation about the URL= suboption, see “(URL= ’Uniform-Resource-Locator’)” on page 451.

Alias

FILE=

Interaction

Using the BODY= option in an ODS markup family statement that refers to an open ODS markup destination forces ODS to close the destination and all associated files. ODS then opens a new instance of the destination. For more information see “Opening and Closing the MARKUP Destination” on page 517.

Note

For some values of TAGSET=, this output is an HTML file. For other TAGSET= values, the output is an XML file, and so on.

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)

specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS
For information about using the SAS Registry, see “Changing SAS Registry Settings for ODS” on page 40.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing.

**CHARSET= character-set**

specifies the character set to be generated in the META declaration for the HTML output.

See For information about the CHARSET= option, see “CHARSET= Option” in SAS National Language Support (NLS): Reference Guide.

**CODE= 'file-specification' <(suboption(s))>**

opens a markup family destination and specifies the file that contains accompanying programming code, such as JavaScript or XSL (Extensible Stylesheet Language). These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

*file-specification*

specifies the file, fileref, or SAS catalog to write to.

*file-specification* is one of the following:

**external-file**

is the name of an external output file.

Requirement You must enclose *external-file* in quotation marks.

**fileref**

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.


**entry.markup**

specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

**suboption(s)**

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

**(DYNAMIC)**

enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 449.
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator’)” on page 451.

CONTENTS= 'file-specification' <(suboption(s))>
opens the HTML5 destination and specifies the file that contains a table of contents for the output. These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.

• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

entry.markup
specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 449.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 449.
(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top
of the output file. For HTML 4.0, the NO_TOP_MATTER option removes
the style sheet.

See  For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 450.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to
appear in the browser window title bar.

title-text
is the text in the metadata of a file that indicates the title.

See  For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 450.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the
filename) in all the links and references that it creates and that point to the
file.

See  For complete documentation about the URL= suboption, see “(URL='Uniform-Resource-Locator’)” on page 451.

CSSSTYLE='file-specification'(<(media-type1<..imedia-type-10>))>
specifies a cascading style sheet to apply to your output.

file-specification
specifies a file, fileref, or URL that contains CSS code.

file-specification is one of the following:

"external-file"
is the name of the external file.

Requirement  You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the
FILENAME statement to assign a fileref.

See  For information about the FILENAME statement, see SAS Statements: Reference.

"URL"
is a URL to an external file.

Requirement  You must enclose URL in quotation marks.

(media-type-1<..media-type-10>)
specifies one or more media blocks that correspond to the type of media that your
output is rendered on. CSS uses media type blocks to specify how a document is
to be presented on different media: on the screen, on paper, with a speech
synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not
contained in any media blocks. By using the media-type suboption, in addition to
the general CSS code, you can import the section of a CSS file intended only for a specific media type.

**Default**

If no **media-type** is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

**Range**

You can specify up to ten different media types.

**Requirements**

You must enclose **media-type** in parentheses.

You must specify **media-type** next to the **file-specification** specified by the CSSSTYLE= option.

**Tip**

If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

** Interaction**

If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

**See**

For an example of a valid for ODS CSS file, see “Example 6: Applying a CSS File to ODS Output” on page 527.

**DOM**<="external-file">

specifies that the ODS document object model is written to the SAS log or an external file.

**external-file**

is the name of an external output file.

**Requirement**

You must enclose **external-file** in quotation marks.

**See**

For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in *SAS Output Delivery System: Advanced Topics*.

**DEVICE= device-driver**

specifies the name of a device driver. ODS automatically selects an optimal default device for each open output destination.

The following table lists the default devices for the most common ODS output destinations. These default devices are used when graphics are created using SAS/GRAPH or ODS Graphics. In the third maintenance release of SAS 9.4, EPUB3 is the default EPUB destination. EPUB2 was the default EPUB version in prior releases of SAS 9.4. This default is set in the Registry. For a complete list of supported devices and file types, see “Supported File Types for Output Destinations” on page 349.

**Table 6.13 Default Devices for ODS Output Destinations**

<table>
<thead>
<tr>
<th>Output Destination</th>
<th>Default Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPUB2</td>
<td>PNG</td>
</tr>
<tr>
<td>EPUB3 (EPUB)</td>
<td>SVG</td>
</tr>
</tbody>
</table>
Tips  Specifying a device on the ODS DEVICE= option takes precedence over the SAS global option and the graphics option.

For information about using the SAS Registry to change defaults, see “Changing SAS Registry Settings for ODS” on page 40.


ENCODING= local-character-set-encoding  overrides the encoding for input or output processing (transcodes) of external files.

See  For information about the ENCODING= option, see SAS National Language Support (NLS): Reference Guide.

FRAME= 'file-specification' <(suboption(s))>  opens a markup family destination and, for HTML output, specifies the file that integrates the table of contents, the page contents, and the body file. If you open the frame file, then you see a table of contents, a table of pages, or both, as well as the body file. For XML output, FRAME= specifies the file that contains the DTD. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification  specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file  is the name of an external output file.

Requirement  You must enclose external-file in quotation marks.
fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

entry.markup
specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 449.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 449.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 450.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

title-text
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text’)” on page 450.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 451.
Restriction  If you specify the FRAME= option, then you must also specify the CONTENTS= option, the PAGE= option, or both.

Example  “Example 2: Creating an XML File and a DTD” on page 520

GFOOTNOTE | NOGFOOTNOTE
controls the location where footnotes are printed in the graphics output.

GFOOTNOTE
writes footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

NOGFOOTNOTE
writes footnotes that are created by ODS, which appear outside the graph borders.

Default  GFOOTNOTE

Restrictions  Footnotes that are displayed by a markup language statement support all SAS/GRAPH FOOTNOTE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH FOOTNOTE statement, see “FOOTNOTE Statement” in SAS/GRAPH: Reference.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

GPATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)
specifies the location for all graphics output that is generated while the destination is open. Use this option when you want to write graphics output files to a location different that specified by the PATH= option for markup files. If you specify an invalid filename, the ActiveX and Java devices send output to the default filename. Other devices create the file as a directory and write output to that directory using the default filename. For information about how ODS names catalog entries and external files, see SAS/GRAPH: Reference.

'aggregate-file-storage-location'
specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement  You must enclose aggregate-file-storage-location in quotation marks.

fileref
is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction  If you specify a fileref in the GPATH= option, then ODS does not use information from the GPATH= option when it constructs links.

See  For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.
libref.catalog
  specifies a SAS catalog to write to.

URL= 'Uniform-Resource-Locator' | NONE
  specifies a URL for file-specification.

Uniform-Resource-Locator
  is the URL that you specify. ODS uses this URL instead of the filename in all
  the links and references that it creates to the file.

Requirement You must enclose Uniform-Resource-Locator in quotation
  marks.

NONE
  specifies that no information from the GPATH= option appears in the links or
  references.

Tip This option is useful for building output files that can be moved from one
  location to another. If the links from the contents and page files are
  constructed with a simple URL (one name), then they resolve, as long as
  the contents, page, and body files are all in the same location.

Default If you omit the GPATH= option, then ODS stores graphics in the location
  that is specified by the PATH= option. If you do not specify the PATH= option,
  then ODS stores the graphics in the current directory. For more
  information, see the PATH= option.

GTITLE | NOGTITLE
  controls the location where titles are printed in the graphics output.

GTITLE
  writes the title that is created by SAS/GRAPH, the SGPLOT procedure, the
  SGPANEL procedure, or the SGSCATTER procedure. The title appears inside
  the graph borders.

NOGTITLE
  writes the title that is created by ODS, which appears outside of the graph
  borders.

Default GTITLE

Restrictions Titles that are displayed by any markup language statement support
  most SAS/GRAPH TITLE statement options. The font must be valid
  for the browser. Options that ODS cannot handle, such as text angle
  specifications, are ignored. For details about the SAS/GRAPH TITLE
  statement, see TITLE statement.

This option applies only to SAS programs that produce one or more
  device-based graphics, or graphics created by the SGPLOT procedure,
  the SGPANEL procedure, or the SGSCATTER procedure.

HEADTEXT= 'markup-document-head'
  specifies markup tags to place between the <HEAD> and </HEAD> tags in all of
  the output files.

markup-document-head
  specifies the markup tags to place between the <HEAD> and </HEAD> tags.

Restriction HEADTEXT= cannot exceed 256 characters.
### Requirement
You must enclose `markup-document-head` in quotation marks.

### Tips
ODS cannot parse the markup that you supply. It should be well-formed markup that is correct in the context of the `<HEAD>` and `</HEAD>` tags.

Use the `HEADTEXT=` option to define programs (such as JavaScript) that you can use later in the file.

### (ID= `identifier`) enables you to run multiple instances of the same destination at the same time. Each instance can have different options.

`identifier` specifies another instance of the destination that is already open. `identifier` is numeric or a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numeric characters.

### Requirement
You must specify the ID= option immediately after the destination name.

### Tip
You can omit the ID= option and instead use a name or a number to identify the instance.

### Example
“Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594

### IMAGE_DPI=
specifies the image resolution for graphical output.

<table>
<thead>
<tr>
<th>Alias</th>
<th>DPI=</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>96</td>
</tr>
</tbody>
</table>

**CAUTION**
Caution: When using high DPI= or DPI_IMAGE values (values over 1000), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500M or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.

### METATEXT= "metatext-for-document-head"
specifies HTML code to use as the `<META>` tag between the `<HEAD>` and `</HEAD>` tags of all of the HTML output files.

"metatext-for-document-head"
specifies the HTML code that provides the browser with information about the document that it is loading. For example, this attribute could specify the content type and the character set to use.

### Requirement
You must enclose `metatext-for-document-head` in quotation marks.

### Default
If you do not specify METATEXT=, then ODS writes a simple `<META>` tag, which includes the content-type of the document and the character set to use, to all the HTML files that it creates.
Restriction  METATEXT= cannot exceed 256 characters.

Tip  ODS cannot parse the HTML code that you supply. It should be well-formed HTML code that is correct in the context of the <HEAD> tags. If you are using METATEXT= as it is intended, then your META tag should look like this:

\[
<\text{META your-metatext-is-here}>
\]

NEWFILE= starting-point
creates a new body file at the specified starting-point.

starting-point
is the location in the output where you want to create a new body file.

ODS automatically names new files by incrementing the name of the body file. In the following example, ODS names the first body file REPORT.XML. Additional body files are named REPORT1.XML, REPORT2.XML, and so on.

Example:

BODY= 'REPORT.XML'

starting-point is one of the following:

BYGROUP
starts a new file for the results of each BY group.

NONE
writes all output to the body file that is currently open.

OUTPUT
starts a new body file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

Alias  TABLE

PAGE
starts a new body file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.

PROC
starts a new body file each time you start a new procedure.

Default  NONE

Restriction  The NEWFILE= option cannot be used in conjunction with the BODY=fileref option.

Tips  If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first body file MAY5.XML. Additional body files are named MAY6.XML, MAY7.XML, and so on.

Example:

BODY= 'MAY5.XML'
OPTIONS ( BITMAP_MODE= | SVG_MODE= | GRAPH_BITMAP_MODE= | STYLE_BITMAP_MODE= | GRAPH_SVG_MODE= | STYLE_SVG_MODE | SHOW_GRAPH_STYLES= | USE_CSS_RESET)
specifies suboptions and a named value for how images are handled using the HTML5 destination.

BITMAP_MODE= | GRAPH_BITMAP_MODE= | STYLE_BITMAP_MODE= specifies how all bit mapped images are inserted into the HTML5 document. Use the GRAPH_BITMAP_MODE and the STYLE_BITMAP_MODE to insert your style images and your graph images differently.

ods html5 options (bitmap_mode='object');

Note: The BITMAP_MODE can be overridden by STYLE_BITMAP_MODE or by GRAPH_BITMAP_MODE.

EMBED creates an HTML5 <embed> tag.
<embed src="c:\Public\arrow.png"/>

IMG creates an HTML5 <img> tag that displays the image file.
<img src="c:\Public\arrow.png"/>

Default IMG is the default for displaying bit maps.

INLINE inserts the image data into the HTML5 file. The image data is base64 encoded and is inserted into the document as a URL for bitmaps.

Restriction Inline graphs and images might not be supported by some email clients.

LINK creates a hyperlink that points to the image file.
<a href="c:\Public\arrow.png" My Arrow PNG</a>

OBJECT creates an HTML5 <object> tag that displays the image file. This tag is used to embed multimedia files and applications into your document (audio, video, Java applets, ActiveX, PDF, and Flash).
<object data="c:\Public\arrow.png"</object>

SVG_MODE= | GRAPH_SVG_MODE= | STYLE_SVG_MODE= specifies how all SVG images are inserted into the HTML5 document. Use the GRAPH_SVG_MODE and the STYLE_SVG_MODE to insert your style images and your graph images differently.

ods html5 options (svg_mode='img');

Note: The SVG_MODE can be overridden by STYLE_SVG_MODE or by GRAPH_SVG_MODE.

EMBED creates an HTML5 <embed> tag.
<embed src="c:\Public\arrow.svg"/>

IMG creates an HTML5 <img> tag that displays the image file.
Restriction

In 9.4, an image map is not generated using SVG with ODS Graphics. The image map data that is used to produce tooltips and links is written directly in the SVG and is not part of the HTML5 document.

See

For information about the browsers that support SVG graphs and compatibility mode, see “Browser Support for Viewing SVG Files” in *SAS/GRAPH: Reference*.

INLINE

inserts the image data into the HTML5 file. The image data is base64 encoded and is inserted into the document as a URL for bitmaps.

Default

INLINE is the default for displaying SVG.

Restrictions

The INLINE value for SVG_MODE is not supported on z/OS.

In 9.4, an image map is not generated using SVG with ODS Graphics. The image map data that is used to produce tooltips and links is written directly in the SVG and is not part of the HTML5 document.

Inline graphs and images might not be supported by some email clients.

LINK

creates a hyperlink that points to the image file.

<object data="c:\Public\arrow.svg" <object>

OBJECT

creates an HTML5 <object> tag that displays the image file. This tag is used to embed multimedia files and applications into your document (audio, video, Java applets, ActiveX, PDF, and Flash).

<object data="c:\Public\arrow.svg" </object>

SHOW_GRAPH_STYLES= 'YES' | 'ON' | 'TRUE' | 'NO' | 'OFF' | 'FALSE'

specifies that the output should contain elements from the graph style that is specified for CSS.

YES | ON | TRUE

specifies that the output should contain elements from the graph style that is specified for CSS. These values must be enclosed in parentheses.

NO | OFF | FALSE

specifies that the output should not contain elements from the graph style that is specified for CSS. These values must be enclosed in parentheses.

Example

ods html5 file=mycss style=styles.seaside dev=svg gpath=gout
options(show_graph_styles='yes' svg_mode='embed');

USE_CSS_RESET= 'ON' | 'OFF'

turns on or off the default CSS styles information. This option works well with the SHOW_GRAPH_STYLES= suboption to provide a starting point to creating a style that works with the CSSSTYLE= option.
OFF
  turns off the default CSS styles information.

ON
  turns on the default CSS styles information.

Requirement  suboption(s) must be enclosed in parentheses.

PACKAGE <package-name>
specifies that the output from the destination be added to a package.

package-name
  specifies the name of a package that was created with the ODS PACKAGE
  statement. If no name is specified, then the output is added to the unnamed
  package that was opened last.

See     “ODS PACKAGE Statement ” on page 554

Example “Example 1: Creating an ODS Package” on page 558

PAGE= 'file-specification' <(suboption(s))>
opens a markup family destination and specifies the file that contains a description
of each page of the body file, and contains links to the body file. ODS produces a new
page of output whenever a procedure requests a new page. These files remain open
until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE
  statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes
  the first file and opens the second file.

file-specification
  specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
  is the name of an external output file.

Requirement  You must enclose external-file in quotation marks.

fileref
  is a file reference that has been assigned to an external file. Use the
  FILENAME statement to assign a fileref.

See  For information about the FILENAME statement, see “FILENAME
Statement” in SAS Statements: Reference.

entry.markup
  specifies an entry in a SAS catalog to write to.

Interaction  If you specify an entry name, you must also specify a library
  and catalog. See the discussion of the PATH= option.

suboption(s)
  specifies one or more suboptions in parentheses. Suboptions are instructions for
  writing the output files. Suboptions can be the following:
(DYNAMIC)

enables you to send output directly to a web server instead of writing it to a file.

See  For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 449.

(NO_BOTTOM_MATTER)

specifies that no ending markup language source code be added to the output file.

See  For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 449.

(NO_TOP_MATTER)

specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See  For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 450.

(TITLE=’title-text’)

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

TITLE

is the text in the metadata of a file that indicates the title.

See  For complete documentation about the TITLE= suboption, see “(TITLE=’title-text’)” on page 450.

(URL= ’Uniform-Resource-Locator’)

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See  For complete documentation about the URL= suboption, see “(URL= ’Uniform-Resource-Locator’)” on page 451.

Interaction

The SAS system option PAGESIZE= has no effect on pages in HTML output except when you are creating batch output. For information about the PAGESIZE= option, see “PAGESIZE= System Option” in SAS System Options: Reference.

PARAMETERS= (parameter-pair-1 ... parameter-pair-n)

writes the specified parameters between the tags that generate dynamic graphics output.

parameter-pair

specifies the name and value of each parameter. parameter-pair has the following form:

’parameter-name’= ’parameter-value’

parameter-name

is the name of the parameter.
parameter-value is the value of the parameter.

Requirement  You must enclose parameter-name and parameter-value in quotation marks.

Interaction Use PARAMETERS= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA, JAVMETA, or ACTIVEX options in the GOPTIONS statement.

See  SAS/GRAPH: Reference for valid parameters for the Graph Applet, Map Applet, Contour Applet, and the MetaView Applet.

PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)

specifies the location of an aggregate storage location or a SAS catalog for all markup files. If the GPATH= option is not specified, all graphics output files are written to the "aggregate-file-storage-specification" or libref.

'aggregate-file-storage-location'
specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement  You must enclose aggregate-file-storage-location in quotation marks.

fileref
is a file reference that has been assigned to an aggregate storage location. Use theFILENAME statement to assign a fileref.

Interaction  If you use a fileref in the PATH= option, then ODS does not use information from PATH= when it constructs links.

See  For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

libref.catalog
specifies a SAS catalog to write to.

See  For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Statements: Reference.

URL= 'Uniform-Resource-Locator' | NONE
specifies a URL for the file-specification.

Uniform-Resource-Locator
is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

NONE
specifies that no information from the PATH= option appears in the links or references.

Tip  This option is useful for building output files that can be moved from one location to another. The links from the contents and page files must be constructed with a single-name URL, and the contents, page, and body files must be in the same location.
Interaction
If you use the BODY= or FILE= external file option in conjunction with the PATH= option, the external file specification should not include path information.

RECORD_SEPARATOR= 'alternative-separator' | NONE
specifies an alternative character or string that separates lines in the output files.

Different operating environments use different separator characters. If you do not specify a record separator, then the files are formatted for the environment where you run the SAS program. However, if you are generating files for viewing in a different operating environment that uses a different separator character, then you can specify a record separator that is appropriate for the target environment.

alternative-separator represents one or more characters in hexadecimal or ASCII format. For example, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= '0D0A'x

Operating Environment Information
In a mainframe environment, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= '0D25'x

Requirement You must enclose alternative-separator in quotation marks.

NONE produces the markup language that is appropriate for the environment where you run the SAS job.

Windows Specifics
In a mainframe environment, by default, ODS produces a binary file that contains embedded record separator characters. This binary file is not restricted by the line-length restrictions on ASCII files. However, if you view the binary files in a text editor, then the lines run together. If you want to format the files so that you can read them with a text editor, then use RECORD_SEPARATOR= NONE. In this case, ODS writes one line of markup language at a time to the file. When you use a value of NONE, the logical record length of the file that you are writing to must be at least as long as the longest line that ODS produces. If the logical record length of the file is not long enough, then the markup language might wrap to another line at an inappropriate place.

Aliases RECSEP=
RS=

SGE=ON | YES | OFF | NO
Generates a file that can be edited only with the ODS Graphics Editor. The file created has an extension of .sge.

See For details about using the ODS Graphics Editor to create SGE files, see SAS ODS Graphics Editor: User's Guide.

Example ods html5 sge=on;
proc sgplot data=sashelp.class;
STYLE= style-template
specifies the style template to use in writing the output files.

style-template
describes how to display the presentation aspects (color, font face, font size, and so on) of your SAS output. A style template determines the overall appearance of the documents that use it. Each style template consists of style elements.

Interaction
The STYLE= option is not valid when you are creating XML output.

Note
If you are using SAS Studio, you do not need to specify the STYLE= option. You can go to Preferences ➔ Results and change the style from the drop-down list for your selected destination.

See
For a complete discussion of style templates, see “TEMPLATE Procedure: Creating a Style Template” in SAS Output Delivery System: Procedures Guide.

Default
If you do not specify a style template, then ODS uses the file that is specified in the SAS registry subkey ODS ➔ DESTINATIONS ➔ MARKUP. By default, this value specifies Default.

Interaction
If you specify the STYLE= option in an ODS HTML5 statement and subsequently need PROC PRINT output to use new style templates in another ODS HTML5 statement, close the first statement before specifying the second statement.

STYLESHEET= 'file-specification' <(suboption(s))>
opens a markup family destination and places the style information for markup output into an external file, or reads style sheet information from an existing file. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement
You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See
For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.
entry.markup

specifies an entry in a SAS catalog to write to.

Interaction  

If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)

enables you to send output directly to a web server instead of writing it to a file.

See  For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 449.

(NO_BOTTOM_MATTER)

specifies that no ending markup language source code be added to the output file.

See  For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 449.

(NO_TOP_MATTER)

specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See  For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 450.

(TITLE='title-text')

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

title-text

is the text in the metadata of a file that indicates the title.

See  For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 450.

(URL= 'Uniform-Resource-Locator' )

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See  For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator' )” on page 451.

Note  

By default, if you do not specifically send the information to a separate file, then the style sheet information is included in the specified HTML file.

Example  

“Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525
TEXT=\textit{text-string}  
inserts text into your document by triggering the paragraph event and specifying a text string to be assigned to the VALUE event variable.

\textbf{Default}  
By default the TEXT= option is used in a paragraph event.

\textbf{See}  
For information about events and event variables, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in \textit{SAS Output Delivery System: Procedures Guide}.

TRANTAB=\texttt{\textbackslash 'translation-table'}

specifies the translation table to use when transcoding a file for output.

\textbf{See}  

\textbf{Suboptions}

\textbf{(DYNAMIC)}

enables you to send output directly to a web server instead of writing it to a file. This option sets the value of the CONTENTTYPE= style attribute. For more information, see \texttt{CONTENTTYPE=} on page 1015 in PROC TEMPLATE.

\textbf{Default}  
If you do not specify DYNAMIC, then ODS sets the value of HTMLCONTENTTYPE= for writing to a file.

\textbf{Restriction}  
If you specify the DYNAMIC suboption with one of the following options in the ODS HTML statement, then you must specify it for all of these options in that statement.

\begin{itemize}
  \item \texttt{BODY=}
  \item \texttt{CONTENTS=}
  \item \texttt{PAGE=}
  \item \texttt{FRAME=}
  \item \texttt{STYLESHEET=}
  \item \texttt{TAGSET=}
\end{itemize}

\textbf{Requirements}  
You must enclose DYNAMIC in parentheses.

You must specify DYNAMIC next to the \textit{file-specification} specified by the \texttt{BODY=}, \texttt{CONTENTS=}, \texttt{PAGE=}, \texttt{FRAME=}, or \texttt{STYLESHEET=} option, or next to the \textit{tagset-name} specified by the \texttt{TAGSET=} option.

\textbf{(NO\_BOTTOM\_MATTER)}

specifies that no ending markup language source code be added to the output file.

\textbf{Alias}  
\texttt{NOBOT}

\textbf{Requirements}  
You must enclose NO\_BOTTOM\_MATTER in parentheses.

You must specify NO\_BOTTOM\_MATTER next to the \textit{file-specification} specified by the \texttt{BODY=}, \texttt{CONTENTS=}, \texttt{PAGE=}, \texttt{FRAME=}, or \texttt{STYLESHEET=} option, or next to the \textit{tagset-name} specified by the \texttt{TAGSET=} option.
If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

**Interactions**

The NO_BOTTOM_MATTER suboption, in conjunction with the NO_TOP_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

**Tip**

If you want to leave a body file in a state that you can append to with ODS, then use NO_BOTTOM_MATTER with the file-specification BODY= option in any markup language statement.

**See**

The NO_TOP_MATTER suboption

(NO_TOP_MATTER)

specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

**Alias**

NOTOP

**Requirements**

You must enclose NO_TOP_MATTER in parentheses.

You must specify NO_TOP_MATTER next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

**Interactions**

The NO_TOP_MATTER suboption, in conjunction with the NO_BOTTOM_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

**See**

The NO_BOTTOM_MATTER suboption and the ANCHOR= option

(TITLE="title-text")

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

**title-text**

is the text in the metadata of a file that indicates the title.

**Requirements**

You must enclose TITLE= in parentheses.

You must enclose title-text in quotation marks.
Tip
If you are creating a web page that uses frames, then it is the TITLE= specification for the frame file that appears in the browser window title bar.

Example
“Example 3: Creating Multiple Markup Output” on page 522

(URL= 'Uniform-Resource-Locator'

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

Requirements
You must enclose URL= 'Uniform-Resource-Locator' in parentheses.

You must enclose Uniform-Resource-Locator in quotation marks.

You must specify URL= 'Uniform-Resource-Locator' next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

Tips
This option is useful for building HTML files that can be moved from one location to another. The links from the contents and page files must be constructed with a single name URL, and the contents, page, and body files must all be in the same location.

You never need to specify this suboption with the FRAME= option because ODS files do not reference the frame file.

Example
“Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

Details

HTML and HTML5 Differences
The ODS HTML5 destination is new for SAS 9.4. There are differences between HTML and HTML5. Refer to the W3C document W3C HTML5 Differences.

HTML Output Defaults
You must use the ODS HTML5 statement if you want output that uses HTML 5.0. The SAS Windowing environment defaults to HTML4. SAS Studio defaults to HTML5. For information about using SAS Studio, see Chapter 4, “SAS Studio and ODS,” on page 93.

Starting in SAS 9.3, in the SAS Windowing environment with the Windows and UNIX operating systems, by default the LISTING destination is closed and the HTML destination is open. Starting in SAS 9.3, you do not have to submit an ODS HTML statement to generate HTML output, and you do not have to use the ODS HTML CLOSE statement to be able to view your output. However, to create LISTING output, you must either submit the ODS LISTING statement or enable the LISTING destination by other means. For more details, see “Working with Output Defaults” on page 21.

Note: HTML and HTML5 use the HTMLBlue style by default.
**HTML5 and Scalable Vector Graphics (SVG)**

The ODS HTML5 statement supports Scalable Vector Graphics (SVG). Scalable Vector Graphics is an XML language for describing two-dimensional vector graphics. SVG is the default file type for the HTML5 destination.

SAS can create SVG documents by using Universal Printers and SAS/GRAPH device drivers. Most often in SAS, the SVG Universal Printers and device drivers are used to create graphs. Graphs can be created by using ODS Graphics or SAS/GRAPH. The ODS HTML5 destination can be used to create SVG documents. SVG is the default Universal Printer and device driver for the ODS HTML5 destination. SVG documents can be stand-alone files or integrated within an HTML5 file.

In order to view SVG documents, you need a viewer or browser that supports Scalable Vector Graphics. For more information, see “Browser Support for Viewing SVG Documents” in *SAS Language Reference: Concepts*.

To view an SVG document in an HTML file, you either create a link to the SVG document, embed the SVG document in the HTML file, or create an SVG graph that is integrated in the HTML. The default value for option SVG_MODE for the HTML5 destination is INLINE. In order to embed the SVG graph, you must specify

```
OPTIONS (SVG_MODE="EMBED")
```

in the ODS HTML5 statement. For more information, see “SVG Documents in HTML Files” in *SAS Language Reference: Concepts*.


For detailed information about the SVG standard, see the W3 documentation at [W3 Scalable Vector Graphics (SVG) document](http://www.w3.org/TR/SVG/).

**HTML5 Support of Embedded Audio and Video**

In the third maintenance release of SAS 9.4, HTML5 supports audio and video. The Report Writing Interface (RWI) is used to accomplish this task. Media methods, AUDIO and VIDEO, generate either an audio tag or a video tag in the HTML5 code. The audio and video elements specify a standard way to embed a video file or an audio file in a web page. How this works depends on the browser that is used.

Information about the RWI methods and the HTML5 tags that support audio and video can be found in the following documentation.

- “AUDIO Method” in *SAS Output Delivery System: Advanced Topics*
- “VIDEO Method” in *SAS Output Delivery System: Advanced Topics*
- HTML5 Audio tag
- HTML5 Video tag
- “RWI Basics” in *SAS Output Delivery System: Advanced Topics*

**Example: ODS Graphics SVG Graph in an HTML5 File**

**Features:**
- ODS HTML5 Statement
- ODS GRAPHICS Statement

**Other features:**
- PROC SGPLOT
Details
To integrate an ODS Graphics SVG graph in an HTML file, you must use the HTML5 destination. By default the HTML5 defaults to putting an SVG graph inline. For this example, `option(SVG_MODE='INLINE')` is not needed, but shows how to specify the option when needed.

Program
```
ods html close;
ods html5 options(svg_mode="inline");
ods graphics /outputfmt=svg;
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2000"d and stock = "IBM");
  title "Stock Trend";
  series x=date y=close;
  series x=date y=low;
  series x=date y=high;
run;
ods html5 close;
ods html;
```

Program Description

**Close the HTML destination.** Close the default HTML destination. The HTML5 destination is being used.
```
ods html close;
```

**Open the HTML5 destination and make the SVG graph appear inline.** Open the HTML5 destination. The SVG_MODE option instructs SAS to place the graph inline.
```
ods html5 options(svg_mode="inline");
```

**Specify the SVG output format for ODS GRAPHICS.** Use ODS GRAPHICS to generate an SVG file.
```
ods graphics /outputfmt=svg;
```

**Use PROC SGPLOT to create a graph.**
```
proc sgplot data=sashelp.stocks
  (where=(date >= "01jan2000"d and stock = "IBM");
  title "Stock Trend";
  series x=date y=close;
  series x=date y=low;
  series x=date y=high;
run;
```

**Close the HTML5 destination.** The ODS HTML5 CLOSE statement closes the HTML5 destination and all the files that are associated with it. Open the HTML5 destination.
```
ods html5 close;
ods html;
```
HTML5 Output

This output is created using the ODS HTML5 destination.

Output 6.39  Integrating an ODS Graphics SVG Graph in an HTML5 File

![](Stock Trend.png)

**ODS LAYOUT ABSOLUTE Statement**

Enables you to specify an exact page location using x and y coordinates. Absolute layout is limited to one page and is supported only by the PRINTER destinations.

**Valid in:** Anywhere

**Category:** ODS: Output Control

**Requirement:** The ODS LAYOUT ABSOLUTE statement must be used with the ODS LAYOUT END statement.

**ODS destination:** ODS LAYOUT ABSOLUTE is supported only for PRINTER destinations (PDF, PS, and PCL).

**Tips:** Absolute layout enables you to specify an exact page location using x and y coordinates. Absolute layout is perfectly suited for static types of output that can be printed on a single page where you want output placed in a specific location. Examples are preprinted forms and cover pages.

Use the ODS REGION statement with the ODS LAYOUT ABSOLUTE statement to create full featured layouts. Refer to “ODS REGION Statement, Absolute” on page 465.

**Example:**

```ods layout absolute y=1.25in x=1in width=6in;
ods region;
   proc print data=sashelp.class;
   run;
ods text= 'layout width=6in';
ods layout end;```
Syntax

ODS LAYOUT ABSOLUTE< option-1>< option-2 ...>

Summary of Optional Arguments

- **HEIGHT=dimension**
  Specify a vertical height of the layout.
- **STYLE=<style-element-name> <[style-attribute-specification(s)]>**
  specifies one or more style elements to use for different parts of the layout.
- **WIDTH=dimension**
  Specify the horizontal width of the layout.
- **X=dimension**
  Specify the horizontal position of the layout.
- **Y=dimension**
  Specify the vertical starting position of the layout.

Optional Arguments

- **HEIGHT=dimension**
  specifies the vertical height of the layout.

  *dimension*
  is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

  Valid units of measure:

  - cm  centimeters
  - em  standard typesetting measurement unit for width
  - ex  standard typesetting measurement unit for height
  - in  inches
  - mm  millimeters
  - pct  a percentage. You can also use the ‘%’ symbol.
  - pt  a printer’s point
  - px  pixels

  Default
  If omitted, the height of the layout defaults to the maximum vertical space needed to display all of the regions.

  Example
  ods layout absolute height=5in;
  proc print data=sashelp.class;
  run;
  ods layout end;

- **STYLE=<style-element-name> <[style-attribute-specification(s)]>**
  specifies the style element to use for the specified locations in the layout.

  Tip
  Font names that contain characters other than letters or underscores must be enclosed in quotation marks.
Example  
ods layout absolute style=[backgroundcolor=yellow];
   proc print data=sashelp.class;run;
   ods layout end;

**WIDTH=dimension**
specifies the horizontal width of the layout.

dimension
is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>centimeters</td>
</tr>
<tr>
<td>em</td>
<td>standard typesetting measurement unit for width</td>
</tr>
<tr>
<td>ex</td>
<td>standard typesetting measurement unit for height</td>
</tr>
<tr>
<td>in</td>
<td>inches</td>
</tr>
<tr>
<td>mm</td>
<td>millimeters</td>
</tr>
<tr>
<td>pct</td>
<td>a percentage. You can also use the ‘%’ symbol.</td>
</tr>
<tr>
<td>pt</td>
<td>a printer’s point</td>
</tr>
<tr>
<td>px</td>
<td>pixels</td>
</tr>
</tbody>
</table>

**Default**
If omitted, the width of the layout defaults to the maximum horizontal space needed to display all of the regions.

Example  
ods layout absolute width=4in;
   proc print data=sashelp.class;
   run;
   ods layout end;

**X=dimension**
specifies the horizontal starting position of the layout. The layout container extends to the right of the X position by the amount specified in the WIDTH= option.

dimension
is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>centimeters</td>
</tr>
<tr>
<td>em</td>
<td>standard typesetting measurement unit for width</td>
</tr>
<tr>
<td>ex</td>
<td>standard typesetting measurement unit for height</td>
</tr>
<tr>
<td>in</td>
<td>inches</td>
</tr>
<tr>
<td>mm</td>
<td>millimeters</td>
</tr>
<tr>
<td>pct</td>
<td>a percentage. You can also use the ‘%’ symbol.</td>
</tr>
<tr>
<td>pt</td>
<td>a printer’s point</td>
</tr>
</tbody>
</table>
Y=dimension

specifies the vertical starting position of the layout. The layout container extends down from the Y position by the amount specified in HEIGHT= option.

dimension

is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- **cm**  centimeters
- **em**  standard typesetting measurement unit for width
- **ex**  standard typesetting measurement unit for height
- **in**  inches
- **mm**  millimeters
- **pct** a percentage. You can also use the ‘%’ symbol.
- **pt**  a printer’s point
- **px**  pixels

Default

If omitted, the Y argument defaults to the current vertical position on the page.

Example

```r
ods layout absolute y=0.5in;
  proc print data=sashelp.class;
  run;
  ods layout end;
```

**Details**

**Using Absolute Layout**

The ODS LAYOUT statements enable you to create custom reports where you can easily mix graphics, images, text, and tables, and arrange them on a page. Absolute Layout enables you to specify an exact page location using x and y coordinates. Each location needs to be explicitly placed to ensure that there is no unintended overlap.

The statement used to create an absolute layout is ODS LAYOUT ABSOLUTE. ODS LAYOUT ABSOLUTE follows the traditional ODS statements 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.

**Note:** Use the ODS REGION statement, absolute version with the ODS LAYOUT ABSOLUTE statement. The ODS REGION statement cannot be used alone.
Maintaining program code written using absolute layout can be challenging. When one container's position is altered, you might need to manually alter all of the other containers to maintain your report's integrity.

You are not limited to a single ODS layout type. An absolute layout can contain gridded layouts and a gridded layout can contain absolute layouts.

Only ODS PRINTER destinations support absolute layout.

**Reasons to Use Absolute Layout**

Absolute layout is restricted to a single page. If the output is too large to fit in the fixed-size container, the output is discarded. You receive a blank region and a warning in your log. Absolute layout is perfectly suited for static types of output.

Absolute layout works well to create the following types of output:

- cover page
- static data
- preprinted form
- single page output

*Note:* Absolute layout works well to create preprinted forms.

**Example: Absolute Layout Using the Region Container**

**Features:**

- ODS LAYOUT ABSOLUTE statement
- ODS LAYOUT END statement
- ODS REGION statement
- ODS REGION
- Y option
- X option
- HEIGHT option
- WIDTH option
- ODS PDF statement
- PROC TEMPLATE
- ODS ESCAPECHAR statement
- ODS TEXT=

**Other features:**

- FOOTNOTE statement
- TITLE statement
- GCHART procedure
- GOPTIONS
- REPORT procedure

**Details**

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

**Program**

```
options nodate nonumber;
```
Example 1: Absolute Layout Using the Region Container

 proc template;
   define style Styles.OrionCalloutBlock;
      parent = Styles.Pearl;
      style LayoutRegion/
         background=cxbbb2e0;
   end;
run;
ods escapechar="~";
title "~{style [preimage='c:\Public\orionstarHeader.jpg' width=100pct
background=cx494068 color=cxbbb2e0 font_size=32pt] Our Company }";
footnote "~{style [font_size=10pt just=right color=cxbbb2e0]
Using ODS Absolute Layout Features.}";
ods pdf file="OrionstarCoInfo.pdf" notoc nogtitle nogfootnote;
ods layout absolute;
ods text="~{style [preimage='c:\Public\starLarge.gif'
font_style=italic font_size=20pt color=cxbbb2e0]
Who we are...}"
ods region y=0.5in x=1in width=6in;
   ods text="The Orion Star Sports & Outdoors Company is a fictional
international retail company that sells sports and outdoor products.
The headquarters is based in the United States. Retail stores are
situated in a number of other countries including Belgium, Holland,
Germany, the United Kingdom, Denmark, France, Italy, Spain, and
Australia.";
ods region y=1.25in x=1in width=4in;
   ods text="Products are sold in physical retail stores, by mail order
catalogs, and through the Internet. Customers who sign up as members
of the Orion Star Club organization can receive favorable special
offers; therefore, most customers enroll in the Orion Star Club. The
sales data in this scenario includes only the purchases of Orion
Star Club members from 1998 through 2002.";
ods region y=2.5in height=1in width=3in;
   ods text="~{newline}Products are organized in a hierarchy
consisting of three levels:";
   ods text="Product Line"
ods text="Product Category"
ods text="Product Group";
ods region y=4.75in height=1in width=5in;
   ods text="~{style [preimage='c:\Public\starLarge.gif'
font_style=italic font_size=20pt color=cxbbb2e0]
Where we generate our profit...}"
ods region y=5.35in width=4.75in height=3.75in;
goptions device=png htext=.1in;
proc gchart data=sashelp.orsales;
  pie product_category / sumvar=profit
    value=none
    percent=outside
    slice=outside;
run;
quit;

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

ods pdf style=Styles.OrionCalloutBlock;
ods region y=1.0625in x=6in width=2in height=1in;
  ods text="~{style [background=cx494068 color=cxbbb2e0
      font_size=24pt just=center font_style=italic width=100pct]
  Our Mission }";
  ods text="~{style [font_style=italic vjust=center font_size=10pt
      just=center]To deliver the best quality sporting equipment,
    accessories, and outdoor equipment for all seasons at the
    most affordable prices.}";
ods region y=2.1875in x=6in width=2in height=1in;
  ods text="~{style [background=cx494068 color=cxbbb2e0
      font_size=24pt just=center font_style=italic width=100pct]
  Our Vision }";
  ods text="~{style [font_style=italic vjust=center font_size=10pt
      just=center]To transform the way the world purchases sporting
    and outdoor equipment.}";
ods region y=3.3125in x=6in width=2in height=1in;
  ods text="~{style [background=cx494068 color=cxbbb2e0
      font_size=24pt just=center font_style=italic width=100pct]
  Our Values }";
  ods text="~{style [font_style=italic vjust=center
      font_size=10pt just=center]To stay Customer focused, Swift
    and Agile, Innovative, and Trustworthy.}";
ods region x=6in y=4.4375in width=2in height=1in;
  ods text="~{style [background=cx494068 color=cxbbb2e0
      font_size=24pt just=center font_style=italic width=100pct]
  Our Goal }";
  ods text="~{style [font_style=italic vjust=center font_size=10pt
      just=center]To grow sales by 15% annually while improving
    profit margins through innovative thinking and operational
    efficiencies.}";
ods layout end;
ods pdf close;
Program Description

Set the SAS system options. Set the NODATE and NONUMBER SAS system options.

    options nodate nonumber;

Set up the style template. Create a custom style template based on Styles.Pearl for use in specified sections of the layout.

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

Set up the titles and footnotes for the page. Set the ESCAPECHAR value to "~" and use the ODS ESCAPECHAR style function to customize the titles and footnotes.

    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. }";

Open the PDF destination and write to a file. Write the PDF output to file OrionstarCoInfo.pdf. In this program, we do not want a TOC to be generated nor a graphics title and footnote.

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

Set the layout to absolute layout.

    ods layout absolute;

Create the first region, "Who we are", using ODS TEXT. Add the star image and text to the first region. Note that the ODS REGION statement was not used to create the first region. A region is created for you if the REGION statement is not used.

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

Create the second region using ODS TEXT. Specify exactly where the region text should go on the page by using the Y=, X=, and WIDTH= region options.

    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. ";

Create the third region using ODS TEXT. Specify the location of the third region on the page by using the Y=, X=, and WIDTH= region options.

    ods region y=1.25in x=1in width=4in;
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.

Create the fourth region, "What we sell", using ODS TEXT. Add the star image and the text for the second item in the list.

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

Create the fifth region using ODS TEXT. Specify the location of the fifth region on the page by using the Y=, X=, WIDTH=, and HEIGHT= region options.

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

Create the sixth region, "Where we generate our profit", using ODS TEXT. Add the star image and the text for the third item in the list.

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

Create the seventh region using ODS GCHART. Create a pie chart and place it exactly where you want it to appear on the page. This pie chart shows the company profits.

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

Create the eighth region using PROC REPORT. Create a table showing the company profits. Place the table exactly where you want it to appear on the page. Note that the REPORT region intentionally overlaps part of the GCHART region.

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

---

Change the style in the following regions. Start using the OrionCalloutBlock style template to change the background color of the following regions.

ods pdf style=Styles.OrionCalloutBlock;

Create the ninth region, “Our Mission”, using ODS TEXT. Specify the location of the
ninth region by using the Y=, X=, WIDTH=, and HEIGHT= region options.

ods region y=1.0625in x=6in width=2in height=1in;
ods text="-{style [background=cx494068 color=cxbbb2e0
    font_size=24pt just=center font_style=italic width=100pct]
    Our Mission }";
ods text="-{style [font_style=italic vjust=center font_size=10pt
    just=center]To deliver the best quality sporting equipment,
    accessories, and outdoor equipment for all seasons at the
    most affordable prices.}";

Create the tenth region, “Our Vision”, using ODS TEXT. Specify the location of the
tenth region by using the Y=, X=, WIDTH=, and HEIGHT= region options.

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

Create the eleventh region, “Our Values”, using ODS TEXT. Specify the location of the
eleventh region by using the Y=, X=, WIDTH=, and HEIGHT= region options.

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

Create the twelfth region, “Our Goal”, using ODS TEXT. Specify the location of the
twelfth region by using the Y=, X=, WIDTH=, and HEIGHT= region options.

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

---

End the layout. The ODS LAYOUT END statement ends the layout. The ODS PDF
CLOSE statement closes the PDF destination.
The following example output mixes graphics, images, text, and tables.

## Output 6.40  ODS Absolute Layout – Company Information

See Also

### ODS Statements

- “ODS REGION Statement, Absolute” on page 465
- “ODS LAYOUT END Statement” on page 484
ODS REGION Statement, Absolute

Creates a region container for absolute layouts. A region can contain output (such as text, tables, images, graphs) or nested layout containers. This ODS REGION statement is sandwiched between the ODS LAYOUT ABSOLUTE statement and the ODS LAYOUT END statement. The ODS LAYOUT ABSOLUTE statement manages the ODS LAYOUT destination when producing one page of output for PRINTER destinations.

Valid in: Anywhere
Category: ODS: Output Control
Requirement: The ODS REGION statement for absolute layout must be used with the ODS LAYOUT ABSOLUTE statement.

ODS destination: ODS LAYOUT ABSOLUTE is supported only for PRINTER destinations (PDF, PS, and PCL).

Tips: Absolute layout enables you to specify the exact location on the page to place a layout and region container. Absolute layout is perfectly suited for static types of output that can be printed on a single page where you want output placed in a specific location. Examples are preprinted forms and cover pages.

Regions can have a fixed size or can be dynamically sized.

Syntax

REGION < option-1>< option-2 ...>

Summary of Optional Arguments

- **HEIGHT=dimension**
  Specify a vertical height of the region.

- **STYLE=<<style-element-name> <[style-attribute-specification(s)]>**
  specifies one or more style elements to use for different parts of the layout.

- **WIDTH=dimension**
  Specify the horizontal width of the region.

- **X=dimension**
  Specify the horizontal starting position of the region.

- **Y=dimension**
  Specify the starting vertical position of the region.

Optional Arguments

- **HEIGHT=dimension**
  specifies the vertical height of the region.

  *dimension* is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

  Valid units of measure:

  - cm centimeters
em  standard typesetting measurement unit for width
ex  standard typesetting measurement unit for height
in  inches
mm  millimeters
pct a percentage. You can also use the ‘%’ symbol.
pt  a printer’s point
px  pixels

Default If omitted, the height of the region defaults to the maximum vertical space needed to display all of the output in that region.

Restrictions The height is restricted by the dimensions of the region.

The sum of all region heights cannot exceed the vertical dimension of the layout container.

Example ods layout absolute;
  ods region height=5in;
  proc print data=sashelp.class; run;
ods layout end;

STYLE=<style-element-name> <![style-attribute-specification(s)>]
specifies the style element to use for the specified locations in the layout.

Tip Font names that contain characters other than letters or underscores must be enclosed in quotation marks.

Example ods layout absolute style=[backgroundcolor=yellow];
  ods region style=[backgroundcolor=lightblue];
  proc print data=sashelp.class; run;
ods layout end;

WIDTH=dimension
specifies the horizontal width of the region.

dimension
is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

cm  centimeters
em  standard typesetting measurement unit for width
ex  standard typesetting measurement unit for height
in  inches
mm  millimeters
pct a percentage. You can also use the ‘%’ symbol.
pt  a printer’s point
px  pixels
**Default**
If omitted, the width of the region defaults to the maximum horizontal space needed to display all of the output in that region.

**Restrictions**
The width is restricted by the dimensions of the region.

The sum of all region widths cannot exceed the horizontal dimension of the layout container.

**Example**
```plaintext
ods layout absolute;
ods region width=5in;
   proc print data=sashelp.class; run;
ods layout end;
```

**X=**`dimension`
specifies the horizontal starting position of the layout. The region extends to the right by the amount specified by the WIDTH= option in the ODS REGION statement.

`dimension`
is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- `cm` centimeters
- `em` standard typesetting measurement unit for width
- `ex` standard typesetting measurement unit for height
- `in` inches
- `mm` millimeters
- `pct` a percentage. You can also use the ‘%’ symbol.
- `pt` a printer’s point
- `px` pixels

**Default**
0.

**Example**
```plaintext
ods layout absolute;
ods region x=2.5in;
   proc print data=sashelp.class; run;
ods layout end;
```

**Y=**`dimension`
specifies the starting vertical position of the region within the layout. The region extends down by the amount specified by the HEIGHT option in the ODS REGION statement.

`dimension`
is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- `cm` centimeters
- `em` standard typesetting measurement unit for width
ex  standard typesetting measurement unit for height
in  inches
mm  millimeters
pct a percentage. You can also use the ‘%’ symbol.
pt  a printer’s point
px  pixels

Default  If omitted, the Y argument defaults to the current vertical position within the layout.

Example  ods layout absolute;
         ods region y=5in;
         proc print data=sashelp.class; run;
         ods layout end;

Details

A region container is an area that contains output such as text, tables, graphics, and images. Region containers can be nested. The ODS REGION statement that is for absolute layout must be used within an ODS LAYOUT ABSOLUTE statement block.

See Also

ODS Statements

•  “ODS LAYOUT ABSOLUTE Statement ” on page 454
•  “ODS LAYOUT END Statement” on page 484

ODS LAYOUT GRIDDED Statement

Enables you to arrange output dynamically in a two-dimensional grid structure. For example, gridded layout enforces automatic alignment of grid cells, allows layout to continue onto the next page, dynamically computes the size of a grid cell, and makes it easier to maintain the integrity of the report. Gridded layout is supported for HTML, POWERPOINT, and PRINTER destinations (PDF, PS, and PCL).

Valid in:  Anywhere
Category:  ODS: Output Control
Requirement:  The ODS LAYOUT GRIDDED statement must be used with the ODS LAYOUT END statement.
ODS destination:  Gridded layout is supported for HTML and PRINTER destinations (PDF, PS, and PCL). It is also supported for the ODS destination for PowerPoint.
Tip:  Gridded layout for dynamically sized regions can accommodate dynamic data, can span more than one page, and allows for easier alignment. Programs created using gridded layout are easier to maintain than those created using absolute layout.
Example:  ods layout gridded width=2in;
          ods region;
          proc print data=sashelp.class;
          run;
ods text= 'layout width=2in';
ods layout end;

Syntax

**ODS LAYOUT GRIDDED**<option-1>< option-2 ...>

**Summary of Optional Arguments**

**ADVANCE=**<BYGROUP | EXPLICIT | PROC | TABLE | OUTPUT >
Dynamically populate the layout grid by groups, tables, pages, procedures, and explicitly.

**COLUMN_GUTTER=**dimension
Specify the horizontal space between each column.

**COLUMN_WIDTHS=**dimension
Specify the width of each column specified.

**COLUMNS=**number
Specify the fixed number of columns in a gridded layout.

**HEIGHT=**dimension
Specify the vertical height of the layout.

**ORDER_TYPE=**<ROW_MAJOR | COLUMN_MAJOR>
Populate the grid by rows or columns.

**ROW_GUTTER=**dimension
Specify the vertical space between each row.

**ROW_HEIGHTS=**dimension
Specify the height of each row specified.

**ROWS=**number
Specify the fixed number of rows in the gridded layout.

**STYLE=**<style-element-name> <[style-attribute-specification(s)]>
specifies one or more style elements to use for different parts of the layout.

**WIDTH=**dimension
Specify the horizontal width of the layout.

**X=**dimension
Specify the horizontal starting position of the layout.

**Y=**dimension
Specify the vertical starting position of the layout.

**Without Arguments**
Without arguments, the default is ONE column and ONE region.

**Optional Arguments**

**ADVANCE=**<BYGROUP | EXPLICIT | PROC | TABLE | OUTPUT >
specifies that the grid is explicitly populated or that the grid is dynamically populated by groups, tables, pages, and procedures.

**BYGROUP**
specifies that the gridded layout dynamically advance to a new region for every BYGROUP encountered.

```
ods layout gridded columns=2 advance=bygroup;
```
EXPLICIT
specifies that the gridded layout dynamically populates the region explicitly before moving on to the next region. All output is in ONE region unless ODS REGION statements are being used.

Note If you are changing ADVANCE= options within the same layout and using ADVANCE=EXPLICIT, the output does not advance until after EXPLICIT is used.

PROC
specifies that the gridded layout dynamically populates the region by procedures before moving on to the next region.

TABLE
specifies that the gridded layout dynamically populates the region by tables before moving on to the next region.

OUTPUT
specifies that the gridded layout dynamically populates the region by output before moving on to the next region.

Default EXPLICIT

COLUMNS=number
specifies the fixed number of columns in a gridded layout.

Default 1

COLUMN_GUTTER=dimension
specifies the horizontal space between each column.

dimension
is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>centimeters</td>
</tr>
<tr>
<td>em</td>
<td>standard typesetting measurement unit for width</td>
</tr>
<tr>
<td>ex</td>
<td>standard typesetting measurement unit for height</td>
</tr>
<tr>
<td>in</td>
<td>inches</td>
</tr>
<tr>
<td>mm</td>
<td>millimeters</td>
</tr>
<tr>
<td>pct</td>
<td>a percentage. You can also use the ‘%’ symbol.</td>
</tr>
<tr>
<td>pt</td>
<td>a printer’s point</td>
</tr>
<tr>
<td>px</td>
<td>pixels</td>
</tr>
</tbody>
</table>

Example
ods layout gridded columns=2 column_gutter=1in;

ods region;
   proc print data=sashelp.air(obs=5);
   run;
ods region;
   proc means data=sashelp.class n mean;
COLUMN_WIDTHS=dimension
specifies the width of each column specified. This is a space-delimited list of
horizontal sizes that correspond to each column.

dimension
is a nonnegative number followed by an optional unit of measure. It is not
recommended that you use pixels because of adverse dependencies on resolution
that can differ between destinations.

Valid units of measure:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>centimeters</td>
</tr>
<tr>
<td>em</td>
<td>standard typesetting measurement unit for width</td>
</tr>
<tr>
<td>ex</td>
<td>standard typesetting measurement unit for height</td>
</tr>
<tr>
<td>in</td>
<td>inches</td>
</tr>
<tr>
<td>mm</td>
<td>millimeters</td>
</tr>
<tr>
<td>pct</td>
<td>a percentage. You can also use the ‘%’ symbol.</td>
</tr>
<tr>
<td>pt</td>
<td>a printer’s point</td>
</tr>
<tr>
<td>px</td>
<td>pixels</td>
</tr>
</tbody>
</table>

Restriction A warning is given and the option is ignored when the number of
column widths does not match the number of columns specified. The
number of column widths must be equal to the number of columns
requested in the LAYOUT statement.

Example

```plaintext
ods layout gridded columns=2 column_widths=(2in 2in);
ods region;
    proc print data=sashelp.class;
    run;
ods region;
    proc print data=sashelp.class;
    run;
ods layout end;
```

HEIGHT=dimension
specifies the vertical height of the layout.

dimension
is a nonnegative number followed by an optional unit of measure. It is not
recommended that you use pixels because of adverse dependencies on resolution
that can differ between destinations.

Valid units of measure:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>centimeters</td>
</tr>
<tr>
<td>em</td>
<td>standard typesetting measurement unit for width</td>
</tr>
</tbody>
</table>
ex standard typesetting measurement unit for height
in inches
mm millimeters
pct a percentage. You can also use the ‘%’ symbol.
pt a printer’s point
px pixels

Default If omitted, the height of the layout defaults to the maximum vertical space needed to display all of the regions.

Example
ods layout gridded height=7in;
   proc print data=sashelp.class; run
ods layout end;

ORDER_TYPE=<ROW_MAJR | COLUMN_MAJR>
populates the grid by rows or columns.

COLUMN_MAJR
specifies that the gridded layout first populates all regions in the first column before moving on to the next column.

ROW_MAJR
specifies that the gridded layout first populates all regions in the row before moving on to the next row.

Default ROW_MAJR

Restrictions The ORDER_TYPE= option is valid only in PRINTER destinations.

The gridded layout option ORDER_TYPE=COLUMN_MAJR does not support the gridded region option COLUMN_SPAN. When these options are specified together, the COLUMN_SPAN= option is ignored and a warning message is written to the SAS log.

Examples
This code populates row1 column1, row1 column2, row2 column1, and row2 column2:
ods layout gridded columns=2 rows=2 order_type=row_major;

This example populates row1 column1, row2 column1, row1 column2, and row2 column2:
ods layout gridded columns=2 rows=2 order_type=column_major;

ROWS=number
specifies the fixed number of rows in the gridded layout.

Default If omitted, the ROWS= option defaults to the maximum number of rows needed to populate the regions created in the vertical direction. If there are two columns, then the number of rows is half of the number of regions.

Example
ods layout gridded rows=2 columns=1;

ods region;
   proc print data=sashelp.class(obs=1); run;
ods region;
   proc print data=sashelp.class(obs=1); run;
ods layout end

**ROW_GUTTER=**_dimension_

specifies the vertical space between each row.

*dimension*

is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- cm  centimeters
- em  standard typesetting measurement unit for width
- ex  standard typesetting measurement unit for height
- in  inches
- mm  millimeters
- pct a percentage. You can also use the ‘%’ symbol.
- pt  a printer’s point
- px  pixels

Example:  ods layout gridded rows=2 row_gutter=1in;

ods region;
   proc print data=sashelp.air(obs=5); run;
ods region;
   proc means data=sashelp.class n mean;
    var height weight;
   run;
ods layout end;

**ROW_HEIGHTS=**_dimension_

specifies the height of each row specified. This is a space delimited list of vertical sizes that correspond to each row.

*dimension*

is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- cm  centimeters
- em  standard typesetting measurement unit for width
- ex  standard typesetting measurement unit for height
- in  inches
- mm  millimeters
pct  a percentage. You can also use the ‘%’ symbol.
pt   a printer’s point
px   pixels

Restriction The number of row heights specified must match the number of rows
specified in the LAYOUT statement.

Example This example shows 3 rows with different row heights.
ods layout gridded rows=3 row_heights=(1in 2in 3in);
  ods region;
  proc print data=sashelp.class(obs=1); run;
  ods region;
  proc means data=sashelp.class n mean; run;
  ods region;
  proc print data=sashelp.class(obs=3); run;
  ods layout end;

STYLE=<style-element-name> <[style-attribute-specification(s)]>
specifies the style element to use for the specified locations in the layout.

Tip    Font names that contain characters other than letters or underscores must
       be enclosed in quotation marks.

Example ods layout gridded columns=2 style=[backgroundcolor=yellow];
         proc print data=sashelp.class; run;
         ods layout end;

WIDTH=dimension
specifies the horizontal width of the layout.

dimension
is a nonnegative number followed by an optional unit of measure. It is not
recommended that you use pixels because of adverse dependencies on resolution
that can differ between destinations.

Valid units of measure:

  cm   centimeters
  em   standard typesetting measurement unit for width
  ex   standard typesetting measurement unit for height
  in   inches
  mm   millimeters
  pct  a percentage. You can also use the ‘%’ symbol.
  pt   a printer’s point
  px   pixels

Default If omitted, the width of the layout defaults to the maximum horizontal
space needed to display all of the regions.
Example:  
ods layout gridded width=7in;  
   proc print data=sashelp.class; run  
ods layout end;

\textbf{X=}dimension  
specifies the horizontal starting position of the layout.  

dimension  
is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:  

cm  centimeters  
em  standard typesetting measurement unit for width  
ex  standard typesetting measurement unit for height  
in  inches  
mm  millimeters  
pct  a percentage. You can also use the ‘\%’ symbol.  
pt  a printer’s point  
px  pixels  

Default:  0. If omitted, gridded layout is centered by default.

Example:  
ods layout gridded x=10cm;  
   proc print data=sashelp.class; run  
ods layout end;

\textbf{Y=}dimension  
specifies the vertical starting position of the layout.  

dimension  
is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:  

cm  centimeters  
em  standard typesetting measurement unit for width  
ex  standard typesetting measurement unit for height  
in  inches  
mm  millimeters  
pct  a percentage. You can also use the ‘\%’ symbol.  
pt  a printer’s point  
px  pixels  

Default:  If omitted, the Y argument defaults to the current vertical position on the page.
Example: ods layout gridded y=10cm;
    proc print data=sashelp.class; run
ods layout end;

Details

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

ODS LAYOUT GRIDDED follows the traditional ODS statements 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 GRIDDED statement is used to provide gridded layout.

Note: A gridded layout can contain absolute layouts, and absolute layouts can contain gridded layouts. You are not limited to a single ODS layout type. However, absolute layouts are limited to PRINTER destinations.

Gridded layout is supported by the following destinations:

- HTML
- PRINTER

Reasons to Use Gridded Layout
Gridded layout enables you to arrange output in a two-dimensional gridded structure, such as a spreadsheet or piece of graph paper. Gridded layout is a mechanism for arranging output dynamically. Gridded layout is a simple, powerful tool for arranging output and managing output on one or more pages. Gridded layouts have the following capabilities:

- provide automatic alignment of respective grid cells
- continue the layout onto the next page when necessary
- dynamically compute the size of a grid cell
- makes it easier to maintain the integrity of the report.

Example: Gridded Layout Titles in a Region

Features:
  ODS LAYOUT GRIDDED statement
  ODS REGION statement
  ODS LAYOUT END statement
  ODS PDF statement

Other features:
  FOOTNOTE statement
  TITLE statement
  GCHART procedure
  PRINT procedure
  GOPTIONS
Details

The following example uses the ODS LAYOUT GRIDDED statement and the ODS REGION statement for gridded layout to show how titles and footnotes are generated.

Program

```sas
options nodate nonumber;
ods pdf file='LayoutGriddedTitles.pdf';
title 'This is TITLE1';
footnote 'This is FOOTNOTE1';
ods layout gridded;
ods region;
title 'This is the REGION TITLE';
footnote 'This is the REGION FOOTNOTE';
proc print data=sashelp.class(obs=10);
run;
goptions hsize=4in vsize=4in;
proc gchart data=sashelp.class;
  vbar age / name='gtitle';
  title 'This is the PROCEDURE TITLE';
  footnote 'This is the PROCEDURE FOOTNOTE';
run;
quit;
ods layout end;
ods pdf close;
```

Program Description

- **Set the SAS system options NODATE and NONUMBER.**
  ```sas
  options nodate nonumber;
  ```

- **Open the PDF destination and write to a file.** Write the PDF output to the file LayoutGriddedTitles.pdf.
  ```sas
  ods pdf file='LayoutGriddedTitles.pdf';
  ```

- **Add a global title and footnote.** The title and footnote are added to the Title and Footnote sections of the document.
  ```sas
  title 'This is TITLE1';
  footnote 'This is FOOTNOTE1';
  ```

- **Set the layout to gridded layout.** Using the ODS LAYOUT GRIDDED statement, write titles and footnotes to the region container. This region contains PROC PRINT and PROC GCHART output. Note that the title and footnote are changed and are placed in the layout region.
  ```sas
  ods layout gridded;
  ods region;
title 'This is the REGION TITLE';
footnote 'This is the REGION FOOTNOTE';
proc print data=sashelp.class(obs=10);
run;
```
Change the PROC GCHART procedure title and footnote.

```sas
goptions hsize=4in vsize=4in;
proc gchart data=sashelp.class;
   vbar age / name='gtitle';
   title 'This is the PROCEDURE TITLE';
   footnote 'This is the PROCEDURE FOOTNOTE';
run;
quit;
```

**End the layout.** The ODS LAYOUT END statement ends the layout. The ODS PDF CLOSE statement closes the PDF destination.

```sas
ods layout end;
ods pdf close;
```

The following output shows how tables, titles, footnotes, and charts are placed on a page using gridded layout:
**See Also**

**ODS Statements**
- “ODS REGION Statement, Gridded” on page 479
- “ODS LAYOUT END Statement” on page 484

**ODS REGION Statement, Gridded**

Creates a region container for gridded layouts. A region can contain output (such as text, tables, images, graphs) or nested layout containers. This ODS REGION statement is sandwiched between the ODS
LAYOUT GRIDDLED statement and the ODS LAYOUT END statement. The ODS LAYOUT GRIDDLED statement manages the ODS LAYOUT destination.

Valid in: Anywhere
Category: ODS: Output Control
Requirement: The ODS REGION statement for gridded layout must be used with the ODS LAYOUT GRIDDLED statement.

ODS destination: ODS LAYOUT GRIDDLED is supported for HTML, PRINTER, and SASREPORT destinations.

Tip: Regions can have a fixed size or can be dynamically sized.

Syntax

ODS REGION (<option-1><option-2 ...>);

Summary of Optional Arguments

COLUMN_SPAN=number
Specify the number of grid columns that the region occupies.

COLUMNS=number
Specify the current grid column position in the gridded layout.

HEIGHT=dimension
Specify the vertical height of the region.

ROW_SPAN=number
Specify the number of grid rows that the region occupies.

ROW=number
Specify the current grid row position in the gridded layout

STYLE=<style-element-name> <[style-attribute-specification(s)]>
specifies one or more style elements to use for different parts of the layout.

WIDTH=dimension
Specify the horizontal width of the region.

Optional Arguments

COLUMNS=number
specifies the current grid column position in the gridded layout. The gridded layout automatically tracks the current grid column position and is incremented for every ODS REGION statement.

Restrictions If you have skipped a gridded column, you cannot go back to it. For example, the following code is not allowed.

```r
ods layout gridded rows=1 columns=3;
ods region column=3;
   proc print data=sashelp.class;
   run;

ods region column=1;
   proc print data=sashelp.class;
   run;
ods layout end;
```
Random access of grid rows and columns is not supported.

**Tip**
The COLUMN= option is useful when you want to skip regions in the gridded layout.

**Example**
Note that the PROC MEANS goes to COLUMN=2 instead of to COLUMN=1.

```plaintext
ods layout gridded columns=3;
ods region column=2;
proc means data=sashelp.class n mean;
  var height weight;
run;
ods layout end;
```

**COLUMN_SPAN=number**
specifies the number of grid columns that the region occupies. The COLUMN_SPAN argument enables you to combine adjacent grid columns in gridded layout.

**Default**
1

**Restriction**
The gridded layout option ORDER_TYPE=COLUMN_MAJOR does not support the gridded region option COLUMN_SPAN=. When these options are specified together, the COLUMN_SPAN= option is ignored and a warning message is written to the SAS log.

**Tip**
When the value specified for COLUMN_SPAN= cannot resolve with other options like ROW= or COLUMNS=, the COLUMN_SPAN= option is ignored and a message is written to the SAS log.

**Example**
In this example, the second region spans two columns.

```plaintext
ods layout gridded columns=3;
ods region;
proc print data=sashelp.class;
run;
ods region column_span=2;
proc means data=sashelp.class;
run;
ods layout end;
```

**HEIGHT=dimension**
specifies the vertical height of the region.

**dimension**
is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- `cm` centimeters
- `em` standard typesetting measurement unit for width
- `ex` standard typesetting measurement unit for height
- `in` inches
- `mm` millimeters
pct a percentage. You can also use the ‘%’ symbol.

pt a printer’s point

px pixels

Default If you omit the HEIGHT= option, the default is the maximum vertical space needed to display the output contained in the region.

Restrictions The height is restricted by the dimensions of the layout container.

The sum of all region heights cannot exceed the vertical dimension of the layout.

Example

```sas
ods layout gridded;
ods region height=5in;
   proc print data=sashelp.class; run;
ods layout end;
```

**ROW=number**

specifies the current grid row position in the gridded layout. The gridded layout automatically tracks the current row position and is incremented for every ODS REGION statement.

Restrictions Random access of grid rows and columns is not supported.

Once you have skipped a gridded row, you cannot go back to it. The following is an example of code that is not allowed.

```sas
ods layout gridded rows=3 columns=1;
ods region row=3;
   proc print data=sashelp.class; run;
ods region row=1;
   proc print data=sashelp.class; run;
ods layout end;
```

Tip The ROW= option is useful when you want to skip regions in the gridded layout.

**ROW_SPAN=number**

specifies the number of grid rows that the region occupies. The ROW_SPAN= option enables you to combine adjacent grid rows in gridded layout.

Default 1

Tip When the value specified for ROW_SPAN= cannot resolve with other options like ROW= or COLUMNS=, the ROW_SPAN option is ignored and a message is output to the SAS log.

Example In the following example code, the second region spans two rows.

```sas
ods layout gridded columns=3;
ods region;
   proc print data=sashelp.class(keep=name obs=1); run;
ods region row_span=2;
   proc print data=sashelp.class(keep=name obs=8); run;
ods region;
   proc print data=sashelp.class(keep=age obs=1); run;
```
ods region;
proc print data=sashelp.class(keep=weight obs=1); run;
ods region;
proc print data=sashelp.class(keep=height obs=1); run;
ods layout end;

STYLE=<style-element-name> <[style-attribute-specification(s)]>
specifies the style element to use for the specified locations in the layout.

Tip Font names that contain characters other than letters or underscores must be enclosed in quotation marks.

Example ods layout gridded columns=2 style=[backgroundcolor=yellow];
       ods region style=[backgroundcolor=lightblue];
       proc print data=sashelp.class; run;
       ods layout end;

WIDTH=dimension
specifies the horizontal width of the region.

dimension
is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

  cm  centimeters
  em  standard typesetting measurement unit for width
  ex  standard typesetting measurement unit for height
  in  inches
  mm  millimeters
  pct a percentage. You can also use the ‘%’ symbol.
  pt  a printer’s point
  px  pixels

Default If you omit the WIDTH=option, the default is the maximum horizontal space needed to display the output contained in the region.

Restrictions The width is restricted by the dimensions of the layout container.

The sum of all region widths cannot exceed the horizontal dimension of the layout container.

Example
ods layout gridded;
ods region width=5in;
       proc print data=sashelp.class; run;
ods layout end;

See Also

ODS Statements
ODS LAYOUT END Statement

Ends a gridded or absolute layout statement block.

Valid in: Anywhere
Category: ODS: Output Control
Requirement: The ODS LAYOUT END statement must be used with the ODS LAYOUT GRIDDED or the ODS LAYOUT ABSOLUTE statements.

Example:
ods layout gridded;
  ods region;
    proc print data=sashelp.class;
    run;
  ods layout end;

Syntax

ODS LAYOUT END;

Without Arguments
Use the ODS LAYOUT END statement to end the ODS LAYOUT GRIDDED or the ODS LAYOUT ABSOLUTE statement block.

ODS LISTING Statement

Opens, manages, or closes the LISTING destination.

Valid in: Anywhere
Category: ODS: SAS Formatted
Default: The default style for the LISTING destination is Listing.
Note: The ODS LISTING statement supports Scalable Vector Graphics. Scalable Vector Graphics (SVG) is an XML language for describing two-dimensional vector graphics. For information about scalable vector graphics, see “Using SVG Graphics” in SAS/GRAPH: Reference.

Syntax

ODS LISTING <action> ;

Without Arguments
If you use the ODS LISTING statement without an action or options, it opens the LISTING destination.
**Actions**
The following actions are available for the ODS LISTING statement:

**CLOSE**
closes the LISTING destination and any files that are associated with it.

**Tip** When you close an ODS destination, ODS does not send output to that destination. Closing an unneeded destination frees some system resources.

**EXCLUDE exclusion(s)| ALL | NONE**
excludes one or more output objects from the LISTING destination.

**Default** NONE

**Restriction** The LISTING destination must be open for this action to take effect.

**See** “ODS EXCLUDE Statement” on page 321

**SELECT selection(s)| ALL | NONE**
selects output objects for the LISTING destination.

**Default** ALL

**Restriction** The LISTING destination must be open for this action to take effect.

**See** “ODS SELECT Statement ” on page 758

**SHOW**
writes the current selection or exclusion list for the LISTING destination to the SAS log.

**Restriction** The LISTING destination must be open for this action to take effect.

**Tip** If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list.

**See** “ODS SHOW Statement” on page 771

**Optional Arguments**

**DATAPANEL=number | DATA | PAGE**
suggests how to split a table that is too wide to fit on a single page into sections of columns and rows. Each section of columns and rows is a data panel. Each data panel has column headings at the top.

**Note:** In this context, a page is what the procedure uses as a page in creating the LISTING output. The SAS system options LINESIZE= and PAGESIZE= generally determine the page size, although some procedures (PROC REPORT, for example) can temporarily override the values that the system options specify.

**number**
writes the specified number of observations in a panel, if possible. More than one panel can occur on every page if space permits.

**Range** 1 to the largest integer that the operating system supports
DATA
bases the size of the panel on how the table is stored in memory. This value provides the fastest performance. However, if the table contains many columns, the number of rows in each panel might be small.

PAGE
tries to make panels that match the page size. If the table contains more columns than can fit on a page, the first page is filled with as many observations as possible for as many columns as can fit on a single line. The second page contains the same observations for the next group of columns, and so on, until all rows and columns have been printed.

This arrangement minimizes the amount of space that is used for column headings because most pages contain observations for only one set of columns.

Restriction If the page size is greater than 200, ODS uses DATAPANEL=200.

Default PAGE

DEVICE= device-driver
specifies the name of a device driver. ODS automatically selects an optimal default device for each open output destination.

The following table lists default devices for the most common ODS output destinations.

<table>
<thead>
<tr>
<th>Output Destination</th>
<th>Default Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPUB</td>
<td>PNG</td>
</tr>
<tr>
<td>HTML</td>
<td>PNG</td>
</tr>
<tr>
<td>HTML5</td>
<td>SVG</td>
</tr>
<tr>
<td>LISTING</td>
<td>PNG</td>
</tr>
<tr>
<td>Measured RTF</td>
<td>EMF</td>
</tr>
<tr>
<td>RTF</td>
<td>EMF</td>
</tr>
<tr>
<td>Markup Tagsets</td>
<td>PNG</td>
</tr>
</tbody>
</table>

Tip Specifying a device on the ODS DEVICE= option takes precedence over the SAS global option and the graphics option.

See DEVICE= System Option in SAS System Options: Reference

FILE= file-specification
specifies the file to write to. file-specification is one of the following:

'external-file'
is the name of an external file to which to write.
**fileref**

is a file reference that has been assigned to an external file. Use the **FILENAME statement** to assign a fileref. For information about the FILENAME statement, see *SAS Statements: Reference*.

**Default** If you do not specify a file to write to, ODS writes the output to the LISTING window.

**GPATH= file-specification <(url='Uniform-Resource-Locator' | NONE)>**

specifies the location for all graphics output that is generated while the destination is open.

**file-specification**

specifies the file or SAS catalog to which to write. ODS names automatically each output object that it places in the file. If you specify an invalid filename, the ActiveX and Java devices send output to the default filename. Other devices create the file as a directory and write output to that directory using the default filename. For more information about how ODS names catalog entries and external files, see *SAS/GRAPH: Reference*. **file-specification** is one of the following:

**external-file**

is the name of an external file to write to.

**Requirement** You must enclose **external-file** in quotation marks.

**fileref**

is a file reference that has been assigned to an external file. Use the **FILENAME statement** to assign a fileref. For information about the FILENAME statement, see *SAS Statements: Reference*.

**Interaction** If you specify a fileref in the GPATH= option, ODS does not use information from the GPATH= option when it constructs links.

**libref.catalog**

specifies a SAS catalog to which to write.

**URL= 'Uniform-Resource-Locator' | NONE**

specifies a URL for **file-specification**.

**Uniform-Resource-Locator**

is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

**Requirement** You must enclose **Uniform-Resource-Locator** in quotation marks.

**NONE**

specifies that no information from the GPATH= option appears in the links or references.

**Tip** This option is useful for building output files that can be moved from one location to another. The links from the contents and page files resolve if they are constructed with a simple URL (one name) and the contents, page, and body files are all in the same location.
IMAGE_DPI=
    specifies the image resolution of ODS graphics output. Output from device-based
    graphics is not affected.

    | Alias  | DPI= |
    |--------|------|
    | Default| 96   |

    | Restriction: The IMAGE_DPI= option affects template-based graphics only.

PACKAGE <package-name>
    specifies that the output from the destination be added to a package.

    package-name
    specifies the name of a package that was created with the ODS PACKAGE
    statement. If no name is specified, then the output is added to the unnamed
    package that was opened last.

    | See also: “ODS PACKAGE Statement” on page 554

SGE= ON | OFF
    determines whether you can edit ODS graphics output with the ODS Graphics
    Editor.

    | Default | OFF |

    | Restriction: The SGE= option affects template-based graphics only.


STYLE=style-definition
    specifies the style template to use in writing the listing output.

    | Default | The default style for the LISTING destination is Listing. |

    | Restriction: The STYLE= option can be used with ODS graphics only.

Details

Beginning with SAS 9.3, by default, in the Windowing environment with the Windows
and UNIX operating systems, the LISTING destination is closed and the HTML
destination is open. You do not have to submit an ODS HTML statement to generate
HTML output, and you do not have to use the ODS HTML CLOSE statement to be able
to view your output. However, to create LISTING output, you must either submit the
ODS LISTING statement or enable the LISTING destination by other means. For more
details, see “Working with Output Defaults” on page 21.

The HTML destination now supports Scalable Vector Graphics (SVG). For information
about scalable vector graphics, see “Using SVG Graphics” in SAS/GRAPH: Reference.

ODS MARKUP Statement

Opens, manages, or closes the MARKUP destination, which produces SAS output that is formatted using
one of many different markup languages.

| Valid in: | Anywhere |
Category: ODS: Third-Party Formatted

Default: The default style for Markup family destinations is HTMLBlue.

Interactions: The output type is determined by the TAGSET | TYPE= option, which specifies the type of markup language that is applied to the output.

By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC PLOT or PROC CHART), ODS formats the output in SAS Monospace font. If you are creating output that will be viewed in an operating environment where SAS software is not installed, this output will not be displayed correctly. This is because without SAS, the SAS Monospace font is not recognized. To make your document display correctly, include the following statement before your SAS program:

```
OPTIONS FORMCHAR="|----|+|---+=|-/<>*";
```

Syntax

```
ODS MARKUP (<ID>= identifier) < action > ;
ODS MARKUP (<ID>= identifier) < option(s) > < TAGSET=tagset-name > < action > ;
```

Summary of Optional Arguments

- **(ID= identifier)**
  - Open multiple instances of the same destination at the same time

- **ANCHOR= 'anchor-name'**
  - Specify a unique base name for the anchor tag that identifies each output object in the current body file

- **ARCHIVE='string'**
  - Specify which applet to use to view ODS HTML output

- **ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)**
  - Specify attributes to write between the tags that generate dynamic graphics output

- **BASE= 'base-text'**
  - Specify text to use as the first part of all links and references that ODS creates in output files

- **BODY= 'file-specification' (suboption(s))**
  - Open a markup family destination and specify the file that contains the primary output that is created by the ODS statement

- **CHARSET= character-set**
  - Specify the character set to be generated in the META declaration for the HTML output

- **CLOSE**
  - Close the destination and the file that is associated with it

- **CODE='file-specification' <(suboption(s))>**
  - Open the HTML destination and specify the file that contains relevant style information

- **CODEBASE='string'**
  - Create a file path that can be used by the GOPTIONS devices

- **CONTENTS= 'file-specification' <(suboption(s))>**
  - Open the HTML destination and specify the file that contains a table of contents for the output

- **CSSSTYLE= 'file-specification'<(media-type-1<…media-type-10>)>**
Specify a cascading style sheet to apply to your output

```
DEVICE= device-driver
```

Specify a device for the output destination

```
DOM="external-file">
```

Specify that the ODS document object model is written to the SAS log or to an external file.

```
ENCODING= local-character-set-encoding
```

Override the encoding for input or output processing (transcodes) of external files

```
EVENT= event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )
```

Specify an event and the value for event variables that is associated with the event

```
EXCLUDE exclusion(s) | ALL | NONE
```

Exclude output objects from the destination

```
FRAME= 'file-specification' <(suboption(s))>
```

Specify the file that integrates the table of contents, the page contents, and the body file

```
GFOOTNOTE | NOGFOOTNOTE
```

Control the location where footnotes are printed in the graphics output

```
GPATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)
```

Specify the location for all graphics output that is generated while the destination is open

```
GTITLE | NOGTITLE
```

Control the location where titles are printed in the graphics output

```
HEADTEXT= 'markup-document-head'
```

Specify HTML tags to place between the `<HEAD>` and `</HEAD>` tags in all of the output files.

```
IMAGE_DPI=
```

Specify the image resolution for graphical output

```
METATEXT= 'metatext-for-document-head'
```

Specify HTML code to use as the `<META>` tag between the `<HEAD>` and `</HEAD>` tags in all of the HTML output files.

```
NEWFILE= starting-point
```

Create a new body file at the specified starting point

```
OPTIONS ( DOC= | <(suboption(s))> )
```

Specify tagset-specific suboptions and a named value

```
PACKAGE <package-name>
```

Specify that the output from the destination be added to an ODS package

```
PAGE= 'file-specification' <(suboption(s))>
```

Open the HTML destination and specify the file that contains a description of each page of the body file and contains links to the body file

```
PARAMETERS= (parameter-pair-1 ... parameter-pair-n)
```

Write the specified parameters between the tags that generate dynamic graphics output

```
PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)
```

Specify the location of an aggregate storage location or a SAS catalog for all markup files

```
RECORD_SEPARATOR= 'alternative-separator' | NONE
``
Specify an alternative character or string to separate lines in the output files.

**SELECT selection(s) | ALL | NONE**
Select output objects for the destination.

**SHOW**
Write to the SAS log the current selection or exclusion list for the destination.

**STYLE= style-template**
Specify a style template to use in writing output files.

**STYLESHEET= 'file-specification' <(suboption(s))>**
Open the HTML destination and place style information for output into an external file, or read style sheet information from an existing file.

**TAGSET= tagset-name**
Specify a keyword value for a tagset. A tagset is a template that defines how to create a markup language output type from a SAS format.

**TEXT= text-string**
Insert text into your document.

**TRANTAB= 'translation-table’**
Specify a translation table to use when transcoding a file for output.

**Actions**

**CLOSE**
closes the destination and any files that are associated with it.

**Tip** When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

**EXCLUDE exclusion(s) | ALL | NONE**
excludes one or more output objects from the destination.

Default: NONE

Restriction: A destination must be open for this action to take effect.

See “ODS EXCLUDE Statement” on page 321

**SELECT selection(s) | ALL | NONE**
selects output objects for the specified destination.

Default: ALL

Restriction: A destination must be open for this action to take effect.

See “ODS SELECT Statement” on page 758

**SHOW**
writes the current selection list or exclusion list for the destination to the SAS log.

Restriction: The destination must be open for this action to take effect.

Tip: If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

See “ODS SHOW Statement” on page 771
### Optional Arguments

**ANCHOR= 'anchor-name'

specifies a unique base name for the anchor tag that identifies each output object in the current body file.

Each output object has an anchor tag for the contents, page, and frame files to reference. The links and references are automatically created by ODS. The links and references point to the name of an anchor. Therefore, each anchor name in a file must be unique.

**anchor-name**

is the base name for the anchor tag that identifies each output object in the current body file.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR= 'TABULATE', then ODS names the first anchor tabulate. The second anchor is named tabulate1; the third is named tabulate2, and so on.

<table>
<thead>
<tr>
<th>Restrictions</th>
<th>Each anchor name in a file must be unique.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Only alphanumeric values, the special characters &quot;$ - . + ! * ' ( ) , &quot; and reserved characters used for their reserved purposes can be used unencoded within a URL.</td>
</tr>
<tr>
<td>Requirement</td>
<td>You must enclose anchor-name in quotation marks.</td>
</tr>
<tr>
<td>Interaction</td>
<td>If you open a file to append to it, be sure to specify a new anchor name to prevent writing the same anchors to the file again. ODS does not recognize anchors that are already in a file when it opens the file.</td>
</tr>
<tr>
<td>Tips</td>
<td>You can change anchor names as often as you want by specifying the ANCHOR= option in a markup family statement anywhere in your program. After you have specified an anchor name, it remains in effect until you specify a new one.</td>
</tr>
<tr>
<td></td>
<td>Specifying new anchor names at various points in your program is useful when you want other web pages to link to specific parts of your markup language output. Because you can control where the anchor name changes, you know in advance what the anchor name is at those points.</td>
</tr>
<tr>
<td></td>
<td>An anchor-name must begin with a letter ([A-Za-z]) and can be followed by any number of letters, digits ([0-9]), hyphens (-), underscores (_), colons (:), and periods (.).</td>
</tr>
</tbody>
</table>

**ARCHIVE='string'

specifies which applet to use to view the ODS HTML output. The ARCHIVE= option is valid only for the GOPTIONS Java device.

The string must be one that the browser can interpret. For example, if the archive file is local to the computer that you are running SAS on, you can use the FILE protocol to identify the file. If you want to point to an archive file that is on a web server, use the HTTP protocol.

| Default | If you do not specify ARCHIVE= and you are using the JAVA device driver, ODS uses the value of the SAS system option |
APPLETOC=. There is no default if you are using the ACTIVEX device driver.

Requirements

You must enclose string in quotation marks.

The ARCHIVE attribute is a feature of Java 1.1. Therefore, if you are using the Java device driver, your browser must support this version of Java. Both Internet Explorer 4.01 and Netscape 4.05 support Java 1.1.

Interaction

Use ARCHIVE= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA or DEVICE=ACTIVEX option in the GOPTIONS statement.

Tips

Typically, this option should not be used, because the SAS server automatically determines the correct SAS/GRAPH applets to view the ODS HTML output. However, if you have renamed the JAR files, or have other applets with which to view the ODS HTML output, this option enables you to access these applets.

Use the CODEBASE= option to specify the file path. It is recommended that you do not put a file path in your ARCHIVE= option.

The value of APPLETOC= points to the location of the Java archive files that are shipped with the SAS system. To find out what the value of this option is, you can either look in the Options window in the Files folder under Environment Control, or you can submit the following procedure step:

```
proc options option=appletloc;
run;
```

ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)

writes the specified attributes between the tags that generate dynamic graphics output.

attribute-pair

specifies the name and value of each attribute. attribute-pair has the following form:

'attribute-name'='attribute-value'

attribute-name

is the name of the attribute.

attribute-value

is the value of the attribute.

Requirement

You must enclose attribute-name and attribute-value in quotation marks.

Interaction

Use the ATTRIBUTES= option in conjunction with SAS/GRAPH procedures and with the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

See

SAS/GRAPH: Reference for valid attributes for the Graph Applet, the Map Applet, the Contour Applet, and the MetaView Applet.
BASE= 'base-text'
specifies the text to use as the first part of all links and references that ODS creates in
the output files.

base-text
is the text that ODS uses as the first part of all links and references that ODS
creates in the file.

Consider this specification:

BASE= 'http://www.your-company.com/local-url/

In this case, ODS creates links that begin with the string http://www.your-
company.com/local-url/. The appropriate anchor-name completes the link.

Requirement You must enclose base-text in quotation marks.

BODY= 'file-specification' (suboption(s))
opens a markup family destination and specifies the file that contains the primary
output that is created by the ODS statement. These files remain open until you do
one of the following:

• close the destination with either an ODS markup-family-destination CLOSE
  statement or ODS _ALL_ CLOSE statement.

• open the same destination with a second markup family statement. This closes
  the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the
FILENAME statement to assign a fileref.

Restriction The BODY=fileref option cannot be used in conjunction with
the NEWFILE= option.

See For information about the FILENAME statement, see

entry.markup
specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library
and catalog. See the discussion of the PATH= option.

(suboption(s))
specifies one or more suboptions in parentheses. Suboptions are instructions for
writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a
file.
See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 515.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 515.

(NO_TOP_MATTER)
See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 516.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 516.

(URL= 'Uniform-Resource-Locator')
See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 516.

Alias FILE=

Interaction Using the BODY= option in an ODS markup family statement that refers to an open ODS markup destination forces ODS to close the destination and all associated files. ODS then opens a new instance of the destination. For more information see “Opening and Closing the MARKUP Destination “ on page 517.

Note For some values of TAGSET=, this output is an HTML file. For other TAGSET= values, the output is an XML file, and so on.

CHARSET= character-set
specifies the character set to be generated in the META declaration for the HTML output.

See For information about the CHARSET= option, see “CHARSET= Option” in SAS National Language Support (NLS): Reference Guide.

CODE= 'file-specification' <(suboption(s))>
opens a markup family destination and specifies the file that contains accompanying programming code, such as JavaScript or XSL (Extensible Stylesheet Language). These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.

• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.
file-specification is one of the following:

external-file
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.


entry.markup
specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 515.

(URL= 'Uniform-Resource-Locator' )
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator' )” on page 516.

CODEBASE='string'
specifies the location of the executable Java applet or the ActiveX control file. string is specified as a pathname or as a URL. The CODEBASE file path option has two definitions, depending on the GOPTIONS device used.

When you generate web presentations with the JAVA and ActiveX device drivers, SAS generates HTML pages that automatically look for the JAVA archive files or the ActiveX control file in the default installation location.

For the ActiveX device:
If you use the ActiveX device driver with ODS to generate output containing an ActiveX control, then specify the CODEBASE= option in the ODS statement. The value of the CODEBASE= option should include the location and the version of the EXE file.

Tip You do not need to specify the CODEBASE= option with the DEVICE=ACTIVEX option unless the users that view your output do not have the ActiveX control installed on their machine. When users that do not have the ActiveX control installed view your output, they are prompted to download the control.
See *SAS/GRAPH: Reference* for information about specifying the location of control and applet files using the CODEBASE= and ARCHIVE= options.

For the Java device:

If you use the Java device driver with ODS to generate output containing a SAS/GRAPH applet, specify the path to the JAR file with the CODEBASE= option in the ODS statement.

When you specify DEVICE=JAVA, the users that view your output must have access to the appropriate Java applet. By default, SAS sets the value of CODEBASE= to refer to the executable file for the applet that is automatically installed with SAS. The default location of the SAS Java archive files is specified by the APPLETLOC= system option. You do not need to specify the CODEBASE= option if both of the following conditions are true.

- The default location is accessible by users who are viewing your web presentation.
- The SAS Java archive is installed at that location.

**Tip** Specify only the directory of the JAR file. The CODEBASE= location can be specified as a pathname or as a URL.

See *SAS/GRAPH: Reference* for information about specifying the location of control and applet files using the CODEBASE= and ARCHIVE= options.

**CONTENTS= 'file-specification' <(suboption(s))>**

opens a markup family destination and specifies the file that contains a table of contents for the output. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

**file-specification**

specifies the file, fileref, or SAS catalog to write to.

**file-specification** is one of the following:

**external-file**

is the name of an external output file.

**Requirement** You must enclose *external-file* in quotation marks.

**fileref**

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in *SAS Statements: Reference*.

**entry.markup**

specifies an entry in a SAS catalog to write to.

**Interaction** If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.
suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 515.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 515.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 516.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

<title-text>
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text’)” on page 516.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator’)” on page 516.

CSSSTYLE='file-specification'\<\(media-type-1<…media-type-10>\)> specifies a cascading style sheet to apply to your output.

<file-specification>
specifies a file, fileref, or URL that contains CSS code.

<file-specification> is one of the following:

"external-file"
is the name of the external file.

Requirement You must enclose external-file in quotation marks.

<fileref>
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.
For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

"URL"

is a URL to an external file.

Requirement You must enclose URL in quotation marks.

(media-type-1<.. media-type-10>)

specifies one or more media blocks that correspond to the type of media that your output is rendered on. CSS uses media type blocks to specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not contained in any media blocks. By using the media-type suboption, in addition to the general CSS code, you can import the section of a CSS file intended only for a specific media type.

Default If no media-type is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

Range You can specify up to ten different media types.

Requirements You must enclose media-type in parentheses.

Tip If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

Restriction The CSSSTYLE= option does not affect SAS/GRAFPH output.

Requirement CSS files must be written in the same type of CSS produced by the ODS HTML statement. To view the CSS code that ODS creates, you can do one of the following:

• specify the ODS TRACE DOM statement
• specify the DOM option

Interaction If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

See For complete documentation about ODS and Cascading Style Sheets, see SAS Output Delivery System: Advanced Topics

Example “Example 6: Applying a CSS File to ODS Output” on page 527

DEVICE= device-driver

specifies the name of a device driver. ODS automatically selects an optimal default device for each open output destination.
The following table lists the default devices for the most common ODS output destinations. These default devices are used when graphics are created using SAS/GRAPH or ODS Graphics. In the third maintenance release of SAS 9.4, EPUB3 is the default EPUB destination. EPUB2 was the default EPUB version in prior releases of SAS 9.4. This default is set in the Registry. For a complete list of supported devices and file types, see “Supported File Types for Output Destinations” on page 349.

**Table 6.14 Default Devices for ODS Output Destinations**

<table>
<thead>
<tr>
<th>Output Destination</th>
<th>Default Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPUB2</td>
<td>PNG</td>
</tr>
<tr>
<td>EPUB3 (EPUB)</td>
<td>SVG</td>
</tr>
<tr>
<td>HTML</td>
<td>PNG</td>
</tr>
<tr>
<td>HTML5</td>
<td>SVG</td>
</tr>
<tr>
<td>LISTING</td>
<td>PNG</td>
</tr>
<tr>
<td>Measured RTF</td>
<td>EMF</td>
</tr>
<tr>
<td>RTF</td>
<td>EMF</td>
</tr>
<tr>
<td>Markup Tagsets</td>
<td>PNG</td>
</tr>
</tbody>
</table>

**Tips**
Specifying a device on the ODS DEVICE= option takes precedence over the SAS global option and the graphics option.

For information about using the SAS Registry to change defaults, see “Changing SAS Registry Settings for ODS” on page 40.

**DOM<="external-file"**
specifies that the ODS document object model is written to the SAS log or an external file.

*external-file*
is the name of an external output file.

**Requirement**
You must enclose *external-file* in quotation marks.

**See**
For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in *SAS Output Delivery System: Advanced Topics*.

**ENCODING= local-character-set-encoding**
overrides the encoding for input or output processing (transcodes) of external files.
EVENT= event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )
specifies an event and the value for event variables that are associated with the event.

(FILE= BODY | CODE | CONTENTS | DATA | FRAME | PAGES | STYLESHEET);
    triggers one of the known types of output files that correspond to the BODY=, CODE=, CONTENTS=, FRAME=, PAGES=, and STYLESHEET= options.

(FINISH)
    triggers the finish section of an event.

See For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(LABEL=’variable-value’)
specifies the value for the LABEL event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the LABEL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(NAME=’variable-value’)
specifies the value for the NAME event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the NAME event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(START)
    triggers the start section of an event.

See For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(STYLE= style-element)
specifies a style element.

See For information about style elements, see “Style Attributes Overview” in SAS Output Delivery System: Procedures Guide.

(TARGET=’variable-value’)
specifies the value for the TARGET event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the TARGET event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(TEXT=’variable-value’)
specifies the value for the TEXT event variable.

Requirement variable-value must be enclosed in quotation marks.
See  For information about the TEXT event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(URL='variable-value')
specifies the value for the URL event variable.

Requirement  variable-value must be enclosed in quotation marks.

See  For information about the URL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

Default  (FILE='BODY')

Requirement  The EVENT= option's suboptions must be enclosed in parentheses.

FRAME= ‘file-specification’ <(suboption(s))>
opens a markup family destination and, for HTML output, specifies the file that integrates the table of contents, the page contents, and the body file. If you open the frame file, then you see a table of contents, a table of pages, or both, as well as the body file. For XML output, FRAME= specifies the file that contains the DTD. These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.

• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement  You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See  For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

entry.markup
specifies an entry in a SAS catalog to write to.

Interaction  If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
ensures you to send output directly to a web server instead of writing it to a file.
(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 515.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 516.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

\texttt{title-text}
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 516.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator’)” on page 516.

Restriction
If you specify the FRAME= option, then you must also specify the CONTENTS= option, the PAGE= option, or both.

Example “Example 2: Creating an XML File and a DTD” on page 520

\textbf{GFOOTNOTE | NOGFOOTNOTE}
controls the location where footnotes are printed in the graphics output.

\textbf{GFOOTNOTE}
writes footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

\textbf{NOGFOOTNOTE}
writes footnotes that are created by ODS, which appear outside the graph borders.

Default GFOOTNOTE

Restrictions Footnotes that are displayed by a markup language statement support all SAS/GRAPH FOOTNOTE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text
angle specifications, are ignored. For details about the SAS/GRAPH FOOTNOTE statement, see “FOOTNOTE Statement” in SAS/GRAPH: Reference.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

**GPATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)**

specifies the location for all graphics output that is generated while the destination is open. Use this option when you want to write graphics output files to a location different that specified by the PATH= option for markup files. If you specify an invalid filename, the ActiveX and Java devices send output to the default filename. Other devices create the file as a directory and write output to that directory using the default filename. For information about how ODS names catalog entries and external files, see SAS/GRAPH: Reference.

'aggregate-file-storage-location'

specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement You must enclose aggregate-file-storage-location in quotation marks.

*fileref*

is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction If you specify a fileref in the GPATH= option, then ODS does not use information from the GPATH= option when it constructs links.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

*libref.catalog*

specifies a SAS catalog to write to.

**URL= 'Uniform-Resource-Locator' | NONE**

specifies a URL for file-specification.

*Uniform-Resource-Locator*

is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

Requirement You must enclose Uniform-Resource-Locator in quotation marks.

*NONE*

specifies that no information from the GPATH= option appears in the links or references.

*Tip* This option is useful for building output files that can be moved from one location to another. If the links from the contents and page files are constructed with a simple URL (one name), then they resolve, as long as the contents, page, and body files are all in the same location.
If you omit the GPATH= option, then ODS stores graphics in the location that is specified by the PATH= option. If you do not specify the PATH= option, then ODS stores the graphics in the current directory. For more information, see the PATH= option.

**GTITLE | NOGTITLE**

controls the location where titles are printed in the graphics output.

**GTITLE**

writes the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.

**NOGTITLE**

writes the title that is created by ODS, which appears outside of the graph borders.

Default GTITLE

Restrictions

Titles that are displayed by any markup language statement support most SAS/GRAPH TITLE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH TITLE statement, see TITLE statement.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

**HEADTEXT= 'markup-document-head'**

specifies markup tags to place between the <HEAD> and </HEAD> tags in all of the output files.

**markup-document-head**

specifies the markup tags to place between the <HEAD> and </HEAD> tags.

Restriction

HEADTEXT= cannot exceed 256 characters.

Requirement

You must enclose markup-document-head in quotation marks.

Tips

ODS cannot parse the markup that you supply. It should be well-formed markup that is correct in the context of the <HEAD> and </HEAD> tags.

Use the HEADTEXT= option to define programs (such as JavaScript) that you can use later in the file.

**(ID= identifier)**

enables you to run multiple instances of the same destination at the same time. Each instance can have different options.

**identifier**

specifies another instance of the destination that is already open. identifier is numeric or a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numeric characters.

Restriction

If identifier is numeric, it must be a positive integer.
Requirement: You must specify the ID= option immediately after the destination name.

Tip: You can omit the ID= option and instead use a name or a number to identify the instance.

Example: “Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594

**IMAGE_DPI=**

specifies the image resolution for graphical output.

Alias: DPI=

Default: 96

**CAUTION**

Caution: When using high DPI= or DPI_IMAGE values (values over 1000), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500M or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.

**METATEXT= 'metatext-for-document-head'**

specifies HTML code to use as the <META> tag between the <HEAD> and </HEAD> tags of all of the HTML output files.

'`metatext-for-document-head`'

specifies the HTML code that provides the browser with information about the document that it is loading. For example, this attribute could specify the content type and the character set to use.

Requirement: You must enclose `metatext-for-document-head` in quotation marks.

Default: If you do not specify METATEXT=, then ODS writes a simple <META> tag, which includes the content-type of the document and the character set to use, to all the HTML files that it creates.

Restriction: METATEXT= cannot exceed 256 characters.

Tip: ODS cannot parse the HTML code that you supply. It should be well-formed HTML code that is correct in the context of the <HEAD> tags. If you are using METATEXT= as it is intended, then your META tag should look like this:

<`META your-metatext-is-here`>

**NEWFILE= `starting-point`**

creates a new body file at the specified `starting-point`.

`starting-point`

is the location in the output where you want to create a new body file.

ODS automatically names new files by incrementing the name of the body file. In the following example, ODS names the first body file REPORT.XML. Additional body files are named REPORT1.XML, REPORT2.XML, and so on.

Example:

BODY= 'REPORT.XML'
**starting-point** is one of the following:

**BYGROUP**
starts a new file for the results of each BY group.

**NONE**
writes all output to the body file that is currently open.

**OUTPUT**
starts a new body file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

*Alias*  **TABLE**

**PAGE**
starts a new body file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.

**PROC**
starts a new body file each time you start a new procedure.

**Default**  **NONE**

**Restriction**
The NEWFILE= option cannot be used in conjunction with the BODY=fileref option.

**Tips**
If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first body file MAY5.XML. Additional body files are named MAY6.XML, MAY7.XML, and so on.

**Example:**

```plaintext
BODY='MAY5.XML'
```

**OPTIONS ( DOC= | <suboption(s)> )**
specifies tagset-specific suboptions and a named value.

**DOC= 'HELP' | 'QUICK' | 'SETTINGS' | 'CHANGELOG'**
provides information about the specified tagset.

**HELP**
provides generic help and information with a quick reference.

**QUICK**
describes the options available for this tagset.

**SETTINGS**
provides the current option settings.

**CHANGELOG**
lists a history of changes made to the tagset. This suboption is supported only on the RTF tagset.

**Requirement**
All values must be enclosed in quotation marks.

**suboption(s)**
specifies one or more suboptions that are valid for the specified tagset.

Suboptions have the following format:

```plaintext
keyword='value'
```
Specify one of the following options when opening an ODS tagset statement, or at any time after the destination has been opened, to get information about suboptions for the tagset.

- `options(doc='help');`
- `options(doc='quick');`
- `options(doc='settings');`

Requirement: `suboption(s)` must be enclosed in parentheses.

Example: “Example: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information” on page 802

**PACKAGE <package-name>**

specifies that the output from the destination be added to a package.

*package-name*

specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

See: “ODS PACKAGE Statement” on page 554

Example: “Example 1: Creating an ODS Package” on page 558

**PAGE= 'file-specification' <(suboption(s))>**

opens a markup family destination and specifies the file that contains a description of each page of the body file, and contains links to the body file. ODS produces a new page of output whenever a procedure requests a new page. These files remain open until you do one of the following:

- close the destination with either an ODS `markup-family-destination` CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

*file-specification*

specifies the file, fileref, or SAS catalog to write to.

`file-specification` is one of the following:

- `external-file`
  
  is the name of an external output file.

  Requirement: You must enclose `external-file` in quotation marks.

- `fileref`
  
  is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

  See: For information about the FILENAME statement, see “FILENAME Statement” in *SAS Statements: Reference*.

- `entry.markup`

  specifies an entry in a SAS catalog to write to.

  Interaction: If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.
suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 515.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 515.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 516.

(TITLE=’title-text’)
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

title-text
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE=’title-text’)” on page 516.

(URL= ’Uniform-Resource-Locator’)
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= ’Uniform-Resource-Locator’)” on page 516.

Interaction The SAS system option PAGESIZE= has no effect on pages in HTML output except when you are creating batch output. For information about the PAGESIZE= option, see “PAGESIZE= System Option” in SAS System Options: Reference.

PARAMETERS= (parameter-pair-1 ... parameter-pair-n)
writes the specified parameters between the tags that generate dynamic graphics output.

parameter-pair
specifies the name and value of each parameter. parameter-pair has the following form:

'parameter-name'= 'parameter-value'
**Parameter**

- **Parameter-name**
  - is the name of the parameter.
  - Requirement: You must enclose *parameter-name* and *parameter-value* in quotation marks.

- **Parameter-value**
  - is the value of the parameter.

**Interaction**

- Use PARAMETERS= in conjunction with SAS/GRAF procedures and the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

**See**

- *SAS/GRAF: Reference* for valid parameters for the Graph Applet, Map Applet, Contour Applet, and the MetaView Applet.

**PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL='Uniform-Resource-Locator' | NONE)**

- Specifies the location of an aggregate storage location or a SAS catalog for all markup files. If the GPATH= option is not specified, all graphics output files are written to the "aggregate-file-storage-specification" or libref.

- **'aggregate-file-storage-location'**
  - specifies an aggregate storage location such as directory, folder, or partitioned data set.
  - Requirement: You must enclose *aggregate-file-storage-location* in quotation marks.

- **fileref**
  - is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.
  - Interaction: If you use a fileref in the PATH= option, then ODS does not use information from PATH= when it constructs links.
  - See: For information about the FILENAME statement, see “FILENAME Statement” in *SAS Statements: Reference*.

- **libref.catalog**
  - specifies a SAS catalog to write to.
  - See: For information about the LIBNAME statement, see “LIBNAME Statement” in *SAS Statements: Reference*.

- **URL= 'Uniform-Resource-Locator' | NONE**
  - specifies a URL for the file-specification.

  - **Uniform-Resource-Locator**
    - is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.
  
  - **NONE**
    - specifies that no information from the PATH= option appears in the links or references.

  - **Tip**: This option is useful for building output files that can be moved from one location to another. The links from the contents and page files must be
constructed with a single-name URL, and the contents, page, and body files must be in the same location.

Interaction

If you use the BODY= or FILE= external file option in conjunction with the PATH= option, the external file specification should not include path information.

**RECORD_SEPARATOR= 'alternative-separator' | NONE**

specifies an alternative character or string that separates lines in the output files.

Different operating environments use different separator characters. If you do not specify a record separator, then the files are formatted for the environment where you run the SAS program. However, if you are generating files for viewing in a different operating environment that uses a different separator character, then you can specify a record separator that is appropriate for the target environment.

*alternative-separator*

represents one or more characters in hexadecimal or ASCII format. For example, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

```
RECORD_SEPARATOR= '0D0A'x
```

**Operating Environment Information**

In a mainframe environment, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

```
RECORD_SEPARATOR= '0D25'x
```

**Requirement**

You must enclose *alternative-separator* in quotation marks.

**NONE**

produces the markup language that is appropriate for the environment where you run the SAS job.

**Windows Specifics**

In a mainframe environment, by default, ODS produces a binary file that contains embedded record separator characters. This binary file is not restricted by the line-length restrictions on ASCII files. However, if you view the binary files in a text editor, then the lines run together. If you want to format the files so that you can read them with a text editor, then use RECORD_SEPARATOR= NONE. In this case, ODS writes one line of markup language at a time to the file. When you use a value of NONE, the logical record length of the file that you are writing to must be at least as long as the longest line that ODS produces. If the logical record length of the file is not long enough, then the markup language might wrap to another line at an inappropriate place.

**Aliases**

RECSEP=

RS=

**STYLE= style-template**

specifies the style template to use in writing the output files.

*style-template*

describes how to display the presentation aspects (color, font face, font size, and so on) of your SAS output. A style template determines the overall appearance of the documents that use it. Each style template consists of style elements.
The STYLE= option is not valid when you are creating XML output.

Note
If you are using SAS Studio, you do not need to specify the STYLE= option. You can go to Preferences ⇒ Results and change the style from the drop-down list for your selected destination.

See
For a complete discussion of style templates, see “TEMPLATE Procedure: Creating a Style Template” in SAS Output Delivery System: Procedures Guide.

If you do not specify a style template, then ODS uses the file that is specified in the SAS registry subkey ODS ⇒ DESTINATIONS ⇒ MARKUP. By default, this value specifies Default.

If you specify the STYLE= option in an ODS HTML4 statement and subsequently need PROC PRINT output to use new style templates in another ODS HTML4 statement, close the first statement before specifying the second statement.

**STYLESHEET= 'file-specification' <(suboption(s))>**

opens a markup family destination and places the style information for markup output into an external file, or reads style sheet information from an existing file. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

**file-specification**
specifies the file, fileref, or SAS catalog to write to.

**file-specification** is one of the following:

**external-file**
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

**fileref**
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

**entry.markup**
specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

**suboption(s)**
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:
(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 515.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 515.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 516.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 516.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 516.

Note By default, if you do not specifically send the information to a separate file, then the style sheet information is included in the specified HTML file.

Example “Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

TAGSET= tagset-name
specifies a keyword value for a tagset. A tagset is a template that defines how to create a markup language output type from a SAS format. Tagsets produce markup output such as Hypertext Language (HTML) and Extensible Language (XML).

An alternate form for specifying a tagset is as follows:

ODS directory.tagset-name file-specification(s)<option(s)> ;
ODS directory.tagset-name action;
A directory can be TAGSET, a user-defined entry, or a libref. By default, the tagsets that SAS supplies are located in the directory TAGSETS, which is within the item
store Sasuser.Tmplmst. For more information about user-defined tagsets and item stores, see “TEMPLATE Procedure: Overview” in *SAS Output Delivery System: Procedures Guide*.

**Alias**

**TYPE=**

**Default**

If you do not specify a TAGSET= value, then the ODS MARKUP statement defaults to XML output.

**Interaction**

Using the TAGSET= option in an ODS markup family statement that refers to an open ODS markup destination forces ODS to close the destination and all associated files. ODS then opens a new instance of the destination. For more information, see “Opening and Closing the MARKUP Destination” on page 517.

**Tip**

SAS provides a set of tagset definitions. To get a list of the tagset names that SAS supplies, plus any tagsets that you created and stored in the Sasuser.Tmplmst template store, submit the following SAS statements:

```sas
proc template;
   list tagsets;
run;
```

For a list of valid tagsets and their descriptions, see “ODS Tagset Statement” on page 772.

For additional information about specifying tagsets, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in *SAS Output Delivery System: Procedures Guide*.

**Examples**

“Example 2: Creating an XML File and a DTD” on page 520

“Example 3: Creating Multiple Markup Output” on page 522

“Example 4: Specifying Tagset Names as ODS Destinations” on page 525

**TEXT=text-string**

Inserts text into your document by triggering the paragraph event and specifying a text string to be assigned to the VALUE event variable.

**Default**

By default the TEXT= option is used in a paragraph event.

**Tip**

You can specify a text-string for a specific event by using the TEXT= option with the EVENT= option by using the following syntax:

```
EVENT=event-name (TEXT=text-string)
```

**See**

For information about events and event variables, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in *SAS Output Delivery System: Procedures Guide*.

**Example**

“Example: Conditionally Excluding Output Objects and Sending Them to Different Output Destinations” on page 325

**TRANTAB= 'translation-table'**

Specifies the translation table to use when transcoding a file for output.

**Suboptions**

**(DYNAMIC)**

enables you to send output directly to a web server instead of writing it to a file. This option sets the value of the CONTENTTYPE= style attribute. For more information, see CONTENTTYPE= on page 1015 in PROC TEMPLATE.

Default If you do not specify DYNAMIC, then ODS sets the value of HTMLCONTENTTYPE= for writing to a file.

Restriction If you specify the DYNAMIC suboption with one of the following options in the ODS HTML statement, then you must specify it for all of these options in that statement.

- BODY=
- CONTENTS=
- PAGE=
- FRAME=
- STYLESHEET=
- TAGSET=

Requirements You must enclose DYNAMIC in parentheses.

You must specify DYNAMIC next to the *file-specification* specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the *tagset-name* specified by the TAGSET= option.

**(NO_BOTTOM_MATTER)**

specifies that no ending markup language source code be added to the output file.

Alias NOBOT

Requirements You must enclose NO_BOTTOM_MATTER in parentheses.

You must specify NO_BOTTOM_MATTER next to the *file-specification* specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the *tagset-name* specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

Interactions The NO_BOTTOM_MATTER suboption, in conjunction with the NO_TOP_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.
Tip
If you want to leave a body file in a state that you can append to with ODS, then use NO_BOTTOM_MATTER with the file-
specification BODY= option in any markup language statement.

See
The NO_TOP_MATTER suboption

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

Alias
NOTOP

Requirements
You must enclose NO_TOP_MATTER in parentheses.

You must specify NO_TOP_MATTER next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

Interactions
The NO_TOP_MATTER suboption, in conjunction with the NO_BOTTOM_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

See
The NO_BOTTOM_MATTER suboption and the ANCHOR= option

(TITLE=’title-text’)
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

Title
is the text in the metadata of a file that indicates the title.

Requirements
You must enclose TITLE= in parentheses.

You must enclose title-text in quotation marks.

Tip
If you are creating a web page that uses frames, then it is the TITLE= specification for the frame file that appears in the browser window title bar.

Example
“Example 3: Creating Multiple Markup Output” on page 522

(URL= ’Uniform-Resource-Locator’)
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

Requirements
You must enclose URL= ’Uniform-Resource-Locator’ in parentheses.
You must enclose Uniform-Resource-Locator in quotation marks.

You must specify URL='Uniform-Resource-Locator' next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

Tips

This option is useful for building HTML files that can be moved from one location to another. The links from the contents and page files must be constructed with a single name URL, and the contents, page, and body files must all be in the same location.

You never need to specify this suboption with the FRAME= option because ODS files do not reference the frame file.

Example

“Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

Details

Opening and Closing the MARKUP Destination

You can modify an open MARKUP destination with many ODS MARKUP options. However, the BODY= and TAGSET= options will automatically close the open destination that is specified in the ODS MARKUP statement. These options also close any files associated with the destination and then open a new instance of the destination. If you use one of these options, it is best if you explicitly close the destination yourself.

Specifying Multiple ODS Destinations

The ODS MARKUP statement opens or closes one destination. Like all single output destinations, you can have only one markup destination open at one time, unless you use the ID= option.

However, you can specify multiple simultaneous ODS destinations to produce multiple markup output by doing both of the following:

- specifying some of the TAGSET= value keywords as a destination
- specifying any two-level tagset name, such as TAGSETS.PYX, TAGSETS.STYLE_DISPLAY, or one of your own tagset names

Specifying a Tagset Keyword as an ODS Destination

You can specify some tagset keywords as ODS destinations. The tagset determines the type of markup that you have in your output file. For example, either of the following sets of statements are acceptable:

- ods markup body='class.html' tagset=phtml;
  ...
  more SAS statements...
  ods markup close;

- ods phtml body='class.html';
  ...
  more SAS statements...
  ods phtml close;

The ODS statement that you use to close a destination must be in the same form as the ODS statement that you used to open the destination. Therefore, the following is not acceptable, because SAS considers MARKUP and PHTML as separate destinations.
ods markup body='class.html' tagset=phtml;
    ...more SAS statements...
ods phtml close;

The tagsets that you can specify as both a TAGSET= value for ODS MARKUP or as a separate ODS destination are as follows:

- CHTML
- CSV
- CSVALL
- HTML4
- PHTML

**Specifying a Two-Level Tagset Name as an ODS Destination**

You can open a destination by specifying the markup that you want to produce by naming its two-level tagset name. You can specify all tagsets in this manner. For example, the following ODS statements open the MYTAGSET destination. The ODS _ALL_ CLOSE statement closes the MYTAGSET destination as well as all other open destinations.

```sas
ods tagsets.mytagset body='test1.xml';
    ...more SAS statements...
ods _all_ close;
```

You can also specify tagset names as follows, using the TYPE= option with a two-level tagset name:

```sas
ods markup type=tagsets.mytagset body='test.xml';
```

**Examples**

**Example 1: Creating an XML FILE**

**Features:**
- ODS MARKUP statement:
  - Action: CLOSE
  - Options: BODY=
- Other features:
  - PROC PRINT
- Data set:
  - StatePop

**Details**

The following ODS MARKUP example creates XML markup from PRINT procedure output. The TAGSET= option for the ODS MARKUP statement is not specified, so the ODS MARKUP statement defaults to XML output.

**Program**

```sas
ods markup body='population.xml';
proc print data=statepop;
run;
```
ods markup close;

Program Description

The ODS MARKUP BODY= statement creates an XML file.

ODS MARKUP BODY='population.xml';

Print the data set. The PRINT procedure prints the data set StatePop.

PROC PRINT DATA=STATEPOP;
RUN;

Close the MARKUP destination. The ODS MARKUP CLOSE statement closes the MARKUP destination and all the files that are associated with it. If you do not close the destination, then you cannot view the files.

ods markup close;
XML Output

The following partial output is tagged with XML (Extensible Markup Language) tags.

**Output 6.42  XML Markup from PRINT Procedure Output**

```xml
<?xml version="1.0" encoding="windows-1252"?>
<odsxml>
<head>
<meta operator="user"/>
</head>
<body>
<proc name="Univariate">
<label name="IDX"/>
<title class="SystemTitle" toc-level="1">US Census of Population and Housing</title>
<proc-title class="ProcTitle" toc-level="1">The UNIVARIATE Procedure</proc-title>
<proc-title class="ProcTitle" toc-level="1">Variable: CityPop_90 (1990 metropolitan pop in millions)</proc-title>
<branch name="Univariate" label="The Univariate Procedure" class="ContentProcName" toc-level="1">
<branch name="CityPop_90" label="CityPop_90" class="ContentFolder" toc-level="2">
<leaf name="Moments" label="Moments" class="ContentItem" toc-level="3">
<output name="Moments" label="Moments" clabel="Moments">
<output-object type="table" class="Table">
<style>
<border spacing="1" padding="7" rules="groups" frame="box"/>
</style>
<colspecs columns="4">
<colgroup>
<colspec name="1" width="15" type="string"/>
<colspec name="2" width="10" align="right" type="string"/>
<colspec name="3" width="16" type="string"/>
<colspec name="4" width="10" align="right" type="string"/>
</colgroup>
... more tagged output ...
</output-object>
</output>
</leaf>
</branch>
<footnote class="SystemFooter" toc-level="1">^[super *]This is a ^S={foreground=black}footnote.</footnote>
</proc>
</body>
</odsxml>
```

**Example 2: Creating an XML File and a DTD**

**Features:**
- ODS MARKUP statement action: `CLOSE`
- ODS MARKUP statement options:
  - `BODY=`
  - `FRAME=`
  - `TAGSET=`

**Other features:**
- PROC UNIVARIATE
TITLE statement

Data set:  
StatePop

Details
The following ODS MARKUP example creates an XML file and its Document Type Definition (DTD) from PROC UNIVARIATE output.

Program

   ods html close;
   ods markup body='statepop.xml'  
       frame='statepop.dtd' tagset=default;
   proc univariate data=statepop;  
      var citypop_90 citypop_80;  
      title 'US Census of Population and Housing';  
      run;
   ods markup close;

Program Description

Close the HTML destination so that no HTML output is produced. The HTML destination is open by default. The ODS HTML statement closes the HTML destination to conserve resources.

   ods html close;

Create XML output and a DTD. The ODS MARKUP BODY= statement creates an XML file. The FRAME= option specifies that you want the DTD in a frame file, and the TAGSET= option specifies that you want the default tagset, which is XML.

   ods markup body='statepop.xml'  
       frame='statepop.dtd' tagset=default;

Generate the statistical tables for the analysis variables. The UNIVARIATE procedure calculates univariate statistics for numeric variables in the StatePop data set. The VAR statement specifies the analysis variables and their order in the output. The TITLE statement specifies a title for the output object.

   proc univariate data=statepop;  
      var citypop_90 citypop_80;  
      title 'US Census of Population and Housing';  
      run;

Close the MARKUP destination. The ODS MARKUP CLOSE statement closes the MARKUP destination and all the files that are associated with it. If you do not close the destination, then you are not able to view the files.

   ods markup close;
Output

This DTD specifies how the markup tags in a group of SGML or XML documents should be interpreted by an application that displays, prints, or otherwise processes the documents.

Output 6.43  DTD Created by the ODS MARKUP Statement

```xml
<!ELEMENT odsxml (head?, body)>  
<!ELEMENT head (meta|css)*>  
<!ELEMENT body (label|page)*|proc)+>  
<!ELEMENT meta EMPTY>  
<!ATTLIST meta  
  operator CDATA #IMPLIED  
  author CDATA #IMPLIED>  
<!ELEMENT css EMPTY>  
<!ATTLIST css  
  file CDATA #IMPLIED>  
<!ELEMENT label EMPTY>  
<!ATTLIST label  
  name ID #IMPLIED>  
<!ELEMENT proc (title|proc-title|note|page|label|style|branch|output)+>  
<!ATTLIST proc  
  class CDATA #IMPLIED>  
... more tagged output ...  
<!ELEMENT br EMPTY>  
<!ELEMENT page EMPTY>  
<!ELEMENT b (#PCDATA|it|b|ul)*>  
<!ELEMENT ul (#PCDATA|it|b|ul)*>  
<!ELEMENT it (#PCDATA|it|b|ul)*>  
<!ELEMENT style (span|align|border)*>  
<!ELEMENT span EMPTY>  
<!ATTLIST span  
  columns CDATA #IMPLIED  
  rows CDATA #IMPLIED>  
<!ELEMENT align EMPTY>  
<!ATTLIST align  
  horiz {left|center|right|justify} "left">  
<!ELEMENT border EMPTY>  
<!ATTLIST border  
  rules {none|groups|rows|cols|all} #IMPLIED  
  frame {void|above|below|hsides|lhs|rhs|vsides|box|border} #IMPLIED  
  padding CDATA #IMPLIED  
  spacing CDATA #IMPLIED>
```

Example 3: Creating Multiple Markup Output

Features:
- ODS CVSALL statement option:  
  BODY=  

- ODS MARKUP statement options:  
  BODY=  
  TAGSET=  
  TITLE=  

Other features:  
- OPTIONS statement  
- PROC PRINT  
- TITLE statement  

Data set:
Grain_Production

Details
The following ODS example creates two different types of markup output from the same procedure output. To create two markup outputs requires two ODS destinations. Because ODS MARKUP is considered one destination, you cannot specify two tagsets without the use of the ID= option. However, you can specify one output using ODS MARKUP. You can then specify the other output using ODS syntax in which the tagset is the destination.

Program

```ods html close;
options obs=15;
ods csvall body='procprintcsvall.csv';
ods markup tagset=chtml body='procprintchtml.html'
   {title= 'This Text Identifies Your Content.'};
title 'Leading Grain-Producing Countries';
proc print data=grain_production;
run;
ods csvall close;
ods markup tagset=chtml close;
```

Program Description

Close the HTML destination so that no HTML output is produced. The HTML destination is open by default. The ODS HTML statement closes the HTML destination to conserve resources. The OPTIONS statement specifies that only fifteen observations be used.

```ods html close;
options obs=15;
```

Create tabular output. The ODS CSV ALL statement produces tabular output with titles that contain columns of data values that are separated by commas.

```ods csvall body='procprintcsvall.csv';
```

Create CHTML output. The ODS MARKUP TAGSET=CHTML statement produces compact, minimal HTML output that does not use style information, and a hierarchical table of contents. The TITLE= option specifies the text that appears in the browser window title bar.

```ods markup tagset=chtml body='procprintchtml.html'
   {title= 'This Text Identifies Your Content.'};
```

Print the data set. The PRINT procedure prints the data set Grain_Production. The TITLE statement specifies the title.

```title 'Leading Grain-Producing Countries';
proc print data=grain_production;
run;
```
Close the open destinations so that you can view or print the output. The ODS CSVALL CLOSE statement closes the CSVALL destination and all of the files that are associated with it. The ODS MARKUP TAGSET=CHTML CLOSE statement closes the MARKUP destination and all of the files that are associated with it. You must close the destinations before you can view the output with a browser or before you can send the output to a physical printer.

```plaintext
ods csvall close;
ods markup tagset=chtml close;
```

Output

The following output was created by specifying the MARKUP TAGSET=CHTML statement. The text "This Text Identifies Your Content." was specified by the TITLE= option.

Output 6.44  CHTML Output

```
<table>
<thead>
<tr>
<th>Obs</th>
<th>Country</th>
<th>Type</th>
<th>Year</th>
<th>Kilotons</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BRZ</td>
<td>Wheat</td>
<td>1995</td>
<td>1516</td>
</tr>
<tr>
<td>2</td>
<td>BRZ</td>
<td>Rice</td>
<td>1995</td>
<td>11236</td>
</tr>
<tr>
<td>3</td>
<td>BRZ</td>
<td>Corn</td>
<td>1995</td>
<td>36276</td>
</tr>
<tr>
<td>4</td>
<td>CHN</td>
<td>Wheat</td>
<td>1995</td>
<td>102207</td>
</tr>
<tr>
<td>5</td>
<td>CHN</td>
<td>Rice</td>
<td>1995</td>
<td>185226</td>
</tr>
<tr>
<td>6</td>
<td>CHN</td>
<td>Corn</td>
<td>1995</td>
<td>112331</td>
</tr>
<tr>
<td>7</td>
<td>IND</td>
<td>Wheat</td>
<td>1995</td>
<td>63007</td>
</tr>
<tr>
<td>8</td>
<td>IND</td>
<td>Rice</td>
<td>1995</td>
<td>122372</td>
</tr>
<tr>
<td>9</td>
<td>IND</td>
<td>Corn</td>
<td>1995</td>
<td>9500</td>
</tr>
<tr>
<td>10</td>
<td>IND</td>
<td>Wheat</td>
<td>1995</td>
<td>11171</td>
</tr>
<tr>
<td>11</td>
<td>IND</td>
<td>Rice</td>
<td>1995</td>
<td>49260</td>
</tr>
<tr>
<td>12</td>
<td>IND</td>
<td>Corn</td>
<td>1995</td>
<td>82223</td>
</tr>
<tr>
<td>13</td>
<td>USA</td>
<td>Wheat</td>
<td>1995</td>
<td>29494</td>
</tr>
<tr>
<td>14</td>
<td>USA</td>
<td>Rice</td>
<td>1995</td>
<td>7888</td>
</tr>
<tr>
<td>15</td>
<td>USA</td>
<td>Corn</td>
<td>1995</td>
<td>187300</td>
</tr>
</tbody>
</table>
```
The following output was created by specifying the ODS CSVALL statement. Note that you cannot specify ODS MARKUP TAGSET=CSVALL and ODS MARKUP TAGSET=CHTML together.

**Output 6.45  CSVALL Output Viewed in Microsoft Excel**

---

### Example 4: Specifying Tagset Names as ODS Destinations

When you specify tagsets and two-level tagset names as destinations, you can open and close multiple destinations, producing multiple markup output. Here is an example:

```sas
ods phtml body='test1.html';
ods chtml body='test2.html';
ods markup body='test1.xml';
ods tagsets.event_map body='test2.xml';
...more SAS statements...
...ods chtml close;
...more SAS statements...
ods _all_ close;
```

---

### Example 5: Including Multiple Cascading Style Sheets in One HTML Document

**Features:**
- ODS HTML statement action: `CLOSE`
- ODS HTML statement options:
  - `BODY=`
  - `STYLESHEET=`
  - `URL=`

**Other features:**
- `OPTIONS` statement
- `PROC PRINT`
TITLE statement

Data set:
Grain_Production

Details

The following example creates one HTML document and two style sheets, which are included in the HTML document. The URLs are created in the order specified by the URL= suboption.

Program

ods html close;
opts obs=15;
ods html body='StylesheetExample.html'
   stylesheet=(url='/css/file1.css /css/file2.css');
proc print data=grain_production;
title 'Leading Grain-Producing Countries';
run;
ods html close;

Program Description

Close the HTML destination so that no HTML output is produced. The HTML destination is open by default. The ODS HTML statement closes the HTML destination to conserve resources. The OPTIONS statement specifies that only fifteen observations be used.

ods html close;
opts obs=15;

Create the HTML output and two style sheets. The ODS HTML statements opens the HTML destination and creates HTML output. The STYLESTHEET= option places the style information for the HTML output into two external files. The URL= suboption specifies a URL for the two files, File1.css and File2.css. ODS uses these URLs (instead of the filename) in all the links and references that it creates and that point to those files.

ods html body='StylesheetExample.html'
   stylesheet=(url='/css/file1.css /css/file2.css');

Print the data set. The PRINT procedure prints the data set Grain_Production. The TITLE statement specifies the title.

proc print data=grain_production;
title 'Leading Grain-Producing Countries';
run;

Close the HTML destination. The ODS HTML CLOSE statement closes the HTML destination and all the files that are associated with it. If you do not close the destination, then you cannot view the files.

ods html close;
Output

The two links to the style sheets that the STYLESHEET= option creates are at the bottom of the partial output. The links are created in the order in which they were specified by the URL= suboption.

Output 6.46  HTML Code

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta name="Generator" content="SAS Software Version 9.4, see www.sas.com">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<title>SAS Output</title>
<link rel="stylesheet" type="text/css" href="/css/file1.css">
<link rel="stylesheet" type="text/css" href="/css/file2.css">
<style type="text/css">
    .l {text-align: left }
    .c {text-align: center }
    .r {text-align: right }
    .d {text-align: right }
    .j {text-align: justify }
    .t {vertical-align: top }
    .m {vertical-align: middle }
    .b {vertical-align: bottom }
    TD, TH {vertical-align: top }
    .stacked_cell{padding: 0 }
    -->
</style>
</head>
</html>
```

Example 6: Applying a CSS File to ODS Output

Features:
- ODS HTML statement options:
  - BODY=
  - CSSSTYLE=media-type
  - TEXT=
- ODS PDF statement options:
  - BODY=
  - CSSSTYLE=media-type
  - STARTPAGE=
  - TEXT=
- ODS RTF statement options:
  - BODY=
  - CSSSTYLE=media-type
  - TEXT=

Other features:
- PROC CONTENTS

Details

The following program applies a style sheet created in a CSS file to HTML, PDF, and RTF output. Because the CSS file has media blocks with additional information for screen and print media types, you can specify that each output destination use the additional style information for a specific media type.

The following code is an example of the external CSS file StyleSheet.css. There are two media types specified in this program, Print and Screen. Copy and paste this code into a text editor and save it as StyleSheet.css.
Program

options nodate pageno=1 linesize=80 pagesize=40 obs=10;
ods html file="StyleSheet.html" cssstyle='stylesheet.css' (screen)
text="Style Sheet Using Screen Media Type";
ods rtf file="StyleSheet.rtf" cssstyle='stylesheet.css' (print)
text="Style Sheet Using Print Media Type";
ods pdf file="StyleSheet.pdf" cssstyle='stylesheet.css' (print screen) startpage=no
   text="Style Sheet Using Both Media Types";
proc contents data=sashelp.class;
run;
ods _all_ close;

Program Description

**Apply the CSS file to your output.** The CSSSTYLE= option on the ODS HTML, ODS RTF, and ODS PDF statements applies the CSS file StyleSheet.css to the output for each destination. In the ODS HTML statement, specifying the CSSSTYLE= option for the media-type Screen applies the style information in the Screen media type block, in addition to style information outside of any media blocks, to the HTML output. Similarly, the RTF output uses the additional information from the Print media block. The PDF output uses all of the code in the CSS file, because both Print and Screen are specified.

options nodate pageno=1 linesize=80 pagesize=40 obs=10;
ods html file="StyleSheet.html" cssstyle='stylesheet.css'(screen)
  text="Style Sheet Using Screen Media Type";
ods rtf file="StyleSheet.rtf" cssstyle='stylesheet.css'(print)
  text="Style Sheet Using Print Media Type";
ods pdf file="StyleSheet.pdf" cssstyle='stylesheet.css'(print screen)
  STARTPAGE=no
  text="Style Sheet Using Both Media Types";

**View the contents of the SAS data set.** The CONTENTS procedure shows the contents of the SAS data set Sashelp.Class.

proc contents data=sashelp.class;
run;

**Close the open destinations.** The ODS _ALL_ CLOSE statement closes all open destinations and the files that are associated with them. If you do not close the destinations, then you cannot view the files.

ods _all_ close;

**Output**

The yellow and green background colors, the white font color, the font size, and border information all come from the Screen media block. All other style information comes
from the code outside of the media blocks. No information from the Print media block is used.

**Output 6.47**  
*HTML Output Using Both a Style Sheet with Screen Media Type*

**Style Sheet Using Screen Media Type**

**The CONTENTS Procedure**

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>SASHELP.CLASS</th>
<th>Observations</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member Type</td>
<td>DATA</td>
<td>Variables</td>
<td>5</td>
</tr>
<tr>
<td>Engine</td>
<td>%9</td>
<td>Indexes</td>
<td>0</td>
</tr>
<tr>
<td>Created</td>
<td>Thursday, May 19, 2005 03:25:02 PM</td>
<td>Observation Length</td>
<td>40</td>
</tr>
<tr>
<td>Last Modified</td>
<td>Thursday, May 19, 2005 03:25:02 PM</td>
<td>Deleted Observations</td>
<td>0</td>
</tr>
<tr>
<td>Protection</td>
<td>Compressed</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Data Set Type</td>
<td>Sorted</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Representation</td>
<td>WINDOWS_22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encoding</td>
<td>w-asci ASCII (ASCII)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Engine/Host Dependent Information**

<table>
<thead>
<tr>
<th>Data Set Page Size</th>
<th>4096</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Data Set Pages</td>
<td>1</td>
</tr>
<tr>
<td>First Data Page</td>
<td>1</td>
</tr>
<tr>
<td>Max Obs per Page</td>
<td>101</td>
</tr>
<tr>
<td>Obs in First Data Page</td>
<td>19</td>
</tr>
<tr>
<td>Number of Data Set Repairs</td>
<td>0</td>
</tr>
<tr>
<td>File Name</td>
<td>C:\SAS9\sasv94\d\wsas92\wsas_d9\data\dictionary\sashelp\class.sas7bdat</td>
</tr>
<tr>
<td>Release Created</td>
<td>9.020160</td>
</tr>
<tr>
<td>Host Created</td>
<td>WIN_PRO</td>
</tr>
</tbody>
</table>

**Alphabetic List of Variables and Attributes**

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Age</td>
<td>Num</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>Height</td>
<td>Num</td>
<td>8</td>
</tr>
</tbody>
</table>
Output 6.48  RTF Output Using a Style Sheet with Print Media Type

The CONTENTS Procedure

Style Sheet Using Print Media Type

<table>
<thead>
<tr>
<th>Data Set Name</th>
<th>SASHELP.CLASS</th>
<th>Observations</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member Type</td>
<td>DATA</td>
<td>Variables</td>
<td>5</td>
</tr>
<tr>
<td>Engine</td>
<td>99</td>
<td>Indexes</td>
<td>0</td>
</tr>
<tr>
<td>Created</td>
<td>Thursday, May 19, 2003 03:25:32 PM</td>
<td>Observation Length</td>
<td>40</td>
</tr>
<tr>
<td>Last Modified</td>
<td>Thursday, May 19, 2003 03:25:02 PM</td>
<td>Deleted Observations</td>
<td>0</td>
</tr>
<tr>
<td>Protection</td>
<td>Compressed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Set Type</td>
<td>WINDOW32</td>
<td>Sorted</td>
<td>NO</td>
</tr>
<tr>
<td>Label</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Representation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encoding</td>
<td>ASCII</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Engine/Host Dependent Information

<table>
<thead>
<tr>
<th>Data Set Page Size</th>
<th>4096</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Data Set Pages</td>
<td>1</td>
</tr>
<tr>
<td>First Data Page</td>
<td>1</td>
</tr>
<tr>
<td>Max Obs per Page</td>
<td>101</td>
</tr>
<tr>
<td>Obs in First Data Page</td>
<td>19</td>
</tr>
<tr>
<td>Number of Data Set Repairs</td>
<td>0</td>
</tr>
<tr>
<td>File Name</td>
<td>C:\SAS\Voage\dev\v920\sas_dv\bin\v920\sas\help\class\sas\v920d.tzt</td>
</tr>
<tr>
<td>Release Created</td>
<td>2001-01-01</td>
</tr>
<tr>
<td>Host Created</td>
<td>WIN_PRO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alphabetical List of Variables and Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Variable TypeLen</td>
</tr>
<tr>
<td>Age</td>
</tr>
<tr>
<td>Height</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Sex</td>
</tr>
<tr>
<td>Weight</td>
</tr>
</tbody>
</table>

The PDF output uses all of the style information in the CSS file, including the information from both media types. However, both the Print and Screen media blocks
have a background color specified for row and column headings. The blue background color is picked up because it is specified last.

**Output 6.49** **PDF Output Using Both a Style Sheet with Both Print and Screen Media Types**

**Example 7: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information**

**Features:**

- **ODS TAGSETS.HTMLPANEL statement action:**
  
  CLOSE

- **ODS TAGSETS.HTMLPANEL statement options:**
  
  OPTIONS (DOC="HELP")
  
  FILE=

**Other features:**

- PROC PRINT

**Details**

The following example prints to the SAS log the OPTIONS suboptions and a description of each available suboption.
Program

ods tagsets.htmlpanel file='Help.html' options (doc="help");
proc print data=Sashelp.Class;
run;
ods _all_ close;

Program Description

Print information about the OPTIONS suboptions to the SAS log file. Specifying the OPTIONS suboption (DOC='HELP') prints Help for the ODS TAGSETS.HTMLPANEL statement suboptions to the SAS log file. The FILE= option prints the results to a file named Help.html.

ods tagsets.htmlpanel file='Help.html' options (doc="help");

Print the data set Sashelp.Class. The PROC PRINT statement prints the Sashelp.Class data set.

proc print data=Sashelp.Class;
run;

Close all destinations. Close the ODS TAGSETS.HTMLPANEL destination and any other open destinations. This statement also closes all the files that are associated with each open destination. If you do not close a destination, then you cannot view the files in a browser window.

ods _all_ close;

SAS Log Output

Specify the "DOC='HELP' suboption to print all of the OPTIONS suboptions and information about each of the suboptions to the SAS log.

---

**ODS NO_DECIMAL_ALIGN Statement**

Right-justifies numeric columns when no justification is specified.

- **Valid in:** Anywhere
- **Category:** ODS: SAS Formatted
- **Alias:** ODS DECIMAL_ALIGN=NO
- **See:** "Values in Table Columns and How They Are Justified" in SAS Output Delivery System: Procedures Guide

**Syntax**

```
ODS NO_DECIMAL_ALIGN;
```

**Without Arguments**

The ODS NO_DECIMAL_ALIGN statement right-justifies values when no justification is specified. ODS NO_DECIMAL_ALIGN is the default setting.
ODS OUTPUT Statement

Produces a SAS data set from an output object and manages the selection and exclusion lists for the OUTPUT destination.

Valid in: Anywhere
Category: ODS: SAS Formatted
Note: When using the ODS OUTPUT statement, set system option REPLACE=YES to ensure that your data set is replaced. See "REPLACE System Option" in SAS System Options: Reference for more details.

Syntax

ODS OUTPUT action;
ODS OUTPUT data-set-definition(s);

Actions

The following actions are available for the ODS OUTPUT statement:

CLEAR
sets the list for the OUTPUT destination to EXCLUDE ALL.

CLOSE
closes the OUTPUT destination. When an ODS destination is closed, ODS does not send output to that destination. Closing a destination frees some system resources.

SHOW
writes to the SAS log the current selection or exclusion list for the OUTPUT destination. If the list is the default list (EXCLUDE ALL), then SHOW also writes the current overall selection or exclusion list.

Required Arguments

data-set-definition
provides instructions for turning an output object into a SAS data set. ODS maintains a list of these definitions. This list is the selection list for the OUTPUT destination. For information about how ODS manages this list, see “Selection and Exclusion Lists” on page 39. Each data-set-definition has the following form:

output-object-specification<=data-set

output-object-specification
has the following form:

output-object<(MATCH_ALL<=macro-var-name) PERSIST=PROC | RUN)>

output-object
identifies one or more output objects to turn into a SAS data set.
To specify an output object, you need to know which output objects your SAS program produces. The ODS TRACE statement writes to the SAS log a trace record that includes the path, the label, and other information about each output object that is produced. For more information, see the ODS TRACE statement on page 854. Output Objects can be specified as the following:

- a full path. For example, the following is the full path of the output object:

  Univariate.City_Pop_90.TestsForLocation

- a partial path. A partial path consists of any part of the full path that begins immediately after a period (.) and continues to the end of the full path. For example, suppose the full path is the following:

  Univariate.City_Pop_90.TestsForLocation
  Then the partial paths are as follows:

  City_Pop_90.TestsForLocation
  Tests For Location

- a label that is enclosed in quotation marks. For example:

  "TestsForLocation"

- a label path. For example, the label path for the output object is as follows:

  "The UNIVARIATE Procedure"."CityPop_90"."Tests For Location"

  Note: The trace record shows the label path only if you specify the LABEL option in the ODS TRACE statement.

- a partial label path. A partial label path consists of any part of the label that begins immediately after a period (.) and continues to the end of the label. For example, suppose the label path is the following:

  "The UNIVARIATE Procedure"."CityPop_90"."Tests For Location"

  Then the partial label paths are as follows:

  >"CityPop_90"."Tests For Location"
  "Tests For Location"</

- a mixture of labels and paths.
- any of the partial path specifications, followed by a pound sign (#) and a number. For example, TestsForLocation#3 refers to the third output object that is named TestsForLocation.

  Tip: To create multiple data sets from the same output object, list the output object as many times as you want. Each time you list the output object, specify a different data set.

MATCH_ALL=<macro-var-name>
creates a new data set for each output object. For an explanation of how ODS names these data sets, see the discussion of the option “data-set” on page 536.
**macro-var-name**

specifies the macro variable where a list of all the data sets that are created are stored. If you want to concatenate all the data sets after the PROC step, you can use the macro variable to specify all the data sets in a DATA step.

**Tip**

The MATCH_ALL option is not needed to merge conflicting output objects into one data set.

**CAUTION**

A data set that is produced by SAS 9.1 without MATCH_ALL might not be identical to a data set that is produced by SAS 9.0 with MATCH_ALL and then concatenated in a DATA step. With SAS 9.0, merging dissimilar output objects with the MATCH_ALL option could result in missing columns or truncated variables. With SAS 9.1, these restrictions do not apply. For more information about merging output objects, see “Merging Dissimilar Output Objects into One Data Set” on page 537.

**PERSIST=PROC | RUN**

determines when ODS closes any data sets that it is creating, and determines when ODS removes output objects from the selection list for the OUTPUT destination.

**PROC**

maintains the list of definitions even after the procedure ends, until you explicitly modify it. To modify the list, use ODS OUTPUT with one or more data-set-specifications. To set the list for the OUTPUT destination to EXCLUDE ALL, use the following statement:

```plaintext
ods output clear;
```

**RUN**

maintains the list of definitions and keeps open the data sets that it is creating even if the procedure or DATA step ends, or until you explicitly modify the list.

**See**

“How ODS Determines the Destinations for an Output Object” on page 38

**data-set**

names the SAS output data set. You can use a one-level or two-level (with a libref) name.

If you are creating a single data set, then the ODS OUTPUT statement simply uses the name that you specify. If you are creating multiple data sets with MATCH_ALL, then the ODS OUTPUT statement appends numbers to the name. For example, if you specify `test` as `data-set` and you create three data sets, then ODS names the first data set `test`. The additional data sets are named `test1` and `test2`.

**Note:** If you end the filename with a number, then ODS begins incrementing the name of the file with that number. For example, if you specify `may5` as `data-set` and you create three data sets, then ODS names the first data set `may5`. The additional data sets are named `may6` and `may7`.

**Default**

If you do not specify a data set, then ODS names the output data set `DATAn`, where `n` is the smallest integer that makes the name unique.
Tip  You can specify data set options in parentheses immediately after data-set.

NOWARN
 suppresses the warning that an output object was requested but not created.

Requirement
 The NOWARN option must be enclosed in parentheses.

Interaction
 The NOWARN option cannot be used with the ALL option or the NONE option.

Example
 The ODS EXCLUDE statement in the following example specifies that no warning is created if the output object Summary is requested but not created.

```plaintext
ods exclude summary (nowarn);

proc contents data=sashelp.class;
run;
```

SHOW
 functions just like the ODS SHOW statement except that it writes only the selection or exclusion list for the OUTPUT destination.

Details

**Merging Dissimilar Output Objects into One Data Set**
By default, the ODS OUTPUT statement puts all output objects that have the same output-path into one SAS data set, regardless of any conflicting variables in the output objects. Variables created by a later output object will get a value of missing in the observations created by the earlier output object. Variables created by an earlier output object that do not exist in a subsequent output object will get a value of missing in the observations added by the later output object. If a variable created by an output object has a different type than a variable with the same name created by an earlier output object, it will be added to the output data set using a new name formed by adding a numeric suffix.

Examples

**Example 1: Creating a Combined Output Data Set**

**Features:**
- ODS _ALL_ CLOSE statement
- ODS HTML statement options:
  - BODY=
  - CONTENTS=
  - FRAME=
  - PAGE=
- ODS LISTING statement:
  - CLOSE
- ODS OUTPUT statement

**Other features:**
- PROC FORMAT
This example routes two output objects that PROC TABULATE produces to both the OUTPUT destination and the HTML destination. The result is two output objects that are combined by the ODS OUTPUT statement to create an output data set formatted as HTML output by the ODS HTML statement.

**Note:** This example uses filenames that might not be valid in all operating environments. To successfully run the example in your operating environment, you might need to change the file specifications. See Appendix 4, “ODS HTML Statements for Running Examples in Different Operating Environments,” on page 1113.

**Program**

```sas
proc format;
  value regfmt 1='Northeast'
          2='South'
          3='Midwest'
          4='West';
  value divfmt 1='New England'
          2='Middle Atlantic'
          3='Mountain'
          4='Pacific';
  value usetype 1='Residential Customers'
                2='Business Customers';
run;

ods output Table=energyoutput(keep=region division type expenditures_sum);
ods html body='your_body_file.html' frame='your_frame_file.html' contents='your_contents_file.html' page='your_page_file.html';
proc tabulate data=energy format=dollar12.;
  by region;
  class division type;
  var expenditures;
  table division,
         type*expenditures;
  format region regfmt. division divfmt. type usetype.;
  title 'Energy Expenditures for Each Region';
  title2 '(millions of dollars)';
run;
ods html path='../../../ods' (url=none)
        body='odsoutput-printbody.htm';
ods html body='your_body_file_2.html';
proc print data=energyoutput noobs;
```
title 'Combined Output Data Set';
run;
ods _all_ close;
ods HTML;

Program Description

Format the variables Region, Division, and Type. PROC FORMAT creates formats for Region, Division, and Type.

proc format;
  value regfmt 1='Northeast'
    2='South'
    3='Midwest'
    4='West';
  value divfmt 1='New England'
    2='Middle Atlantic'
    3='Mountain'
    4='Pacific';
  value usetype 1='Residential Customers'
    2='Business Customers';
run;

Create the SAS output data set and specify the variables that you want to be written to the output SAS data set. The ODS OUTPUT statement creates the SAS data set EnergyOutput from the output objects that PROC TABULATE produces. The name of each output object is Table. You can determine the name of the output objects by using the ODS TRACE ON statement. The KEEP= data set option limits the variables in the output data set EnergyOutput to Region, Division, and Expenditures_sum. The variable name Expenditures_sum is generated by PROC TABULATE to indicate that the sum statistic was generated for the Expenditures variable. For more information, see “ODS TRACE Statement” on page 854.

ods output Table=energyoutput(keep=region division type expenditures_sum);

Create HTML output. The ODS HTML statement creates the body, frame, contents, and pages files. The output from PROC TABULATE is sent to the body file. FRAME=, CONTENTS=, and PAGE= create a frame that includes a table of contents and a table of pages that link to the contents of the body file. The body file also appears in the frame.

ods html body='your_body_file.html'
    frame='your_frame_file.html'
    contents='your_contents_file.html'
    page='your_page_file.html';

Create output data sets and an HTML report. This PROC TABULATE step creates two output objects named Table, one for each BY group, and adds them to the EnergyOutput data set. Because the HTML destination is open, ODS writes the output to the body file.

proc tabulate data=energy format=dollar12.;
  by region;
  class division type;
  var expenditures;
  table division,
    type*expenditures;
Close the current body file and open a new file. Create HTML output. The ODS HTML BODY= statement closes the original body file and opens a new one. The contents, page, and frame files remain open. The contents and page files will contain links to both body files. The ODS HTML statement opens the HTML destination and creates HTML output. The output from PROC TABULATE is sent to the body file. FRAME=, CONTENTS=, and PAGE= create a frame that includes a table of contents and a table of pages that link to the contents of the body file. The body file also appears in the frame.

ods html body='your_body_file_2.html';

Print the combined data set. This PROC PRINT step prints the data set EnergyOutput that contains both BY groups. The output is added to the current body file, your_body_file_2.html.

proc print data=energyoutput noobs;  
   title 'Combined Output Data Set';  
run;

Close all of the open destinations. The ODS _ALL_ CLOSE statement closes all open ODS output destinations. To return ODS to its default setup, the ODS HTML statement opens the HTML destination.

ods _all_ close;  
ods HTML;
HTML Output

The following HTML output shows the output data set that is created by the ODS OUTPUT statement.

**Output 6.50 Combined Data Set**

![Combined Output Data Set](image)

The following output shows the two separate BY groups that are created by the TABULATE procedure.

**Output 6.51 Output Objects Created by PROC TABULATE**

![Output Objects Created by PROC TABULATE](image)
Example 2: Using Different Procedures to Create a Data Set from Similar Output Objects

Features:
- ODS HTML statement options:
  - BODY=
  - CONTENTS=
  - FRAME=
- ODS OUTPUT statement
- ODS SELECT statement

Other features:
- PROC GLM
- PROC PRINT
- PROC REG

Data set:
- Iron

Details
This example creates and prints a data set that is created from the parameter estimates that PROC REG and PROC GLM generate. These procedures are part of SAS/STAT software.

Note: This example uses filenames that might not be valid in all operating environments. To successfully run the example in your operating environment, you might need to change the file specifications. See Appendix 4, “ODS HTML Statements for Running Examples in Different Operating Environments,” on page 1113.

Program
```sas
options nodate pageno=1 pagesize=60 linesize=72;
ods html body='parameter-estimates-body.htm'
   frames='parameter-estimates-frame.htm'
   contents='parameter-estimates-contents.htm';
ods select ParameterEstimates(persist);
ods output ParameterEstimates(persist=proc)=IronParameterEstimates;
proc reg data=iron;
   model loss=fe;
   title 'Parameter Estimate from PROC REG';
run;
quit;
proc glm data=iron;
   model loss=fe;
   title 'Parameter Estimate from PROC GLM';
run;
quit;
ods select all;
proc print data=IronParameterEstimates noobs;
   title 'PROC PRINT Report of the Data set from PROC REG';
run;
```
Example 2: Using Different Procedures to Create a Data Set from Similar Output Objects

ods _all_ close;
ods html;

Program Description

Set the SAS system options for the LISTING output. The NODATE option suppresses the display of the date and time in the LISTING output. The PAGENO= option specifies the starting page number. The PAGESIZE= option specifies the number of lines on an output page. The LINESIZE= option specifies the output line length.

    options nodate pageno=1 pagesize=60 linesize=72;

Create HTML output. The ODS HTML statement creates the body, frame, and contents files. The FRAME= and CONTENTS= options create a frame that includes a table of contents that links to the contents of the body file. The body file also appears in the frame.

    ods html body='parameter-estimates-body.htm'
        frame='parameter-estimates-frame.htm'
        contents='parameter-estimates-contents.htm';

Specify the output objects to be sent to all open ODS destinations. The ODS SELECT statement specifies that output objects named ParameterEstimates should be sent to all open ODS destinations that do not specifically exclude them. The LISTING destination is open by default, and its default list is SELECT ALL. The ODS HTML statement has opened the HTML destination, and its default list is also SELECT ALL. Thus, any object that is named ParameterEstimates will go to both these destinations. The PERSIST option specifies that ParameterEstimates should remain in the overall selection list until the list is explicitly modified.

    ods select ParameterEstimates(persist);

Create the IronParameterEstimates data set. The ODS OUTPUT statement opens the OUTPUT destination and creates the SAS data set IronParameterEstimates. By default, the list for the OUTPUT destination is EXCLUDE ALL. This ODS OUTPUT statement puts ParameterEstimates in the selection list for the destination. The PERSIST=PROC option specifies that ParameterEstimates should remain in the overall selection list until the procedure ends or the list is explicitly modified.

    ods output ParameterEstimates(persist=proc)=IronParameterEstimates;

Create the output objects. PROC REG and PROC GLM each produce an output object named ParameterEstimates. Because the data set definition persists when the procedure ends, ODS creates an output object from each one.

    proc reg data=iron;
        model loss=fe;
        title 'Parameter Estimate from PROC REG';
        run;
        quit;
    proc glm data=iron;
        model loss=fe;
        title 'Parameter Estimate from PROC GLM';
        run;
        quit;
Enable all open destinations to receive output objects. The ODS SELECT ALL statement sets the lists for all destinations to their defaults so that ODS sends all output objects to the HTML and LISTING destinations. (Without this statement, none of the output objects from the following PROC PRINT steps would be sent to the open destinations.)

    ods select all;

Print the reports. The PROC PRINT steps print the data set that ODS created from PROC REG and PROC GLM. The output from these steps goes to both the HTML and the LISTING destinations. Links to the HTML output are added to the contents file.

    proc print data=IronParameterEstimates noobs; 
    title 'PROC PRINT Report of the Data set from PROC REG';
    run;

Close the OUTPUT and HTML destinations. The ODS _ALL_ CLOSE statement closes all open destinations. The ODS HTML statement opens the HTML destination and returns ODS to its default setting.

    ods _all_ close;
    ods html;

HTML Output

The HTML output includes the parameter estimates from PROC REG, the parameter estimates from PROC GLM, and a report of the data set that ODS created from each set of parameter estimates.

The table of contents identifies output objects by their labels. The label for ParameterEstimates in PROC REG is Parameter Estimates. The corresponding label in PROC GLM is Solution. Notice how the column widths in the HTML output are automatically adjusted to fit the data. Compare this layout to the layout of the columns in the LISTING output.
Example 3: Creating a Data Set with and without the MATCH_ALL Option

Features:

ODS HTML statement options:

BODY=

ODS LISTING

ODS OUTPUT statement:

MATCH_ALL

ODS TRACE statement

Other features:

PROC PRINT

PROC REG

Data set:

Model

Details

This example illustrates the differences in the data sets created by specifying the MATCH_ALL option and by not specifying the MATCH_ALL option. The first program creates a merged data set by specifying the MATCH_ALL option. The second program creates a merged data set without specifying the MATCH_ALL option.

The data sets that are printed are parameter estimates that PROC REG generates. The PROC REG procedure is part of SAS/STAT software.
**Note:** This example uses filenames that might not be valid in all operating environments. To successfully run the example in your operating environment, you might need to change the file specifications. See Appendix 4, “ODS HTML Statements for Running Examples in Different Operating Environments,” on page 1113.

**Program 1**

```sas
ods output SelectionSummary(match_all=list) = summary;
title1 'Using the MATCH_ALL Option Produces Two Data Sets With Different Columns';
ods trace on;
proc reg data=model;
  model r33=a b r4 r8 c d e r23 r24 r29/ selection=forward
      sle=.5 maxstep=3;
  model r33=a b r4 r8 c d e r23 r24 r29/ selection=backward
      sls=0.05 maxstep=3;
run;
ods trace off;
title2 'The First Data Set Has the VARENTERED Column';
proc print data=summary;
run;
title1;
title2 'The Second Data Set Has the VAREMOVED Column';
proc print data=summary1;
run;
data summarym;
  set &list;
run;
title1;
title2 'The Merged Data Set Has Both Columns';
proc print data=summarym;
run;
```

**Program Description**

**Prepare a SAS data set to be created.** The ODS OUTPUT statement opens the OUTPUT destination. By default, the list for the OUTPUT destination is EXCLUDE ALL. This ODS OUTPUT statement puts SelectionSummary in the selection list for the destination. The MATCH_ALL option produces a SAS data set for each instance of SelectionSummary. The name of the first data set is Summary, and the name of the second data set is Summary1. ODS stores a list of these names in the macro variable list. This variable is used later in the example to combine the data sets.

```sas
ods output SelectionSummary(match_all=list) = summary;
title1 'Using the MATCH_ALL Option Produces Two Data Sets With Different Columns';
```

**Create the output objects and view a record of them in the log.** PROC REG creates the output objects. The ODS TRACE statement writes to the SAS log a record of each output object that is created. The ODS TRACE OFF statement represses the printing of the records.

```sas
ods trace on;
proc reg data=model;
  model r33=a b r4 r8 c d e r23 r24 r29/ selection=forward
```
Print the reports. The PROC PRINT steps print the data sets that ODS created from PROC REG. The output from these steps is sent to the HTML destination.

```
title2 'The First Data Set Has the VARENTERED Column';
proc print data=summary;
run;
title1;
title2 'The Second Data Set Has the VARREMOVED Column';
proc print data=summary1;
run;
```

Create a data set that contains all of the data sets. The data set SummaryM combines all the data sets that were created by the ODS OUTPUT statement. The macro variable &list contains the list of data set names.

```
data summarym;
   set &list;
run;
```

Print the merged report and specify the title. The PROC PRINT step prints the merged data set created from the DATA step. The output from this step is sent to the HTML destination. The TITLE1 statement cancels the first title, and the TITLE2 statements specify a new title for the output.

```
title1;
title2 'The Merged Data Set Has Both Columns';
proc print data=summarym;
run;
```

HTML Output

The first data set created when using the MATCH_ALL option: This HTML output contains a printed report of the Summary data set created by the ODS OUTPUT statement with the MATCH_ALL option specified. It has no VARREMOVED column.

The second data set created when using the MATCH_ALL option: This HTML output contains a printed report of the Summary1 data set created by the ODS OUTPUT statement with the MATCH_ALL option specified. It has no VARENTERED column.

The merged data set created when using the MATCH_ALL option: This HTML output contains a printed report of the SummaryM data set created by the ODS OUTPUT statement with the MATCH_ALL option specified. This is the data set created from Summary and Summary1. It contains both the VARENTERED and VARREMOVED columns.
Output 6.53 Three Data Sets Created When Using the MATCH_ALL Option

Program 2

ods output SelectionSummary=summary;
title1 'Without the MATCH_ALL Option, ODS Produces a Single Data Set With All Of the Columns';
ods trace on;
proc reg data=model;
  model r33=a b r4 r8 c d e r23 r24 r29/ selection=forward
     sles=.5 maxstep=3;
  model r33=a b r4 r8 c d e r23 r24 r29/ selection=backward
     sls=0.05 maxstep=3;
run;
ods trace off;
proc print data=summary;
run;

Program Description

Prepare a SAS data set to be created. The ODS OUTPUT statement opens the OUTPUT destination and creates the SAS data set Summary. Because the MATCH_ALL option is not specified, ODS creates one data set that contains all instances of the output object SelectionSummary.

Create the output objects and view a record of them in the log. PROC REG creates the output objects. The ODS TRACE statement writes to the SAS log a record of each output object that is created. The ODS TRACE OFF statement represses the printing of the records.
ods trace on;
proc reg data=model;
  model r33=a b r4 r8 c d e r23 r24 r29/ selection=forward
   sls=.5 maxstep=3;
  model r33=a b r4 r8 c d e r23 r29/ selection=backward
   sls=0.05 maxstep=3;
run;
ods trace off;

Print the combined data set. The PROC PRINT step prints the merged data set created by ODS. The output from this step is sent to the HTML destination.

proc print data=summary;
run;

HTML Output
This HTML output contains a printed report of the Summary data set created by the ODS OUTPUT statement without the MATCH_ALL option specified. Note that to merge data sets, you do not have to specify the MATCH_ALL option.

Output 6.54 Using the ODS OUTPUT Statement without the MATCH_ALL Option to Combine Data Sets

Example 4: Customizing BY Lines
Features:
- ODS PDF statement option
  - FILE=
- ODS HTML
- ODS LISTING statement
- ODS OUTPUT statement
- PROC DOCUMENT

Other features:
- OPTIONS statement
- PROC PRINT
- PROC SORT
- PROC TABULATE
- DATA step
- MACRO language
Details

The use of BY variables sometimes results in more text than is desired in the tables of contents. The following example shows the use of ODS, PROC DOCUMENT, and the MACRO language to do the following:

- automate the labeling of the BYLINE entries from “By variable=By value” to “By Value”
- compress the table of contents entries to show one node for each table

For complete documentation about the MACRO language, see *SAS Macro Language: Reference*. For documentation about the DOCUMENT procedure, see “The DOCUMENT Procedure” in *SAS Output Delivery System: Procedures Guide*.

Step 1: Create an ODS Document

```sas
ods html close;
ods pdf file="DefaultTOC.pdf";
proc sort data=sashelp.cars out=cars;
  by origin;
run;
ods document name=Reorder(write);
proc tabulate data=cars;
  title "#byvall1";
  by origin;
  class cylinders;
  var mpg_highway;
  table cylinders, mpg_highway*mean;
run;
ods _all_ close;
```

Program Description

The first step is to create an ODS document to store your PROC TABULATE output. You only have to create the procedure output one time and save it in an ODS document. This document, named Reorder, is used to manipulate the output from this point on.

**Close the HTML destination and open the PDF destination.** The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources. If the destination were left open, then ODS would produce both HTML and PDF output. The ODS PDF statement opens the PDF destination and names the file the PDF output is written to.

```sas
ods html close;
ods pdf file="DefaultTOC.pdf";
```

**Sort the data set Sashelp.Cars by the variable ORIGIN**

```sas
proc sort data=sashelp.cars out=cars;
  by origin;
run;
```

**Create the ODS document Reorder.** The ODS DOCUMENT statement opens the DOCUMENT destination and specifies the name of a new document, Reorder. Reorder
contains the PROC TABULATE output objects. This example uses Reorder in the new step to manipulate the TOC entries using PROC DOCUMENT.

```r
ods document name=Reorder(write);

Create the TABULATE procedure output.

proc tabulate data=cars;
title "#byval1";
  by origin;
  class cylinders;
  var mpg_highway;
  table cylinders, mpg_highway*mean;
run;
```

Close all open destinations.

```r
ods _all_ close;
```

The following output shows the Bookmark list in the PDF file. The same text will show in the printable table of contents, if CONTENTS=YES is specified.

**Output 6.55  Default Table of Contents**

**Step 2: Create a DATA Set**

```r
ods output properties=Props;

proc document name=reorder;
  list / levels=all;
  run;
  quit;
```

```r
ods listing;
  proc print data=Props;
  run;
  ods listing close;
```

**Program Description**

Next, move the contents of the Reorder document into a DATA set. This enables you to automate the logic in step 3.
Create the Props data set. The ODS OUTPUT statement creates a data set name Props.

```ods output properties=Props;```

List the contents of the REORDER document and save them into the Props data set. The PROC DOCUMENT statement opens the document Reorder. The LIST statement lists the contents of Reorder. The contents are saved in the data set Props.

```proc document name=reorder;
list / levels=all;
run;
quit;```

Print the Props data set to see the document level names. At this point you might want to look at the contents of the Props data set. You can use the PROC PRINT and ODS LISTING statement to view the document level names.

```ods listing;
proc print data=props;
run;
ods listing close;```

Notice that some of the levels are directories (Dir) and some are tables (Table).

Output 6.56  Document Level Names in LISTING Output

<table>
<thead>
<tr>
<th>Obs</th>
<th>Path</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tabulate#1</td>
<td>Dir</td>
</tr>
<tr>
<td>2</td>
<td>Tabulate#1\ByGroup1#1</td>
<td>Dir</td>
</tr>
<tr>
<td>3</td>
<td>Tabulate#1\ByGroup1#1\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>4</td>
<td>Tabulate#1\ByGroup1#1\Report#1\Table1</td>
<td>Table</td>
</tr>
<tr>
<td>5</td>
<td>Tabulate#1\ByGroup2#1</td>
<td>Dir</td>
</tr>
<tr>
<td>6</td>
<td>Tabulate#1\ByGroup2#1\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>7</td>
<td>Tabulate#1\ByGroup2#1\Report#1\Table1</td>
<td>Table</td>
</tr>
<tr>
<td>8</td>
<td>Tabulate#1\ByGroup3#1</td>
<td>Dir</td>
</tr>
<tr>
<td>9</td>
<td>Tabulate#1\ByGroup3#1\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>10</td>
<td>Tabulate#1\ByGroup3#1\Report#1\Table1</td>
<td>Table</td>
</tr>
</tbody>
</table>

Step 3: Customize the Table of Contents

```data _null_;
set props end=last;
  if type in('Table') then do;
    count+1;
    call symputx('patht'||trim(left(count)),path);
  end;
call symputx('total',count);
run;

%macro order;
proc document name=work.reorder;
  %do i=1 %to &total;
    setlabel &&patht&&i "#byval1";
    move &&patht&&i to ^;
  %end;
%end;```
Program Description

Now you can start modifying your table of contents. PROC DOCUMENT syntax changes the text and structure of the table of contents. The macro language enables you to automate the process.

**Place the table entries into macro variables.** This DATA step selects only the entries labeled Table. It then places path values into the macro variables PathtN and Total. These macro variables are used in a later %DO loop with macro logic.

```sas
data _null_;
set props end=last;
  if type in("Table") then do;
    count+1;
    call symputx('patht'||trim(left(count)),path);
  end;
  call symputx('total',count);
run;
```

**Reorder and rename the table of contents, and begin the macro definition.** The %MACRO statement creates the macro definition name Order. The MOVE statement moves the contents of the variable Path to the current working directory. This creates one node for each table in the table of contents. The SETLABEL statement renames the node entries to the name of the By variable.

```sas
%macro order;
proc document name=work.reorder;
  %do i=1 %to &total;
    setlabel &&patht&&i "#byval1";
    move &&patht&&i to ^;
  %end;
%mend;
```

**Replay the output to the ODS PDF destination.** Create the PDF output by replaying the output to the PDF destination.

```sas
ods pdf file="replayed.pdf";
  replay;
run;
ods pdf close;
```
Show the modified document entries in the LISTING destination and end the macro definition. You can view all of the modified entries using the LIST statement and the ODS LISTING destination. The %MEND statement ends the macro definition.

```sas
ods listing;
   list / levels=all;
run;
ods listing close;
quit;
%mend;
```

Invoke the macro %ORDER. Now that your program is stored in a macro, invoke the macro %ORDER in order to create the output.

```
%order
```

Output 6.57  Customized BY Line in PDF Output

Output 6.58  Modified Document Entries

<table>
<thead>
<tr>
<th>Obs</th>
<th>Template</th>
<th>Label</th>
<th>Page Break</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Tabulate Procedure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Origin-Asia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Cross-tabular summary report</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Origin-Europe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Cross-tabular summary report</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Origin-USA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Cross-tabular summary report</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>*byvall</td>
<td></td>
<td>Before</td>
</tr>
<tr>
<td>9</td>
<td>*byvall</td>
<td></td>
<td>Before</td>
</tr>
<tr>
<td>10</td>
<td>*byvall</td>
<td></td>
<td>Before</td>
</tr>
</tbody>
</table>

Number of levels: All

<table>
<thead>
<tr>
<th>Obs</th>
<th>Path</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>\Tabulate#1</td>
<td>Dir</td>
</tr>
<tr>
<td>2</td>
<td>\Tabulate#1\ByGroup#1</td>
<td>Dir</td>
</tr>
<tr>
<td>3</td>
<td>\Tabulate#1\ByGroup#1\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>4</td>
<td>\Tabulate#1\ByGroup#1\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>5</td>
<td>\Tabulate#1\ByGroup#1\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>6</td>
<td>\Tabulate#1\ByGroup#1\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>7</td>
<td>\Tabulate#1\ByGroup#1\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>8</td>
<td>Table#1</td>
<td>Table</td>
</tr>
<tr>
<td>9</td>
<td>Table#2</td>
<td>Table</td>
</tr>
<tr>
<td>10</td>
<td>Table#3</td>
<td>Table</td>
</tr>
</tbody>
</table>

ODS PACKAGE Statement

Opens, adds to, publishes, or closes one SAS Output Delivery System (ODS) package object.
Valid in: Anywhere
Category: Data Access
Requirement: The destination must specify the PACKAGE option to connect with the package.

Syntax

```
ODS PACKAGE (<name>) OPEN <options> ;
ODS PACKAGE (<name>) PUBLISH
    transport PROPERTIES(transport-property-1="value-1" ...transport-property-n="value-n");
ODS PACKAGE (<name>) ADD FILE=file-specification | DATA=member-specification
    MIMETYPE="string" <PATH=path-specification> <options> ;
ODS PACKAGE (<name>) CLOSE <CLEAR> ;
```

Required Arguments

ADD
adds a file or data set to an ODS package using the specified Multipurpose Internet Mail Extensions (MIME) type.

Requirement When using the ADD argument, you must also use the MIMETYPE=, FILE=, or DATA= arguments to specify a file or data set and a MIME type.

```
FILE="file-specification" <TEXT | BINARY>
specifies the file that you want to add to an ODS package.

file-specification
    specifies one of the following:
    
    external-file
        is the name of an external file to add.
        
        Requirement You must enclose external-file in quotation marks.
    
    fileref
        is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.
    
    TEXT
        specifies that the file is a text file.
    
    BINARY
        specifies that the file is a binary file.
```

Default If you do not specify the TEXT or BINARY values, then the file is added as binary, unless it is a text file. Text files are added as text by default.

Restrictions You can use the FILE= argument only with the ADD argument.

You cannot add a file and a data set to an ODS package.
Example  Use the following statement to add the Test.SAS file as plain text to the ODS package directory SAS:
ods package add file="test.sas" mimtype="text/plain" path="sas/";

**DATA=**<em>member-specification</em>
specifies the data set that you want to add to an ODS package. <em>member-specification</em> can be in the form <em>libname.membername</em> or <em>membername</em>.

**Restrictions**  You can use the DATA= argument only with the ADD argument.
You cannot add a file and a data set to an ODS package.

**MIMETYPE="string"**
specifies the Multipurpose Internet Mail Extensions (MIME) type for the file or data set that you are adding to an ODS package.

**Restriction**  You can use the MIMETYPE= argument only with the ADD argument.

**OPEN**
creates the ODS package object to which the ODS destinations can connect. The ODS package object holds the package metadata and tracks the locations of any files that are added to the package metadata.

Example  The following ODS PACKAGE statement opens an unnamed package with an abstract and a description:
ods package open abstract="this is my abstract" description="this is description";

**PUBLISH**
builds the ODS package and sends it to the chosen delivery transport.

**CLOSE**
deletes the package object. As long as you have not closed a package, you can publish it as many ways and times as you want.

**Tip**  Use the CLEAR option to remove files that have been added to the package.

**transport**
specifies the deliver transport to use with the PUBLISH action. <em>transport</em> can be one of the following:

**ARCHIVE PROPERTIES**(transport-property-1="value-1"… transport-property-n="value-n")
publishes a package to an archive. For a list of transport properties and their values, see the section on transport properties in <em>SAS Publishing Framework: Developer's Guide</em>.

Example  The following statement publishes an ODS package to the archive Test.spk:
ods package publish archive properties(archive_path="/" archive_name="test.spk");

**EMAIL PROPERTIES**(transport-property-1="value-1"… transport-property-n="value-n") ADDRESSES("e-mail-address-1"… "e-mail-address-n")
publishes a package to one or more e-mail addresses. For a list of transport properties and their values, see the section on transport properties in <em>SAS Publishing Framework: Developer's Guide</em>. 
Example  The following statement publishes an ODS package to the e-mail addresses your.email@company.com and your.second.email@company.com:

```
ods package publish email addresses("your.email@company.com"
   "your.second.email@company.com")
   properties(archive_name="testPackage" archive_path="./");
```

```
QUEUE PROPERTIES(transport-property-1="value-1" . . . transport-property-n="value-n") QUEUES("queue-1" . . . "queue-n")
```

publishes a package to one or more message queues. For a list of transport properties and their values, see the section on transport properties in *SAS Publishing Framework: Developer's Guide*.

```
SUBSCRIBERS PROPERTIES(transport-property-1="value-1" . . . transport-property-n="value-n")
```

publishes a package to subscribers who are associated with the specified channel. For a list of transport properties and their values, see the section on transport properties in *SAS Publishing Framework: Developer's Guide*.

```
WEBDAV PROPERTIES(transport-property-1="value-1" . . . transport-property-n="value-n")
```

publishes a package to a WebDAV-compliant server. For a list of transport properties and their values, see the section on transport properties in *SAS Publishing Framework: Developer's Guide*.

**Optional Arguments**

**ABSTRACT=string**

specifies a string for the abstract metadata of the package or file.

Restriction  You can use the ABSTRACT= option only with the ADD or OPEN arguments.

Example  The following ODS PACKAGE statement opens an unnamed package with an abstract and a description:

```
ods package open abstract="this is my abstract" description="this is description";
```

**CLEAR**

specifies that all files that were automatically added to the package will be removed from the location to which ODS wrote them.

Restriction  You can use the CLEAR option only with the CLOSE argument.

**DESCRIPTION=string**

specifies a string for the description metadata for the package or file.

Restriction  You can use the DESCRIPTION= option only with the ADD or OPEN arguments.

Example  The following ODS PACKAGE statement opens an unnamed package with an abstract and a description:

```
ods package open abstract="this is my abstract" description="this is description";
```

**EXPIRATION= <'expiration-date'>**

specifies an expiration date for the package. The date must be a SAS date value.
Restriction You can use the EXPIRATION= option only with the PUBLISH or OPEN arguments.

Requirement expiration-date must be enclosed in quotation marks.

\(<(name)\>

specifies the name of a package. Naming a package enables you to open more than one package at a time. Each destination can connect with any package by specifying the package name in the same way.

Restriction The NAMEVALUE= option can be used only with the OPEN argument.

Requirements You must place name directly after the PACKAGE keyword in the ODS PACKAGE statement.

name must be enclosed in parentheses.

NAMEVALUE="\(<name-1=\"value-1\\\" \ldots name-n=\"value-n\\\">\"

specifies a string of name/value pairs for the name/value metadata on the package or file.

Restriction The NAMEVALUE= option can be used only with the ADD or OPEN arguments.

PATH="path-specification"

places the file or data set at the specified pathname within an ODS package.

Restriction You can use the PATH= option only with the ADD argument.

Example Use the following statement to add the Test.SAS file as plain text to the ODS package directory SAS:

```
ods package add file="test.sas" mimetype="text/plain" path="sas/";
```

Details

A package is a container for digital content that is generated or collected for delivery to a consumer. ODS packages allow ODS destinations to use the SAS Publishing Framework. An ODS package is an object that contains output files and data sets that are associated with any open ODS destinations. ODS packages hold the package metadata and track the output from any active destinations that connect to it. After the destinations are closed, the package can be published to any of the publish destinations. You can continue to use the package, or you can close it. A package remains active until explicitly closed.

Examples

Example 1: Creating an ODS Package

The following example creates a simple ODS package. The package is created in your default directory, if you do not specify a different directory.

Program

```
goptions dev=gif xpixels=480 ypixels=320;
```
Example 1: Creating an ODS Package

ods package open;
ods html package;

proc gplot data=sashelp.class;
   plot height*weight;
   by name;
run;
quit;
ods html close;

ods package publish archive properties
   (archive_name="SimpleExample.zip" archive_path="./");
ods package close;
ods html;

Program Description

Specify graphical options with the GOPTIONS statement.

goptions dev=gif xpixels=480 ypixels=320;

Open an ODS package and specify that HTML output be added to the package. The ODS PACKAGE statement opens an ODS package with no name. The PACKAGE option specified by the ODS HTML statement specifies that output from the HTML destination be added to the package.

ods package open;
ods html package;

Create graphical output with the GPLOT statement and close the HTML destination.

proc gplot data=sashelp.class;
   plot height*weight;
   by name;
run;
quit;
ods html close;

Build the package and publish it to an archive. The PUBLISH option builds the ODS package. The ARCHIVE property publishes the package to the archive named SimpleExample.zip in the default directory.

ods package publish archive properties
   (archive_name="SimpleExample.zip" archive_path="./");
ods package close;
ods html;
Simple ODS Package

Example 2: Listing Package Contents with the ODS DOCUMENT Statement

In the following program, PROC DOCUMENT imports the archive SimpleExample.zip into a PROC DOCUMENT package named myPackage. You can then use PROC DOCUMENT to list the contents and details of the package.

Program

```sas
proc document name=archive;
import archive="SimpleExample.zip" to myPackage;
list/levels=all;
run;
dir myPackage;
list 'sashtml.htm'n/details;
run;
quit;
```

Program Description

Create an ODS document and import SimpleExample.zip. The DOCUMENT procedure creates the ODS document Archive. The IMPORT TO statement imports SimpleExample.zip into the package myPackage. The LIST statement lists all of the levels of Archive.

```sas
proc document name=archive;
import archive="SimpleExample.zip" to myPackage;
list/levels=all;
```
Example 2: Listing Package Contents with the ODS DOCUMENT Statement

run;

List the details of the file SasHtml.htm. The DIR statement changes the directory to myPackage. The LIST statement lists the details of SasHtml.htm.

dir myPackage;
list 'sashtml.htm'n/details;
run;
quit;
### Program Output

**Output 6.59  Listing of Work.Archive and Details of HTM File**

#### The SAS System

<table>
<thead>
<tr>
<th>Obs</th>
<th>Path</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>myPackage#1</code></td>
<td>Dir</td>
</tr>
<tr>
<td>2</td>
<td><code>myPackage#1\'sashtml.htm'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>3</td>
<td><code>myPackage#1\'gplot1.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>4</td>
<td><code>myPackage#1\'gplot2.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>5</td>
<td><code>myPackage#1\'gplot3.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>6</td>
<td><code>myPackage#1\'gplot4.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>7</td>
<td><code>myPackage#1\'gplot5.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>8</td>
<td><code>myPackage#1\'gplot6.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>9</td>
<td><code>myPackage#1\'gplot7.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>10</td>
<td><code>myPackage#1\'gplot8.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>11</td>
<td><code>myPackage#1\'gplot9.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>12</td>
<td><code>myPackage#1\'gplot10.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>13</td>
<td><code>myPackage#1\'gplot11.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>14</td>
<td><code>myPackage#1\'gplot12.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>15</td>
<td><code>myPackage#1\'gplot13.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>16</td>
<td><code>myPackage#1\'gplot14.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>17</td>
<td><code>myPackage#1\'gplot15.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>18</td>
<td><code>myPackage#1\'gplot16.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>19</td>
<td><code>myPackage#1\'gplot17.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>20</td>
<td><code>myPackage#1\'gplot18.gif'n#1</code></td>
<td>File</td>
</tr>
<tr>
<td>21</td>
<td><code>myPackage#1\'gplot19.gif'n#1</code></td>
<td>File</td>
</tr>
</tbody>
</table>

#### The SAS System

<table>
<thead>
<tr>
<th>Path</th>
<th>Type</th>
<th>Size in Bytes</th>
<th>Created</th>
<th>Modified</th>
<th>Symbolic Link</th>
<th>Template</th>
<th>Label</th>
<th>Page Break</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>myPackage#1\'sashtml.htm'n#1</code></td>
<td>File</td>
<td>4984</td>
<td>28JAN2011:11:50:10</td>
<td>28JAN2011:11:50:10</td>
<td>tags-HTML4, Proc, Gplot</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**See Also**

ODS PATH Statement

Specifies locations to write to or read from when creating or using PROC TEMPLATE definitions and the order in which to search for them.

- **Valid in:** Anywhere
- **Category:** ODS: Output Control
- **Tips:** This statement overrides the ODS PATH statement for the duration of a PROC TEMPLATE step.
  
  You can use the SYSODSPATH automatic macro variable to store the current ODS path. For information about the SYSODSPATH macro variable, see *SAS Macro Language: Reference*.

- **Example:** "Creating a Customized Crosstabulation Table Template with No Legend" in *SAS Output Delivery System: Procedures Guide*

### Syntax

**PATH** *(APPEND) | (PREPEND) | (REMOVE) > location(s);*

**PATH** *path-argument;*

### Required Arguments

**location(s)**

specifies one or more locations to write to or read from when creating or using PROC TEMPLATE definitions and the order in which to search for them. ODS searches the locations in the order in which they appear on the statement. It uses the first definition that it finds that has the appropriate access mode (Read, Write, or Update) set.

Each *location* has the following form:

*<libref.:item-store>* *(READ | UPDATE | WRITE)*

*<libref.:item-store>* identifies an item store to read from, to write to, or to update. If an item store does not already exist, then the ODS PATH statement will create it.

*(READ | UPDATE | WRITE)* specifies the access mode for the definition. The access mode is one of the following:

- **READ** provides Read-Only access.
- **WRITE** provides Write access (always creating a new template store) as well as Read access.
- **UPDATE** provides Update access (creating a new template store only if the specified one does not exist) as well as Read access.

**Default** READ
Default  The general default path is as follows:

1. Sasuser.Templat (UPDATE)
2. Sashelp.Tmplmst (READ)

If you have the RSASUSER SAS system option specified, the default path is as follows:

1. Work.Templat (UPDATE).
2. Sasuser.Templat (READ).
3. Sashelp.Tmplmst (READ). Note that SAS stores all the definitions that it provides in Sashelp.Tmplmst.

For more information, see “RSASUSER System Option” in SAS System Options: Reference.

Interaction  You can use the PATH statement in a PROC TEMPLATE step to temporarily override the ODS PATH statement. For more information, see “PATH Statement” in SAS Output Delivery System: Procedures Guide.

Tip  To ignore all definitions that you create, keep them in their own item stores instead of the list of item stores that ODS searches.

path-argument specifies the setting or displaying of the ODS path.

path-argument can be one of the following:

RESET  sets the ODS path to the default settings Sasuser.Templat (UPDATE) and Sashelp.Tmplmst (READ).

SHOW  displays the current ODS path.

VERIFY  sets the ODS path to include only templates supplied by SAS. VERIFY is the same as specifying ODS PATH Sashelp.Tmplmst (READ).

Optional Argument  

(APPEND | PREPEND | REMOVE )

adds or removes one or more locations to a path.

APPEND  adds one or more locations to the end of a path. When you append a location to a path, all duplicate instances (same name and same permissions) of that item store are removed from the path. Only the last item store with the same name and permissions are kept.

PREPEND  adds one or more locations to the beginning of a path. When you prepend a location with Update permissions to a path, all duplicate instances (same name
and same permissions) of that item store are removed from the path. Only the first item store with the same name and permissions are kept.

**REMOVE**
removes one or more locations from a path.

**Default**
If you do not specify an APPEND, PREPEND, or REMOVE option, then the ODS PATH statement overwrites the complete path.

---

**ODS PCL Statement**

Opens, manages, or closes the PCL destination, which produces printable output for PCL (HP LaserJet) files.

**Valid in:** Anywhere

**Category:** ODS: Third-Party Formatted

**Default:** The default style for PRINTER destinations is Pearl.

**Note:** By default, the ODS PCL statement creates Scalable Vector Graphics. Scalable Vector Graphics (SVG) is an XML language for describing two-dimensional vector graphics. For information about scalable vector graphics, see “Using SVG Graphics” in SAS/GRAPH: Reference.

**Syntax**

```
ODS PCL (<ID=> identifier) < action> ;
ODS PCL (<ID=> identifier) <option(s)> ;
```

**Summary of Optional Arguments**

- `<ID=> identifier>`
  Open multiple instances of the same destination at the same time

- `BOX_SIZING=(CONTENT_BOX | BORDER_BOX)`
  Specify how to measure the width of cells. Use to override the default value of BOX_SIZING for a destination

- `CLOSE`
  Close the destination and the file that is associated with it

- `COLOR=FULL | GRAY | MONO | NO | YES`
  Apply a specified color scheme to your output

- `CLOCKS=n`
  Specify the number of columns to create on each page of output

- `CSSSTYLE='file-specification'<(media-type-1…media-type-10)>`
  Specify a cascading style sheet to apply to your output

- `DOM="external-file">`
  Specify that the ODS document object model is written to the SAS log or to an external file.

- `DPI=`
  Specify the image resolution in dots per inch for output images

- `EXCLUDE exclusion(s) | ALL | NONE`
  Exclude output objects from the destination
FILE='external-file' | fileref
Specify the output file.

GFOOTNOTE | NOGFOOTNOTE
Specify the location where footnotes are printed in the graphics output

GTITLE | NOGTITLE
Control the location where titles are printed in the graphics output

NEWFILE= starting-point
Create a new file at the specified starting-point

PACKAGE <package-name>
Specify that the output from the destination be added to an ODS package

SELECT selection(s) | ALL | NONE
Select output objects for the destination

SHOW
Write to the SAS log the current selection or exclusion list for the destination

STARTPAGE=NEVER | NO | NOW | YES | BYGROUP
Control page breaks

STYLE=style-template
Specify the style template to use in writing the PDF output

TEXT=’text-string’
Insert text into your output

UNIFORM
For multi-page tables, provide uniformity from page to page within a single table

**Without Arguments**
If you use the ODS PCL statement without an action or options, then it opens the PCL destination and creates PCL output.

**Actions**
The following actions are available for the ODS PCL statement:

**CLOSE**
closes the destination and any files that are associated with it.

**Tip** When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

**EXCLUDE exclusion(s) | ALL | NONE**
excludes one or more output objects from the destination.

Default | NONE
Restriction | A destination must be open for this action to take effect.
See | “ODS EXCLUDE Statement” on page 321

**SELECT selection(s) | ALL | NONE**
selects output objects for the specified destination.

Default | ALL
Restriction | A destination must be open for this action to take effect.
See | “ODS SELECT Statement ” on page 758
SHOW writes the current selection list or exclusion list for the destination to the SAS log.

Restriction The destination must be open for this action to take effect.

Tip If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

See “ODS SHOW Statement” on page 771

Optional Arguments

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing.

COLOR=FULL | GRAY | MONO | NO | YES applies the specified color scheme to your output.

FULL
creates full color output for both text and graphics.

GRAY
creates gray scale output for both text and graphics.

Alias GREY

MONO
creates monochromatic output for both text and graphics.

Alias BW

NO
does not use all the color information that the style template provides. If you specify COLOR=NO, then the destination does this:

• generates black and white output
• creates all text and rules in black
• sets the SAS/GRAPH device to produce SAS/GRAPH output in gray scale
• ignores specifications for a background color from the style template except for the purposes of determining whether to print rules for the table

YES
uses all the color information that a style template provides, including background color. To print in color, you must also do the following:

• use a printer that is capable of printing in color.
• use the COLORPRINTING SAS system option. For information about the COLORPRINTING system option, see SAS System Options: Reference.

Default YES
Tip If you choose color output for a printer that does not support color, then your output might be difficult to read.

**COLUMNS=n**
specifies the number of columns to create on each page of output.

\[n\]
is the number columns per page.

Default 1

**CSSSTYLE='file-specification'\(<(media-type-1<..media-type-10>)>\)**

specifies a cascading style sheet to apply to your output.

*file-specification*
specifies a file, fileref, or URL that contains CSS code.

*file-specification* is one of the following:

"external-file"
is the name of the external file.

Requirement You must enclose external-file in quotation marks.

*fileref* is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

"URL"
is a URL to an external file.

Requirement You must enclose URL in quotation marks.

\(<(media-type-1<..media-type-10>)>\)
specifies one or more media blocks that correspond to the type of media that your output is rendered on. CSS uses media type blocks to specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not contained in any media blocks. By using the media-type suboption, in addition to the general CSS code, you can import the section of a CSS file intended only for a specific media type.

Default If no media-type is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

Range You can specify up to ten different media types.

Requirements You must enclose media-type in parentheses.

Tip If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if
there is duplicate style information in different media blocks, then the styles from the last media block are used.

Restriction  The CSSSTYLE= option does not affect SAS/GRAPH output.

Requirement  CSS files must be written in the same type of CSS produced by the ODS HTML statement. To view the CSS code that ODS creates, you can do one of the following:
  • specify the ODS TRACE DOM statement
  • specify the DOM option

Interaction  If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

See  For complete documentation about ODS and Cascading Style Sheets, see SAS Output Delivery System: Advanced Topics

Example  “Example 6: Applying a CSS File to ODS Output” on page 527

DOM="external-file">
specifies that the ODS document object model is written to the SAS log or an external file.

external-file  is the name of an external output file.

Requirement  You must enclose external-file in quotation marks.

See  For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in SAS Output Delivery System: Advanced Topics.

DPI=
specifies the image resolution for output files.

Default  150

Restriction  The DPI= option takes effect only if specified at the opening of a file.

CAUTION  When using high DPI= or DPI_IMAGE= values (values over 600), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500 or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.

FILE='external-file' | fileref
specifies the output file.

external-file  is the name of an external file.

Requirement  You must enclose external-file in quotation marks.

fileref  is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.
Restriction

The FILE=fileref option cannot be used in conjunction with the NEWFILE= option.

See

For information about the FILENAME statement, see SAS Statements: Reference.

Default

If you do not specify an output file, then ODS writes to the file that is specified by two SAS system options. Use the SYSPRINT= system option if you are using the Windows operating environment and do not specify any of the following options: PCL, PDFMARK, POSTSCRIPT, PS, or SAS. Use the PRINTERPATH= system option in all other cases. If the system option does not specify a file, then ODS writes to the default printer. For more information, see the PRINTER= option.

Interaction

In an ODS printer family statement that refers to an open ODS PRINTER destination, the FILE= option forces ODS to close the destination and all files that are associated with it. ODS then opens a new instance of the destination. For more information, see “Opening and Closing the PRINTER Destination” on page 708.

See

For information about the FILENAME statement, see SAS Statements: Reference.

GFOOTNOTE | NOGFOOTNOTE

controls the location of the footnotes that are defined by the graphics program that generates the Printer output.

GFOOTNOTE

includes all of the currently defined footnotes within the graphics output.

NOGFOOTNOTE

prevents all of the currently defined footnotes from appearing in the graphics file. Instead, they become part of the Printer file.

Default

GFOOTNOTE

Restriction

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

See

For more information, see “Customizing Titles and Footnotes” on page 40.

GTITLE | NOGTITLE

controls the location of the titles that are defined by the graphics program that generates the Printer output.

GTITLE

includes all of the currently defined titles within the graphics output.

NOGTITLE

prevents all of the currently defined titles from appearing in the graphics output. Instead, the titles become part of the Printer file.

Default

GTITLE
Restriction: This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

See: For more information, see “Customizing Titles and Footnotes” on page 40.

(<ID=> identifier)

enables you to open multiple instances of the same destination at the same time. Each instance can have different options.

identifier can be numeric or can be a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numerals.

Restriction: If identifier is numeric, it must be a positive integer.

Requirement: The ID= option must be specified immediately after the destination name.

NEWFILE= starting-point

creates a new file at the specified starting-point.

starting-point is the location in the output where you want to create a new file.

ODS automatically names new files by incrementing the name of the file. In the following example, ODS names the first file REPORT.PS. Additional body files are named REPORT1.PS, REPORT2.PS, and so on.

Example:

FILE= 'REPORT.PS'

starting-point can be one of the following:

BYGROUP starts a new file for the results of each BY group.

NONE writes all output to the file that is currently open.

OUTPUT starts a new file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

Alias TABLE

PAGE starts a new file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.

PROC starts a body file each time you start a new procedure.

Default: NONE
Restrictions

The NEWFILE= option cannot be used in conjunction with the FILE=fileref option.

The NEWFILE= option cannot be used if you are sending output to a physical printer.

Tips

If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first file MAY5.PS. Additional body files are named MAY6.PS, MAY7.PS, and so on.

Example:

FILE= 'MAY5.PS'

PACKAGE <package-name>

specifies that the output from the destination be added to a package.

package-name

specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

See “ODS PACKAGE Statement ” on page 554

STARTPAGE=NEVER | NO | NOW | YES | BYGROUP

controls page breaks.

BYGROUP

specifies to insert page breaks after each BY group.

NEVER

specifies not to insert page breaks, even before graphics procedures.

CAUTION:

Each graph normally requires an entire page. The default behavior forces a new page after a graphics procedure. STARTPAGE=NEVER turns off that behavior, so specifying STARTPAGE= NEVER might cause graphics to overprint.

NO

specifies that no new pages be inserted at the beginning of each procedure, or within certain procedures, even if new pages are requested by the procedure code. A new page begins only when a page is filled or when you specify STARTPAGE=NOW.

CAUTION:

Each graph normally requires an entire page. The default behavior forces a new page after a graphics procedure, even if you use STARTPAGE=NO. STARTPAGE=NEVER turns off that behavior, so specifying STARTPAGE= NEVER might cause graphics to overprint.

Alias

OFF

Tip

When you specify STARTPAGE=NO, system titles and footnotes are still produced only at the top and bottom of each physical page, regardless of the setting of this option. Thus, some system titles and footnotes that you specify might not appear when this option is specified.
NOW
forces the immediate insertion of a new page.

Tip
This option is useful primarily when the current value of the
STARTPAGE= option is NO. Otherwise, each new procedure forces a
new page automatically.

Example
“Example 7: Toggling Page Breaks” on page 615

YES
inserts a new page at the beginning of each procedure, and within certain
procedures, as requested by the procedure code.

Alias ON

Default YES

STYLE=style-template
specifies the style template to use in writing the printer output.

Default If you do not specify a style template, then ODS uses the style template
that is specified in the SAS registry subkey: ODS ⇄ DESTINATIONS
⇌ PRINTER. By default, this value is Pearl for the PRINTER, PDF, PS
and PCL destinations.

Note If you are using SAS Studio, you do not need to specify the STYLE=
option. You can go to Preferences ⇄ Results and change the style from
the drop-down list for your selected destination.

See For a complete discussion of style templates, see “Working with Styles ”

For instructions on making your own user-defined style templates, see
“TEMPLATE Procedure: Creating a Style Template ” in SAS Output
Delivery System: Procedures Guide.

Examples “Example 3: Customizing the Table of Contents” on page 600

“Example 6: Adding Text That Imitates a System Title” on page 611

TEXT='text-string'
inserts a text string into your output.

text-string
is the text that you want to insert into your output.

Requirement You must enclose text-string in quotation marks.
Tip If you are submitting more than one procedure step and you do not specify the STARTPAGE=NO option, each procedure forces a new page before the output. Therefore, any text that you specify with TEXT= is on the same page as the previous procedure.

See “Example: Conditionally Excluding Output Objects and Sending Them to Different Output Destinations” on page 325

UNIFORM
for multiple page tables, ensures uniformity from page to page within a single table. When the UNIFORM option is in effect, ODS reads the entire table first, so that it can determine the column widths that are necessary to accommodate all the data. These column widths are applied to all pages of a multiple page table.

Default If you do not specify the UNIFORM option, then ODS prints a table one page at a time. This approach ensures that SAS does not run out of memory while processing very large tables. However, it can also mean that column widths vary from one page to the next.

Note With BY-group processing, SAS writes the results of each BY group to a separate table, so the output might not be uniform across BY groups.

Tip The UNIFORM option can cause SAS to run out of memory if you are printing a very large table. If this happens, then you can explicitly set the width of each of the columns in the table, and then print the table one page at a time. To do so, you must edit the table template that you use. For more information, see “What You Can Do with Table Templates” in SAS Output Delivery System: Procedures Guide.

Details

Opening and Closing the PCL Destination
You can modify an open PCL destination with many ODS PCL options. However, the FILE= and SAS options perform the following actions on an open PCL destination.

• close the open destination referred to in the ODS PCL statement
• close any files associated with the open PCL destination
• open a new instance of the PCL destination

If you use one of these options, it is best if you explicitly close the destination yourself.

The ODS Printer Family of Statements
The ODS PCL statement is part of the ODS printer family of statements. Statements in the printer family open the PCL, PDF, PRINTER, or PS destination, producing output that is suitable for a high-resolution printer. The ODS PDF, ODS PRINTER, and ODS PS statements are also members of the ODS printer family of statements.

See Also
• “The Third-Party Formatted Destinations” on page 35

Statements
• “ODS PDF Statement ” on page 575
ODS PDF Statement

Opens, manages, or closes the PDF destination, which produces PDF output, a form of output that is read by Adobe Acrobat and other applications.

Valid in: Anywhere
Category: ODS: Third-Party Formatted
Default: The default style for PRINTER destinations is Pearl.
Restriction: PDF does not support double-byte Type1 fonts.
Notes: By default, the ODS PDF statement creates Scalable Vector Graphics. Scalable Vector Graphics (SVG) is an XML language for describing two-dimensional vector graphics. For information about scalable vector graphics, see “Using SVG Graphics” in SAS/GRAPH: Reference.
You can add drill-down graphs in your PDF file. For detailed information about drill-down graphs and about writing graphs to a PDF file, refer to “Writing Your Graphs to a PDF File” in SAS/GRAPH: Reference.
Tips: The PDF driver that SAS uses does not recognize all Microsoft Windows fonts. You must enter any such fonts into the SAS registry in order for SAS to find them. See “The SAS Registry” in SAS Language Reference: Concepts.
If the orientation of a PDF document is changed after the PDF destination is opened and before the PDF destination is closed, any setting for margins is taken from the OPTIONS statement in place before the ODS PDF FILE= statement. If no OPTIONS statement is used to explicitly set the margins, the margin settings are retrieved from the SAS registry.
Example: For an example about using PROC TEMPLATE to customize titles and footnotes in PDF output, see “Enhancing Titles and Footnotes in PDF Output” in SAS Output Delivery System: Procedures Guide.

Syntax

ODS PDF (<ID=> identifier) < action> ;
ODS PDF (<ID=> identifier) <option(s)> ;

Summary of Optional Arguments

(<ID=> identifier)
Open multiple instances of the same destination at the same time
ANCHOR='anchor-name'
Specify the root name for the anchor tag that identifies each output object in the current file
AUTHOR='author-text'
Insert the text string that you specify as the author into the metadata of a file
BASE='base-text'
Specify a string to use as the first part of all references that ODS creates in the file
BOOKMARKGEN | NOBOOKMARKGEN | BOOKMARKGEN=
Control the generation of bookmarks in PDF and PS files

BOOKMARKLIST= HIDE | NONE | SHOW
Specify whether to generate and display the list of bookmarks for PDF and PS files

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
Specify how to measure the width of cells. Use to override the default value of BOX_SIZING for a destination

CLOSE
Close the destination and the file that is associated with it

COLOR=FULL | GRAY | MONO | NO | YES
Apply a specified color scheme to your output

COLUMNS=n
Specify the number of columns to create on each page of output

COMPRESS=n
Specify the compression of a PDF file. Compression reduces the size of the file

CONTENTS= NO | YES
Control the generation of a printable table of contents

CSSSTYLE='file-specification'<(media-type-1<…media-type-10>?)>
Specify a cascading style sheet to apply to your output

DOM<="external-file"
Specify that the ODS document object model is written to the SAS log or to an external file.

DPI=
Specify the image resolution in dots per inch for output images

EXCLUDE exclusion(s) | ALL | NONE
Exclude output objects from the destination

FILE='external-file' | fileref
Specify the output file.

GFOOTNOTE | NOGFOOTNOTE
Specify the location where footnotes are printed in the graphics output

GTITLE | NOGTITLE
Control the location where titles are printed in the graphics output

KEYWORDS='keywords-text'
Insert a string of keywords into the output file's metadata

NEWFILE= starting-point
Create a new file at the specified starting-point

NOTOC
Omit the table of contents (Bookmark list) that is produced by default when producing PDF or PDFMARK output

PACKAGE <package-name>
Specify that the output from the destination be added to an ODS package

PDFNOTE | NOPDFNOTE
Control whether notes are added to a PDF file for items that are associated with the FLYOVER= style attribute

PDFTOC=n
Control the level of the expansion of the table of contents in PDF documents

SELECT selection(s) | ALL | NONE
Select output objects for the destination
SHOW
Write to the SAS log the current selection or exclusion list for the destination

STARTPAGE=NEVER | NO | NOW | YES | BYGROUP
Control page breaks

STYLE=style-template
Specify the style template to use in writing the PDF output

SUBJECT='subject-text'
Insert the text string that you specify as the subject in the metadata of a file

TEXT='text-string'
Insert text into your output

TITLE='title-text'
Insert the text string that you specify as the title in the metadata of a file

UNIFORM
For multi-page tables, provide uniformity from page to page within a single table

**Without Arguments**
If you use the ODS PDF statement without an action or options, then it opens the PDF destination and creates PDF output.

**Actions**
The following actions are available for the ODS PDF statement:

**CLOSE**
closes the destination and any files that are associated with it.

*Tip* When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

EXCLUDE exclusion(s) | ALL | NONE
excludes one or more output objects from the destination.

Default none

Restriction A destination must be open for this action to take effect.

See “ODS EXCLUDE Statement” on page 321

SELECT selection(s) | ALL | NONE
selects output objects for the specified destination.

Default all

Restriction A destination must be open for this action to take effect.

See “ODS SELECT Statement” on page 758

SHOW
writes the current selection list or exclusion list for the destination to the SAS log.

Restriction The destination must be open for this action to take effect.

*Tip* If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For
information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

See “ODS SHOW Statement” on page 771

Optional Arguments

ANCHOR='anchor-name'
specifies the root name for the anchor tag that identifies each output object in the current file.

Each output object must have an anchor tag for the bookmarks to reference. The references are automatically created by ODS. These references, point to the name of an anchor. Therefore, each anchor name in a file must be unique.

anchor-name
is the root name for the anchor tag that identifies each output object in the current file.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR='TABULATE', then ODS names the first anchor tabulate. The second anchor is named tabulate1; the third is named tabulate2, and so on.

Requirement You must enclose anchor-name in quotation marks.

Alias NAMED_DEST= | BOOKMARK=

Restriction Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

Tips You can change anchor names as often as you want by submitting the ANCHOR= option in a valid statement anywhere in your program. After you have specified an anchor name, it remains in effect until you specify a new one.

Specifying new anchor names at various points in your program is useful when you want to link to specific parts of your PRINTER output. Because you can control where the anchor name changes, you know in advance what the anchor name is at those points.

AUTHOR= 'author-text'
inserts the text string that you specify as the author into the metadata of a file.

author-text
is the text in the metadata of an open file that indicates the author.

Restrictions Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

The AUTHOR= option takes effect only if specified at the opening of a file.

Requirement You must enclose author-text in quotation marks.
BASE='base-text'

specifies the text to use as the first part of all references that ODS creates in the output file.

base-text

is the text that ODS uses as the first part of all references that ODS creates in the file.

Consider this specification:

BASE='http://www.your-company.com/local-url/'

In this case, ODS creates references that begin with the string http://www.your-company.com/local-url/. The appropriate anchor-name completes the link.

Restriction  Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

Requirement  You must enclose base-text in quotation marks.

BOOKMARKLIST= HIDE | NONE | SHOW

specifies whether to generate and display the list of bookmarks for PDF and PS files.

HIDE

generates a list of bookmarks for your PDF and PS files. The bookmarks are not automatically displayed when you open the PDF and PS files.

NONE

specifies not to generate a list of bookmarks for your PDF and PS files.

Aliases  NO | OFF

NOBOOKMARKLIST is an alias for BOOKMARKLIST=NONE | NO | OFF.

SHOW

generates a list of bookmarks for your PDF and PS files. The bookmarks are automatically displayed when you open the PDF and PS files.

Aliases  YES | ON

BOOKMARKLIST is an alias for BOOKMARKLIST=SHOW | YES | ON.

Example  “Example 2: Creating a Printable Table of Contents” on page 598

Default  SHOW

Restrictions  This option can be set only when you first open the destination.

This option has an effect only when creating PDF, PDFMARK, PS output.

Interaction  The NOTOC option specifies BOOKMARKLIST= OFF and CONTENTS= OFF.

Note  The generation of the bookmarks is not affected by the setting of this option. Bookmarks are generated by the BOOKMARKGEN= option.
Example  “Example 2: Creating a Printable Table of Contents” on page 598

BOOKMARKGEN | NOBOOKMARKGEN | BOOKMARKGEN=
controls the generation of bookmarks in PDF and PS files.

BOOKMARKGEN
specifies to generate bookmarks in PDF and PS files.

BOOKMARKGEN=
controls the generation of bookmarks in PDF and PS files.

NO
specifies not to generate bookmarks in PDF and PS files.

Alias OFF

YES
specifies to generate bookmarks in PDF and PS files.

Alias ON

NOBOOKMARKGEN
specifies not to generate bookmarks in the PDF and PS files.

Default YES or BOOKMARKGEN

Interaction If you set BOOKMARKGEN=NO, then the BOOKMARKLIST option is set to NO also.

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing.

COLOR=FULL | GRAY | MONO | NO | YES
applies the specified color scheme to your output.

FULL
creates full color output for both text and graphics.

GRAY
creates gray scale output for both text and graphics.

Alias GREY

MONO
creates monochromatic output for both text and graphics.

Alias BW

NO
does not use all the color information that the style template provides. If you specify COLOR=NO, then the destination does this:

• generates black and white output
• creates all text and rules in black
• sets the SAS/GRAPH device to produce SAS/GRAPH output in gray scale
• ignores specifications for a background color from the style template except for the purposes of determining whether to print rules for the table

YES
uses all the color information that a style template provides, including background color. To print in color, you must also do the following:
• use a printer that is capable of printing in color.
• use the COLORPRINTING SAS system option. For information about the COLORPRINTING system option, see SAS System Options: Reference.

Default YES
Tip If you choose color output for a printer that does not support color, then your output might be difficult to read.

COLUMNS=n
specifies the number of columns to create on each page of output.

\(n\)

is the number columns per page.

Default 1

COMPRESS=n
controls the compression of a PDF file. Compression reduces the size of the file.

\(n\)

specifies the level of compression. The larger the number, the greater the compression. For example, \(n=0\) is completely uncompressed, and \(n=9\) is the maximum compression level.

Default 6
Range 0–9
Restrictions Use this option only with the ODS PDF statement and the ODS PRINTER statement with the PDF option specified. PostScript output cannot be compressed.

The COMPRESS= option takes effect only if specified at the opening of a file.

Interactions The COMPRESS= option overrides the DEFLATION system option. First, the DEFLATION system option checked. Next, the ODS PDF statement COMPRESS= option is checked. If the COMPRESS= option is specified, that value is used regardless of the value specified for the DEFLATION system option. For more information, see the DEFLATION option.

The COMPRESS= option overrides the UPRINTCOMPRESSION option. If COMPRESS= is specified, the UPRINTCOMPRESSION system option is then queried. If the system option is off, it is turned on for this one PDF statement and the PDF file is compressed. When compression is complete, the UPRINTCOMPRESSION system option is again enabled for all other files to use. For more information, see the UPRINTCOMPRESSION system option.
CONTENTS= NO | YES
controls the generation of a printable table of contents.

NO
does not generate a printable table of contents.

Alias
NOCONTENTS is an alias for CONTENTS=NO.

YES
generates a printable table of contents.

Alias
CONTENTS is an alias for CONTENTS=YES.

Default
NO

Example
“Example 2: Creating a Printable Table of Contents” on page 598

CSSSTYLE= '<file-specification'><(media-type-1<.. media-type-10>)>'
specifies a cascading style sheet to apply to your output.

file-specification
specifies a file, fileref, or URL that contains CSS code.

file-specification is one of the following:

"external-file"
is the name of the external file.

Requirement
You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the
FILENAME statement to assign a fileref.

See
For information about the FILENAME statement, see “FILENAME
Statement” in SAS Statements: Reference.

"URL"
is a URL to an external file.

Requirement
You must enclose URL in quotation marks.

(media-type-1<.. media-type-10>)
specifies one or more media blocks that correspond to the type of media that your
output is rendered on. CSS uses media type blocks to specify how a document is
to be presented on different media: on the screen, on paper, with a speech
synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not
contained in any media blocks. By using the media-type suboption, in addition to
the general CSS code, you can import the section of a CSS file intended only for
a specific media type.

Default
If no media-type is specified in your ODS statement, but you do
have media types specified in your CSS file, then ODS uses the
Screen media type.

Range
You can specify up to ten different media types.
Requirements

You must enclose *media-type* in parentheses.

You must specify *media-type* next to the *file-specification* specified by the CSSSTYLE= option.

Tip

If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

Restriction

The CSSSTYLE= option does not affect SAS/GRAph output.

Requirement

CSS files must be written in the same type of CSS produced by the ODS HTML statement. To view the CSS code that ODS creates, you can do one of the following:

• specify the ODS TRACE DOM statement
• specify the DOM option

Interaction

If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

See

For complete documentation about ODS and Cascading Style Sheets, see *SAS Output Delivery System: Advanced Topics*

Example

“Example 6: Applying a CSS File to ODS Output” on page 527

**DOM<="external-file"**

specifies that the ODS document object model is written to the SAS log or an external file.

*external-file*

is the name of an external output file.

Requirement

You must enclose *external-file* in quotation marks.

See

For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in *SAS Output Delivery System: Advanced Topics*.

**DPI=**

specifies the image resolution for output files.

Default

150

Restriction

The DPI= option takes effect only if specified at the opening of a file.

**CAUTION**

When using high DPI= or DPI_IMAGE= values (values over 600), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500 or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.

**FILE='external-file' | fileref**

specifies the output file.
**Requirements**

*external-file*

is the name of an external file.

**You must enclose *external-file* in quotation marks.**

*fileref*

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

**Restriction**

The FILE=fileref option cannot be used in conjunction with the NEWFILE= option.

**See**

For information about the FILENAME statement, see *SAS Statements: Reference.*

**Default**

If you do not specify an output file, then ODS writes to the file that is specified by two SAS system options. Use the SYSPRINT= system option if you are using the Windows operating environment and do not specify any of the following options: PCL, PDFMARK, POSTSCRIPT, PS, or SAS. Use the PRINTERPATH= system option in all other cases. If the system option does not specify a file, then ODS writes to the default printer. For more information, see the PRINTER= option.

**Interaction**

In an ODS printer family statement that refers to an open ODS PRINTER destination, the FILE= option forces ODS to close the destination and all files that are associated with it. ODS then opens a new instance of the destination. For more information, see “Opening and Closing the PRINTER Destination” on page 708.

**See**

For information about the FILENAME statement, see *SAS Statements: Reference.*

**GFOOTNOTE | NOGFOOTNOTE**

controls the location of the footnotes that are defined by the graphics program that generates the Printer output.

**GFOOTNOTE**

includes all of the currently defined footnotes within the graphics output.

**NOGFOOTNOTE**

prevents all of the currently defined footnotes from appearing in the graphics file. Instead, they become part of the Printer file.

**Default**

GFOOTNOTE

**Restriction**

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

**See**

For more information, see “Customizing Titles and Footnotes” on page 40.

**GTITLE | NOGTITLE**

controls the location of the titles that are defined by the graphics program that generates the Printer output.

**GTITLE**

includes all of the currently defined titles within the graphics output.
NOGTITLE
prevents all of the currently defined titles from appearing in the graphics output. Instead, the titles become part of the Printer file.

Default  GTITLE

Restriction  This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

See  For more information, see “Customizing Titles and Footnotes” on page 40.

(<ID=> identifier)
enables you to open multiple instances of the same destination at the same time. Each instance can have different options.

identifier  can be numeric or can be a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numerals.

Restriction  If identifier is numeric, it must be a positive integer.

Requirement  The ID= option must be specified immediately after the destination name.

KEYWORDS='keywords-text'
inserts a string of keywords into the output file's metadata. The keywords enable a document management system to do topic-based searches.

keywords-text  is the string of keywords.

Restrictions  Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

The KEYWORDS= option takes effect only if specified at the opening of a file.

Requirement  You must enclose keywords-text in quotation marks.

NEWFILE= starting-point  creates a new file at the specified starting-point.

starting-point  is the location in the output where you want to create a new file.

ODS automatically names new files by incrementing the name of the file. In the following example, ODS names the first file REPORT.PS. Additional body files are named REPORT1.PS, REPORT2.PS, and so on.

Example:
FILE= 'REPORT.PS'

starting-point can be one of the following:

BYGROUP  starts a new file for the results of each BY group.
NONE
writes all output to the file that is currently open.

OUTPUT
starts a new file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

Alias  TABLE

PAGE
starts a new file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.

PROC
starts a body file each time you start a new procedure.

Default  NONE

Restrictions
The NEWFILE= option cannot be used in conjunction with the FILE=fileref option.

The NEWFILE= option cannot be used if you are sending output to a physical printer.

Tips
If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first file MAY5.PS. Additional body files are named MAY6.PS, MAY7.PS, and so on.

Example:
FILE= 'MAY5.PS'

NOTOC
specifies that ODS omit the table of contents (Bookmark list) that is produced by default when producing PDF or PDFMARK output.

Interaction
The NOTOC option specifies BOOKMARKLIST=OFF and CONTENTS= OFF.

Examples
“Example 5: Combining a Table and Image on the Same Page” on page 608
“Example 6: Adding Text That Imitates a System Title” on page 611
“Example 7: Toggling Page Breaks” on page 615
“Example 8: Suppressing a Page Break” on page 619

PACKAGE <package-name>
specifies that the output from the destination be added to a package.

package-name
specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

See  “ODS PACKAGE Statement ” on page 554
PDFNOTE | NOPDFNOTE
controls whether notes are added to a PDF file for items that are associated with the FLYOVER= style attribute.

PDFNOTE
adds notes to a PDF file for items that are associated with the FLYOVER= style attribute.

NOPDFNOTE
modifies the behavior of PDFMARK so that notes are not added to the file for items that are associated with the FLYOVER= style attribute.

Default PDFNOTE

Restriction Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and ODS PRINTER statement with the PDFMARK option specified.

PDFTOC=n
controls the level of the expansion of the table of contents in PDF documents.

n specifies the level of expansion. For example, PDFTOC=0 results in a fully expanded table of contents. PDFTOC=2 results in a table of contents that is expanded to two levels.

Default 0

Tip The PDFTOC= can be set after the file has been opened, but only the last specification for a given file is used.

See “Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594

STARTPAGE=NEVER | NO | NOW | YES | BYGROUP
controls page breaks.

BYGROUP
specifies to insert page breaks after each BY group.

NEVER
specifies not to insert page breaks, even before graphics procedures.

CAUTION:
Each graph normally requires an entire page. The default behavior forces a new page after a graphics procedure. STARTPAGE=NEVER turns off that behavior, so specifying STARTPAGE= NEVER might cause graphics to overprint.

NO
specifies that no new pages be inserted at the beginning of each procedure, or within certain procedures, even if new pages are requested by the procedure code. A new page begins only when a page is filled or when you specify STARTPAGE=NOW.

CAUTION:
Each graph normally requires an entire page. The default behavior forces a new page after a graphics procedure, even if you use STARTPAGE=NO. STARTPAGE=NEVER turns off that behavior, so specifying STARTPAGE= NEVER might cause graphics to overprint.
<table>
<thead>
<tr>
<th>Alias</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tip</td>
<td>When you specify STARTPAGE=NO, system titles and footnotes are still produced only at the top and bottom of each physical page, regardless of the setting of this option. Thus, some system titles and footnotes that you specify might not appear when this option is specified.</td>
</tr>
<tr>
<td>Examples</td>
<td>“Example 5: Combining a Table and Image on the Same Page” on page 608</td>
</tr>
<tr>
<td></td>
<td>“Example 6: Adding Text That Imitates a System Title” on page 611</td>
</tr>
<tr>
<td></td>
<td>“Example 7: Toggling Page Breaks” on page 615</td>
</tr>
<tr>
<td></td>
<td>“Example 8: Suppressing a Page Break” on page 619</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NOW</th>
<th>forces the immediate insertion of a new page.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tip</td>
<td>This option is useful primarily when the current value of the STARTPAGE= option is NO. Otherwise, each new procedure forces a new page automatically.</td>
</tr>
<tr>
<td>Example</td>
<td>“Example 7: Toggling Page Breaks” on page 615</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>YES</th>
<th>inserts a new page at the beginning of each procedure, and within certain procedures, as requested by the procedure code.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>ON</td>
</tr>
<tr>
<td>Default</td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STYLE=style-template</th>
<th>specifies the style template to use in writing the printer output.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>If you do not specify a style template, then ODS uses the style template that is specified in the SAS registry subkey: ODS \ DESTINATIONS \ PRINTER. By default, this value is Pearl for the PRINTER, PDF, PS and PCL destinations.</td>
</tr>
<tr>
<td>Note</td>
<td>If you are using SAS Studio, you do not need to specify the STYLE= option. You can go to Preferences \ Results and change the style from the drop-down list for your selected destination.</td>
</tr>
<tr>
<td>See</td>
<td>For a complete discussion of style templates, see “Working with Styles ” in SAS Output Delivery System: Procedures Guide.</td>
</tr>
<tr>
<td></td>
<td>For instructions on making your own user-defined style templates, see “TEMPLATE Procedure: Creating a Style Template ” in SAS Output Delivery System: Procedures Guide.</td>
</tr>
<tr>
<td>Examples</td>
<td>“Example 3: Customizing the Table of Contents” on page 600</td>
</tr>
<tr>
<td></td>
<td>“Example 6: Adding Text That Imitates a System Title” on page 611</td>
</tr>
</tbody>
</table>
SUBJECT='subject-text'
inserts into the metadata of a file the text string that you specify as the subject.

subject-text
is the text in the metadata of a file that indicates the subject.

Restrictions
Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

The SUBJECT= option takes effect only if specified at the opening of a file.

Requirement
You must enclose subject-text in quotation marks.

TEXT='text-string'
inserts a text string into your output.

text-string
is the text that you want to insert into your output.

Requirement
You must enclose text-string in quotation marks.

Tip
If you are submitting more than one procedure step and you do not specify the STARTPAGE=NO option, each procedure forces a new page before the output. Therefore, any text that you specify with TEXT= is on the same page as the previous procedure.

See
“Example: Conditionally Excluding Output Objects and Sending Them to Different Output Destinations” on page 325

TITLE='title-text'
inserts into the metadata of a file the text string that you specify as the title.

title-text
is the text in the metadata of a file that indicates the title.

Restrictions
Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

The TITLE= option takes effect only if specified at the opening of a file.

Requirement
You must enclose title-text in quotation marks.

UNIFORM
for multiple page tables, ensures uniformity from page to page within a single table. When the UNIFORM option is in effect, ODS reads the entire table first, so that it can determine the column widths that are necessary to accommodate all the data. These column widths are applied to all pages of a multiple page table.

Default
If you do not specify the UNIFORM option, then ODS prints a table one page at a time. This approach ensures that SAS does not run out of memory while processing very large tables. However, it can also mean that column widths vary from one page to the next.

Note
With BY-group processing, SAS writes the results of each BY group to a separate table, so the output might not be uniform across BY groups.
Tip
The UNIFORM option can cause SAS to run out of memory if you are printing a very large table. If this happens, then you can explicitly set the width of each of the columns in the table, and then print the table one page at a time. To do so, you must edit the table template that you use. For more information, see “What You Can Do with Table Templates” in SAS Output Delivery System: Procedures Guide.

Details

The ODS Printer Family of Statements
The ODS PDF statement is part of the ODS printer family of statements. Statements in the printer family open the PCL, PDF, PRINTER, or PS destination, producing output that is suitable for a high-resolution printer. The ODS PCL, ODS PRINTER, and ODS PS statements are also members of the ODS printer family of statements.

Opening and Closing the PDF Destination
You can modify an open PDF destination with many ODS PDF options. However, the FILE= and SAS options perform the following actions on an open PDF destination:

- close the open destination referred to in the ODS PDF statement
- close any files associated with the open PDF destination
- open a new instance of the PDF destination

If you use one of these options, it is best if you explicitly close the destination yourself.

Securing ODS-Generated PDF Files
You can use the ODS PRINTER statement or the ODS PDF statement to generate PDF output. By default, PDF files are not password protected, so any user can view, and edit the PDF files without restrictions. However, you can use SAS system options to restrict or allow users' ability to access, assemble, copy, or modify the ODS PDF files. Other SAS system options control whether the user can fill in forms and set the print resolution.

Setting the security of a PDF file involves setting an encryption level and setting PDF document properties. You use the following SAS system options to secure and configure document properties for PDF files:

<table>
<thead>
<tr>
<th>Action</th>
<th>System Option</th>
<th>Document Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifies whether text and graphics from PDF documents can be read by screen readers for the visually impaired</td>
<td>PDFACCESS</td>
<td>Content Accessibility Enabled</td>
</tr>
<tr>
<td>See PDFACCESS.</td>
<td></td>
<td>Content Accessibility Enabled</td>
</tr>
<tr>
<td>Specifies whether PDF documents can be assembled</td>
<td>PDFASSEMBLY</td>
<td>Document Assembly</td>
</tr>
<tr>
<td>Action</td>
<td>System Option</td>
<td>Document Property</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------</td>
<td>-----------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Specifies whether PDF document comments can be modified</td>
<td>PDFCOMMENT</td>
<td>Commenting</td>
</tr>
<tr>
<td>Specifies whether the contents of a PDF document can be changed</td>
<td>PDFCONTENT</td>
<td>Changing the Document</td>
</tr>
<tr>
<td>Specifies whether text and graphics from a PDF document can be copied</td>
<td>PDFCOPY</td>
<td>Content Copying</td>
</tr>
<tr>
<td>Specifies whether PDF forms can be filled in</td>
<td>PDFFILLIN</td>
<td>Form Field Fill-in or Signing</td>
</tr>
<tr>
<td>Specifies the password to use to open a PDF document and the password used by a PDF document owner</td>
<td>PDFPASSWORD=</td>
<td>Security Method</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Document Open Password</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Permissions Password</td>
</tr>
<tr>
<td>Specifies the resolution used to print the PDF document</td>
<td>PDFPRINT=</td>
<td>Printing</td>
</tr>
<tr>
<td>Specifies the level of encryption for PDF documents</td>
<td>PDFSECURITY=</td>
<td>Encryption Level</td>
</tr>
</tbody>
</table>

The PDF system options are documented in *SAS System Options: Reference*.

*Note:* The SAS/SECURE SSL software that is used to encrypt PDF files is included in the SAS installation software only for countries that allow the importation of encryption software.

You secure a PDF file by setting the PDFSECURITY= system option to an encryption level. Valid security levels for the PDFSECURITY= option are NONE or HIGH. SAS sets the default PDF document properties based on the encryption level.

PDFSECURITY=NONE sets no encryption level or document property restrictions for the document. All of the PDF document properties are set to Allowed. Setting other PDF system options has no effect on PDF document properties when PDFSECURITY=NONE.

PDFSECURITY=HIGH sets the encryption level to 128-bit RC4.

When the PDFSECURITY= option is set to HIGH, you must specify one or more document passwords using the PDFPASSWORD= option. Passwords are required to open a secure document. An optional permissions password can be required to validate the document owner. Use the OPEN= "pw" argument to specify a password to open a document. Use the OWNER= "pw" argument to specify a permissions password for the document owner.

To view the document properties for a PDF file, open the PDF file, right-click in the document, select **Document Properties** from the menu, and click **Show Details**. The Document Security window appears with the document property values.

*Note:* The **Security** tab in the Document Properties window displays the security settings. When PDFSECURITY=NONE, the **Show Details** button is inactive and the **Document Restrictions Summary** section displays the document property value of Allowed for all properties. If PDFSECURITY= is set HIGH, ignore the **Document Restrictions Summary** section. The PDF document properties are displayed...
properly only from the Document Security window, which you access with the Show Details button.

The Yes and No values for the Document Open Password and the Permissions Password document properties indicate whether password security has been set for a document. These values are determined by the values of the PDFSECURITY= option and the PDFPASSWORD= option as shown in this table:

<table>
<thead>
<tr>
<th>PDFPASSWORD=</th>
<th>PDFSECURITY=HIGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Method</td>
<td>Password Security</td>
</tr>
<tr>
<td>Document Open Password</td>
<td>OPEN=&quot;pw&quot; Yes</td>
</tr>
<tr>
<td>OWNER=&quot;pw&quot; No</td>
<td></td>
</tr>
<tr>
<td>OPEN=&quot;pw&quot; OWNER=&quot;pw&quot; Yes</td>
<td></td>
</tr>
<tr>
<td>Permissions Password</td>
<td>OPEN=&quot;pw&quot; No</td>
</tr>
<tr>
<td>OWNER=&quot;pw&quot; Yes</td>
<td></td>
</tr>
<tr>
<td>OPEN=&quot;pw&quot; OWNER=&quot;pw&quot; Yes</td>
<td></td>
</tr>
</tbody>
</table>

Nearly all other document properties can be set to Allowed or Not Allowed by using other PDF system options. The Page Extraction property cannot be set by using a system option. To see how the individual options set the document properties, see the documentation for the PDF system options in SAS System Options: Reference.

The following table shows the default PDF document properties for the two values of the PDFSECURITY= option:

<table>
<thead>
<tr>
<th>PDFSECURITY=NONE</th>
<th>PDFSECURITY=HIGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Printing Allowed</td>
<td>High Resolution</td>
</tr>
<tr>
<td>Changing the Document Allowed</td>
<td>Not Allowed</td>
</tr>
<tr>
<td>Commenting Allowed</td>
<td>Not Allowed</td>
</tr>
<tr>
<td>Form Field Fill-in or Signing Allowed</td>
<td>Allowed</td>
</tr>
<tr>
<td>Document Assembly Allowed</td>
<td>Not Allowed</td>
</tr>
<tr>
<td>Content Copying Allowed</td>
<td>Allowed</td>
</tr>
<tr>
<td>Content Accessibility Enabled Allowed</td>
<td>Allowed</td>
</tr>
<tr>
<td>Page Extraction Allowed</td>
<td>Not Allowed</td>
</tr>
<tr>
<td>Encryption Level None</td>
<td>128-bit RC4</td>
</tr>
</tbody>
</table>

* Documents that are created when PDFSECURITY=HIGH can be viewed using Acrobat 5.0 and later.

Some document properties are set by SAS system options only when PDFSECURITY=HIGH.
Importing Existing Images into a PDF File

Images can be included in PDF files created with SAS using the style attributes BACKGROUNDIMAGE, POSTIMAGE, and PREIMAGE. These can be included using PROC TEMPLATE or inline style syntax.

In order to get the best looking images included in your ODS PDF file, the images should follow these parameters:

- Images should be in one of the following formats: JPEG, PNG, GIF, BMP, TIFF.
- Under z/OS, you must enable the UNIX file system HFS.
- The image should have 150 dots per inch (DPI). It is important that all images have a consistent DPI. For example, if you have more than one image to include in your PDF file(s), they should all have the same DPI (150 for best results).
- If you use JPG files, these should conform to the JFIF standard, use RGB colors, and should not use transparency in the image.
- It is recommended that you create the image in the height and width that is desired in the final report output. For example, if you need to display a banner image that is 2 inches by 6 inches in size, the file should be 150 DPI, use RGB colors, and have the dimensions of 2 inches by 6 inches. ODS PDF will then accurately include the image in the resulting PDF file.

Working with the Table of Contents

The ODS PDF destination provides the following two navigation tools:

- The default table of contents (TOC), which is a clickable bookmark tree that is not printed.

Output 6.60  PDF Output Default Bookmark Tree

- A printable table of contents, which is generated using the CONTENTS=YES option on the ODS PDF FILE= statement. The output created this way is static and, just like the table of contents in a book, does not count toward the page count of the PDF file. The text “Table of Contents” is customizable using PROC TEMPLATE, and the text of each of the entries is customizable with the ODS PROCLABEL statement and CONTENTS= options on some of the PROC statements.
The text displayed by the nodes of each tool is controlled with the following:

- the ODS PROCLABEL statement
- the CONTENTS=, the DESCRIPTION=, and the OBJECTLABEL= options
- the DOCUMENT destination and procedure
- the TEMPLATE procedure

Examples

**Example 1: Opening Multiple Instances of the Same Destination at the Same Time**

**Features:**
- ODS PDF statement option:
  - ID=
  - STYLE=
  - FILE=
- Other features:
  - PROC FORMAT
  - PROC SORT
  - PROC REPORT
  - NOBYLINE|BYLINE system option
  - #BYVAL parameter in titles

**Data set:**
- Grain_Production

**Format:**
- $CNTRY

**Details**

This example opens multiple instances of the PDF destination to create PDF output. One instance uses the default style template and the second instance uses the STYLE= option to specify the Sapphire style template.

**Program**

```sas
proc sort data=grain_production;
   by year country type;
run;
ods HTML close;
opts nobyline nodate;
title 'Leading Grain-Producing Countries';
```
title2 'for #byval(year)';
ods pdf file='grain-1.pdf' pdftoc=2;
ods pdf (id=SapphireStyle) style=Sapphire file='grain-2.pdf' pdftoc=3;
proc report data=grain_production nowindows;
   by year;
   column country type kilotons;
   define country / group width=14 format=$cntry.;
   define type     / group 'Type of Grain';
   define kilotons / format=comma12.;
   footnote 'Measurements are in metric tons.';
run;
options byline;

Program Description

Sort the data set Grain_Production. SORT sorts the data first by values of Year, then by values of Country, and finally by values of Type.

proc sort data=grain_production;
   by year country type;
run;

Close the HTML destination so that no HTML output is produced. The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources. If the destination were left open, then ODS would produce both HTML and PDF output.

ods HTML close;

Suppress the default BY line, suppress the printing of the date, and use the BY value in a title. The NOBYLINE option suppresses the BY line. The #BYVAL specification inserts the current value of the BY variable Year into the title.

options nobyline nodate;
   title 'Leading Grain-Producing Countries';
   title2 'for #byval(year)';

Create two different PDF output files at the same time. The ODS PDF statement opens the PDF destination and creates PDF output. The file Grain-1.pdf is created by the first ODS PDF statement. Because no style template is specified, the default style,
Styles.Pearl, is used. The PDFTOC=2 option specifies that the table of contents is expanded two levels. The file Grain-2.pdf is created by the second ODS PDF statement with the ID= option specified. The STYLE= option specifies that ODS use the style template Sapphire. The ID= option gives this instance of the PDF destination the name SapphireStyle. The PDFTOC=3 option specifies that the table of contents is expanded three levels. If you do not specify the ID= option, this ODS PDF statement closes the instance of the PDF destination that was opened by the previous ODS PDF statement and opens a new instance of the PDF destination. The file Grain-1.pdf contains no output.

```plaintext
ods pdf file='grain-1.pdf' pdftoc=2;
ods pdf (id=SapphireStyle) style=Sapphire file='grain-2.pdf' pdftoc=3;
```

### Produce a report
This PROC REPORT step produces a report on grain production. Each BY group produces a page of output.

```plaintext
proc report data=grain_production nowindows;
  by year;
  column country type kilotons;
  define country / group width=14 format=$cntry.;
  define type / group 'Type of Grain';
  define kilotons / format=comma12.;
  footnote 'Measurements are in metric tons.';
run;
```

### Restore the BY line and clear the second title statement
The BYLINE option restores the BY line. The TITLE2 statement clears the second TITLE statement.

```plaintext
options byline;
title2;
```

### Produce a report that contains one table for each year
The TABLE statement in this PROC TABULATE step has Year as the page dimension. Therefore, PROC TABULATE explicitly produces one table for 1995 and one for 1996.

```plaintext
proc tabulate data=grain_production format=comma12.;
  class year country type;
  var kilotons;
  table year,
    country*type,
    kilotons*sum=' ' / box=_page_ misstext='No data';
  format country $cntry.;
  footnote 'Measurements are in metric tons.';
run;
```

### Close the open destinations so that you can view or print the output
The ODS PDF CLOSE statement closes the first instance of the PDF destination and all of the files that are associated with it. The ODS PDF (ID=SapphireStyle) statement closes the second instance of the PDF destination and all of the files that are associated with it. You must close the destinations before you can view the output with a browser or before you can send the output to a physical printer.

```plaintext
ods pdf close;
ods pdf(id=SapphireStyle) close;
```

### Open the HTML destination
The ODS HTML statement opens the HTML destination and returns SAS to the default ODS destination.

```plaintext
ods html;
```
PDF Output

The default style for the ODS PDF, ODS PS, ODS PCL, and ODS PRINTER statements is Pearl.

**Output 6.62  PDF Output with the Default Style Applied**

### Leading Grain-Producing Countries for 1995

<table>
<thead>
<tr>
<th>Country</th>
<th>Type of Grain</th>
<th>Kilotons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brazil</td>
<td>Corn</td>
<td>36,276</td>
</tr>
<tr>
<td></td>
<td>Rice</td>
<td>11,236</td>
</tr>
<tr>
<td></td>
<td>Wheat</td>
<td>1,516</td>
</tr>
<tr>
<td>China</td>
<td>Corn</td>
<td>112,331</td>
</tr>
<tr>
<td></td>
<td>Rice</td>
<td>185,226</td>
</tr>
<tr>
<td></td>
<td>Wheat</td>
<td>102,207</td>
</tr>
<tr>
<td>India</td>
<td>Corn</td>
<td>9,800</td>
</tr>
<tr>
<td></td>
<td>Rice</td>
<td>122,372</td>
</tr>
<tr>
<td></td>
<td>Wheat</td>
<td>63,007</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Corn</td>
<td>8,223</td>
</tr>
<tr>
<td></td>
<td>Rice</td>
<td>49,860</td>
</tr>
<tr>
<td></td>
<td>Wheat</td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>Corn</td>
<td>187,300</td>
</tr>
<tr>
<td></td>
<td>Rice</td>
<td>7,888</td>
</tr>
<tr>
<td></td>
<td>Wheat</td>
<td>59,494</td>
</tr>
</tbody>
</table>
Example 2: Creating a Printable Table of Contents

Features:
- ODS PDF statement option:
  BOOKMARKLIST=
  CONTENTS=
  FILE=

Other features:
- OPTIONS statement
- PROC FREQ
- PROC PRINT

Details
By default, ODS PDF does not create a printable table of contents, only a clickable bookmark tree. This example shows you how to create a printable table of contents.

Program
```
ods html close;

  title "Create a Table of Contents";
```
options nodate;
ods pdf file="MyDefaultToc.pdf" contents=yes bookmarklist=hide;

proc freq data=sashelp.cars;
  tables origin*type;
run;

proc print data=sashelp.cars;
run;

ods pdf close;
ods html;

**Program Description**

---

**Close the HTML destination so that no HTML output is produced.** The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources. If the destination were left open, then ODS would produce both HTML and PDF output.

```
ods html close;
```

---

**Specify a title and set the SAS system options.**

```
title "Create a Table of Contents";
options nodate;
```

---

**Open the PDF destination and specify the ODS PDF statement options.** The ODS PDF statement opens the PDF destination and the FILE= option specifies PDF filename. The CONTENTS=YES option specifies that a table of contents is created. Because you are creating a table of contents, you might not need the bookmark tree. The BOOKMARKLIST=HIDE option specifies that a bookmark tree is created but hidden.

```
ods pdf file="MyDefaultToc.pdf" contents=yes bookmarklist=hide;
```

---

**Create the procedure output.**

```
proc freq data=sashelp.cars;
  tables origin*type;
run;

proc print data=sashelp.cars;
run;
```

---

**Close the PDF destination and open the HTML destination.** The ODS PDF CLOSE statement closes the PDF destination and all of the files that are associated with it. You must close the destinations before you can view the output with a browser or before you can send the output to a physical printer. The ODS HTML statement opens the HTML destination and returns SAS to the default ODS destination.

```
ods pdf close;
ods html;
```
**Example 3: Customizing the Table of Contents**

**Features:**

- ODS PDF statement option
  
  FILE=
  STYLE=

- PROC TEMPLATE options:
  
  DEFINE statement
  PARENT= statement
  CLASS statement
  PRETEXT= style attribute
  ContentTitle style element

- ODS PROCLABEL
- ODS HTML

**Other features:**

- OPTIONS statement
- PROC FREQ
- PROC PRINT

**Details**

This example shows you how to customize the table of contents with the following tasks:

- PROC TEMPLATE is used to change the table of contents title text.
- ODS PROCLABEL is used to change the first-level node text for the PROC FREQ and PROC PRINT output.
- The CONTENTS= option is used in the TABLES statement in PROC FREQ to eliminate the second node of text for the PROC FREQ crosstab table.
- The CONTENTS= option is used in the PROC PRINT statement to change the text of the second node created by the PROC PRINT table.

This example requires some knowledge of style templates, style elements, and style attributes. For complete documentation about styles, see “Understanding Styles, Style Elements, and Style Attributes” on page 913. For a table of style elements affecting the table of contents and table of pages, see Table 11.4 on page 967. For a table of style attributes, see Table 12.1 on page 994.

**Program**

```plaintext
ods html close;
```
proc template;
define style Styles.CustomTitle;
parent=Styles.pearl;
   class ContentTitle from ContentTitle /
      pretext='My Customized Title';
end;
run;
title "Create a Custom Table of Contents";
options nodate;
ods pdf file="CustomTOC.pdf" style=Styles.CustomTitle;
ods proclabel "Crosstab of SASHELP.CARS";
proc freq data=sashelp.cars;
   tables origin*type / contents="";
run;
ods proclabel "All variables: SASHELP.CARS";
proc print data=sashelp.cars contents="Second level";
run;
ods pdf close;
ods html;
proc template;
delete Styles.CustomTitle;
run;

Program Description

Close the HTML destination so that no HTML output is produced. The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources. If the destination were left open, then ODS would produce both HTML and PDF output.

ods html close;

Create a style to change the table of contents title text. The first step in using PROC TEMPLATE to modify the title text is to create a new style. The DEFINE STYLE statement in PROC TEMPLATE specifies that CustomTitle is the name of the new style, and that it is stored in the Styles item store. The PARENT= statement specifies that all style elements and attributes are inherited from the style Styles.Pearl, which is the PDF default style.

proc template;
define style Styles.CustomTitle;
parent=Styles.pearl;

Specify the text for the custom title. The ContentTitle style element controls the appearance of the contents title. The CLASS statement specifies that the style element ContentTitle is going to be modified. The PRETEXT= style attribute specifies the new text for the table of contents title.

   class ContentTitle from ContentTitle /
      pretext='My Customized Title';
end;
Specify a title and set the SAS system options.

```sas
title "Create a Custom Table of Contents";
options nodate;
```

Open the PDF destination and specify the ODS PDF statement options. The ODS PDF statement opens the PDF destination and the FILE= option specifies PDF filename. The STYLE= option specifies that the custom style Styles.CustomTitle is applied to the output.

```sas
ods pdf file="CustomTOC.pdf" style=Styles.CustomTitle;
```

For PROC FREQ, change the text for the first-level node and suppress the second-level node. The ODS PROCLABEL statement specifies that the text "Crosstab of SASHELP.CARS" overrides the default procedure label, which is used as the first-level node. The PROC FREQ option CONTENTS=" " suppresses the text for the second-level node.

```sas
ods proclabel "Crosstab of SASHELP.CARS";
proc freq data=sashelp.cars;
   tables origin*type / contents="";
run;
```

For PROC PRINT, change the text for both the first-level node and the second-level node. The ODS PROCLABEL statement specifies that the text "All variables: SASHELP.CARS" overrides the default procedure label, which is used as the first-level node. The PROC PRINT option CONTENTS= specifies custom text for the second-level node.

```sas
ods proclabel "All variables: SASHELP.CARS";
proc print data=sashelp.cars contents="Second level";
run;
```

Close the PDF destination and open the HTML destination. The ODS PDF CLOSE statement closes the PDF destination and all of the files that are associated with it. You must close the destinations before you can view the output with a browser or before you can send the output to a physical printer. The ODS HTML statement opens the HTML destination and returns SAS to the default ODS destination.

```sas
ods pdf close;
ods html;
```

After your output is produced, you can remove the custom style. If you no longer want to keep the new style, you can use the DELETE statement in PROC TEMPLATE to remove the custom style.

```sas
proc template;
delete Styles.CustomTitle;
run;
```
Example 4: Customizing BY Lines

Features:
- ODS PDF statement option
  FILE=
- ODS HTML
- ODS LISTING statement
- ODS OUTPUT statement
- PROC DOCUMENT

Other features:
- OPTIONS statement
- PROC PRINT
- PROC SORT
- PROC TABULATE
- DATA step
- MACRO language

Details
The use of BY variables sometimes results in more text than is desired in the tables of contents. The following example shows the use of ODS, PROC DOCUMENT, and the MACRO language to do the following:

- automate the labeling of the BYLINE entries from “By variable=By value” to “By Value”
- compress the table of contents entries to show one node for each table
For complete documentation about the MACRO language, see *SAS Macro Language: Reference*. For documentation about the DOCUMENT procedure, see “The DOCUMENT Procedure” in *SAS Output Delivery System: Procedures Guide*.

**Step 1: Create an ODS Document**

```plaintext
ods html close;
ods pdf file="DefaultTOC.pdf";
proc sort data=sashelp.cars out=cars;
  by origin;
run;
ods document name=Reorder(write);
proc tabulate data=cars;
  title "#byval1";
  by origin;
  class cylinders;
  var mpg_highway;
  table cylinders, mpg_highway*mean;
run;
ods _all_ close;
```

**Program Description**

The first step is to create an ODS document to store your PROC TABULATE output. You only have to create the procedure output one time and save it in an ODS document. This document, named Reorder, is used to manipulate the output from this point on.

---

**Close the HTML destination and open the PDF destination.** The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources. If the destination were left open, then ODS would produce both HTML and PDF output. The ODS PDF statement opens the PDF destination and names the file the PDF output is written to.

```plaintext
ods html close;
ods pdf file="DefaultTOC.pdf";
```

---

**Sort the data set Sashelp.Cars by the variable ORIGIN**

```plaintext
proc sort data=sashelp.cars out=cars;
  by origin;
run;
```

---

**Create the ODS document Reorder.** The ODS DOCUMENT statement opens the DOCUMENT destination and specifies the name of a new document, Reorder. Reorder contains the PROC TABULATE output objects. This example uses Reorder in the new step to manipulate the TOC entries using PROC DOCUMENT.

```plaintext
ods document name=Reorder(write);
```

---

**Create the TABULATE procedure output.**

```plaintext
proc tabulate data=cars;
  title "#byval1";
  by origin;
  class cylinders;
```
var mpg_highway;
table cylinders, mpg_highway*mean;
run;

Close all open destinations.
ods _all_ close;

The following output shows the Bookmark list in the PDF file. The same text will show in the printable table of contents, if CONTENTS=YES is specified.

Output 6.67 Default Table of Contents

Step 2: Create a DATA Set
ods output properties=Props;
proc document name=reorder;
list / levels=all;
run;
quit;
ods listing;
proc print data=props;
run;
ods listing close;

Program Description
Next, move the contents of the Reorder document into a DATA set. This enables you to automate the logic in step 3.

Create the Props data set. The ODS OUTPUT statement creates a data set name Props.
ods output properties=Props;

List the contents of the REORDER document and save them into the Props data set. The PROC DOCUMENT statement opens the document Reorder. The LIST statement lists the contents of Reorder. The contents are saved in the data set Props.
proc document name=reorder;
list / levels=all;
run;
quit;

Print the Props data set to see the document level names. At this point you might want to look at the contents of the Props data set. You can use the PROC PRINT and ODS LISTING statement to view the document level names.

ods listing;
proc print data=props;
run;
ods listing close;

Notice that some of the levels are directories (Dir) and some are tables (Table).

Output 6.68  Document Level Names in LISTING Output

<table>
<thead>
<tr>
<th>Obs</th>
<th>Path</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tabulate#1</td>
<td>Dir</td>
</tr>
<tr>
<td>2</td>
<td>Tabulate#1\ByGroup1#1</td>
<td>Dir</td>
</tr>
<tr>
<td>3</td>
<td>Tabulate#1\ByGroup1\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>4</td>
<td>Tabulate#1\ByGroup1\Report#1\Table#1</td>
<td>Table</td>
</tr>
<tr>
<td>5</td>
<td>Tabulate#1\ByGroup2#1</td>
<td>Dir</td>
</tr>
<tr>
<td>6</td>
<td>Tabulate#1\ByGroup2\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>7</td>
<td>Tabulate#1\ByGroup2\Report#1\Table#1</td>
<td>Table</td>
</tr>
<tr>
<td>8</td>
<td>Tabulate#1\ByGroup3#1</td>
<td>Dir</td>
</tr>
<tr>
<td>9</td>
<td>Tabulate#1\ByGroup3\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>10</td>
<td>Tabulate#1\ByGroup3\Report#1\Table#1</td>
<td>Table</td>
</tr>
</tbody>
</table>

Step 3: Customize the Table of Contents

data _null_
set props end=last;
  if type in(‘Table’) then do;
    count+1;
    call symputx(‘patht‘|trim(left(count)),path);
  end;
call symputx(‘total’,count);
run;

%macro order;
proc document name=work.reorder;
  %do i=1 %to &total;
    setlabel &&patht&amp;i "#byvall";
    move &&patht&amp;i to ^;
  %end;
ods pdf file="replayed.pdf";
  replay ;
run;
ods pdf close;
ods listing;
  list / levels=all;
run;
ods listing close;
Program Description

Now you can start modifying your table of contents. PROC DOCUMENT syntax changes the text and structure of the table of contents. The macro language enables you to automate the process.

Place the table entries into macro variables. This DATA step selects only the entries labeled Table. It then places path values into the macro variables PathN and Total. These macro variables are used in a later %DO loop with macro logic.

```sas
data _null_; set props end=last; if type in("Table") then do; count+1; call symputx('patht'||trim(left(count)),path); end; call symputx('total',count); run;
```

Reorder and rename the table of contents, and begin the macro definition. The %MACRO statement creates the macro definition name Order. The MOVE statement moves the contents of the variable Path to the current working directory. This creates one node for each table in the table of contents. The SETLABEL statement renames the node entries to the name of the By variable.

```sas
%macro order; proc document name=work.reorder; %do i=1 %to &total; setlabel &&patht &i "#byval1"; move &&patht &i to ^; %end;
```

Replay the output to the ODS PDF destination. Create the PDF output by replaying the output to the PDF destination.

```sas
ods pdf file="replayed.pdf"; replay ; run; ods pdf close;
```

Show the modified document entries in the LISTING destination and end the macro definition. You can view all of the modified entries using the LIST statement and the ODS LISTING destination. The %MEND statement ends the macro definition.

```sas
ods listing; list / levels=all; run; ods listing close; quit;
```
Invoke the macro %ORDER. Now that your program is stored in a macro, invoke the macro %ORDER in order to create the output.

%order

Output 6.69  Customized BY Line in PDF Output

Output 6.70  Modified Document Entries

<table>
<thead>
<tr>
<th>Obs</th>
<th>Template</th>
<th>Label</th>
<th>Page Break</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>The Tabulate Procedure</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Origin-Asia</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Cross-tabular summary report</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Origin-Europe</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Cross-tabular summary report</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Origin-USA</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Cross-tabular summary report</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>*byvall</td>
<td>Before</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>*byvall</td>
<td>Before</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>*byvall</td>
<td>Before</td>
<td></td>
</tr>
</tbody>
</table>

Number of levels: All

<table>
<thead>
<tr>
<th>Obs</th>
<th>Path</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>\Tabulate+1</td>
<td>Dir</td>
</tr>
<tr>
<td>2</td>
<td>\Tabulate+1\ByGroup1+1</td>
<td>Dir</td>
</tr>
<tr>
<td>3</td>
<td>\Tabulate+1\ByGroup1\Report+1</td>
<td>Dir</td>
</tr>
<tr>
<td>4</td>
<td>\Tabulate+1\ByGroup2+1</td>
<td>Dir</td>
</tr>
<tr>
<td>5</td>
<td>\Tabulate+1\ByGroup2\Report+1</td>
<td>Dir</td>
</tr>
<tr>
<td>6</td>
<td>\Tabulate+1\ByGroup3+1</td>
<td>Dir</td>
</tr>
<tr>
<td>7</td>
<td>\Tabulate+1\ByGroup3\Report+1</td>
<td>Dir</td>
</tr>
<tr>
<td>8</td>
<td>Table+1</td>
<td>Table</td>
</tr>
<tr>
<td>9</td>
<td>Table+2</td>
<td>Table</td>
</tr>
<tr>
<td>10</td>
<td>Table+3</td>
<td>Table</td>
</tr>
</tbody>
</table>

Example 5: Combining a Table and Image on the Same Page

Features:
- ODS PDF statement option
  - FILE=
  - NOTOC
  - STARTPAGE=NO

ODS NOPROCTITLE statement
ODS HTML

Other features:
- OPTIONS statement
- PROC MEANS
- PROC SGPlot
Details

ODS PDF will paginate according to the individual procedures’ behavior. For example, PROC PRINT will generate a new page break for every BY group if the PAGEBY statement is used. PROC REPORT will generate a new page break for every BY group if a BY statement is used; it will also generate a new page if a BREAK statement (after or before) includes the PAGE option. SAS/GRAPH and ODS GRAPHICS procedures will place one image on a page then produce a page break. To override this behavior, the STARTPAGE= option is available in the ODS PDF statement.

Program

```sas
ods html close;
title "Eliminating Page Breaks";
options nodate;
ods pdf file="file.pdf" notoc startpage=no nogtitle;
ods noproctitle;
proc means data=sashelp.cars;
class cylinders;
var mpg_city mpg_highway;
run;
ods graphics / reset noborder width=6in;
proc sgplot data=sashelp.cars;
scatter x=cylinders y=mpg_highway;
scatter x=cylinders y=mpg_city;
yaxis label=" ";
keylegend / noborder;
run;
ods pdf close;
ods html;
```

Program Description

Close the HTML destination so that no HTML output is produced. The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources. If the destination were left open, then ODS would produce both HTML and PDF output.

```sas
ods html close;
```

Specify a title and set the SAS system options.

```sas
title "Eliminating Page Breaks";
options nodate;
```

Open the PDF destination and specify the ODS PDF statement options. The ODS PDF statement opens the PDF destination and the FILE= option specifies the PDF filename. The NOTOC option specifies that no table of contents is created. The STARTPAGE=NO option specifies that no new pages are inserted at the beginning of each procedure, or within certain procedures, even if new pages are requested by the procedure code. The NOGTITLE option specifies that the title is not inserted into the graphic image. The ODS NOPROCTITLE statement suppresses the writing of the procedure title that produces the results.

```sas
proc means data=sashelp.cars;
class cylinders;
var mpg_city mpg_highway;
run;
```
ods pdf file="file.pdf" notoc startpage=no nogtitle;
ods noproc title;

Create the MEANS procedure output.

proc means data=sashelp.cars;
  class cylinders;
  var mpg_city mpg_highway;
run;

Create a scatter plot with ODS graphics.

ods graphics / reset noborder width=6in;
proc sgplot data=sashelp.cars;
  scatter x=cylinders y=mpg_highway;
  scatter x=cylinders y=mpg_city;
  yaxis label="";
  keylegend / noborder;
run;

Close the PDF destination and open the HTML destination. The ODS PDF CLOSE statement closes the PDF destination and all of the files that are associated with it. You must close the destinations before you can view the output with a browser or before you can send the output to a physical printer. The ODS HTML statement opens the HTML destination and returns SAS to the default ODS destination.

ods pdf close;
ods html;
Example 6: Adding Text That Imitates a System Title

Features:
- ODS PDF statement option
  - FILE=
  - NOTOC
  - STARTPAGE=NO
STYLE=
ODS NOPROCTITLE statement
ODS HTML
ODS TEXT statement
PROC TEMPLATE

Other features:
OPTIONS statement
PROC MEANS
PROC PRINT
TITLE statement

Details
SAS titles and footnotes are displayed once per page in the PDF destination. Therefore, when the STARTPAGE= option is set to NO (OFF) and output from more than one procedure or DATA _NULL_ step is routed to ODS PDF, only the first set of titles and footnotes are written to the output file. This example shows how to use the TEXT= option to mimic an interim title, displayed above the second procedure output. PROC TEMPLATE is used to create a custom style template that mimics the style of the Systemtitle element. Systemtitle is the style element that controls the appearance of titles.

This example requires some knowledge of style templates, style elements, and style attributes. For complete documentation about styles, see “Understanding Styles, Style Elements, and Style Attributes ” on page 913. For a table of style elements affecting the table of contents and table of pages, see Table 11.4 on page 967. For a table of style attributes, see Table 12.1 on page 994.

Program
ods html close;
options nodate;
proc template;
define style styles.mimictitle;
parent=styles.pearl;
class usertext from systemtitle / just=c;
end;
run;
ods pdf file="file.pdf" notoc startpage=no style=styles.mimictitle;

title "Overriding the Default Procedure Title";
ods noproctitle;
proc means data=sashelp.cars;
class cylinders;
var mpg_city mpg_highway;
run;
ods text="My Custom PROC PRINT Output Title";
proc print data=sashelp.cars noobs;
  where mpg_highway gt 45;
run;

ods pdf close;
ods html;
proc template;
delete Styles.Mimictitle;
run;

Program Description

Close the HTML destination so that no HTML output is produced. The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources. If the destination were left open, then ODS would produce both HTML and PDF output.

ods html close;

Set the SAS system options.

options nodate;

Create a custom style template. The Usertext style element controls the appearance of user text, which includes text specified by the TEXT= option. The Systemtitle style element controls the appearance of the default SAS titles, which include the text specified by the TITLE= statement. This PROC TEMPLATE step creates a new style named Styles.Mimictitle, which contains all of the style elements and style attributes that Styles.Pearl contains. However, the CLASS statement modifies the Usertext style element to produce center-justified text, just as the Systemtitle style element does. This ensures that text specified by the TEXT= option will look the same as the text specified by the TITLE= statement.

proc template;
  define style styles.mimictitle;
  parent=styles.pearl;
  class usertext from systemtitle /
    just=c;
  end;
run;

Open the PDF destination and specify the ODS PDF statement options. The ODS PDF statement opens the PDF destination and the FILE= option specifies the PDF filename. The NOTOC option specifies that no table of contents is created. The STARTPAGE=NO option specifies that no new pages are inserted at the beginning of each procedure, or within certain procedures, even if new pages are requested by the procedure code. The STYLE= option specifies the style to use for the output, which is Styles.Mimictitle in this example.

ods pdf file="file.pdf" notoc startpage=no style=styles.mimictitle;

Specify a title.

title "Overriding the Default Procedure Title";
Create the MEANS procedure output and suppress the procedure title. The ODS NOPROCTITLE statement suppresses the writing of the procedure title that produces the results.

```sas
ods noproctitle;
proc means data=sashelp.cars;
   class cylinders;
   var mpg_city mpg_highway;
run;
```

Specify the text to use as the second procedure title. The TEXT= option specifies the text that appears above the PRINT procedure output. Because the custom style Styles.Mimictitle was specified in the ODS PDF statement, this text will look like a title specified by the TITLE= statement.

```sas
ods text="My Custom PROC PRINT Output Title";
```

Create the PRINT procedure output.

```sas
proc print data=sashelp.cars noobs;
   where mpg_highway gt 45;
run;
```

Close the PDF destination and open the HTML destination. The ODS PDF CLOSE statement closes the PDF destination and all of the files that are associated with it. You must close the destinations before you can view the output with a browser or before you can send the output to a physical printer. The ODS HTML statement opens the HTML destination and returns SAS to the default ODS destination.

```sas
ods pdf close;
ods html;
```

After your output is produced, you can remove the custom style. The DELETE statement in PROC TEMPLATE removes the custom style.

```sas
proc template;
   delete Styles.Mimictitle;
run;
```
PDF Output

The following image shows the text “My Custom PROC PRINT Output Title” above the PROC PRINT table in the same style as the title.

Output 6.72  PDF Output with Custom Procedure Title

Example 7: Toggling Page Breaks

Features:
- ODS PDF statement option
  - FILE=
  - NOTOC
  - STARTPAGE=NO
  - STARTPAGE=NOW
- ODS GRAPHICS statement
- ODS HTML

Other features:
- FOOTNOTE statement
- OPTIONS statement
- PROC REPORT
- PROC SGPLOT
- TITLE statement

Details

After pagination is turned off with STARTPAGE=NO, the setting stays in effect until it is overridden. If pagination is needed immediately and only once, the STARTPAGE=NOW setting is helpful. In the following example, PROC REPORT and PROC SGPLOT results are grouped onto two separate pages. The first REPORT table...
Program

ods html close;
options nodate;
ods pdf file="file.pdf" notoc startpage=no;

title "Top of the First Page Title";
footnote "Bottom of the first page footnote";

proc report data=sashelp.cars;
   col origin enginesize mpg_city;
   where mpg_highway gt 45;
run;

title;
footnote;
ods graphics on / reset noborder;
proc sgplot data=sashelp.cars;
   where mpg_highway gt 45;
   scatter x=enginesize y=mpg_city;
run;
ods pdf startpage=now;

title "Top of the Second Page Title";
footnote "Bottom of the second page footnote";

proc report data=sashelp.cars;
   col make model type;
   where mpg_highway gt 45;
run;

title;
footnote;
proc sgplot data=sashelp.cars;
   where mpg_highway gt 45;
   scatter x=enginesize y=mpg_highway;
run;

ods pdf close;
ods html;
Program Description

Close the HTML destination so that no HTML output is produced. The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources. If the destination were left open, then ODS would produce both HTML and PDF output.

```sas
ods html close;
```

Set the SAS system options.

```sas
options nodate;
```

Open the PDF destination and specify the ODS PDF statement options. The ODS PDF statement opens the PDF destination and the FILE= option specifies PDF filename. The NOTOC option specifies that no table of contents is created. The STARTPAGE=NO option specifies that no new pages are inserted at the beginning of each procedure, or within certain procedures, even if new pages are requested by the procedure code.

```sas
ods pdf file="file.pdf" notoc startpage=no ;
```

Specify the title and footnote for the first page of output. The title and footnote specified in this step are displayed at the top and bottom of the first page of output.

```sas
title "Top of the First Page Title";
footnote "Bottom of the first page footnote";
```

Create the REPORT procedure output for the first page.

```sas
proc report data=sashelp.cars ;
   col origin enginesize mpg_city;
   where mpg_highway gt 45;
run;
```

Clear the title and footnote. Specifying a blank TITLE statement prevents the title from printing before the SGPLOT procedure output. Specifying a blank FOOTNOTE statement prevents the footnote from printing after the SGPLOT procedure output.

```sas
title;
footnote;
```

Create the SGPLOT procedure output for the first page.

```sas
ods graphics on / reset noborder;
proc sgplot data=sashelp.cars;
   where mpg_highway gt 45;
   scatter x=enginesize y=mpg_city;
run;
```

Begin a new page and specify a title and footnote for the page. The STARTPAGE=NOW option forces the immediate insertion of a new page. The TITLE and FOOTNOTE statements specify a title and footnote for the new page.

```sas
ods pdf startpage=now;

   title "Top of the Second Page Title";
```
footnote "Bottom of the second page footnote";

Create the REPORT procedure output for the second page.

proc report data=sashelp.cars ;
  col make model type;
  where mpg_highway gt 45;
run;

Clear the title and footnote. Specifying a blank TITLE statement prevents the title from printing before the SGPLOT procedure output. Specifying a blank FOOTNOTE statement prevents the footnote from printing after the SGPLOT procedure output.

title;
footnote;

Create the SGPLOT procedure output for the second page.

proc sgplot data=sashelp.cars;
  where mpg_highway gt 45;
  scatter x=enginesize y=mpg_highway;
run;

Close the PDF destination and open the HTML destination. The ODS PDF CLOSE statement closes the PDF destination and all of the files that are associated with it. You must close the destinations before you can view the output with a browser or before you can send the output to a physical printer. The ODS HTML statement opens the HTML destination and returns SAS to the default ODS destination.

ods pdf close;
ods html;
PDF Output

If you had used STARTPAGE=YES instead of STARTPAGE=NOW in this example, a three-page PDF file is created with second SGPLOT output on the third page.

Output 6.73  PDF Output with Customized Title and Footnote Behavior

Example 8: Suppressing a Page Break

Features:

- ODS PDF statement option:
  
  FILE=
  NOTOC
  STARTPAGE=NO

- ODS GRAPHICS statement
- ODS HTML
- FOOTNOTE statement
- OPTIONS statement
- PROC REPORT
- PROC SGSCATTER
- TITLE statement

Details

The following example illustrates a basic behavior of the STARTPAGE= option. The STARTPAGE= option does not have to be specified on the first ODS PDF statement. In
this example the STARTPAGE=NO setting is made after the first PROC step and takes effect immediately to combine the PROC SGSCATTER results on the first page.

Also, this example illustrates that populated title and footnotes are included in the PROC SGSCATTER portion of the next example. Title and footnote text is embedded in any image created by SAS/GRAPH and ODS GRAPHICS output.

**Program**

```sas
ods html close;
options nodate;
ods pdf file="file.pdf" notoc;

title "Top Of the Page Title";
footnote "Bottom of the page footnote";

proc report data=sashelp.cars (obs=15);
   col origin enginesize mpg_city mpg_highway type cylinders ;
run;

ods pdf startpage=no;
ods graphics on / reset noborder;
title "Title Embedded In the Image";
footnote "Footnote embedded in the image";
proc sgscatter data=sashelp.cars;
   plot mpg_highway*weight mpg_city*weight;
run;

ods pdf close;
ods html;
```

**Program Description**

---

**Close the HTML destination so that no HTML output is produced.** The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources. If the destination were left open, then ODS would produce both HTML and PDF output.

```sas
ods html close;
```

---

**Set the SAS system options.**

```sas
options nodate;
```

---

**Open the PDF destination and specify a title and footnote.** The ODS PDF statement opens the PDF destination and the FILE= option specifies the PDF filename. The NOTOC option specifies that no table of contents is created. The TITLE and FOOTNOTE statements specify a title and footnote for the top and bottom of the page.

```sas
ods pdf file="file.pdf" notoc;

title "Top Of the Page Title";
footnote "Bottom of the page footnote";
```
Create the REPORT procedure output.

```sas
proc report data=sashelp.cars (obs=15);
  col origin enginesize mpg_city mpg_highway type cylinders;
run;
```

Prevent a page break. The STARTPAGE=NO option specified in the ODS PDF statement prevents a page break.

```sas
ods pdf startpage=no;
```

Create the SGSCATTER procedure output and specify a title and footnote. The title and footnote specified before an ODS graphics procedure step are embedded in the image.

```sas
ods graphics on / reset noborder;
  title "Title Embedded In the Image";
  footnote "Footnote embedded in the image";
  proc sgscatter data=sashelp.cars;
    plot mpg_highway*weight mpg_city*weight;
  run;
```

Close the PDF destination and open the HTML destination. The ODS PDF CLOSE statement closes the PDF destination and all of the files that are associated with it. You must close the destinations before you can view the output with a browser or before you can send the output to a physical printer. The ODS HTML statement opens the HTML destination and returns SAS to the default ODS destination.

```sas
ods pdf close;
ods html;
```
PDF Output

**Output 6.74** Preventing a Page Break in PDF Output

### Top Of the Page Title

<table>
<thead>
<tr>
<th>Origin</th>
<th>Engine Size (L)</th>
<th>MPG (City)</th>
<th>MPG (Highway)</th>
<th>Type</th>
<th>Cylinders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>3.5</td>
<td>17</td>
<td>23</td>
<td>SUV</td>
<td>6</td>
</tr>
<tr>
<td>Asia</td>
<td>2</td>
<td>24</td>
<td>31</td>
<td>Sedan</td>
<td>4</td>
</tr>
<tr>
<td>Asia</td>
<td>2.4</td>
<td>22</td>
<td>29</td>
<td>Sedan</td>
<td>4</td>
</tr>
<tr>
<td>Asia</td>
<td>3.2</td>
<td>20</td>
<td>28</td>
<td>Sedan</td>
<td>6</td>
</tr>
<tr>
<td>Asia</td>
<td>3.5</td>
<td>18</td>
<td>24</td>
<td>Sedan</td>
<td>6</td>
</tr>
<tr>
<td>Asia</td>
<td>3.5</td>
<td>18</td>
<td>24</td>
<td>Sedan</td>
<td>6</td>
</tr>
<tr>
<td>Asia</td>
<td>3.2</td>
<td>17</td>
<td>24</td>
<td>Sports</td>
<td>6</td>
</tr>
<tr>
<td>Europe</td>
<td>1.8</td>
<td>22</td>
<td>31</td>
<td>Sedan</td>
<td>4</td>
</tr>
<tr>
<td>Europe</td>
<td>1.8</td>
<td>23</td>
<td>30</td>
<td>Sedan</td>
<td>4</td>
</tr>
<tr>
<td>Europe</td>
<td>3</td>
<td>20</td>
<td>23</td>
<td>Sedan</td>
<td>6</td>
</tr>
<tr>
<td>Europe</td>
<td>3</td>
<td>17</td>
<td>25</td>
<td>Sedan</td>
<td>6</td>
</tr>
<tr>
<td>Europe</td>
<td>3</td>
<td>18</td>
<td>25</td>
<td>Sedan</td>
<td>6</td>
</tr>
<tr>
<td>Europe</td>
<td>3</td>
<td>20</td>
<td>27</td>
<td>Sedan</td>
<td>6</td>
</tr>
<tr>
<td>Europe</td>
<td>3</td>
<td>18</td>
<td>25</td>
<td>Sedan</td>
<td>6</td>
</tr>
<tr>
<td>Europe</td>
<td>3</td>
<td>20</td>
<td>27</td>
<td>Sedan</td>
<td>6</td>
</tr>
</tbody>
</table>

### Title Embedded In the image

#### Footnote embedded in the image

### Bottom of the page footnote
See Also

Other ODS Features

• “The DOCUMENT Procedure” in SAS Output Delivery System: Procedures Guide
• For complete documentation about styles, see “Understanding Styles, Style Elements, and Style Attributes ” on page 913.
• For a table of style elements affecting the table of contents and table of pages, see Table 11.4 on page 967.
• For a table of style attributes, see Table 12.1 on page 994.

Statements

• “ODS PCL Statement” on page 565
• “ODS PRINTER Statement ” on page 690
• “ODS PS Statement” on page 717

ODS POWERPOINT Statement

Opens, manages, or closes the ODS destination for PowerPoint, which produces PowerPoint output.

Valid in: Anywhere

Category: ODS: Third-Party Formatted

Defaults: The default style for PowerPoint is PowerPointLight. The default device for PowerPoint is PNG.

Interaction: By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC CONTENTS), ODS formats the output in SAS Monospace font. If you are creating output that is viewed in an operating environment where SAS software is not installed, this output is not displayed correctly. This is because without SAS, the SAS Monospace font is not recognized. To make your document display correctly, include the following statement before your SAS program:

```sas
OPTIONS FORMCHAR="|----|+|---+=|-=/<>*";
```

z/OS specifics: On z/OS, the ODS destination for PowerPoint only works with the HFS file system. You must use the FILESYSTEM=HFS option. The external file specified by the FILE= option must be an HFS file. If the WORK= option is used, the directory must be an HFS directory.

Tip: You must use either the PowerPointLight or PowerPointDark style, or a style that inherits from one of these two styles. The use of any other style will not produce satisfactory results.

Examples:

“Example 1: Creating Content for PowerPoint Slides” on page 639
“Example 2: ODS PowerPoint Using Layouts, Styles, and Background Gradients” on page 641
“Example 3: Customizing the Background and Transition between the PowerPoint Slides” on page 649
“Example 4: PowerPoint Slides with Background Image Specified by Path ” on page 651
Syntax

ODS POWERPOINT <\(\text{<ID=> identifier}> <\text{action}> ;
ODS POWERPOINT <\(\text{<ID=> identifier}> <\text{option(s)}> ;

Summary of Optional Arguments

AUTHOR='text-string'
Specify the author of the PowerPoint document

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
Specify how to measure the width of cells. Use to override the default value of BOX_SIZING for a destination

CATEGORY='text-string'
Specify the category of the PowerPoint document

CLOSE
Close the destination and the file that is associated with it

COMMENTS='text-string'
Add comments to the properties of the PowerPoint document

EXCLUDE exclusion(s) | ALL | NONE
Exclude output objects from the destination

FILE='file-specification'
Specify the file that contains the PowerPoint created by the destination

GFOOTNOTE | NOGFOOTNOTE
Control the location where footnotes are printed in the graphics output

GTITLE | NOGTITLE
Control the location where titles are printed in the graphics output

IMAGE_DPI
Specify the image resolution for the graphical output

KEYWORDS='text-string'
Add keywords to the PowerPoint document properties

LAYOUT=layout-name
Specify a predefined gridded layout

OPTIONS ( ADVANCE_AFTER= | ADVANCE_ON_MOUSE_CLICK= | BACKGROUNDCOLOR= | BACKGROUNDIMAGE= | BACKGROUNDREPEAT= | DURATION= | EFFECT_OPTION= | SOUND= | TRANSITION= )
Specify suboptions and a named value when using the ODS POWERPONT destination

SASDATE
Insert the standard SAS date in the document in place of the default PowerPoint date and time field

SELECT selection(s) | ALL | NONE
Select output objects for the destination

SHOW
Write to the SAS log the current selection or exclusion list for the destination

STATUS=‘text-string’
Insert the status of the PowerPoint document

STYLE= style-override(s)
Specifies one or more style-override(s) to use when writing output files

TITLE=‘text-string’
Specify a title for the PowerPoint document

WORK=‘fileref’ | ‘directory-name’
Specify an alternate directory for the temporary files

Without Arguments
If you use the ODS POWERPOINT statement without an action or options, then it opens the ODS destination for PowerPoint and creates PowerPoint output.

Actions
The following actions are available for the ODS POWERPOINT statement.

CLOSE
   closes the destination and any files that are associated with it.

   Tip  When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

EXCLUDE exclusion(s) | ALL | NONE
   excludes one or more output objects from the destination.

   Default  NONE

   Restriction  A destination must be open for this action to take effect.

   See  “ODS EXCLUDE Statement” on page 321

SELECT selection(s) | ALL | NONE
   selects output objects for the specified destination.

   Default  ALL

   Restriction  A destination must be open for this action to take effect.

   See  “ODS SELECT Statement” on page 758

SHOW
   writes the current selection list or exclusion list for the destination to the SAS log.

   Restriction  The destination must be open for this action to take effect.

   Tip  If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

   See  “ODS SHOW Statement” on page 771
Optional Arguments

**AUTHOR='text-string'**
specifies the author of the PowerPoint document. This information can be seen in the document properties.

**BOX_SIZING=(CONTENT_BOX | BORDER_BOX)**
specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry. For information about using the SAS Registry, see “Changing SAS Registry Settings for ODS” on page 40.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing.

**CATEGORY='text-string'**
specifies the category of the PowerPoint document. This information can be seen in the document properties.

**COMMENTS='text-string'**
adds comments to the properties of the PowerPoint document. This information can be seen in the document properties.

**FILE='file-specification'**
specifies the file that contains the PowerPoint created by the destination.

`'file-specification'`
specifies the file or fileref to receive output.

`file-specification` is one of the following:

- **external-file**
is the name of an external file to receive output.

  Requirement You must enclose `external-file` in quotation marks.

- **fileref**
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

  Default ODS uses the filename that is specified in the SAS registry. The default filename for the ODS destination for PowerPoint is “saspres.pptx”.

**GFOOTNOTE | NOGFOOTNOTE**
controls the location where footnotes are printed in the graphics output.

- **GFOOTNOTE**
  prints footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

- **NOGFOOTNOTE**
  prints footnotes that are created by ODS, which appear outside the graph borders.

  Default GFOOTNOTE

  Restriction This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.
GTITLE | NOGTITLE
controls the location where titles are printed in the graphics output.

GTITLE
prints the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.

NOGTITLE
prints the title that is created by ODS, which appears outside of the graph borders.

Default GTITLE

Restriction This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

See For details about the SAS/GRAPH TITLE statement, see “TITLE, FOOTNOTE, and NOTE Statements” in SAS/GRAPH: Reference.

IMAGE_DPI
specifies the image resolution for graphical output.

Alias DPI=

Default 150

CAUTION Caution: When using high DPI= or DPI_IMAGE values (values over 1000), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500M or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.

KEYWORDS='text-string'
provides keywords in the PowerPoint document. This information can be seen in the document properties.

Note: The KEYWORDS values are listed next to “Tags” in the properties pane.

LAYOUT=layout-name
specifies a predefined gridded layout. The layout-name value is one of the following:

Titleslide
specifies the layout for the title slide.

TitleandContent
specifies a slide with a title and content.

TwoContent< (ADVANCE=options)>
specifies slides with a two-column layout. The options that can be specified for ADVANCE= are listed below.

ADVANCE= BYGROUP | OUTPUT | PROC | TABLE
specifies that the grid is dynamically populated by groups, tables, output, and procedures. If nothing is specified for TwoContent, the grid is populated by PROC.
BYGROUP specifies that the gridded layout dynamically advance to a new region for every BYGROUP encountered.

Example

```bash
ods powerpoint file="test_bygroup.pptx"
layout=twocontent(advance=bygroup);
```

OUTPUT specifies that the gridded layout dynamically populates the region by output before moving on to the next region.

Example

```bash
ods powerpoint file="test_proc.pptx"
layout=twocontent(advance=output);
```

PROC specifies that the gridded layout dynamically populates the region by procedures before moving on to the next region.

Default PROC is the default option.

Example

```bash
ods powerpoint file="test_proc.pptx"
layout=twocontent(advance=proc);
```

TABLE specifies that the gridded layout dynamically populates the region by tables before moving on to the next region.

Example

```bash
ods powerpoint file="test_proc.pptx"
layout=twocontent(advance=table);
```

Tips ODS LAYOUT statements for gridded layout work with the ODS destination for PowerPoint.

When using the LAYOUT= option and OPTIONS options, you will want to specify the LAYOUT option and the OPTIONS option each time you change these options.

OPTIONS ( ADVANCE_AFTER= | ADVANCE_ON_MOUSE_CLICK= | BACKGROUND_COLOR= | BACKGROUND_IMAGE= | BACKGROUND_REPEAT= | DURATION= | EFFECT_OPTION= | SOUND= | TRANSITION= )

specifies suboptions and a named value when using the ODS POWERPOINT destination. The OPTIONS option is new in the third release of SAS 9.4.

ADVANCE_AFTER=seconds specifies the number of seconds to display the current slide before automatically proceeding to the next slide. This option is used to create presentations that advance automatically.

Note You can specify a fraction, such as `advance_after='2.5'`.

Example

```bash
ods powerpoint options(advance_after='5');
```

ADVANCE_ON_MOUSE_CLICK="YES" | "NO"
is used in conjunction with the ADVANCE_AFTER= option. The default is “YES”. By default PowerPoint waits for a mouse click to proceed to the next slide. Specify ADVANCE_ON_MOUSE_CLICK="NO" to disable this and
proceed automatically. Leave ADVANCE_ON_MOUSE_CLICK="YES" to allow a mouse click to advance to the next slide.

Example  ods powerpoint options {advance_on_mouse_click='yes'};

BACKGROUND_COLOR=color
specifies the color of the background of the slides.

Tip When present, the value of this option overrides the BACKGROUND_COLOR attribute of the BODY style element in the style for all the slides created after the option was used.

See See “color” on page 1037.

Example ods powerpoint file="test.pptx" options(backgroundcolor="#93ACFF" transition="push" effect_option="from_left" );
proc print data=sashelp.class(obs=5);
run;

Example “Example 3: Customizing the Background and Transition between the PowerPoint Slides” on page 649

BACKGROUND_IMAGE='file-specification' | 'LINEAR-GRADIENT' | 'RADIAL-GRADIENT'
specifies an image or a gradient to use as the background for the slides.

'file-specification'
is the name of a GIF, JPEG, or PNG file. Use a filename, a fileref, or a complete path. However, placing image files in a local directory and then specifying a filename might be the simplest approach.

file-specification is one of the following:

external-file
is the name and optional path of an external image file.

Examples ods powerpoint options
( backgroundimage="ppt-logo-background.gif");
ods powerpoint options
(backgroundimage='c:\Public\green.jpg');

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

Example filename backgrnd "tilepattern.gif";
ods powerpoint options(backgroundimage="backgrnd" backgroundrepeat="repeat");

Requirement You must enclose file-specification in quotation marks.

Example “Example 8: Creating Transition Effects for PowerPoint Slides” on page 658

LINEAR-GRADIENT (side-or-corner | angle, color-stop [, color-stop]+)
A linear gradient is defined by a gradient line on which the color-stops are arranged. One way to describe the gradient line is by naming a side or corner.
Another way is to describe the gradient line as a line passing through the center of the gradient box and rotated by an angle.

**side-or-corner, color-stop**

The gradient line begins at the named side or corner and extends to the opposite side or corner. For example, if the starting point is “top”, the gradient ray extends to the bottom.

The following are the side and corner names:

- bottom
- left
- right
- top
- topleft
- topright
- bottomleft
- bottomright

**Default**

By default the first color-stop is the starting side or corner.

**Notes**

The side and corner names do not require quotation marks.

The gradient must include at least two color-stops.

**Example**

linear-gradient (top, black, white);

**angle, color-stop[+, color-stop]**

You can describe the gradient line as a line passing through the center of the gradient box and rotated by an angle. The angle is measured in degrees. The angle 0 degrees points to the right and 90 degrees points up. Positive angles go counterclockwise. In the direction of the angle, the ending-point is the point on the gradient-line where a line drawn perpendicular to the gradient-line would intersect the corner of the box in that direction. The starting-point is determined identically, except in the opposite direction of the angle. See Figure 6.1 on page 638.

The gradient must include at least two color stops. At each color-stop, the line perpendicular to the gradient line has the specified color. Between color stops, the color is linearly interpolated between the colors of the adjacent color stops.

**Default**

By default the first color-stop is the starting side or corner.

**Notes**

The gradient must include at least two color stops.

The side and corner names do not require quotation marks.

**Example**

linear-gradient (45deg, #98f5ff, #7eb6ff, #3a5fcd, #4d71a3, #3a66a7);

linear-gradient(180deg, purple 30%, yellow 75%);

**color-stop <position>**

A color-stop argument specifies a color. The position argument is optional. If a position is present, it is specified as a percentage between
0% and 100%. At each color-stop, the gradient line or ray is the color of the color-stop. Between two color-stops, the color is linearly interpolated between the colors of the two color stops.

The color in a color-stop argument can be specified by name or by RGB value. The color might be a CSS color name such as red or yellow. The value might also be specified in the hexadecimal notation #0000FF or in a functional form such as rgba(). In hexadecimal notation, the red, green, and blue color channels are specified as hexadecimal values between 00 and FF. For example, the color yellow is specified as #FFFF00. The functional form rgba() supports an additional alpha channel. In this form, the red, green, and blue channels can be specified either as decimal values between 0 and 255, or percentages between 0% and 100%. The alpha channel is specified as a fraction between 0 and 1, where 0 is transparent and 1 is opaque. For example, rgba(100%, 0%, 0%, 1) is opaque red. The value rgba(100%, 100%, 0%, 0.1) is a very transparent yellow.

Default By default the first color-stop is the starting side or corner.

Note The side and corner names do not require quotation marks.

Example linear-gradient(180deg, purple 30%, yellow 75%);

Examples “Example 6: PowerPoint Slides with Linear Gradient Filled Background” on page 654

“Example 4: PowerPoint Slides with Background Image Specified by Path” on page 651

RADIAL-GRADIENT (<shape> size position, color-stop[, color-stop]+)
A radial gradient is defined by a gradient ray that starts in the center of the gradient and extends outward. Color-stops are positioned along the gradient ray starting in the center of the gradient toward the right. At each color-stop, the circle surrounding the center of the gradient has the specified color. Between color-stops the color is linearly interpolated between the colors of the adjacent color-stops.

Note: These three elements can be specified in any order.

<shape>
If present it can be either CIRCLE or ELLIPSE. The destination for PowerPoint always uses CIRCLE no matter which shape is specified or if no shape is specified.

Notes The gradient must include at least two colors.

In the radial gradient syntax, when shape is specified, there is no comma before listing size and position.

Example radial-gradient(circle 100% 100%, aqua, yellow)

Example “Example 8: Creating Transition Effects for PowerPoint Slides” on page 658
size

The gradient is always sized to meet the corner of the box farthest from the center. This sizing is called “farthest-corner” or “cover” in the CSS language.

Note: The SIZE portion of the specification, if present, is also ignored.

Example  radial-gradient(center, #fff200, #ff7a00 45%, #ff0300 70%, #4d0808)

position

The position designates the center of the gradient in the box in which the gradient appears. It can be center or one of the corners topleft, topright, bottomleft, or bottomright.

The position can also be specified as a pair of percentages. In the case, the first percentage is the distance down from the top of the box percentage and the second percentage is the distance to the right from the left side. That is, “0% 0%” is the top left corner, “50% 50%” is the center, and “100% 100%” is the bottom right corner. Although you can specify any positive integer between 0 and 100 for each value in the pair, a pair of percentages that does not exactly describe a corner or the center, will be adjusted to the nearest corner or to the center.

Example  radial-gradient(center, #fff200, #ff7a00 45%, #ff0300 70%, #4d0808)

color-stop <position>

A color-stop specifies a color. The position argument is optional. If a position is present, it is specified as a percentage between 0% and 100%. At each color-stop, the gradient line or ray is the color of the color-stop. Between two color-stops, the color is linearly interpolated between the colors of the two color-stops.

The color in a color-stop can be specified by name or by RGB value. The color might be a CSS color name such as red or yellow. The value might also be specified in the hexadecimal notation #RRGGBB or in a functional form such as rgba(). In hexadecimal notation, the red, green, and blue color channels are specified as hexadecimal values between 00 and FF. For example, the color yellow is specified as #FFFF00. The functional form rgba() supports an additional alpha channel. In this form, the red, green, and blue channels can be specified either as decimal values between 0 and 255, or percentages between 0% and 100%. The alpha channel is specified as a fraction between 0 and 1, where 0 is transparent and 1 is opaque. For example, rgba(100%, 0%, 0%, 1) is opaque red. The value rgba(100%, 100%, 0%, 0.1) is a very transparent yellow.

If the first color-stop does not have a position it is assigned 0%. If the last color-stop does not have a position, it is assigned 100%. If any color-stop has a position that is less than the position of a color-stop before it in the list, it is assigned a position equal to the largest specified position of any color-stop before it. If any color-stop still does not have a position, then each run of adjacent color-stops without positions is evenly spaced between the preceding and following color-stops with positions.

Example  red
          blue 25%
          #dde6cf
Restrictions  The W3C gradient syntax supports repeating gradients. The ODS destination for PowerPoint does not.

PowerPoint supports rectangular and path gradients. The ODS destination for PowerPoint does not.

Note  The ODS destination for PowerPoint might silently reject gradient definitions that use unsupported syntax elements.

Tip  When present, the value of this option overrides the BACKGROUNDIMAGE attribute of the BODY style element in the style for all the slides created after the option was used.

Examples  “Example 4: PowerPoint Slides with Background Image Specified by Path ” on page 651

“Example 7: PowerPoint Slides with Radial Gradient Filled Background ” on page 656

BACKGROUNDREPEAT=option
specifies whether the image used for the slide background is repeated horizontally, vertically, both, or not repeated. option can be one of the following:

NO_REPEAT
specifies that the image is not repeated.

REPEAT
specifies that the image is repeated both horizontally and vertically.

Example  ods powerpoint options(backgroundimage="backgrnd" backgroundrepeat="repeat");

DURATION= 'seconds'
specifies a non-default value for the duration of the transition. All transitions have a default duration. Use this option to specify a non-default value for the duration of the transition. The option value is a decimal number of seconds. For example, duration="1.5" sets the duration to one and a half seconds.

ods powerpoint options(duration="1.5");

EFFECT_OPTION= 'options'
Specifies a variation of the transition. All of the EFFECT_OPTION options supported by Microsoft PowerPoint 2013 are supported by the ODS POWERPOINT destination. See Table A5.1 on page 1115 for a list of the options supported for the EFFECT_OPTION.

Default  See Table A5.1 on page 1115.

Tip  If the name of the effect option used by PowerPoint contains a blank, substitute an underscore. For example, if PowerPoint calls the effect option Through Black, use EFFECT OPTION="THROUGH_BLACK". EFFECT OPTION="NONE" specifies that the default effect option takes effect for the transition.
Example

```ods powerpoint file="test.pptx"
  options(backgroundcolor="#93ACFF"
    transitions="push"
    effect_option="from_left" );
  proc print data=sashelp.class(obs=5);
  run;
```

Examples

- “Example 3: Customizing the Background and Transition between the PowerPoint Slides” on page 649
- “Example 8: Creating Transition Effects for PowerPoint Slides” on page 658

**SOUND= 'filename'**

specifies the name of a .wav file to play during the transition.

**filename**

specifies the name of a .wav file to play during the transition. *Filename* can be either a fileref or the path to a .wav file. For example, if the fileref MYSOUND is associated with the file `c:sounds\thunder.wav`, you can specify `SOUND="MYSOUND"`. Alternatively, you can specify the path explicitly: `SOUND="c:sounds\thunder.wav"`.

Example

```sound="c:sounds\thunder.wav"
```

**TRANSITION= '<name>' | 'NONE'**

specifies the name of a transition to be used between slides. The transition applies starting with the next slide created and is applied between every succeeding slide until another transition is specified or **TRANSITION="NONE"** is specified.

**<NAME>**

can be any of the 47 transitions supported by PowerPoint 2013. See Table A5.1 on page 1115 for a list of the transition names.

**Note:** Many of the Microsoft PowerPoint 2013 transitions are not supported by Microsoft PowerPoint 2010. If you try to show a presentation containing a Microsoft PowerPoint 2013 transition with Microsoft PowerPoint 2010, the results are unpredictable.

**'NONE'**

specifies that the default transition is used.

**Requirement**

*suboption(s)* must be enclosed in parentheses.

**Tips**

Option values stay in effect until you change them. If you specify **TRANSITION="fade"**, all slides will get the fade transition until you specify a different transition or you specify **TRANSITION="none"**. Because, effect options work with transitions, if you specify **EFFECT_OPTION="through_black"**, all transitions get that effect option as well until you specify another effect option or **EFFECT_OPTION="none"**.

If a procedure creates multiple slides, the current transition is applied to all the slides.

**See**

See Table A5.1 on page 1115 for a list of the transition names.
Example

```sas
ods powerpoint file="test.pptx"
options(backgroundcolor="#93ACFF"
    transition="push"
    effect_option="from_left" );
proc print data=sashelp.class(obs=5);
run;
```

Examples

"Example 3: Customizing the Background and Transition between the PowerPoint Slides" on page 649

"Example 8: Creating Transition Effects for PowerPoint Slides" on page 658

SASDATE

inserts the standard SAS date in the document in place of the default PowerPoint date and time field. PowerPoint formats this date field using the format specified by PowerPoint’s Header and Footer dialog. The date field is updated whenever you open the presentation. When the SASDATE option is used, instead of a date field, the ODS destination for PowerPoint inserts the date and time that you started your SAS session. PowerPoint does not update the date and time.

STATUS='text-string'

specifies the status of the PowerPoint document. This information can be seen in the document properties.

STYLE= style-override(s)

specifies one or more style-override(s) to use when writing output files. PowerPointLight is the default style for PowerPoint documents. However, you can also use PowerPointDark or a style derived from one of these two styles.

You can specify a style override in two ways:

- Specify a style element. A style element is a collection of style attributes that apply to a particular part of the output for a SAS program.
- Specify a style attribute. A style attribute is a name-value pair that describes a single behavioral or visual aspect of a piece of output. This is the most specific method of changing the appearance of your output.

style-override(s) has the following form:

```
style-element-name | [style-attribute-name-1=style-attribute-value-1
<style-attribute-name-2=style-attribute-value-2 ...>]
```

Default

PowerPointLight is the default style. PowerPointDark is another style that was created specifically for the ODS destination for PowerPoint.

Note

There is no way to toggle styles within one presentation.

See

For a complete discussion about styles, see Chapter 9, “Overview,” on page 913.

TITLE='text-string'

specifies a title for the PowerPoint document. This information can be seen in the document properties.

WORK='fileref' | 'directory-name'

specifies an alternate directory for the temporary files. By default, the ODS destination for PowerPoint uses the SAS Work library to hold temporary files. The WORK= option specifies an alternate directory for the temporary files.
fileref

is a file reference that has been assigned to a directory. Use the FILENAME
statement to assign a fileref.

directory-name

is the name of the directory.

Details

Overview
PowerPoint presentations consist of a number of individual pages or *slides*. Slides can contain text, graphics, and tables. The ODS POWERPOINT statement opens, manages, or closes the ODS destination for PowerPoint and generates PowerPoint output that has the .pptx extension.

By default, the ODS destination for PowerPoint puts one output object (table, graph, output from PROC ODSTEXT and ODSLIST) on each slide and centers the output. The ODS destination for PowerPoint adds all of the titles and footnotes that have been defined. SAS renders the titles first, the footnotes next, then the content.

*Note:* When multiple titles and footnotes are defined and there is not enough room to render the table and its header, the ODS destination for PowerPoint issues a warning message and simply outputs one row of the table without including the header. The rest of the table is rendered on the slides that follow.

If the DATE and NUMBER options are in effect, the ODS destination for PowerPoint adds the date and time and the slide number to each slide. The date and time are formatted according to the settings specified by PowerPoint when the presentation is viewed.

Layout in PowerPoint
The ODS destination for PowerPoint supports gridded layout. Three predefined gridded layouts are supported. These gridded layouts correspond to the similarly named built-in PowerPoint layouts. The following are the predefined gridded layouts:

<table>
<thead>
<tr>
<th>Layout Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TitleSlide</td>
<td>The TitleSlide layout is used to create a title slide. You can have one or two titles on the slide. Specifying more than two titles produces undefined output. The titles are created by P statements in PROC ODSTEXT. All TITLE statements are ignored. Footnotes specified after the first footnote are ignored.</td>
</tr>
<tr>
<td>TitleAndContent</td>
<td>The TitleAndContent layout is very similar to the default (no layout) slides. The difference is that the number of output objects that fit on a slide appear on the slide. The first two titles and the first two footnotes specified appear in the layout. Others are ignored. If you specify the TitleAndContent layout, the ODS destination for PowerPoint uses that layout for all of the slides until you change or stop the layout. To return to the default layout state, specify LAYOUT=<em>NULL</em>.</td>
</tr>
<tr>
<td>TwoContent</td>
<td>The TwoContent layout creates two-column output. This layout is similar to specifying the COLUMNS=2 option when you are using ODS LAYOUT. It outputs one procedure in one column and another procedure in the other column, but does not wrap. The first two titles and the first two footnotes specified appear in the layout. Others are ignored. If you specify the TwoContent layout, the ODS destination for PowerPoint uses that layout for</td>
</tr>
</tbody>
</table>
all the slides until you change or stop the layout. To return to the default state, specify LAYOUT=_NULL_.

Here is example code where the Titleslide layout is used with PROC ODSTEXT:

```sas
ods powerpoint file="title-example.pptx" layout=titleslide;
proc odstext;
   p "My Presentation" / style=PresentationTitle;
   p "ABC Company"/ style=PresentationTitle2;
run;
ods powerpoint close;
```

The ODS destination for PowerPoint finishes the current slide and the subsequent output goes onto a new slide when any of the following conditions are met:

- You change the layout.
- You start using a layout after the output has been created with the default layout (no layout).
- You switch from a layout to no layout using LAYOUT=_NULL_.

**Note:** When using a predefined layout, a new slide is not created for each BY group.

Also, a new slide is not automatically created by a new PROC statement.

The PresentationTitle and PresentationTitle2 style classes are predefined style classes. These style classes are customized for use with the TitleSlide layout and the PowerPointLight and PowerPointDark style templates. You can modify these style classes using style overrides or you can use your own style classes. With the TitleSlide template, the ODS POWERPOINT statement ignores all title statements. The title slide can have at most one footnote. The title slide shows the date, time, and page number if the DATE and NUMBER options are in effect.

**Note:** You get only one title slide at a time using the TitleSlide layout. If you want to create multiple title slides in a row, you have to specify LAYOUT=TitleSlide between each ODSTEXT procedure.

For graphical file types supported by the PowerPoint destination, refer to Table 6.10 on page 349.

**Gradient Fills in the ODS Destination for PowerPoint**

There are two types of gradient fills for the ODS destination for PowerPoint, linear and radial. These gradients are specified using the BACKGROUNDIMAGE= option.

A linear gradient is defined by a gradient line on which the color stops are arranged. The gradient line begins at the named side or corner and extends to the opposite side or corner. For example, if the starting point is “top” then the gradient ray extends to the bottom. By default the first color stop is the starting side or corner.

As an alternative, you can describe the gradient line as a line passing through the center of the gradient box and rotated by an angle. The angle is measured in degrees. The angle 0 degree points to the right and 90 degree points up. Positive angles go counterclockwise. In the direction of the angle, the ending-point is the point on the gradient-line where a line drawn perpendicular to the gradient-line would intersect the corner of the box in that direction. The starting-point is determined identically, except in the opposite direction of the angle. See Figure 6.1 on page 638.
A radial gradient is defined by a gradient ray that starts in the center of the gradient and extends outward. Color-stops are positioned along the gradient ray starting in the center of the gradient toward the right. At each color-stop, the circle surrounding the center of the gradient has the specified color. The angle is measured in degrees. The angle 0 degree points to the right and 90 degree points up. Positive angles go counterclockwise. In the direction of the angle, the ending-point is the point on the gradient-line where a line drawn perpendicular to the gradient-line would intersect the corner of the box in that direction. The starting-point is determined identically, except in the opposite direction of the angle. See “Example 7: PowerPoint Slides with Radial Gradient Filled Background ” on page 656.

Cautions when using gradient fills are as follows:

1. CSS gradients are new. As of this writing, the W3C syntax is still being drafted. Consequently, each major web browser supports its own dialect. The gradient syntax used by the ODS destination for PowerPoint is based on a W3C draft recommendation from 17 February 2011. See the list of documents listed below.

2. Some elements of the W3C syntax, such as color-stop positions measured in pixels, are not supported. This chapter of the document describes the supported syntax. The supported syntax is described in BACKGROUNDIMAGE= on page 629.
3. The W3C gradient syntax supports repeating gradients. The ODS destination for PowerPoint does not.

4. PowerPoint supports rectangular and path gradients. The ODS destination for PowerPoint does not.

See the following documents from the World Wide Web Consortium (W3C) for more information.

- CSS Image Values and Replaced Content Module Level 3
- CSS3 Values and Units
- CSS Color Module Level 3

Examples

**Example 1: Creating Content for PowerPoint Slides**

**Features:**
- ODS POWERPOINT statement
- CLOSE
- PROC ODSLIST statement
- ITEM statement
- P statement
- STYLE= option

**Other features:**
- FOOTNOTE statement
- ODS HTML CLOSE statement
- TITLE statement
- goptions
- GMAP procedure

**Details**

The following example produces PowerPoint output.

**Program**

```sas
ods html close;

title1 'PowerPoint Using Template Layout Twocontent with ODSLIST/GMAP';
footnote "The ODS Output Destination for PowerPoint";
ods powerpoint file="Layout2List.ppt" layout=twocontent
     nogtitle nogfootnote;
proc odslist;
  item 'Pre-defined template';
  item 'Side-by-side output';
  item;
  p 'Use: ';
  list / style=[bullet=check];
  item 'Tables';
  item 'Graphs';
  item 'Lists';
  item 'Text';
end;
```
run;

goptions hsize=4.5in vsize=4.5in;
proc gmap map=maps.us data=maps.us all;
   id state;
   choro statecode/statistic=frequency discrete nolegend;
run;
quit;

ods _all_ close;

Program Description

**Close the HTML destination.** The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources.

```ods html close;
```

**Specify titles and specify footnotes.**

```title1 'PowerPoint Using Template Layout Twocontent with ODSLIST/GMAP';
footnote "The ODS Output Destination for PowerPoint";
```

**Create a PowerPoint file and two-column layout.**

```ods powerpoint file="Layout2List.ppt" layout=twocontent	nogtitle nogfootnote;
```

**Create a list for the left column of the slide.**

```proc odslist;
   item 'Pre-defined template';
   item 'Side-by-side output';
   item;
      p 'Use:';
      list / style=[bullet=check];
   item 'Tables';
   item 'Graphs';
   item 'Lists';
   item 'Text';
   end;
run;
```

**Create PROC GMAP output for the right column of the slide and close open destinations.**

```goptions hsize=4.5in vsize=4.5in;
proc gmap map=maps.us data=maps.us all;
   id state;
   choro statecode/statistic=frequency discrete nolegend;
run;
quit;

ods _all_ close;
```
Output

A PowerPoint slide is produced using PROC ODSLIST and PROC GMAP using a two-column layout.

Output 6.75  PowerPoint Slide with PROC ODSLIST and PROC GMAP

Example 2: ODS PowerPoint Using Layouts, Styles, and Background Gradients

Features:
ODS POWERPOINT statement
   LAYOUT= option
   STYLE= option
   NOGTITLE option
   NOGFOOTNOTE option
ESCAPECHAR statement
PROC ODSTEXT
   STYLE= option
P statement
   STYLE= option
LIST statement
   STYLE= option
ITEM statement
   STYLE= option
Other features:
   FOOTNOTE statement
   ODS HTML CLOSE statement
Details
This sample program shows how to use the different PowerPoint layouts and styles.

Program
ods html close;
title1 'PowerPoint - Various Layouts and Styles';
footnote 'The PowerPoint Destination';
proc template;
define style styles.example;
  parent= styles.powerpointlight;
  class body /
    backgroundimage="linear-gradient(top, white, white 41%, #8599D4 72%, #2E438C)";
  class headersandfooters /
    backgroundcolor=a00000000
    color=accent3;
  class data, table, headersandfooters /
    bordercolor=dark2;
end;
run;
ods escapechar = "^*;"
ods powerPoint file="powerptOptions.ppt" style=styles.example;
ods powerpoint layout=titleslide;
proc odstext;
p "The ODS Destination for PowerPoint" / style=presentationtitle;
p "The Power to Know^{super ^{unicode 00AE}}" / style=presentationtitle2;
run;
ods powerpoint layout=_null_;    
footnote '{style [color=white]The ODS Destination for PowerPoint}';
title 'Supported Output';
proc odstext;
list / style=[color=accent3];
  item 'Graphics';
  item 'Tables';
  item 'Lists';
  item 'Paragraphs';
  item 'ODS TEXT= output';
end;
run;
title "Features";
ods powerpoint layout=twocontent;
proc odstext;
  list;
    item 'Light and dark styles';
Example 2: ODS PowerPoint Using Layouts, Styles, and Background Gradients

p 'Gradient backgrounds';
list / style=[liststyletype=check];
   item 'Linear';
   item 'Radial';
end;
end;
run;

proc odstext;
list;
item;
p 'Template layouts';
list / style=[liststyletype=decimal];
   item 'Titleslide';
   item 'TitleandContent';
   item 'TwoContent';
end;
end;
item 'Graphics support';
item 'Layout Support';
item 'Images';
end;
run;
ods powerpoint layout=twocontent;
title1 "2-Column Layout";
proc means data=sashelp.class min max ;
run;

goptions hsize=3in vsize=3in dev=png;
pattern color="#a78d84";

proc gchart data=sashelp.class;
vbar age / name='pptall0'
color="#fba16c"
coutline="red";
run;
quit;
ods powerpoint close;

Program Description

Close the HTML destination and specify the titles and footnotes. The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources.

ods html close;
title1 'PowerPoint - Various Layouts and Styles';
footnote 'The PowerPoint Destination';

Create a custom style to use for all slides.

proc template;
define style styles.example;
   parent= styles.powerpointlight;
   class body /
Set the escape character. The escape character is used to add style options.

```plaintext
ods escapechar = "\"; 
```

Open the ODS destination for PowerPoint and specify the options. The FILE= option specifies a filename for the PowerPoint slide. The STYLE= option specifies Styles.Example as the style to apply to the output.

```plaintext
ods powerPoint file="powerptOptions.ppt" style=styles.example; 
```

Create the title slide. The LAYOUT= output specifies TITLESLIDE as the layout.

```plaintext
ods powerpoint layout=titleslide; 
```

Create content for the title slide. The ODSTEXT procedure creates output for the title slide.

```plaintext
proc odstext; 
p "The ODS Destination for PowerPoint" / style=presentationtitle; 
p "The Power to Know"{super "{unicode 00AE}"* / style=presentationtitle2; 
run; 
```

Reset the layout template and specify a title and footnote. The ODS ESCAPECHAR style function sets the styles for the footnote.

```plaintext
ods powerpoint layout=_null_; 
footnote '{style [color=white]The ODS Destination for PowerPoint}'; 
title 'Supported Output'; 
```

Create the content for the second PowerPoint slide. The ODSTEXT procedure creates the content for the second slide.

```plaintext
proc odstext; 
list / style=[color=accent3]; 
  item 'Graphics'; 
  item 'Tables'; 
  item 'Lists'; 
  item 'Paragraphs'; 
  item 'ODS TEXT= output'; 
end; 
run; 
```

Create content for the first column of the third slide. The LAYOUT=TWOCOLUMN option in the ODS POWERPOINT statement specifies that the slide has two columns. The ODSTEXT procedure creates the text for the first column of slide three.

```plaintext
title "Features"; 
ods powerpoint layout=twocontent; 
proc odstext; 
```
list;
  item 'Light and dark styles';
  item;
    p 'Gradient backgrounds';
    list / style=[liststyletype=check];
      item 'Linear';
      item 'Radial';
    end;
  end;
run;

Create content for the second column. The second PROC ODSTEXT step creates the content for the second column of slide three.

proc odstext;
  list;
  item;
    p 'Template layouts';
    list / style=[liststyletype=decimal];
      item 'Titleslide';
      item 'TitleandContent';
      item 'TwoContent';
    end;
  end;
  item 'Graphics support';
  item 'Layout Support';
  item 'Images';
end;
run;

Create the fourth slide. The LAYOUT=TWOCONTENT option in the ODS POWERPOINT statement specifies that the fourth column has two columns. The PROC MEANS and PROC GCHART procedures create the content for the fourth slide.

ods powerpoint layout=twocontent;
title1 "2-Column Layout";
proc means data=sashelp.class min max ;
run;

goptions hsize=3in vsize=3in dev=png;
pattern color="#a78d84";

proc gchart data=sashelp.class;
  vbar age / name='pptall0'
    ctext="#fba16c"
    coutline="red";
run;
quit;

Close the PowerPoint destination.

  ods powerpoint close;

Output

The following PowerPoint slides provide information about what the ODS POWERPOINT statement can do.
The ODS Destination for PowerPoint

The Power to Know®
Supported Output

- Graphics
- Tables
- Lists
- Paragraphs
- ODS TEXT= output
Features

• Light and dark styles
• Gradient backgrounds
  ✓ Linear
  ✓ Radial

• Template layouts
  1. Titleslide
  2. TitleAndContent
  3. TwoContent

• Graphics support
• Layout Support
• Images
Example 3: Customizing the Background and Transition between the PowerPoint Slides

Features:
- ODS POWERPOINT statement
  - BACKGROUNDCOLOR option
  - TRANSITION option
  - EFFECT_OPTION option
- Other features:
  - PRINT statement
  - TITLE statement

Details
The following example produces PowerPoint output where the background of the worksheet has been customized. The OPTIONS suboptions BACKGROUNDCOLOR, TRANSITION, and EFFECT_OPTION have been used.

Program
```plaintext
ods html close;
title "Background Options";
title2 "Blue Background";
```
Program Description

**Close the HTML destination.** The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources.

```ods html close;
```

**Specify titles.**

```title "Background Options";
title2 "Blue Background";
```

**Create a PowerPoint file and customize the background.** Create a PowerPoint slide where the background is blue, the transition is push, and the effect is from the left.

```ods powerpoint file="test.pptx"
options {bgcolor="#93ACFF"
    transition="push"
    effect_option="from_left"};
```

**Print the customized data.**

```proc print data=sashelp.class;
run;
```

**Close the PowerPoint destination.**

```ods powerpoint close;
```

**Output**

A PowerPoint slide is produced using OPTIONS BACKGROUNDCOLOR, TRANSITION, and EFFECT.
Example 4: PowerPoint Slides with Background Image Specified by Path

Features:
- ODS POWERPOINT statement
  - BACKGROUNDIMAGE option
- TABULATE Procedure
- TITLE statement

Details
The following example produces PowerPoint output where the background of the worksheet has been customized using the BACKGROUNDIMAGE= suboption.

Program

```r
ods html close;
title1 'Image Background';
title2 'Specified By Path';
ods powerpoint options(backgroundimage='c:\Public\green.jpg');
proc tabulate data=sashelp.iris;
class species;
var petalwidth;
tables species, petalwidth*mean;
run;
ods powerpoint close;
```

Program Description
Close the HTML destination. The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources.

    ods html close;

Specify titles.

    title1 'Image Background';
    title2 'Specified By Path';

Create a PowerPoint file and customize the background with your own image.

    ods powerpoint options(backgroundimage='c:\Public\green.jpg');

Use PROC TABULATE to calculate the pedal width of Iris species

    proc tabulate data=sashelp.iris;
    class species;
    var petalwidth;
    tables species, petalwidth*mean;
    run;

Close the PowerPoint destination.

    ods powerpoint close;

Output

A PowerPoint slide is produced with OPTIONS BACKGROUNDIMAGE. The full path to the image is specified.

Output 6.81  PowerPoint Slide Specified by a Full Path to the Background Image
Example 5: PowerPoint Slides with Tile Background Image Specified by FILEREF

Features:
- ODS POWERPOINT statement
  - BACKGROUNDIMAGE option
  - BACKGROUNDREPEAT option
- FILEREF

Other features:
- PRINT statement
- TITLE statement

Details
The following example produces PowerPoint output where the background of the worksheet has been customized. The OPTIONS suboptions BACKGROUNDCOLOR and BACKGROUNDREPEAT have been used.

Program
```sas
ods html close;

title;
title2 'Tiled Image Background';
title3 'Specified By a Fileref';
filename backgrnd "tilepattern.gif";
ods powerpoint options(backgroundimage="backgrnd"
backgroundrepeat="repeat");

proc print data=sashelp.class(obs=5);
run;
ods powerpoint close;
```

Program Description

Close the HTML destination. The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources.
```sas
ods html close;
```

Specify titles.
```sas
title;
title2 'Tiled Image Background';
title3 'Specified By a Fileref';
```

Create a PowerPoint slide with a tiled background specified by using a fileref.
```sas
filename backgrnd "tilepattern.gif";
ods powerpoint options(backgroundimage="backgrnd"
backgroundrepeat="repeat");
```

Print the data with the customized background image.
```sas
proc print data=sashelp.class(obs=5);
run;
```
Close the PowerPoint destination.

ods powerpoint close;

Output

A PowerPoint slide is produced with OPTIONS BACKGROUNDIMAGE. The full path to the image is specified.

Output 6.82  PowerPoint Slide with OPTIONS BACKGROUNDIMAGE= Fileref to the Image

Example 6: PowerPoint Slides with Linear Gradient Filled Background

Features:
- ODS POWERPOINT statement
  - BACKGROUNDIMAGE option
  - LINEAR_GRADIENT option

Other features:
- PRINT statement
- TITLE statement

Details

The following example produces PowerPoint output where the background of the worksheet has been filled using a linear gradient.

Program

ods html close;

title;
title2 'Linear Gradient Background';
title3 'linear-gradient(top, #93acff, #82ff8e)';
Example 6: PowerPoint Slides with Linear Gradient Filled Background

ods powerpoint options(backgroundimage="linear-gradient(90deg,#93ACFF,#82FF8E));
proc print data=sashelp.class(obs=5);
run;
ods powerpoint close;

Program Description

Close the HTML destination. The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources.

    ods html close;

Specify titles.

    title;
    title2 'Linear Gradient Background';
    title3 'linear-gradient(top,#93ACFF,#82FF8E)';

Create a PowerPoint slide with a background that is filled by using the LINEAR_GRADIENT suboption.

    ods powerpoint options(backgroundimage="linear-gradient(90deg,#93ACFF,#82FF8E));

Print the data with the customized background image.

    proc print data=sashelp.class(obs=5);
run;

Close the PowerPoint destination.

    ods powerpoint close;

Output

The following example produces PowerPoint output where the background of the worksheet has been filled using a linear gradient.
Output 6.83  PowerPoint Slide with OPTIONS BACKGROUNDIMAGE= LINEAR_GRADIENT

Example 7: PowerPoint Slides with Radial Gradient Filled Background

Features:
- ODS POWERPOINT statement
  - BACKGROUNDIMAGE option
  - RADIAL_GRADIENT option

Other features:
- PRINT statement
- TITLE statement

Details

The following example produces PowerPoint output where the background of the worksheet has been filled using a linear gradient. The OPTIONS (BACKGROUNDIMAGE=) is used.

Program

ods html close;
title;
title2 'Radial Gradient Background';
title3 'radial-gradient(bottom left circle farthest-corner,' #ffff00, #00ffff 33%, #ffffff )';
ods powerpoint options(backgroundimage= "radial-gradient(bottom left circle farthest-corner, #ffff00, #00ffff 33%, #ffffff )");
proc print data=sashelp.class(obs=5);
run;
ods powerpoint close;

Program Description

Close the HTML destination. The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources.

```
ods html close;
```

Specify titles.

```
title;
title2 'Radial Gradient Background';
title3 'radial-gradient(bottom left circle farthest-corner,
                   #ffff00, #00ffff 33%, #ffffff)';
```

Create a PowerPoint slide with a background that is filled by using the RADIAL_GRADIENT suboption.

```
ods powerpoint options(backgroundimage=
                        "radial-gradient(bottom left circle farthest-corner,
                                       #ffff00, #00ffff 33%, #ffffff ) ");
```

Print the data with the customized background.

```
proc print data=sashelp.class(obs=5);
run;
```

Close the PowerPoint destination.

```
ods powerpoint close;
```

Output

The following example produces PowerPoint output where the background of the worksheet has been filled using a linear gradient.
Example 8: Creating Transition Effects for PowerPoint Slides

Features:
- ODS POWERPOINT statement options
  - BACKGROUNDIMAGE option
  - RADIAL_GRADIENT option
  - NOGTITLE
  - NOGFOOTNOTE
  - BACKGROUNDIMAGE=
  - TRANSITION=
  - EFFECT_OPTION=

Other features:
- DATA step
- ODS SELECT statement
- OPTIONS statement
- PRINT statement
- PROC GAM
- TITLE statement

Details
This example shows you how to use the ODS POWERPOINT statement to create transitional effects for your PowerPoint slides.

Program

data diabetes;
  input Age BaseDeficit CPeptide @@;
  logCP = log(CPeptide);
datalines;
Example 8: Creating Transition Effects for PowerPoint Slides

Create the input data set.

```plaintext
options nodate nonumber;
ods html close;
ods select fitsummary anodev smoothingcomponentplot;
ods powerpoint file="transition.pptx" nogtitle nogfootnote
   options(backgroundimage="radial-gradient(top left, white, white 36%, #C2C2C2 73%, #363636"
   transition="rotate" effect_option="from_top");
title 'Patterns of Diabetes';
footnote 'ODS Destination for PowerPoint';
proc gam data=diabetes;
   model logCP = spline(Age) spline(BaseDeficit);
run;
ods powerpoint close;
```

Program Description

Create the input data set.

```plaintext
data diabetes;
   input Age BaseDeficit CPeptide @@;
   logCP = log(CPeptide);
datalines;
5.2  -8.1  4.8  8.8  -16.1  4.1  10.5  -0.9  5.2
10.6  -7.8  5.5  10.4  -29.0  5.0  1.8  -19.2  3.4
12.7  -18.9  3.4  15.6  -10.6  4.9  5.8  -2.8  5.6
1.9  -25.0  3.7  2.2  -3.1  3.9  4.8  -7.8  4.5
7.9  -13.9  4.8  5.2  -4.5  4.9  0.9  -11.6  3.0
11.8  -2.1  4.6  7.9  -2.0  4.8  11.5  -9.0  5.5
10.6  -11.2  4.5  8.5  -0.2  5.3  11.1  -6.1  4.7
12.8  -1.0  6.6  11.3  -3.6  5.1  1.0  -8.2  3.9
14.5  -0.5  5.7  11.9  -2.0  5.1  8.1  -1.6  5.2
13.8  -11.9  3.7  15.5  -0.7  4.9  9.8  -1.2  4.8
11.0  -14.3  4.4  12.4  -0.8  5.2  11.1  -16.8  5.1
5.1  -5.1  4.6  4.8  -9.5  3.9  4.2  -17.0  5.1
6.9  -3.3  5.1  13.2  -0.7  6.0  9.9  -3.3  4.9
12.5  -13.6  4.1  13.2  -1.9  4.6  8.9  -10.0  4.9
10.8  -13.5  5.1;
```

options nodate nonumber;
ods html close;
ods select fitsummary anodev smoothingcomponentplot;
ods powerpoint file="transition.pptx" nogtitle nogfootnote
   options(backgroundimage="radial-gradient(top left, white, white 36%, #C2C2C2 73%, #363636"
   transition="rotate" effect_option="from_top");
title 'Patterns of Diabetes';
footnote 'ODS Destination for PowerPoint';
proc gam data=diabetes;
   model logCP = spline(Age) spline(BaseDeficit);
run;
ods powerpoint close;
```
Set the SAS system options and close the ODS HTML destination. The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources.

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

Select the output objects. The PROC GAM produces many output objects. The ODS SELECT statement specifies the output objects that are included in the output. Each output object is displayed on a separate slide.

```sas
ods select fitsummary anodev smoothingcomponentplot;
```

Open the ODS destination for PowerPoint and specify the options. The ODS POWERPOINT statement opens the ODS destination for PowerPoint. The BACKGROUNDIMAGE= creates the radial gradient.

```sas
ods powerpoint file="transition.pptx" nogtitle nogfootnote
   options(backgroundimage="radial-gradient(top left, white, white 36%, #C2C2C2 73%, #363636)",
            transition="rotate" effect_option="from_top");
```

Specify the title, specify the footnote, and create the procedure output.

```sas
title 'Patterns of Diabetes';
footnote 'ODS Destination for PowerPoint';
proc gam data=diabetes;
   model logCP = spline(Age) spline(BaseDeficit);
run;
```

Close the ODS destination for PowerPoint.

```sas
ods powerpoint close;
```
## Patterns of Diabetes

### The GAM Procedure

**Dependent Variable:** \( \text{logCP} \)

**Smoothing Model Component(s):** \( \text{spline(Age)} \) \( \text{spline(BaseDeficit)} \)

### Smoothing Model Analysis

Fit Summary for Smoothing Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Smoothing Parameter</th>
<th>DF</th>
<th>GCV</th>
<th>Num Unique Obs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(Age)</td>
<td>0.995582</td>
<td>3.000000</td>
<td>0.011675</td>
<td>37</td>
</tr>
<tr>
<td>Spline(BaseDeficit)</td>
<td>0.995299</td>
<td>3.000000</td>
<td>0.012437</td>
<td>39</td>
</tr>
</tbody>
</table>

**Output 6.85 Customized PowerPoint Slide 2**
Patterns of Diabetes

The GAM Procedure

Dependent Variable: logCP

Smoothing Model Component(s): spline(Age) spline(BaseDeficit)

Output 6.86  Transition from Slide 2 to Slide 3

Patterns of Diabetes

The GAM Procedure

Dependent Variable: logCP

Smoothing Model Component(s): spline(Age) spline(BaseDeficit)

Output 6.87  Customized PowerPoint Slide 3
ODS PREFERENCES Statement

Reverts the ODS settings back to start-up defaults.

Valid in: Anywhere
Category: ODS: Output Control
Tip: It is useful to specify this statement if you are creating graphics and have changed your default output directory to something other than Work. When you specify the ODS PREFERENCES statement, the default directory is set to Work, which is the original default. Any output is then generated in your Work folder.

Syntax

ODS PREFERENCES;

Without Arguments

The ODS PREFERENCES statement reverts the ODS back to the default behavior:

• HTML output is created.
• LISTING output is not created.
• Both HTML and graph image files are saved in the Work folder (and not your current directory).
• Default style is HTMLBlue.
• ODS Graphics is enabled.

The changes made by the ODS PREFERENCES statement last only for the duration of your SAS session, or until you change them with an ODS statement or option. The ODS PREFERENCES statement does not change the settings in the TOOLS ⇒ Options ⇒ Preferences ⇒ Results dialog box.

ODS PHTML Statement

Opens, manages, or closes the PHTML destination, which produces simple HTML output that uses twelve style elements and no class attributes for the presentation. Class attributes are used only for the justification.

Valid in: Anywhere
Category: ODS: Third-Party Formatted
Default: The default style for Markup family destinations is HTMLBlue.
### Syntax

**ODS PHTML** *action*;
**ODS PHTML** <option(s)> ;

### Summary of Optional Arguments

(ID= *identifier*)
Open multiple instances of the same destination at the same time

ANCHOR= 'anchor-name'
Specify a unique base name for the anchor tag that identifies each output object in the current body file

ARCHIVE='*string*'
Specify which applet to use to view ODS HTML output

ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)
Specify attributes to write between the tags that generate dynamic graphics output

BASE= 'base-text'
Specify text to use as the first part of all links and references that ODS creates in output files

BODY= 'file-specification' (suboption(s))
Open a markup family destination and specify the file that contains the primary output that is created by the ODS statement

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
Specify how to measure the width of cells. Use to override the default value of BOX_SIZING for a destination

CHARSET= *character-set*
Specify the character set to be generated in the META declaration for the HTML output

CLOSE
Close the destination and the file that is associated with it

CODE= 'file-specification' <(suboption(s))>
Open the HTML destination and specify the file that contains relevant style information

CODEBASE='*string*'
Create a file path that can be used by the GOPTIONS devices

CONTENTS= 'file-specification' <(suboption(s))>
Open the HTML destination and specify the file that contains a table of contents for the output

CSSSTYLE= 'file-specification'(media-type-1…media-type-10)>
Specify a cascading style sheet to apply to your output

DOM="*external-file*>
Specify that the ODS document object model is written to the SAS log or to an external file.

ENCODING= local-character-set-encoding
Override the encoding for input or output processing (transcodes) of external files

EVENT=event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )
Specify an event and the value for event variables that is associated with the event
EXCLUDE exclusion(s) | ALL | NONE
Exclude output objects from the destination

FRAME='file-specification' <(suboption(s))>
Specify the file that integrates the table of contents, the page contents, and the body file

GFOOTNOTE | NOGFOOTNOTE
Control the location where footnotes are printed in the graphics output

GPATH='aggregate-file-storage-specification' | fileref | libref.catalog (URL='Uniform-Resource-Locator' | NONE)
Specify the location for all graphics output that is generated while the destination is open

GTITLE | NOGTITLE
Control the location where titles are printed in the graphics output

HEADTEXT='markup-document-head'
Specify HTML tags to place between the <HEAD> and </HEAD> tags in all of the output files.

METATEXT='metatext-for-document-head'
Specify HTML code to use as the <META> tag between the <HEAD> and </HEAD> tags in all of the HTML output files.

NEWFILE=starting-point
Create a new body file at the specified starting point

OPTIONS ( DOC= | <suboption(s)> )
Specify tagset-specific suboptions and a named value

PACKAGE <package-name>
Specify that the output from the destination be added to an ODS package

PAGE='file-specification' <(suboption(s))>
Open the HTML destination and specify the file that contains a description of each page of the body file and contains links to the body file

PARAMETERS= (parameter-pair-1 ... parameter-pair-n)
Write the specified parameters between the tags that generate dynamic graphics output

PATH='aggregate-file-storage-specification' | fileref | libref.catalog (URL='Uniform-Resource-Locator' | NONE)
Specify the location of an aggregate storage location or a SAS catalog for all markup files

RECORD_SEPARATOR='alternative-separator' | NONE
Specify an alternative character or string to separate lines in the output files

SELECT selection(s) | ALL | NONE
Select output objects for the destination

SHOW
Write to the SAS log the current selection or exclusion list for the destination

STYLE=style-template
Specify a style template to use in writing output files

STYLESHEET='file-specification' <(suboption(s))>
Open the HTML destination and place style information for output into an external file, or read style sheet information from an existing file

TEXT=text-string
Insert text into your document

TRANTAB='translation-table'
Specify a translation table to use when transcoding a file for output
Without Arguments
If you use the ODS PHTML statement without an action or options, then it opens the PHTML destination and creates PHTML output.

Actions
The following actions are available for the ODS PHTML statement.

CLOSE
   closes the destination and any files that are associated with it.
   
   Tip   When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

EXCLUDE exclusion(s) | ALL | NONE
   excludes one or more output objects from the destination.
   
   Default   NONE
   
   Restriction   A destination must be open for this action to take effect.
   
   See   “ODS EXCLUDE Statement” on page 321

SELECT selection(s) | ALL | NONE
   selects output objects for the specified destination.
   
   Default   ALL
   
   Restriction   A destination must be open for this action to take effect.
   
   See   “ODS SELECT Statement ” on page 758

SHOW
   writes the current selection list or exclusion list for the destination to the SAS log.
   
   Restriction   The destination must be open for this action to take effect.
   
   Tip   If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.
   
   See   “ODS SHOW Statement” on page 771

Optional Arguments
The following options are available for the ODS PHTML statement, which is part of the markup family of statements.

ANCHOR= 'anchor-name'
   specifies a unique base name for the anchor tag that identifies each output object in the current body file.
   
   Each output object has an anchor tag for the contents, page, and frame files to reference. The links and references are automatically created by ODS. The links and references point to the name of an anchor. Therefore, each anchor name in a file must be unique.
anchor-name

is the base name for the anchor tag that identifies each output object in the current body file.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR= 'TABULATE', then ODS names the first anchor **tabulate**. The second anchor is named **tabulate1**; the third is named **tabulate2**, and so on.

**Restrictions**

Each anchor name in a file must be unique.

Only alphanumeric values, the special characters "$ - _ . + ! * ' () , " and reserved characters used for their reserved purposes can be used unencoded within a URL.

**Requirement**

You must enclose *anchor-name* in quotation marks.

**Interaction**

If you open a file to append to it, be sure to specify a new anchor name to prevent writing the same anchors to the file again. ODS does not recognize anchors that are already in a file when it opens the file.

**Tips**

You can change anchor names as often as you want by specifying the ANCHOR= option in a markup family statement anywhere in your program. After you have specified an anchor name, it remains in effect until you specify a new one.

Specifying new anchor names at various points in your program is useful when you want other web pages to link to specific parts of your markup language output. Because you can control where the anchor name changes, you know in advance what the anchor name is at those points.

An *anchor-name* must begin with a letter ([A-Za-z]) and can be followed by any number of letters, digits ([0-9]), hyphens (-), underscores (_), colons (:), and periods (.).

**ARCHIVE='string'**

specifies which applet to use to view the ODS HTML output. The ARCHIVE= option is valid only for the GOPTIONS Java device.

The string must be one that the browser can interpret. For example, if the archive file is local to the computer that you are running SAS on, you can use the FILE protocol to identify the file. If you want to point to an archive file that is on a web server, use the HTTP protocol.

**Default**

If you do not specify ARCHIVE= and you are using the JAVA device driver, ODS uses the value of the SAS system option APPLETOC=. There is no default if you are using the ACTIVEX device driver.

**Requirements**

You must enclose *string* in quotation marks.

The ARCHIVE attribute is a feature of Java 1.1. Therefore, if you are using the Java device driver, your browser must support this version of Java. Both Internet Explorer 4.01 and Netscape 4.05 support Java 1.1.
Interaction
Use ARCHIVE= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA or DEVICE=ACTIVEX option in the GOPTIONS statement.

Tips
Typically, this option should not be used, because the SAS server automatically determines the correct SAS/GRAPH applets to view the ODS HTML output. However, if you have renamed the JAR files, or have other applets with which to view the ODS HTML output, this option enables you to access these applets.

Use the CODEBASE= option to specify the file path. It is recommended that you do not put a file path in your ARCHIVE= option.

The value of APPLETOC= points to the location of the Java archive files that are shipped with the SAS system. To find out what the value of this option is, you can either look in the Options window in the Files folder under Environment Control, or you can submit the following procedure step:

```plaintext
proc options option=appletloc;
run;
```

ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)
writes the specified attributes between the tags that generate dynamic graphics output.

attribute-pair
specifies the name and value of each attribute. attribute-pair has the following form:

'attribute-name' = 'attribute-value'

attribute-name
is the name of the attribute.

attribute-value
is the value of the attribute.

Requirement
You must enclose attribute-name and attribute-value in quotation marks.

Interaction
Use the ATTRIBUTES= option in conjunction with SAS/GRAPH procedures and with the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

See
SAS/GRAPH: Reference for valid attributes for the Graph Applet, the Map Applet, the Contour Applet, and the MetaView Applet.

BASE= 'base-text'
specifies the text to use as the first part of all links and references that ODS creates in the output files.

base-text
is the text that ODS uses as the first part of all links and references that ODS creates in the file.

Consider this specification:

```plaintext
BASE= 'http://www.your-company.com/local-url/
```
In this case, ODS creates links that begin with the string http://www.your-
company.com/local-url/. The appropriate anchor-name completes the link.

Requirement You must enclose base-text in quotation marks.

BODY= 'file-specification' (suboption(s))
opens a markup family destination and specifies the file that contains the primary
output that is created by the ODS statement. These files remain open until you do
one of the following:

• close the destination with either an ODS markup-family-destination CLOSE
statement or ODS _ALL_ CLOSE statement.

• open the same destination with a second markup family statement. This closes
the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the
FILENAME statement to assign a fileref.

Restriction The BODY=fileref option cannot be used in conjunction with
the NEWFILE= option.

See For information about the FILENAME statement, see
"FILENAME Statement" in SAS Statements: Reference.

entry.markup
specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library
and catalog. See the discussion of the PATH= option.

(suboption(s))
specifies one or more suboptions in parentheses. Suboptions are instructions for
writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a
file.

See For complete documentation about the DYNAMIC suboption, see
“(DYNAMIC)” on page 688.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output
file.

See For complete documentation about the NO_BOTTOM_MATTER
suboption, see “(NO_BOTTOM_MATTER)” on page 688.
(NO_TOP_MATTER)
See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 689.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.
See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 689.

(URL='Uniform-Resource-Locator')
See For complete documentation about the URL= suboption, see “(URL='Uniform-Resource-Locator')” on page 690.

Alias FILE=

Interaction Using the BODY= option in an ODS markup family statement that refers to an open ODS markup destination forces ODS to close the destination and all associated files. ODS then opens a new instance of the destination. For more information see “Opening and Closing the MARKUP Destination” on page 517.

Note For some values of TAGSET=, this output is an HTML file. For other TAGSET= values, the output is an XML file, and so on.

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry. For information about using the SAS Registry, see “Changing SAS Registry Settings for ODS” on page 40.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing.

CHARSET= character-set
specifies the character set to be generated in the META declaration for the HTML output.
See For information about the CHARSET= option, see “CHARSET= Option” in SAS National Language Support (NLS): Reference Guide.

CODE='file-specification' <(suboption(s))>
opens a markup family destination and specifies the file that contains accompanying programming code, such as JavaScript or XSL (Extensible Stylesheet Language). These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:
**external-file**

is the name of an external output file.

**Requirement**

You must enclose *external-file* in quotation marks.

**fileref**

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

**See**

See “FILENAME Statement” in *SAS Statements: Reference*.

**entry.markup**

specifies an entry in a SAS catalog to write to.

**Interaction**

If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

**suboption(s)**

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

**(DYNAMIC)**

enables you to send output directly to a web server instead of writing it to a file.

**See**

For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 688.

**(URL= 'Uniform-Resource-Locator' )**

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

**See**

For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator' )” on page 690.

**CODEBASE='string'**

specifies the location of the executable Java applet or the ActiveX control file. *string* is specified as a pathname or as a URL. The CODEBASE file path option has two definitions, depending on the GOPTIONS device used.

When you generate web presentations with the JAVA and ActiveX device drivers, SAS generates HTML pages that automatically look for the JAVA archive files or the ActiveX control file in the default installation location.

For the ActiveX device:

If you use the ActiveX device driver with ODS to generate output containing an ActiveX control, then specify the CODEBASE= option in the ODS statement. The value of the CODEBASE= option should include the location and the version of the EXE file.

**Tip**

You do not need to specify the CODEBASE= option with the DEVICE=ACTIVEX option unless the users that view your output do not have the ActiveX control installed on their machine. When users that do not have the ActiveX control installed view your output, they are prompted to download the control.
See *SAS/GRAPH: Reference* for information about specifying the location of control and applet files using the CODEBASE= and ARCHIVE= options.

For the Java device:
If you use the Java device driver with ODS to generate output containing a SAS/GRAPH applet, specify the path to the JAR file with the CODEBASE= option in the ODS statement.

When you specify DEVICE=JAVA, the users that view your output must have access to the appropriate Java applet. By default, SAS sets the value of CODEBASE= to refer to the executable file for the applet that is automatically installed with SAS. The default location of the SAS Java archive files is specified by the APPLETLOC= system option. You do not need to specify the CODEBASE= option if both of the following conditions are true.

- The default location is accessible by users who are viewing your web presentation.
- The SAS Java archive is installed at that location.

Tip Specify only the directory of the JAR file. The CODEBASE= location can be specified as a pathname or as a URL.

See *SAS/GRAPH: Reference* for information about specifying the location of control and applet files using the CODEBASE= and ARCHIVE= options.

**CONTENTS= 'file-specification' <(suboption(s))>**
opens a markup family destination and specifies the file that contains a table of contents for the output. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

**file-specification**
specifies the file, fileref, or SAS catalog to write to.

**file-specification** is one of the following:

**external-file**
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

**fileref**
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in *SAS Statements: Reference.*

**entry.markup**
specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.
suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see "(DYNAMIC)" on page 688.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see "(NO_BOTTOM_MATTER)" on page 688.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see "(NO_TOP_MATTER)" on page 689.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

See For complete documentation about the TITLE= suboption, see "(TITLE='title-text')" on page 689.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see "(URL= 'Uniform-Resource-Locator')" on page 690.

CSSSTYLE= 'file-specification'<(media-type-1<…media-type-10>)> specifies a cascading style sheet to apply to your output.

file-specification specifies a file, fileref, or URL that contains CSS code.

file-specification is one of the following:

"external-file"
is the name of the external file.

Requirement You must enclose external-file in quotation marks.

fileref is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.
See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

"URL"
is a URL to an external file.

Requirement You must enclose URL in quotation marks.

(media-type-1<.. media-type-10>)specifies one or more media blocks that correspond to the type of media that your output is rendered on. CSS uses media type blocks to specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not contained in any media blocks. By using the media-type suboption, in addition to the general CSS code, you can import the section of a CSS file intended only for a specific media type.

Default If no media-type is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

Range You can specify up to ten different media types.

Requirements You must enclose media-type in parentheses.

You must specify media-type next to the file-specification specified by the CSSSTYLE= option.

Tip If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

Restriction The CSSSTYLE= option does not affect SAS/GRAPH output.

Requirement CSS files must be written in the same type of CSS produced by the ODS HTML statement. To view the CSS code that ODS creates, you can do one of the following:
• specify the ODS TRACE DOM statement
• specify the DOM option

Interaction If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

See For complete documentation about ODS and Cascading Style Sheets, see SAS Output Delivery System: Advanced Topics

Example “Example 6: Applying a CSS File to ODS Output” on page 527

DOM="external-file”>specifies that the ODS document object model is written to the SAS log or an external file.
**external-file**

is the name of an external output file.

**Requirement**

You must enclose *external-file* in quotation marks.

**See**

For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in *SAS Output Delivery System: Advanced Topics*.

**ENCODING=** `local-character-set-encoding`

overrides the encoding for input or output processing (transcodes) of external files.

**See**

For information about the ENCODING= option, see *SAS National Language Support (NLS): Reference Guide*.

**EVENT=** `event-name` *(FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )*

specifies an event and the value for event variables that are associated with the event.

***FILE=*** `BODY | CODE | CONTENTS | DATA | FRAME | PAGES | STYLESHEET`;

triggers one of the known types of output files that correspond to the BODY=, CODE=, CONTENTS=, FRAME=, PAGES=, and STYLESHEET= options.

***FINISH***

triggers the finish section of an event.

**See**

For information about events, see “Understanding Events” in *SAS Output Delivery System: Procedures Guide*.

***LABEL=*** `'variable-value'`

specifies the value for the LABEL event variable.

**Requirement**

*variable-value* must be enclosed in quotation marks.

**See**

For information about the LABEL event variable, see “Event Variables” in *SAS Output Delivery System: Procedures Guide*.

***NAME=*** `'variable-value'`

specifies the value for the NAME event variable.

**Requirement**

*variable-value* must be enclosed in quotation marks.

**See**

For information about the NAME event variable, see “Event Variables” in *SAS Output Delivery System: Procedures Guide*.

***START***

triggers the start section of an event.

**See**

For information about events, see “Understanding Events” in *SAS Output Delivery System: Procedures Guide*.

***STYLE=*** `style-element`

specifies a style element.

**See**

For information about style elements, see “Style Attributes Overview” in *SAS Output Delivery System: Procedures Guide*.
(TARGET='variable-value')
specifies the value for the TARGET event variable.

Requirement  
variable-value must be enclosed in quotation marks.

See  
For information about the TARGET event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(TEXT='variable-value')
specifies the value for the TEXT event variable.

Requirement  
variable-value must be enclosed in quotation marks.

See  
For information about the TEXT event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(URL='variable-value')
specifies the value for the URL event variable.

Requirement  
variable-value must be enclosed in quotation marks.

See  
For information about the URL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

Default  
(FILE='BODY')

Requirement  
The EVENT= option's suboptions must be enclosed in parentheses.

FRAME= 'file-specification' <(suboption(s))>
opens a markup family destination and, for HTML output, specifies the file that integrates the table of contents, the page contents, and the body file. If you open the frame file, then you see a table of contents, a table of pages, or both, as well as the body file. For XML output, FRAME= specifies the file that contains the DTD. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.

file-specification is one of the following:

external-file
is the name of an external output file.

Requirement  
You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See  
For information about the FILENAME Statement, see “FILENAME Statement” in SAS Statements: Reference.
**entry.markup**
specifies an entry in a SAS catalog to write to.

**Interaction**
If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

**suboption(s)**
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

**(DYNAMIC)**
enables you to send output directly to a web server instead of writing it to a file.

**See**
For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 688.

**(NO_BOTTOM_MATTER)**
specifies that no ending markup language source code be added to the output file.

**See**
For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 688.

**(NO_TOP_MATTER)**
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

**See**
For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 689.

**(TITLE='title-text')**
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

**title-text**
is the text in the metadata of a file that indicates the title.

**See**
For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 689.

**(URL= 'Uniform-Resource-Locator')**
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

**See**
For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 690.

**Restriction**
If you specify the FRAME= option, then you must also specify the CONTENTS= option, the PAGE= option, or both.

**Example**
“Example 2: Creating an XML File and a DTD” on page 520

**GFOOTNOTE | NOGFOOTNOTE**
controls the location where footnotes are printed in the graphics output.
GFOOTNOTE
writes footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

NOGFOOTNOTE
writes footnotes that are created by ODS, which appear outside the graph borders.

Default GFOOTNOTE

Restrictions
Footnotes that are displayed by a markup language statement support all SAS/GRAPH FOOTNOTE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH FOOTNOTE statement, see “FOOTNOTE Statement” in SAS/GRAPH: Reference.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

GPATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)
specifies the location for all graphics output that is generated while the destination is open. Use this option when you want to write graphics output files to a location different that specified by the PATH= option for markup files. If you specify an invalid filename, the ActiveX and Java devices send output to the default filename. Other devices create the file as a directory and write output to that directory using the default filename. For information about how ODS names catalog entries and external files, see SAS/GRAPH: Reference.

'aggregate-file-storage-location'
specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement You must enclose aggregate-file-storage-location in quotation marks.

fileref
is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction If you specify a fileref in the GPATH= option, then ODS does not use information from the GPATH= option when it constructs links.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

libref.catalog
specifies a SAS catalog to write to.

URL= 'Uniform-Resource-Locator' | NONE
specifies a URL for file-specification.

Uniform-Resource-Locator
is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.
Requirement  You must enclose Uniform-Resource-Locator in quotation marks.

NONE

specifies that no information from the GPATH= option appears in the links or references.

Tip  This option is useful for building output files that can be moved from one location to another. If the links from the contents and page files are constructed with a simple URL (one name), then they resolve, as long as the contents, page, and body files are all in the same location.

Default  If you omit the GPATH= option, then ODS stores graphics in the location that is specified by the PATH= option. If you do not specify the PATH= option, then ODS stores the graphics in the current directory. For more information, see the PATH= option.

GTITLE | NOGTITLE

controls the location where titles are printed in the graphics output.

GTITLE  
writes the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.

NOGTITLE  
writes the title that is created by ODS, which appears outside of the graph borders.

Default  GTITLE

Restrictions  Titles that are displayed by any markup language statement support most SAS/GRAPH TITLE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH TITLE statement, see TITLE statement.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

HEADTEXT= 'markup-document-head'

specifies markup tags to place between the <HEAD> and </HEAD> tags in all of the output files.

markup-document-head  
specifies the markup tags to place between the <HEAD> and </HEAD> tags.

Restriction  HEADTEXT= cannot exceed 256 characters.

Requirement  You must enclose markup-document-head in quotation marks.

Tips  ODS cannot parse the markup that you supply. It should be well-formed markup that is correct in the context of the <HEAD> and </HEAD> tags.

Use the HEADTEXT= option to define programs (such as JavaScript) that you can use later in the file.
**ID= identifier**

enables you to run multiple instances of the same destination at the same time. Each instance can have different options.

**identifier**

specifies another instance of the destination that is already open. **identifier** is numeric or a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numeric characters.

**Restriction**

If **identifier** is numeric, it must be a positive integer.

**Requirement**

You must specify the ID= option immediately after the destination name.

**Tip**

You can omit the ID= option and instead use a name or a number to identify the instance.

**Example**

“Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594

**METATEXT= 'metatext-for-document-head'**

specifies HTML code to use as the <META> tag between the <HEAD> and </HEAD> tags of all of the HTML output files.

'**metatext-for-document-head**'

specifies the HTML code that provides the browser with information about the document that it is loading. For example, this attribute could specify the content type and the character set to use.

**Requirement**

You must enclose **metatext-for-document-head** in quotation marks.

**Default**

If you do not specify **METATEXT=**, then ODS writes a simple <META> tag, which includes the content-type of the document and the character set to use, to all the HTML files that it creates.

**Restriction**

**METATEXT=** cannot exceed 256 characters.

**Tip**

ODS cannot parse the HTML code that you supply. It should be well-formed HTML code that is correct in the context of the <HEAD> tags. If you are using **METATEXT=** as it is intended, then your META tag should look like this:

```html
<META your-metatext-is-here>
```

**NEWFILE= starting-point**

creates a new body file at the specified **starting-point**.

**starting-point**

is the location in the output where you want to create a new body file.

ODS automatically names new files by incrementing the name of the body file. In the following example, ODS names the first body file **REPORT.XML**.

Additional body files are named **REPORT1.XML**, **REPORT2.XML**, and so on.

**Example:**

**BODY= 'REPORT.XML'**

**starting-point** is one of the following:
BYGROUP
starts a new file for the results of each BY group.

NONE
writes all output to the body file that is currently open.

OUTPUT
starts a new body file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

Alias TABLE

PAGE
starts a new body file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.

PROC
starts a new body file each time you start a new procedure.

Default NONE

Restriction The NEWFILE= option cannot be used in conjunction with the BODY=fileref option.

Tips If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first body file MAY5.XML. Additional body files are named MAY6.XML, MAY7.XML, and so on.

Example:
BODY= 'MAY5.XML'

OPTIONS ( DOC= | <suboption(s)> )
specifies tagset-specific suboptions and a named value.

(DOC= 'HELP' | 'QUICK' | 'SETTINGS' | 'CHANGELOG')
provides information about the specified tagset.

HELP
provides generic help and information with a quick reference.

QUICK
describes the options available for this tagset.

SETTINGS
provides the current option settings.

CHANGELOG
lists a history of changes made to the tagset. This suboption is supported only on the RTF tagset.

Requirement All values must be enclosed in quotation marks.

suboption(s)
specifies one or more suboptions that are valid for the specified tagset. Suboptions have the following format:

keyword='value'
Specify one of the following options when opening an ODS tagset statement, or at any time after the destination has been opened, to get information about suboptions for the tagset.

- `options(doc='help');`
- `options(doc='quick');`
- `options(doc='settings');`

Requirement: `suboption(s)` must be enclosed in parentheses.

Example: “Example: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information” on page 802

**PACKAGE <package-name>**

specifies that the output from the destination be added to a package.

- `package-name` specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

See: “ODS PACKAGE Statement” on page 554

Example: “Example 1: Creating an ODS Package” on page 558

**PAGE= 'file-specification' <(suboption(s))>**

opens a markup family destination and specifies the file that contains a description of each page of the body file, and contains links to the body file. ODS produces a new page of output whenever a procedure requests a new page. These files remain open until you do one of the following:

- close the destination with either an ODS `markup-family-destination CLOSE` statement or ODS `_ALL_ CLOSE` statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

- `file-specification` specifies the file, fileref, or SAS catalog to write to.

  - `file-specification` is one of the following:

    - `external-file`
      is the name of an external output file.

      Requirement: You must enclose `external-file` in quotation marks.

    - `fileref`
      is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

      See: For information about the FILENAME statement, see “FILENAME Statement” in *SAS Statements: Reference*.

    - `entry.markup`
      specifies an entry in a SAS catalog to write to.

      Interaction: If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.
suboption(s)
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

**(DYNAMIC)**
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 688.

**(NO_BOTTOM_MATTER)**
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 688.

**(NO_TOP_MATTER)**
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 689.

**(TITLE='title-text')**
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

*title-text*
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 689.

**(URL= 'Uniform-Resource-Locator')**
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 690.

Interaction The SAS system option PAGESIZE= has no effect on pages in HTML output except when you are creating batch output. For information about the PAGESIZE= option, see “PAGESIZE= System Option” in SAS System Options: Reference.

**PARAMETERS= (parameter-pair-1 ... parameter-pair-n)**
writes the specified parameters between the tags that generate dynamic graphics output.

*parameter-pair*
specifies the name and value of each parameter. *parameter-pair* has the following form:

'parameter-name'= 'parameter-value'
**parameter-name**

is the name of the parameter.

**parameter-value**

is the value of the parameter.

**Requirement**

You must enclose **parameter-name** and **parameter-value** in quotation marks.

**Interaction**

Use `PARAMETERS=` in conjunction with SAS/GRAPH procedures and the `DEVICE=JAVA`, `JAVAก็ตาม`, or `ACTIVEX` options in the `GOPTIONS` statement.

**See**

SAS/GRAPH: Reference for valid parameters for the Graph Applet, Map Applet, Contour Applet, and the MetaView Applet.

**PATH** = `'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)`

specifies the location of an aggregate storage location or a SAS catalog for all markup files. If the GPATH= option is not specified, all graphics output files are written to the "aggregate-file-storage-specification" or libref.

`'aggregate-file-storage-location'`

specifies an aggregate storage location such as directory, folder, or partitioned data set.

**Requirement**

You must enclose **aggregate-file-storage-location** in quotation marks.

**fileref**

is a file reference that has been assigned to an aggregate storage location. Use the `FILENAME` statement to assign a fileref.

**Interaction**

If you use a fileref in the `PATH=` option, then ODS does not use information from `PATH=` when it constructs links.

**See**

For information about the `FILENAME` statement, see “FILENAME Statement” in SAS Statements: Reference.

**libref.catalog**

specifies a SAS catalog to write to.

**See**

For information about the `LIBNAME` statement, see “LIBNAME Statement” in SAS Statements: Reference.

**URL** = `'Uniform-Resource-Locator' | NONE`

specifies a URL for the `file-specification`.

**Uniform-Resource-Locator**

is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

**NONE**

specifies that no information from the `PATH=` option appears in the links or references.

**Tip**

This option is useful for building output files that can be moved from one location to another. The links from the contents and page files must be
constructed with a single-name URL, and the contents, page, and body files must be in the same location.

Interaction If you use the BODY= or FILE= external file option in conjunction with the PATH= option, the external file specification should not include path information.

RECORD_SEPARATOR= ’alternative-separator’ | NONE
specifies an alternative character or string that separates lines in the output files.

Different operating environments use different separator characters. If you do not specify a record separator, then the files are formatted for the environment where you run the SAS program. However, if you are generating files for viewing in a different operating environment that uses a different separator character, then you can specify a record separator that is appropriate for the target environment.

alternative-separator
represents one or more characters in hexadecimal or ASCII format. For example, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= ’0D0A’x

Operating Environment Information
In a mainframe environment, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= ’0D25’x

Requirement You must enclose alternative-separator in quotation marks.

NONE
produces the markup language that is appropriate for the environment where you run the SAS job.

Windows Specifics
In a mainframe environment, by default, ODS produces a binary file that contains embedded record separator characters. This binary file is not restricted by the line-length restrictions on ASCII files. However, if you view the binary files in a text editor, then the lines run together. If you want to format the files so that you can read them with a text editor, then use RECORD_SEPARATOR= NONE. In this case, ODS writes one line of markup language at a time to the file. When you use a value of NONE, the logical record length of the file that you are writing to must be at least as long as the longest line that ODS produces. If the logical record length of the file is not long enough, then the markup language might wrap to another line at an inappropriate place.

Aliases RECSEP=

RS=

STYLE= style-template
specifies the style template to use in writing the output files.

style-template
describes how to display the presentation aspects (color, font face, font size, and so on) of your SAS output. A style template determines the overall appearance of the documents that use it. Each style template consists of style elements.
The STYLE= option is not valid when you are creating XML output.

If you are using SAS Studio, you do not need to specify the STYLE= option. You can go to Preferences ⇒ Results and change the style from the drop-down list for your selected destination.

For a complete discussion of style templates, see “TEMPLATE Procedure: Creating a Style Template” in SAS Output Delivery System: Procedures Guide.

If you do not specify a style template, then ODS uses the file that is specified in the SAS registry subkey ODS $DESTINATIONS$ MARKUP. By default, this value specifies Default.

If you specify the STYLE= option in an ODS HTML4 statement and subsequently need PROC PRINT output to use new style templates in another ODS HTML4 statement, close the first statement before specifying the second statement.

`STYLESEET= 'file-specification' <(suboption(s))>`

opens a markup family destination and places the style information for markup output into an external file, or reads style sheet information from an existing file. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

`file-specification`

specifies the file, fileref, or SAS catalog to write to.

`file-specification` is one of the following:

- `external-file`
  is the name of an external output file.

  **Requirement**
  You must enclose `external-file` in quotation marks.

- `fileref`
  is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

  **See**
  For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

- `entry.markup`
  specifies an entry in a SAS catalog to write to.

  **Interaction**
  If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

`suboption(s)`

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:
(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 688.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 688.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 689.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

$title-text$
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 689.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 690.

Note By default, if you do not specifically send the information to a separate file, then the style sheet information is included in the specified HTML file.

Example “Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

TEXT=text-string
inserts text into your document by triggering the paragraph event and specifying a text string to be assigned to the VALUE event variable.

Default By default the TEXT= option is used in a paragraph event.

Tip You can specify a text-string for a specific event by using the TEXT= option with the EVENT= option by using the following syntax:

EVENT=event-name (TEXT=text-string)
See For information about events and event variables, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in SAS Output Delivery System: Procedures Guide.

Example “Example: Conditionally Excluding Output Objects and Sending Them to Different Output Destinations” on page 325

TRANTAB= ’translation-table’
specifies the translation table to use when transcoding a file for output.


Suboptions

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file. This option sets the value of the CONTENTTYPE= style attribute. For more information, see CONTENTTYPE= on page 1015 in PROC TEMPLATE.

Default If you do not specify DYNAMIC, then ODS sets the value of HTMLCONTENTTYPE= for writing to a file.

Restriction If you specify the DYNAMIC suboption with one of the following options in the ODS HTML statement, then you must specify it for all of these options in that statement.
- BODY=
- CONTENTS=
- PAGE=
- FRAME=
- STYLESHEET=
- TAGSET=

Requirements You must enclose DYNAMIC in parentheses.

You must specify DYNAMIC next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

Alias NOBOT

Requirements You must enclose NO_BOTTOM_MATTER in parentheses.

You must specify NO_BOTTOM_MATTER next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.
Interactions

The NO_BOTTOM_MATTER suboption, in conjunction with the NO_TOP_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

Tip

If you want to leave a body file in a state that you can append to with ODS, then use NO_BOTTOM_MATTER with the file-specification BODY= option in any markup language statement.

See

The NO_TOP_MATTER suboption

(NO_TOP_MATTER)

specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

Alias

NOTOP

Requirements

You must enclose NO_TOP_MATTER in parentheses.

You must specify NO_TOP_MATTER next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

Interactions

The NO_TOP_MATTER suboption, in conjunction with the NO_BOTTOM_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

See

The NO_BOTTOM_MATTER suboption and the ANCHOR= option

(TITLE="title-text")

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

\textit{title-text}

is the text in the metadata of a file that indicates the title.

Requirements

You must enclose TITLE= in parentheses.

You must enclose \textit{title-text} in quotation marks.

Tip

If you are creating a web page that uses frames, then it is the TITLE= specification for the frame file that appears in the browser window title bar.
Example

“Example 3: Creating Multiple Markup Output” on page 522

(URL= 'Uniform-Resource-Locator')

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

Requirements

You must enclose URL= 'Uniform-Resource-Locator' in parentheses.

You must enclose Uniform-Resource-Locator in quotation marks.

You must specify URL= 'Uniform-Resource-Locator' next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

Tips

This option is useful for building HTML files that can be moved from one location to another. The links from the contents and page files must be constructed with a single name URL, and the contents, page, and body files must all be in the same location.

You never need to specify this suboption with the FRAME= option because ODS files do not reference the frame file.

Example

“Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

Details

The ODS PHTML statement is part of the ODS markup family of statements. ODS statements in the markup family produce output that is formatted using one of many different markup languages, such as HTML (Hypertext Markup Language) and XML (Extensible Markup Language). You can specify a markup language that SAS supplies, or create one of your own and store it as a user-defined markup language.

**ODS PRINTER Statement**

Opens, manages, or closes the PRINTER destination, which produces printable output.

**Valid in:** Anywhere

**Category:** ODS: Third-Party Formatted

**Default:** The default style for PRINTER destinations is Pearl.

**Interaction:** By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC PLOT or PROC CHART), ODS formats the output in SAS Monospace font. If you create output that is not displayed correctly. This is because without SAS, the SAS Monospace font is not recognized. To make your document display correctly, include the following statement before your SAS program:

```
OPTIONS FORMCHAR="|-----|+|---+=-|/\<>*;"
```

**CAUTION:** For PostScript output, verify that your online viewer or printer is set to use the same paper size as the value that is specified by the OPTIONS PAPERSIZE= statement. Otherwise, some parts of your output might appear to be missing.
Syntax

ODS PRINTER <(<ID=> identifier)> < action> ;
ODS PRINTER <(<ID=> identifier)> <option(s)> ;

Summary of Optional Arguments

(<ID=> identifier)
Open multiple instances of the same destination at the same time

ANCHOR='anchor-name'
Specify the root name for the anchor tag that identifies each output object in the current file

AUTHOR= 'author-text'
Insert the text string that you specify as the author into the metadata of a file

BASE='base-text'
Specify a string to use as the first part of all references that ODS creates in the file

BOOKMARKGEN | NOBOOKMARKGEN | BOOKMARKGEN=
Control the generation of bookmarks in PDF and PS files

BOOKMARKLIST= HIDE | NONE | SHOW
Specify whether to generate and display the list of bookmarks for PDF and PS files

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
Specify how to measure the width of cells. Use to override the default value of BOX_SIZING for a destination

CLOSE
Close the destination and the file that is associated with it

COLOR=FULL | GRAY | MONO | NO | YES
Apply a specified color scheme to your output

COLUMNS=n
Specify the number of columns to create on each page of output

COMPRESS=n
Specify the compression of a PDF file. Compression reduces the size of the file

CONTENTS= NO | YES
Control the generation of a printable table of contents

CSSSTYLE= 'file-specification'<(<media-type-1<…media-type-10>)}
Specify a cascading style sheet to apply to your output

DOM="external-file">
Specify that the ODS document object model is written to the SAS log or to an external file.

DPI=
Specify the image resolution in dots per inch for output images

EXCLUDE exclusion(s) | ALL | NONE
Exclude output objects from the destination

FILE='external-file' | fileref
Specify the output file.

GFOOTNOTE | NOGFOOTNOTE
Specify the location where footnotes are printed in the graphics output

GTITLE | NOGTITLE
Control the location where titles are printed in the graphics output
HOST
   Use the printer drivers that the host system provides

KEYWORDS='keywords-text'
   Insert a string of keywords into the output file's metadata

NEWFILE=starting-point
   Create a new file at the specified starting-point

NOTOC
   Omit the table of contents (Bookmark list) that is produced by default when producing PDF or PDFMARK output

PACKAGE <package-name>
   Specify that the output from the destination be added to an ODS package

PCL
   Create PCL output

PDF
   Create PDF output

PDFMARK
   Insert special markup that is used when converting a PostScript file to a PDF file

PDFNOTE | NOPDFNOTE
   Control whether notes are added to a PDF file for items that are associated with the FLYOVER= style attribute

PDFTOC=n
   Control the level of the expansion of the table of contents in PDF documents

PRINTER=printer-name
   Create output that is formatted for the specified printer

PS
   Create PostScript output

SELECT selection(s) | ALL | NONE
   Select output objects for the destination

SHOW
   Write to the SAS log the current selection or exclusion list for the destination

STARTPAGE=NEVER | NO | NOW | YES | BYGROUP
   Control page breaks

STYLE=style-template
   Specify the style template to use in writing the PDF output

SUBJECT='subject-text'
   Insert the text string that you specify as the subject in the metadata of a file

TEXT='text-string'
   Insert text into your output

TITLE='title-text'
   Insert the text string that you specify as the title in the metadata of a file

UNIFORM
   For multi-page tables, provide uniformity from page to page within a single table

Without Arguments
When you use the ODS PRINTER statement in UNIX or z/OS operating environments without an action or options, the PRINTER destination is opened and PostScript output is created unless otherwise configured by your system administrator.
If you use the ODS PRINTER statement in the Windows operating environment without an action or options, then the output is printed to the default Windows printer.

**Actions**
The following actions are available for the ODS PRINTER statement:

**CLOSE**
closes the destination and any files that are associated with it.

*Tip* When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

**EXCLUDE exclusion(s) | ALL | NONE**
excludes one or more output objects from the destination.

*Default* NONE

*Restriction* A destination must be open for this action to take effect.

*See* “ODS EXCLUDE Statement” on page 321

**SELECT selection(s) | ALL | NONE**
selects output objects for the specified destination.

*Default* ALL

*Restriction* A destination must be open for this action to take effect.

*See* “ODS SELECT Statement” on page 758

**SHOW**
writes the current selection list or exclusion list for the destination to the SAS log.

*Restriction* The destination must be open for this action to take effect.

*Tip* If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

*See* “ODS SHOW Statement” on page 771

**Optional Arguments**

**ANCHOR='anchor-name'**
specifies the root name for the anchor tag that identifies each output object in the current file.

Each output object must have an anchor tag for the bookmarks to reference. The references are automatically created by ODS. These references, point to the name of an anchor. Therefore, each anchor name in a file must be unique.

*anchor-name* is the root name for the anchor tag that identifies each output object in the current file.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR='TABULATE', then ODS names the first
anchor \textit{tabulate}. The second anchor is named \textit{tabulate1}; the third is named \textit{tabulate2}, and so on.

**Requirement**  
You must enclose \textit{anchor-name} in quotation marks.

**Alias**  
NAMED_DEST= | BOOKMARK=

**Restriction**  
Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

**Tips**  
You can change anchor names as often as you want by submitting the ANCHOR= option in a valid statement anywhere in your program. After you have specified an anchor name, it remains in effect until you specify a new one.

Specifying new anchor names at various points in your program is useful when you want to link to specific parts of your PRINTER output. Because you can control where the anchor name changes, you know in advance what the anchor name is at those points.

\[ \text{AUTHOR= 'author-text'} \]

inserts the text string that you specify as the author into the metadata of a file.

\textit{author-text}

is the text in the metadata of an open file that indicates the author.

**Restrictions**  
Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

The AUTHOR= option takes effect only if specified at the opening of a file.

**Requirement**  
You must enclose \textit{author-text} in quotation marks.

\[ \text{BASE='base-text'} \]

specifies the text to use as the first part of all references that ODS creates in the output file.

\textit{base-text}

is the text that ODS uses as the first part of all references that ODS creates in the file.

Consider this specification:

\[ \text{BASE='http://www.your-company.com/local-url/'} \]

In this case, ODS creates references that begin with the string \texttt{http://www.your-company.com/local-url/}. The appropriate \textit{anchor-name} completes the link.

**Restriction**  
Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

**Requirement**  
You must enclose \textit{base-text} in quotation marks.
BOOKMARKLIST= HIDE | NONE | SHOW
specifies whether to generate and display the list of bookmarks for PDF and PS files.

HIDE
generates a list of bookmarks for your PDF and PS files. The bookmarks are not automatically displayed when you open the PDF and PS files.

NONE
specifies not to generate a list of bookmarks for your PDF and PS files.

Aliases | NO | OFF
NOBOOKMARKLIST is an alias for BOOKMARKLIST=NONE | NO | OFF.

SHOW
generates a list of bookmarks for your PDF and PS files. The bookmarks are automatically displayed when you open the PDF and PS files.

Aliases | YES | ON
BOOKMARKLIST is an alias for BOOKMARKLIST=SHOW | YES | ON.

Example | “Example 2: Creating a Printable Table of Contents” on page 598

Default | SHOW

Restrictions | This option can be set only when you first open the destination.
This option has an effect only when creating PDF, PDFMARK, PS output.

Interaction | The NOTOC option specifies BOOKMARKLIST= OFF and CONTENTS= OFF.

Note | The generation of the bookmarks is not affected by the setting of this option. Bookmarks are generated by the BOOKMARKGEN= option.

Example | “Example 2: Creating a Printable Table of Contents” on page 598

BOOKMARKGEN | NOBOOKMARKGEN | BOOKMARKGEN=
controls the generation of bookmarks in PDF and PS files.

BOOKMARKGEN
specifies to generate bookmarks in PDF and PS files.

BOOKMARKGEN=
controls the generation of bookmarks in PDF and PS files.

NO
specifies not to generate bookmarks in PDF and PS files.

Alias | OFF

YES
specifies to generate bookmarks in PDF and PS files.

Alias | ON
NOBOOKMARKGEN
 specifies not to generate bookmarks in the PDF and PS files.

Default   YES or BOOKMARKGEN

Interaction    If you set BOOKMARKGEN=NO, then the BOOKMARKLIST option is set to NO also.

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)
 specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing.

COLOR=FULL | GRAY | MONO | NO | YES
 applies the specified color scheme to your output.

FULL
 creates full color output for both text and graphics.

GRAY
 creates gray scale output for both text and graphics.

Alias   GREY

MONO
 creates monochromatic output for both text and graphics.

Alias   BW

NO
 does not use all the color information that the style template provides. If you specify COLOR=NO, then the destination does this:

• generates black and white output
• creates all text and rules in black
• sets the SAS/GRAPH device to produce SAS/GRAPH output in gray scale
• ignores specifications for a background color from the style template except for the purposes of determining whether to print rules for the table

YES
 uses all the color information that a style template provides, including background color. To print in color, you must also do the following:

• use a printer that is capable of printing in color.
• use the COLORPRINTING SAS system option. For information about the COLORPRINTING system option, see SAS System Options: Reference.

Default    YES

Tip   If you choose color output for a printer that does not support color, then your output might be difficult to read.

COLUMNS=n
 specifies the number of columns to create on each page of output.
\[ n \]
is the number columns per page.

Default 1

**COMPRESS=n**
controls the compression of a PDF file. Compression reduces the size of the file.

\[ n \]
specifies the level of compression. The larger the number, the greater the compression. For example, \( n=0 \) is completely uncompressed, and \( n=9 \) is the maximum compression level.

Default 6

Range 0–9

Restrictions Use this option only with the ODS PDF statement and the ODS PRINTER statement with the PDF option specified. PostScript output cannot be compressed.

The COMPRESS= option takes effect only if specified at the opening of a file.

Interactions The COMPRESS= option overrides the DEFLATION system option. First, the DEFLATION system option checked. Next, the ODS PDF statement COMPRESS= option is checked. If the COMPRESS= option is specified, that value is used regardless of the value specified for the DEFLATION system option. For more information, see the DEFLATION option.

The COMPRESS= option overrides the UPRINTCOMPRESSION option. If COMPRESS= is specified, the UPRINTCOMPRESSION system option is then queried. If the system option is off, it is turned on for this one PDF statement and the PDF file is compressed. When compression is complete, the UPRINTCOMPRESSION system option is again enabled for all other files to use. For more information, see the UPRINTCOMPRESSION system option.

**CONTENTS= ** NO | YES
controls the generation of a printable table of contents.

NO does not generate a printable table of contents.

Alias NOCONTENTS is an alias for CONTENTS=NO.

YES generates a printable table of contents.

Alias CONTENTS is an alias for CONTENTS=YES.

Default NO

Example “Example 2: Creating a Printable Table of Contents” on page 598

**CSSSTYLE=’file-specification’<(media-type-1…media-type-10)>**
specifies a cascading style sheet to apply to your output.
file-specification
specifies a file, fileref, or URL that contains CSS code.

file-specification is one of the following:

"external-file"
is the name of the external file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

"URL"
is a URL to an external file.

Requirement You must enclose URL in quotation marks.

(media-type-1<.. media-type-10>)specifies one or more media blocks that correspond to the type of media that your output is rendered on. CSS uses media type blocks to specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not contained in any media blocks. By using the media-type suboption, in addition to the general CSS code, you can import the section of a CSS file intended only for a specific media type.

Default If no media-type is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

Range You can specify up to ten different media types.

Requirements You must enclose media-type in parentheses.

You must specify media-type next to the file-specification specified by the CSSSTYLE= option.

Tip If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

Restriction The CSSSTYLE= option does not affect SAS/GRAPH output.

Requirement CSS files must be written in the same type of CSS produced by the ODS HTML statement. To view the CSS code that ODS creates, you can do one of the following:
• specify the ODS TRACE DOM statement
• specify the DOM option
Interaction

If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

See

For complete documentation about ODS and Cascading Style Sheets, see SAS Output Delivery System: Advanced Topics

Example

“Example 6: Applying a CSS File to ODS Output” on page 527

DOM="external-file">
specifies that the ODS document object model is written to the SAS log or an external file.

external-file

is the name of an external output file.

Requirement

You must enclose external-file in quotation marks.

See

For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in SAS Output Delivery System: Advanced Topics.

DPI=
specifies the image resolution for output files.

Default

The default is determined by the default DPI of the specified PRINTER= value or the PRINTERPATH option.

Restriction

The DPI= option takes effect only if specified at the opening of a file.

CAUTION

When using high DPI= or DPI_IMAGE= values (values over 600), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500 or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.

FILE='external-file' | fileref

specifies the output file.

external-file

is the name of an external file.

Requirement

You must enclose external-file in quotation marks.

fileref

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

Restriction

The FILE=fileref option cannot be used in conjunction with the NEWFILE= option.

See

For information about the FILENAME statement, see SAS Statements: Reference.

Default

If you do not specify an output file, then ODS writes to the file that is specified by two SAS system options. Use the SYSPRINT= system option if you are using the Windows operating environment and do not
specify any of the following options: PCL, PDFMARK, POSTSCRIPT, PS, or SAS. Use the PRINTERPATH= system option in all other cases. If the system option does not specify a file, then ODS writes to the default printer. For more information, see the PRINTER= option.

Interaction

In an ODS printer family statement that refers to an open ODS PRINTER destination, the FILE= option forces ODS to close the destination and all files that are associated with it. ODS then opens a new instance of the destination. For more information, see “Opening and Closing the PRINTER Destination” on page 708.

See

For information about the FILENAME statement, see SAS Statements: Reference.

GFOOTNOTE | NOGFOOTNOTE
controls the location of the footnotes that are defined by the graphics program that generates the Printer output.

GFOOTNOTE
includes all of the currently defined footnotes within the graphics output.

NOGFOOTNOTE
prevents all of the currently defined footnotes from appearing in the graphics file. Instead, they become part of the Printer file.

Default GFOOTNOTE

Restriction This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

See For more information, see “Customizing Titles and Footnotes” on page 40.

GTITLE | NOGTITLE
controls the location of the titles that are defined by the graphics program that generates the Printer output.

GTITLE
includes all of the currently defined titles within the graphics output.

NOGTITLE
prevents all of the currently defined titles from appearing in the graphics output. Instead, the titles become part of the Printer file.

Default GTITLE

Restriction This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

See For more information, see “Customizing Titles and Footnotes” on page 40.

HOST
specifies that ODS use the printer drivers that the host system provides.

Interaction In an ODS printer family statement that refers to an open ODS PRINTER destination, the HOST option forces ODS to close the
destination and all files that are associated with it. ODS then opens a new instance of the destination. For more information, see “Opening and Closing the PRINTER Destination” on page 708.

\(<\text{ID=>} \text{identifier}\) enables you to open multiple instances of the same destination at the same time. Each instance can have different options.

\text{identifier}

can be numeric or can be a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numerals.

\textbf{Restriction} If \text{identifier} is numeric, it must be a positive integer.

\textbf{Requirement} The ID= option must be specified immediately after the destination name.

\textbf{KEYWORDS='keywords-text'}

inserts a string of keywords into the output file's metadata. The keywords enable a document management system to do topic-based searches.

\text{keywords-text}

is the string of keywords.

\textbf{Restrictions} Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

The KEYWORDS= option takes effect only if specified at the opening of a file.

\textbf{Requirement} You must enclose keywords-text in quotation marks.

\textbf{NEWFILE= starting-point}

creates a new file at the specified starting-point.

\text{starting-point}

is the location in the output where you want to create a new file.

ODS automatically names new files by incrementing the name of the file. In the following example, ODS names the first file \texttt{REPORT.PS}. Additional body files are named \texttt{REPORT1.PS, REPORT2.PS}, and so on.

Example:

\texttt{FILE= 'REPORT.PS'}

\text{starting-point} can be one of the following:

\textbf{BYGROUP}

starts a new file for the results of each BY group.

\textbf{NONE}

writes all output to the file that is currently open.

\textbf{OUTPUT}

starts a new file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.
PAGE
starts a new file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.

PROC
starts a body file each time you start a new procedure.

Default
NONE

Restrictions
The NEWFILE= option cannot be used in conjunction with the FILE=fileref option.

The NEWFILE= option cannot be used if you are sending output to a physical printer.

Tips
If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first file MAY5.PS. Additional body files are named MAY6.PS, MAY7.PS, and so on.

Example:
FILE= 'MAY5.PS'

NOTOC
specifies that ODS omit the table of contents (Bookmark list) that is produced by default when producing PDF or PDFMARK output.

Interaction
The NOTOC option specifies BOOKMARKLIST=OFF and CONTENTS= OFF.

Examples
“Example 5: Combining a Table and Image on the Same Page” on page 608
“Example 6: Adding Text That Imitates a System Title” on page 611
“Example 7: Toggling Page Breaks” on page 615
“Example 8: Suppressing a Page Break” on page 619

PACKAGE <package-name>
specifies that the output from the destination be added to a package.

package-name
specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

See “ODS PACKAGE Statement” on page 554

PCL
creates PCL output.

Restriction
Do not use this option in conjunction with the PDF or PS option.
Interaction If you use the PCL option in an ODS PRINTER statement that refers to an open ODS PRINTER destination, the option forces ODS to close the destination and all files that are associated with it. ODS then opens a new instance of the destination. For more information, see “Opening and Closing the PRINTER Destination” on page 708.

PDF creates PDF output.

Restrictions Do not use this option in conjunction with the PCL or PS options.

- PDF does not support double-byte Type1 fonts.

Interaction If you use the PDF option in an ODS PRINTER statement that refers to an open ODS PRINTER destination, the option forces ODS to close the destination and all files that are associated with it. ODS then opens a new instance of the destination. For more information, see “Opening and Closing the PRINTER Destination” on page 708.

PDFMARK enables ODS to insert special tags into a PostScript file. When you use software such as Adobe Acrobat (not Adobe Viewer), Acrobat Distiller interprets the tags to create a PDF file that contains the following items:

- bookmarks for each section of the output and for each table.
- references for items that are associated with the URL= style attribute.
- notes for items that are associated with the FLYOVER= style attribute. Notes are optional, and are based on the PDFNOTE option.
- author, keywords, subject, and title in the metadata of a file.

Default Because using PDFMARK implies PostScript output, SAS automatically uses the PostScript driver that SAS supplies with this option.

Restriction You cannot use the PRINTER= option with the PDFMARK option.

Requirement To create a PDF file, you must use specialized software, such as Adobe Acrobat Distiller to convert the marked-up PostScript file into a PDF formatted file.

Interaction In an ODS printer family statement that refers to an open ODS PRINTER destination, the PDFMARK option forces ODS to close the destination and all files that are associated with it. ODS then opens new instance of the destination. For more information, see “Opening and Closing the PRINTER Destination” on page 708.

Tip Use this option only if you plan to distill the output. Otherwise, it uses excess resources and does not enhance the results.

PDFNOTE | NOPDFNOTE controls whether notes are added to a PDF file for items that are associated with the FLYOVER= style attribute.

PDFNOTE adds notes to a PDF file for items that are associated with the FLYOVER= style attribute.
NOPDFNOTE modifies the behavior of PDFMARK so that notes are not added to the file for items that are associated with the FLYOVER= style attribute.

**Default** PDFNOTE

**Restriction** Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and ODS PRINTER statement with the PDFMARK option specified.

**PDFTOC=n**
controls the level of the expansion of the table of contents in PDF documents.

*n* specifies the level of expansion. For example, PDFTOC=0 results in a fully expanded table of contents. PDFTOC=2 results in a table of contents that is expanded to two levels.

**Default** 0

**Tip** The PDFTOC= can be set after the file has been opened, but only the last specification for a given file is used.

**See** “Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594

**PRINTER= printer-name**
creates output that is formatted for the specified printer.

*printer-name* is the name of the printer for which you want output formatted.

**Alias** PRT

**Default** If you do not specify a printer, then ODS formats the printer output for the printer that is specified by one of two SAS system options. Use the SYSPRINT= system option if you are using the Windows operating environment and do not specify any of the following options: PCL, PDFMARK, POSTSCRIPT, PS, or SAS. Use the PRINTERPATH= system option in all other cases. If the system option does not specify a printer, then ODS writes to the default printer driver as specified in the SAS registry or the Windows registry. In the SAS registry, the default printer is specified in CORE ⇒ PRINTING ⇒ Default Printer. In SAS 9.3, the default printer value can be modified in the SAS Registry. For information on how to change the default printer, see “Changing SAS Registry Settings for ODS” on page 40.

**Restrictions** *printer-name* must match a subkey in either the SAS registry or the Windows printer registry.

You cannot use the PRINTER= option with the PCL, PDF, PDFMARK, or PS options.

**Interaction** In an ODS printer family statement that refers to an open ODS PRINTER destination, the PRINTER= option forces ODS to close the destination and all files that are associated with it. ODS then opens a new instance of the destination. For more information, see “Opening and Closing the PRINTER Destination” on page 708.
Note  
printer-name is not necessarily a physical printer. It is a description that tells SAS how to format the output, and where the output is located. For example, it could be a file on a disk.

Tips  
The description of the printer includes its destination and device type. If you are using the SAS printer drivers, then you can find a description of the printer in CORE ↪ PRINTING ↪ PRINTERS ↪ selected-printer ↪ PRINTER SETUP ↪ OUTPUT.

In the Windows operating environment, if you do not specify the SAS option in the ODS PRINTER statement, then a description of the printer is located in the Windows registry.

To see a list of available printers for SAS printing, use the REGEDIT command. The printers are listed in the Registry Editor window under CORE ↪ PRINTING ↪ PRINTERS.

PS  
creates PostScript output.

Alias  
POSTSCRIPT

Restrictions  
Do not use this option in conjunction with the PDF or PCL options.

PS does not support double-byte Type1 fonts.

Interaction  
If you use the PS option in an ODS PRINTER statement that refers to an open ODS PRINTER destination, the option forces ODS to close the destination and all files that are associated with it. ODS then opens a new instance of the destination. For more information, see “Opening and Closing the PRINTER Destination” on page 708.

Tip  
Specifying this option is equivalent to specifying both the SAS option and PRINTER= POSTSCRIPT.

STARTPAGE=NEVER | NO | NOW | YES | BYGROUP  
controls page breaks.

BYGROUP  
specifies to insert page breaks after each BY group.

NEVER  
specifies not to insert page breaks, even before graphics procedures.

CAUTION:  
Each graph normally requires an entire page. The default behavior forces a new page after a graphics procedure. STARTPAGE=NEVER turns off that behavior, so specifying STARTPAGE= NEVER might cause graphics to overprint.

NO  
specifies that no new pages be inserted at the beginning of each procedure, or within certain procedures, even if new pages are requested by the procedure code. A new page begins only when a page is filled or when you specify STARTPAGE=NOW.

CAUTION:  
Each graph normally requires an entire page. The default behavior forces a new page after a graphics procedure, even if you use
STARTPAGE=NO. STARTPAGE=NEVER turns off that behavior, so specifying STARTPAGE= NEVER might cause graphics to overprint.

**Alias**
OFF

**Tip**
When you specify STARTPAGE=NO, system titles and footnotes are still produced only at the top and bottom of each physical page, regardless of the setting of this option. Thus, some system titles and footnotes that you specify might not appear when this option is specified.

**Examples**
- “Example 5: Combining a Table and Image on the Same Page” on page 608
- “Example 6: Adding Text That Imitates a System Title” on page 611
- “Example 7: Toggling Page Breaks” on page 615
- “Example 8: Suppressing a Page Break” on page 619

**NOW**
forces the immediate insertion of a new page.

**Tip**
This option is useful primarily when the current value of the STARTPAGE= option is NO. Otherwise, each new procedure forces a new page automatically.

**Example**
“Example 7: Toggling Page Breaks” on page 615

**YES**
inserts a new page at the beginning of each procedure, and within certain procedures, as requested by the procedure code.

**Alias**
ON

**Default**
YES

**STYLE=style-template**
specifies the style template to use in writing the printer output.

**Default**
If you do not specify a style template, then ODS uses the style template that is specified in the SAS registry subkey: ODS ⇐ DESTINATIONS ⇐ PRINTER. By default, this value is Pearl for the PRINTER, PDF, PS and PCL destinations.

**Note**
If you are using SAS Studio, you do not need to specify the STYLE= option. You can go to Preferences ⇐ Results and change the style from the drop-down list for your selected destination.

**See**
For a complete discussion of style templates, see “Working with Styles” in SAS Output Delivery System: Procedures Guide.

For instructions on making your own user-defined style templates, see “TEMPLATE Procedure: Creating a Style Template” in SAS Output Delivery System: Procedures Guide.

**Examples**
“Example 3: Customizing the Table of Contents” on page 600
SUBJECT=’subject-text’
Inserts into the metadata of a file the text string that you specify as the subject.

*subject-text*

is the text in the metadata of a file that indicates the subject.

**Restrictions**
Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

The SUBJECT= option takes effect only if specified at the opening of a file.

**Requirement**
You must enclose *subject-text* in quotation marks.

TEXT=’text-string’
Inserts a text string into your output.

*text-string*

is the text that you want to insert into your output.

**Requirement**
You must enclose *text-string* in quotation marks.

**Tip**
If you are submitting more than one procedure step and you do not specify the STARTPAGE=NO option, each procedure forces a new page before the output. Therefore, any text that you specify with TEXT= is on the same page as the previous procedure.

**See**
“Example: Conditionally Excluding Output Objects and Sending Them to Different Output Destinations” on page 325

TITLE=’title-text’
Inserts into the metadata of a file the text string that you specify as the title.

*title-text*

is the text in the metadata of a file that indicates the title.

**Restrictions**
Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

The TITLE= option takes effect only if specified at the opening of a file.

**Requirement**
You must enclose *title-text* in quotation marks.

**UNIFORM**
For multiple page tables, ensures uniformity from page to page within a single table.

When the UNIFORM option is in effect, ODS reads the entire table first, so that it can determine the column widths that are necessary to accommodate all the data. These column widths are applied to all pages of a multiple page table.

**Default**
If you do not specify the UNIFORM option, then ODS prints a table one page at a time. This approach ensures that SAS does not run out of memory while processing very large tables. However, it can also mean that column widths vary from one page to the next.
Note  With BY-group processing, SAS writes the results of each BY group to a separate table, so the output might not be uniform across BY groups.

Tip  The UNIFORM option can cause SAS to run out of memory if you are printing a very large table. If this happens, then you can explicitly set the width of each of the columns in the table, and then print the table one page at a time. To do so, you must edit the table template that you use. For more information, see “What You Can Do with Table Templates” in SAS Output Delivery System: Procedures Guide.

Details

Opening and Closing the PRINTER Destination
You can modify an open PRINTER destination with many ODS PRINTER options. However, the FILE=, HOST, PCL, PDF, PDFMARK, PRINTER=, PS, or SAS options perform the following actions on an open PRINTER destination:

• close the open destination referred to in the ODS PRINTER statement
• close any files associated with the open PRINTER destination
• open a new instance of the PRINTER destination

If you use one of these options, it is best to explicitly close the destination yourself.

For example, in the following ODS program, the second ODS PRINTER statement closes the PRINTER destination that is opened by the first ODS PRINTER statement. Therefore, the file Blockprintstyle.ps does not contain output that is formatted with the Journal style. However, the second ODS PRINTER statement does not affect the PS destination that is opened by the ODS PS statement. The PS destination is still open and the file nostyle.ps could be modified.

The ODS PRINTER statement opens the PRINTER destination and creates PostScript output.

ods printer ps style=blockprint file='Blockprintstyle.ps';
proc print data=statepop;
run;

The ODS PS statement opens the PS destination and creates PostScript output.

ods ps file='nostyle.ps';
proc print data=statepop;
run;

The ODS PRINTER statement closes the open PRINTER destination and the files that are associated with it. It then opens a new instance of the PRINTER destination and creates PostScript output.

ods printer ps style=Journal file='Journalstyle.ps';
proc print data=statepop;
run;
ods printer ps close;
ods ps close;

Printing Output Directly to a Printer
Printing output directly to a printer using the ODS PRINTER statement depends on your host operating environment.
Note: To print directly to a printer in the z/OS and UNIX operating environments, you can use the FILENAME statement. Specific information about your operating environment is required when using the FILENAME statement. See the SAS documentation for your operating environment before using this statement. Commands are also available in some operating environments that associate a fileref with a file and that break that association.

Table 6.16 Methods for Sending SAS Output to a Printer

<table>
<thead>
<tr>
<th>Platform</th>
<th>Method for Sending SAS Output to a Printer</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS</td>
<td>Use the FILENAME statement with the SYSOUT= data set option specified. You can then print to the fileref. Syntax: filename your-fileref sysout=a dest=printer-name; ods printer file=your-fileref; Example: filename local sysout=a dest=chpljj21; ods printer file=local;</td>
</tr>
<tr>
<td>UNIX</td>
<td>Use the FILENAME statement with the PIPE command to associate a fileref with your lpr print command. Syntax: filename your-fileref pipe 'lpr -P printer-name'; ods printer file=your-fileref; Example: filename local pipe 'lpr -P chpljj21'; ods printer file=local;</td>
</tr>
<tr>
<td>Windows</td>
<td>If you want to print to your default printer, use this code. Syntax: ods printer; If you want to print to a printer that is not the default, then use the PRINTER= option to specify the printer name. Syntax: ods printer printer=printer-name; Example: ods printer printer=chpljj21; To find out what printers are available, select Start ⇒ Settings ⇒ Printers from the Taskbar. If a printer is listed there, then you can use it with the ODS PRINTER statement. If the printer name has spaces, then you must put the printer name in quotation marks.</td>
</tr>
</tbody>
</table>

Using ODS PRINTER with Windows

In the Windows operating environment, the ODS PRINTER statement produces output that is formatted for your default Windows printer unless you specify a different printer by using the PRINTER= option. For more information about the PRINTER= option,
see the PRINTER= option on page 704. You can also produce printable output files in PCL, PDF, or PostScript format by using the appropriate option.

Starting in SAS 9.3, you can change the default printer values in the SAS Registry. See “Changing SAS Registry Settings for ODS” on page 40 for more detailed information about this “Default Printer” attribute and how it can be modified.

**Using ODS PRINTER with All Other Hosts**
Universal Printing is the print mechanism used for all non-windowing hosts. Refer to “Universal Printing” in SAS Language Reference: Concepts for details.

The ODS PRINTER statement produces PostScript output files by default unless you change the default in the SAS Registry. You can also produce printable output files in PCL or PDF format by using the appropriate option or registry setting. See “PRINTER= printer-name” on page 704 for more detailed information about this “Default Printer” attribute and how it can be modified.

**PDF Security**
You can easily encrypt and password-protect your PDF output files by specifying the PDFSECURITY= system option. By specifying system option PDFSECURITY=HIGH, SAS encrypts PDF documents using a 128-bit encryption algorithm. When PDFSECURITY=HIGH is set, at least one password must be set using the PDFPASSWORD= system option. A password is required to open a PDF file that has been generated with ODS.

To enable encryption and password protection, specify the OPTIONS statement. The following code shows how to encrypt your PDF output file with a high level of encryption that is password protected:

```sas
options pdfsecurity=high pdfpw=(open=testpw);
```

The following code shows the PDF security option used with the PDF destination:

```sas
options pdfsecurity=high pdfpw=(open=testpw);
ods pdf file="secure.pdf";
proc contents data=sashelp.class;
run;
ods pdf close;
```

For detailed information about the PDF Security options, see “Securing ODS-Generated PDF Files” on page 590.

*Note:* Encryption requires Acrobat version 5.0 or later.

**PDF Views**
System options PDFPAGELAYOUT and PDFPAGEVIEW enable you to control how you view your PDF document. The PDFPAGELAYOUT system option controls the page layout. This setting is equivalent to selecting View ➤ Page Display in Adobe Acrobat Reader when a document is open. The PDFPAGEVIEW system option controls the page viewing mode. This setting is equivalent to selecting View ➤ Zoom in Adobe Acrobat Reader.

For detailed information about these system options, see SAS System Options: Reference.

**Example: Selecting Output for the HTML and PRINTER Destinations**

**Features:**
Example 1: Selecting Output for the HTML and PRINTER Destinations

ODS _ALL_ CLOSE

ODS HTML statement:
   BODY=

ODS PRINTER statement:
   FILE=
      PS

ODS LISTING statement:
   CLOSE

ODS SELECT statement:
   with label
   with name
   with path

Other features:
   PROC UNIVARIATE

Data set:
   StatePop

This example selects three output objects from a UNIVARIATE procedure step to send to both the HTML destination and to the PRINTER destination.

Program

   title;
   options nodate nonumber;
   ods html body='your_file.html';
   ods printer ps file='your_file.ps';
   ods select BasicMeasures
      'Tests For Location'
      Univariate.CityPop_90.ExtremeObs;
   proc univariate data=statepop mu0=3.5;
      var citypop_90 citypop_80;
   run;
   ods printer close;

Program Description

   title;

   Set the SAS system options. The OPTIONS statement controls several aspects of the PRINTER output. The NODATE system option specifies that SAS not print the date and the time. The NONUMBER system option specifies that SAS not print the page number on the first title line of each page of SAS output. These options do not affect the HTML output.

   options nodate nonumber;

   Create HTML output. The BODY= option sends all output objects to the external file that you specify. Some browsers require an extension of HTM or HTML on the filename.
Create PostScript output. The ODS PRINT statement opens the PRINTER destination and the PS option specifies PostScript output. FILE= sends all output objects to the external file that you specify.

ods printer ps file='your_file.ps';

Specify the output objects to send to the open destinations. The ODS SELECT statement specifies three output objects to send to all open destinations. The first output object is selected by its name, BasicMeasures. The second output object is selected by its label, Tests For Location. These two selection criteria select the output objects for the analysis of both variables. The third output object is selected by its full path Univariate.CityPop_90.ExtremeObs. This selection criterion selects the output object for only one variable, CityPop_90.

ods select BasicMeasures 'Tests For Location' Univariate.CityPop_90.ExtremeObs;

Compute descriptive statistics for two variables. PROC UNIVARIATE computes descriptive statistics for two variables, CityPop_80 and CityPop_90. ODS routes the selected output objects to the HTML and PRINTER destinations.

proc univariate data=statepop mu0=3.5;
  var citypop_90 citypop_80;
run;

Close the open destinations so that you can view or print the output. The ODS PRINT statement closes the printer destination and all of the files that are associated with them. You must close the printer destination before you can send the output to a physical printer.

ods printer close;

HTML Output

The HTML output includes three output objects for the variable CityPop_90, and two output objects for the variable CityPop_80.
Example 1: Selecting Output for the HTML and PRINTER Destinations

Output 6.88  HTML Output for the Variables CityPop_90 and CityPop_80
The UNIVARIATE Procedure
Variable: CityPop_80 (1980 metropolitan pop in millions)

<table>
<thead>
<tr>
<th>Basic Statistical Measures</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Variability</td>
</tr>
<tr>
<td>Mean 3.468471</td>
<td>Std Deviation 4.42799</td>
</tr>
<tr>
<td>Median 2.114000</td>
<td>Variance 19.60710</td>
</tr>
<tr>
<td>Mode .</td>
<td>Range 22.77400</td>
</tr>
<tr>
<td>Interquartile Range 3.21000</td>
<td></td>
</tr>
</tbody>
</table>

Tests for Location: Mu0=3.5

<table>
<thead>
<tr>
<th>Test</th>
<th>Statistic</th>
<th>p Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student's t</td>
<td>-0.05085</td>
<td>Pr &gt;</td>
</tr>
<tr>
<td>Sign</td>
<td>-10.5</td>
<td>Pr &gt;=</td>
</tr>
<tr>
<td>Signed Rank</td>
<td>-190</td>
<td>Pr &gt;=</td>
</tr>
</tbody>
</table>

Printer Output

The printer output includes three output objects for the variable CityPop_90, and two output objects for the variable CityPop_80.
Output 6.89  PostScript Output for the Variables CityPop_90 and CityPop_80
ODS PROCLABEL Statement

Enables you to change a procedure label.

Valid in: Anywhere
Category: ODS: Output Control
Interaction: This statement applies to all open destinations, except for the output destination, where a procedure label is not an option. However, this setting lasts for only one procedure step. You must issue an ODS PROCLABEL statement for each procedure step that you have.

Examples: “Example 6: Adding Text That Imitates a System Title” on page 611
“Example 3: Customizing the Table of Contents” on page 600

Syntax

ODS PROCLABEL 'string';
ODS PROCLABEL= 'string';

Required Argument

'string'

is the procedure label that you specify.

Interaction The NOLABEL system option overrides the ODS PROCLABEL statement. Therefore, to produce labels using the ODS PROCLABEL statement, you must specify the LABEL system option also.

Details

ODS PROCLABEL affects the item names in the outer list of the table of contents.

See Also

System Option

• LABEL System Option

ODS PROCTITLE Statement

Determines whether to write the title that identifies the procedure that produces the results in the output.

Valid in: Anywhere
Category: ODS: Output Control
Interaction: This statement applies to all open destinations, except for the output destination, where a procedure label is not an option. This setting persists until you issue an ODS NOPROCTITLE statement. You do not have to issue an ODS PROCTITLE statement for each procedure step.

Examples: “Example 5: Combining a Table and Image on the Same Page” on page 608
Example 6: Adding Text That Imitates a System Title

Syntax

**ODS PROCTITLE;**
**ODS NOPROCTITLE;**

*Required Arguments*

**ODS PROCTITLE**
writes, in the output, the name of the procedure that produces the results.

*Note:* Not all procedures use a procedure title.

*Default*  
ODS PROCTITLE is the default.

**ODS NOPROCTITLE**
suppresses the writing of the title of the procedure that produces the results.

*Details*

The following table lists the aliases for the ODS PROCTITLE statement:

**Table 6.17  Aliases for ODS PROCTITLE**

<table>
<thead>
<tr>
<th>Statement</th>
<th>Alias</th>
</tr>
</thead>
</table>
| ODS PROCTITLE   | ODS PROCTITLE=ON  
|                 | ODS PTITLE    |
|                 | ODS PTITLE=ON  |
|                 | ODS PTITLE=YES|

<table>
<thead>
<tr>
<th>ODS NOPROCTITLE</th>
<th>ODS PROCTITLE=OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ODS NOPTITLE</td>
</tr>
<tr>
<td></td>
<td>ODS PTITLE=OFF</td>
</tr>
<tr>
<td></td>
<td>ODS PTITLE=NO</td>
</tr>
</tbody>
</table>

**ODS PS Statement**

Opens, manages, or closes the PS destination, which produces PostScript (PS) output.

*Valid in:* Anywhere

*Category:* ODS: Third-Party Formatted

*Default:* The default style for PRINTER destinations is Pearl.

*Restriction:* PS does not support double-byte Type1 fonts.

*Interaction:* By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC PLOT or PROC CHART), ODS formats the output in SAS Monospace font. If you are creating output that will be viewed in an operating
environment where SAS software is not installed, this output will not be displayed correctly. This is because without SAS, the SAS Monospace font is not recognized. To make your document display correctly, include the following statement before your SAS program:

```
OPTIONS FORMCHAR="|----|+|---+=-|/>.\*";
```

**Note:** By default, the ODS PS statement creates PNG images.

**CAUTION:** For PostScript output, verify that your online viewer or printer is set to use the same paper size as the value that is specified by the OPTIONS PAPERSIZE= statement. Otherwise, some parts of your output might appear to be missing.

### Syntax

```
ODS PS (<ID>= identifier) <action> ;
ODS PS (<ID>= identifier) <option(s)> ;
```

### Summary of Optional Arguments

- **(<ID>= identifier)**
  - Open multiple instances of the same destination at the same time
- **ANCHOR='anchor-name'**
  - Specify the root name for the anchor tag that identifies each output object in the current file
- **AUTHOR= 'author-text'**
  - Insert the text string that you specify as the author into the metadata of a file
- **BASE='base-text'**
  - Specify a string to use as the first part of all references that ODS creates in the file
- **BOOKMARKGEN | NOBOOKMARKGEN | BOOKMARKGEN=**
  - Control the generation of bookmarks in PDF and PS files
- **BOOKMARKLIST= HIDE | NONE | SHOW**
  - Specify whether to generate and display the list of bookmarks for PDF and PS files
- **BOX_SIZING=(CONTENT_BOX | BORDER_BOX)**
  - Specify how to measure the width of cells. Use to override the default value of BOX_SIZING for a destination
- **CLOSE**
  - Close the destination and the file that is associated with it
- **COLOR=FULL | GRAY | MONO | NO | YES**
  - Apply a specified color scheme to your output
- **COLUMNS=n**
  - Specify the number of columns to create on each page of output
- **CSSSTYLE= 'file-specification'<(media-type-1<…media-type-10>)>**
  - Specify a cascading style sheet to apply to your output
- **DOM="external-file"**
  - Specify that the ODS document object model is written to the SAS log or to an external file.
- **DPI=**
  - Specify the image resolution in dots per inch for output images
- **EXCLUDE exclusion(s) | ALL | NONE**
Exclude output objects from the destination

**FILE=’external-file’ | fileref**
Specify the output file.

**GFOOTNOTE | NOGFOOTNOTE**
Specify the location where footnotes are printed in the graphics output

**GTITLE | NOGTITLE**
Control the location where titles are printed in the graphics output

**KEYWORDS=’keywords-text’**
Insert a string of keywords into the output file's metadata

**NEWFILE= starting-point**
Create a new file at the specified starting-point

**PACKAGE <package-name>**
Specify that the output from the destination be added to an ODS package

**PDFMARK**
Insert special markup that is used when converting a PostScript file to a PDF file

**PDFNOTE | NOPDFNOTE**
Control whether notes are added to a PDF file for items that are associated with the FLYOVER= style attribute

**SELECT selection(s) | ALL | NONE**
Select output objects for the destination

**SHOW**
Write to the SAS log the current selection or exclusion list for the destination

**STARTPAGE=NEVER | NO | NOW | YES | BYGROUP**
Control page breaks

**STYLE=style-template**
Specify the style template to use in writing the PDF output

**TEXT=’text-string’**
Insert text into your output

**UNIFORM**
For multi-page tables, provide uniformity from page to page within a single table

### Without Arguments

If you use the ODS PS statement without an action or options, then it opens the PS destination and creates PostScript output.

### Actions

The following actions are available for the ODS PS statement:

**CLOSE**
closes the destination and any files that are associated with it.

**Tip**  
When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

**EXCLUDE exclusion(s) | ALL | NONE**
excludes one or more output objects from the destination.

<table>
<thead>
<tr>
<th>Default</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Restriction</strong></td>
<td>A destination must be open for this action to take effect.</td>
</tr>
</tbody>
</table>
See “ODS EXCLUDE Statement” on page 321

**SELECT selection(s) | ALL | NONE**
selects output objects for the specified destination.

Default ALL

Restriction A destination must be open for this action to take effect.

See “ODS SELECT Statement” on page 758

**SHOW**
writes the current selection list or exclusion list for the destination to the SAS log.

Restriction The destination must be open for this action to take effect.

Tip If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

See “ODS SHOW Statement” on page 771

**Optional Arguments**

**ANCHOR='anchor-name'**
specifies the root name for the anchor tag that identifies each output object in the current file.

Each output object must have an anchor tag for the bookmarks to reference. The references are automatically created by ODS. These references, point to the name of an anchor. Therefore, each anchor name in a file must be unique.

anchor-name
is the root name for the anchor tag that identifies each output object in the current file.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR='TABULATE', then ODS names the first anchor tabulate. The second anchor is named tabulate1, the third is named tabulate2, and so on.

Requirement You must enclose anchor-name in quotation marks.

Alias NAMED_DEST= | BOOKMARK=

Restriction Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

Tips You can change anchor names as often as you want by submitting the ANCHOR= option in a valid statement anywhere in your program. After you have specified an anchor name, it remains in effect until you specify a new one.

Specifying new anchor names at various points in your program is useful when you want to link to specific parts of your PRINTER
output. Because you can control where the anchor name changes, you know in advance what the anchor name is at those points.

**AUTHOR= 'author-text'**

inserts the text string that you specify as the author into the metadata of a file.

*author-text*

is the text in the metadata of an open file that indicates the author.

**Restrictions**

Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

The AUTHOR= option takes effect only if specified at the opening of a file.

**Requirement**

You must enclose *author-text* in quotation marks.

**BASE='base-text'**

specifies the text to use as the first part of all references that ODS creates in the output file.

*base-text*

is the text that ODS uses as the first part of all references that ODS creates in the file.

Consider this specification:

**BASE='http://www.your-company.com/local-url/'**

In this case, ODS creates references that begin with the string http:// www.your-company.com/local-url/. The appropriate anchor-name completes the link.

**Restriction**

Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and the ODS PRINTER statement with the PDFMARK option specified.

**Requirement**

You must enclose *base-text* in quotation marks.

**BOOKMARKLIST= HIDE | NONE | SHOW**

specifies whether to generate and display the list of bookmarks for PDF and PS files.

**HIDE**

generates a list of bookmarks for your PDF and PS files. The bookmarks are not automatically displayed when you open the PDF and PS files.

**NONE**

specifies not to generate a list of bookmarks for your PDF and PS files.

**Aliases**

NO | OFF

NOBOOKMARKLIST is an alias for BOOKMARKLIST=NONE | NO | OFF.

**SHOW**

generates a list of bookmarks for your PDF and PS files. The bookmarks are automatically displayed when you open the PDF and PS files.

**Aliases**

YES | ON
BOOKMARKLIST is an alias for BOOKMARKLIST=SHOW | YES | ON.

Example

“Example 2: Creating a Printable Table of Contents” on page 598

Default

SHOW

Restrictions

This option can be set only when you first open the destination.

This option has an effect only when creating PDF, PDFMARK, PS output.

Interaction

The NOTOC option specifies BOOKMARKLIST= OFF and CONTENTS= OFF.

Note

The generation of the bookmarks is not affected by the setting of this option. Bookmarks are generated by the BOOKMARKGEN= option.

Example

“Example 2: Creating a Printable Table of Contents” on page 598

BOOKMARKGEN | NOBOOKMARKGEN | BOOKMARKGEN=

controls the generation of bookmarks in PDF and PS files.

BOOKMARKGEN

specifies to generate bookmarks in PDF and PS files.

BOOKMARKGEN=

controls the generation of bookmarks in PDF and PS files.

NO

specifies not to generate bookmarks in PDF and PS files.

Alias OFF

YES

specifies to generate bookmarks in PDF and PS files.

Alias ON

NOBOOKMARKGEN

specifies not to generate bookmarks in the PDF and PS files.

Default

YES or BOOKMARKGEN

Interaction

If you set BOOKMARKGEN=NO, then the BOOKMARKLIST option is set to NO also.

BOX_SIZING=(CONTENT_BOX | BORDER_BOX)

specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing.

COLOR=FULL | GRAY | MONO | NO | YES

applies the specified color scheme to your output.
FULL
creates full color output for both text and graphics.

GRAY
creates gray scale output for both text and graphics.
Alias GREY

MONO
creates monochromatic output for both text and graphics.
Alias BW

NO
does not use all the color information that the style template provides. If you specify COLOR=NO, then the destination does this:
• generates black and white output
• creates all text and rules in black
• sets the SAS/GRAPH device to produce SAS/GRAPH output in gray scale
• ignores specifications for a background color from the style template except for the purposes of determining whether to print rules for the table

YES
uses all the color information that a style template provides, including background color. To print in color, you must also do the following:
• use a printer that is capable of printing in color.
• use the COLORPRINTING SAS system option. For information about the COLORPRINTING system option, see SAS System Options: Reference.

Default YES
Tip If you choose color output for a printer that does not support color, then your output might be difficult to read.

COLUMNS=n
specifies the number of columns to create on each page of output.

n is the number columns per page.

Default 1

CSSSTYLE='file-specification'<(media-type-1<…media-type-10>)>
specifies a cascading style sheet to apply to your output.
file-specification specifies a file, fileref, or URL that contains CSS code.
file-specification is one of the following:
"external-file"
is the name of the external file.
Requirement You must enclose external-file in quotation marks.
**fileref**

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in *SAS Statements: Reference*.

"URL"

is a URL to an external file.

**Requirement** You must enclose URL in quotation marks.

(media-type-1<.. media-type-10>)
specifies one or more media blocks that correspond to the type of media that your output is rendered on. CSS uses media type blocks to specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not contained in any media blocks. By using the media-type suboption, in addition to the general CSS code, you can import the section of a CSS file intended only for a specific media type.

**Default** If no media-type is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

**Range** You can specify up to ten different media types.

**Requirements** You must enclose media-type in parentheses.

You must specify media-type next to the file-specification specified by the CSSSTYLE= option.

**Tip** If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

**Restriction** The CSSSTYLE= option does not affect SAS/GRAPH output.

**Requirement** CSS files must be written in the same type of CSS produced by the ODS HTML statement. To view the CSS code that ODS creates, you can do one of the following:

- specify the ODS TRACE DOM statement
- specify the DOM option

**Interaction** If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

**See** For complete documentation about ODS and Cascading Style Sheets, see *SAS Output Delivery System: Advanced Topics*

**Example** “Example 6: Applying a CSS File to ODS Output” on page 527
DOM="external-file">
  specifies that the ODS document object model is written to the SAS log or an external file.

  external-file
    is the name of an external output file.

  Requirement You must enclose external-file in quotation marks.

See For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in SAS Output Delivery System: Advanced Topics.

DPI=
  specifies the image resolution for output files.

  Default 150

  Restriction The DPI= option takes effect only if specified at the opening of a file.

  CAUTION When using high DPI= or DPI_IMAGE= values (values over 600), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500 or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.

FILE='external-file' | fileref
  specifies the output file.

  external-file
    is the name of an external file.

  Requirement You must enclose external-file in quotation marks.

  fileref
    is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

  Restriction The FILE=fileref option cannot be used in conjunction with the NEWFILE= option.

See For information about the FILENAME statement, see SAS Statements: Reference.

  Default If you do not specify an output file, then ODS writes to the file that is specified by two SAS system options. Use the SYSPRINT= system option if you are using the Windows operating environment and do not specify any of the following options: PCL, PDFMARK, POSTSCRIPT, PS, or SAS. Use the PRINTERPATH= system option in all other cases. If the system option does not specify a file, then ODS writes to the default printer. For more information, see the PRINTER= option.

  Interaction In an ODS printer family statement that refers to an open ODS PRINTER destination, the FILE= option forces ODS to close the destination and all files that are associated with it. ODS then opens a new instance of the destination. For more information, see “Opening and Closing the PRINTER Destination” on page 708.
For information about the FILENAME statement, see *SAS Statements: Reference*.

**GFOOTNOTE | NOGFOOTNOTE**  
controls the location of the footnotes that are defined by the graphics program that generates the Printer output.  

**GFOOTNOTE**  
includes all of the currently defined footnotes within the graphics output.  

**NOGFOOTNOTE**  
prevents all of the currently defined footnotes from appearing in the graphics file. Instead, they become part of the Printer file.  

**Default**  
**GFOOTNOTE**  

**Restriction**  
This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.  

**See**  
For more information, see “Customizing Titles and Footnotes” on page 40.

**GTITLE | NOGTITLE**  
controls the location of the titles that are defined by the graphics program that generates the Printer output.  

**GTITLE**  
includes all of the currently defined titles within the graphics output.  

**NOGTITLE**  
prevents all of the currently defined titles from appearing in the graphics output. Instead, the titles become part of the Printer file.  

**Default**  
**GTITLE**  

**Restriction**  
This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.  

**See**  
For more information, see “Customizing Titles and Footnotes” on page 40.

**(<ID=> identifier)**  
enables you to open multiple instances of the same destination at the same time. Each instance can have different options.  

**identifier**  
can be numeric or can be a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numerals.  

**Restriction**  
If identifier is numeric, it must be a positive integer.  

**Requirement**  
The ID= option must be specified immediately after the destination name.
KEYWORDS='keywords-text'
inserts a string of keywords into the output file's metadata. The keywords enable a
document management system to do topic-based searches.

keywords-text
is the string of keywords.

Restrictions
Use this option only with the ODS PDF statement, the ODS PS
statement with the PDFMARK option specified, and the ODS
PRINTER statement with the PDFMARK option specified.

The KEYWORDS= option takes effect only if specified at the
opening of a file.

Requirement
You must enclose keywords-text in quotation marks.

NEWFILE= starting-point
creates a new file at the specified starting-point.

starting-point
is the location in the output where you want to create a new file.

ODS automatically names new files by incrementing the name of the file. In the
following example, ODS names the first file REPORT.PS. Additional body files
are named REPORT1.PS, REPORT2.PS, and so on.

Example:
FILE= 'REPORT.PS'

starting-point can be one of the following:

BYGROUP
starts a new file for the results of each BY group.

NONE
writes all output to the file that is currently open.

OUTPUT
starts a new file for each output object. For SAS/GRAPH this means that
ODS creates a new file for each SAS/GRAPH output file that the program
generates.

Alias TABLE

PAGE
starts a new file for each page of output. A page break occurs when a
procedure explicitly starts a new page (not because the page size was
exceeded) or when you start a new procedure.

PROC
starts a body file each time you start a new procedure.

Default NONE

Restrictions
The NEWFILE= option cannot be used in conjunction with the
FILE=fileref option.

The NEWFILE= option cannot be used if you are sending output to a
physical printer.
Tips

If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first file `MAY5.PS`. Additional body files are named `MAY6.PS`, `MAY7.PS`, and so on.

Example:

```plaintext
FILE= 'MAY5.PS'
```

**PACKAGE <package-name>**

specifies that the output from the destination be added to a package.

`package-name` specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

See “ODS PACKAGE Statement” on page 554

**PDFMARK**

enables ODS to insert special tags into a PostScript file. When you use software such as Adobe Acrobat (not Adobe Viewer), Acrobat Distiller interprets the tags to create a PDF file that contains the following items:

- bookmarks for each section of the output and for each table.
- references for items that are associated with the URL= style attribute.
- notes for items that are associated with the FLYOVER= style attribute. Notes are optional, and are based on the PDFNOTE option.
- author, keywords, subject, and title in the metadata of a file.

Default

Because using PDFMARK implies PostScript output, SAS automatically uses the PostScript driver that SAS supplies with this option.

Restriction

You cannot use the PRINTER= option with the PDFMARK option.

Requirement

To create a PDF file, you must use specialized software, such as Adobe Acrobat Distiller to convert the marked-up PostScript file into a PDF formatted file.

Interaction

In an ODS printer family statement that refers to an open ODS PRINTER destination, the PDFMARK option forces ODS to close the destination and all files that are associated with it. ODS then opens new instance of the destination. For more information, see “Opening and Closing the PRINTER Destination” on page 708.

Tip

Use this option only if you plan to distill the output. Otherwise, it uses excess resources and does not enhance the results.

**PDFNOTE | NOPDFNOTE**

controls whether notes are added to a PDF file for items that are associated with the FLYOVER= style attribute.

**PDFNOTE**

adds notes to a PDF file for items that are associated with the FLYOVER= style attribute.
NOPDFNOTE
modifies the behavior of PDFMARK so that notes are not added to the file for items that are associated with the FLYOVER= style attribute.

Default PDFNOTE

Restriction Use this option only with the ODS PDF statement, the ODS PS statement with the PDFMARK option specified, and ODS PRINTER statement with the PDFMARK option specified.

STARTPAGE=NEVER | NO | NOW | YES | BYGROUP
controls page breaks.

BYGROUP
specifies to insert page breaks after each BY group.

NEVER
specifies not to insert page breaks, even before graphics procedures.

CAUTION:
Each graph normally requires an entire page. The default behavior forces a new page after a graphics procedure. STARTPAGE=NEVER turns off that behavior, so specifying STARTPAGE= NEVER might cause graphics to overprint.

NO
specifies that no new pages be inserted at the beginning of each procedure, or within certain procedures, even if new pages are requested by the procedure code. A new page begins only when a page is filled or when you specify STARTPAGE=NOW.

CAUTION:
Each graph normally requires an entire page. The default behavior forces a new page after a graphics procedure, even if you use STARTPAGE=NO. STARTPAGE=NEVER turns off that behavior, so specifying STARTPAGE= NEVER might cause graphics to overprint.

Alias OFF

Tip When you specify STARTPAGE=NO, system titles and footnotes are still produced only at the top and bottom of each physical page, regardless of the setting of this option. Thus, some system titles and footnotes that you specify might not appear when this option is specified.

Examples
“Example 5: Combining a Table and Image on the Same Page” on page 608
“Example 6: Adding Text That Imitates a System Title” on page 611
“Example 7: Toggling Page Breaks” on page 615
“Example 8: Suppressing a Page Break” on page 619

NOW
forces the immediate insertion of a new page.
Tip
This option is useful primarily when the current value of the STARTPAGE= option is NO. Otherwise, each new procedure forces a new page automatically.

Example
“Example 7: Toggling Page Breaks” on page 615

YES
inserts a new page at the beginning of each procedure, and within certain procedures, as requested by the procedure code.

Alias
ON

Default
YES

STYLE=style-template
specifies the style template to use in writing the printer output.

Default
If you do not specify a style template, then ODS uses the style template that is specified in the SAS registry subkey: ODS ⇒ DESTINATIONS ⇒ PRINTER. By default, this value is Pearl for the PRINTER, PDF, PS and PCL destinations.

Note
If you are using SAS Studio, you do not need to specify the STYLE= option. You can go to Preferences ⇒ Results and change the style from the drop-down list for your selected destination.

See
For a complete discussion of style templates, see “Working with Styles” in SAS Output Delivery System: Procedures Guide.

For instructions on making your own user-defined style templates, see “TEMPLATE Procedure: Creating a Style Template” in SAS Output Delivery System: Procedures Guide.

Examples
“Example 3: Customizing the Table of Contents” on page 600

“Example 6: Adding Text That Imitates a System Title” on page 611

TEXT='text-string'
inserts a text string into your output.

text-string
is the text that you want to insert into your output.

Requirement
You must enclose text-string in quotation marks.

Tip
If you are submitting more than one procedure step and you do not specify the STARTPAGE=NO option, each procedure forces a new page before the output. Therefore, any text that you specify with TEXT= is on the same page as the previous procedure.

See
“Example: Conditionally Excluding Output Objects and Sending Them to Different Output Destinations” on page 325

UNIFORM
for multiple page tables, ensures uniformity from page to page within a single table. When the UNIFORM option is in effect, ODS reads the entire table first, so that it
can determine the column widths that are necessary to accommodate all the data. These column widths are applied to all pages of a multiple page table.

**Default**
If you do not specify the UNIFORM option, then ODS prints a table one page at a time. This approach ensures that SAS does not run out of memory while processing very large tables. However, it can also mean that column widths vary from one page to the next.

**Note**
With BY-group processing, SAS writes the results of each BY group to a separate table, so the output might not be uniform across BY groups.

**Tip**
The UNIFORM option can cause SAS to run out of memory if you are printing a very large table. If this happens, then you can explicitly set the width of each of the columns in the table, and then print the table one page at a time. To do so, you must edit the table template that you use. For more information, see “What You Can Do with Table Templates” in *SAS Output Delivery System: Procedures Guide*.

**Details**
The ODS PS statement is part of the ODS printer family of statements. Statements in the printer family open the PCL, PDF, PRINTER, or PS destination, producing output that is suitable for a high-resolution printer. The ODS PCL, ODS PDF, and ODS PRINTER statements are also members of the ODS printer family of statements.

**Opening and Closing the PS Destination**
You can modify an open PS destination with many ODS PS options. However, the FILE=, PDFMARK, and SAS options perform the following actions on an open PS destination:

- close the open destination referred to in the ODS PS statement
- close any files associated with the open PS destination
- open a new instance of the PS destination

If you use one of these options, it is best if you explicitly close the destination yourself.

**See Also**
- “The Third-Party Formatted Destinations” on page 35

**Statements**
- “ODS PCL Statement” on page 565
- “ODS PDF Statement” on page 575
- “ODS PRINTER Statement” on page 690

**ODS RESULTS Statement**
Tracks ODS output in the Results window.

**Valid in:** Anywhere

**Category:** ODS: Output Control
Alias: ODS RESULTS|NORESULTS;
Restriction: Valid in a windowing environment only, not in batch mode.

Syntax

ODS RESULTS ON | OFF;
ODS RESULTS= ON | OFF;

Required Arguments

ON
tracks output that ODS generates in the Results window.

OFF
turns off the tracking of output that ODS generates in the Results window.

Details

Using ODS RESULTS ON sends all output to the Results window. This is the default setting. Using ODS RESULTS OFF disables ODS tracking, and output is not sent to the Results window. The OFF option is recommended for long-running jobs, such as regression analyses, when you do not want to track all of the output.

ODS RTF Statement

Opens, manages, or closes the RTF destination, which produces output written in Rich Text Format for use with Microsoft Word 2002.

Valid in: Anywhere
Category: ODS: Third-Party Formatted
Default: The default style for the RTF destination is RTF.
Restriction: When a table that is generated using SAS is copied and pasted into or imported into a Word document, it is assigned the Word style "Normal". As a result, most of the custom ODS formatting is lost, and the output does not display as expected.
Interactions: By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC PLOT or PROC CHART), ODS formats the output in SAS Monospace font. If you are creating output that will be viewed in an operating environment that does not have SAS software installed, this output will not be displayed correctly. The SAS Monospace font is not recognized if SAS is not installed. For the correct display of your document, include the following statement before your SAS program:

```sas
OPTIONS FORMCHAR="|----|+|--|--|--"; 
```

To change the page orientation of the RTF output, specify the system option ORIENTATION=. To change the orientation, you will need to trigger the change by issuing the ODS RTF statement after the global options statement. See "Example 3: RTF Interaction with the ORIENTATION= System Option" on page 755 for details.

Tips: Microsoft Word 2002 is the current, official, minimum level that is supported. However, no problems have been found with Microsoft Word 2000 and SAS RTF files.
When producing large tables, use the ODS TAGSETS.RTF statement. For detailed information, see “ODS TAGSETS.RTF Statement” on page 803.

Syntax

**ODS RTF**

```
<(<ID=> identifier)> action;
```

```
ODS RTF <(<ID=> identifier)> <option(s)> ;
```

**Summary of Optional Arguments**

- **(ID= identifier)**
  - Open multiple instances of the same destination at the same time
- **ANCHOR= 'anchor-name'**
  - Specify a unique base name for the anchor tag that identifies each output object in the current body file
- **AUTHOR= 'author-text'**
  - Specify the text string that identifies the author. This text string is inserted into the metadata of a file.
- **BASE= 'base-text'**
  - Specify text to use as the first part of all links and references that ODS creates in output files
- **BODYTITLE**
  - Specify that the titles and footnotes are to be placed into the body of the RTF document and not into the header and footer sections
- **BODYTITLE_AUX**
  - Specify that the titles and footnotes are to be placed into the body of the RTF document and not into the header and footer sections. The titles and footnotes are also placed into cells or tables
- **BOX_SIZING=(CONTENT_BOX | BORDER_BOX)**
  - Specify how to measure the width of cells. Use to override the default value of BOX_SIZING for a destination
- **CLOSE**
  - Close the destination and the file that is associated with it
- **COLUMNS= n | MAX**
  - Specify the number of columns to create on each page of output
- **CONTENTS**
  - Specify whether to produce a table of contents page
- **CSSSTYLE= 'file-specification'(media-type-1<…media-type-10>)**
  - Specify a cascading style sheet to apply to your output
- **DEVICE= device-driver**
  - Specify a device for the RTF output destination
- **DOM<="external-file">**
  - Specify that the ODS document object model is written to the SAS log or to an external file.
- **ENCODING= local-character-set-encoding**
  - Override the encoding for input or output processing (transcodes) of external files
- **EXCLUDE exclusion(s) | ALL | NONE**
  - Exclude output objects from the destination
- **FILE= 'external-file' | fileref**
Open the ODS RTF destination and specify the name of the file to which to write information.

**GFOOTNOTE | NOGFOOTNOTE**
Specify the location where footnotes are printed in the graphics output.

**GTITLE | NOGTITLE**
Control the location where titles are printed in the graphics output.

**IMAGE_DPI**
Specify the image resolution for the graphical output.

**KEEPN | NOKEEPN**
Control where tables split on a page.

**NEWFILE= starting-point**
Create a new body file at the specified starting point.

**NOTOC_DATA**
Specify whether contents data is inserted into the RTF file.

**OPERATOR= 'text-string'**
Insert the text that you specify into the metadata of the RTF file.

**PACKAGE <package-name>**
Specify that the output from the destination be added to an ODS package.

**PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)**
Specify the location of an aggregate storage location or a SAS catalog for all RTF files.

**PREPAGE='text-string'**
Specify a text string that occurs before a table on a page.

**RECORD_SEPARATOR= 'alternative-separator' | NONE**
Specify an alternative character or string to separate lines in the output files.

**SASDATE**
Write to the RTF file the time and date that you started your SAS session.

**SELECT selection(s) | ALL | NONE**
Select output objects for the destination.

**SHOW**
Write to the SAS log the current selection or exclusion list for the destination.

**STARTPAGE= BYGROUP | YES | NO | NOW**
Control page breaks.

**STYLE= style-template**
Specify a style template to use in writing the RTF files.

**TEXT='text-string'**
Insert text into your RTF output.

**TITLE='title-text'**
Insert the text string that you want as your title into the metadata of a file.

**TOC_DATA | NOTOC_DATA**
Specify whether contents data is inserted into the RTF file.

**TRANTAB= translation-table**
Specify a translation table to use when you transcode a file for output.

**Actions**
The following actions are available for the ODS RTF statement:

**CLOSE**
closes the destination and any files that are associated with it.
Tip When an ODS destination is closed, ODS does not send output to that
destination. Closing an unneeded destination conserves system resources.

**EXCLUDE** exclusion(s) | ALL | NONE
excludes one or more output objects from the destination.

Default NONE

Restriction A destination must be open for this action to take effect.

See “ODS EXCLUDE Statement” on page 321

**SELECT** selection(s) | ALL | NONE
selects output objects for the specified destination.

Default ALL

Restriction A destination must be open for this action to take effect.

See “ODS SELECT Statement” on page 758

**SHOW** writes the current selection list or exclusion list for the destination to the SAS log.

Restriction The destination must be open for this action to take effect.

Tip If the selection or exclusion list is the default list (SELECT ALL), then
SHOW also writes the entire selection or exclusion list. For
information about selection and exclusion lists, see “Selection and
Exclusion Lists” on page 39.

See “ODS SHOW Statement” on page 771

**Optional Arguments**

**ANCHOR= 'anchor-name'**
specifies the base name for the RTF anchor tag that identifies each output object in
the current file.

Each output object must have an anchor tag to which other files link or reference.
The references, which ODS automatically creates, point to the name of an anchor.
Therefore, each anchor name in a file must be unique.

anchor-name is the base name for the RTF anchor tag that identifies each output object in the
current file.

ODS increments the name that you specify and creates unique anchor names. For
example, if you specify ANCHOR= 'tabulate', then ODS names the first anchor
**tabulate**. The second anchor is named **tabulatel**; the third is named
**tabulate2**, and so on.

Requirement You must enclose anchor-name in quotation marks.

**Alias** NAMED_DEST= | BOOKMARK=

**Tips** It is useful to specify new anchor names at various points in your program
when you want other RTF files to link to specific parts of your RTF output.
Because you can control where the anchor name changes, you know in advance what the anchor name is at those points.

You can change anchor names as often as you want by submitting the ANCHOR= option in an ODS RTF statement anywhere in your program. After you specify an anchor name, it remains in effect until you specify a new one.

**AUTHOR='author-text'**

inserts the text string that you specify as the author into the metadata of a file.

*author-text*

is the text in the metadata of an open file that indicates the author.

**Requirement** You must enclose *author-text* in quotation marks.

**BASE='base-text'**

specifies the text to use as the first part of references that ODS creates in the output file.

*base-text*

is the text that ODS uses as the first part of all references that ODS creates in the file.

Consider this specification:

```
BASE='http://www.your-company.com/local-url/'
```

In this case, ODS creates links that begin with the string `http://www.your-company.com/local-url/`.

**Requirement** You must enclose *base-text* in quotation marks.

**BODYTITLE**

specifies that SAS titles and footnotes are placed into the body of the RTF document instead of into the headers and footers section of the RTF document.

**Restriction** The BODYTITLE option can be specified only when you create a new RTF file.

**Interactions** When you set the STARTPAGE= option to YES (the default), ODS inserts a new page at the start of each procedure. ODS relies on Word to place headers and footers correctly before and after the procedures. When you specify BODYTITLE, titles and footnotes are removed from the header and footer sections of the RTF document. Titles and footnotes are then placed into the body of the document, and are appended to every TABLE. Therefore, when you set the STARTPAGE= option to YES and specify the BODYTITLE option, the titles and footnotes might not repeat on every page. For example, if there is a table that spans multiple pages, the title is on the first page only, and the footnote is on the last page only.

When you specify the BODYTITLE option, Microsoft Word no longer controls the placement of the header and footer text. However, Microsoft Word still controls other header and footer information, such as page number and date.

**Tip** The background is not honored on the title cells.
BODYTITLE_AUX option. Use the BODYTITLE_AUX option when you want titles and footnotes placed in tables in the body of the RTF document.

**BODYTITLE_AUX**

specifies that SAS titles and footnotes be placed into the body of the RTF document instead of into the headers and footers section of the RTF document. These titles and footnotes are put into cells, which allows titles and footnotes to be centered, left-justified, or right-justified.

**Restriction**

You can specify the BODYTITLE_AUX option only when you are creating a new RTF file.

**Interactions**

When you set the STARTPAGE= option to YES (the default), ODS inserts a new page at the start of each procedure. ODS relies on Word for the correct placement of headers and footers before and after the procedures. When you specify BODYTITLE_AUX, titles and footnotes are removed from the header and footer sections of the RTF document. Titles and footnotes are then placed into the body of the document, and they are appended to every TABLE. Therefore, when you set the STARTPAGE= option to YES and you specify the BODYTITLE_AUX option, the titles and footnotes might not repeat on every page. For example, if there is a table that spans multiple pages, then the title is placed on the first page only, and the footnote is placed on the last page only.

When you specify the BODYTITLE_AUX option, Microsoft Word no longer controls the placement of the header and footer text. However, Microsoft Word still controls other header and footer information, such as page number and date.

**Example**

“Example 2: Justifying Title and Footnotes When You Specify the BODYTITLE_AUX Option” on page 753

**BOX_SIZING=(CONTENT_BOX | BORDER_BOX)**

specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing.

**COLUMNS= n | MAX**

specifies the number of columns to place across each page of output.

\( n \)

is the number of one-inch columns that you want on the page.

**MAX**

specifies the maximum number of columns for the paper size and margin setting. This value is dependent upon the paper size and page orientation.

**Default**

The number of columns that fit on the page
Interaction

When you specify the `COLUMNS=` option, the `STARTPAGE=NO` option is not honored.

Tips

Titles are considered tables and not RTF instructions in Measured RTF (ODS TAGSETS.RTF statement). When you use the `COLUMNS=` option with Measured RTF, titles appear at the top of each column. However, ODS truncates the titles to fit the column width.

If you specify a value greater than the maximum number of columns that can fit on the page, a note is written to the SAS log that states what the maximum value can be for that page.

CONTENTS

produces a table of contents page for RTF documents that are opened in Microsoft Word. The table of contents page contains a Table of Contents field, which puts all of the contents information that is embedded in the document into a table of contents. To expand the table of contents, right-click under the title in Microsoft Word and select Update Field from the selection list.

Restriction

Do not use the CONTENTS option with the NEWFILE option.

Tips

To go to a specific topic in the document, you can double-click or hold down the Ctrl key and click on the topic in the table of contents. You might have to configure Microsoft Word to use the Ctrl + click method by selecting Tools ⇒ Options ⇒ Edit and checking Use Ctrl + Click to follow hyperlink.

You must specify the TOC_DATA option to view the text that is captured in the Table of Contents. If not, the Table of Contents page displays the error message "Error! No table of contents entries found." NOTOC_DATA is the default option that is used.

See

TOC_DATA option

Example

“Example 1: Creating a Table of Contents from Embedded Data” on page 749

CSSSTYLE='file-specification'&lt;(media-type-1&hellip;media-type-10)&gt;

specifies a cascading style sheet to apply to your output.

file-specification

specifies a file, fileref, or URL that contains CSS code.

file-specification is one of the following:

"external-file"

is the name of the external file.

Requirement

You must enclose external-file in quotation marks.

fileref

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See

For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

"URL"

is a URL to an external file.
Requirement
You must enclose URL in quotation marks.

\[ (media-type-1<.. media-type-10>) \]

specifies one or more media blocks that correspond to the type of media that your output is rendered on. CSS uses media type blocks to specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not contained in any media blocks. By using the media-type suboption, in addition to the general CSS code, you can import the section of a CSS file intended only for a specific media type.

Default
If no media-type is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

Range
You can specify up to ten different media types.

Requirements
You must enclose media-type in parentheses.

You must specify media-type next to the file-specification specified by the CSSSTYLE= option.

Tip
If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

Restriction
The CSSSTYLE= option does not affect SAS/GRAPH output.

Requirement
CSS files must be written in the same type of CSS produced by the ODS HTML statement. To view the CSS code that ODS creates, you can do one of the following:
• specify the ODS TRACE DOM statement
• specify the DOM option

Interaction
If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

See
For complete documentation about ODS and Cascading Style Sheets, see *SAS Output Delivery System: Advanced Topics*

Example
“Example 6: Applying a CSS File to ODS Output” on page 527

DEVICE= device-driver
specifies the name of a device driver. ODS automatically selects an optimal default device for each open output destination.

The following table lists the default devices for the most common ODS output destinations. These default devices are used when graphics are created using SAS/GRAPH or ODS Graphics. In the third maintenance release of SAS 9.4, EPUB3 is the default EPUB destination. EPUB2 was the default EPUB version in prior releases of SAS 9.4. This default is set in the Registry. For a complete list of
supported devices and file types, see “Supported File Types for Output Destinations” on page 349.

Table 6.18 Default Devices for ODS Output Destinations

<table>
<thead>
<tr>
<th>Output Destination</th>
<th>Default Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPUB2</td>
<td>PNG</td>
</tr>
<tr>
<td>EPUB3 (EPUB)</td>
<td>SVG</td>
</tr>
<tr>
<td>HTML</td>
<td>PNG</td>
</tr>
<tr>
<td>HTML5</td>
<td>SVG</td>
</tr>
<tr>
<td>LISTING</td>
<td>PNG</td>
</tr>
<tr>
<td>Measured RTF</td>
<td>EMF</td>
</tr>
<tr>
<td>RTF</td>
<td>EMF</td>
</tr>
<tr>
<td>Markup Tagsets</td>
<td>PNG</td>
</tr>
</tbody>
</table>

Tips

Specifying a device on the ODS DEVICE= option takes precedence over the SAS global option and the graphics option.

For information about using the SAS Registry to change defaults, see “Changing SAS Registry Settings for ODS” on page 40.

DOM=""external-file"">

specifies that the ODS document object model is written to the SAS log or an external file.

external-file

is the name of an external output file.

Requirement You must enclose external-file in quotation marks.

See For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in SAS Output Delivery System: Advanced Topics.

ENCODING= local-character-set-encoding

overrides the encoding for input or output processing (transcodes) of external files.

See For information about the ENCODING= option, see SAS National Language Support (NLS): Reference Guide.
FILE= 'external-file' | fileref

opens the RTF destination and specifies the RTF file or SAS catalog to which to write. This file remains open until you do one of the following actions:

• Close the RTF destination with ODS RTF CLOSE or ODS ALL CLOSE.
• Specify a different file to which to write.

external-file

is the name of an external file to which to write.

Requirement You must enclose external-file in quotation marks.

fileref

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

Restriction You cannot use the FILE=fileref option with the NEWFILE= option.

See The section on statements in SAS Statements: Reference for information about the FILENAME statement.

Alias BODY=

Interaction In an ODS RTF statement that refers to an open RTF destination, the FILE= option forces ODS to close the destination and all files that are associated with it. ODS then opens a new instance of the destination. For more information, see “Opening and Closing the RTF Destination” on page 748.

See NEWFILE= option

GFOOTNOTE | NOGFOOTNOTE

controls the location of the footnotes that are defined by the graphics program that generates the RTF output.

GFOOTNOTE

includes all of the currently defined footnotes within the graphics output.

NOGFOOTNOTE

prevents all of the currently defined footnotes from appearing in the graphics file. Instead, they become part of the RTF file.

Default GFOOTNOTE

Restriction This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

GTITLE | NOGTITLE

controls the location of the titles that are defined by the graphics program that generates the RTF output.

GTITLE

includes all of the currently defined titles within the graphics output that is called by the body file.
NOGTITLE
prevents all of the currently defined titles from appearing in the graphics output. Instead, the titles become part of the RTF file.

Default GTITLE

Restriction This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

(ID= identifier)
enables you to run multiple instances of the same destination at the same time. Each instance can have different options.

identifier specifies another instance of the destination that is already open. identifier is numeric or a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numeric characters.

Restriction If identifier is numeric, it must be a positive integer.

Requirement You must specify the ID= option immediately after the destination name.

Tip You can omit the ID= option and instead use a name or a number to identify the instance.

Example “Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594

IMAGE_DPI
specifies the image resolution for graphical output.

Alias DPI=

Default 200

CAUTION Caution: When using high DPI= or DPI_IMAGE values (values over 1000), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500M or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.

KEEPN | NOKEEPN
controls where tables split on a page.

KEEPN ODS allows table splits only if the entire table cannot fit on one page.

NOKEEPN ODS lets a table split at a page break.

Tip Although KEEPN minimizes page breaks in tables, it might use substantially more paper than NOKEEPN. This is because the KEEPN option issues a page break before starting to print any table that does not fit on the remainder of the page.

NEWFILE= starting-point
creates a new file at the specified starting-point.
starting-point can be one of the following:

BYGROUP
starts a new file for the results of each BY group.

NONE
writes all output to the body file that is currently open.

OUTPUT
starts a new file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

Alias TABLE

PROC
starts a new file each time you start a new procedure.

Default NONE

Restriction You cannot use the NEWFILE= option with the FILE=fileref option.

Tip If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first body file MAY5.XML, and names additional body files MAY6.XML, MAY7.XML, and so on.

NOTOC_DATA
See the description of TOC_DATA in this section.

OPERATOR= 'text-string'
inserts the text that you specify into the metadata of the RTF file.

text-string
is the text in the metadata of a file that indicates the author.

Requirement You must enclose text-string in quotation marks.

PACKAGE <package-name>
specifies that the output from the destination be added to a package.

package-name
specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

See “ODS PACKAGE Statement ” on page 554

PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)
specifies the location of an aggregate storage location or a SAS catalog for all RTF files. If the GPATH= option is not specified, all graphics output files are written to the "aggregate-file-storage-specification" or libref.

'aggregate-file-storage-location'
specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement You must enclose aggregate-file-storage-location in quotation marks.
fileref

is a file reference that has been assigned to an aggregate storage location. Use the
FILENAME statement to assign a fileref.

Interaction If you use a fileref in the PATH= option, then ODS does not use
information from PATH= when it constructs links.


libref.catalog

specifies a SAS catalog to write to.

See “LIBNAME Statement” in SAS Statements: Reference.

URL= 'Uniform-Resource-Locator' | NONE

specifies a URL for the file-specification.

Uniform-Resource-Locator

is the URL that you specify. ODS uses this URL instead of the filename in all
the links and references that it creates to the file.

NONE

specifies that no information from the PATH= option appears in the links or
references.

Tip This option is useful for building output files that can be moved from one
location to another. The links from the contents and page files must be
constructed with a single-name URL, and the contents, page, and body files
must be in the same location.

Interaction If you use the BODY= or FILE= external file option in conjunction
with the PATH= option, the external file specification should not
include path information.

PREPAGE='text-string'

specifies a text string that occurs before a table on a page.

text-string

is the text at the top of the table, after the titles. The text is placed before any
tables created by the procedure.

Requirement You must enclose text-string in quotation marks.

RECORD_SEPARATOR= 'alternative-separator' | NONE

specifies an alternative record separator. This separator is a character or string that
separates lines in the output files.

Different operating environments use different separator characters. If you do not
specify a record separator, ODS formats the RTF files for the environment in which
you run the SAS job. However, if you are generating files in one operating
environment to view in another operating environment that uses a different separator
character, you can specify a record separator that is appropriate for the target
environment.

alternative-separator

represents one or more characters in hexadecimal or ASCII format. For example,
the following option specifies a record separator of a carriage-return character
and a linefeed character (on an ASCII file system):

RECORD_SEPARATOR= '0D0A'x
Operating Environment Information
In a mainframe environment, the following option specifies a record separator for a carriage-return character and a linefeed character for use with an ASCII file system:

```
RECORD_SEPARATOR= '0D25'x
```

Requirement You must enclose alternative-separator in quotation marks.

NONE produces RTF output that is appropriate for the environment in which you run the SAS job.

Operating Environment Information
In many operating environments, using a value of NONE has the same result as omitting the RECORD_SEPARATOR option.

Operating Environment Information
In a mainframe environment, by default, ODS produces a binary file that contains embedded record-separator characters. This approach means that the file is not restricted by the line-length restrictions on ASCII files. However, this also means that the lines are concatenated if you view the file in an editor. If you want to format the RTF files in a manner that enables you to read them with an editor, use RECORD_SEPARATOR= NONE. In this case, ODS writes one line of RTF at a time to the file. When you use a value of NONE, the logical record length of the file to which you are writing must be at least as long as the longest line that ODS produces. Otherwise, RTF might wrap to another line at an inappropriate place.

Aliases RECSEP=
RS=

SASDATE writes to the RTF file the time and the date that you started your SAS session.

Restriction You can specify SASDATE only when you open a new file. If you specify the option at any other time, ODS writes a warning message to the SAS log.

Interaction To reset the SAS session time that is input into the RTF file, use the DTRESET system option.

See For information about the DTRESET system option, see SAS System Options: Reference.

STARTPAGE= BYGROUP | YES | NO | NOW controls page breaks.

BYGROUP specifies to insert page breaks after each BY group.

YES inserts a new page at the start of each procedure and within certain procedures, as is requested by the procedure code.

Alias ON

Interactions When the STARTPAGE= option is set to YES (the default), ODS inserts a new page at the start of each procedure. ODS relies on
Word for the correct placement of headers and footers before and after the procedures. When you specify BODYTITLE, titles and footnotes are removed from the header and footer sections of the RTF document. Titles and footnotes are then placed into the body of the document, and they are appended to every TABLE. Therefore, when you set the STARTPAGE= option to YES and you specify the BODYTITLE option, the titles and footnotes might not repeat on every page. For example, if there is a table that spans multiple pages, the title appears on only the first page, and the footnote appears on only the last page.

Note that when you specify the BODYTITLE option, Microsoft Word no longer controls the placement of the headers and footers text. However, Word still controls other header and footer information, such as page number and date.

**NO**

instructs ODS not to insert any new pages at the start of each procedure or within certain procedures, even if the procedure code requests new pages. A new page begins only when a page is filled or when you specify STARTPAGE=NOW.

**Alias** NEVER

**Interaction** When you specify the COLUMNS= option, the STARTPAGE=NO option is not honored.

**NOW**

forces the immediate insertion of a new page.

**Tip** This option is useful primarily when the current value of the STARTPAGE= option is NO. Otherwise, each new procedure forces a new page automatically.

**Default** YES

**Tip** Specifying STARTPAGE= NO prevents forced page breaks. You can turn on forced page breaking again by specifying STARTPAGE=YES. You can insert a page break at any time by specifying STARTPAGE=NOW.

**STYLE= style-template**

specifies the style template for ODS to use to write the RTF files.

**style-template**

describes how to display the presentation aspects (color, font face, font size, and so on) of your SAS output. A style template determines the overall appearance of the documents that use that style template. Each style template consists of style elements.

**Note** If you are using SAS Studio, you do not need to specify the STYLE= option. You can go to Preferences ⇒ Results and change the style from the drop-down list for your selected destination.

**See** For a complete discussion of style templates, see “TEMPLATE Procedure: Creating a Style Template” in SAS Output Delivery System: Procedures Guide.
If you do not specify a style template, ODS uses the file that is specified in the SAS registry subkey: `ODS DESTINATIONS RTF`. By default, this value specifies RTF for traditional RTF and Measured RTF.

Do not use the function syntax for ODS ESCAPECHAR to produce superscripts with the ODS RTF destination. Instead, use the traditional ODS ESCAPECHAR syntax. For example, use the following statement:

```sas```
proc report style(report)=\[posttext="SuperScript test \super 2"];
```
instead of this statement:

```sas```
proc report style(report)=\[posttext="SuperScript test \^{super 2}"];
```

**TEXT= 'text-string'**

inserts text into your RTF output.

*text-string* is the text that you want to insert into your RTF output. You can also use TEXT= to annotate other output.

You must enclose a *text-string* in quotation marks.

**TITLE= 'title-text'**

inserts the text string that you specify as the title into the metadata of a file.

*title-text* is the text in the metadata of a file that indicates the title.

You must enclose a *title-text* in quotation marks.

**TOC_DATA | NOTOC_DATA**

specifies whether contents data is embedded in the RTF file as hidden text.

**NOTOC_DATA**

instructs ODS not to insert contents data into the RTF file.

**TOC_DATA**

instructs ODS to insert contents data into the RTF file.

**Tip**

Insertion of table of contents data can be resumed in the middle of a SAS program by including the following statement:

```sas```
ods rtf toc_data;
```

**Default**

NOTOC_DATA

**Tip**

To create a visible table of contents from the inserted table of contents data, specify the CONTENTS option.

**See**

CONTENTS option

**Example**

“Example 1: Creating a Table of Contents from Embedded Data” on page 749

**TRANTAB= translation-table**

specifies the translation table for ODS to use when it transcodes a file for output.

**See**

For more information, see “TRANTAB= System Option” in *SAS National Language Support (NLS): Reference Guide*. 
Details

Opening and Closing the RTF Destination
You can modify an open RTF destination with many ODS RTF options. However, the FILE= option performs the following actions on an open RTF destination:

• close the open destination referred to in the ODS RTF statement
• close any files associated with the open RTF destination
• open a new instance of the RTF destination

If you use the FILE= option, you should explicitly close the destination yourself.

Understanding How RTF Formats Output
RTF produces output for Microsoft Word 2002. Although other applications can read RTF files, the RTF output might not work successfully with the other applications.

The RTF destination enables you to view and edit the RTF output. ODS does not define the vertical measurement, which means that SAS does not determine the optimal place to position each item on the page. For example, page breaks are not always fixed because you do not want your RTF output tables to split at inappropriate places when you edit your text. Your tables remain intact on one page, or break where you specify.

However, Microsoft Word needs to know the widths of table columns; and Microsoft Word cannot adjust tables if they are too wide for the page. Therefore, ODS measures the width of the text and tables (horizontal measurement). All of the column widths can be set properly by SAS and the table can be divided into panels if it is too wide to fit on a single page.

In short, when producing RTF output for input to Microsoft Word, SAS determines the horizontal measurement and Microsoft Word controls the vertical measurement. Because Microsoft Word can determine how much room there is on the page, your tables are displayed consistently even after you modify your RTF file.

However, in SAS version 9.2, the ODS Measured tagset is introduced. This tagset enables users to specify how and where page breaks occur and when to place titles and footnotes into the body of a page. For information about using Measured RTF, see “ODS TAGSETS.RTF Statement ” on page 803.

Note: Complex tables that contain a large number of observations can reduce system efficiencies and take longer to process.

ODS RTF and Graphics
ODS RTF produces output in rich text format, which supports three formats for graphics that Microsoft Word can read.

<table>
<thead>
<tr>
<th>Format for Graphics</th>
<th>Corresponding SAS Graphics Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>emfbllips</td>
<td>EMF</td>
</tr>
<tr>
<td>pngblips</td>
<td>PNG</td>
</tr>
<tr>
<td>jpegblips</td>
<td>JPEG</td>
</tr>
</tbody>
</table>

When you do not specify a target device, the default target is EMF. You can also use the ACTIVEX, ACTXIMG, JAVAIMG graphics drivers to generate graphics in your RTF
documents. The ACTIVEX driver generates an ActiveX control. The ACTXIMG and
JAVAIMG drivers generate PNG files with the ACTIVEX Control or JAVA Applets
appropriately. For more information about graphics devices, see SAS/GRAPH:
Reference.

Note: When you specify the JAVA device in the ODS RTF statement, the JAVAIMG
driver is used.

Examples

Example 1: Creating a Table of Contents from Embedded Data

Features:
- ODS RTF statement action:
  - CLOSE
- ODS RTF statement options:
  - CONTENTS
  - NOTOC_DATA
  - TOC_DATA

Other features:
- #BYVAL parameter in titles
- NOBYLINE|BYLINE system option
- OPTIONS statement
- PROC FORMAT
- PROC PRINT
- PROC SORT
- PROC REPORT
- PROC TABULATE
- TITLE statement

Data set:
- Grain_Production

Format:
- $CNTRY.

The following example creates a table of contents page that contains embedded table of
contents data for some procedures but not for others. The insertion of the table of
contents data can be turned on and off in the middle of a program.

Program

```plaintext
proc sort data=Grain_Production;
    by year country type;
run;
ods html close;
ods rtf file='Grain.Rtf' contents toc_data;
options nobyline;
title 'Leading Grain-Producing Countries';
title2 'for #byval(year)';
proc report data=Grain_Production nowindows;
    by year;
column country type kilotons;
define country / group width=14 format=$cntry.;
```
Program Description

Sort the data set Grain_Production. PROC SORT sorts the data, first by values of the variable Year, then by values of the variable Country, and finally by values of the variable Type.

```
proc sort data=Grain_Production;
  by year country type;
run;
```

Close the HTML destination so that no HTML output is produced. The HTML destination is open by default. The ODS HTML statement closes the HTML destination to conserve resources.

```
ods html close;
```

Create RTF output and create a new body file for each page of output. The ODS RTF statement opens the RTF destination and creates RTF output. The CONTENTS option creates a table of contents page that contains a Table of Contents field. All of the contents information that is embedded in the document is placed in the table of contents. However, the table of contents information is not embedded by default into your RTF file. The default is NOTOC_DATA. The embedded TOC data is not shown until you specify the option TOC_DATA.

```
ods rtf file='Grain.Rtf' contents toc_data;
```

Replace the default BY line with a new value in the BY line. The NOBYLINE option suppresses the default BY line variable. The #BYVAL parameter specification inserts the current value of the BY variable Year into the title.

```
options nobyline;
  title 'Leading Grain-Producing Countries';
```
Produce a report. This PROC REPORT step produces a report on grain production. Each BY group produces a page of output, and ODS creates a new body file for each BY group. The NOWINDOWS option instructs PROC REPORT to run without the REPORT window and to send its output to the open output destinations.

```proc report data=Grain_Production nowindows;
  by year;
  column country type kilotons;
  define country / group width=14 format=$cntry.;
  define type   / group 'Type of Grain';
  define kilotons / format=comma12.;
  footnote 'Measurements are in metric tons.';
run;
```

Restore the default BY line and clear the second TITLE statement. The BYLINE option restores the default BY line. The TITLE2 statement clears the second TITLE statement.

```options byline;
title2;
```

Suppress the insertion of table of contents data into the RTF file. The NOTOC_DATA option instructs ODS not to insert the table of contents data into the RTF file. There will be no entry for the TABULATE procedure in the table of contents page.

```ods rtf notoc_data;
```

The TABLE statement in the PROC TABULATE step uses three dimensions. Year defines pages, Country and Type define the rows, and Kilotons defines the columns. Therefore, PROC TABULATE explicitly produces one page of output for 1995 and one page for 1996, based on the years specified in the Grain_Production data set. ODS also starts a new body file for each page.

```proc tabulate data=Grain_Production format=comma12.;
  class year country type;
  var kilotons;
  table year,
    country*type,
    kilotons*sum=' ' / box=_page_ misstext='No data';
  format country $cntry.;
  footnote 'Measurements are in metric tons.';
run;
```

Enable the insertion of table of contents data into the RTF file. The TOC_DATA option instructs ODS to insert the table of contents data into the RTF file. There will be an entry for the PRINT procedure in the table of contents page.

```ods rtf toc_data;
```

Print the Grain_Production data set.

```proc print data=Grain_Production;
run;
```
Close the RTF destination. The ODS RTF CLOSE statement closes the RTF destination and all the files that are associated with it. If you do not close the destination, you cannot view the files in a browser window.

```r
ods rtf close;
ods html;
```

RTF Output

By default the table of contents is collapsed on the table of contents page. To expand the table of contents, right-click under the title in Microsoft Word and select Update Field from the selection list.

The table of contents contains only entries for PROC REPORT and PROC PRINT. By default the table of contents data is not embedded in the RTF document. To embed the table of contents data, specify the TOC_DATA option, which results in an entry for PROC REPORT. If you specify the NOTOC_DATA option before the TABULATE procedure, ODS does not insert contents information into the RTF document, and no entry for PROC TABULATE appears in the table of contents. If you specify the
Example 2: Justifying Title and Footnotes When You Specify the BODYTITLE_AUX Option

Features:
- ODS RTF statement action:
  
  CLOSE
- ODS RTF statement options:
  
  BODYTITLE_AUX
  
  FILE=

Other features:
- OPTIONS statement
- PROC PRINT
- TITLE statement

When you want to place the titles and footnotes in the body of the RTF output, you usually specify the BODYTITLE option. However, to center your titles and footnotes or to justify them, you need to specify the BODYTITLE_AUX option. The preferred way to accomplish this functionality is to use the measured ODS TAGSETS.RTF statement. For more information, see “ODS TAGSETS.RTF Statement” on page 803.

Program

OPTIONS NODATE NOSTIMER LS=78 PS=60;
ods html close;
ods rtf file="bodytitle_aux.rtf" bodytitle_aux;
proc print data=sashelp.class;
run;
title j=1 "left" j=c "center" j=r "right";
title2 j=1 "left";
title3 j=c "center";
title4 j=r "right";
footnote j=1 "left" j=c "center" j=r "right";
run;
ods rtf close;
ods html;

Program Description
The following example shows how to left-justify, right-justify, and center titles and footnotes in the body of the output.

Specify the layout of the output. Instruct ODS not to print the date or time on the page and not to write any SAS statistics to the SAS log. Set the page size to 60 and the line size to 78.

```
OPTIONS NODATE NOSTIMER LS=78 PS=60;
```

Close the HTML destination so that no HTML output is produced The HTML destination is open by default. The ODS HTML statement closes the HTML destination to conserve resources.

```
ods html close;
```

Create RTF output. The ODS RTF statement opens the RTF destination and creates RTF output. The BODYTITLE_AUX option tells SAS to place the titles and footnotes in the body of the output. In addition, this option places the titles and footnotes into cells.

```
ods rtf file="bodytitle_aux.rtf" bodytitle_aux;
```

Print the Sashelp.Class data set.
```
proc print data=sashelp.class;
run;
```

Add titles and footnotes to the output. Because you have specified the BODYTITLE_AUX option, ODS adds the titles and footnotes to the body of the output and places the text into cells. The J= style specifies the position of the title and footnote text on the page: left, center, or right.

```
title j=l "left" j=c "center" j=r "right";
title2 j=l "left";
title3 j=c "center";
title4 j=r "right";
footnote j=l "left" j=c "center" j=r "right";
run;
```

Close the RTF destination. The ODS RTF CLOSE statement closes the RTF destination and all the files that are associated with it. If you do not close the destination, you cannot view the files in a browser window.

```
ods rtf close;
ods html;
```

The following output shows how ODS places the titles and footnotes into the body of the output when you specify the BODYTITLE_AUX option. The text of the titles and
footnotes are then placed into cells and tables. The JUSTIFY style element is then used to center, right-justify, or left-justify the title and footnote text.

Example 3: RTF Interaction with the ORIENTATION= System Option

Features:
- ODS RTF statement action:
  
  CLOSE
  
- ODS RTF statement option:
  
  FILE=

Other features:
- OPTIONS statement: ORIENTATION option
- PROC PRINT
- TITLE statement

When you want to change the page orientation for RTF, specify the ORIENTATION= system option. To activate or trigger this change of the page orientation, the ODS RTF statement needs to follow the ORIENTATION= option. The following example provides example code for specifying a page orientation change within an RTF file.

Program

```plaintext
OPTIONS NODATE NOSTIMER LS=78 PS=60;
```
Program Description

Specify the layout of the output. Instruct ODS not to print the date or time on the page and not to write any SAS statistics to the SAS log. Set the page size to 60 and the line size to 78.

```sas
OPTIONS NODATE NOSTIMER LS=78 PS=60;
```

Close the HTML destination so that no HTML output is produced. The HTML destination is open by default. The ODS HTML statement closes the HTML destination to conserve resources.

```sas
ods html close;
```

Add titles and footnotes to the output. Add a title for the overall file output and then titles that describe the changing orientation.

```sas
title 'Page Orientation';
title2 'Default';
```

Create RTF output. The ODS RTF statement opens the RTF destination and creates RTF output. In this case, the statement also triggers the change in the page orientation from the default.

```sas
ods rtf file="ChgOrientation.rtf";
```

Print the Sashelp.Class data set with only one observation. The page orientation is the default orientation, which is portrait.

```sas
proc print data=sashelp.class (obs=1);
run;
```

Add a title to change the page orientation in the output file. Add a title to change the page orientation to landscape.

```sas
title 'Page Orientation';
title2 'Landscape';
```

Specify the system option that changes the page orientation.
options orientation=landscape;

Trigger the page orientation change. This RTF statement triggers the change of the page orientation from portrait to landscape.

ods rtf;

Print the Sashelp.Class data set with only one observation.

    proc print data=sashelp.class (obs=1);
    run;

Close the RTF destination. The ODS RTF CLOSE statement closes the RTF destination and all the files that are associated with it. If you do not close the destination, you cannot view the files in a browser window.

    ods rtf close;
    ods html;

RTF Output

The following shows the RTF output for the first page. The orientation is portrait, which is the default.
The following shows the RTF output for the second page. The orientation was changed to landscape.

![Page Orientation Landscape](image)

**ODS SELECT Statement**

Specifies output objects for ODS destinations.

**Valid in:** Anywhere  
**Category:** ODS: Output Control  
**Tip:** You can maintain a selection list for one destination and an exclusion list for another. However, it is easier to understand the results if you maintain the same types of lists for all of the destinations to which you route output.

**See:** “ODS EXCLUDE Statement” on page 321  
**Example:** “Table Header and Footer Border Formatting” in *SAS Output Delivery System: Procedures Guide*

---

**Syntax**

```
ODS <ODS-destination> SELECT selection(s) | ALL | NONE;
```

**Required Arguments**

`selection(s)`  
specifies output objects to add to a selection list. ODS sends the items in the selection list to all active ODS destinations. By default, ODS automatically modifies selection lists when a DATA step that uses ODS or a procedure step ends. For information about modifying these lists, see “Selection and Exclusion Lists” on page 39. For information about ending DATA and procedure steps, see the section on DATA Step Processing in *SAS Language Reference: Concepts*.

Each `selection` has the following form:
**output-object** <-(PERSIST)>

**output-object**

specifies the output object to select.

To specify an output object, you need to know which output objects your SAS program produces. The ODS TRACE statement writes to the SAS log a trace record that includes the path, the label, and other information about each output object that your SAS program produces. You can specify an output object as one of the following:

- a full path. For example, the following is the full path of the output object:
  
  `Univariate.City.Pop_90.TestsForLocation`

- a partial path. A partial path consists of any part of the full path that begins immediately after a period (.) and continues to the end of the full path. For example, suppose the full path is the following:
  
  `Univariate.City.Pop_90.TestsForLocation`

  Then the partial paths are as follows:
  
  `City.Pop_90.TestsForLocation`
  `TestsForLocation`

- a label that is enclosed in quotation marks.

  For example:

  "The UNIVARIATE Procedure"

- a label path. For example, the label path for the output object is as follows:

  "The UNIVARIATE Procedure"."City.Pop_90"."Tests For Location"

  **Note:** The trace record shows the label path only if you specify the LABEL option in the ODS TRACE statement.

- a partial label path. A partial label path consists of any part of the label that begins immediately after a period (.) and continues to the end of the label. For example, suppose the label path is the following:

  "The UNIVARIATE Procedure"."City.Pop_90"."Tests For Location"

  Then the partial label paths are as follows:

  "City.Pop_90"."Tests For Location"
  "Tests For Location"

- a mixture of labels and paths.

- any of the partial path specifications, followed by a pound sign (#) and a number. For example, `TestsForLocation#3` refers to the third output object that is named `TestsForLocation`.

  **See**  “ODS TRACE Statement” on page 854

**(PERSIST)**

keeps the **output-object** that precedes the PERSIST option in the selection list, even if the DATA or procedure step ends, until you explicitly modify the list with one of the following:

- any ODS EXCLUDE statement

- ODS SELECT NONE
- ODS SELECT ALL
- an ODS SELECT statement that applies to the same output object but does not specify PERSIST

Requirements: You must enclose PERSIST in parentheses.

ALL
specifies that ODS send all of the output objects to the open destination.

Alias: ODS SELECT DEFAULT

Interaction: If you specify ALL without specifying a destination, ODS sets the overall list to SELECT ALL and sets all other lists to their defaults.

NONE
specifies that ODS does not send any output objects to the open destination.

Interaction: If you specify NONE and you do not specify a destination, ODS sets the overall list to SELECT NONE and sets all other lists to their defaults.

Tips: Using the NONE action is different from closing a destination. The output destination is still open, but ODS restricts the output that it sends to the destination.

To temporarily suspend a destination, use ODS SELECT NONE. Use ODS SELECT ALL when you want to resume sending output to the suspended destination.

Optional Arguments

(NOWARN)
suppresses the warning that an output object was requested but not created.

Requirement: The NOWARN option must be enclosed in parentheses.

Interaction: The NOWARN option cannot be used with the ALL option or the NONE option.

Example: The ODS SELECT statement in the following example specifies that no warning is created if the output object Summary is requested but not created.

```ods select summary (nowarn);```
```proc contents data=sashelp.class; run;```
Tip To set the selection list for the Output destination to something other than the default, see the “ODS OUTPUT Statement ” on page 534.

See “Understanding ODS Destinations” on page 33 for a discussion of ODS destinations.

WHERE=where-expression
selects output objects that meet a particular condition. For example, the following statement selects only output objects with the word "Histogram" in their name:

```sas
ods select where=_name_ ? 'Histogram';
```

where-expression is an arithmetic or logical expression that consists of a sequence of operators and operands. where-expression has this form:

```
(subsetting-variable <comparison-operator where-expression-n>)
```

subsetting-variable
Subsetting variables are a special type of WHERE expression operand used by SAS to help you find common values in items. For example, this ODS SELECT statement selects only output objects with the path `City_Pop_90.TestsForLocation`:

```sas
ods select / where=_path_ = 'City_Pop_90.TestsForLocation';
```

subsetting-variable is one of the following:

- _LABEL_
  - is the label of the output object.

- _LABELPATH_
  - is the label path of the output object.

- _NAME_
  - is the name of the output object.

- _PATH_
  - is the full or partial path of the output object.

operator compares a variable with a value or with another variable. operator can be AND, OR NOT, OR, AND NOT, or a comparison operator.

The following table lists some comparison operators:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Mnemonic Equivalent</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>EQ</td>
<td>Equal to</td>
</tr>
<tr>
<td>^= or -= or == or &lt;&gt;</td>
<td>NE</td>
<td>Not equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>GT</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;</td>
<td>LT</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>GE</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt;=</td>
<td>LE</td>
<td>Less than or equal to</td>
</tr>
</tbody>
</table>
Details

For each ODS destination, ODS maintains either a selection list or an exclusion list of output objects. You can use the default output objects selected or excluded for each destination or you can specify which output object you want to produce by selecting or excluding them from a list.

A selection list is a list of output objects that are sent to an ODS destination. An exclusion list is a list of output objects that are excluded from an ODS destination. ODS also maintains an overall selection or exclusion list of output objects. By checking the destination-specific lists and the overall list, ODS determines what output objects to produce. These lists can be modified by using the ODS SELECT statement and the ODS EXCLUDE statement.

**T I P** You can maintain a selection list for one destination and an exclusion list for another. However, the results are less complicated if you maintain the same types of lists for all the destinations to which you route output.

You can view the contents of the exclusion and selection lists by using the ODS SHOW statement. The current selection list is written to the SAS log.

EXCLUDE ALL is the default setting for the ODS OUTPUT destination. SELECT ALL is the default setting for all other destinations. To change the default selection and exclusion lists, use the ODS SELECT or ODS EXCLUDE statements or use the exclude and select actions that are available for some of the ODS statements. However, to set the exclusion list for the OUTPUT destination to something other than the default, use the “ODS OUTPUT Statement” on page 534. For a list of ODS output destinations and explanations of each, see “Understanding ODS Destinations” on page 33.

In order to view output objects that are selected or excluded from your program, use the ODS TRACE statement. The ODS TRACE statement prints the output objects that are selected and excluded and puts the information in a trace record that is written to the SAS log. The trace provides the path, the label, and other information about output objects that are selected and excluded. For complete documentation about viewing and selecting output objects, see the “ODS SELECT Statement” on page 758, the “ODS EXCLUDE Statement” on page 321, and the “ODS TRACE Statement” on page 854.

Examples

**Example 1: Using a Selection List with Multiple Procedure Steps**

**Features:**
- ODS SELECT statement:
  - with label
  - with name
  - with and without PERSIST
  - ALL

- ODS SHOW statement

- ODS HTML statement options:
  - BODY=
  - CONTENTS=
Other features:
PROC GLM
PROC PRINT
PROC PLOT

Data set:
Iron

Details
This example runs the same procedures multiple times to illustrate how ODS maintains and modifies a selection list. The ODS SHOW statement writes the overall selection list to the SAS log. The example does not alter selection lists for individual destinations. The contents file that is generated by the ODS HTML statement shows which output objects are routed to both the HTML and the LISTING destinations.

This example creates and prints data sets from the parameter estimates that PROC GLM generates. This procedure is part of SAS/STAT software.

Program
ods html body='odspersist-body.htm'
   frame='odspersist-frame.htm'
   contents='odspersist-contents.htm'
   pages='odspersist-page.htm';
ods show;
ods select ParameterEstimates
   "Type III Model ANOVA";
ods show;
proc glm data=iron;
   model loss=fe;
   title 'Parameter Estimates and Type III Model ANOVA';
run;
ods show;
quit;
ods show;
proc glm data=iron;
   model loss=fe;
   title 'All Output Objects Selected';
run;
quit;
ods select OverallANOVA(persist) "Fit Statistics";
proc glm data=iron;
   model loss=fe;
   title 'OverallANOVA and Fitness Statistics';
run;
quit;
ods show;
proc glm data=iron;
   model loss=fe;
   title 'Overall ANOVA';
   title2 'Part of the Selection List Persists';
run;
quit;
proc print data=iron;
   title 'The IRON Data Set';
run;
ods select all;
proc plot data=iron;
   plot fe*loss='*' / vpos=25 ;
   label fe='Iron Content'
   loss='Weight Loss';
   title 'Plot of Iron Versus Loss';
run;
quit;
ods html close;

Program Description

Create HTML output. The ODS HTML creates the body, contents, frame, and page files. The output from the procedures is sent to the file odspersist-body.htm. The FRAME=, CONTENTS=, and PAGE= options create the files OdsPersist-Frame.htm, OdsPersist-Contents.htm, and OdsPersist-Page.htm, respectively. These files, together with the file OdsPersist-Body.htm, create a frame that includes a table of contents and a table of pages that link to the contents of the body file.

ods html body='odspersist-body.htm'
   frame='odspersist-frame.htm'
   contents='odspersist-contents.htm'
   pages='odspersist-page.htm';

Write the overall selection list to the SAS log. The ODS SHOW statement writes to the SAS log the overall list, which is set to SELECT ALL by default. See the “SAS Log” on page 767.

ods show;

Specify the output objects that will be sent to the open destinations. The ODS SELECT statement determines which output objects ODS sends to the LISTING and HTML destinations. In this case, ODS sends all output objects that are named ParameterEstimates and all output objects that are labeled "Type III Model ANOVA" to the two destinations.

ods select ParameterEstimates
   "Type III Model ANOVA";

Write the modified overall selection list to the SAS log. The ODS SHOW statement writes to the SAS log the overall selection list, which now contains the two items that were specified in the ODS SELECT statement. See the “SAS Log” on page 767.

ods show;
Create the output objects and send the selected output objects to the open destinations. As PROC GLM sends each output object to the Output Delivery System, ODS sends the two output objects from PROC GLM that match the items in the selection list to the open destinations. See 1. in the table of contents in “HTML Output” on page 768. Note that it is the label of an output object, not its name, that appears in the table of contents. The label for ParameterEstimates is "Solution".

```sas
proc glm data=iron;
  model loss=fe;
  title 'Parameter Estimates and Type III Model ANOVA';
run;
```

Write the overall selection list to the SAS log. PROC GLM supports run-group processing. Therefore, the RUN statement does not end the procedure, and ODS does not automatically modify the selection list. See the “SAS Log” on page 767.

```sas
ods show;
```

End the GLM procedure. The QUIT statement ends the procedure. ODS removes all objects that are not specified with PERSIST from the selection list. Because this action removes all objects from the list, ODS sets the list to its default, SELECT ALL.

```sas
quit;
```

Write the current selection list to the SAS log. The ODS SHOW statement writes the current selection list to the SAS log. See the “SAS Log” on page 767.

```sas
ods show;
```

Create the output objects, send the selected output objects to the open destinations, and end the procedure. As PROC GLM sends each output object to the Output Delivery System, ODS sends all the output objects to the HTML and LISTING destinations. See 2. in the table of contents in “HTML Output” on page 768. The QUIT statement ends the procedure. Because the list uses the argument ALL, ODS does not automatically modify it when the PROC step ends.

```sas
proc glm data=iron;
  model loss=fe;
  title 'All Output Objects Selected';
run;
quit;
```

Modify the overall selection lists. This ODS SELECT statement modifies the overall selection list. It sends all output objects that are named OverallANOVA, and all output objects that are labeled Fit Statistics, to both the HTML and LISTING destinations. The PERSIST option specifies that OverallANOVA should remain in the selection list when ODS automatically modifies it.

```sas
ods select OverallANOVA(persist) "Fit Statistics";
```

Create the output objects and send the selected output objects to the open destinations. As PROC GLM sends each output object to the Output Delivery System, ODS sends the two output objects from PROC GLM that match the items in the selection list to the HTML and LISTING destinations. See 3. in the table of contents in “HTML Output” on page 768.

```sas
proc glm data=iron;
  model loss=fe;
```
title 'OverallANOVA and Fitness Statistics';
run;

End the GLM procedure and automatically modify the selection list. When the QUIT statement ends the procedure, ODS automatically modifies the selection list. Because OverallANOVA was specified with the PERSIST option, it remains in the selection list. Because Fitness Statistics was not specified with the PERSIST option, ODS removes it from the selection list.
quit;

Write the current selection list to the SAS log. The ODS SHOW statement writes the current selection list to the SAS log. See the “SAS Log” on page 767.
ods show;

Create the output objects and send the selected output objects to the open destinations. As PROC GLM sends each output object to the Output Delivery System, ODS sends only the output object that is named OverallANOVA to the HTML and LISTING destinations. See 4. in the table of contents in “HTML Output” on page 768.
proc glm data=iron;
  model loss=fe;
  title 'OverallANOVA';
  title2 'Part of the Selection List Persists';
run;

End the GLM procedure and automatically modify the selection list. When the QUIT statement ends the procedure, ODS automatically modifies the selection list. Because OverallANOVA was specified with the PERSIST option, it remains in the selection list.
quit;

PROC PRINT does not produce any output that is named OverallANOVA. Therefore, no PROC PRINT output is sent to the ODS destinations.
proc print data=iron;
  title 'The IRON Data Set';
run;

Reset all selection lists. This ODS SELECT statement resets all selection lists to their defaults.
ods select all;

Create the plots. As PROC PLOT creates and sends each output object to the Output Delivery System, ODS sends each one to the HTML and LISTING destinations because their lists and the overall list are set to SELECT ALL (the default).
proc plot data=iron;
  plot fe*loss='*' / vpos=25 ;
  label fe='Iron Content'
  loss='Weight Loss';
  title 'Plot of Iron Versus Loss';
run;

End the PLOT procedure. The QUIT statement ends the PLOT procedure. Because the list uses the argument ALL, ODS does not automatically modify the list when the PROC step ends.
quit;

Close the HTML destination. This ODS HTML statement closes the HTML destination and all the files that are associated with it.

ods html close;

SAS Log

Output 6.90 The ODS SHOW Statement Writes the Current Selection List to the SAS Log.
The contents file shows the output objects from each procedure that ODS sent to the open ODS destinations. You can see that no output was written to the HTML destination for PROC PRINT (because PROC PRINT did not produce anything whose name matched the name in the selection list). You can also see that the PROC PLOT output was written to the HTML destination after the ODS SELECT ALL statement was executed.

### Example 2: Conditionally Selecting Output Objects

**Features:**
- ODS SELECT statement option: `WHERE=`
- ODS TRACE statement options:
  - `LABEL`
  - `EXCLUDED`
ODS HTML statement

Other features:
PROC UNIVARIATE

Program

data BPressure;
  length PatientID $2;
  input PatientID $ Systolic Diastolic @@;
  datalines;
  CK 120 50  SS 96  60 FR 100 70
  CP 120 75  BL 140 90 ES 120 70
  CP 165 110 JI 110 40 MC 119 66
  FC 125 76 RW 133 60 KD 108 54
  DS 110 50 JW 130 80 BH 120 65
  JW 134 80 SB 118 76 NS 122 78
  GS 122 70 AB 122 78 EC 112 62
  HH 122 82
;  
run;

title 'Systolic and Diastolic Blood Pressure';
ods trace on / label excluded;
ods select where=( _path_ ? "Diastolic" and _name_='Moments') ;
proc univariate data=BPressure;
  var Systolic Diastolic;
  run;

Program Description

Create the BPressure data set.

data BPressure;
  length PatientID $2;
  input PatientID $ Systolic Diastolic @@;
  datalines;
  CK 120 50  SS 96  60 FR 100 70
  CP 120 75  BL 140 90 ES 120 70
  CP 165 110 JI 110 40 MC 119 66
  FC 125 76 RW 133 60 KD 108 54
  DS 110 50 JW 130 80 BH 120 65
  JW 134 80 SB 118 76 NS 122 78
  GS 122 70 AB 122 78 EC 112 62
  HH 122 82
;  
run;

title 'Systolic and Diastolic Blood Pressure';

Add a title.

title 'Systolic and Diastolic Blood Pressure';

Specify that SAS write the trace record to the SAS log. This ODS TRACE statement writes the trace record to the SAS log. The LABEL option includes label paths in the
trace record. The EXCLUDED option includes information about output objects that SAS excludes from the output destination.

```sas
ods trace on / label excluded;
```

**Select output objects.** The ODS SELECT statement with the WHERE = option specified selects output objects that are named 'Moments' and that have 'Diastolic' in the pathname.

```sas
ods select where=(path_ ? "Diastolic" and _name_='Moments') ;
```

Create the output objects and send the selected output objects to the open destination. As PROC UNIVARIATE sends each output object to the Output Delivery System, ODS sends the output object from PROC UNIVARIATE that matches the items in the selection list to the open destination.

```sas
proc univariate data=BPressure;
  var Systolic Diastolic;
run;
```

**SAS Log: Trace Record**

**Output 6.92** SAS Log Including Trace Record
ODS SHOW Statement

Writes the specified selection or exclusion list to the SAS log.

Valid in: Anywhere
Category: ODS: Output Control

Syntax

ODS <ODS-destination> SHOW;

Optional Argument

ODS-destination

specifies which ODS destination's selection or exclusion list to write to the SAS log. ODS-destination must be a valid ODS destination. For information about ODS destinations, see “Understanding ODS Destinations” on page 33. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

See Also

Statements
• “ODS EXCLUDE Statement” on page 321
• “ODS SHOW Statement” on page 771
• “ODS TRACE Statement” on page 854
If you omit ODS-destination, ODS SHOW writes the overall selection or exclusion list.

See Also

Statements

• “ODS EXCLUDE Statement” on page 321
• “ODS SELECT Statement” on page 758
• “ODS TRACE Statement” on page 854

ODS Tagset Statement

Opens, manages, or closes the specified tagset destination.

Valid in: Anywhere
Category: ODS: Third-Party Formatted
Interaction: Using the ODS Tagset statement in an ODS markup family statement that refers to an open ODS markup destination forces ODS to close the destination and all associated files, and then open a new instance of the destination. For more information, see “Opening and Closing the MARKUP Destination” on page 517.

See: For additional information about specifying tagsets, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in SAS Output Delivery System: Procedures Guide or “ODS MARKUP Statement” on page 488.

Syntax

ODS directory.tagset-name file-specification <option(s)>;
ODS directory.tagset-name file-specification action;

Summary of Optional Arguments

(ID= identifier)
Open multiple instances of the same destination at the same time

ANCHOR= 'anchor-name'
Specify a unique base name for the anchor tag that identifies each output object in the current body file

ARCHIVE='string'
Specify which applet to use to view ODS HTML output

ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)
Specify attributes to write between the tags that generate dynamic graphics output

BASE= 'base-text'
Specify text to use as the first part of all links and references that ODS creates in output files

BODY= 'file-specification' (suboption(s))
Open a markup family destination and specify the file that contains the primary output that is created by the ODS statement
CHARSET= character-set
   Specify the character set to be generated in the META declaration for the HTML output

CLOSE
   Close the destination and the file that is associated with it

CODE='file-specification' <(suboption(s))>
   Open the HTML destination and specify the file that contains relevant style information

CODEBASE='string'
   Create a file path that can be used by the GOPTIONS devices

CONTENTS='file-specification' <(suboption(s))>
   Open the HTML destination and specify the file that contains a table of contents for the output

CSSSTYLE='file-specification'<(media-type-1<…media-type-10>)>
   Specify a cascading style sheet to apply to your output

DOM="external-file">
   Specify that the ODS document object model is written to the SAS log or to an external file.

ENCODING= local-character-set-encoding
   Override the encoding for input or output processing (transcodes) of external files

EVENT=event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )
   Specify an event and the value for event variables that is associated with the event

EXCLUDE exclusion(s) | ALL | NONE
   Exclude output objects from the destination

FRAME='file-specification' <(suboption(s))>
   Specify the file that integrates the table of contents, the page contents, and the body file

GFOOTNOTE | NOGFOOTNOTE
   Control the location where footnotes are printed in the graphics output

GPATH='aggregate-file-storage-specification'|fileref|libref.catalog (URL=
   'Uniform-Resource-Locator'|NONE)
   Specify the location for all graphics output that is generated while the destination is open

GTITLE | NOGTITLE
   Control the location where titles are printed in the graphics output

HEADTEXT='markup-document-head'
   Specify HTML tags to place between the <HEAD> and </HEAD> tags in all of the output files.

METATEXT='metatext-for-document-head'
   Specify HTML code to use as the <META> tag between the <HEAD> and </HEAD> tags in all of the HTML output files.

NEWFILE= starting-point
   Create a new body file at the specified starting point

OPTIONS ( DOC= ) | sub-option(s)
   Specify ODS tagset-specific suboptions and a named value.

PAGE='file-specification' <(suboption(s))>
   Open the HTML destination and specify the file that contains a description of each page of the body file and contains links to the body file
PARAMETERS= (parameter-pair-1 ... parameter-pair-n)
  Write the specified parameters between the tags that generate dynamic
graphics output

PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL=
  'Uniform-Resource-Locator' | NONE)
  Specify the location of an aggregate storage location or a SAS catalog for all
markup files

RECORD_SEPARATOR= 'alternative-separator' | NONE
  Specify an alternative character or string to separate lines in the output files

SELECT selection(s) | ALL | NONE
  Select output objects for the destination

SHOW
  Write to the SAS log the current selection or exclusion list for the destination

STYLE= style-template
  Specify a style template to use in writing output files

STYLESHEET= 'file-specification' <(suboption(s))>
  Open the HTML destination and place style information for output into an
external file, or read style sheet information from an existing file

TEXT= text-string
  Insert text into your document

TRANTAB= 'translation-table'
  Specify a translation table to use when transcoding a file for output

Actions
The following actions are available for the ODS Tagset statement.

CLOSE
  closes the destination and any files that are associated with it.
  Tip    When an ODS destination is closed, ODS does not send output to that
destination. Closing an unneeded destination conserves system resources.

EXCLUDE exclusion(s) | ALL | NONE
  excludes one or more output objects from the destination.
  Default    NONE
  Restriction   A destination must be open for this action to take effect.
  See "ODS EXCLUDE Statement" on page 321

SELECT selection(s) | ALL | NONE
  selects output objects for the specified destination.
  Default    ALL
  Restriction   A destination must be open for this action to take effect.
  See "ODS SELECT Statement" on page 758

SHOW
  writes the current selection list or exclusion list for the destination to the SAS log.
  Restriction   The destination must be open for this action to take effect.
Tip
If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

See “ODS SHOW Statement” on page 771

Required Arguments
The following arguments are available for the ODS Tagset statement. For additional tagsets that are available for the ODS Tagset statement, see “Diagnostic Tagsets” on page 797.

directory
specifies the directory where the specified tagset is stored. directory can be a directory supplied by SAS, a user-defined entry, or a libref. By default, the tagsets that SAS supplies are located in the directory TAGSETS, which is within the item store Sashelp.Tmplmst.
tagset-name
specifies the name of the tagset. tagset-name can be one of the following:

CHTML
produces compact, minimal HTML output that does not use style information. It does produce a hierarchical table of contents.

See “ODS CHTML Statement” on page 170

CORE
contains a table of Unicode values and mnemonics. For a detailed description of using this tagset, see “Using Unicode Symbols” on page 272.

CSV
produces tabular output that contains columns of data values that are separated by commas.

Interaction The TEXT= option has no affect in the CSV file output.


CSVALL
produces output containing columns of data values that are separated by commas, and produces tabular output with titles, notes, and BY lines.

Interaction The TEXT= option has no affect in the CSV file output.

See “ODS CSVALL Statement” on page 195

Example “Example 3: Creating Multiple Markup Output” on page 522

CSVBYLINE
produces output with comma-separated values and columns of data that are separated by commas.

Interaction The TEXT= option has no affect in the CSV file output.

DEFAULT
produces XML output.
ExcelXP  
produces Microsoft spreadsheetML XML. This tagset is used to import data into Excel. Execute the following code to get detailed information about this tagset:

```plaintext
ods tagsets.excelxp file='test.xml' options(doc='help');
```

HTML4  
produces HTML 4.0 embedded style sheets.  

See "ODS HTML Statement " on page 379

HTMLPANEL  
creates panels for BY grouped graphs. It also has controls for semi-automatic and manually controlled paneling. This tagset makes it easy to put graphs and tables side-by-side on a page. Also included are controls for titles, footnotes, and BY lines.  

To get detailed help on this tagset, any of the following three lines of code can be executed:

```plaintext
ods tagsets.htmlpanel file="gbypanel.html" options(doc='help');
ods tagsets.htmlpanel options(doc='quick');
ods tagsets.htmlpanel options(doc='settings');
```

MSOFFICE2K  
produces HTML code for output generated by ODS for Microsoft Office products.

PHTML  
produces simple HTML output that uses twelve style elements and no class attributes for the presentation. Class attributes are used only for the justification.  

See “ODS PHTML Statement” on page 663

RTF  
produces measured RTF. This tagset allows the user to specify how and where page breaks occur and when to place titles and footnotes into the body of a page. The RTF tagset enables SAS to place titles and footnotes into the body of the document so that it is outside of the control of Microsoft Word. Therefore, SAS becomes responsible for the implicit page breaks.  

For details about how to use the RTF tagset, see “ODS TAGSETS.RTF Statement ” on page 803.

**user-defined-tagset**  
specifies the tagset that you created using PROC TEMPLATE.  


---

**Optional Arguments**

The following options are available for the ODS Tagset statement, which is part of the markup family of statements.

**ANCHOR= 'anchor-name'**  
specifies a unique base name for the anchor tag that identifies each output object in the current body file.  

Each output object has an anchor tag for the contents, page, and frame files to reference. The links and references are automatically created by ODS. The links and
references point to the name of an anchor. Therefore, each anchor name in a file must be unique.

**anchor-name**

is the base name for the anchor tag that identifies each output object in the current body file.

ODS creates unique anchor names by incrementing the name that you specify. For example, if you specify ANCHOR= 'TABULATE', then ODS names the first anchor **tabulate**. The second anchor is named **tabulate1**, the third is named **tabulate2**, and so on.

**Restrictions**

Each anchor name in a file must be unique.

Only alphanumeric values, the special characters "$ - . + ! * ' () , " and reserved characters used for their reserved purposes can be used unencoded within a URL.

**Requirement**

You must enclose *anchor-name* in quotation marks.

**Interaction**

If you open a file to append to it, be sure to specify a new anchor name to prevent writing the same anchors to the file again. ODS does not recognize anchors that are already in a file when it opens the file.

**Tips**

You can change anchor names as often as you want by specifying the ANCHOR= option in a markup family statement anywhere in your program. After you have specified an anchor name, it remains in effect until you specify a new one.

Specifying new anchor names at various points in your program is useful when you want other web pages to link to specific parts of your markup language output. Because you can control where the anchor name changes, you know in advance what the anchor name is at those points.

An *anchor-name* must begin with a letter ([A-Za-z]) and can be followed by any number of letters, digits ([0-9]), hyphens (-), underscores (_), colons (:), and periods (.).

**ARCHIVE='string'**

specifies which applet to use to view the ODS HTML output. The ARCHIVE= option is valid only for the GOPTIONS Java device.

The string must be one that the browser can interpret. For example, if the archive file is local to the computer that you are running SAS on, you can use the FILE protocol to identify the file. If you want to point to an archive file that is on a web server, use the HTTP protocol.

**Default**

If you do not specify ARCHIVE= and you are using the JAVA device driver, ODS uses the value of the SAS system option APPLETOC=. There is no default if you are using the ACTIVEX device driver.

**Requirements**

You must enclose *string* in quotation marks.

The ARCHIVE attribute is a feature of Java 1.1. Therefore, if you are using the Java device driver, your browser must support this...
Both Internet Explorer 4.01 and Netscape 4.05 support Java 1.1.

Interaction

Use ARCHIVE= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA or DEVICE=ACTIVEX option in the GOPTIONS statement.

Tips

Typically, this option should not be used, because the SAS server automatically determines the correct SAS/GRAPH applets to view the ODS HTML output. However, if you have renamed the JAR files, or have other applets with which to view the ODS HTML output, this option enables you to access these applets.

Use the CODEBASE= option to specify the file path. It is recommended that you do not put a file path in your ARCHIVE= option.

The value of APPLETOC= points to the location of the Java archive files that are shipped with the SAS system. To find out what the value of this option is, you can either look in the Options window in the Files folder under Environment Control, or you can submit the following procedure step:

```sas
proc options option=appletloc;
run;
```

ATTRIBUTES= (attribute-pair-1 ... attribute-pair-n)

writes the specified attributes between the tags that generate dynamic graphics output.

attribute-pair

specifies the name and value of each attribute. attribute-pair has the following form:

'attribute-name'='attribute-value'

attribute-name

is the name of the attribute.

attribute-value

is the value of the attribute.

Requirement

You must enclose attribute-name and attribute-value in quotation marks.

Interaction

Use the ATTRIBUTES= option in conjunction with SAS/GRAPH procedures and with the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

See

SAS/GRAPH: Reference for valid attributes for the Graph Applet, the Map Applet, the Contour Applet, and the MetaView Applet.

BASE= 'base-text'

specifies the text to use as the first part of all links and references that ODS creates in the output files.

base-text

is the text that ODS uses as the first part of all links and references that ODS creates in the file.
Consider this specification:

```
BASE = 'http://www.your-company.com/local-url/'
```

In this case, ODS creates links that begin with the string `http://www.your-company.com/local-url/`. The appropriate anchor-name completes the link.

**Requirement** You must enclose base-text in quotation marks.

**BODY = 'file-specification' (suboption(s))**

opens a markup family destination and specifies the file that contains the primary output that is created by the ODS statement. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

**file-specification**

specifies the file, fileref, or SAS catalog to write to.

**file-specification** is one of the following:

- **external-file**
  is the name of an external output file.
  
  **Requirement** You must enclose external-file in quotation marks.

- **fileref**
  is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.
  
  **Restriction** The BODY=fileref option cannot be used in conjunction with the NEWFILE= option.

  **See** For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

- **entry.markup**
  specifies an entry in a SAS catalog to write to.
  
  **Interaction** If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

**(suboption(s))**

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

- **(DYNAMIC)**
  enables you to send output directly to a web server instead of writing it to a file.
  
  **See** For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 798.

- **(NO_BOTTOM_MATTER)**
  specifies that no ending markup language source code be added to the output file.
See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 798.

(NO_TOP_MATTER)
See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 799.

(TITLE=’title-text’)
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.
See For complete documentation about the TITLE= suboption, see “(TITLE=’title-text’)” on page 799.

(URL= ’Uniform-Resource-Locator’ )
See For complete documentation about the URL= suboption, see “(URL=’Uniform-Resource-Locator’ )” on page 800.

Alias FILE=

Interaction Using the BODY= option in an ODS markup family statement that refers to an open ODS markup destination forces ODS to close the destination and all associated files. ODS then opens a new instance of the destination. For more information see “Opening and Closing the MARKUP Destination “ on page 517.

Note For some values of TAGSET=, this output is an HTML file. For other TAGSET= values, the output is an XML file, and so on.

CHARSET= character-set
specifies the character set to be generated in the META declaration for the HTML output.
See For information about the CHARSET= option, see “CHARSET= Option” in SAS National Language Support (NLS): Reference Guide.

CODE= ’file-specification’ <(suboption(s))>
opens a markup family destination and specifies the file that contains accompanying programming code, such as JavaScript or XSL (Extensible Stylesheet Language). These files remain open until you do one of the following:

• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
• open the same destination with a second markup family statement. This closes the first file and opens the second file.

file-specification
specifies the file, fileref, or SAS catalog to write to.
file-specification is one of the following:

external-file
is the name of an external output file.

Requirement You must enclose external-file in quotation marks.
fileref

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.


entry.markup

specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

suboption(s)

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

(DYNAMIC)

enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 798.

(URL= 'Uniform-Resource-Locator' )

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator' )” on page 800.

CODEBASE='string'

specifies the location of the executable Java applet or the ActiveX control file. string is specified as a pathname or as a URL. The CODEBASE file path option has two definitions, depending on the GOPTIONS device used.

When you generate web presentations with the JAVA and ActiveX device drivers, SAS generates HTML pages that automatically look for the JAVA archive files or the ActiveX control file in the default installation location.

For the ActiveX device:

If you use the ActiveX device driver with ODS to generate output containing an ActiveX control, then specify the CODEBASE= option in the ODS statement. The value of the CODEBASE= option should include the location and the version of the EXE file.

Tip You do not need to specify the CODEBASE= option with the DEVICE=ACTIVEX option unless the users that view your output do not have the ActiveX control installed on their machine. When users that do not have the ActiveX control installed view your output, they are prompted to download the control.

See SAS/GRAPH: Reference for information about specifying the location of control and applet files using the CODEBASE= and ARCHIVE= options.

For the Java device:

If you use the Java device driver with ODS to generate output containing a SAS/GRAPH applet, specify the path to the JAR file with the CODEBASE= option in the ODS statement.
When you specify DEVICE=JAVA, the users that view your output must have access to the appropriate Java applet. By default, SAS sets the value of CODEBASE= to refer to the executable file for the applet that is automatically installed with SAS. The default location of the SAS Java archive files is specified by the APPLETLOC= system option. You do not need to specify the CODEBASE= option if both of the following conditions are true.

- The default location is accessible by users who are viewing your web presentation.
- The SAS Java archive is installed at that location.

**Tip** Specify only the directory of the JAR file. The CODEBASE= location can be specified as a pathname or as a URL.

See *SAS/GRAPH: Reference* for information about specifying the location of control and applet files using the CODEBASE= and ARCHIVE= options.

**CONTENTS= 'file-specification' <(suboption(s))>**

opens a markup family destination and specifies the file that contains a table of contents for the output. These files remain open until you do one of the following:

- close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

**file-specification** specifies the file, fileref, or SAS catalog to write to.

**file-specification** is one of the following:

**external-file** is the name of an external output file.

- **Requirement** You must enclose external-file in quotation marks.

**fileref** is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in *SAS Statements: Reference*.

**entry.markup** specifies an entry in a SAS catalog to write to.

- **Interaction** If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

**suboption(s)** specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

**(DYNAMIC)** enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 798.
(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 798.

(NO_TOP_MATTER)
specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 799.

(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

title-text
is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 799.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL='Uniform-Resource-Locator')” on page 800.

CSSSTYLE= 'file-specification'<(media-type-1<…media-type-10>)>
specifies a cascading style sheet to apply to your output.

file-specification
specifies a file, fileref, or URL that contains CSS code.

file-specification is one of the following:

"external-file"
is the name of the external file.

Requirement You must enclose external-file in quotation marks.

fileref
is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

"URL"
is a URL to an external file.

Requirement You must enclose URL in quotation marks.
specifies one or more media blocks that correspond to the type of media that your output is rendered on. CSS uses media type blocks to specify how a document is to be presented on different media: on the screen, on paper, with a speech synthesizer, with a braille device, and so on.

The media block is added to your output in addition to the CSS code that is not contained in any media blocks. By using the media-type suboption, in addition to the general CSS code, you can import the section of a CSS file intended only for a specific media type.

**Default**
If no media-type is specified in your ODS statement, but you do have media types specified in your CSS file, then ODS uses the Screen media type.

**Range**
You can specify up to ten different media types.

**Requirements**
You must enclose media-type in parentheses.

You must specify media-type next to the file-specification specified by the CSSSTYLE= option.

**Tip**
If you specify multiple media types, all of the style information in all of the media types is applied to your output. However, if there is duplicate style information in different media blocks, then the styles from the last media block are used.

**Restriction**
The CSSSTYLE= option does not affect SAS/GRAPH output.

**Requirement**
CSS files must be written in the same type of CSS produced by the ODS HTML statement. To view the CSS code that ODS creates, you can do one of the following:
- specify the ODS TRACE DOM statement
- specify the DOM option

**Interaction**
If both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.

**See**
For complete documentation about ODS and Cascading Style Sheets, see SAS Output Delivery System: Advanced Topics

**Example**
“Example 6: Applying a CSS File to ODS Output” on page 527

DOM<="external-file"> specifies that the ODS document object model is written to the SAS log or an external file.

**external-file** is the name of an external output file.

**Requirement**
You must enclose external-file in quotation marks.

**See**
For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in SAS Output Delivery System: Advanced Topics.
ENCODING= local-character-set-encoding
overrides the encoding for input or output processing (transcodes) of external files.

See For information about the ENCODING= option, see SAS National Language Support (NLS): Reference Guide.

EVENT=event-name (FILE= | FINISH | LABEL= | NAME= | START | STYLE= | TARGET= | TEXT= | URL= )
specifies an event and the value for event variables that are associated with the event.

(FILE= BODY | CODE | CONTENTS | DATA | FRAME | PAGES | STYLESHEET);
triggers one of the known types of output files that correspond to the BODY=, CODE=, CONTENTS=, FRAME=, PAGES=, and STYLESHEET= options.

(FINISH)
triggers the finish section of an event.

See For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(LABEL='variable-value')
specifies the value for the LABEL event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the LABEL event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(NAME='variable-value')
specifies the value for the NAME event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the NAME event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(START)
triggers the start section of an event.

See For information about events, see “Understanding Events” in SAS Output Delivery System: Procedures Guide.

(STYLE=style-element)
specifies a style element.

See For information about style elements, see “Style Attributes Overview” in SAS Output Delivery System: Procedures Guide.

(TARGET='variable-value')
specifies the value for the TARGET event variable.

Requirement variable-value must be enclosed in quotation marks.

See For information about the TARGET event variable, see “Event Variables” in SAS Output Delivery System: Procedures Guide.

(TEXT='variable-value')
specifies the value for the TEXT event variable.
Requirement \textit{variable-value} must be enclosed in quotation marks.

See For information about the TEXT event variable, see “Event Variables” in \textit{SAS Output Delivery System: Procedures Guide}.

(\texttt{URL='variable-value'}) specifies the value for the URL event variable.

Requirement \textit{variable-value} must be enclosed in quotation marks.

See For information about the URL event variable, see “Event Variables” in \textit{SAS Output Delivery System: Procedures Guide}.

Default (\texttt{FILE='BODY'})

Requirement The EVENT= option's suboptions must be enclosed in parentheses.

\texttt{FRAME='file-specification' \langle suboption(s)\rangle}

opens a markup family destination and, for HTML output, specifies the file that integrates the table of contents, the page contents, and the body file. If you open the frame file, then you see a table of contents, a table of pages, or both, as well as the body file. For XML output, FRAME= specifies the file that contains the DTD. These files remain open until you do one of the following:

- close the destination with either an ODS \texttt{markup-family-destination CLOSE} statement or ODS \_\_\_CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

\textit{file-specification} specifies the file, fileref, or SAS catalog to write to.

\textit{file-specification} is one of the following:

\texttt{external-file}

is the name of an external output file.

Requirement You must enclose \texttt{external-file} in quotation marks.

\texttt{fileref}

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

See For information about the FILENAME statement, see “FILENAME Statement” in \textit{SAS Statements: Reference}.

\texttt{entry.markup}

specifies an entry in a SAS catalog to write to.

Interaction If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

\texttt{suboption(s)}

specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:
(DYNAMIC)

enables you to send output directly to a web server instead of writing it to a file.

See For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 798.

(NO_BOTTOM_MATTER)

specifies that no ending markup language source code be added to the output file.

See For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 798.

(NO_TOP_MATTER)

specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 799.

(TITLE=’title-text’)

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

title-text

is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE=’title-text’)” on page 799.

(URL= ‘Uniform-Resource-Locator’ )

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= ‘Uniform-Resource-Locator’)” on page 800.

Restriction

If you specify the FRAME= option, then you must also specify the CONTENTS= option, the PAGE= option, or both.

Example

“Example 2: Creating an XML File and a DTD” on page 520

GFOOTNOTE | NOGFOOTNOTE

controls the location where footnotes are printed in the graphics output.

GFOOTNOTE

writes footnotes that are created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The footnotes appear inside the graph borders.

NOGFOOTNOTE

writes footnotes that are created by ODS, which appear outside the graph borders.

Default GFOOTNOTE
Footnotes that are displayed by a markup language statement support all SAS/GRAPH FOOTNOTE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH FOOTNOTE statement, see “FOOTNOTE Statement” in SAS/GRAPH: Reference.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

GPATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)
specifies the location for all graphics output that is generated while the destination is open. Use this option when you want to write graphics output files to a location different that specified by the PATH= option for markup files. If you specify an invalid filename, the ActiveX and Java devices send output to the default filename. Other devices create the file as a directory and write output to that directory using the default filename. For information about how ODS names catalog entries and external files, see SAS/GRAPH: Reference.

'aggregate-file-storage-location'
specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement You must enclose aggregate-file-storage-location in quotation marks.

fileref
is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction If you specify a fileref in the GPATH= option, then ODS does not use information from the GPATH= option when it constructs links.

See For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

libref.catalog
specifies a SAS catalog to write to.

URL= 'Uniform-Resource-Locator' | NONE
specifies a URL for file-specification.

Uniform-Resource-Locator
is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

Requirement You must enclose Uniform-Resource-Locator in quotation marks.

NONE
specifies that no information from the GPATH= option appears in the links or references.

Tip This option is useful for building output files that can be moved from one location to another. If the links from the contents and page files are
constructed with a simple URL (one name), then they resolve, as long as the contents, page, and body files are all in the same location.

Default
If you omit the GPATH= option, then ODS stores graphics in the location that is specified by the PATH= option. If you do not specify the PATH= option, then ODS stores the graphics in the current directory. For more information, see the PATH= option.

GTITLE | NOGTITLE
controls the location where titles are printed in the graphics output.

GTITLE
writes the title that is created by SAS/GRAPH, the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure. The title appears inside the graph borders.

NOGTITLE
writes the title that is created by ODS, which appears outside of the graph borders.

Default GTITLE

Restrictions
Titles that are displayed by any markup language statement support most SAS/GRAPH TITLE statement options. The font must be valid for the browser. Options that ODS cannot handle, such as text angle specifications, are ignored. For details about the SAS/GRAPH TITLE statement, see TITLE statement.

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

HEADTEXT= 'markup-document-head'
specifies markup tags to place between the <HEAD> and </HEAD> tags in all of the output files.

markup-document-head
specifies the markup tags to place between the <HEAD> and </HEAD> tags.

Restriction
HEADTEXT= cannot exceed 256 characters.

Requirement
You must enclose markup-document-head in quotation marks.

Tips
ODS cannot parse the markup that you supply. It should be well-formed markup that is correct in the context of the <HEAD> and </HEAD> tags.

Use the HEADTEXT= option to define programs (such as JavaScript) that you can use later in the file.

(ID= identifier)
enables you to run multiple instances of the same destination at the same time. Each instance can have different options.

identifier
specifies another instance of the destination that is already open. identifier is numeric or a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numeric characters.
Restriction If identifier is numeric, it must be a positive integer.

Requirement You must specify the ID= option immediately after the destination name.

Tip You can omit the ID= option and instead use a name or a number to identify the instance.

Example “Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594

METATEXT= 'metatext-for-document-head'
specifies HTML code to use as the <META> tag between the <HEAD> and </HEAD> tags of all of the HTML output files.

'metatext-for-document-head'
specifies the HTML code that provides the browser with information about the document that it is loading. For example, this attribute could specify the content type and the character set to use.

Requirement You must enclose metatext-for-document-head in quotation marks.

Default If you do not specify METATEXT=, then ODS writes a simple <META> tag, which includes the content-type of the document and the character set to use, to all the HTML files that it creates.

Restriction METATEXT= cannot exceed 256 characters.

Tip ODS cannot parse the HTML code that you supply. It should be well-formed HTML code that is correct in the context of the <HEAD> tags. If you are using METATEXT= as it is intended, then your META tag should look like this:

<META your-metatext-is-here>

NEWFILE= starting-point
creates a new body file at the specified starting-point.

starting-point
is the location in the output where you want to create a new body file.

ODS automatically names new files by incrementing the name of the body file. In the following example, ODS names the first body file REPORT.XML. Additional body files are named REPORT1.XML, REPORT2.XML, and so on.

Example:

BODY= 'REPORT.XML'

starting-point is one of the following:

BYGROUP
starts a new file for the results of each BY group.

NONE
writes all output to the body file that is currently open.
OUTPUT
starts a new body file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

Alias TABLE

PAGE
starts a new body file for each page of output. A page break occurs when a procedure explicitly starts a new page (not because the page size was exceeded) or when you start a new procedure.

PROC
starts a new body file each time you start a new procedure.

Default NONE

Restriction The NEWFILE= option cannot be used in conjunction with the BODY=fileref option.

Tips If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first body file MAY5.XML. Additional body files are named MAY6.XML, MAY7.XML, and so on.

Example:
BODY= 'MAY5.XML'

OPTIONS ( DOC= ) | sub-option(s)
specifies ODS tagset-specific suboptions and a named value.

(DOC='QUICK' | 'HELP' | 'SETTINGS')
provides information about the specified tagset.

QUICK
describes the options available for this tagset.

HELP
provides generic help and information with a quick reference.

SETTINGS
provides the current option settings.

Requirement All values must be enclosed in quotation marks.

sub-option(s)
specifies one or more suboptions that are valid for the specified tagset. To list suboptions that are valid for a tagset, specify DOC="HELP" or DOC="QUICK" with the OPTIONS option.

Requirement The OPTION suboptions must be enclosed in parentheses.

Example "Example: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information " on page 802

PAGE='file-specification' <(suboption(s))>
opens a markup family destination and specifies the file that contains a description of each page of the body file, and contains links to the body file. ODS produces a new
page of output whenever a procedure requests a new page. These files remain open until you do one of the following:

- close the destination with either an ODS `markup-family-destination` CLOSE statement or ODS `_ALL_` CLOSE statement.
- open the same destination with a second markup family statement. This closes the first file and opens the second file.

**file-specification**

specifies the file, fileref, or SAS catalog to write to.

**file-specification** is one of the following:

- **external-file**
  - is the name of an external output file.
  
  **Requirement** You must enclose **external-file** in quotation marks.

- **fileref**
  - is a file reference that has been assigned to an external file. Use the `FILENAME` statement to assign a fileref.

  **See** For information about the `FILENAME` statement, see “`FILENAME Statement`” in *SAS Statements: Reference*.

- **entry.markup**
  - specifies an entry in a SAS catalog to write to.

  **Interaction** If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

- **suboption(s)**
  - specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

  - **(DYNAMIC)**
    - enables you to send output directly to a web server instead of writing it to a file.

    **See** For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 798.

  - **(NO_BOTTOM_MATTER)**
    - specifies that no ending markup language source code be added to the output file.

    **See** For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 798.

  - **(NO_TOP_MATTER)**
    - specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

    **See** For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 799.
(TITLE='title-text')
inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

*title-text* is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 799.

(URL= 'Uniform-Resource-Locator')
specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 800.

Interaction The SAS system option PAGESIZE= has no effect on pages in HTML output except when you are creating batch output. For information about the PAGESIZE= option, see “PAGESIZE= System Option” in SAS System Options: Reference.

PARAMETERS= (parameter-pair-1 ... parameter-pair-n)
writes the specified parameters between the tags that generate dynamic graphics output.

*parameter-pair*
specifies the name and value of each parameter. *parameter-pair* has the following form:

'parameter-name'='parameter-value'

*parameter-name* is the name of the parameter.

*parameter-value* is the value of the parameter.

Requirement You must enclose *parameter-name* and *parameter-value* in quotation marks.

Interaction Use PARAMETERS= in conjunction with SAS/GRAPH procedures and the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

See SAS/GRAPH: Reference for valid parameters for the Graph Applet, Map Applet, Contour Applet, and the MetaView Applet.

PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)
specifies the location of an aggregate storage location or a SAS catalog for all markup files. If the GPATH= option is not specified, all graphics output files are written to the "aggregate-file-storage-specification" or libref.

'aggregate-file-storage-location'
specifies an aggregate storage location such as directory, folder, or partitioned data set.
Requirement  You must enclose aggregate-file-storage-location in quotation marks.

fileref

is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction  If you use a fileref in the PATH= option, then ODS does not use information from PATH= when it constructs links.

See  For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

libref,catalog

specifies a SAS catalog to write to.

See  For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Statements: Reference.

URL= 'Uniform-Resource-Locator' | NONE

specifies a URL for the file-specification.

Uniform-Resource-Locator

is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

NONE

specifies that no information from the PATH= option appears in the links or references.

Tip  This option is useful for building output files that can be moved from one location to another. The links from the contents and page files must be constructed with a single-name URL, and the contents, page, and body files must be in the same location.

Interaction  If you use the BODY= or FILE= external file option in conjunction with the PATH= option, the external file specification should not include path information.

RECORD_SEPARATOR= 'alternative-separator' | NONE

specifies an alternative character or string that separates lines in the output files.

Different operating environments use different separator characters. If you do not specify a record separator, then the files are formatted for the environment where you run the SAS program. However, if you are generating files for viewing in a different operating environment that uses a different separator character, then you can specify a record separator that is appropriate for the target environment.

alternative-separator

represents one or more characters in hexadecimal or ASCII format. For example, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= '0D0A'x

Operating Environment Information

In a mainframe environment, the following option specifies a record separator for a carriage return character and a linefeed character for use with an ASCII file system:
RECORD_SEPARATOR='\0D25'x

Requirement You must enclose alternative-separator in quotation marks.

NONE produces the markup language that is appropriate for the environment where you run the SAS job.

Windows Specifics
In a mainframe environment, by default, ODS produces a binary file that contains embedded record separator characters. This binary file is not restricted by the line-length restrictions on ASCII files. However, if you view the binary files in a text editor, then the lines run together. If you want to format the files so that you can read them with a text editor, then use RECORD_SEPARATOR=NONE. In this case, ODS writes one line of markup language at a time to the file. When you use a value of NONE, the logical record length of the file that you are writing to must be at least as long as the longest line that ODS produces. If the logical record length of the file is not long enough, then the markup language might wrap to another line at an inappropriate place.

Aliases RECSEP=
        RS=

STYLE= style-template specifies the style template to use in writing the output files.

style-template describes how to display the presentation aspects (color, font face, font size, and so on) of your SAS output. A style template determines the overall appearance of the documents that use it. Each style template consists of style elements.

Interaction The STYLE= option is not valid when you are creating XML output.

Note If you are using SAS Studio, you do not need to specify the STYLE= option. You can go to Preferences ⇒ Results and change the style from the drop-down list for your selected destination.

See For a complete discussion of style templates, see “TEMPLATE Procedure: Creating a Style Template” in SAS Output Delivery System: Procedures Guide.

Default If you do not specify a style template, then ODS uses the file that is specified in the SAS registry subkey ODS ⇒ DESTINATIONS ⇒ MARKUP. By default, this value specifies Default.

Interaction If you specify the STYLE= option in an ODS HTML4 statement and subsequently need PROC PRINT output to use new style templates in another ODS HTML4 statement, close the first statement before specifying the second statement.

STYLESHEET='file-specification' <(suboption(s))>
opens a markup family destination and places the style information for markup output into an external file, or reads style sheet information from an existing file. These files remain open until you do one of the following:
• close the destination with either an ODS markup-family-destination CLOSE statement or ODS _ALL_ CLOSE statement.

• open the same destination with a second markup family statement. This closes the first file and opens the second file.

**file-specification**
specifies the file, fileref, or SAS catalog to write to.

**file-specification** is one of the following:

- **external-file**
  is the name of an external output file.

  **Requirement** You must enclose `external-file` in quotation marks.

- **fileref**
  is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

  **See** For information about the FILENAME statement, see “FILENAME Statement” in SAS Statements: Reference.

- **entry.markup**
  specifies an entry in a SAS catalog to write to.

  **Interaction** If you specify an entry name, you must also specify a library and catalog. See the discussion of the PATH= option.

**suboption(s)**
specifies one or more suboptions in parentheses. Suboptions are instructions for writing the output files. Suboptions can be the following:

- **(DYNAMIC)**
  enables you to send output directly to a web server instead of writing it to a file.

  **See** For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 798.

- **(NO_BOTTOM_MATTER)**
  specifies that no ending markup language source code be added to the output file.

  **See** For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 798.

- **(NO_TOP_MATTER)**
  specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

  **See** For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 799.

- **(TITLE='title-text')**
  inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.
*title-text*

is the text in the metadata of a file that indicates the title.

See For complete documentation about the TITLE= suboption, see “(TITLE='title-text')” on page 799.

*(URL= 'Uniform-Resource-Locator' )*

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator')” on page 800.

Note By default, if you do not specifically send the information to a separate file, then the style sheet information is included in the specified HTML file.

Example “Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

**TEXT=text-string**

inserts text into your document by triggering the paragraph event and specifying a text string to be assigned to the VALUE event variable.

Default By default the TEXT= option is used in a paragraph event.

Tip You can specify a text-string for a specific event by using the TEXT= option with the EVENT= option by using the following syntax:

EVENT=event-name (TEXT=text-string)

See For information about events and event variables, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in *SAS Output Delivery System: Procedures Guide*.

Example “Example: Conditionally Excluding Output Objects and Sending Them to Different Output Destinations” on page 325

**TRANTAB= 'translation-table'**

specifies the translation table to use when transcoding a file for output.


---

**Diagnostic Tagsets**

The following diagnostic tagsets are available for the ODS Tagset statement.

**EVENT_MAP**

creates XML output that shows which events are being triggered and which variables are used by an event to send output from a SAS process to an output file. When you run a SAS process with EVENT_MAP, ODS writes XML to an output file that shows all event names and variable names as tags. The output helps you create your own tagsets.
NAMEDHTML
creates HTML output similar to STYLE_POPUP on page 798, but with all the objects labeled as they are when using ODS TRACE.

SHORT_MAP
creates a subset of the XML output that is created by the EVENT_MAP tagset.

STYLE_DISPLAY
creates a sample page of HTML output that is similar to STYLE_POPUP output. The output helps you create and modify styles.

See STYLE_POPUP on page 798

STYLE_POPUP
displays the resolved ODS style template for any item that you select. This information is displayed in a window if you are using Internet Explorer.

TEXT_MAP
creates text output that shows which events are being triggered as ODS handles the output objects.

Tip You can use the TEXT_MAP output as an alternative to the output that is created by the EVENT_MAP tagset.

See EVENT_MAP on page 797

Suboptions

(DYNAMIC)
enables you to send output directly to a web server instead of writing it to a file. This option sets the value of the CONTENTTYPE= style attribute. For more information, see CONTENTTYPE= on page 1015 in PROC TEMPLATE.

Default If you do not specify DYNAMIC, then ODS sets the value of HTMLCONTENTTYPE= for writing to a file.

Restriction If you specify the DYNAMIC suboption with one of the following options in the ODS HTML statement, then you must specify it for all of these options in that statement.

• BODY=
• CONTENTS=
• PAGE=
• FRAME=
• STYLESHEET=
• TAGSET=

Requirements You must enclose DYNAMIC in parentheses.

You must specify DYNAMIC next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

(NO_BOTTOM_MATTER)
specifies that no ending markup language source code be added to the output file.

Alias NOBOT
**Requirements**

You must enclose NO_BOTTOM_MATTER in parentheses.

You must specify NO_BOTTOM_MATTER next to the **file-specification** specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the **tagset-name** specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

**Interactions**

The NO_BOTTOM_MATTER suboption, in conjunction with the NO_TOP_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

**Tip**

If you want to leave a body file in a state that you can append to with ODS, then use NO_BOTTOM_MATTER with the **file-specification** BODY= option in any markup language statement.

**See**

The NO_TOP_MATTER suboption

---

**NO_TOP_MATTER**

specifies that no beginning markup language source code be added to the top of the output file. For HTML 4.0, the NO_TOP_MATTER option removes the style sheet.

**Alias**

NOTOP

**Requirements**

You must enclose NO_TOP_MATTER in parentheses.

You must specify NO_TOP_MATTER next to the **file-specification** specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the **tagset-name** specified by the TAGSET= option.

If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

**Interactions**

The NO_TOP_MATTER suboption, in conjunction with the NO_BOTTOM_MATTER suboption, makes it possible for you to add output to an existing file. You can then put your own markup language between output objects in the file.

When you are opening a file that ODS has previously written to, use the ANCHOR= option to specify a new base name for the anchors. This step prevents duplicate anchors.

**See**

The NO_BOTTOM_MATTER suboption and the ANCHOR= option

---

**(TITLE=’title-text’)**

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.
**title-text**

is the text in the metadata of a file that indicates the title.

**Requirements**

You must enclose TITLE= in parentheses.

You must enclose *title-text* in quotation marks.

**Tip**

If you are creating a web page that uses frames, then it is the TITLE= specification for the frame file that appears in the browser window title bar.

**Example**

“Example 3: Creating Multiple Markup Output” on page 522

**Example**

*(URL= 'Uniform-Resource-Locator' )*

specifies a URL for the file-specification. ODS uses this URL (instead of the filename) in all the links and references that it creates and that point to the file.

**Requirements**

You must enclose URL= 'Uniform-Resource-Locator' in parentheses.

You must enclose *Uniform-Resource-Locator* in quotation marks.

You must specify URL= 'Uniform-Resource-Locator' next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESHEET= option, or next to the tagset-name specified by the TAGSET= option.

**Tips**

This option is useful for building HTML files that can be moved from one location to another. The links from the contents and page files must be constructed with a single name URL, and the contents, page, and body files must all be in the same location.

You never need to specify this suboption with the FRAME= option because ODS files do not reference the frame file.

**Example**

“Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 525

**Details**

**Understanding Tagsets**

A tagset is a type of template that defines how to generate a markup language output type from SAS data. A markup language is a set of tags and format codes that are embedded in text in order to define layout and certain content.

You can use the ODS Tagset statement to specify a tagset to create markup language output from the Output Delivery System. SAS provides tagset definitions for a variety of markup language output. For example, there are several SAS tagsets for XML output, HTML output, XSL, and so on. In addition to using the tagsets provided by SAS, you can modify the SAS tagsets, and you can create your own. By supplying new tagset definitions, ODS output and XML LIBNAME engine output is user-configurable, generating a wider variety of markup language output. For information about modifying SAS tagsets and creating your own tagsets, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in *SAS Output Delivery System: Procedures Guide.*
**Listing Tagset Names**

To see a list of available tagsets, issue the following SAS statements or view them in the Templates window.

- **Templates window:**
  
  To display a list of the available tagsets using the SAS Explorer window, follow these steps:

  1. From any window in an interactive SAS session, select **View ⇒ Results**.
  2. In the Results window, select **View ⇒ Templates**.
  3. In the Templates window, select and open `Sashelp.Tmplmst`.
  4. Select and open the **Tagsets** folder, which contains a list of available tagsets. If you want to view the underlying SAS code for a tagset, then select the tagset and open it.

*Windows Specifics*

For information about navigating in the Explorer window without a mouse, see "Window Controls and General Navigation" in the SAS documentation for your operating environment.

- **TEMPLATE procedure:**

  You can also display a list of the available tagsets by submitting the following PROC TEMPLATE statements:

  ```sas
  proc template;
  list tagsets;
  quit;
  ```

  By default, PROC TEMPLATE lists the tagsets in Sashelp.Tmplmst and Sasuser.Templat. Typically, Sashelp.Tmplmst is a read-only item store for the SAS tagsets, and Sasuser.Templat is the item store for user-defined tagsets.

**Viewing the Source of a Tagset**

To see the source for a tagset definition, you can either open the tagset in the SAS Explorer window, or use PROC TEMPLATE and specify the two-level name of the tagset. To see the source of the SAS tagset CHTML, issue these SAS statements:

```sas
proc template;
  source tagsets.chtml;
quit;
```

**Viewing Available Options for a Tagset**

To view the options that are available for a specific tagset, use the OPTIONS (DOC=) option with one of the following specified:

- **QUICK**
  
  describes the options available for this tagset.

- **HELP**
  
  provides generic help and information with a quick reference.

- **SETTINGS**
  
  provides the current option settings.
Example: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information

Features:
ODS TAGSETS.HTMLPANEL statement action:
   CLOSE

ODS TAGSETS.HTMLPANEL statement options:
   OPTIONS
      (DOC="HELP")
      FILE=

Other features:
   PROC PRINT

Details
The following example prints to the SAS log the OPTIONS suboptions for the HTMLPANEL tagset and a description of each available suboption.

Program
ods tagsets.htmlpanel file='Help.html' options (doc="help");
proc print data=Sashelp.Class;
  run;
ods _all_ close;

Program Description

Print information about the OPTIONS suboptions to the SAS log file.
   ods tagsets.htmlpanel file='Help.html' options (doc="help");

Print the data set Sashelp.Class. The PROC PRINT statement prints the Sashelp.Class data set.
   proc print data=Sashelp.Class;
     run;

Close all destinations. Close the ODS TAGSETS.HTMLPANEL destination and any other open destinations. This statement also closes all the files that are associated with each open destination. If you do not close a destination, then you cannot view the files in a browser window.
   ods _all_ close;

Output
Specify the "DOC='HELP'" suboption to print all of the OPTIONS suboptions and information about each of the suboptions to the SAS log.
ODS TAGSETS.RTF Statement

Opens, manages, or closes the RTF destination, which produces measured output that is written in Rich Text Format.

Valid in: Anywhere

Category: ODS: Third-Party Formatted

Restriction: When a SAS generated table is copied and pasted into or imported into a Word document, it is assigned the Word style "Normal". As a result, most of the custom ODS formatting is lost, and the output does not display as expected.

Interaction: By default, when you execute a procedure that uses the FORMCHAR system option (for example, PROC CONTENTS), ODS formats the output in SAS Monospace font. If you are creating output that will be viewed in an operating environment that does not have SAS software installed, this output will not be displayed correctly. The SAS Monospace font is not recognized if SAS is not installed. For the correct display of your document, include the following statement before your SAS program:

```sas
OPTIONS FORMCHAR='|----|+|---+=-|\<>*';
```

Tip: Microsoft Word 2002 is the current official minimum level that is supported. However, no problems have been found with Microsoft Word 2000 and SAS RTF files.

See Also


---

**ODS TAGSETS.RTF Statement**

These are the options supported by this tagset.

Sample usage:

```sas
ods tagsets.htmlpanel file='template.html'
```

---

Tip: Microsoft Word 2002 is the current official minimum level that is supported. However, no problems have been found with Microsoft Word 2000 and SAS RTF files.
Syntax

**ODS TAGSETS.RTF** 

\(<\langle ID\rangle=\text{identifier}\rangle\) \textit{action};

**ODS TAGSETS.RTF** 

\(<\langle ID\rangle=\text{identifier}\rangle\) \textit{<option(s)>};

**Summary of Optional Arguments**

\((ID=\text{identifier})\)
Open multiple instances of the same destination at the same time

**ANCHOR= 'anchor-name'**
Specify a unique base name for the anchor tag that identifies each output object in the current body file

**AUTHOR= 'author-text'**
Specify the text string that identifies the author. This text string is inserted into the metadata of a file.

**BASE= 'base-text'**
Specify text to use as the first part of all links and references that ODS creates in output files

**BOX_SIZING=(CONTENT_BOX | BORDER_BOX)**
Specify how to measure the width of cells. Use to override the default value of BOX_SIZING for a destination

**CLOSE**
Close the destination and the file that is associated with it

**COLUMNS= n | MAX**
Specify the number of columns to create on each page of output

**DEVICE= device-driver**
Specify a device for the RTF output destination

**ENCODING= local-character-set-encoding**
Override the encoding for input or output processing (transcodes) of external files

**EXCLUDE exclusion(s) | ALL | NONE**
Exclude output objects from the destination

**FILE= 'external-file\| fileref**
Open the ODS RTF destination and specify the name of the file to which to write information

**GFOOTNOTE | NOGFOOTNOTE**
Specify the location where footnotes are printed in the graphics output

**GTITLE | NOGTITLE**
Control the location where titles are printed in the graphics output

**IMAGE_DPI**
Specify the image resolution for the graphical output

**NEWFILE= starting-point**
Create a new body file at the specified starting point

**OPERATOR= 'text-string'**
Insert the text that you specify into the metadata of the RTF file

**OPTIONS (CONTENTS= | CONTINUE_TAG= | DOC= | SECT= | TABLES_OFF= | TOC_DATA= | TOC_LEVEL= | TROWD= | TRHDR= | TROWHDRCELL= | VSPACE= | WATERMARK=)**
Specify TAGSETS.RTF-specific options

**PACKAGE <package-name>**
Specify that the output from the destination be added to an ODS package
**PAGEPANELS= n | NONE**
Specify the number of panels that are rendered for a multipanel table

**PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL='Uniform-Resource-Locator' | NONE)**
Specify the location of an aggregate storage location or a SAS catalog for all RTF files

**PREPAGE='text-string'**
Specify a text string that occurs before a table on a page

**RECORD_SEPARATOR= 'alternative-separator' | NONE**
Specify an alternative character or string to separate lines in the output files

**SELECT selection(s) | ALL | NONE**
Select output objects for the destination

**SHOW**
Write to the SAS log the current selection or exclusion list for the destination

**STARTPAGE= BYGROUP | YES | NO | NOW**
Control page breaks

**STYLE= style-template**
Specify a style template to use in writing the RTF files

**TABLEROWS= n**
Specify the number of rows that are rendered in a table

**TEXT= 'text-string'**
Insert text into your RTF output

**TRANTAB= translation-table**
Specify a translation table to use when you transcode a file for output

**UNIFORM**
Specify that every page of a table is formatted the same

**Actions**
The following actions are available for the ODS TAGSETS.RTF statement:

**CLOSE**
closes the destination and any files that are associated with it.

**Tip**
When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

**EXCLUDE exclusion(s) | ALL | NONE**
excludes one or more output objects from the destination.

**Default**
NONE

**Exclusion(s)**

**Restriction**
A destination must be open for this action to take effect.

**See**
“ODS EXCLUDE Statement” on page 321

**SELECT selection(s) | ALL | NONE**
selects output objects for the specified destination.

**Default**
ALL

**Selection(s)**

**Restriction**
A destination must be open for this action to take effect.

**See**
“ODS SELECT Statement ” on page 758
SHOW
writes the current selection list or exclusion list for the destination to the SAS log.

Restriction  The destination must be open for this action to take effect.

Tip  If the selection or exclusion list is the default list (SELECT ALL), then SHOW also writes the entire selection or exclusion list. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 39.

See  “ODS SHOW Statement” on page 771

Optional Arguments

ANCHOR= 'anchor-name'
specifies the base name for the RTF anchor tag that identifies each output object in the current file.

Each output object must have an anchor tag to which other files link or reference. The references, which ODS automatically creates, point to the name of an anchor. Therefore, each anchor name in a file must be unique.

anchor-name
is the base name for the RTF anchor tag that identifies each output object in the current file.

ODS increments the name that you specify and creates unique anchor names. For example, if you specify ANCHOR= 'tabulate', then ODS names the first anchor tabulate. The second anchor is named tabulate1; the third is named tabulate2, and so on.

Requirement  You must enclose anchor-name in quotation marks.

Alias  NAMED_DEST= | BOOKMARK=

Tips  It is useful to specify new anchor names at various points in your program when you want other RTF files to link to specific parts of your RTF output. Because you can control where the anchor name changes, you know in advance what the anchor name is at those points.

You can change anchor names as often as you want by submitting the ANCHOR= option in an ODS RTF statement anywhere in your program. After you specify an anchor name, it remains in effect until you specify a new one.

AUTHOR= 'author-text'
inserts the text string that you specify as the author into the metadata of a file.

author-text
is the text in the metadata of an open file that indicates the author.

Requirement  You must enclose author-text in quotation marks.

BASE= 'base-text'
specifies the text to use as the first part of references that ODS creates in the output file.
**base-text**

is the text that ODS uses as the first part of all references that ODS creates in the file.

Consider this specification:

```
BASE='http://www.your-company.com/local-url/'
```

In this case, ODS creates links that begin with the string `http://www.your-company.com/local-url/`.

**Requirement**

You must enclose *base-text* in quotation marks.

**BOX_SIZING=(CONTENT_BOX | BORDER_BOX)**

specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for a destination. The default value can be found in the SAS registry.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at [http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing](http://www.w3.org/TR/2002/WD-css3-box-20021024/#box-sizing).

**COLUMNS=n | MAX**

specifies the number of columns to place across each page of output.

\[ n \]

is the number of one-inch columns that you want on the page.

\[ MAX \]

specifies the maximum number of columns for the paper size and margin setting. This value is dependent upon the paper size and page orientation.

**Default**

The number of columns that fit on the page

**Interaction**

When you specify the COLUMNS= option, the STARTPAGE=NO option is not honored.

**Tips**

Titles are considered tables and not RTF instructions in Measured RTF (ODS TAGSETS.RTF statement). When you use the COLUMNS= option with Measured RTF, titles appear at the top of each column. However, ODS truncates the titles to fit the column width.

If you specify a value greater than the maximum number of columns that can fit on the page, a note is written to the SAS log that states what the maximum value can be for that page.

**DEVICE= device-driver**

specifies the name of a device driver. ODS automatically selects an optimal default device for each open output destination.

The following table lists the default devices for the most common ODS output destinations. These default devices are used when graphics are created using SAS/GRAPH or ODS Graphics. In the third maintenance release of SAS 9.4, EPUB3 is the default EPUB destination. EPUB2 was the default EPUB version in prior releases of SAS 9.4. This default is set in the Registry. For a complete list of
supported devices and file types, see “Supported File Types for Output Destinations” on page 349.

### Table 6.20 Default Devices for ODS Output Destinations

<table>
<thead>
<tr>
<th>Output Destination</th>
<th>Default Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPUB2</td>
<td>PNG</td>
</tr>
<tr>
<td>EPUB3 (EPUB)</td>
<td>SVG</td>
</tr>
<tr>
<td>HTML</td>
<td>PNG</td>
</tr>
<tr>
<td>HTML5</td>
<td>SVG</td>
</tr>
<tr>
<td>LISTING</td>
<td>PNG</td>
</tr>
<tr>
<td>Measured RTF</td>
<td>EMF</td>
</tr>
<tr>
<td>RTF</td>
<td>EMF</td>
</tr>
<tr>
<td>Markup Tagsets</td>
<td>PNG</td>
</tr>
</tbody>
</table>

**Tips**

Specifying a device on the ODS DEVICE= option takes precedence over the SAS global option and the graphics option.

For information about using the SAS Registry to change defaults, see “Changing SAS Registry Settings for ODS” on page 40.

**See**


**ENCODING= local-character-set-encoding**

overrides the encoding for input or output processing (transcodes) of external files.

**See**

For information about the ENCODING= option, see *SAS National Language Support (NLS): Reference Guide*.

**FILE= 'external-file' | fileref**

opens the RTF destination and specifies the RTF file or SAS catalog to which to write. This file remains open until you do one of the following actions:

- Close the RTF destination with ODS RTF CLOSE or ODS _ALL_ CLOSE.
- Specify a different file to which to write.

**external-file**

is the name of an external file to which to write.

**Requirement** You must enclose *external-file* in quotation marks.

**fileref**

is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.
Restriction

You cannot use the FILE=fileref option with the NEWFILE= option.

See

The section on statements in *SAS Statements: Reference* for information about the FILENAME statement.

Alias

BODY=

Interaction

In an ODS RTF statement that refers to an open RTF destination, the FILE= option forces ODS to close the destination and all files that are associated with it. ODS then opens a new instance of the destination. For more information, see “Opening and Closing the RTF Destination” on page 748.

See

NEWFILE= option

**GFOOTNOTE | NOGFOOTNOTE**

controls the location of the footnotes that are defined by the graphics program that generates the RTF output.

**GFOOTNOTE**

includes all of the currently defined footnotes within the graphics output.

**NOGFOOTNOTE**

prevents all of the currently defined footnotes from appearing in the graphics file. Instead, they become part of the RTF file.

Default

GFOOTNOTE

Restriction

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

**GTITLE | NOGTITLE**

controls the location of the titles that are defined by the graphics program that generates the RTF output.

**GTITLE**

includes all of the currently defined titles within the graphics output that is called by the body file.

**NOGTITLE**

prevents all of the currently defined titles from appearing in the graphics output. Instead, the titles become part of the RTF file.

Default

GTITLE

Restriction

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SGPLOT procedure, the SGPANEL procedure, or the SGSCATTER procedure.

**(ID= identifier)**

enables you to run multiple instances of the same destination at the same time. Each instance can have different options.

**identifier**

specifies another instance of the destination that is already open. *identifier* is numeric or a series of characters that begin with a letter or an underscore. Subsequent characters can include letters, underscores, and numeric characters.
**Restriction** If `identifier` is numeric, it must be a positive integer.

**Requirement** You must specify the ID= option immediately after the destination name.

**Tip** You can omit the ID= option and instead use a name or a number to identify the instance.

**Example** “Example 1: Opening Multiple Instances of the Same Destination at the Same Time” on page 594

---

**IMAGE_DPI**
specifies the image resolution for graphical output.

<table>
<thead>
<tr>
<th>Alias</th>
<th>DPI=</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>200</td>
</tr>
</tbody>
</table>

**CAUTION** Caution: When using high DPI= or DPI_IMAGE values (values over 1000), you might need to increase memory allocations. To increase memory, set the MEMSIZE= system option to 500M or higher. You can also decrease the DPI= value to ensure that you do not run out of memory.

**NEWFILE= starting-point**
creates a new file at the specified `starting-point`.

`starting-point` can be one of the following:

- **BYGROUP**
  starts a new file for the results of each BY group.

- **NONE**
  writes all output to the body file that is currently open.

- **OUTPUT**
  starts a new file for each output object. For SAS/GRAPH this means that ODS creates a new file for each SAS/GRAPH output file that the program generates.

<table>
<thead>
<tr>
<th>Alias</th>
<th>TABLE</th>
</tr>
</thead>
</table>

**PROC**
starts a new file each time you start a new procedure.

<table>
<thead>
<tr>
<th>Default</th>
<th>NONE</th>
</tr>
</thead>
</table>

**Restriction** You cannot use the NEWFILE= option with the FILE=fileref option.

**Tip** If you end the filename with a number, then ODS begins incrementing with that number. In the following example, ODS names the first body file MAY5.XML, and names additional body files MAY6.XML, MAY7.XML, and so on.

**OPERATOR= ’text-string’**
inserts the text that you specify into the metadata of the RTF file.

`text-string` is the text in the metadata of a file that indicates the author.
**Requirement**

You must enclose *text-string* in quotation marks.

**OPTIONS (CONTENTS= | CONTINUE_TAG= | DOC= | SECT= | TABLES_OFF= | TOC_DATA= | TOC_LEVEL= | TROWD= | TRHDR= | TROWHDRCELL= | VSPACE= | WATERMARK=)**

specifies ODS TAGSETS.RTF-specific suboptions and a named value.

**(CONTENTS= 'YES')**

produces a table of contents (TOC) page for RTF documents that are opened in Microsoft Word. The table of contents page contains a Table of Contents field that puts all of the contents information that is embedded in the document into a table of contents. To display the captured TOC data, you must turn on the option `TOC_DATA` on page 813. To expand the table of contents, right-click under the title in Microsoft Word and select **Update Field** from the selection list.

**Note:** From Microsoft Word, you might need to right-click lower on the page to get the **Update Field** value to appear in the selection list.

**YES**

adds a table of contents page to the top of the RTF file. This table of contents page is followed by a page break.

**Alias** ON

**Requirement**

All values must be enclosed in quotation marks.

**Tips**

To go to a specific topic in the document, you can double-click or hold down the Ctrl key and click on the topic in the table of contents. You might have to configure Microsoft Word to use the Ctrl key method. Select **Tools** ⇒ **Options** ⇒ **Edit** and then select **Use Ctrl + Click to follow hyperlink**.

The `TOC_DATA` option must be set to YES to capture TOC data. If you specify `CONTENTS=YES`, but you do not specify `TOC_DATA`, no Table of Contents data is captured. The error displayed on the Table of Contents page is "Error! No table of contents entries found".

**See**

`TOC_DATA` suboption for details about displaying the contents embedded in the document.

**Example**

“Example 1: Creating a Table of Contents” on page 823

**(CONTINUE_TAG ='ON' | 'OFF')**

specifies whether to add a continue tag to the RTF file when a table breaks and is continued to the next page.

**ON**

instructs ODS to add a continue tag to the RTF file when a table breaks and is continued to the next page.

**Alias** YES

**OFF**

instructs ODS not to add the continue tag when a table breaks and is continued to the next page.

**Alias** NO
Requirement You must enclose all values in quotation marks.

Example “Example 1: Creating a Table of Contents” on page 823

(\texttt{DOC=}'QUICK' | 'HELP' | 'SETTINGS')
provides information about the tagset.

\texttt{QUICK}
describes the options available for this tagset.

\texttt{HELP}
provides generic help and information with a quick reference.

\texttt{SETTINGS}
provides the current option settings.

Requirement All values must be enclosed in quotation marks.

Example “Example: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information “ on page 802

(\texttt{SECT=}''rtf\_control\_string'' | 'OFF' | 'NONE')
inserts RTF control words into the section data specifications.

\texttt{rtf\_control\_string}
specifies RTF control words used to format the section data.

\texttt{OFF}
turns off the usage of RTF control words for the section data and resets the \texttt{rtf\_control\_string} to null.

Alias \texttt{NO}

\texttt{NONE}
stops new RTF control words from being inserted into the file for the section data. ODS continues to use the section data information that was set before the use of NONE until it is reset.

Requirement All values must be enclosed in quotation marks.

Tip To reset the \texttt{rtf\_control\_string}, assign a different value or use the OFF or NO values.


(\texttt{TABLES\_OFF=}'style\_elements' | 'STYLE\_ELEMENTS' | 'OFF' )
determines whether tables are used. A table can consist of one cell or many cells. SAS puts all of the text that you create into tables for RTF output. Use this suboption for tables that are text holders like titles, footnotes, and TEXT=. You should not use this suboption for tables produced by reporting procedures.

Note: You can turn on Gridlines in Microsoft Word.

\texttt{style\_elements}
specifies the style element for formatting. For example, the following statement turns off tables that use the USERTEXT style element. The text specified by the TEXT= option is not placed in the table.
ODS TAGSETS.RTF Statement

ods tagsets.rtf options (Tables_OFF='usertext');
ods tagsets.rtf text="Text is not placed in a table");

STYLE_ELEMENTS
  lists the output style elements in the SAS log.

  OFF
  turns the option off. Therefore, ODS places the information output next into
  the RTF file inside a table. This action is the default option.

    Alias   NO

    Requirement   You must enclose all values in quotation marks.

    See   “General ODS Style Elements ” on page 965 for information
          about style elements

    Example   “Example 3: Using the TABLES_OFF Suboption” on page 828

(TOC_DATA ='ON' | 'OFF')
  specifies whether to show the contents data in the RTF file.

  OFF
  instructs ODS not to display the table of contents data in the RTF file.

    Alias   NO

  ON
  instructs ODS to display the hidden text of the table of contents in the RTF
  file.

    Alias   YES

    Requirement   You must enclose all values in quotation marks.

    Example   “Example 1: Creating a Table of Contents” on page 823

(TOC_LEVEL ='n')
  controls the level of the expansion of the table of contents in RTF documents.
  This option must be used with the (CONTENTS=YES) and (TOC_DATA=YES)
  options specified.

    n
    specifies the level of expansion. For example, TOC_LEVEL="0" results in a
    fully expanded table of contents. TOC_LEVEL="2" results in a table of
    contents that is expanded to two levels.

  OFF
  restores all levels of expansion that are shown in the Table of Contents.

    Alias   NO

    Requirement   You must enclose all values in quotation marks.

    Tip   The TOC_DATA= and CONTENTS= suboptions must be set to
          YES to capture TOC data. For more information about viewing
          the TOC, see “Example 1: Creating a Table of Contents” on page
          823.
See TOC_DATA= and the CONTENTS= suboption for details about displaying the contents embedded in the document.

Example  “Example 1: Creating a Table of Contents” on page 823

(TROWD='rtf_control_string' | 'OFF')
inserts raw RTF specifications directly into header descriptions of the table row.

- rtf_control_string
  specifies RTF control words and symbols.
- OFF
  RTF controls are no longer inserted.
- Alias NO
  Requirement You must enclose all values in quotation marks.
  Tip If the RTF code inserted in the document is invalid, the code is either ignored or causes the document to be unusable.
  Example  “Example 4: Column Heading Rotation Using the TRHDR, TROWHDRCELL, and TROWD Options” on page 830

(TRHDR='rtf_control_string' | 'OFF')
inserts raw table row RTF specifications directly into the header description of the table row.

- rtf_control_string
  specifies Microsoft RTF control words or symbols.
- OFF
  RTF controls are no longer inserted.
- Alias NO
  Requirement You must enclose all values in quotation marks.
  Tip If the RTF code inserted in the document is invalid, the code is either ignored or causes the document to be unusable.
  Example  “Example 4: Column Heading Rotation Using the TRHDR, TROWHDRCELL, and TROWD Options” on page 830

(TROWHDRCELL='text_string' | 'OFF')
iinserts raw text into the table row cells. If the RTF Reader does not recognize this text_string, it applies the raw text to the location where the RTF is being written in the documentation. Otherwise, the RTF Reader interprets the text_string as RTF control words.
text_string
   any text specified.

OFF
   inserts a null string. Text is no longer inserted.
   Alias  NO
   Requirement You must enclose all values in quotation marks.
See
Example
   “Example 4: Column Heading Rotation Using the TRHDR, TROWHDRCELL, and TROWD Options” on page 830

(VSPACE=’ON’ | ’OFF’)
specifies whether to add or remove space before and after tables.

Note: The Parskip style element can be used to change the spacing before and after tables. This suboption overrides the spacing controlled by the Parskip style element.

OFF
   specifies that no space is added before and after tables in the RTF file.
   Alias  NO
ON
   specifies that space is added before and after tables in the RTF file.
   Alias  YES
   Default  YES or ON
   Requirement You must enclose all values in quotation marks.
Example
   ods tagsets.rtf file='myrtf.rtf' OPTIONS(VSPACE=’NO’);
Example
   “Example 7: Spacing Using Parskip Style Element and VSPACE= Suboption” on page 844

(WATERMARK=’"text_string"’ | ‘’ )
inserts a watermark that is displayed diagonally across each page of the RTF document.

   text_string
       specifies the text string that appears diagonally across each page of the RTF document.

   ‘’
       turns off the watermark text. Use a blank character within the quotation marks.

NO | OFF
appear as watermark text.
Restriction To turn off the watermark, you must use a blank space within the quotation marks. Empty quotation marks do not turn off the watermark.

Requirement WATERMARK='text_string' must be enclosed in parentheses.

Tip You can also use the BACKGROUNDIMAGE= style attribute to apply a watermark image to the RTF document.

PACKAGE <package-name>

specifies that the output from the destination be added to a package.

package-name specifies the name of a package that was created with the ODS PACKAGE statement. If no name is specified, then the output is added to the unnamed package that was opened last.

See “ODS PACKAGE Statement” on page 554

PAGEPANELS= n | NONE

specifies the number of panels permitted per page before ODS inserts a page break.

n specifies a positive integer.

Default 0

Tip Setting the value to 0 resets the action to the default action.

NONE specifies that paneling is handled the same ways as traditional ODS RTF. That is, all of the first panel is written, then all of the second panel, and so on, until all of the table information is written.

Defaults If you do not specify paneling, ODS tries to fit the full set of panels on a single page. ODS measures the width of the text and tables (horizontal measurement) and determines what the column widths should be. ODS then divides the page into panels if it is too wide to fit on a page.

ODS always determines the column widths and determines whether panels are required. When there are multiple panels, ODS attempts to place a reasonable number of rows in each panel.

Example “Example 5: Paneling Using the TABLEROWS and PAGEPANELS Options” on page 832

PATH= 'aggregate-file-storage-specification' | fileref | libref.catalog (URL= 'Uniform-Resource-Locator' | NONE)

specifies the location of an aggregate storage location or a SAS catalog for all RTF files. If the GPATH= option is not specified, all graphics output files are written to the "aggregate-file-storage-specification" or libref.

'aggregate-file-storage-location'
specifies an aggregate storage location such as directory, folder, or partitioned data set.

Requirement You must enclose aggregate-file-storage-location in quotation marks.
fileref
is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.

Interaction If you use a fileref in the PATH= option, then ODS does not use information from PATH= when it constructs links.


libref.catalog
specifies a SAS catalog to write to.

See “LIBNAME Statement” in SAS Statements: Reference.

URL= 'Uniform-Resource-Locator' | NONE
specifies a URL for the file-specification.

Uniform-Resource-Locator
is the URL that you specify. ODS uses this URL instead of the filename in all the links and references that it creates to the file.

NONE
specifies that no information from the PATH= option appears in the links or references.

Tip This option is useful for building output files that can be moved from one location to another. The links from the contents and page files must be constructed with a single-name URL, and the contents, page, and body files must be in the same location.

Interaction If you use the BODY= or FILE= external file option in conjunction with the PATH= option, the external file specification should not include path information.

PREPAGE= 'text-string'
specifies a text string that occurs before a table on a page.

text-string
is the text at the top of the table, after the titles. The text is placed before any tables created by the procedure.

Requirement You must enclose text-string in quotation marks.

RECORD_SEPARATOR= 'alternative-separator' | NONE
specifies an alternative record separator. This separator is a character or string that separates lines in the output files.

Different operating environments use different separator characters. If you do not specify a record separator, ODS formats the RTF files for the environment in which you run the SAS job. However, if you are generating files in one operating environment to view in another operating environment that uses a different separator character, you can specify a record separator that is appropriate for the target environment.

alternative-separator
represents one or more characters in hexadecimal or ASCII format. For example, the following option specifies a record separator of a carriage-return character and a linefeed character (on an ASCII file system):

RECORD_SEPARATOR= '0D0A'x
Operating Environment Information
In a mainframe environment, the following option specifies a record separator for a carriage-return character and a linefeed character for use with an ASCII file system:

RECORD_SEPARATOR= '0D25'x

Requirement You must enclose alternative-separator in quotation marks.

NONE produces RTF output that is appropriate for the environment in which you run the SAS job.

Operating Environment Information
In many operating environments, using a value of NONE has the same result as omitting the RECORD_SEPARATOR option.

Operating Environment Information
In a mainframe environment, by default, ODS produces a binary file that contains embedded record-separator characters. This approach means that the file is not restricted by the line-length restrictions on ASCII files. However, this also means that the lines are concatenated if you view the file in an editor. If you want to format the RTF files in a manner that enables you to read them with an editor, use RECORD_SEPARATOR= NONE. In this case, ODS writes one line of RTF at a time to the file. When you use a value of NONE, the logical record length of the file to which you are writing must be at least as long as the longest line that ODS produces. Otherwise, RTF might wrap to another line at an inappropriate place.

Aliases
RECSEP=

RS=

STARTPAGE= BYGROUP | YES | NO | NOW
controls page breaks.

BYGROUP
specifies to insert page breaks after each BY group.

YES
inserts a new page at the start of each procedure and within certain procedures, as is requested by the procedure code.

Alias
ON

Interactions
When the STARTPAGE= option is set to YES (the default), ODS inserts a new page at the start of each procedure. ODS relies on Word for the correct placement of headers and footers before and after the procedures.

Note that when you specify the BODYTITLE option, Microsoft Word no longer controls the placement of the headers and footers text. However, Word still controls other header and footer information, such as page number and date.

NO
instructs ODS not to insert any new pages at the start of each procedure or within certain procedures, even if the procedure code requests new pages. A new page begins only when a page is filled or when you specify STARTPAGE= NOW.
NEVER

**Interaction**
When you specify the `COLUMNS=` option, the `STARTPAGE=NO` option is not honored.

**NOW**
forces the immediate insertion of a new page.

**Tip** This option is useful primarily when the current value of the `STARTPAGE=NO` option is NO. Otherwise, each new procedure forces a new page automatically.

**Default** YES

**Tip** Specifying `STARTPAGE=NO` prevents forced page breaks. You can turn on forced page breaking again by specifying `STARTPAGE=YES`. You can insert a page break at any time by specifying `STARTPAGE=NOW`.

**STYLE=** `style-template`
specifies the style template for ODS to use to write the RTF files.

**style-template**
describes how to display the presentation aspects (color, font face, font size, and so on) of your SAS output. A style template determines the overall appearance of the documents that use that style template. Each style template consists of style elements.

**Note** If you are using SAS Studio, you do not need to specify the `STYLE=` option. You can go to Preferences ⇒ Results and change the style from the drop-down list for your selected destination.

**See** For a complete discussion of style templates, see “TEMPLATE Procedure: Creating a Style Template” in SAS Output Delivery System: Procedures Guide.

**Default** If you do not specify a style template, ODS uses the file that is specified in the SAS registry subkey: `ODS ⇒ DESTINATIONS ⇒ RTF`. By default, this value specifies `RTF` for traditional RTF and Measured RTF.

**TABLEROWS=** `n`
specifies the number of rows in each table before ODS inserts a page break. If the table is narrow enough to fit on a page, `n` lines are written to the table before a page break. If the table is too wide for a page, the page is broken into panels. In each panel, `n` rows are written. When all the panels for `n` rows have been written, a page break is inserted before the next group of panels is written.

**Note**: Page breaks are not forced between panels.

`n` is a positive integer.

**Alias** 0 | NONE

**Default** Allow SAS to determine the number of rows per table.

**Tip** 0 or NONE returns to the default, which allows SAS to determine the number of rows per table.
Example  “Example 5: Paneling Using the TABLEROWS and PAGEPANELS Options” on page 832

TEXT= 'text-string'

inserts text into your RTF output.

text-string

is the text that you want to insert into your RTF output. You can also use TEXT= to annotate other output.

Requirement  You must enclose a text-string in quotation marks.

TRANTAB= translation-table

specifies the translation table for ODS to use when it transcodes a file for output.

See  For more information, see “TRANTAB= System Option” in SAS National Language Support (NLS): Reference Guide.

UNIFORM

ensures uniformity from page to page within a single table that requires multiple pages. When the UNIFORM option is in effect, ODS reads the entire table first, and determines the column widths that are necessary to accommodate all of the data. ODS applies these column widths to all pages of a multiple page table.

Note:  With BY-group processing, SAS writes the results of each BY group to a separate table, so the output might not be uniform across BY groups.

Default  If you do not specify the UNIFORM option, ODS prints a table one page at a time. This approach ensures that SAS does not run out of memory while it processes very large tables. However, column widths might vary from one page to the next.

Tips  After this option is turned on, you cannot turn it off for that SAS session.

The UNIFORM option can cause SAS to run out of memory if you are printing a very large table. If this happens, you can specify the width of each of the columns in the table. Then print the table one page at a time. To do so, you must edit the table template that you use. For more information, see “What You Can Do with Table Templates” in SAS Output Delivery System: Procedures Guide.

Example  “Example 6: Repeating Headers Using the UNIFORM Option” on page 837

Diagnostic Tagsets

The following diagnostic tagsets are available for the ODS TAGSET.RTF statement.

MEAS_EVENT_MAP

creates XML output that shows which events are being triggered and which variables are used by an event to send output from a SAS process to an output file. When you run a SAS process with MEAS_EVENT_MAP, ODS writes XML to an output file that shows all event names and variable names as tags. The output helps you create your own tagsets.

MEAS_SHORT_MAP

creates a subset of the XML output that is created by the MEAS_EVENT_MAP tagset.
MEAS_TEXT_MAP
creates text output that shows which events are being triggered as ODS handles the output objects.

Tip You can use the MEAS_TEXT_MAP output as an alternative to the output that is created by the MEAS_EVENT_MAP tagset.

Details

Opening and Closing the ODS TAGSETS.RTF Destination
You can modify and open an RTF destination with many ODS TAGSETS.RTF options. However, the FILE= option automatically closes the open destination that is referred to by the ODS TAGSETS.RTF statement. The option also closes any files associated with it and opens a new instance of the destination. If you use one of the ODS TAGSETS.RTF options, you should close the destination yourself.

Understanding How Traditional RTF Formats Output
RTF produces output for Microsoft Word. Although other applications can read RTF files, the RTF output might not work successfully with them.

The RTF destination enables you to view and edit the RTF output. ODS does not define the “vertical measurement,” which means that SAS does not determine the optimal place to position each item on the page. For example, page breaks are not always fixed because you do not want your RTF output tables to split at inappropriate places. Your tables can remain intact on one page, or can have logical breaks where you specify.

Microsoft Word needs to know the widths of table columns, and it cannot adjust tables if they are too wide for the page. However, ODS measures the width of the text and tables (horizontal measurement). Therefore, all the column widths can be set properly by SAS, and the table can be divided into panels if it is too wide to fit on a single page.

In short, when producing RTF output for input to Microsoft Word, SAS determines the horizontal measurement and Microsoft Word controls the vertical measurement. Because Microsoft Word can determine how much room there is on the page, your tables will be displayed consistently even after you modify your RTF file.

Note: The creation of complex tables that contain a large number of observations can reduce system efficiencies and increase processing time.

ODS Measured RTF versus Traditional ODS RTF
The ODS TAGSETS.RTF is also referred to as the measured tagset. This tagset enables users to specify how and where page breaks occur and when to place titles and footnotes into the body of a page. Traditional ODS RTF relies on Microsoft Word to make implicit page breaks for tables that are too long to fit on a single page. Traditional RTF also places titles and footnotes in the RTF instructions that enable Microsoft Word to apply them to pages as they are needed. In contrast, the RTF tagset (TAGSETS.RTF) enables SAS to place titles and footnotes into the body of the document so that it is outside of the control of Microsoft Word. Therefore, SAS becomes responsible for the implicit page breaks.

RTF Tagset Features

Overview of the RTF Tagset Features
The measured RTF tagset does the following:

• controls page breaks on very large tables
• supports RTF readers other than Microsoft Word
• controls titles, footnotes, and other page elements

**Controlling Page Breaks in Long Tables**

Multiple-page tables can be a problem for ODS RTF. Like the ODS PRINTER destinations, SAS determines where to wrap a wide table. But for a long table, the entire table is loaded into memory before being rendered. When tables become longer than a physical page, Microsoft Word determines the page break. Microsoft word re-creates the column heading information in the table and applies titles and footnotes as needed. If a table is later edited in Microsoft Word, the information remains valid.

Unfortunately, a lot of information is associated with each cell of a table. No matter how much memory is added to the system, it is possible to create a table that can exceed it. Furthermore, an exhausted memory condition cannot be anticipated because it varies with the machine setup and with the table that you are creating.

However, with the ODS RTF tagset, SAS determines where to break the page and puts the titles and footnotes in the body of the document. When the table is broken into pages and SAS controls the page breaks, approximately a page of data is needed in memory at any one time. Therefore, a much smaller memory footprint is consumed and extremely large tables can be created. The ODS RTF tagset accommodates users who need large tables and users who want the old style RTF behavior. Both RTF implementations can be supported simultaneously.

**Controlling Titles, Footnotes, and Other Page Elements**

Measured RTF uses a tagset that places the titles and footnotes on the page as tables instead of as RTF control words that are passed to Microsoft Word. With traditional RTF, the titles and footnotes are placed in the RTF header and footer information unless you specify the BODYTITLE option. Because the headers and footers are automatically placed in the body of the document with measured RTF, the TAGSET.RTF destination does not need the BODYTITLE option.

**Measured RTF and Graphics**

Measured RTF produces output in rich text format, which supports three formats for graphics that Microsoft Word can read.

<table>
<thead>
<tr>
<th>Format for Graphics</th>
<th>Corresponding SAS Graphics Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>emfblips</td>
<td>EMF</td>
</tr>
<tr>
<td>pngblips</td>
<td>PNG</td>
</tr>
<tr>
<td>jpegblips</td>
<td>JPEG</td>
</tr>
</tbody>
</table>

When you do not specify a target device, the default target is EMF. You can also use the ACTIVEX, ACTXIMG, JAVA IMG graphics drivers to generate graphics in your measured RTF documents. The ACTIVEX driver generates an ActiveX control. The ACTXIMG and JAVA IMG drivers generate PNG files with the ACTIVEX Control or JAVA Applets appropriately. For more information about graphics devices, see SAS/GRAPH: Reference.

**Note:** When you specify the JAVA device in the ODS TAGSET.RTF statement, the JAVA IMG driver is used.
Note: You cannot use UTF-8 encoding with the ACTIVEX device in RTF. When UTF-8 encoding is used, the ACTXIMG (activex image) device is used.

Examples

**Example 1: Creating a Table of Contents**

**Features:**
- ODS TAGSETS.RTF statement:
  - Action: CLOSE
  - Option: CONTENTS
  - Option: TOC_DATA

**Other features:**
- OPTIONS statement
- PROC FORMAT
- PROC PRINT
- PROC SORT
- PROC REPORT
- PROC TABULATE

**Data set:**
- Grain_Production

**Format:**
- $CNTRY.

The following example creates a table of contents page that contains embedded table of contents data for some procedures, but not for others. The insertion of the table of contents data can be turned on and off in the middle of a program.

**Program**

```r
ods html close;
proc sort data=Grain_Production;
  by year country type;
run;
ods tagsets.rtf file='Grain_Tagset.rtf' options(contents='yes' toc_data='yes');
options nobyline;
title 'Leading Grain-Producing Countries';
title2 'for #byval(year)';
proc report data=Grain_Production nowindows;
  by year;
  column country type kilotons;
  define country / group width=14 format=$cntry.;
  define type / group 'Type of Grain';
  define kilotons / format=comma12.;
  footnote 'Measurements are in metric tons.';
run;
options byline;
title2;
ods tagsets.rtf options(toc_data='no');
proc tabulate data=Grain_Production format=comma12.;
  class year country type;
```
Program Description

Close the HTML destination. The HTML destination is open by default. The ODS HTML statement closes the HTML destination to conserve resources.

ods html close;

Sort the data set Grain_Production. PROC SORT sorts the data, first by values of the variable Year, then by values of the variable Country, and finally by values of the variable Type.

proc sort data=Grain_Production;
  by year country type;
run;

Create RTF output and create a new body file for each page of output. The ODS TAGSETS.RTF statement opens the RTF destination and creates RTF output. The CONTENTS suboption creates a table of contents page. The table of contents page contains a table of contents field that puts all of the contents information that is embedded in the document into a table of contents. This action occurs only if the TOC_DATA suboption is specified along with the CONTENTS suboption. The table of contents information is not embedded by default into the RTF file. You can turn on the insertion of the TOC data by specifying TOC_DATA='YES' or instruct ODS to not insert this information by specifying TOC_DATA='NO'.

ods tagsets.rtf file='Grain_Tagset.rtf' options(contents='yes' toc_data='yes');

Suppress the default BY line and specify a new value into the BY line. The NOBYLINE option suppresses the default BY line variable. The #BYVAL parameter specification inserts the current value of the BY variable Year into the title.

options nobyline;
  title 'Leading Grain-Producing Countries';
  title2 'for #byval(year)';

Produce a report. This PROC REPORT step produces a report on grain production. Each BY group produces a page of output. ODS creates a new body file for each BY group. The NOWINDOWS option instructs ODS to run PROC REPORT without the REPORT window and sends its output to the open output destinations.

proc report data=Grain_Production nowindows;
  by year;
  column country type kilotons;

define country / group width=14 format=$cntry.;
define type     / group 'Type of Grain';
define kilotons / format=comma12.;
footnote 'Measurements are in metric tons.';
run;

**Restore the default BY line and clear the second TITLE statement.** The BYLINE option restores the default BY line. The TITLE2 statement clears the second TITLE statement.

```plaintext
options byline;
title2;
```

**Suppress the insertion of table of contents data into the RTF file.** The TOC_DATA='NO' option instructs ODS not to insert the table of contents data into the RTF file. Therefore, because the TABULATE procedure follows the TOC_DATA='NO' option, there is no entry for the TABULATE procedure in the table of contents page.

```plaintext
ods tagsets.rtf options(toc_data='no');
```

**Produce a report.** The TABLE statement in the PROC TABULATE step uses three dimensions. Year defines pages, Country and Type define the rows, and Kilotons defines the columns. Therefore, PROC TABULATE explicitly produces one page of output for 1995 and one page for 1996 based on the years specified in the Grain_Production data set. ODS also starts a new body file for each page.

```plaintext
proc tabulate data=Grain_Production format=comma12.;
class year country type;
var kilotons;
table year,
country*type,
kilotons*sum=' ' / box=_page_ misstext='No data';
    format country $cntry.;
    footnote 'Measurements are in metric tons.';
run;
```

**Enable the insertion of table of contents data into the RTF file.** The TOC_DATA='YES' option instructs ODS to insert the table of contents data into the RTF file. There is an entry for the PRINT procedure in the table of contents page when the PROC PRINT statement is executed.

```plaintext
ods tagsets.rtf options(toc_data='yes');
```

**Print the Grain_Production data set.**

```plaintext
proc print data=Grain_Production;
run;
```

**Close the RTF destination.** The ODS TAGSETS.RTF CLOSE statement closes the RTF destination and all the files that are associated with it. If you do not close the destination, then you cannot view the files in a browser window. Open the HTML destination to return ODS to its default setting.

```plaintext
ods tagsets.rtf close;
ods html;
```
RTF Output

By default the table of contents is collapsed on the table of contents page. To expand the table of contents from Microsoft Word, right-click beneath the title until the **Update Field** option is shown in the selection list. Then select **Update Field**.

The table of contents contains entries for PROC REPORT and PROC PRINT only. By default, ODS does not embed the table of contents data in the RTF document until you specify the TOC_DATA=’YES’ option, which results in an entry for PROC REPORT and all other data. If you turn off the TOC_DATA option before the TABULATE procedure, ODS does not insert information into the RTF document for PROC TABULATE. No other contents information is inserted into the RTF document until you specify TOC_DATA=’YES’. In this example, the TOC_DATA=’YES’ option is specified before
the PRINT procedure. Therefore, ODS inserts contents data for PROC PRINT into the table of contents.

**Example 2: Using the DOC Suboption to Get ODS TAGSETS.RTF Information**

**Features:**
- ODS TAGSETS.RTF statement:
  - Action: CLOSE
  - Option: OPTIONS (DOC="HELP")
  - Option: FILE=

**Other features:**
- PROC PRINT

The following example is printed to the SAS log the OPTIONS suboptions and a description of each available suboption.

**Program**

```sas
ods tagsets.rtf file='Help.rtf' options (doc='help');
proc print data=Sashelp.Class;
run;
ods tagsets.rtf close;
```

**Program Description**

Print information about the OPTIONS suboptions to the SAS log file. Specifying the OPTIONS suboption (DOC="HELP") prints Help for the ODS TAGSETS.RTF statement suboptions to the SAS log file. The FILE= option prints the data results to an RTF file named Help.rtf.
ods tagsets.rtf file='Help.rtf' options (doc='help');

Print the data set Sashelp.Class. The PROC PRINT statement prints the Sashelp.Class data set.

    proc print data=Sashelp.Class;
    run;

Close the TAGSETS.RTF destination. If you do not close this destination, then you cannot view the output.

    ods tagsets.rtf close;

Output

Specify the “DOC=’HELP’ suboption to print all of the OPTIONS suboptions and information about each of the suboptions to the SAS log.

Example 3: Using the TABLES_OFF Suboption

Features:

ODS TAGSETS.RTF statement:

    Action: CLOSE
    Option: OPTIONS (TABLES_OFF="OFF") (TABLES_OFF="USERTEXT")
              (TABLES_OFF="STYLES_ELEMENTS")
Example 3: Using the TABLES_OFF Suboption

Option: FILE=
Option: TEXT=

Other features:
PROC PRINT

The following example turns on and off the tables in RTF output and applies the style element specified by the TABLES_OFF suboption.

Program

ods tagsets.rtf file='tablesOff.rtf' options(TABLES_OFF='STYLE_ELEMENTS');
proc print data=sashelp.class(obs=1) ;
run;
ods tagsets.rtf text='TEXT is placed in a table by default' ;
ods tagsets.rtf options(TABLES_OFF='usertext' );
ods tagsets.rtf text='TEXT is not placed in a table (table is removed when style element is specified)' ;
ods tagsets.rtf options(TABLES_OFF='off' );
ods tagsets.rtf text='TEXT is placed in a table (returned to default when tables_off is set to off)' ;
ods tagsets.rtf close;

Program Description

List the style elements that can be applied in the SAS log file. ODS TAGSETS.RTF enables you to apply a style element to the RTF output. To determine the style elements that you can use, list them by specifying the TABLES_OFF suboption. This information is written to the SAS log. Notice that you can use different style elements with each statement.

ods tagsets.rtf file='tablesOff.rtf' options(TABLES_OFF='STYLE_ELEMENTS');
proc print data=sashelp.class(obs=1) ;
run;
ods tagsets.rtf text='TEXT is placed in a table by default' ;

Turn off tables and apply the USERTEXT style element. Specifying TABLES_OFF='USERTEXT' applies the USERTEXT style to the text being written.

ods tagsets.rtf options(TABLES_OFF='usertext' );
ods tagsets.rtf text='TEXT is not placed in a table (table is removed when style element is specified)' ;

Return to the default, tables are on. Specifying TABLES_OFF='OFF', returns the option to the default and turns the tables back on.

ods tagsets.rtf options(TABLES_OFF='off' );
ods tagsets.rtf text='TEXT is placed in a table (returned to default when tables_off is set to off)' ;

Close the RTF destination. If you do not close this destination, then you cannot view the output.

ods tagsets.rtf close;
RTF Output

If you specify the ODS TAGSETS.RTF suboption, TABLES_OFF= style_element lists the style elements that are being used and are written to the SAS log.

The following output illustrates what happens when the TABLES_OFF suboption is used. In this example, ODS places the output text in a table by default. Specifying TABLES_OFF='USERTEXT' turns off the table and applies the USERTEXT style to the output. Lastly, TABLES_OFF='OFF' is specified, which causes the text to be written in a table.

Example 4: Column Heading Rotation Using the TRHDR, TROWHDRCELL, and TROWD Options

Features:

ODS TAGSETS.RTF statement:
Action: CLOSE
Option: OPTIONS TRHDR= TROWHDRCELL= TROWD=

Other features:
PROC PRINT
OPTIONS statement

The following example creates an RTF file in which the headers and contents of the row and column headings are rotated within the table.

Program

options orientation=landscape nodate nonumber;

ods html close;
ods tagsets.rtf file='Mrotate.rtf'
OPTIONS (TRHDR='\trrh750'
TROWHDRCELL='\cltxbtlr'
TROWD='\rtlrow');
proc print data=Sashelp.Class(obs=5);
run;
ods tagsets.rtf close;
ods html;

Program Description

Specify the orientation of the page. The ORIENTATION option sets the page to landscape. The NODATE option turns off the output of the date and time. The NONUMBER option tells SAS not to print the page number on the first title line of each page of output.

options orientation=landscape nodate nonumber;

Close the HTML destination. The HTML destination is open by default. The ODS HTML statement closes the HTML destination to conserve resources.

ods html close;

Create RTF output using the ODS TAGSETS.RTF statement and rotate the rows and header information in the table. The ODS TAGSETS.RTF statement opens the RTF destination and creates RTF output that is sent to the Mrotate.rtf file. The three options enable you to manipulate the row and header descriptions. TRHDR enables change to the table row headers. In this example, the RTF string that is specified adds more space to the row headers. TROWHDRCELL enables you to manipulate the table-row cell information. In this case, the information is rotated to vertical. The TROWD option enables you to change the table row description. The RTF string specified changes the first table row to the rightmost row.

ods tagsets.rtf file='Mrotate.rtf'
OPTIONS (TRHDR='\trrh750'
TROWHDRCELL='\cltxbtlr'
TROWD='\rtlrow');

Print the Sashelp.Class data set.
proc print data=Sashelp.Class(obs=5);
 run;

Close the TAGSETS.RTF destination. If you do not close this destination, then you cannot view the output. Open the HTML destination to return ODS to its default setting.

ods tagsets.rtf close;
ods html;

RTF Output

The Mrotate.rtf output shows how ODS has rotated the first row of the table to the rightmost column. ODS added more space to the row headers and made the cell contents of the header row vertical. This table manipulation was caused by using the TRHDR=, TROWHDCELL=, and TROWD= suboptions of OPTIONS.

Example 5: Paneling Using the TABLEROWS and PAGEPANELS Options

Features:
- ODS TAGSETS.RTF statement
  - CLOSE action
  - TABLEROWS option
  - PAGEPANELS option

Other features:
- OPTIONS statement
- PROC PRINT
- DATA statement
The following program provides an example of how ODS creates panels when a table is wider than a page and presents options for controlling the paneling.

**Program**

```sas
option nodate nonumber;
ods html close;
ods tagsets.rtf file='Panel.rtf';
data temp;
array values val1-val50;
do j = 1 to 6;
   do i = 1 to dim(values);
      values(i) = i;
   end;
   output;
end;
run;
ods tagsets.rtf;
title Default Paneling;
proc print data=Temp;
run;
ods tagsets.rtf tablerows=5 pagepanels=2;
title 'Paneling with TABLEROWS=5 and PAGEPANELS=2';
proc print data=Temp;
run;
ods tagsets.rtf close;
ods html;
```

**Program Description**

**Specify the system options.** The NODATE option turns off the output of the date and time. The NONUMBER option tells SAS not to print the page number on the first title line of each page of output.

```
option nodate nonumber;
```

**Close the HTML destination.** The HTML destination is open by default. The ODS HTML statement closes the HTML destination to conserve resources.

```
ods html close;
```

**Open the RTF and file destination.** Open the RTF destination and name the output file Panel.rtf. If you do not specify a filename, the output filename defaults to Sasmeas.rtf.

```
ods tagsets.rtf file='Panel.rtf';
```

**Produce a large data set.** Create a large data set in order to show how paneling works.

```
data temp;
array values val1-val50;
do j = 1 to 6;
   do i = 1 to dim(values);
      values(i) = i;
   end;
end;
```
Create RTF output that uses the default paneling. The ODS TAGSETS.RTF statement opens the RTF destination and creates RTF output. Default paneling is used to print the TEMP data set that was created earlier in this program. The title of the table is “Default Paneling”.

```
ods tagsets.rtf;
title Default Paneling;
proc print data=Temp;
run;
```

Create RTF output where the number of panels is specified. The ODS TAGSETS.RTF statement opens the RTF destination and creates RTF output. RTF tagset options TABLEROWS and PAGEPANELS enable you to control the number of panels on a page and the number of rows of data that you want output for each table. The title of this multi-paneled table is 'Paneling with TABLEROWS=5 and PAGEPANELS=4';.

```
ods tagsets.rtf tablerows=5 pagepanels=2;
title 'Paneling with TABLEROWS=5 and PAGEPANELS=2';
proc print data=Temp;
run;
```

Close the RTF destination. If you do not close this destination, then you cannot view the output. Open the HTML destination to return ODS to its default setting.

```
ods tagsets.rtf close;
ods html;
```

RTF Output

Page paneling occurs when a table is wider than a page. By default in measured ODS RTF, panels are grouped together so that all observations are close together. The first
column holds as many columns as can fit on one line. The number of rows in each panel is determined by the number that fit on a logical page.

**Output 6.94** RTF Output with Default Page Paneling - Page 1

**Default Paneling**

| Obs | val11 | val12 | val13 | val14 | val15 | val16 | val17 | val18 | val19 | val20 | val21 | val22 | val23 | val24 | val25 | val26 | val27 | val28 | val29 | val30 | val31 | val32 | val33 | val34 | val35 |
|-----|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1   | 1     | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   | 16   | 17   | 18   | 19   | 20   | 21   | 22   | 23   | 24   | 25   | 26   |
| 2   | 1     | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   | 16   | 17   | 18   | 19   | 20   | 21   | 22   | 23   | 24   | 25   | 26   | 27   |
| 3   | 1     | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   | 16   | 17   | 18   | 19   | 20   | 21   | 22   | 23   | 24   | 25   | 26   | 27   | 28   |
| 4   | 1     | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   | 16   | 17   | 18   | 19   | 20   | 21   | 22   | 23   | 24   | 25   | 26   | 27   | 28   |
| 5   | 1     | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   | 16   | 17   | 18   | 19   | 20   | 21   | 22   | 23   | 24   | 25   | 26   | 27   | 28   |
| 6   | 1     | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   | 16   | 17   | 18   | 19   | 20   | 21   | 22   | 23   | 24   | 25   | 26   | 27   | 28   |

The TABLEROWS option enables you to specify the number of rows that are output for each panel. In this example, five rows are specified, which causes each panel output to contain a maximum of five rows. PAGEPANELS=2 then causes only two panels to be
output onto each page. What is generated is four pages of output with a maximum of two panels per page where each table contains a maximum of five rows each.

**Output 6.95**  RTF Output with Options PAGEPANELS and TABLEROWS

### Paneling with TABLEROWS=5 and PAGEPANELS=2

<table>
<thead>
<tr>
<th>Obs</th>
<th>val1</th>
<th>val2</th>
<th>val3</th>
<th>val4</th>
<th>val5</th>
<th>val6</th>
<th>val7</th>
<th>val8</th>
<th>val9</th>
<th>val10</th>
<th>val11</th>
<th>val12</th>
<th>val13</th>
<th>val14</th>
<th>val15</th>
<th>val16</th>
<th>val17</th>
<th>val18</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>val19</th>
<th>val20</th>
<th>val21</th>
<th>val22</th>
<th>val23</th>
<th>val24</th>
<th>val25</th>
<th>val26</th>
<th>val27</th>
<th>val28</th>
<th>val29</th>
<th>val30</th>
<th>val31</th>
<th>val32</th>
<th>val33</th>
<th>val34</th>
<th>val35</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>2</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
</tbody>
</table>

(Continued)

### Paneling with TABLEROWS=5 and PAGEPANELS=2

<table>
<thead>
<tr>
<th>Obs</th>
<th>val36</th>
<th>val37</th>
<th>val38</th>
<th>val39</th>
<th>val40</th>
<th>val41</th>
<th>val42</th>
<th>val43</th>
<th>val44</th>
<th>val45</th>
<th>val46</th>
<th>val47</th>
<th>val48</th>
<th>val49</th>
<th>val50</th>
<th>j</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>1</td>
<td>51</td>
</tr>
<tr>
<td>2</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>2</td>
<td>51</td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>3</td>
<td>51</td>
</tr>
<tr>
<td>4</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>4</td>
<td>51</td>
</tr>
<tr>
<td>5</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>5</td>
<td>51</td>
</tr>
</tbody>
</table>

### Paneling with TABLEROWS=5 and PAGEPANELS=2

| Obs | val1 | val2 | val3 | val4 | val5 | val6 | val7 | val8 | val9 | val10 | val11 | val12 | val13 | val14 | val15 | val16 | val17 | val18 | val19 | val20 | val21 | val22 | val23 | val24 | val25 | val26 | val27 | val28 | val29 | val30 | val31 | val32 | val33 | val34 | val35 |
|-----|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 6   | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10    | 11    | 12    | 13    | 14    | 15    | 16    | 17    | 18    | 19    | 20    | 21    | 22    | 23    | 24    | 25    | 26    | 27    | 28    | 29    | 30    | 31    | 32    | 33    | 34    | 35    |

<table>
<thead>
<tr>
<th>Obs</th>
<th>val19</th>
<th>val20</th>
<th>val21</th>
<th>val22</th>
<th>val23</th>
<th>val24</th>
<th>val25</th>
<th>val26</th>
<th>val27</th>
<th>val28</th>
<th>val29</th>
<th>val30</th>
<th>val31</th>
<th>val32</th>
<th>val33</th>
<th>val34</th>
<th>val35</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
</tbody>
</table>
Example 6: Repeating Headers Using the UNIFORM Option

Features:

ODS TAGSETS.RTF statement:
   Action: CLOSE
   Option: UNIFORM
   Option: FILE=

ODS RTF statement

Other features:
   OPTIONS statement
   PROC FORMAT
   PROC TABULATE
   DATA statement

The following example creates a multi-page table that is uniform across several pages. The row and column heading labels are also carried over to each page.

Program

ods html close;

options orientation=landscape;
ods rtf file='RtfTab.rtf';
ods tagsets.rtf file='MrtfTab.rtf' uniform;

data one;
   do a=1 to 2;
      do b=1 to 2;
         do c=1 to 3;
            do d=1 to 3;
               do e=1 to 5;
                  output;
               end;
            end;
         end;
   end;
run;

proc format;
   value cars 1='DATSUN 200SX'
      2='PONTIAC FIERO';
   value colors 1='RED'
       2='LIGHT BLUE'
       3='YELLOW'
       4='GREEN';

Paneling with TABLEROWS=5 and PAGEPANELS=2

| Obs | val36 | val37 | val38 | val39 | val40 | val41 | val42 | val43 | val44 | val45 | val46 | val47 | val48 | val49 | val50 | j   | i   |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|----|
| 6   | 36    | 37    | 38    | 39    | 40    | 41    | 42    | 43    | 44    | 45    | 46    | 47    | 48    | 49    | 50   | 6   | 51  |
5='BROWN';
value luxury 1='ALL THE WAY'
             2='STANDARD OPTIONS'
             3='STRIPPED DOWN';
value opts  1='POWER STEERING'
             2='SUN ROOF'
             3='AUTOMATIC'
             4='T-TOP'
             5='HATCHBACK'
             6='FUEL-INJECTION'
             7='HUBCAPS'
             8='AM/FM STEREO'
             9='FLOOR MATS'
            10='CASSETTE PLAYER';
value perform 1='VERY SLOW'
               2='SLOW'
               3='AVERAGE'
               4='FAST'
               5='VERY FAST';
run;
data two (keep=model color luxury options perform);
length model color luxury options perform  $ 20;
set one;
model=put(a,cars.);
color=put(b,colors.);
luxury=put(c,luxury.);
options=put(d,opts.);
perform=put(e,perform.);
run;
title2 'My Favorite Cars';
title3 '(large data set)';
proc tabulate data=two order=data ;
class model color luxury options perform;
table model*color*luxury*options*perform,n / indent=4 condense;
label model='MODEL CAR'
       color='COLOR OF CAR'
       luxury='CONDITION OF CAR'
       perform='SPEED';
keylabel n='NUMBER';
run;
ods _all_ close;
ods html;

**Program Description**

**Close the HTML destination.** The HTML destination is open by default. The ODS HTML statement closes the HTML destination to conserve resources.

```
ods html close;
```

**Specify the orientation of the page and name the RTF output.** Specify landscape as the orientation of the page. Name the RTF output file to RtfTab.rtf.

```
options orientation=landscape;
```
Open the RTF file and create output that has UNIFORM header information. The ODS TAGSETS.RTF statement opens the RTF file. The UNIFORM option ensures that the column headings and header information appear on each page.

```sas
ods tagsets.rtf file='MrtfTab.rtf' uniform;
```

Create the data set One. Create a data set that has five columns. Each column consists of one to five subcolumns.

```sas
data one;
    do a=1 to 2;
        do b=1 to 2;
            do c=1 to 3;
                do d=1 to 3;
                    do e=1 to 5;
                        output;
                    end;
                end;
            end;
        end;
    end;
run;
```

Create user-defined formats. PROC FORMAT creates the formats that SAS uses in the columns and subcolumns of the table.

```sas
proc format;
    value cars 1='DATSUN 200SX'
        2='PONTIAC FIERO';
    value colors 1='RED'
        2='LIGHT BLUE'
        3='YELLOW'
        4='GREEN'
        5='BROWN';
    value luxury 1='ALL THE WAY'
        2='STANDARD OPTIONS'
        3='STRIPPED DOWN';
    value opts 1='POWER STEERING'
        2='SUN ROOF'
        3='AUTOMATIC'
        4='T-TOP'
        5='HATCHBACK'
        6='FUEL-INJECTION'
        7='HUBCAPS'
        8='AM/FM STEREO'
        9='FLOOR MATS'
        10='CASSETTE PLAYER';
    value perform 1='VERY SLOW'
        2='SLOW'
        3='AVERAGE'
        4='FAST'
        5='VERY FAST';
run;
```
Create data set Two. Data set Two populates the data set with the formats supplied by PROC FORMAT.

```sas
data two (keep=model color luxury options perform);
  length model color luxury options perform $ 20;
  set one;
  model=put(a,cars.);
  color=put(b,colors.);
  luxury=put(c,luxury.);
  options=put(d,opts.);
  perform=put(e,perform.);
run;
```

Create titles for the Output. Provide two titles for the output.

```sas
title2 'My Favorite Cars';
title3 '(large data set)';
```

Produce a report. PROC TABULATE creates the table of cars and their attributes.

```sas
proc tabulate data=two order=data ;
  class model color luxury options perform;
  table model*color*luxury*options*perform,n / indent=4 condense;
  label model='MODEL CAR'
    color='COLOR OF CAR'
    luxury='CONDITION OF CAR'
    perform='SPEED';
  keylabel n='NUMBER';
run;
```

Close all destinations. The ODS_ALL_CLOSE statement closes any open destinations and all of the files that are associated with them. If you do not close the destination, you cannot view the files in a browser window.

```sas
ods _all_ close;
ods html;
```

Output

The following output is from the measured RTF output file Mrtftab.rtf. This output is generated using the ODS TAGSETS.RTF statement. Note the differences between the measured output and the traditional RTF output. Note that the cell heading information is
Example 6: Repeating Headers Using the UNIFORM Option

carried to each page and that the word “Continued” appears at the bottom of each page of RTF output.

Output 6.96  Measured RTF Output

<table>
<thead>
<tr>
<th>DATSUN 280X</th>
<th>RED</th>
<th>ALL THE WAY</th>
<th>POWER STEERING</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>VERY SLOW</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SLOW</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AVERAGE</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FAST</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VERY FAST</td>
<td>1</td>
</tr>
<tr>
<td>SUN ROOF</td>
<td></td>
<td></td>
<td>VERY SLOW</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SLOW</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AVERAGE</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FAST</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VERY FAST</td>
<td>1</td>
</tr>
<tr>
<td>AUTOMATIC</td>
<td></td>
<td></td>
<td>VERY SLOW</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SLOW</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AVERAGE</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FAST</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VERY FAST</td>
<td>1</td>
</tr>
<tr>
<td>STANDARD OPTIONS</td>
<td></td>
<td></td>
<td>POWER STEERING</td>
<td>VERY SLOW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SLOW</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AVERAGE</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FAST</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VERY FAST</td>
<td>1</td>
</tr>
<tr>
<td>SUN ROOF</td>
<td></td>
<td></td>
<td>VERY SLOW</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SLOW</td>
<td>1</td>
</tr>
</tbody>
</table>

(Continued)
The following output is a portion of the Rtftab.rtf file that was generated using the traditional ODS RTF statement. Notice that header information is not carried over to
Example 6: Repeating Headers Using the UNIFORM Option

Output 6.97  Traditional RTF Output

<table>
<thead>
<tr>
<th>The SAS System</th>
<th>My Favorite Cars (large data set)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA: NISSAN</td>
<td>R3D</td>
</tr>
<tr>
<td></td>
<td>ALL THE WAY</td>
</tr>
<tr>
<td></td>
<td>POWER STEERING</td>
</tr>
<tr>
<td></td>
<td>NUMBER</td>
</tr>
<tr>
<td>DATSUN 205SX</td>
<td>SLOW</td>
</tr>
<tr>
<td></td>
<td>AVERAGE</td>
</tr>
<tr>
<td></td>
<td>FAST</td>
</tr>
<tr>
<td></td>
<td>VERY FAST</td>
</tr>
<tr>
<td>SUN ROOF</td>
<td>VERY SLOW</td>
</tr>
<tr>
<td></td>
<td>SLOW</td>
</tr>
<tr>
<td></td>
<td>AVERAGE</td>
</tr>
<tr>
<td></td>
<td>FAST</td>
</tr>
<tr>
<td></td>
<td>VERY FAST</td>
</tr>
<tr>
<td>AUTOMATIC</td>
<td>VERY SLOW</td>
</tr>
<tr>
<td></td>
<td>SLOW</td>
</tr>
<tr>
<td></td>
<td>AVERAGE</td>
</tr>
<tr>
<td></td>
<td>FAST</td>
</tr>
<tr>
<td></td>
<td>VERY FAST</td>
</tr>
</tbody>
</table>

Page two of the output. Also note that page one does not indicate that more pages of output follow.
Example 7: Spacing Using Parskip Style Element and VSPACE= Suboption

Features:
- PROC TEMPLATE
  - Parskip Style Element
- ODS TAGSETS.RTF statement
  - VSPACE= suboption
  - STYLE= option

Other features:
- OPTIONS statement
- PROC PRINT
- DATA statement

The following program provides an example of how to use the Parskip style element and the VSPACE= suboption to control the spacing before and after tables.

Program
```
option nodate;
ods html close;
proc template;
    define style test1;
    parent=styles.rtf;
```
Example 7: Spacing Using Parskip Style Element and VSPACE= Suboption

```sas
style parskip / fontsize = 40pt;
end;
run;

ods tagsets.rtf file="mparskip.rtf" style=test1;
title "Parskip Style Element Spacing between the Title and Table";
footnote "Parskip Style Element Spacing between the Table and Footnote";
proc print data=sashelp.class (obs=5); run;
ods tagsets.rtf style=test1 OPTIONS(vspace="NO");
title "Spacing between the Title and the Table Using VSPACE=No";
footnote "Spacing between the Table and Footnote Using VSPACE=No"
proc print data=sashelp.class (obs=5);
run;
ods tagsets.rtf close;
ods html;
```

**Program Description**

**Specify the system options.** The NODATE option turns off the output of the date and time.
```sas
option nodate;
```

**Close the HTML destination.** The HTML destination is open by default.
```sas
ods html close;
```

**Create the Styles Using PROC TEMPLATE** Use the default RTF styles and specify the parskip style element. This style element is used to determine the spacing before and after tables.
```sas
proc template;
  define style test1;
  parent=styles.rtf;
  style parskip / fontsize = 40pt;
end;
run;
```

**Open the RTF file destination and specify a style.** Open the RTF destination and name the output file mparskip.rtf. If you do not specify a filename, the output filename defaults to Sasmeas.rtf. Use the style defined by PROC TEMPLATE, test1.
```sas
ods tagsets.rtf file="mparskip.rtf" style=test1;
```

**Print the title and footnote using the parskip style element spacing.** The title and footnotes are printed in the RTF file. The spacing used between the title and footnote is determined by the parskip style element.
```sas
  title "Parskip Style Element Spacing between the Title and Table";
  footnote "Parskip Style Element Spacing between the Table and Footnote";
  proc print data=sashelp.class (obs=5); run;
```

**Change the spacing between the titles and footnotes using the VSPACE= suboption** Change the spacing between the titles and footnotes using the VSPACE= suboption. With VSPACE=’NO’ there is no vertical space placed before and after the table. The spacing specified by the Parskip style element is not honored.
ods tagsets.rtf style=test1 OPTIONS(vspace="NO");
title "Spacing between the Title and the Table Using VSPACE=No";
footnote "Spacing between the Table and Footnote Using VSPACE=No"
proc print data=sashelp.class (obs=5);
run;

Close the RTF destination. Close the RTF destination and Open the HTML destination to return ODS to its default settings.

ods tagsets.rtf close;
ods html;

RTF Output
The vertical spacing placed between the title and the table and the footnote and the table is determined by the parskip style element.

Output 6.98 RTF Output Using Parskip Style Element Spacing

Parskip Style Element Spacing between the Title and Table

<table>
<thead>
<tr>
<th>Obs</th>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69.0</td>
<td>112.5</td>
</tr>
<tr>
<td>2</td>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84.0</td>
</tr>
<tr>
<td>3</td>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98.0</td>
</tr>
<tr>
<td>4</td>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
<td>102.5</td>
</tr>
<tr>
<td>5</td>
<td>Henry</td>
<td>M</td>
<td>14</td>
<td>63.5</td>
<td>102.5</td>
</tr>
</tbody>
</table>

Parskip Style Element Spacing between the Table and Footnote
The vertical spacing specified by the Parskip style element is not honored when the VSPACE=NO suboption is used. No vertical space is added between the title and the table and the footnote and the table.

**Output 6.99**  
**RTF Output Using VSPACE=’NO’ - Parskip Style Spacing Is Not Honored**

### Spacing between the Title and the Table Using VSPACE=No

<table>
<thead>
<tr>
<th>Obs</th>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69.0</td>
<td>112.5</td>
</tr>
<tr>
<td>2</td>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84.0</td>
</tr>
<tr>
<td>3</td>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98.0</td>
</tr>
<tr>
<td>4</td>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
<td>102.5</td>
</tr>
<tr>
<td>5</td>
<td>Henry</td>
<td>M</td>
<td>14</td>
<td>63.5</td>
<td>102.5</td>
</tr>
</tbody>
</table>

### Spacing between the Table and Footnote Using VSPACE=No

---

**ODS TEXT= Statement**

Inserts text into your ODS output.

**Valid in:** Anywhere  
**Category:** ODS: Output Control  
**Tip:** The ODS TEXT= statement is sent only to output destinations that are open. Therefore, it must be specified after an ODS destination statement.

---

**Syntax**

**ODS TEXT= ’text-string’**

**Required Argument**

*text-string*

specifies the text to insert into your output. This text is sent to all open supported output destinations.

**Restriction**

The ODS TEXT= statement does not support the OUTPUT destination or the LISTING destination. All other ODS destinations are supported.

**Requirement**

You must enclose ‘text-string’ in quotes.

**Tip**

The UserText style element controls text specified with the TEXT= statement.
Examples

Example 1: Adding Text to Multiple Destinations

Features:
- ODS HTML statement
- ODS PDF statement
- ODS RTF statement
- ODS TEXT= statement

PROC TEMPLATE:
- DEFINE STYLE statement
- PARENT= statement
- STYLE statement

Other features:
- PROC PRINT

Data set:
- Exprev

Details

The following example uses a single ODS TEXT= statement to add text to PDF, HTML, and traditional RTF output. PROC TEMPLATE modifies the UserText style element that controls the font style, font color, and other attributes of the text that the ODS TEXT= statement adds.

Program

```sas
options obs=10;
proc template;
  define style Styles.MyStyle;
  parent=styles.htmlblue;
  style usertext from usertext /
    foreground=red;
  end;
run;
ods html file="text.html" style=Styles.MyStyle ;
ods pdf file="text.pdf" startpage=never notoc style=Styles.MyStyle ;
ods rtf file="text_trad.rtf" style=Styles.MyStyle ;
title "January Orders ";
footnote " For All Employees";
ods text="My Text 1";
ods text="My Text 2";
proc print data=exprev;
run;
ods text="My Text 3";
ods pdf close;
ods rtf close;
ods html close;
title;
footnote;
```
proc template;
    delete Styles.MyStyle ;
run;

Program Description

options obs=10;

Create the Styles.MyStyle style template. The Styles.MyStyle style template modifies
the Usertext style element to change the font color of text created by the TEXT=
statement to red.

proc template;
    define style Styles.MyStyle;
    parent=styles.htmlblue;
    style usertext from usertext /
        foreground=red;
    end;
run;

Send output to multiple ODS destinations. The following statements open the HTML,
PDF, and RTF destinations. The STYLE= options specifies that the style Styles.MyStyle
is applied to the output.

ods html file="text.html" style=Styles.MyStyle ;
ods pdf file="text.pdf" startpage=never notoc style=Styles.MyStyle ;
ods rtf file="text_trad.rtf" style=Styles.MyStyle ;

Add a title and footnote. The TITLE and FOOTNOTE statements specify the title and
footnote. You must place the TITLE and FOOTNOTE statements before the ODS
TEXT= statements.

title "January Orders ";
footnote " For All Employees";

Add text strings before the output is printed. The ODS TEXT= statements add the
text strings "My Text 1" and "My Text 2" The text is added to the output before the data
set is printed.

ods text="My Text 1";
ods text="My Text 2";

Print the data set Experev. The PRINT procedure prints the Experev data set.

proc print data=experev;
run;

Add a third text string after the data set. The third ODS TEXT= statement adds the
text string "My Text 3" after the data set is printed.

ods text="My Text 3";

Close the RTF, HTML, and PRINTER destinations and remove the titles and
footnotes. The ODS RTF CLOSE statement closes the RTF destination. The ODS PDF
CLOSE statement closes the PRINTER destination. The ODS HTML CLOSE statement
closes the HTML destination. The TITLE and FOOTNOTE statements remove any titles
and footnotes previously specified.

ods pdf close;
ods rtf close;
ods html close;
title;
footnote;

After your output is created, you can delete the Styles.MyStyle style template. The DELETE statement deletes the Styles.MyStyle style template.

proc template;
   delete Styles.MyStyle ;
run;

Output

Output 6.100  HTML Output with Text Added

Output 6.101  PDF Output with Text Added
Example 2: Adding Text That Imitates a System Title

Features:
- ODS PDF statement option
  - FILE=
  - NOTOC
  - STARTPAGE=NO
  - STYLE=
  - ODS NOPROCTITLE statement
  - ODS HTML
  - ODS TEXT statement
  - PROC TEMPLATE

Other features:
- OPTIONS statement
- PROC MEANS
- PROC PRINT
- TITLE statement

Details

SAS titles and footnotes are displayed once per page in the PDF destination. Therefore, when the STARTPAGE= option is set to NO (OFF) and output from more than one procedure or DATA_NULL_step is routed to ODS PDF, only the first set of titles and footnotes are written to the output file. This example shows how to use the TEXT= option to mimic an interim title, displayed above the second procedure output. PROC TEMPLATE is used to create a custom style template that mimics the style of the Systemtitle element. Systemtitle is the style element that controls the appearance of titles.

This example requires some knowledge of style templates, style elements, and style attributes. For complete documentation about styles, see “Understanding Styles, Style Elements, and Style Attributes” on page 913. For a table of style elements affecting
the table of contents and table of pages, see Table 11.4 on page 967. For a table of style attributes, see Table 12.1 on page 994.

**Program**

```sas
ods html close;
options nodate;

proc template;
  define style styles.mimictitle;
  parent=styles.pearl;
  class usertext from systemtitle /
    just=c;
  end;
run;
ods pdf file="file.pdf" notoc startpage=no style=styles.mimictitle;

title "Overriding the Default Procedure Title";
ods noproctitle;
proc means data=sashelp.cars;
  class cylinders;
  var mpg_city mpg_highway;
run;

ods text="My Custom PROC PRINT Output Title";

proc print data=sashelp.cars noobs;
  where mpg_highway gt 45;
run;

ods pdf close;
ods html;
proc template;
delete Styles.Mimictitle;
run;
```

**Program Description**

**Close the HTML destination so that no HTML output is produced.** The HTML destination is open by default. The ODS HTML CLOSE statement closes the HTML destination to conserve resources. If the destination were left open, then ODS would produce both HTML and PDF output.

```sas
ods html close;
```

**Set the SAS system options.**

```sas
options nodate;
```
Create a custom style template. The Usertext style element controls the appearance of user text, which includes text specified by the TEXT= option. The Systemtitle style element controls the appearance of the default SAS titles, which include the text specified by the TITLE= statement. This PROC TEMPLATE step creates a new style named Styles.Mimictitle, which contains all of the style elements and style attributes that Styles.Pearl contains. However, the CLASS statement modifies the Usertext style element to produce center-justified text, just as the Systemtitle style element does. This ensures that text specified by the TEXT= option will look the same as the text specified by the TITLE= statement.

```
proc template;
   define style styles.mimictitle;
   parent=styles.pearl;
   class usertext from systemtitle /
       just=c;
   end;
run;
```

Open the PDF destination and specify the ODS PDF statement options. The ODS PDF statement opens the PDF destination and the FILE= option specifies the PDF filename. The NOTOC option specifies that no table of contents is created. The STARTPAGE=NO option specifies that no new pages are inserted at the beginning of each procedure, or within certain procedures, even if new pages are requested by the procedure code. The STYLE= option specifies the style to use for the output, which is Styles.Mimictitle in this example.

```
ods pdf file="file.pdf" notoc startpage=no style=styles.mimictitle;
```

Specify a title.

```
title "Overriding the Default Procedure Title";
```

Create the MEANS procedure output and suppress the procedure title. The ODS NOPROCTITLE statement suppresses the writing of the procedure title that produces the results.

```
ods noproctitle;
proc means data=sashelp.cars;
   class cylinders;
   var mpg_city mpg_highway;
run;
```

Specify the text to use as the second procedure title. The TEXT= option specifies the text that appears above the PRINT procedure output. Because the custom style Styles.Mimictitle was specified in the ODS PDF statement, this text will look like a title specified by the TITLE= statement.

```
ods text="My Custom PROC PRINT Output Title";
```

Create the PRINT procedure output.

```
proc print data=sashelp.cars noobs;
   where mpg_highway gt 45;
```
Close the PDF destination and open the HTML destination. The ODS PDF CLOSE statement closes the PDF destination and all of the files that are associated with it. You must close the destinations before you can view the output with a browser or before you can send the output to a physical printer. The ODS HTML statement opens the HTML destination and returns SAS to the default ODS destination.

```sas
ods pdf close;
ods html;
```

After your output is produced, you can remove the custom style. The DELETE statement in PROC TEMPLATE removes the custom style.

```sas
proc template;
  delete Styles.Mimictitle;
run;
```

PDF Output

The following image shows the text “My Custom PROC PRINT Output Title” above the PROC PRINT table in the same style as the title.

Output 6.103  PDF Output with Custom Procedure Title

ODS TRACE Statement

Writes to the SAS log a record of each output object that is created, or suppresses the writing of this record.
Valid in: Anywhere
Category: ODS: Output Control
Default: OFF
Example: “Example 3: Creating a Data Set with and without the MATCH_ALL Option” on page 545

Syntax

ODS TRACE ON <option(s)>;
ODS TRACE OFF;

Required Arguments

OFF
  turns off the writing of the trace record.
  
  Alias  NO

ON
  turns on the writing of the trace record.
  
 Aliases  OUTPUT
  YES

Optional Arguments

DOM<="external-file">
  specifies that the ODS document object model is written to the SAS log or an external file.
  
  external-file
  is the name of an external output file.
  
  Requirement  You must enclose external-file in quotation marks.
  
  See  For complete documentation about the ODS document object model, see “Working with the ODS Document Object Model” in SAS Output Delivery System: Advanced Topics.

EXCLUDED
  includes, in the trace record, information for excluded output objects.
  
  Example  “Example 2: Conditionally Selecting Output Objects” on page 768

LABEL
  includes the label path for the output object in the record. You can use a label path anywhere that you can use a path.
  
  Tip  This option is helpful for users who are running a localized version of SAS, because the labels are translated from English to the local language. The names and paths of output objects are not translated because they are part of the syntax of the Output Delivery System.
LISTING
writes the trace record to the LISTING destination, so that each part of the trace record immediately precedes the output object that it describes.

Details

Contents of the Trace Record
ODS produces an output object by combining data from the data component with a table template. The trace record provides information about the data component, the table template, and the output object. By default, the record that the ODS TRACE statement produces contains these items:

Name
is the name of the output object. You can use the name to reference this output object and others with the same name. For details about how to reference an output object, see “How ODS Determines the Destinations for an Output Object” on page 38. For example, you could use this name in an ODS OUTPUT statement to make a data set from the output object. You could also use this name in an ODS SELECT or an ODS EXCLUDE statement.

TIP The name is the rightmost part of the path that appears in the trace record.

Label
briefly describes the contents of the output object. This label also identifies the output object in the Results window.

Data name
is the name of the data component that was used to create this output object. The data name appears only if it differs from the name of the output object.

Data label
describes the contents of the data.

Template
is the name of the table template that ODS uses to format the output object. You can modify this definition with PROC TEMPLATE. See the “EDIT Statement” in SAS Output Delivery System: Procedures Guide for more information.

Path
is the path of the output object. You can use the path to reference this output object. For example, you could use the path in the ODS OUTPUT statement to make a data set from the output. You could also use the path in an ODS SELECT or an ODS EXCLUDE statement.

The LABEL option modifies the trace record by including the label path for the object in the record. See the discussion of the option LABEL on page 856.

Specifying an Output Object
After you have determined which output objects your SAS program produces, you can specify the output objects in statements such as ODS EXCLUDE, ODS SELECT, and so on. You can specify an output object by using one of the following:

• a full path. For example, the following is the full path of the output object:

Univariate.City_Pop_90.TestsForLocation

• a partial path. A partial path consists of any part of the full path that begins immediately after a period (.) and continues to the end of the full path. For example, suppose the full path is the following:

Univariate.City_Pop_90.TestsForLocation
Then the partial paths are as follows:

City_Pop_90.TestsForLocation
TestsForLocation

• a label that is enclosed in quotation marks. For example:

"The UNIVARIATE Procedure"

• a label path. For example, the following is the label path for the output object:

"The UNIVARIATE Procedure"."CityPop_90"."Tests For Location"

Note: The trace record shows the label path only if you specify the LABEL option in the ODS TRACE statement.

• a partial label path. A partial label path consists of any part of the label that begins immediately after a period (.) and continues to the end of the label. For example, suppose the label path is the following:

"The UNIVARIATE Procedure"."CityPop_90"."Tests For Location"

Then the partial label paths are as follows:

"CityPop_90"."Tests For Location"
"Tests For Location"

• a mixture of labels and paths.

• any of the partial path specifications, followed by a pound sign (#) and a number. For example, TestsForLocation#3 refers to the third output object that is named TestsForLocation.

Example: Determining Which Output Objects a Procedure Creates

Features:
ODS TRACE statement:
   LABEL
   OFF
   ON

Other features:
   PROC UNIVARIATE

Data set:
   StatePop

Details
This example shows how to determine the names and labels of the output objects that a procedure creates. You can use this information to select and exclude output objects.

Program
ods trace on / label;
proc univariate data=statepop mu0=3.5;
   var citypop_90 citypop_80;
run;
ods trace off;
Program Description

Specify that SAS write the trace record to the SAS log and include label paths. This ODS TRACE statement writes the trace record to the SAS log. The LABEL option includes label paths in the trace record.

    ods trace on / label;

Create descriptive statistics for two variables. PROC UNIVARIATE computes descriptive statistics for two variables, CityPop_80 and CityPop_90. As PROC UNIVARIATE sends each output object to the Output Delivery System, ODS writes the pertinent information for that output object to the trace record.

    proc univariate data=statepop mu0=3.5;
        var citypop_90 citypop_80;
    run;

Specify that SAS stop writing the trace record. The ODS TRACE OFF statement stops the writing of the trace record to the SAS log.

    ods trace off;

SAS Log

This partial SAS log shows the trace record that the ODS TRACE statement creates. For each analysis variable, PROC UNIVARIATE creates five output objects: Moments, BasicMeasures, TestsForLocation, Quantiles, and ExtremeObs. Notice that an output object has the same name and label, regardless of which variable is analyzed. Therefore, you can select all the moments tables that PROC UNIVARIATE produces by using the name or label in an ODS SELECT statement. The path and label path are unique for each output object because they include the name of the variable that is analyzed. You can, therefore, select an individual moments table by using the path or the label path in an ODS SELECT statement.
Example 1: Determining Which Output Objects a Procedure Creates

Output Added:

Name: Moments
Label: Moments
Template: base.univariate.Moments
Path: Univariate.CityPop_90.Moments
Label Path: "The Univariate Procedure"."CityPop_90"."Moments"

Output Added:

Name: BasicMeasures
Label: Basic Measures of Location and Variability
Template: base.univariate.Measures
Path: Univariate.CityPop_90.BasicMeasures
Label Path: "The Univariate Procedure"."CityPop_90"."Basic Measures of Location and Variability"

Output Added:

Name: TestsForLocation
Label: Tests For Location
Template: base.univariate.Location
Path: Univariate.CityPop_90.TestsForLocation
Label Path: "The Univariate Procedure"."CityPop_90"."Tests For Location"

Output Added:

Name: Quantiles
Label: Quantiles
Template: base.univariate.Quantiles
Path: Univariate.CityPop_90.Quantiles
Label Path: "The Univariate Procedure"."CityPop_90"."Quantiles"

Output Added:

Name: ExtremeObs
Label: Extreme Observations
Template: base.univariate.ExtObs
Path: Univariate.CityPop_90.ExtremeObs
Label Path: "The Univariate Procedure"."CityPop_90"."Extreme Observations"
### Output Added:

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th>Template</th>
<th>Path</th>
<th>Label Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moments</td>
<td>Moments</td>
<td>base.univariate.Moments</td>
<td>Univariate.CityPop_80.Moments</td>
<td>&quot;The Univariate Procedure&quot;.&quot;CityPop_80&quot;.&quot;Moments&quot;</td>
</tr>
<tr>
<td>TestsForLocation</td>
<td>Tests For Location</td>
<td>base.univariate.Location</td>
<td>Univariate.CityPop_80.TestsForLocation</td>
<td>&quot;The Univariate Procedure&quot;.&quot;CityPop_80&quot;.&quot;Tests For Location&quot;</td>
</tr>
<tr>
<td>Quantiles</td>
<td>Quantiles</td>
<td>base.univariate.Quantiles</td>
<td>Univariate.CityPop_80.Quantiles</td>
<td>&quot;The Univariate Procedure&quot;.&quot;CityPop_80&quot;.&quot;Quantiles&quot;</td>
</tr>
<tr>
<td>ExtremeObs</td>
<td>Extreme Observations</td>
<td>base.univariate.ExtObs</td>
<td>Univariate.CityPop_80.ExtremeObs</td>
<td>&quot;The Univariate Procedure&quot;.&quot;CityPop_80&quot;.&quot;Extreme Observations&quot;</td>
</tr>
</tbody>
</table>

### See Also

#### Statements

- “ODS EXCLUDE Statement” on page 321
- “ODS SELECT Statement” on page 758

### ODS USEGOPT Statement

Determines whether ODS uses traditional SAS/GRAF option settings.

**Valid in:** Anywhere  
**Category:** ODS: Output Control  
**Restrictions:** The ODS USEGOPTS option has no effect on graphics produced as a result of any of the ODS graphics functionality or the ODS GRAPHICS statement. The ODS USEGOPTS option only affects the titles and footnotes for tables, not the contents of the table.
Syntax

**ODS USEGOPT | NOUSEGOPT:**

Required Arguments

**ODS USEGOPT**

specifies that ODS use traditional SAS/GRAPH option settings for non-graphical output.

**ODS NOUSEGOPT**

specifies that ODS not use traditional SAS/GRAPH option settings for non-graphical output.

Details

**Enabling Traditional SAS/GRAPH Graphics Options**

While ODS USEGOPT is in effect, the settings for the following graphics options will affect all of your ODS output, including tables:

- CTEXT=
- CTITLE=
- FTITLE=
- FTEXT=
- HTEXT=
- HTITLE=

If ODS NOUSEGOPT is in effect, the settings for these graphics options will not override the value in the style template for titles and footnotes in your ODS output.

Example: Enabling and Disabling Graphics Options

**Features:**

- ODS HTML statement option:
  - FILE=
- ODS LISTING statement action:
  - CLOSE
- ODS NOUSEGOPT statement
- ODS USEGOPT statement

**Other features:**

- GOPTIONS statement:
  - FCTEXT=
  - FTITLE=
  - HTEXT=

- PROC PRINT
  - TITLE statement

**Data set:**

- Exprev
Details
This example creates two HTML reports, one with the GOPTIONS enabled by using the ODS USEGOPT statement, and one with GOPTIONS disabled by using the ODS NOUSEGOPT statement.

Program

goptions reset=all  htext=2 ftitle=script ftext=script;
ods usegopt;
ods html file='opts.html';
title 'This Title Was Created With the USEGOPT Option Specified ' ;
title2 'The Graphics Option Settings are Turned On';
proc print data=exprev(obs=2);
run;
ods nousegopt;
title 'This Title Was Created With the NOUSEGOPT Option Specified' ;
title2 'The Graphics Option Settings are Turned Off';
proc print data=exprev (obs=2) ;
run;

Program Description

**Specify the GOPTIONS.** The RESET=ALL option sets all graphics options to their default values and cancels all global statements. The HTEXT= option specifies that the text height for titles and footnotes be two units. The FTITLE= option specifies the font for titles and footnotes. The FTEXT option specifies the font for the text.

goptions reset=all  htext=2 ftitle=script ftext=script;

**Enable the graphics options.** While ODS USEGOPT is in effect, the settings for HTEXT= and CTEXT= graphics options will override values that are specified for titles and footnotes in the style template.

ods usegopt;

**Create HTML output, specify titles, and print the data set.** The ODS HTML statement opens the HTML destination and creates HTML output. The output from PROC PRINT is sent to the body file specified by the FILE= option. The TITLE statements specify the titles for your output. The PRINT procedure prints the SAS data set Exprev. The OBS= option specifies two observations to be printed.

ods html file='opts.html';
title 'This Title Was Created With the USEGOPT Option Specified ' ;
title2 'The Graphics Option Settings are Turned On';
proc print data=exprev(obs=2);
run;

**Disable the graphics options.** The NOUSEGOPT statement suppresses the use of the HTEXT= and CTEXT= graphics option settings for your output.

ods nousegopt;

**Create HTML output, specify titles, and print the data set.** The ODS HTML statement opens the HTML destination and creates HTML output. The output from PROC PRINT
is sent to the body file specified by the FILE= option. The TITLE statements specify the
titles for your output. The PRINT procedure prints the SAS data set Exprev. The OBS=
option specifies two observations to be printed.

```sas
title 'This Title Was Created With the NOUSEGOPT Option Specified' ;
title2 'The Graphics Option Settings are Turned Off';
proc print data=exprev (obs=2) ;
run;
```

**HTML Output**

In the following example, the heights and fonts for the titles of the first table are
specified by the FTITLE, FTEXT, and HTEXT options in the GOPTIONS statement.
The heights and fonts for the titles of the second table are specified by the default style
template.

---

<table>
<thead>
<tr>
<th>Obs</th>
<th>Country</th>
<th>Emp_ID</th>
<th>Order_Date</th>
<th>Ship_Date</th>
<th>Sale_Type</th>
<th>Quantity</th>
<th>Price</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Antarctica</td>
<td>99999999</td>
<td>1/1/05</td>
<td>1/7/05</td>
<td>Internet</td>
<td>2</td>
<td>92.6</td>
<td>20.7</td>
</tr>
<tr>
<td>2</td>
<td>Puerto</td>
<td>Rico</td>
<td>99999999</td>
<td>1/5/05</td>
<td>Catalog</td>
<td>14</td>
<td>51.2</td>
<td>12.1</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Obs</th>
<th>Country</th>
<th>Emp_ID</th>
<th>Order_Date</th>
<th>Ship_Date</th>
<th>Sale_Type</th>
<th>Quantity</th>
<th>Price</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Antarctica</td>
<td>99999999</td>
<td>1/1/05</td>
<td>1/7/05</td>
<td>Internet</td>
<td>2</td>
<td>92.6</td>
<td>20.7</td>
</tr>
<tr>
<td>2</td>
<td>Puerto</td>
<td>Rico</td>
<td>99999999</td>
<td>1/5/05</td>
<td>Catalog</td>
<td>14</td>
<td>51.2</td>
<td>12.1</td>
</tr>
</tbody>
</table>

---

**ODS VERIFY Statement**

Prints or suppresses a message indicating that a style template or a table template being used is not supplied by SAS.

- **Valid in:** Anywhere
- **Category:** ODS: Output Control
- **Default:** If you do not specify the ODS VERIFY statement, then ODS runs with the verification process turned off. If you specify the ODS VERIFY statement but do not specify an argument, then ODS runs with verification turned on.
- **See:** For information about how to ignore user-created definitions, see "ODS PATH Statement" on page 563.
Syntax

ods verify <on | off | error | warn> ;

Optional Arguments

ON
prints the warning and sends output objects to open destinations.

Aliases  ODS verify
        yes

OFF
suppresses the warning.

Aliases  ODS noverify
        no

ERROR
prints an error message instead of a warning message and does not send output objects to open destinations.

WARN
prints a warning message and does not send output objects to open destinations.

Details

Using the ODS VERIFY Statement
PROC TEMPLATE can modify the values in an output object. None of the definitions that SAS provides modifies any values. If you receive a warning from the ODS VERIFY statement, then look at the source code to verify that the values have not been modified.

PUT Statement for ODS

Writes data values to a special buffer from which they can be written to the data component and then formatted by ODS.

Valid in:  DATA step
Category:  File-Handling
Type:     Executable
Requirement: If you use the _ods_ option in the PUT statement, then you must use the FILE PRINT ODS statement.

Note:  This syntax shows only the ODS form of the PUT statement when you are binding to a template. For the complete syntax, see the "PUT Statement" in SAS Statements: Reference.

Syntax

put <specification> <_ods_> <@|@@> ;
**Optional Arguments**

**specification**

specifies one or more variables to write and where to write them. specification has the following form:

<ods-pointer-control-1> variable-1 <...<ods-pointer-control-n> variable-n>

**ods-pointer-control**

moves the pointer in the buffer to a specified line or column.

See “When the Pointer Moves Past the End of a Line ” on page 867

**variable**

identifies the variable to write.

**Example** “Example 4: Creating and Using a User-Defined Table Template” on page 85

_**ODS_**

specifies that the PUT statement writes values to the data component for each of the variables that were defined as columns with the FILE PRINT ODS COLUMNS= statement.

**Default**

The order of these columns is determined by the order that is specified by the COLUMNS= suboption in the FILE PRINT ODS statement. If you omit the COLUMNS= suboption, then the order of the variables in the program data vector determines their order in the output object.

**Requirement**

If you specify the _**ODS_** option, then you must use the FILE PRINT ODS statement and the FILE PRINT ODS statement must precede the PUT _**ODS_** statement.

**Interaction**

You can use _**ODS_** in a PUT statement that specifies the placement of individual variables. _**ODS_** writes to a particular row and column only if another PUT statement has not already written a variable to that same row and column. The position of _**ODS_** in the PUT statement does not affect the outcome in the data component.

**Tip**

By default, the order of the columns in the data component matches the order of the columns in the buffer. However, if you have specified a table template, it might override this order. For more information, see the discussion of ORDER_DATA.

See For more information, see ODS<=(ODS-suboptions)> on page 155.

_**@** | **@@**_

holds an output line for the execution of the next PUT statement across iterations of the DATA step. The line-hold specifiers are called trailing @ and double trailing @.

**Default**

If you do not use @ or @@, then each PUT statement in a DATA step writes a new line to the buffer.

See “When the Pointer Moves Past the End of a Line ” on page 867
Details

**ODS Column Pointer Controls**
ODS column pointer controls differ slightly from column pointer controls in a PUT statement that does not use ODS. An ODS column refers not to a single character space but to a column that contains an entire variable value. Therefore, an ODS column pointer control moves from one entire value to the next, not from one character space to another. Column 1 contains values for the first variable in the output; column 2 contains values for the second variable, and so on.

ODS column pointer controls have the following general forms:

1. **@ods-column**
   - Moves the pointer to the specified ODS column. *ods-column* is a number, a numeric variable, or an expression that identifies the column to write to.
   - **Default** If *ods-column* exceeds the number of columns in the data component, then ODS writes the current line, moves the pointer to the first ODS column on the next line, and continues to process the PUT statement.
   - **Requirement** If *ods-column* is a number, then it must be a positive integer. If *ods-column* is a numeric variable or an expression, then SAS treats it as follows. If *ods-column* is not an integer, then SAS truncates the decimal portion and uses only the integer value. If *ods-column* is 0 or negative, then SAS moves the pointer to column 1.
   - **Tip** You can alter the default behavior with options in the FILE PRINT ODS statement. For more information, see the discussion of overflow control on page 155.

2. **+ods-column**
   - Moves the pointer by the specified number of ODS columns. *ods-column* is a number, a numeric variable, or an expression that specifies the number of columns to move the pointer.
   - **Requirement** If *ods-column* is a number, then it must be an integer. If *ods-column* is a numeric variable or an expression, then it does not have to be an integer. If it is not an integer, then SAS truncates the decimal portion and uses only the integer value. If *ods-column* is a positive integer, SAS moves the pointer to the right. If *ods-column* is a negative integer, SAS moves the pointer to the left. If *ods-column* is 0, SAS does not move the pointer.
   - **Tip** If the current column position becomes less than 1, then the pointer moves to column 1. If the current column position exceeds the number of columns in the data component, then ODS writes the current line, moves the pointer to the first ODS column on the next line, and continues to process the PUT statement.

**Example**
“Example 4: Creating and Using a User-Defined Table Template” on page 85
@ 'column-name'
    moves the pointer to the ODS column identified by 'column-name'. The column
    name is a data component variable name.

    Requirement  column-name must be enclosed in quotation marks.

**ODS Line Pointer Controls**

Line pointer controls in a DATA step that uses ODS are the same as line pointer controls
in a DATA step that does not use ODS. However, you can use only those listed below
with ODS. Line pointer controls have the following general forms:

`#line`
    moves the pointer to the specified line. line is a number, a numeric variable, or an
    expression that identifies the line to write to.

    Requirement  If line is a number, then it must be an integer. If line is a numeric
    variable or an expression, it does not have to be an integer. If it is not
    an integer, then SAS truncates the decimal portion and uses only the
    integer value.

`/`
    moves the pointer to the first column of the next line.

Example  “Example 4: Creating and Using a User-Defined Table Template” on page
85

**Note:** If you use a line pointer control to skip lines in ODS output, then all columns that
are not referenced on the current line, or skipped lines, are set to a missing value.
Columns that contain numeric values display a period for the missing value. If you
prefer not to include these periods in your ODS output, you can display missing
numeric values as a blank by using the MISSING statement (or the MISSING=
system option). For more information about the statement, see “MISSING
Statement” in *SAS Statements: Reference*. For more information about the system
option, see “MISSING= System Option” in *SAS System Options: Reference*.

**When the Pointer Moves Past the End of a Line**

In a DATA step that uses ODS, the number of columns in the buffer and in the data
component are determined in one of three ways:

- By default, the number of variables in the program data vector determines the
  number of ODS columns.
- You can override the default by defining ODS columns with the COLUMNS=
  suboption in the FILE PRINT ODS statement.
- If you associate a template with the data component, then the specifications in the
  template take precedence. As a result, the number of columns that actually appear in
  the output object could change.

When using pointer controls and the @ or @@, you might inadvertently position the
pointer beyond the last ODS column. You can control how SAS handles this situation
with options in the FILE PRINT ODS statement. For more information, see the
discussion of overflow control on page 155.

**See Also**

-  Output Delivery System and the DATA Step on page 53
• Examples on page 69

Statement
• “FILE Statement for ODS” on page 155
Part 6

Arranging Tabular and Graphic Information

Chapter 7
Arranging Output with ODS LAYOUT .......................... 871
Chapter 7
Arranging Output with ODS LAYOUT

ODS Layout Overview

The act of arranging tabular and graphic information about a page is called layout. In SAS, pages are arranged in layout containers and region containers. Titles and footnotes can be placed in layout containers or region containers. SAS supports two types of layout, absolute and gridded.

Absolute layout enables you to specify an exact page location using x and y coordinates. Absolute layout is limited to one page and is supported only by the PRINTER destinations. Use the ODS LAYOUT ABSOLUTE statement to produce absolute layout.

Gridded layout is a mechanism for arranging output dynamically. Gridded layout enables you to arrange output in a two-dimensional grid structure. For example, gridded layout enforces automatic alignment of grid cells, allows layout to continue onto the next page, dynamically computes the size of a grid cell, and makes it easier to maintain the integrity of the report. Use the ODS LAYOUT GRIDDED statement to produce gridded layout.

Gridded layout is supported for HTML, POWERPOINT, and PRINTER destinations (PDF, PS, and PCL).

The following terms are used to describe ODS layout:

footnotes

are text that appears at the bottom of a container. Footnote processing is always done before any output is produced on the physical page. This is often referred to as page initialization. Footnotes can also be specified inside regions, but only the footnotes that have changed are displayed.
layout container
is an area that contains a collection of regions or nested layouts. Layout containers can have a fixed size (WIDTH=3in and HEIGHT=4in) or can be dynamically sized to accommodate the regions.

region container
is an area that contains output (such as text, tables, images, graphics), or nested layout containers. Regions can also have a fixed size or can be dynamically sized to accommodate the collection of output.

titles
are text that appears at the top of a container or page. Title processing is always done before any output is produced on the physical page. This is often referred to as page initialization. Titles can also be specified inside regions, but only the titles that have changed are displayed.

The following image shows the various layout regions and containers.

---

Concepts

Absolute layout and gridded layout are the two types of layouts supported by SAS. Gridded and absolute layouts can be used together and can be nested.

**Absolute Layout**

Absolute layout enables you to specify an exact page location using x and y coordinates. Absolute layout is limited to one page and is supported only by the PRINTER destinations.
Each container must be explicitly placed to ensure no unwanted overlap. If the output is too large to fit in the fixed-size container, the output is discarded. You receive a blank region and a warning in your log.

Absolute layouts are suitable for static types of output, as in the following situations:

- placing output in a specific location on a pre-printed form
- creating cover pages
- precisely placing output in a nested region container

The following ODS statements are used to accomplish absolute layout:

- “ODS LAYOUT ABSOLUTE Statement” on page 873
- “ODS REGION Statement, Absolute” on page 884
- “ODS LAYOUT END Statement” on page 903

See “Example: Absolute Layout Using the Region Container” on page 877 for an example of absolute layout.

**Gridded Layout**

Gridded layout is a mechanism for arranging output dynamically. Gridded layout enables you to arrange output in a two-dimensional grid structure. For example, gridded layout enforces automatic alignment of grid cells, allows layout to continue onto the next page, dynamically computes the size of a grid cell, and makes it easier to maintain the integrity of the report. Gridded layout is supported for HTML, POWERPOINT, and PRINTER destinations (PDF, PS, and PCL).

Gridded layouts enable you to do the following:

- enforce automatic alignment of respective grid cells
- continue the layout onto the next page if necessary
- dynamically compute the size of a grid cell
- make it easier to maintain the integrity of the report

The following ODS statements are used to accomplish gridded layout:

- “ODS LAYOUT GRIDDED Statement” on page 888
- “ODS REGION Statement, Gridded” on page 899
- “ODS LAYOUT END Statement” on page 903

See “Example: Gridded Layout Titles in a Region” on page 896 for an example of using gridded layout.

**Dictionary**

**ODS LAYOUT ABSOLUTE Statement**

Enables you to specify an exact page location using x and y coordinates. Absolute layout is limited to one page and is supported only by the PRINTER destinations

**Valid in:** Anywhere
The ODS LAYOUT ABSOLUTE statement must be used with the ODS LAYOUT END statement.

**Tips:** Absolute layout enables you to specify an exact page location using x and y coordinates. Absolute layout is perfectly suited for static types of output that can be printed on a single page where you want output placed in a specific location. Examples are preprinted forms and cover pages.

Use the ODS REGION statement with the ODS LAYOUT ABSOLUTE statement to create full featured layouts. Refer to “ODS REGION Statement, Absolute” on page 884.

**Example:**

```
ods layout absolute y=1.25in x=1in width=6in;
  ods region;
    proc print data=sashelp.class;
    run;
    ods text= 'layout width=6in';
  ods layout end;
```

**Syntax**

```
ODS LAYOUT ABSOLUTE < option-1 > < option-2 ... >
```

**Summary of Optional Arguments**

- **HEIGHT=** *dimension*
  Specify a vertical height of the layout.

- **STYLE=** *<style-element-name> <style-attribute-specification(s)>]*
  Specifies one or more style elements to use for different parts of the layout.

- **WIDTH=** *dimension*
  Specify the horizontal width of the layout.

- **X=** *dimension*
  Specify the horizontal position of the layout.

- **Y=** *dimension*
  Specify the vertical starting position of the layout.

**Optional Arguments**

- **HEIGHT=** *dimension*
  Specifies the vertical height of the layout.

  *dimension* is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

  Valid units of measure:

  - cm centimeters
  - em standard typesetting measurement unit for width
  - ex standard typesetting measurement unit for height
in  inches
mm  millimeters
pct a percentage. You can also use the ‘%’ symbol.
pt  a printer’s point
px  pixels

Default  If omitted, the height of the layout defaults to the maximum vertical space needed to display all of the regions.

Example  ods layout absolute height=5in;
         proc print data=sashelp.class;
         run;
         ods layout end;

STYLE=<style-element-name> <[style-attribute-specification(s)]>
specifies the style element to use for the specified locations in the layout.

Tip  Font names that contain characters other than letters or underscores must be enclosed in quotation marks.

Example  ods layout absolute style=[backgroundcolor=yellow];
         proc print data=sashelp.class;run;
         ods layout end;

WIDTH=dimension
specifies the horizontal width of the layout.

dimension  is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

   cm  centimeters
   em  standard typesetting measurement unit for width
   ex  standard typesetting measurement unit for height
   in  inches
   mm  millimeters
   pct a percentage. You can also use the ‘%’ symbol.
   pt  a printer’s point
   px  pixels

Default  If omitted, the width of the layout defaults to the maximum horizontal space needed to display all of the regions.

Example  ods layout absolute width=4in;
         proc print data=sashelp.class;
         run;
         ods layout end;
**X=dimension**

specifies the horizontal starting position of the layout. The layout container extends to the right of the X position by the amount specified in the WIDTH= option.

*dimension*

is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- `cm` centimeters
- `em` standard typesetting measurement unit for width
- `ex` standard typesetting measurement unit for height
- `in` inches
- `mm` millimeters
- `pct` a percentage. You can also use the ‘%’ symbol.
- `pt` a printer’s point
- `px` pixels

Default 0

Example

```plaintext
ods layout absolute x=2.5in;
proc print data=sashelp.class;
run;
ods layout end;
```

**Y=dimension**

specifies the vertical starting position of the layout. The layout container extends down from the Y position by the amount specified in HEIGHT= option.

*dimension*

is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- `cm` centimeters
- `em` standard typesetting measurement unit for width
- `ex` standard typesetting measurement unit for height
- `in` inches
- `mm` millimeters
- `pct` a percentage. You can also use the ‘%’ symbol.
- `pt` a printer’s point
- `px` pixels

Default

If omitted, the Y argument defaults to the current vertical position on the page.
Details

Using Absolute Layout

The ODS LAYOUT statements enable you to create custom reports where you can easily mix graphics, images, text, and tables, and arrange them on a page. Absolute Layout enables you to specify an exact page location using x and y coordinates. Each location needs to be explicitly placed to ensure that there is no unintended overlap.

The statement used to create an absolute layout is ODS LAYOUT ABSOLUTE. ODS LAYOUT ABSOLUTE follows the traditional ODS statements 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.

Note: Use the ODS REGION statement, absolute version with the ODS LAYOUT ABSOLUTE statement. The ODS REGION statement cannot be used alone.

Maintaining program code written using absolute layout can be challenging. When one container's position is altered, you might need to manually alter all of the other containers to maintain your report's integrity.

You are not limited to a single ODS layout type. An absolute layout can contain gridded layouts and a gridded layout can contain absolute layouts.

Only ODS PRINTER destinations support absolute layout.

Reasons to Use Absolute Layout

Absolute layout is restricted to a single page. If the output is too large to fit in the fixed-size container, the output is discarded. You receive a blank region and a warning in your log. Absolute layout is perfectly suited for static types of output.

Absolute layout works well to create the following types of output:

- cover page
- static data
- preprinted form
- single page output

Note: Absolute layout works well to create preprinted forms.

Example: Absolute Layout Using the Region Container

Features:

- ODS LAYOUT ABSOLUTE statement
- ODS LAYOUT END statement
- ODS REGION statement

- ODS REGION
  - Y option
  - X option
  - HEIGHT option
WIDTH option
ODS PDF statement
PROC TEMPLATE
ODS ESCAPECHAR statement
ODS TEXT=

Other features:
FOOTNOTE statement
TITLE statement
GCHART procedure
GOPTIONS
REPORT procedure

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

Program

options nodate nonumber;
proc template;
   define style Styles.OrionCalloutBlock;
      parent = Styles.Pearl;
      style LayoutRegion/
         background=cxbbb2e0;
   end;
run;
ods escapechar="-";
title "~{style [preimage='c:\Public\orionstarHeader.jpg' width=100pct
   background=cx494068 color=cxbbb2e0 font_size=32pt] Our Company }";
footnote "~{style [font_size=10pt just=right color=cxbbb2e0]
   Using ODS Absolute Layout Features.}";
ods pdf file="OrionstarCoInfo.pdf" notoc nogtitle nogfootnote;
ods layout absolute;
ods text="~{style [preimage='c:\Public\starLarge.gif'
   font_style=italic font_size=20pt color=cxbbb2e0]Who we are...}";
ods region y=0.5in x=1in width=6in;
   ods text="The Orion Star Sports & Outdoors Company is a fictional
   international retail company that sells sports and outdoor products.
   The headquarters is based in the United States. Retail stores are
   situated in a number of other countries including Belgium, Holland,
   Germany, the United Kingdom, Denmark, France, Italy, Spain, and
   Australia.";
ods region y=1.25in x=1in width=4in;
   ods text="Products are sold in physical retail stores, by mail order
   catalogs, and through the Internet. Customers who sign up as members
   of the Orion Star Club organization can receive favorable special
   offers; therefore, most customers enroll in the Orion Star Club. The
   sales data in this scenario includes only the purchases of Orion

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

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

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

ods region y=5.35in width=4.75in height=3.75in;
goptions device=png htext=.1in;
proc gchart data=sashelp.orsales;
  pie product_category / sumvar=profit
    value=none
    percent=outside
    slice=outside;
run;
quit;
ods region y=5.5in x=4.625in width=3in height=3.7in;
proc report nowd data=sashelp.orsales
  style(header)={background=cx494068 color=cxbbb2e0};
  columns product_category profit;
  define product_category / group;
  define profit /analysis sum format=dollar14.;
run;

ods pdf style=Styles.OrionCalloutBlock;
ods region y=1.0625in x=6in width=2in height=1in;
ods text="-\{style [background=cx494068 color=cxbbb2e0 font_size=24pt just=center font_style=italic width=100pct]Our Mission\}"
ods text="-\{style [font_style=italic vjust=center font_size=10pt just=center]To deliver the best quality sporting equipment, accessories, and outdoor equipment for all seasons at the most affordable prices.\}"
ods region y=2.1875in x=6in width=2in height=1in;
ods text="-\{style [background=cx494068 color=cxbbb2e0 font_size=24pt just=center font_style=italic width=100pct]Our Vision\}"
ods text="-\{style [font_style=italic vjust=center font_size=10pt just=center]To transform the way the world purchases sporting and outdoor equipment.\}"
Program Description

Set the SAS system options. Set the NODATE and NONUMBER SAS system options.

```
options nodate nonumber;
```

Set up the style template. Create a custom style template based on Styles.Pearl for use in specified sections of the layout.

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

Set up the titles and footnotes for the page. Set the ESCAPECHAR value to "~" and use the ODS ESCAPECHAR style function to customize the titles and footnotes.

```
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.}";
```

Open the PDF destination and write to a file. Write the PDF output to file OrionstarCoInfo.pdf. In this program, we do not want a TOC to be generated nor a graphics title and footnote.

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

Set the layout to absolute layout.

```
ods layout absolute;
```
Create the first region, "Who we are", using ODS TEXT. Add the star image and text to the first region. Note that the ODS REGION statement was not used to create the first region. A region is created for you if the REGION statement is not used.

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

Create the second region using ODS TEXT. Specify exactly where the region text should go on the page by using the Y=, X=, and WIDTH= region options.

```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.";```

Create the third region using ODS TEXT. Specify the location of the third region on the page by using the Y=, X=, and WIDTH= region options.

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

Create the fourth region, "What we sell", using ODS TEXT. Add the star image and the text for the second item in the list.

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

Create the fifth region using ODS TEXT. Specify the location of the fifth region on the page by using the Y=, X=, WIDTH=, and HEIGHT= region options.

```ods region y=3in x=1in width=4in height=1.75in;
ods text="Approximately 5500 different sports and outdoor products are offered at Orion Star. Products are sold in volumes that reflect the different types of sports and outdoor activities that are performed in each country. Therefore, some products are not sold in certain countries. All of the product names are fictitious.

- {newline}Products are organized in a hierarchy consisting of three levels:
  - Product Line
  - Product Category
  - Product Group";```

Create the sixth region, "Where we generate our profit", using ODS TEXT. Add the star image and the text for the third item in the list.

```ods region y=4.75in height=1in width=5in;
ods text="-{style [preimage='c:\Public\starLarge.gif' font_style=italic font_size=20pt color=cxbbb2e0]Where we generate our profit...}";```
Create the seventh region using ODS GCHART. Create a pie chart and place it exactly where you want it to appear on the page. This pie chart shows the company profits.

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

Create the eighth region using PROC REPORT. Create a table showing the company profits. Place the table exactly where you want it to appear on the page. Note that the REPORT region intentionally overlaps part of the GCHART region.

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

Change the style in the following regions. Start using the OrionCalloutBlock style template to change the background color of the following regions.

```plaintext
ods pdf style=Styles.OrionCalloutBlock;
```

Create the ninth region, “Our Mission”, using ODS TEXT. Specify the location of the ninth region by using the Y=, X=, WIDTH=, and HEIGHT= region options.

```plaintext
ods region y=1.0625in x=6in width=2in height=1in;
ods text="~{style [background=cx494068 color=cxbbb2e0
   font_size=24pt just=center font_style=italic width=100pct]}
   Our Mission }*
ods text="~{style [font_style=italic vjust=center font_size=10pt
   just=center]To deliver the best quality sporting equipment,
   accessories, and outdoor equipment for all seasons at the
   most affordable prices.}*
```

Create the tenth region, “Our Vision”, using ODS TEXT. Specify the location of the tenth region by using the Y=, X=, WIDTH=, and HEIGHT= region options.

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

Create the eleventh region, “Our Values”, using ODS TEXT. Specify the location of the eleventh region by using the Y=, X=, WIDTH=, and HEIGHT= region options.

```plaintext
ods region y=3.3125in x=6in width=2in height=1in;
```
Create the twelfth region, "Our Goal", using ODS TEXT. Specify the location of the twelfth region by using the Y=, X=, WIDTH=, and HEIGHT= region options.

ods region x=6in y=4.4375in width=2in height=1in;
ods text="-{} Our Goal }";
ods text="-{} To grow sales by 15% annually while improving profit margins through innovative thinking and operational efficiencies.}"

End the layout. The ODS LAYOUT END statement ends the layout. The ODS PDF CLOSE statement closes the PDF destination.

ods layout end;
ods pdf close;

The following example output mixes graphics, images, text, and tables.
Output 7.1 ODS Absolute Layout – Company Information

Our Company

Who we are...
The Orion Sports & Outdoors Company is a leading international retail company that sells sports and outdoor products. The headquarters is located 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 special offers, therefore, retail customers enroll in the Orion Star Club. The sales data in this company includes only the purchases of Orion Star Club members from 1990 through 2002.

What we sell...
Approximately 500 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 sold in certain countries. All of the product names are fictitious.

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

Where we generate our profit...

The image shows a pie chart and a table. The pie chart represents the contribution of different product categories to profit, and the table provides a breakdown of profit by product category. The pie chart is color-coded to indicate the relative contribution of each category.

See Also

ODS Statements
- “ODS REGION Statement, Absolute” on page 884
- “ODS LAYOUT END Statement” on page 903

ODS REGION Statement, Absolute

Creates a region container for absolute layouts. A region can contain output (such as text, tables, images, graphs) or nested layout containers. This ODS REGION statement is sandwiched between the ODS LAYOUT ABSOLUTE statement and the ODS LAYOUT END statement. The ODS LAYOUT ABSOLUTE
statement manages the ODS LAYOUT destination when producing one page of output for PRINTER destinations.

Valid in: Anywhere
Category: ODS: Output Control
Requirement: The ODS REGION statement for absolute layout must be used with the ODS LAYOUT ABSOLUTE statement.

ODS destination: ODS LAYOUT ABSOLUTE is supported only for PRINTER destinations (PDF, PS, and PCL).

Tips: Absolute layout enables you to specify the exact location on the page to place a layout and region container. Absolute layout is perfectly suited for static types of output that can be printed on a single page where you want output placed in a specific location. Examples are preprinted forms and cover pages. Regions can have a fixed size or can be dynamically sized.

Syntax

REGION < option-1>< option-2 ...>

Summary of Optional Arguments

HEIGH T=dimension
Specify a vertical height of the region.

STYLE=<style-element-name> <[style-attribute-specification(s)]>
specifies one or more style elements to use for different parts of the layout.

WIDTH=dimension
Specify the horizontal width of the region.

X=dimension
Specify the horizontal starting position of the region.

Y=dimension
Specify the starting vertical position of the region.

Optional Arguments

HEIGHT=dimension
specifies the vertical height of the region.

dimension
is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

cm centimeters
em standard typesetting measurement unit for width
ex standard typesetting measurement unit for height
in inches
mm millimeters
pct a percentage. You can also use the ‘%’ symbol.
STYLE=<style-element-name> <[style-attribute-specification(s)]>
specifies the style element to use for the specified locations in the layout.

Tip
Font names that contain characters other than letters or underscores must be enclosed in quotation marks.

Example
ods layout absolute style=[backgroundcolor=yellow];
ods region style=[backgroundcolor=lightblue];
proc print data=sashelp.class; run;
ods layout end;

WIDTH=dimension
specifies the horizontal width of the region.

dimension
is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- cm centimeters
- em standard typesetting measurement unit for width
- ex standard typesetting measurement unit for height
- in inches
- mm millimeters
- pct a percentage. You can also use the ‘%’ symbol.
- pt a printer’s point
- px pixels

Default
If omitted, the width of the region defaults to the maximum horizontal space needed to display all of the output in that region.

Restrictions
The width is restricted by the dimensions of the region.

The sum of all region widths cannot exceed the horizontal dimension of the layout container.
Example
ods layout absolute;
  ods region width=sin;
    proc print data=sashelp.class; run;
  ods layout end;

\textbf{X=dimension}

specifies the horizontal starting position of the layout. The region extends to the right by the amount specified by the \texttt{WIDTH=} option in the ODS REGION statement.

\textit{dimension} is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- \texttt{cm} centimeters
- \texttt{em} standard typesetting measurement unit for width
- \texttt{ex} standard typesetting measurement unit for height
- \texttt{in} inches
- \texttt{mm} millimeters
- \texttt{pct} a percentage. You can also use the ‘\%’ symbol.
- \texttt{pt} a printer’s point
- \texttt{px} pixels

Default 0.

Example
ods layout absolute;
  ods region x=2.5in;
    proc print data=sashelp.class; run;
  ods layout end;

\textbf{Y=dimension}

specifies the starting vertical position of the region within the layout. The region extends down by the amount specified by the \texttt{HEIGHT} option in the ODS REGION statement.

\textit{dimension} is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- \texttt{cm} centimeters
- \texttt{em} standard typesetting measurement unit for width
- \texttt{ex} standard typesetting measurement unit for height
- \texttt{in} inches
- \texttt{mm} millimeters
- \texttt{pct} a percentage. You can also use the ‘\%’ symbol.
- \texttt{pt} a printer’s point
px  pixels

Default  If omitted, the Y argument defaults to the current vertical position within the layout.

Example  ods layout absolute;
   ods region y=5in;
       proc print data=sashelp.class; run;
   ods layout end;

Details

A region container is an area that contains output such as text, tables, graphics, and images. Region containers can be nested. The ODS REGION statement that is for absolute layout must be used within an ODS LAYOUT ABSOLUTE statement block.

See Also

ODS Statements

- “ODS LAYOUT ABSOLUTE Statement” on page 873
- “ODS LAYOUT END Statement” on page 903

ODS LAYOUT GRIDDED Statement

Enables you to arrange output dynamically in a two-dimensional grid structure. For example, gridded layout enforces automatic alignment of grid cells, allows layout to continue onto the next page, dynamically computes the size of a grid cell, and makes it easier to maintain the integrity of the report. Gridded layout is supported for HTML, POWERPOINT, and PRINTER destinations (PDF, PS, and PCL).

Valid in:  Anywhere
Category:  ODS: Output Control
Requirement:  The ODS LAYOUT GRIDDED statement must be used with the ODS LAYOUT END statement.
ODS destination:  Gridded layout is supported for HTML and PRINTER destinations (PDF, PS, and PCL). It is also supported for the ODS destination for PowerPoint.
Tip:  Gridded layout for dynamically sized regions can accommodate dynamic data, can span more than one page, and allows for easier alignment. Programs created using gridded layout are easier to maintain than those created using absolute layout.

Example:  ods layout gridded width=2in;
   ods region;
       proc print data=sashelp.class;
       run;
   ods text= 'layout width=2in';
   ods layout end;

Syntax

ODS LAYOUT GRIDDED<option-1><option-2 ...>
Summary of Optional Arguments

**ADVANCE**=<BYGROUP | EXPLICIT | PROC | TABLE | OUTPUT >
Dynamically populate the layout grid by groups, tables, pages, procedures, and explicitly.

**COLUMN_GUTTER**=dimension
Specify the horizontal space between each column.

**COLUMN_WIDTHS**=dimension
Specify the width of each column specified.

**COLUMNS**=number
Specify the fixed number of columns in a gridded layout.

**HEIGHT**=dimension
Specify the vertical height of the layout.

**ORDER_TYPE**=<ROW_MAJOR | COLUMN_MAJOR>
Populate the grid by rows or columns.

**ROW_GUTTER**=dimension
Specify the vertical space between each row.

**ROW_HEIGHTS**=dimension
Specify the height of each row specified.

**ROWS**=number
Specify the fixed number of rows in the gridded layout.

**STYLE**=<style-element-name> <[style-attribute-specification(s)]>
specifies one or more style elements to use for different parts of the layout.

**WIDTH**=dimension
Specify the horizontal width of the layout.

**X**=dimension
Specify the horizontal starting position of the layout.

**Y**=dimension
Specify the vertical starting position of the layout.

Without Arguments
Without arguments, the default is ONE column and ONE region.

Optional Arguments

**ADVANCE**=<BYGROUP | EXPLICIT | PROC | TABLE | OUTPUT >
specifies that the grid is explicitly populated or that the grid is dynamically populated by groups, tables, pages, and procedures.

**BYGROUP**
specifies that the gridded layout dynamically advance to a new region for every BYGROUP encountered.

ods layout gridded columns=2 advance=bygroup;

**EXPLICIT**
specifies that the gridded layout dynamically populates the region explicitly before moving on to the next region. All output is in ONE region unless ODS REGION statements are being used.

**Note**
If you are changing ADVANCE= options within the same layout and using ADVANCE=EXPLICIT, the output does not advance until after EXPLICIT is used.
PROC specifies that the gridded layout dynamically populates the region by procedures before moving on to the next region.

TABLE specifies that the gridded layout dynamically populates the region by tables before moving on to the next region.

OUTPUT specifies that the gridded layout dynamically populates the region by output before moving on to the next region.

Default EXPLICIT

COLUMN=number specifies the fixed number of columns in a gridded layout.

Default 1

COLUMN_GUTTER=dimension specifies the horizontal space between each column.

dimension is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- cm centimeters
- em standard typesetting measurement unit for width
- ex standard typesetting measurement unit for height
- in inches
- mm millimeters
- pct a percentage. You can also use the ‘%’ symbol.
- pt a printer’s point
- px pixels

Example  ods layout gridded columns=2 column_gutter=1in;

ods region;
  proc print data=sashelp.air(obs=5);
  run;

ods region;
  proc means data=sashelp.class n mean;
  var height weight;
  run;

ods layout end;

COLUMN_WIDTHS=dimension specifies the width of each column specified. This is a space-delimited list of horizontal sizes that correspond to each column.
**dimension**

is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- **cm** centimeters
- **em** standard typesetting measurement unit for width
- **ex** standard typesetting measurement unit for height
- **in** inches
- **mm** millimeters
- **pct** a percentage. You can also use the ‘%’ symbol.
- **pt** a printer’s point
- **px** pixels

**Restriction**

A warning is given and the option is ignored when the number of column widths does not match the number of columns specified. The number of column widths must be equal to the number of columns requested in the LAYOUT statement.

**Example**

```plaintext
ods layout gridded columns=2 column_widths=(2in 2in);

ods region;
  proc print data=sashelp.class;
  run;
ods region;
  proc print data=sashelp.class;
  run;
ods layout end;
```

**HEIGHT=**

specifies the vertical height of the layout.

**dimension**

is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- **cm** centimeters
- **em** standard typesetting measurement unit for width
- **ex** standard typesetting measurement unit for height
- **in** inches
- **mm** millimeters
- **pct** a percentage. You can also use the ‘%’ symbol.
- **pt** a printer’s point
- **px** pixels
If omitted, the height of the layout defaults to the maximum vertical space needed to display all of the regions.

Example
ods layout gridded height=7in;
   proc print data=sashelp.class; run
ods layout end;

ORDER_TYPE=\texttt{ROW\_MAJOR | COLUMN\_MAJOR}
populates the grid by rows or columns.

\texttt{COLUMN\_MAJOR}
specifies that the gridded layout first populates all regions in the first column before moving on to the next column.

\texttt{ROW\_MAJOR}
specifies that the gridded layout first populates all regions in the row before moving on to the next row.

Default \texttt{ROW\_MAJOR}

Restrictions
The \texttt{ORDER\_TYPE=} option is valid only in PRINTER destinations.

The gridded layout option \texttt{ORDER\_TYPE=COLUMN\_MAJOR} does not support the gridded region option \texttt{COLUMN\_SPAN}. When these options are specified together, the \texttt{COLUMN\_SPAN=} option is ignored and a warning message is written to the SAS log.

Examples
This code populates row1 column1, row1 column2, row2 column1, and row2 column2:
ods layout gridded columns=2 rows=2 order_type=row_major;

This example populates row1 column1, row2 column1, row1 column2, and row2 column2:
ods layout gridded columns=2 rows=2 order_type=column_major;

\texttt{ROWS=}\texttt{number}
specifies the fixed number of rows in the gridded layout.

Default If omitted, the \texttt{ROWS=} option defaults to the maximum number of rows needed to populate the regions created in the vertical direction. If there are two columns, then the number of rows is half of the number of regions.

Example
ods layout gridded rows=2 columns=1;
   ods region;
   proc print data=sashelp.class(obs=1); run;
ods region;
   proc print data=sashelp.class(obs=1); run;
ods layout end

\texttt{ROW\_GUTTER=}\texttt{dimension}
specifies the vertical space between each row.
**dimension**

is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- cm centimeters
- em standard typesetting measurement unit for width
- ex standard typesetting measurement unit for height
- in inches
- mm millimeters
- pct a percentage. You can also use the ‘%’ symbol.
- pt a printer’s point
- px pixels

**Example**

```
ods layout gridded rows=2 row_gutter=1in;

ods region;
    proc print data=sashelp.air(obs=5);
    run;

ods region;
    proc means data=sashelp.class n mean;
    var height weight;
    run;

ods layout end;
```

**ROW_HEIGHTS=** *dimension*

specifies the height of each row specified. This is a space delimited list of vertical sizes that correspond to each row.

**dimension**

is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- cm centimeters
- em standard typesetting measurement unit for width
- ex standard typesetting measurement unit for height
- in inches
- mm millimeters
- pct a percentage. You can also use the ‘%’ symbol.
- pt a printer’s point
- px pixels
Restriction The number of row heights specified must match the number of rows specified in the LAYOUT statement.

Example This example shows 3 rows with different row heights.
```ods layout gridded rows=3 row_heights=(1in 2in 3in);
ods region;
   proc print data=sashelp.class(obs=1); run;
ods region;
   proc means data=sashelp.class n mean; run;
ods region;
   proc print data=sashelp.class(obs=3); run;
ods layout end;
```

STYLE=<style-element-name> <[style-attribute-specification(s)]>
specifies the style element to use for the specified locations in the layout.

Tip Font names that contain characters other than letters or underscores must be enclosed in quotation marks.

Example ods layout gridded columns=2 style=[backgroundcolor=yellow];
   proc print data=sashelp.class;run;
ods layout end;

WIDTH=dimension
specifies the horizontal width of the layout.

dimension is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:
- cm centimeters
- em standard typesetting measurement unit for width
- ex standard typesetting measurement unit for height
- in inches
- mm millimeters
- pct a percentage. You can also use the ‘%’ symbol.
- pt a printer’s point
- px pixels

Default If omitted, the width of the layout defaults to the maximum horizontal space needed to display all of the regions.

Example ods layout gridded width=7in;
   proc print data=sashelp.class; run
ods layout end;
X=\textit{dimension}

specifies the horizontal starting position of the layout.

\textit{dimension} is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- cm centimeters
- em standard typesetting measurement unit for width
- ex standard typesetting measurement unit for height
- in inches
- mm millimeters
- pct a percentage. You can also use the ‘%’ symbol.
- pt a printer’s point
- px pixels

Default If omitted, gridded layout is centered by default.

Example
\begin{verbatim}
ods layout gridded x=10cm;
proc print data=sashelp.class; run
ods layout end;
\end{verbatim}

Y=\textit{dimension}

specifies the vertical starting position of the layout.

\textit{dimension} is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- cm centimeters
- em standard typesetting measurement unit for width
- ex standard typesetting measurement unit for height
- in inches
- mm millimeters
- pct a percentage. You can also use the ‘%’ symbol.
- pt a printer’s point
- px pixels

Default If omitted, the Y argument defaults to the current vertical position on the page.

Example
\begin{verbatim}
ods layout gridded y=10cm;
    proc print data=sashelp.class; run
ods layout end;
\end{verbatim}
Details

Using ODS Gridded Layout

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

ODS LAYOUT GRIDDED follows the traditional ODS statements 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 GRIDDED statement is used to provide gridded layout.

Note: A gridded layout can contain absolute layouts, and absolute layouts can contain gridded layouts. You are not limited to a single ODS layout type. However, absolute layouts are limited to PRINTER destinations.

Gridded layout is supported by the following destinations:

- HTML
- PRINTER

Reasons to Use Gridded Layout

Gridded layout enables you to arrange output in a two-dimensional gridded structure, such as a spreadsheet or piece of graph paper. Gridded layout is a mechanism for arranging output dynamically. Gridded layout is a simple, powerful tool for arranging output and managing output on one or more pages. Gridded layouts have the following capabilities:

- provide automatic alignment of respective grid cells
- continue the layout onto the next page when necessary
- dynamically compute the size of a grid cell
- makes it easier to maintain the integrity of the report.

Example: Gridded Layout Titles in a Region

Features:

- ODS LAYOUT GRIDDED statement
- ODS REGION statement
- ODS LAYOUT END statement
- ODS PDF statement

Other features:

- FOOTNOTE statement
- TITLE statement
- GCHART procedure
- PRINT procedure
- GOPTIONS

Details

The following example uses the ODS LAYOUT GRIDDED statement and the ODS REGION statement for gridded layout to show how titles and footnotes are generated.
Program

options nodate nonumber;
ods pdf file='LayoutGriddedTitles.pdf';
title 'This is TITLE1';
footnote 'This is FOOTNOTE1';
ods layout gridded;
ods region;
title 'This is the REGION TITLE';
footnote 'This is the REGION FOOTNOTE';
proc print data=sashelp.class(obs=10);
run;

goptions hsize=4in vsize=4in;
proc gchart data=sashelp.class;
  vbar age / name='gtitle';
  title 'This is the PROCEDURE TITLE';
  footnote 'This is the PROCEDURE FOOTNOTE';
run;
quit;
ods layout end;
ods pdf close;

Program Description

Set the SAS system options NODATE and NONUMBER.

options nodate nonumber;

Open the PDF destination and write to a file. Write the PDF output to the file LayoutGriddedTitles.pdf.

ods pdf file='LayoutGriddedTitles.pdf';

Add a global title and footnote. The title and footnote are added to the Title and Footnote sections of the document.

title 'This is TITLE1';
footnote 'This is FOOTNOTE1';

Set the layout to gridded layout. Using the ODS LAYOUT GRIDDED statement, write titles and footnotes to the region container. This region contains PROC PRINT and PROC GCHART output. Note that the title and footnote are changed and are placed in the layout region.

ods layout gridded;
ods region;
title 'This is the REGION TITLE';
footnote 'This is the REGION FOOTNOTE';
proc print data=sashelp.class(obs=10);
run;

Change the PROC GCHART procedure title and footnote.

goptions hsize=4in vsize=4in;
proc gchart data=sashelp.class;
vbar age / name='gttitle';
title 'This is the PROCEDURE TITLE';
footnote 'This is the PROCEDURE FOOTNOTE';
run;
quit;

**End the layout.** The ODS LAYOUT END statement ends the layout. The ODS PDF CLOSE statement closes the PDF destination.

```
ods layout end;
ods pdf close;
```

The following output shows how tables, titles, footnotes, and charts are placed on a page using gridded layout:

**Output 7.2 ODS Gridded Layout – Changing Titles and Footnotes**

This is TITLE1

This is the REGION TITLE

<table>
<thead>
<tr>
<th>Obs</th>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alfred</td>
<td>M</td>
<td>40</td>
<td>69.0</td>
<td>112.3</td>
</tr>
<tr>
<td>2</td>
<td>Alice</td>
<td>F</td>
<td>35</td>
<td>58.5</td>
<td>84.0</td>
</tr>
<tr>
<td>3</td>
<td>Barbara</td>
<td>F</td>
<td>65</td>
<td>65.3</td>
<td>96.0</td>
</tr>
<tr>
<td>4</td>
<td>Carol</td>
<td>F</td>
<td>44</td>
<td>62.8</td>
<td>102.8</td>
</tr>
<tr>
<td>5</td>
<td>Henry</td>
<td>M</td>
<td>63</td>
<td>63.5</td>
<td>102.8</td>
</tr>
<tr>
<td>6</td>
<td>James</td>
<td>M</td>
<td>12</td>
<td>57.3</td>
<td>83.0</td>
</tr>
<tr>
<td>7</td>
<td>Jane</td>
<td>F</td>
<td>12</td>
<td>59.8</td>
<td>84.5</td>
</tr>
<tr>
<td>8</td>
<td>Janet</td>
<td>F</td>
<td>15</td>
<td>62.5</td>
<td>112.8</td>
</tr>
<tr>
<td>9</td>
<td>Jeffrey</td>
<td>M</td>
<td>63</td>
<td>62.5</td>
<td>96.0</td>
</tr>
<tr>
<td>10</td>
<td>John</td>
<td>M</td>
<td>12</td>
<td>59.0</td>
<td>96.0</td>
</tr>
</tbody>
</table>

This is the REGION FOOTNOTE

This is the PROCEDURE TITLE

FREQUENCY

<table>
<thead>
<tr>
<th>Age MIDPOINT</th>
<th>FREQUENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.4</td>
<td>1</td>
</tr>
<tr>
<td>12.8</td>
<td>7</td>
</tr>
<tr>
<td>13.8</td>
<td>5</td>
</tr>
<tr>
<td>15.0</td>
<td>3</td>
</tr>
<tr>
<td>16.2</td>
<td>1</td>
</tr>
</tbody>
</table>

This is FOOTNOTE1
See Also

**ODS Statements**

- “ODS REGION Statement, Gridded” on page 899
- “ODS LAYOUT END Statement” on page 903

---

**ODS REGION Statement, Gridded**

Creates a region container for gridded layouts. A region can contain output (such as text, tables, images, graphs) or nested layout containers. This ODS REGION statement is sandwiched between the ODS LAYOUT GRIDDED statement and the ODS LAYOUT END statement. The ODS LAYOUT GRIDDED statement manages the ODS LAYOUT destination.

**Valid in:** Anywhere

**Category:** ODS: Output Control

**Requirement:** The ODS REGION statement for gridded layout must be used with the ODS LAYOUT GRIDDED statement.

**ODS destination:** ODS LAYOUT GRIDDED is supported for HTML, PRINTER, and SASREPORT destinations.

**Tip:** Regions can have a fixed size or can be dynamically sized.

---

**Syntax**

`ODS REGION (<option-1><option-2 ...>);`

---

**Summary of Optional Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN_SPAN=number</td>
<td>Specify the number of grid columns that the region occupies.</td>
</tr>
<tr>
<td>COLUMNS=number</td>
<td>Specify the current grid column position in the gridded layout.</td>
</tr>
<tr>
<td>HEIGHT=dimension</td>
<td>Specify the vertical height of the region.</td>
</tr>
<tr>
<td>ROW_SPAN=number</td>
<td>Specify the number of grid rows that the region occupies.</td>
</tr>
<tr>
<td>ROW=number</td>
<td>Specify the current grid row position in the gridded layout.</td>
</tr>
<tr>
<td>STYLE=</td>
<td>Specifies one or more style elements to use for different parts of the layout.</td>
</tr>
<tr>
<td>WIDTH=dimension</td>
<td>Specify the horizontal width of the region.</td>
</tr>
</tbody>
</table>

---

**Optional Arguments**

**COLUMNS= number**

specifies the current grid column position in the gridded layout. The gridded layout automatically tracks the current grid column position and is incremented for every ODS REGION statement.
Restrictions  If you have skipped a gridded column, you cannot go back to it. For example, the following code is not allowed.

```sas
ods layout gridded rows=1 columns=3;
ods region column=3;
   proc print data=sashelp.class;
   run;

ods region column=1;
   proc print data=sashelp.class;
   run;
ods layout end;
```

Random access of grid rows and columns is not supported.

Tip  The COLUMN= option is useful when you want to skip regions in the gridded layout.

Example  Note that the PROC MEANS goes to COLUMN=2 instead of to COLUMN=1.

```sas
ods layout gridded columns=3;
ods region column=2;
   proc means data=sashelp.class n mean;
      var height weight;
   run;
ods layout end;
```

**COLUMN_SPAN=number**  
specifies the number of grid columns that the region occupies. The COLUMN_SPAN argument enables you to combine adjacent grid columns in gridded layout.

Default  1

Restriction  The gridded layout option ORDER_TYPE=COLUMN_MAJOR does not support the gridded region option COLUMN_SPAN=. When these options are specified together, the COLUMN_SPAN= option is ignored and a warning message is written to the SAS log.

Tip  When the value specified for COLUMN_SPAN= cannot resolve with other options like ROW= or COLUMNS=, the COLUMN_SPAN= option is ignored and a message is written to the SAS log.

Example  In this example, the second region spans two columns.

```sas
ods layout gridded columns=3;
ods region;
   proc print data=sashelp.class;
   run;
ods region column_span=2;
   proc means data=sashelp.class n mean;
      var height weight;
   run;
ods layout end;
```

**HEIGHT=dimension**  
specifies the vertical height of the region.
dimension

is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- cm centimeters
- em standard typesetting measurement unit for width
- ex standard typesetting measurement unit for height
- in inches
- mm millimeters
- pct a percentage. You can also use the ‘%’ symbol.
- pt a printer’s point
- px pixels

Default

If you omit the HEIGHT= option, the default is the maximum vertical space needed to display the output contained in the region.

Restrictions

The height is restricted by the dimensions of the layout container.

The sum of all region heights cannot exceed the vertical dimension of the layout.

Example

```ods layout gridded;
ods region height=5in;
    proc print data=sashelp.class; run;
ods layout end;
```

ROW=number

specifies the current grid row position in the gridded layout. The gridded layout automatically tracks the current row position and is incremented for every ODS REGION statement.

Restrictions

Random access of grid rows and columns is not supported.

Once you have skipped a gridded row, you cannot go back to it. The following is an example of code that is not allowed.

```ods layout gridded rows=3 columns=1;
ods region row=3;
    proc print data=sashelp.class; run;
ods region row=1;
    proc print data=sashelp.class; run;
ods layout end;
```

Tip

The ROW= option is useful when you want to skip regions in the gridded layout.

ROW_SPAN=number

specifies the number of grid rows that the region occupies. The ROW_SPAN= option enables you to combine adjacent grid rows in gridded layout.
When the value specified for `ROW_SPAN=` cannot resolve with other options like `ROW=` or `COLUMNS=`, the `ROW_SPAN` option is ignored and a message is output to the SAS log.

Example

In the following example code, the second region spans two rows.

```sas
ods layout gridded columns=3;
ods region;
   proc print data=sashelp.class(keep=name obs=1); run;
ods region row_span=2;
   proc print data=sashelp.class(keep=name obs=8);run;
ods region;
   proc print data=sashelp.class(keep=age obs=1); run;
ods region;
   proc print data=sashelp.class(keep=weight obs=1); run;
ods region;
   proc print data=sashelp.class(keep=height obs=1); run;
ods layout end;
```

**STYLE=** `<style-element-name> <[style-attribute-specification(s)]>`

specifies the style element to use for the specified locations in the layout.

Tip

Font names that contain characters other than letters or underscores must be enclosed in quotation marks.

Example

```sas
ods layout gridded columns=2 style=[backgroundcolor=yellow];
ods region style=[backgroundcolor=lightblue];
   proc print data=sashelp.class;run;
ods layout end;
```

**WIDTH=** `dimension`

specifies the horizontal width of the region.

`dimension`

is a nonnegative number followed by an optional unit of measure. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

Valid units of measure:

- `cm` centimeters
- `em` standard typesetting measurement unit for width
- `ex` standard typesetting measurement unit for height
- `in` inches
- `mm` millimeters
- `pct` a percentage. You can also use the ‘%’ symbol.
- `pt` a printer’s point
- `px` pixels

Default

If you omit the `WIDTH=` option, the default is the maximum horizontal space needed to display the output contained in the region.
Restrictions

The width is restricted by the dimensions of the layout container.

The sum of all region widths cannot exceed the horizontal dimension of the layout container.

Example

ods layout gridded;
  ods region width=5in;
    proc print data=sashelp.class; run;
ods layout end;

See Also

ODS Statements

- “ODS LAYOUT GRIDDED Statement” on page 888
- “ODS LAYOUT END Statement” on page 903

ODS LAYOUT END Statement

Ends a gridded or absolute layout statement block.

Valid in: Anywhere

Category: ODS: Output Control

Requirement: The ODS LAYOUT END statement must be used with the ODS LAYOUT GRIDDED or the ODS LAYOUT ABSOLUTE statements.

Example: ods layout gridded;
  ods region;
    proc print data=sashelp.class;
    run;
ods layout end;

Syntax

ODS LAYOUT END;

Without Arguments

Use the ODS LAYOUT END statement to end the ODS LAYOUT GRIDDED or the ODS LAYOUT ABSOLUTE statement block.
Chapter 8
System Options for ODS

Dictionary

ODSDEST= System Option
Changes the default ODS destination. Starting in SAS 9.3, HTML is the default output destination for the SAS Windowing environment in the Microsoft Windows and UNIX operating systems. For all other platforms, LISTING is the default destination.

Valid in: Configuration file, SAS invocation
Category: ODS Printing
PROC OPTIONS GROUP= ODSPRINT
Restriction: ODSDEST= can be configured only at SAS start-up. Subsequent to start up, use the corresponding ODS statements. ODSDEST= applies only to the default destination, and not to any subsequent user-created destinations. ODSDEST=HTML applies only to the TAGSETS.HTML4 destination.
Operating environment: UNIX, Windows, z/OS

Syntax
ODSDEST= HTML | LISTING | AUTO

Syntax Description

HTML
specifies that HTML (TAGSETS.HTML4) is the default output destination in the SAS Windowing environment on the Microsoft Windows and UNIX operating systems.
LISTING
specifies that the LISTING destination is the default output destination.

AUTO
specifies that the SAS registry settings determine the default destination setting. This setting is HTML by default in SAS 9.4 in the SAS Windowing environment for Microsoft Windows and UNIX operating systems. HTML 4.0 is the HTML version supported by default.

*Note:* HTML output in the SAS Windowing environment is the default for Microsoft Windows and UNIX. When you run SAS in batch mode or on other operating systems, the LISTING destination is open and is the default, ODS Graphics is not enabled by default, and the default style for HTML output is HTMLBlue.

Details
At SAS session start-up, you can change the default output destination by adding ODSDEST= to the configuration file. If you do not configure the ODSDEST= option, the default destination is taken from the Registry setting. HTML is the default registry setting for the SAS Windowing environment in the Microsoft Windows and UNIX operating systems. The LISTING destination is the default output destination for all other environments and operating systems.

See Also
“Working with Output Defaults” on page 21

**ODSGRAPHICS= System Option**
Controls ODS Graphics processing.

**Valid in:** Configuration file, SAS invocation

**Category:** ODS Printing

**PROC OPTIONS GROUP=:** ODSPRINT

**Restriction:** ODSGRAPHICS= can be configured only at SAS start-up. Subsequent to start up, use the corresponding ODS statements.

**Operating environment:** UNIX, Windows, z/OS

**Syntax**

`ODSGRAPHICS=` **ON | OFF | AUTO**

**Syntax Description**

**ON**
- enables ODS Graphics processing by default.

**OFF**
- disables ODS graphics processing.
AUTO
specifies that the SAS registry setting determines the default destination setting.

Note: ODS Graphics is enabled in the SAS Windowing environment for Microsoft Windows and UNIX. When you run SAS in batch mode or on other operating systems, ODS Graphics is not enabled by default, the LISTING destination is open and is the default, and the default style for HTML output is HTMLBlue.

Details
ODS Graphics is enabled by default in the SAS Windowing environment for Microsoft Windows and UNIX. When ODS graphics processing is enabled, the graphs are integrated with tables and all output is displayed in the same HTML file using the HTMLBlue style. This new style is an all-color style, which is designed to integrate tables and modern statistical graphics.

When you run large computational programs, you might not want to create graphs. In those cases, you should disable ODS Graphics to improve the performance of your program. You can enable and disable ODS Graphics in your SAS programs with the ODS GRAPHICS OFF and ODS GRAPHICS ON statements. You can also change the ODS Graphics default in the Preference window under the Results tab.

Prior to SAS 9.3, ODS Graphics was disabled by default on all platforms. This default behavior changed in SAS 9.3. For more information about the default behavior, see “Working with Output Defaults” on page 21.

Example
In the following example, ODS Graphics is disabled, the ODS default destination is LISTING, and the style used is the default style as specified in the SAS Registry. In the SAS Windowing environment on UNIX and Microsoft Windows, ODS Graphics is enabled, HTML is the default output destination, and the default style is HTMLBlue.

   options odsgraphics=off odsdest=listing odsstyle=default;

See Also
“Working with Output Defaults” on page 21

ODSSTYLE= System Option
Specifies the default style to use.

Valid in: Configuration file, SAS invocation, OPTIONS Statement, Systems Options window
Category: ODS Printing
PROC OPTIONS GROUP= ODSPRINT
Operating environment: UNIX, Windows, z/OS

Syntax
ODSSTYLE= style-name | AUTO
Syntax Description

style-name

specifies the default style for ODS HTML destinations. Starting in SAS 9.3, HTMLBlue is the default style used for HTML output in the SAS Windowing environment for UNIX and Microsoft Windows. In 9.4, HTMLBlue is the default style for HTML on all platforms.

Note: By default, ODS displays the procedure or DATA step results in a style. The TEMPLATE procedure creates and modifies styles. The Output Delivery System uses these styles to produce customized formatted output.

See For a list of styles that are included with the SAS product, see Chapter 10, “Style Templates,” on page 923.

AUTO

specifies that the SAS registry setting determine the default style setting. In SAS 9.3, HTMLBlue is the default style for HTML in the SAS Windowing environment on UNIX and Microsoft Windows. Starting in SAS 9.4, HTMLBlue is the default style for HTML produced on all platforms.

In HTML, the style impacts TABULAR and GRAPHICS output.

Details

The ODSSTYLE= option can be specified at any time during the SAS session.

Example

The following example shows how to change the HTML style that is output in the Display Manager from the default style to the Sapphire style.

```sas
options odsstyle=Sapphire;
ods html close;
ods html;
```

See Also

“Working with Output Defaults” on page 21
Part 8

ODS Styles Reference

Chapter 9
  Overview .......................................................... 913

Chapter 10
  Style Templates .................................................. 923

Chapter 11
  Style Elements .................................................... 965

Chapter 12
  Style Attributes .................................................. 993
Understanding Styles, Style Elements, and Style Attributes

The appearance of SAS output is controlled by style templates (styles). A style is a type of ODS template that defines the visual aspects (colors, fonts, lines, markers, and so on) of SAS output. A style determines the overall look of the documents that use it. Style templates consist of style elements and style attributes.

- A style element is a named collection of style attributes that apply to a particular part of the output. Each area of ODS output has a style element name that is associated with it. The style element name specifies where the style attributes are applied. For example, a style element might contain instructions for the presentation of column headings or for the presentation of the data inside the cells. Style elements might also specify default colors and fonts for output that uses the style.

- A style attribute is a visual property, such as color, font properties, and line characteristics, that is defined in ODS with a reserved name and value. Style attributes are collectively referenced by a style element within a style template. Each style attribute specifies a value for one aspect of the presentation. For example, the BACKGROUNDCOLOR= attribute specifies the color for the background of an HTML table or for a colored table in printed output. The FONTSTYLE= attribute specifies whether to use a Roman font or an italic font.

Note: Because styles control the presentation of the data, they have no effect on output objects that go to the LISTING, DOCUMENT, or OUTPUT destination.

Available styles are in the SASHELP.TMPLMST item store. In SAS Enterprise Guide, the list of style sheets is shown by the Style Wizard. In batch mode or SAS Studio, you can display the list of available style templates by submitting this code:

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

For complete information about viewing ODS styles, see “Viewing ODS Styles Supplied by SAS” on page 923.
By default, HTML output uses the HTMLBlue style template. To help you become familiar with styles, style elements, and style attributes, look at the relationship between them. The diagram that follows shows the relationship between the style, the style elements, and the style attributes. The following figure illustrates the structure of a style:
Figure 9.1  Diagram of the HtmlBlue Style

```plaintext
proc template;
  define style Styles.HTMLBlue;
    parent = styles.statistical;
    class GraphColors /
      'gblockheader' = cxcfd5de
      'gcphasebox' = cxd299e1
      'gphasebox' = cxd2eb8f2
      'gczonec' = cxeceee0
      'gzonec' = cxccdcee
      'gczoneb' = cxcddcee
      'gzoneb' = cxd2e5f3
      'gzonea' = cxedf7
      'gconramp3cend' = cxc9c1c00
      'gconramp3cneutral' = cxc222222
      'gconramp3cstart' = cx0e36ac
      'gramp3cend' = cxd05b5b
      'gramp3neutral' = cxfafbe
      'gramp3start' = cx66fafa2
      'gcontrollim' = cx6f2ff
      'gccontrollim' = cx8f2ff9
      'gruntest' = cxcae3ff
      'gcruntest' = cxbf94d4d
      'gclipping' = cxfffc6
      'gclipping' = cxc1c100
  end;

...more style elements and style attributes...

  class Header /
    bordercolor = cxb07bb
    bgcolor = cxedf2f9
    color = cx112277;

  class Footer /
    bordercolor = cxb07bb
    bgcolor = cxedf2f9
    color = cx112277;

  class RowHeader /
    bordercolor = cxb07bb
    bgcolor = cxedf2f9
    color = cx112277;

  class RowFooter /
    bordercolor = cxb07bb
    bgcolor = cxedf2f9
    color = cx112277;

  class Table /
    cellpadding = 5;

  class Graph /
    attrpriority = "Color";

  class GraphFit2 /
    linestyle = 1;

  class GraphClipping /
    markerstyle = "circlefilled";
end;
run;
*** END OF TEXT ***
```
The following list corresponds to the numbered items in the preceding figure:

1. Styles.HtmlBlue is the style. Styles describe how to display presentation aspects (color, font, font size, and so on) of the SAS output. A style determines the overall appearance of the ODS documents that use it. The default style for HTML output is HtmlBlue. Each style consists of style elements. Each destination has a default style that is applied to all output that is written to the destination.
   - The default style for HTML output is HTMLBlue.
   - The default style for PRINTER output is Pearl.
   - The default style for RTF output is RTF.

You can create new styles with the “DEFINE STYLE Statement” in SAS Output Delivery System: Procedures Guide. New styles can be created independently or from an existing style. You can use “PARENT= Statement” in SAS Output Delivery System: Procedures Guide to create a new style from an existing style. For complete documentation about ODS styles, see Chapter 10, “Style Templates,” on page 923.

2. Header and Footer are examples of style elements. A style element is a collection of style attributes that apply to a particular part of the output for a SAS program. For example, a style element might contain instructions for the presentation of column headings or for the presentation of the data inside table cells. Style elements might also specify default colors and fonts for output that uses the style. Style elements exist inside styles and consist of one or more style attributes. Style elements can be user-defined or supplied by SAS. User-defined style elements can be created by the “STYLE Statement” in SAS Output Delivery System: Procedures Guide.

   Note: For a list of the default style elements used for HTML and markup languages and their inheritance, see Chapter 11, “Style Elements,” on page 965.

3. BORDERCOLOR=, BACKGROUND_COLOR=, and COLOR= are examples of style attributes. Style attributes specify a value for one aspect of the area of the output that its style element applies to. For example, the COLOR= attribute specifies the value cx112277 for the font color. For a list of style attributes supplied by SAS, see Chapter 12, “Style Attributes,” on page 993.

   Style attributes can be referenced with style references. See “style-reference” on page 1041 for more information about style references.

The following table shows commonly used style attributes that you can set with the STYLE= option in PROC PRINT, PROC TABULATE, and PROC REPORT. Most of these attributes apply to parts of the table other than cells (for example, table borders and the lines between columns and rows). Note that not all attributes are valid in all destinations. For more information about these style attributes, their valid values, and their applicable destinations, see “Style Attributes Tables ” in SAS Output Delivery System: Procedures Guide.

### Table 9.1 Style Attributes for PROC REPORT, PROC TABULATE, and PROC PRINT

<table>
<thead>
<tr>
<th>Attribute</th>
<th>PROC REPORT Statement Table Area</th>
<th>PROC REPORT Areas: CALLDEF, COLUMN, HEADER, LINES, SUMMARY</th>
<th>PROC TABULATE STATEMENTS VAR, CLASS, BOX, CLASSLEVEL, KEYWORD</th>
<th>PROC PRINT Table Location</th>
<th>PROC PRINT: all locations other than TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASIS=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Attribute</td>
<td>PROC REPORT STATEMENT REPORT Area</td>
<td>PROC REPORT Areas: CALLDEF, COLUMN, HEADER, LINES, SUMMARY</td>
<td>PROC TABULATE STATEMENT TABLE</td>
<td>PROC TABULATE STATEMENTS VAR, CLASS, BOX, CLASSLEV, KEYWORD</td>
<td>PROC PRINT: all locations other than TABLE</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------</td>
<td>-------------------------------</td>
<td>---------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>BACKGROUNDCOLOR</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>BACKGOUNDIMAGE</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>BORDERBOTTOMCOLOR</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERBOTTOMSTYLE</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERBOTTOMWIDTH</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERLEFTCOLOR</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERLEFTSTYLE</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERLEFTWIDTH</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERCOLOR</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>BORDERCOLORDARK</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>BORDERCOLORLIGHT</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>BODERRIGHTCOLOR</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BODERRIGHTSTYLE</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BODERRIGHTWIDTH</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERTOPCOLOR</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BODERTOPSTYLE</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BODERTOPWIDTH</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERWIDTH</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CELLPADDING</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CELSPACING</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>PROC REPORT STATEMENT REPORT Area</td>
<td>PROC REPORT Areas: CALLDEF, COLUMN, HEADER, LINES, SUMMARY</td>
<td>PROC TABULATE STATEMENT TABLE</td>
<td>PROC TABULATE STATEMENTS VAR, CLASS, BOX, CLASSLEV, KEYWORD</td>
<td>PROC PRINT TABLE location</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------------</td>
<td>----------------------------------------------------------</td>
<td>-------------------------------</td>
<td>------------------------------------------------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>CELLWIDTH=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CLASS=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>COLOR=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FLYOVER=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FONT=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FONTFAMILY=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FONTSIZE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FONTSTYLE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FONTSIZE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>FRAME=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>HEIGHT=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>HREFTARGET=</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>HTMLSTYLE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NOBREAKSPACE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>OUTPUTWIDTH=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>POSTHTML=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>POSTIMAGE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>POSTTEXT=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PREHTML=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PREIMAGE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PRETEXT=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PROTECTSPECIALCHARS=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>RULES=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TAGATTR=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Using Styles with Base SAS Procedures

Most Base SAS procedures that support ODS use one or more table templates to produce output objects. These table templates include templates for table elements: columns, headers, and footers. Each table element can specify the use of one or more style elements for various parts of the output. These style elements cannot be specified within the syntax of the procedure, but you can use customized styles for the ODS destinations that you use. For more information about customizing tables and styles, see “TEMPLATE Procedure: Creating a Style Template” in SAS Output Delivery System: Procedures Guide.

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 the STYLE= option with these procedure statements to modify the appearance of your report. The STYLE= option enables you to make changes in sections of output without changing the default style for all of the output. You can customize specific sections of procedure output by specifying the STYLE= option in specific statements within the procedure.

The following program uses the STYLE= option to create the background colors in the PROC REPORT 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;
```
The following program uses the `STYLE=` option to create the colors in the PROC TABULATE output below:

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

proc tabulate data=prdsale;
   class region division prodtype / style=[background=lightgreen];
   classlev region division prodtype / style=[background=yellow];
   var actual / style=[background=tan];
   keyword all sum / style=[background=linen color=blue];
   keylabel all='Total';
   table (region all)*(division all),
     (prodtype all)*(actual*f=dollar10.) /
     box=[label='Region by Division and Type' style=[backgroundcolor=orange]]; 

   title 'Actual Product Sales';
   title2 '(millions of dollars)';
   run;
```
The following program uses the `STYLE=` option to create the colors in the PROC PRINT output below:

```sas
proc print data=exprev noobs sumlabel='Total' GRANDTOTAL_LABEL="Grand Total"
  style(table)=[frame=box rules=groups]
  style(bysumline)=[background=red foreground=linen]
  style(grandtotal)=[foreground=green]
  style(header)=[font_style=italic background=orange];
by sale_type order_date;
sum price quantity;
sumby sale_type;
label sale_type='Sale Type' order_date='Sale Date';
format price dollar10.2 cost dollar10.2;
  var Country / style(data)=[font_face=arial font_weight=bold background=linen];
  var Price / style(data)=[font_style=italic background=yellow];
  var Cost / style(data)=[foreground=hgt. background=lightgreen];
title 'Retail and Quantity Totals for Each Sale Type';
run;
```

For the complete input data set, see “EXPREV” in *Base SAS Procedures Guide*.
## Retail and Quantity Totals for Each Sale Type

### Sale Type=Catalog Sale Date=1/1/12

<table>
<thead>
<tr>
<th>Country</th>
<th>Price</th>
<th>Cost</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Puerto Rico</td>
<td>$51.20</td>
<td>$12.10</td>
<td>14</td>
</tr>
<tr>
<td>Aruba</td>
<td>$123.70</td>
<td>$59.00</td>
<td>30</td>
</tr>
<tr>
<td>Bahamas</td>
<td>$113.40</td>
<td>$28.45</td>
<td>8</td>
</tr>
<tr>
<td>Bermuda</td>
<td>$41.00</td>
<td>$9.25</td>
<td>7</td>
</tr>
</tbody>
</table>

### Sale Type=Catalog Sale Date=1/2/12

<table>
<thead>
<tr>
<th>Country</th>
<th>Price</th>
<th>Cost</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>British Virgin Islands</td>
<td>$40.20</td>
<td>$20.20</td>
<td>11</td>
</tr>
<tr>
<td>Canada</td>
<td>$11.80</td>
<td>$5.00</td>
<td>100</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$381.30</strong></td>
<td><strong>$170</strong></td>
<td></td>
</tr>
</tbody>
</table>

### Sale Type=In Store Sale Date=1/1/12

<table>
<thead>
<tr>
<th>Country</th>
<th>Price</th>
<th>Cost</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virgin Islands (U.S.)</td>
<td>$31.10</td>
<td>$15.66</td>
<td>25</td>
</tr>
</tbody>
</table>

### Sale Type=In Store Sale Date=1/2/12

<table>
<thead>
<tr>
<th>Country</th>
<th>Price</th>
<th>Cost</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Belize</td>
<td>$146.40</td>
<td>$36.70</td>
<td>2</td>
</tr>
<tr>
<td>Cayman Islands</td>
<td>$71.00</td>
<td>$32.30</td>
<td>20</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$248.50</strong></td>
<td><strong>47</strong></td>
<td></td>
</tr>
</tbody>
</table>

### Sale Type=Internet Sale Date=1/1/12

<table>
<thead>
<tr>
<th>Country</th>
<th>Price</th>
<th>Cost</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antarctica</td>
<td>$92.60</td>
<td>$20.70</td>
<td>2</td>
</tr>
<tr>
<td><strong>Grand Total</strong></td>
<td><strong>$722.40</strong></td>
<td></td>
<td><strong>219</strong></td>
</tr>
</tbody>
</table>

Figure 9.4 Enhanced PROC PRINT Output
Chapter 10
Style Templates

Viewing ODS Styles Supplied by SAS ........................................ 923
Table of Suggested ODS Styles ............................................. 925
Program for Viewing Multiple Styles ................................. 926
ODS Styles Gallery ......................................................... 928
  EPUB Daisy Style ...................................................... 928
  HTML Styles ............................................................ 930
  Printer Styles ........................................................... 957
  Styles for the ODS Destination for PowerPoint .................... 962
  Excel Style ............................................................... 963

Viewing ODS Styles Supplied by SAS

To view the names of all of the style templates that are shipped with SAS, submit the following program. The style templates are all located in the Styles folder by default.

    proc template;
    list styles;
    run;

To view the source code of a specified style, submit the following code:

    proc template;
    source styles.style-name;
    run;
Log 10.1 Partial Source Code for the HTMLBlue Style

```plaintext
proc template;
source styles.htmlBlue;
define style Styles.htmlBlue;
  parent = styles.statistical;
  class GraphColors /
    'gndata12' = cxECE8C4
    'gndata11' = cxDBD8F8
    'gndata10' = cxC6E4BF
    'gndata9' = cxE6CEAD
    'gndata8' = cxE5C1D4
    'gndata7' = cxC9D9F0
    'gndata6' = cxDDDEB5
    'gndata5' = cxDBC7E7
    'gndata4' = cxD5C6B4
    'gndata3' = cxB7D4D3
    'gndata2' = cxE7B3B4
    'gndata1' = cxBBC2DC
    'gndata' = cxC8C9CB
```

*Note:* If you are using SAS Studio, you do not need to specify the STYLE= option. You can go to Preferences ⇒ Results and change the style from the drop-down list for your selected destination.
Table of Suggested ODS Styles

With ODS, you can use any style with any output destination. However, for each destination, SAS supplies one or more styles that are optimized to work with the output the destination creates.

Table 10.1  Recommended Styles for ODS Destinations

<table>
<thead>
<tr>
<th>Destination</th>
<th>Recommended Styles</th>
<th>Default Style</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPUB</td>
<td>Daisy</td>
<td>Daisy</td>
</tr>
<tr>
<td></td>
<td>Moonflower</td>
<td></td>
</tr>
<tr>
<td>ODS destination for Excel</td>
<td>Excel</td>
<td>Excel</td>
</tr>
<tr>
<td>Printer family of statements</td>
<td>FancyPrinter</td>
<td>Pearl</td>
</tr>
<tr>
<td></td>
<td>FestivalPrinter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GrayscalePrinter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MeadowPrinter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MonoChromePrinter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monospace</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NormalPrinter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pearl</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Printer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sapphire</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SasDocPrinter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SeasidePrinter</td>
<td></td>
</tr>
<tr>
<td>RTF</td>
<td>RTF</td>
<td>RTF</td>
</tr>
<tr>
<td>TAGSETS.RTF</td>
<td>RTF</td>
<td>RTF</td>
</tr>
<tr>
<td>ODS destination for PowerPoint</td>
<td>PowerPointDark</td>
<td>PowerPointLight</td>
</tr>
<tr>
<td></td>
<td>PowerPointLight</td>
<td></td>
</tr>
<tr>
<td>LISTING</td>
<td>Listing</td>
<td>Listing</td>
</tr>
</tbody>
</table>
### Program for Viewing Multiple Styles

This program creates a sample report in HTML, PDF, and RTF of every style supplied by SAS. The output appears in your working directory. Although you can apply most

---

### Destination | Recommended Styles | Default Style
---

TAGSETS.EXCELXP | Default | Default

* The Moonflower style for ODS EPUB is designed for nighttime or low-light reading.
Program for Viewing Multiple Styles

927

SAS styles to any destination, SAS supplies one or more styles that are optimized to
work with the output the destination creates. For a table of suggested ODS styles for
each destination, see Table 10.1 on page 925.
The table Gallery is created and is used in a subsequent DATA step to generate a list of
each style for each destination. The destination link is a hyperlink.
ods _all_ close;
proc template;
define table gallery;
column libname memname style
define libname ;
blank_dups=on;
end;
define links;
header = 'Samples';
compute as '<a href="' ||
'<a href="' ||
'<a href="' ||
end;
end;
run;

links;

trim(style) ||'.html">HTML</a> ' ||
trim(style) ||'.pdf">PDF</a> ' ||
trim(style) ||'.rtf">RTF</a>';

The DATA step creates an index of all available styles supplied by SAS from the
template store Sashelp.Tmplmst.
ods html file="index.html";
title "Index of all styles";
data _null_;
set sashelp.Gallery(where=(libname="SASHELP"));
file print ods=(template='Gallery');
put _ods_;
run;
ods html close;

The ODS destination statements create the output. You can add additional destinations
by specifying the following statement for each destination: ods destination
file="&style..destination-extension" style=&style;
%macro generateods();
options nodate;
ods html file="&style..html" style=&style;
ods pdf file="&style..pdf" style=&style;
ods rtf file="&style..rtf" style=&style;
title "Style is: &style";

The ODS NOPTITLE statement removes the procedure title.
ods noptitle;

The ODS SELECT statement selects the Variables table for the gallery.
ods select variables;
proc contents data=sashelp.class;
run;
ods _all_ close;
%mend;


The ODS NORESULTS statement prevents an entry in the results window for each of the subsequent PROC CONTENTS steps that are generated.

```sas
ods noresults;
```

This DATA step creates a sample of each style.

```sas
data _null_
   set sashelp.Gallery(where=(libname="SASHELP"));
   call symputx('style', style);
   call execute('%generateods');
run;
```

The ODS RESULTS and ODS PREFERENCES statements set the ODS options back to defaults.

```sas
ods results;
ods preferences;
```

---

**ODS Styles Gallery**

**EPUB Daisy Style**

*Output 10.1  EPUB Book Title Page*
## Output 10.2  EPUB Book, Page 1

The GLM Procedure

### Class Level Information

<table>
<thead>
<tr>
<th>Class Levels</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driv</td>
<td>A D F</td>
</tr>
</tbody>
</table>

Number of Observations Read: 30
Number of Observations Used: 30

## Output 10.3  EPUB Book, Page 2

The GLM Procedure

### Dependent Variable: PostTreatment

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Type I SS</th>
<th>Mean Square</th>
<th>F</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>6</td>
<td>6911.42048</td>
<td>178.226410</td>
<td>10.78</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>Error</td>
<td>24</td>
<td>397.557952</td>
<td>16.364915</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>29</td>
<td>1288.700000</td>
<td>44.600000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

R-Square: 0.691505
Coeff Var: 51.51901
Root MSE: 4.070003
PostTreatment Mean: 7.900000

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Type III SS</th>
<th>Mean Square</th>
<th>F</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drug</td>
<td>2</td>
<td>290.0000000</td>
<td>145.800000</td>
<td>8.85</td>
<td>0.0013</td>
</tr>
<tr>
<td>PreTreat</td>
<td>1</td>
<td>517.6874030</td>
<td>517.687403</td>
<td>34.85</td>
<td>&lt; 0.001</td>
</tr>
<tr>
<td>PreTreat*Drug</td>
<td>2</td>
<td>19.6449451</td>
<td>9.8222230</td>
<td>0.55</td>
<td>0.9506</td>
</tr>
<tr>
<td>Error</td>
<td>24</td>
<td>397.557952</td>
<td>16.364915</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>29</td>
<td>1288.700000</td>
<td>44.600000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

R-Square: 0.6921505
Coeff Var: 51.51901
Root MSE: 4.070003
PostTreatment Mean: 7.900000
You can view and modify the default HTML style by selecting **Tools \rightarrow Options \rightarrow Preferences** from the menu at the top of the main SAS window. Then open the **Results** tab. You can change the style by selecting a style from the **Style** drop-down menu. The settings in your Preferences window persist until you explicitly change them. The following display shows the **Results** tab with the new HTML style specified:

**Note:** If you are using SAS Studio, you do not need to specify the **STYLE=** option. You can go to **Preferences \rightarrow Results** and change the style from the drop-down list for your selected destination.
Figure 10.2  Changing the HTML Style with the Preferences Window
Figure 10.3 Changing the HTML Style with SAS Studio
Output 10.5  Analysis Style

![Bar chart showing sales of different furniture items (bed, chair, desk, sofa, and table) across regions (East and West).](chart1)

![Scatter plot showing the relationship between height and weight with confidence and prediction limits.](chart2)

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$925.00</td>
<td>$850.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$989.00</td>
<td>$927.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$606.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$683.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$655.00</td>
<td>$696.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$948.00</td>
<td>$986.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$114.00</td>
<td>$504.00</td>
</tr>
</tbody>
</table>
Output 10.6  BarretttsBlue Style

![Graphs showing sales data across different regions for different products (BED, CHAIR, DESK, SOFA, TABLE).](image)

![Graph showing the relationship between height and weight with confidence and prediction intervals.](image)

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$925.00</td>
<td>$850.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$999.00</td>
<td>$397.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$608.00</td>
<td>$846.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$656.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$848.00</td>
<td>$436.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$114.00</td>
<td>$554.00</td>
</tr>
</tbody>
</table>
Output 10.7  BlockPrint Style

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$923.00</td>
<td>$850.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$999.00</td>
<td>$297.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$608.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$656.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$948.00</td>
<td>$486.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
</tbody>
</table>
Output 10.8  Default Style
Output 10.9  Dove Style

![Graph showing sales data for different regions and products.](image)

![Graph showing weight vs. height relationship with confidence and prediction limits.](image)

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$925.00</td>
<td>$850.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$999.00</td>
<td>$897.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$508.00</td>
<td>$846.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$942.00</td>
<td>$553.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$858.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$948.00</td>
<td>$486.00</td>
</tr>
</tbody>
</table>
Output 10.10  Dtree Style

Dtree Style

---

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$925.00</td>
<td>$950.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$999.00</td>
<td>$297.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$608.00</td>
<td>$846.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$656.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$948.00</td>
<td>$498.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$444.00</td>
<td>$564.00</td>
</tr>
</tbody>
</table>
Output 10.11  EGDefault Style

![Graph showing data for BED, CHAIR, and DESK across regions EAST and WEST, with predicted and actual sales comparisons.]

![Graph showing scatter plot with regression line, 95% confidence limits, and 95% prediction limits.]

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$925.00</td>
<td>$950.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$999.00</td>
<td>$937.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$608.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$656.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$948.00</td>
<td>$436.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$114.00</td>
<td>$564.00</td>
</tr>
</tbody>
</table>
Output 10.12  Festival Style

![Graph showing sales by region for different furniture items.](image)

![Graph showing the relationship between height and weight with regression line and confidence limits.](image)

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$625.00</td>
<td>$800.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$668.00</td>
<td>$287.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$600.00</td>
<td>$546.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$353.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$658.00</td>
<td>$546.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$648.00</td>
<td>$498.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$114.00</td>
<td>$504.00</td>
</tr>
</tbody>
</table>
Output 10.13  Gantt Style

The Gantt chart shows the sales data for different products (BED, CHAIR, DESK, SOFA, TABLE) across two regions (EAST, WEST). The chart includes predicted and actual sales data.

The second chart illustrates a scatter plot with a regression line indicating the relationship between weight and height. The plot includes 95% confidence limits and prediction limits.

The table below provides a summary of the sales data for SOFA in the EAST region:

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$925.00</td>
<td>$850.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$999.00</td>
<td>$297.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$608.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$656.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$948.00</td>
<td>$486.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$644.00</td>
<td>$674.00</td>
</tr>
</tbody>
</table>
Output 10.14  Harvest Style

---

### Table: Predicted vs. Actual Sales

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$325.00</td>
<td>$300.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$300.00</td>
<td>$287.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$308.00</td>
<td>$346.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$352.00</td>
<td>$333.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$355.00</td>
<td>$446.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$345.00</td>
<td>$459.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$312.00</td>
<td>$377.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$314.00</td>
<td>$364.00</td>
</tr>
</tbody>
</table>
Output 10.16  HTMLBlue Style

![Graphs showing sales data](image)

![Graph showing scatter plot with regression line and confidence limits](image)

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$625.00</td>
<td>$850.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$988.00</td>
<td>$297.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$608.00</td>
<td>$546.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$656.00</td>
<td>$546.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$940.00</td>
<td>$466.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$114.00</td>
<td>$564.00</td>
</tr>
</tbody>
</table>
Output 10.17  Journal Style

[Graph showing sales of BED, CHAIR, SOFA, and DESK in EAST and WEST regions with predicted and actual sales data.]

[Graph showing weight and height data with 95% confidence and prediction limits, and regression line.]

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$625.00</td>
<td>$650.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$999.00</td>
<td>$297.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$832.00</td>
<td>$845.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$542.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$500.00</td>
<td>$500.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$942.00</td>
<td>$488.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$12.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$114.00</td>
<td>$204.00</td>
</tr>
</tbody>
</table>
Output 10.18  Meadow Style

![Graphs showing sales data for different regions and products.](image)

![Graph showing weight vs. height with regression line and confidence limits.](image)

### Table

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$925.00</td>
<td>$950.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$900.00</td>
<td>$907.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$960.00</td>
<td>$940.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$942.00</td>
<td>$953.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$950.00</td>
<td>$946.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$940.00</td>
<td>$940.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$910.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$114.00</td>
<td>$584.00</td>
</tr>
</tbody>
</table>
Output 10.19  Minimal Style

![Graphs depicting sales data and regression analysis]

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$925.00</td>
<td>$850.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$999.00</td>
<td>$920.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$608.00</td>
<td>$946.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$656.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$948.00</td>
<td>$486.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$114.00</td>
<td>$564.00</td>
</tr>
</tbody>
</table>
Output 10.20  Netdraw and NoFontDefault Styles

![Graph showing sales data for different products in different regions.](image)

![Graph showing the relationship between height and weight with confidence intervals.](image)

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$925.00</td>
<td>$650.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$999.00</td>
<td>$297.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$608.00</td>
<td>$846.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$626.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$648.00</td>
<td>$466.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$644.00</td>
<td>$561.00</td>
</tr>
</tbody>
</table>
Output 10.21  Normal Style

---

**Table:**

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$925.00</td>
<td>$850.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$900.00</td>
<td>$897.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$908.00</td>
<td>$846.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$942.00</td>
<td>$933.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$950.00</td>
<td>$940.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$946.00</td>
<td>$496.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$812.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$114.00</td>
<td>$584.00</td>
</tr>
</tbody>
</table>
Output 10.22  Ocean Style

![Graph showing sales data for different regions and products]

![Graph showing weight vs height relationship]

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$925.00</td>
<td>$850.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$999.00</td>
<td>$977.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$968.00</td>
<td>$846.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$656.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$948.00</td>
<td>$486.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$114.00</td>
<td>$564.00</td>
</tr>
</tbody>
</table>
Output 10.23  Plateau Style
Output 10.24  Raven Style
Output 10.25  SasWeb Style

![Graph of predicted vs. actual sales for different products and regions.](image)

![Graph showing weight vs. height with regression line.](image)

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$625.00</td>
<td>$650.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$600.00</td>
<td>$597.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$600.00</td>
<td>$648.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$633.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$668.00</td>
<td>$646.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$648.00</td>
<td>$490.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$3114.00</td>
<td>$504.00</td>
</tr>
</tbody>
</table>
Output 10.26 Seaside Style
Output 10.27  StatDoc Style

---

![Graph showing predicted and actual sales for different regions.](image)

---

![Graph showing weight vs height with regression line and confidence intervals.](image)

---

<table>
<thead>
<tr>
<th>Obs</th>
<th>PRODUCT</th>
<th>REGION</th>
<th>ACTUAL</th>
<th>PREDICT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOFA</td>
<td>EAST</td>
<td>$620.00</td>
<td>$650.00</td>
</tr>
<tr>
<td>2</td>
<td>SOFA</td>
<td>EAST</td>
<td>$600.00</td>
<td>$607.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$608.00</td>
<td>$640.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$642.00</td>
<td>$632.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$600.00</td>
<td>$640.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$648.00</td>
<td>$488.00</td>
</tr>
<tr>
<td>7</td>
<td>SOFA</td>
<td>EAST</td>
<td>$612.00</td>
<td>$717.00</td>
</tr>
<tr>
<td>8</td>
<td>SOFA</td>
<td>EAST</td>
<td>$7114.00</td>
<td>$664.00</td>
</tr>
</tbody>
</table>
Output 10.28 Statistical Style

- Bed sales comparison by region (EAST, WEST).
- Chair sales comparison by region (EAST, WEST).
- Desk sales comparison by region (EAST, WEST).

- Sofa sales comparison by region (EAST, WEST).
- Table sales comparison by region (EAST, WEST).

- Scatter plot showing weight vs. height with 95% confidence limits, 95% prediction limits, and regression line.

- Table showing observed (Actual) and predicted (Predict) values for different products (SOFA) in different regions (EAST).
**Printer Styles**

**Output 10.29  FancyPrinter Style**

<table>
<thead>
<tr>
<th>Obs</th>
<th>Make</th>
<th>Model</th>
<th>Type</th>
<th>Origin</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acura</td>
<td>MDX</td>
<td>SUV</td>
<td>Asia</td>
<td>$33,337</td>
</tr>
<tr>
<td>2</td>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$21,761</td>
</tr>
<tr>
<td>3</td>
<td>Acura</td>
<td>TSX 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$24,647</td>
</tr>
<tr>
<td>4</td>
<td>Acura</td>
<td>TL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$30,299</td>
</tr>
<tr>
<td>5</td>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$39,014</td>
</tr>
<tr>
<td>6</td>
<td>Acura</td>
<td>3.5 RL w/Navigation 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$41,100</td>
</tr>
<tr>
<td>7</td>
<td>Acura</td>
<td>NSX coupe 2dr manual S</td>
<td>Sports</td>
<td>Asia</td>
<td>$79,978</td>
</tr>
<tr>
<td>8</td>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$23,508</td>
</tr>
<tr>
<td>9</td>
<td>Audi</td>
<td>A4 1.8T convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$32,506</td>
</tr>
<tr>
<td>10</td>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$28,846</td>
</tr>
<tr>
<td>11</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr manual</td>
<td>Sedan</td>
<td>Europe</td>
<td>$30,366</td>
</tr>
<tr>
<td>12</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr auto</td>
<td>Sedan</td>
<td>Europe</td>
<td>$31,388</td>
</tr>
<tr>
<td>13</td>
<td>Audi</td>
<td>A6 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$33,129</td>
</tr>
<tr>
<td>14</td>
<td>Audi</td>
<td>A6 3.0 Quattro 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$35,992</td>
</tr>
<tr>
<td>15</td>
<td>Audi</td>
<td>A4 3.0 convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$38,325</td>
</tr>
</tbody>
</table>

**Output 10.30  FestivalPrinter Style**

<table>
<thead>
<tr>
<th>Obs</th>
<th>Make</th>
<th>Model</th>
<th>Type</th>
<th>Origin</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acura</td>
<td>MDX</td>
<td>SUV</td>
<td>Asia</td>
<td>$33,337</td>
</tr>
<tr>
<td>2</td>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$21,761</td>
</tr>
<tr>
<td>3</td>
<td>Acura</td>
<td>TSX 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$24,647</td>
</tr>
<tr>
<td>4</td>
<td>Acura</td>
<td>TL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$30,299</td>
</tr>
<tr>
<td>5</td>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$39,014</td>
</tr>
<tr>
<td>6</td>
<td>Acura</td>
<td>3.5 RL w/Navigation 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$41,100</td>
</tr>
<tr>
<td>7</td>
<td>Acura</td>
<td>NSX coupe 2dr manual S</td>
<td>Sports</td>
<td>Asia</td>
<td>$79,978</td>
</tr>
<tr>
<td>8</td>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$23,508</td>
</tr>
<tr>
<td>9</td>
<td>Audi</td>
<td>A4 1.8T convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$32,506</td>
</tr>
<tr>
<td>10</td>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$28,846</td>
</tr>
<tr>
<td>11</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr manual</td>
<td>Sedan</td>
<td>Europe</td>
<td>$30,366</td>
</tr>
<tr>
<td>12</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr auto</td>
<td>Sedan</td>
<td>Europe</td>
<td>$31,388</td>
</tr>
<tr>
<td>13</td>
<td>Audi</td>
<td>A6 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$33,129</td>
</tr>
<tr>
<td>14</td>
<td>Audi</td>
<td>A6 3.0 Quattro 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$35,992</td>
</tr>
<tr>
<td>15</td>
<td>Audi</td>
<td>A4 3.0 convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$38,325</td>
</tr>
</tbody>
</table>
Output 10.31  GrayscalePrinter Style

<table>
<thead>
<tr>
<th>Obs</th>
<th>Make</th>
<th>Model</th>
<th>Type</th>
<th>Origin</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acura</td>
<td>MDX</td>
<td>SUV</td>
<td>Asia</td>
<td>$33,337</td>
</tr>
<tr>
<td>2</td>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$21,761</td>
</tr>
<tr>
<td>3</td>
<td>Acura</td>
<td>TSX 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$24,647</td>
</tr>
<tr>
<td>4</td>
<td>Acura</td>
<td>TL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$30,299</td>
</tr>
<tr>
<td>5</td>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$39,014</td>
</tr>
<tr>
<td>6</td>
<td>Acura</td>
<td>3.5 RL w/Navigation 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$41,100</td>
</tr>
<tr>
<td>7</td>
<td>Acura</td>
<td>NSX coupe 2dr manual S</td>
<td>Sports</td>
<td>Asia</td>
<td>$79,978</td>
</tr>
<tr>
<td>8</td>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$23,508</td>
</tr>
<tr>
<td>9</td>
<td>Audi</td>
<td>A4 1.8T convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$32,506</td>
</tr>
<tr>
<td>10</td>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$28,946</td>
</tr>
<tr>
<td>11</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr manual</td>
<td>Sedan</td>
<td>Europe</td>
<td>$30,366</td>
</tr>
<tr>
<td>12</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr auto</td>
<td>Sedan</td>
<td>Europe</td>
<td>$31,388</td>
</tr>
<tr>
<td>13</td>
<td>Audi</td>
<td>A6 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$33,129</td>
</tr>
<tr>
<td>14</td>
<td>Audi</td>
<td>A6 3.0 Quattro 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$35,992</td>
</tr>
<tr>
<td>15</td>
<td>Audi</td>
<td>A4 3.0 convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$38,325</td>
</tr>
</tbody>
</table>

Output 10.32  MeadowPrinter Style

<table>
<thead>
<tr>
<th>Obs</th>
<th>Make</th>
<th>Model</th>
<th>Type</th>
<th>Origin</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acura</td>
<td>MDX</td>
<td>SUV</td>
<td>Asia</td>
<td>$33,337</td>
</tr>
<tr>
<td>2</td>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$21,761</td>
</tr>
<tr>
<td>3</td>
<td>Acura</td>
<td>TSX 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$24,647</td>
</tr>
<tr>
<td>4</td>
<td>Acura</td>
<td>TL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$30,299</td>
</tr>
<tr>
<td>5</td>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$39,014</td>
</tr>
<tr>
<td>6</td>
<td>Acura</td>
<td>3.5 RL w/Navigation 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$41,100</td>
</tr>
<tr>
<td>7</td>
<td>Acura</td>
<td>NSX coupe 2dr manual S</td>
<td>Sports</td>
<td>Asia</td>
<td>$79,978</td>
</tr>
<tr>
<td>8</td>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$23,508</td>
</tr>
<tr>
<td>9</td>
<td>Audi</td>
<td>A4 1.8T convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$32,506</td>
</tr>
<tr>
<td>10</td>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$28,946</td>
</tr>
<tr>
<td>11</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr manual</td>
<td>Sedan</td>
<td>Europe</td>
<td>$30,366</td>
</tr>
<tr>
<td>12</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr auto</td>
<td>Sedan</td>
<td>Europe</td>
<td>$31,388</td>
</tr>
<tr>
<td>13</td>
<td>Audi</td>
<td>A6 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$33,129</td>
</tr>
<tr>
<td>14</td>
<td>Audi</td>
<td>A6 3.0 Quattro 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$35,992</td>
</tr>
<tr>
<td>15</td>
<td>Audi</td>
<td>A4 3.0 convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$38,325</td>
</tr>
</tbody>
</table>
### Output 10.33  MonoChromePrinter Style

<table>
<thead>
<tr>
<th>Obs</th>
<th>Make</th>
<th>Model</th>
<th>Type</th>
<th>Origin</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acura</td>
<td>MDX</td>
<td>SUV</td>
<td>Asia</td>
<td>$33,337</td>
</tr>
<tr>
<td>2</td>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$21,761</td>
</tr>
<tr>
<td>3</td>
<td>Acura</td>
<td>TSX 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$24,647</td>
</tr>
<tr>
<td>4</td>
<td>Acura</td>
<td>TL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$30,299</td>
</tr>
<tr>
<td>5</td>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$39,014</td>
</tr>
<tr>
<td>6</td>
<td>Acura</td>
<td>3.5 RL w/Nav</td>
<td>Sedan</td>
<td>Asia</td>
<td>$41,100</td>
</tr>
<tr>
<td>7</td>
<td>Acura</td>
<td>NSX coupe 2dr</td>
<td>Sports</td>
<td>Asia</td>
<td>$79,978</td>
</tr>
<tr>
<td>8</td>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$23,508</td>
</tr>
<tr>
<td>9</td>
<td>Audi</td>
<td>A4 1.8T convert</td>
<td>Sedan</td>
<td>Europe</td>
<td>$32,506</td>
</tr>
<tr>
<td>10</td>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$38,846</td>
</tr>
<tr>
<td>11</td>
<td>Audi</td>
<td>A4 3.0 Quattro</td>
<td>Sedan</td>
<td>Europe</td>
<td>$30,366</td>
</tr>
<tr>
<td>12</td>
<td>Audi</td>
<td>A4 3.0 Quattro</td>
<td>Sedan</td>
<td>Europe</td>
<td>$31,388</td>
</tr>
<tr>
<td>13</td>
<td>Audi</td>
<td>A6 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$33,129</td>
</tr>
<tr>
<td>14</td>
<td>Audi</td>
<td>A6 3.0 Quattro</td>
<td>Sedan</td>
<td>Europe</td>
<td>$35,792</td>
</tr>
<tr>
<td>15</td>
<td>Audi</td>
<td>A4 3.0 convert</td>
<td>Sedan</td>
<td>Europe</td>
<td>$38,325</td>
</tr>
</tbody>
</table>

### Output 10.34  Monospace Style

<table>
<thead>
<tr>
<th>Obs</th>
<th>Make</th>
<th>Model</th>
<th>Type</th>
<th>Origin</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acura</td>
<td>MDX</td>
<td>SUV</td>
<td>Asia</td>
<td>$33,337</td>
</tr>
<tr>
<td>2</td>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$21,761</td>
</tr>
<tr>
<td>3</td>
<td>Acura</td>
<td>TSX 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$24,647</td>
</tr>
<tr>
<td>4</td>
<td>Acura</td>
<td>TL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$30,299</td>
</tr>
<tr>
<td>5</td>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$39,014</td>
</tr>
<tr>
<td>6</td>
<td>Acura</td>
<td>3.5 RL w/Nav</td>
<td>Sedan</td>
<td>Asia</td>
<td>$41,100</td>
</tr>
<tr>
<td>7</td>
<td>Acura</td>
<td>NSX coupe 2dr</td>
<td>Sports</td>
<td>Asia</td>
<td>$79,978</td>
</tr>
<tr>
<td>8</td>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$23,508</td>
</tr>
<tr>
<td>9</td>
<td>Audi</td>
<td>A4 1.8T convert</td>
<td>Sedan</td>
<td>Europe</td>
<td>$32,506</td>
</tr>
<tr>
<td>10</td>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$38,846</td>
</tr>
<tr>
<td>11</td>
<td>Audi</td>
<td>A4 3.0 Quattro</td>
<td>Sedan</td>
<td>Europe</td>
<td>$30,366</td>
</tr>
<tr>
<td>12</td>
<td>Audi</td>
<td>A4 3.0 Quattro</td>
<td>Sedan</td>
<td>Europe</td>
<td>$31,388</td>
</tr>
<tr>
<td>13</td>
<td>Audi</td>
<td>A6 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$33,129</td>
</tr>
<tr>
<td>14</td>
<td>Audi</td>
<td>A6 3.0 Quattro</td>
<td>Sedan</td>
<td>Europe</td>
<td>$35,792</td>
</tr>
<tr>
<td>15</td>
<td>Audi</td>
<td>A4 3.0 convert</td>
<td>Sedan</td>
<td>Europe</td>
<td>$38,325</td>
</tr>
</tbody>
</table>
Pearl is the default style for PRINTER output.

### Output 10.35  NormalPrinter Style

<table>
<thead>
<tr>
<th>Obs</th>
<th>Make</th>
<th>Model</th>
<th>Type</th>
<th>Origin</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acura</td>
<td>MDX</td>
<td>SUV</td>
<td>Asia</td>
<td>$33,337</td>
</tr>
<tr>
<td>2</td>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$21,761</td>
</tr>
<tr>
<td>3</td>
<td>Acura</td>
<td>TSX 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$24,647</td>
</tr>
<tr>
<td>4</td>
<td>Acura</td>
<td>TL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$30,269</td>
</tr>
<tr>
<td>5</td>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$39,014</td>
</tr>
<tr>
<td>6</td>
<td>Acura</td>
<td>3.5 RL w/Navigation 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$41,160</td>
</tr>
<tr>
<td>7</td>
<td>Acura</td>
<td>NSX coupe 2dr manual S</td>
<td>Sports</td>
<td>Asia</td>
<td>$79,979</td>
</tr>
<tr>
<td>8</td>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$33,500</td>
</tr>
<tr>
<td>9</td>
<td>Audi</td>
<td>A4 1.8T convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$32,506</td>
</tr>
<tr>
<td>10</td>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$28,846</td>
</tr>
<tr>
<td>11</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr manual</td>
<td>Sedan</td>
<td>Europe</td>
<td>$30,366</td>
</tr>
<tr>
<td>12</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr auto</td>
<td>Sedan</td>
<td>Europe</td>
<td>$37,369</td>
</tr>
<tr>
<td>13</td>
<td>Audi</td>
<td>A6 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$33,129</td>
</tr>
<tr>
<td>14</td>
<td>Audi</td>
<td>A6 3.0 Quattro 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$35,992</td>
</tr>
<tr>
<td>15</td>
<td>Audi</td>
<td>A4 3.0 convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$38,325</td>
</tr>
</tbody>
</table>

### Output 10.36  Pearl Style

<table>
<thead>
<tr>
<th>Obs</th>
<th>Make</th>
<th>Model</th>
<th>Type</th>
<th>Origin</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acura</td>
<td>MDX</td>
<td>SUV</td>
<td>Asia</td>
<td>$33,337</td>
</tr>
<tr>
<td>2</td>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$21,761</td>
</tr>
<tr>
<td>3</td>
<td>Acura</td>
<td>TSX 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$24,647</td>
</tr>
<tr>
<td>4</td>
<td>Acura</td>
<td>TL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$30,269</td>
</tr>
<tr>
<td>5</td>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$39,014</td>
</tr>
<tr>
<td>6</td>
<td>Acura</td>
<td>3.5 RL w/Navigation 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$41,160</td>
</tr>
<tr>
<td>7</td>
<td>Acura</td>
<td>NSX coupe 2dr manual S</td>
<td>Sports</td>
<td>Asia</td>
<td>$79,979</td>
</tr>
<tr>
<td>8</td>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$33,500</td>
</tr>
<tr>
<td>9</td>
<td>Audi</td>
<td>A4 1.8T convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$32,506</td>
</tr>
<tr>
<td>10</td>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$28,846</td>
</tr>
<tr>
<td>11</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr manual</td>
<td>Sedan</td>
<td>Europe</td>
<td>$30,366</td>
</tr>
<tr>
<td>12</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr auto</td>
<td>Sedan</td>
<td>Europe</td>
<td>$37,369</td>
</tr>
<tr>
<td>13</td>
<td>Audi</td>
<td>A6 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$33,129</td>
</tr>
<tr>
<td>14</td>
<td>Audi</td>
<td>A6 3.0 Quattro 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$35,992</td>
</tr>
<tr>
<td>15</td>
<td>Audi</td>
<td>A4 3.0 convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$38,325</td>
</tr>
</tbody>
</table>
### Output 10.37  Sapphire Style

<table>
<thead>
<tr>
<th>Obs</th>
<th>Make</th>
<th>Model</th>
<th>Type</th>
<th>Origin</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acura</td>
<td>MDX</td>
<td>SUV</td>
<td>Asia</td>
<td>$33,337</td>
</tr>
<tr>
<td>2</td>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$21,761</td>
</tr>
<tr>
<td>3</td>
<td>Acura</td>
<td>TSX 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$24,647</td>
</tr>
<tr>
<td>4</td>
<td>Acura</td>
<td>TL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$30,299</td>
</tr>
<tr>
<td>5</td>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$39,014</td>
</tr>
<tr>
<td>6</td>
<td>Acura</td>
<td>3.5 RL w/Navigation 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$41,100</td>
</tr>
<tr>
<td>7</td>
<td>Acura</td>
<td>NSX coupe 2dr manual</td>
<td>Sedan</td>
<td>Asia</td>
<td>$79,978</td>
</tr>
<tr>
<td>8</td>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$23,508</td>
</tr>
<tr>
<td>9</td>
<td>Audi</td>
<td>A4 1.8T convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$32,506</td>
</tr>
<tr>
<td>10</td>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$28,846</td>
</tr>
<tr>
<td>11</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$30,366</td>
</tr>
<tr>
<td>12</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr auto</td>
<td>Sedan</td>
<td>Europe</td>
<td>$31,388</td>
</tr>
<tr>
<td>13</td>
<td>Audi</td>
<td>A6 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$33,129</td>
</tr>
<tr>
<td>14</td>
<td>Audi</td>
<td>A6 3.0 Quattro 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$35,992</td>
</tr>
<tr>
<td>15</td>
<td>Audi</td>
<td>A4 3.0 convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$38,325</td>
</tr>
</tbody>
</table>

### Output 10.38  SasDocPrinter Style

<table>
<thead>
<tr>
<th>Obs</th>
<th>Make</th>
<th>Model</th>
<th>Type</th>
<th>Origin</th>
<th>Invoice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acura</td>
<td>MDX</td>
<td>SUV</td>
<td>Asia</td>
<td>$33,337</td>
</tr>
<tr>
<td>2</td>
<td>Acura</td>
<td>RSX Type S 2dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$21,761</td>
</tr>
<tr>
<td>3</td>
<td>Acura</td>
<td>TSX 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$24,647</td>
</tr>
<tr>
<td>4</td>
<td>Acura</td>
<td>TL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$30,299</td>
</tr>
<tr>
<td>5</td>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$39,014</td>
</tr>
<tr>
<td>6</td>
<td>Acura</td>
<td>3.5 RL w/Navigation 4dr</td>
<td>Sedan</td>
<td>Asia</td>
<td>$41,100</td>
</tr>
<tr>
<td>7</td>
<td>Acura</td>
<td>NSX coupe 2dr manual</td>
<td>Sedan</td>
<td>Asia</td>
<td>$79,978</td>
</tr>
<tr>
<td>8</td>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$23,508</td>
</tr>
<tr>
<td>9</td>
<td>Audi</td>
<td>A4 1.8T convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$32,506</td>
</tr>
<tr>
<td>10</td>
<td>Audi</td>
<td>A4 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$28,846</td>
</tr>
<tr>
<td>11</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$30,366</td>
</tr>
<tr>
<td>12</td>
<td>Audi</td>
<td>A4 3.0 Quattro 4dr auto</td>
<td>Sedan</td>
<td>Europe</td>
<td>$31,388</td>
</tr>
<tr>
<td>13</td>
<td>Audi</td>
<td>A6 3.0 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$33,129</td>
</tr>
<tr>
<td>14</td>
<td>Audi</td>
<td>A6 3.0 Quattro 4dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$35,992</td>
</tr>
<tr>
<td>15</td>
<td>Audi</td>
<td>A4 3.0 convertible 2dr</td>
<td>Sedan</td>
<td>Europe</td>
<td>$38,325</td>
</tr>
</tbody>
</table>
**Output 10.39  Seaside Printer Style**

![Seaside Printer Style](image)

**Styles for the ODS Destination for PowerPoint**

PowerPointLight is the default style for output created by the ODS destination for PowerPoint.

**Output 10.40  PowerPointLight Style**

![PowerPointLight Style](image)
### Excel Style

#### Output 10.42  Excel Style

<table>
<thead>
<tr>
<th>Region</th>
<th>Division</th>
<th>Product type</th>
<th>Total Predicted Sales</th>
<th>Total Actual Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>$11,081</td>
<td>$12,483</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFFICE</td>
<td>$21,939</td>
<td>$16,991</td>
</tr>
<tr>
<td></td>
<td>EDUCATION</td>
<td>FURNITURE</td>
<td>$12,972</td>
<td>$14,467</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFFICE</td>
<td>$16,434</td>
<td>$20,189</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Division Total</td>
<td>$62,426</td>
<td>$64,130</td>
</tr>
<tr>
<td>WEST</td>
<td>CONSUMER</td>
<td>FURNITURE</td>
<td>$10,286</td>
<td>$10,380</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFFICE</td>
<td>$16,042</td>
<td>$16,371</td>
</tr>
<tr>
<td></td>
<td>EDUCATION</td>
<td>FURNITURE</td>
<td>$12,816</td>
<td>$11,234</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFFICE</td>
<td>$17,759</td>
<td>$18,905</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Division Total</td>
<td>$56,903</td>
<td>$56,890</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Grand Total</td>
<td>$119,329</td>
<td>$121,020</td>
</tr>
</tbody>
</table>
Chapter 11

Style Elements

General ODS Style Elements

The following table lists all the style elements available for ODS style definitions. The table provides a brief description of each style element and indicates the style elements from which it inherits its attributes. An abstract style element is one that is not used to generate any style element but provides a parent for one or more style elements to inherit.

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miscellaneous</td>
<td>Controls all container-oriented elements</td>
<td>Container *</td>
</tr>
<tr>
<td>Container *</td>
<td>Controls continued flag when a table breaks across a page (paginated destinations only)</td>
<td>TitlesAndFooters</td>
</tr>
<tr>
<td>Continued</td>
<td>Controls page numbers for paginated destinations</td>
<td>TitlesAndFooters</td>
</tr>
<tr>
<td>ExtendedPage</td>
<td>Message when page won't fit (Printer only)</td>
<td>TitlesAndFooters</td>
</tr>
<tr>
<td>(PageNo)</td>
<td>Controls space between tables in RTF output</td>
<td>TitlesAndFooters</td>
</tr>
<tr>
<td>PrePage</td>
<td>Controls the ODS RTF/MEASURED PREPAGE= style</td>
<td>TitlesAndFooters</td>
</tr>
</tbody>
</table>
## Style Elements

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Miscellaneous</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StartUpFunction</td>
<td>This is a Javascript function that is added to the HTML output. Any Javascript code in the TAGATTR= attribute is executed when the page is loaded.</td>
<td></td>
</tr>
<tr>
<td>ShutDownFunction</td>
<td>Controls the Shut-Down function. This is a Javascript function that is added to the HTML output. Any Javascript code in the TAGATTR= attribute is executed when the page is exited.</td>
<td></td>
</tr>
<tr>
<td>UserText</td>
<td>Controls the ODS TEXT= style</td>
<td>Note</td>
</tr>
</tbody>
</table>

* An abstract style element. Abstract elements are not explicitly used in the ODS output. They are used for inheritance purposes only. Because of this, abstract styles will not appear in the output of destinations that generate a style sheet.

### Table 11.2 Style Elements Affecting Documents

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Documents</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document</td>
<td>Controls the various document bodies. This generally includes things like the page background color and page margins.</td>
<td>Container *</td>
</tr>
<tr>
<td>Body</td>
<td>Controls the Body file</td>
<td>Document</td>
</tr>
<tr>
<td>Frame</td>
<td>Controls the Frame file for HTML</td>
<td>Document</td>
</tr>
<tr>
<td>Contents</td>
<td>Controls the Contents file</td>
<td>Document</td>
</tr>
<tr>
<td>Pages</td>
<td>Controls the Page file</td>
<td>Document</td>
</tr>
</tbody>
</table>

* An abstract style element. Abstract elements are not explicitly used in the ODS output. They are used for inheritance purposes only. Because of this, abstract styles will not appear in the output of destinations that generate a style sheet.
### Table 11.3  Style Elements Affecting Dates

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td>BodyDate</td>
<td>Controls the date field in the Contents file</td>
<td>ContentsDate</td>
</tr>
<tr>
<td>Date</td>
<td>Controls how date fields look</td>
<td>Container*</td>
</tr>
<tr>
<td>PagesDate</td>
<td>Controls the date field in the Pages file</td>
<td>Date</td>
</tr>
</tbody>
</table>

* An abstract style element. Abstract elements are not explicitly used in the ODS output. They are used for inheritance purposes only. Because of this, abstract styles will not appear in the output of destinations that generate a style sheet.

### Table 11.4  Style Elements Affecting Table of Contents and Table of Pages

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndexItem</td>
<td>Controls list items and folders for Contents and Pages</td>
<td>Container*</td>
</tr>
<tr>
<td>ContentFolder</td>
<td>Controls the folders in the Contents file</td>
<td>IndexItem</td>
</tr>
<tr>
<td>ByContentFolder</td>
<td>Controls the byline folders in the Contents file</td>
<td>ContentFolder</td>
</tr>
<tr>
<td>ContentItem</td>
<td>Controls the items in the Contents file</td>
<td>IndexItem</td>
</tr>
<tr>
<td>PagesItem</td>
<td>Controls the items in the Pages file</td>
<td>IndexItem</td>
</tr>
<tr>
<td>Index</td>
<td>Controls miscellaneous Contents and Pages components</td>
<td>Container*</td>
</tr>
<tr>
<td>IndexProcName</td>
<td>Controls the PROC name in the Contents and Pages files</td>
<td>Index*</td>
</tr>
<tr>
<td>ContentProcName</td>
<td>Controls the PROC name in the Contents file</td>
<td>IndexProcName</td>
</tr>
<tr>
<td>ContentProcLabel</td>
<td>Controls the PROC label in the Contents file</td>
<td>ContentProcName</td>
</tr>
<tr>
<td>PagesProcName</td>
<td>Controls the PROC name in the Pages file</td>
<td>IndexProcName</td>
</tr>
</tbody>
</table>
An abstract style element. Abstract elements are not explicitly used in the ODS output. They are used for inheritance purposes only. Because of this, abstract styles will not appear in the output of destinations that generate a style sheet.

### Table 11.5  *Style Elements Affecting Titles and Footers*

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Titles and Footers</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SysTitleAndFooterContainer</td>
<td>Controls the container for system page title and system page footer. This element is usually used to add borders around a title.</td>
<td>Container</td>
</tr>
<tr>
<td>TitlesAndFooters</td>
<td>Controls system page title text and system page footer text</td>
<td>Container*</td>
</tr>
<tr>
<td>SystemTitle</td>
<td>Controls system title text</td>
<td>TitlesAndFooters</td>
</tr>
<tr>
<td>SystemTitle2</td>
<td>Controls system title2 text</td>
<td>SystemTitle</td>
</tr>
<tr>
<td>SystemTitle3</td>
<td>Controls system title3 text</td>
<td>SystemTitle2</td>
</tr>
<tr>
<td>SystemTitle4</td>
<td>Controls system title4 text</td>
<td>SystemTitle3</td>
</tr>
<tr>
<td>SystemTitle5</td>
<td>Controls system title5 text</td>
<td>SystemTitle4</td>
</tr>
<tr>
<td>SystemTitle6</td>
<td>Controls system title6 text</td>
<td>SystemTitle5</td>
</tr>
<tr>
<td>SystemTitle7</td>
<td>Controls system title7 text</td>
<td>SystemTitle6</td>
</tr>
</tbody>
</table>
An abstract style element. Abstract elements are not explicitly used in the ODS output. They are used for inheritance purposes only. Because of this, abstract styles will not appear in the output of destinations that generate a style sheet.

**Table 11.6  ** Style Elements Affecting Procedure Titles

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC Titles</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TitleAndNoteContainer</td>
<td>Controls the container for procedure-defined titles and notes</td>
<td>Container</td>
</tr>
<tr>
<td>ProcTitle</td>
<td>Controls procedure title text</td>
<td>TitlesAndFooters</td>
</tr>
<tr>
<td>ProcTitleFixed</td>
<td>Controls procedure title text that requests a fixed font</td>
<td>ProcTitle</td>
</tr>
</tbody>
</table>
### Table 11.7 Style Elements Affecting Bylines

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bylines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BylineContainer</td>
<td>Controls the container for the byline. This is generally used to add borders to a byline.</td>
<td>Container</td>
</tr>
<tr>
<td>Byline</td>
<td>Controls byline text</td>
<td>TitlesAndFooters</td>
</tr>
</tbody>
</table>

### Table 11.8 Style Elements Affecting Notes, Warnings, and Errors

Notes, warnings, and errors consist of two pieces: a banner area and a content area as shown in the following diagram. The Banner elements generally print the content of the banner (that is, "NOTE: ", "WARNING: ", and so on) using the PRETEXT= attribute.

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notes, Warnings, and Errors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td>Controls the container for note banners and note contents</td>
<td>Container*</td>
</tr>
<tr>
<td>NoteBanner</td>
<td>Controls the banner for NOTE:s</td>
<td>Note</td>
</tr>
<tr>
<td>NoteContent</td>
<td>Controls the contents for NOTE:s</td>
<td>Note</td>
</tr>
<tr>
<td>NoteContentFixed</td>
<td>Controls the contents for NOTE:s. Fixed font.</td>
<td>NoteContent</td>
</tr>
<tr>
<td>WarnBanner</td>
<td>Controls the banner for WARNING:s</td>
<td>Note</td>
</tr>
<tr>
<td>WarnContent</td>
<td>Controls the contents of WARNING:s</td>
<td>Note</td>
</tr>
<tr>
<td>WarnContentFixed</td>
<td>Controls the contents for WARNING:s. Fixed font.</td>
<td>WarnContent</td>
</tr>
<tr>
<td>ErrorBanner</td>
<td>Controls the banner for ERROR:s</td>
<td>Note</td>
</tr>
<tr>
<td>ErrorContent</td>
<td>Controls the contents of ERROR:s</td>
<td>Note</td>
</tr>
<tr>
<td>ErrorContentFixed</td>
<td>Controls the contents for ERROR:s. Fixed font.</td>
<td>ErrorContent</td>
</tr>
<tr>
<td>Style Element</td>
<td>Description</td>
<td>Inherits from</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td><strong>FatalBanner</strong></td>
<td>Controls the banner for FATAL:s</td>
<td>Note</td>
</tr>
<tr>
<td><strong>FatalContent</strong></td>
<td>Controls the contents of FATAL:s</td>
<td>Note</td>
</tr>
<tr>
<td><strong>FatalContentFixed</strong></td>
<td>Controls the contents for FATAL:s. Fixed font.</td>
<td>FatalContent</td>
</tr>
</tbody>
</table>

* An abstract style element. Abstract elements are not explicitly used in the ODS output. They are used for inheritance purposes only. Because of this, abstract styles will not appear in the output of destinations that generate a style sheet.

### Table 11.9  Style Elements Affecting Tables and Batch Output

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output</strong></td>
<td>Controls basic output forms. This is generally used to control the borders (using the FRAME=, RULES=, and individual border control attributes), cell spacing, cell padding, and background color.</td>
<td>Container*</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>Controls overall table style</td>
<td>Output</td>
</tr>
<tr>
<td><strong>Batch</strong></td>
<td>Controls batch mode output</td>
<td>Output</td>
</tr>
<tr>
<td><strong>TableHeaderContainer</strong></td>
<td>Places and controls the box around all column headings (RTF only)</td>
<td>Container*</td>
</tr>
<tr>
<td><strong>TableFooterContainer</strong></td>
<td>Places and controls the box around all column footers (RTF only)</td>
<td>Container*</td>
</tr>
</tbody>
</table>
An abstract style element. Abstract elements are not explicitly used in the ODS output. They are used for inheritance purposes only. Because of this, abstract styles will not appear in the output of destinations that generate a style sheet.

**Table 11.10  Style Elements Affecting Data Cells in Tables**

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table Data Cells</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cell</td>
<td>Controls data, header, and footer cells</td>
<td>Container*</td>
</tr>
<tr>
<td>Data</td>
<td>Default style for data cells</td>
<td>Cell</td>
</tr>
<tr>
<td>DataFixed</td>
<td>Default style for data cells that request a fixed font</td>
<td>Data</td>
</tr>
<tr>
<td>DataEmpty</td>
<td>Controls emphasized data cells</td>
<td>Data</td>
</tr>
<tr>
<td>DataEmphasis</td>
<td>Controls emphasized data cells</td>
<td>Data</td>
</tr>
<tr>
<td>DataEmphasisFixed</td>
<td>Controls emphasized data cells that request a fixed font</td>
<td>DataEmphasis</td>
</tr>
<tr>
<td>Style Element</td>
<td>Description</td>
<td>Inherits from</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Table Data Cells</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataStrong</td>
<td>Controls strong (more emphasized) data cells</td>
<td>Data</td>
</tr>
<tr>
<td>DataStrongFixed</td>
<td>Controls strong (more emphasized) data cells that request a fixed font</td>
<td>DataStrong</td>
</tr>
</tbody>
</table>

* An abstract style element. Abstract elements are not explicitly used in the ODS output. They are used for inheritance purposes only. Because of this, abstract styles will not appear in the output of destinations that generate a style sheet.

### Table 11.11  Style Elements Affecting Header and Footer Cells

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Header and Footer Cells</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HeadersAndFooters</td>
<td>Controls table headers and footers</td>
<td>Cell*</td>
</tr>
<tr>
<td>Header</td>
<td>Controls the headers of a table</td>
<td>HeadersAndFooters</td>
</tr>
<tr>
<td>HeaderFixed</td>
<td>Controls the header of a table that request a fixed font</td>
<td>Header</td>
</tr>
<tr>
<td>HeaderEmpty</td>
<td>Controls empty table header cells</td>
<td>Header</td>
</tr>
<tr>
<td>HeaderEmphasis</td>
<td>Controls emphasized table header cells that request a fixed font</td>
<td>Header</td>
</tr>
<tr>
<td>HeaderEmphasisFixed</td>
<td>Controls emphasized table header cells that request a fixed font</td>
<td>HeaderEmphasis</td>
</tr>
<tr>
<td>HeaderStrong</td>
<td>Controls strong (more emphasized) table header cells</td>
<td>Header</td>
</tr>
<tr>
<td>HeaderStrongFixed</td>
<td>Controls strong (more emphasized) table header cells</td>
<td>Header</td>
</tr>
<tr>
<td>RowHeader</td>
<td>Controls row headers</td>
<td>Header</td>
</tr>
<tr>
<td>RowHeaderFixed</td>
<td>Controls row headers that request a fixed font</td>
<td>RowHeader</td>
</tr>
<tr>
<td>RowHeaderEmpty</td>
<td>Controls empty row headers</td>
<td>RowHeader</td>
</tr>
<tr>
<td>RowHeaderEmphasis</td>
<td>Controls emphasized row headers</td>
<td>RowHeader</td>
</tr>
<tr>
<td>Style Element</td>
<td>Description</td>
<td>Inherits from</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td><strong>Table Header and Footer Cells</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RowHeaderEmphasisFixed</td>
<td>Controls emphasized row headers that request a fixed font</td>
<td>RowHeaderEmphasis</td>
</tr>
<tr>
<td>RowHeaderStrong</td>
<td>Controls strong (more emphasized) row headers</td>
<td>RowHeader</td>
</tr>
<tr>
<td>RowHeaderStrongFixed</td>
<td>Controls strong (more emphasized) row headers that request a fixed font</td>
<td>RowHeaderStrong</td>
</tr>
<tr>
<td>Footer</td>
<td>Controls table footers</td>
<td>HeadersAndFooters</td>
</tr>
<tr>
<td>FooterFixed</td>
<td>Controls table footers that request a fixed font</td>
<td>Footer</td>
</tr>
<tr>
<td>FooterEmpty</td>
<td>Controls empty table footers</td>
<td>Footer</td>
</tr>
<tr>
<td>FooterEmphasis</td>
<td>Controls emphasized table footers</td>
<td>Footer</td>
</tr>
<tr>
<td>FooterEmphasisFixed</td>
<td>Controls emphasized table footers that request a fixed font</td>
<td>FooterEmphasis</td>
</tr>
<tr>
<td>FooterStrong</td>
<td>Controls strong (more emphasized) table footers</td>
<td>Footer</td>
</tr>
<tr>
<td>FooterStrongFixed</td>
<td>Controls strong (more emphasized) table footers that request a fixed font</td>
<td>FooterStrong</td>
</tr>
<tr>
<td>RowFooter</td>
<td>Controls a row footer (label)</td>
<td>Footer</td>
</tr>
<tr>
<td>RowFooterFixed</td>
<td>Controls a row footer (label) that request a fixed font</td>
<td>RowFooter</td>
</tr>
<tr>
<td>RowFooterEmpty</td>
<td>Controls an empty row footer (label)</td>
<td>RowFooter</td>
</tr>
<tr>
<td>RowFooterEmphasis</td>
<td>Controls an emphasized row footer (label)</td>
<td>RowFooter</td>
</tr>
<tr>
<td>RowFooterEmphasisFixed</td>
<td>Controls an emphasized row footer (label) that request a fixed font</td>
<td>RowFooterEmphasis</td>
</tr>
<tr>
<td>RowFooterStrong</td>
<td>Controls a strong (more emphasized) row footer (label)</td>
<td>RowFooter</td>
</tr>
</tbody>
</table>
Style Elements Affecting Template-Based Graphics

The following style elements affect template-based graphics and can be specified by Graph Template Language appearance options or used in styles. Template-based graphics include all SAS/GRAPH output where a compiled ODS template of type STATGRAPH is used to produce graphical output. Supplied templates are stored in Sashelp.Tmplmst. Device drivers and some global statements such as SYMBOL, PATTERN, AXIS, and LEGEND have no affect on this form of graphics. Common SAS/GRAPH procedures that produce template-based graphics are SGPLOT, SGPANEL, SGSCATTER, and SGRENDER in addition to many SAS/STAT, SAS/ETS, and SAS/QC procedures. ODS graphics always produce output as image files and use the ODS GRAPHICS statement to control the graphical environment.

Certain style elements were created to be used with specific plots or graphs. For example, the style element GraphFit2 is best used to modify secondary fit lines. The style element GraphConfidence2 was created to modify secondary confidence bands. The table below lists each style element, the portion of the graph that it affects or was created to use with, and the default attribute values. Attribute values can be changed with PROC TEMPLATE, as stated above.
For complete documentation on the style attributes that can be specified in each style element, see “Style Attributes Overview” in *SAS Output Delivery System: Procedures Guide*.

**Table 11.13  Graph Style Elements: General Graph Appearance**

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Portion of Graph Affected</th>
<th>Recognized Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph</td>
<td>Graph size and outer border appearance</td>
<td>OutputWidth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OutputHeight</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BorderColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BorderWidth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CellPadding</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CellSpacing</td>
</tr>
<tr>
<td>GraphAnnoLine</td>
<td>Annotation lines</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td>GraphAnnoShape</td>
<td>Annotation closed shapes such as circles, and squares</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transparency</td>
</tr>
<tr>
<td>GraphAnnoText</td>
<td>Annotation text</td>
<td>Font</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or font-attributes*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td>GraphAxisLines</td>
<td>X, Y and Z axis lines</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TickDisplay</td>
</tr>
<tr>
<td>GraphBackground</td>
<td>Background of the graph</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transparency</td>
</tr>
<tr>
<td>GraphBorderLines</td>
<td>Border around graph wall, legend border, borders to complete axis frame</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td>GraphDataText</td>
<td>Text font and color for point and line labels</td>
<td>Font</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or font-attributes*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Color</td>
</tr>
<tr>
<td>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Recognized Attributes</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>GraphFootnoteText</td>
<td>Text font and color for footnote(s)</td>
<td>Font or font-attributes* Color</td>
</tr>
<tr>
<td>GraphGridLines</td>
<td>Horizontal and vertical grid lines drawn at major tick marks</td>
<td>Color ContrastColor DisplayOpts LineStyle LineThickness</td>
</tr>
<tr>
<td>GraphHeaderBackground</td>
<td>Background color of the legend title</td>
<td>Color FrameBorder</td>
</tr>
<tr>
<td>GraphLabelText</td>
<td>Text font and color for axis labels and legend titles</td>
<td>Font or font-attributes* Color</td>
</tr>
<tr>
<td>GraphLegendBackground</td>
<td>Background color of the legend</td>
<td>Color FrameBorder</td>
</tr>
<tr>
<td>GraphMinorGridLines</td>
<td>Appearance of the grid lines.</td>
<td>ContrastColor DisplayOpts LineStyle LineThickness</td>
</tr>
<tr>
<td>GraphOutlines</td>
<td>Outline properties for fill areas such as bars, pie slices, box plots, ellipses, and histograms</td>
<td>Color ContrastColor LineStyle LineThickness</td>
</tr>
<tr>
<td>GraphReference</td>
<td>Horizontal and vertical reference lines and drop lines</td>
<td>ContrastColor LineStyle LineThickness</td>
</tr>
<tr>
<td>GraphTitleText</td>
<td>Text font and color for title(s)</td>
<td>Font or font-attributes* Color</td>
</tr>
<tr>
<td>GraphUnicodeText</td>
<td>Text font for Unicode values</td>
<td>Font or font-attributes* Color</td>
</tr>
<tr>
<td>GraphValueText</td>
<td>Text font and color for axis tick values and legend values</td>
<td>Font or font-attributes* Color</td>
</tr>
<tr>
<td>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Recognized Attributes</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>GraphWalls</td>
<td>Vertical wall(s) bounded by axes</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FrameBorder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ContrastColor</td>
</tr>
<tr>
<td>GraphBoxMean</td>
<td>Marker for mean</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td>GraphBoxMedian</td>
<td>Line for median</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td>GraphBoxWhisker</td>
<td>Box whiskers and serifs</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td>GraphConfidence</td>
<td>Primary confidence lines and bands,</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td>colors for bands and lines</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td>GraphConfidence2</td>
<td>Secondary confidence lines and bands,</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td>color for bands, and contrast color for</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td>lines</td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td>GraphConnectLine</td>
<td>Line for connecting boxes or bars</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
</tbody>
</table>

* Font-attributes can be one of the following: FONTFAMILY=, FONTSIZE=, FONTSTYLE=, FONTWEIGHT=.

Table 11.14 Style Elements Affecting Graphical Data Representation
<table>
<thead>
<tr>
<th>Style Element</th>
<th>Portion of Graph Affected</th>
<th>Recognized Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraphDataDefault</td>
<td>Primitives related to non-grouped data items, colors for filled areas, markers, and lines</td>
<td>ContrastColor, Color, MarkerSize, MarkerSymbol, LineStyle, LineThickness, StartColor, NeutralColor, EndColor</td>
</tr>
<tr>
<td>GraphCutLine</td>
<td>Cutline attributes for a dendogram</td>
<td>Color, LineStyle</td>
</tr>
<tr>
<td>GraphDataDefault</td>
<td>Primitives related to non-grouped data items, colors for filled areas, markers, and lines</td>
<td>Color, ContrastColor, MarkerSymbol, MarkerSize, LineStyle, LineThickness, StartColor, NeutralColor, EndColor</td>
</tr>
<tr>
<td>GraphError</td>
<td>Error line or error bar fill, ContrastColor for lines, Color for bar fill</td>
<td>CapStyle, ContrastColor, Color, LineStyle, Transparency</td>
</tr>
<tr>
<td>GraphFit</td>
<td>Primary fit lines such as a normal density curve</td>
<td>ContrastColor, Color, MarkerSize, MarkerSymbol, LineStyle, LineThickness</td>
</tr>
<tr>
<td>GraphFit2</td>
<td>Secondary fit lines such as a kernel density curve</td>
<td>ContrastColor, Color, MarkerSize, MarkerSymbol, LineStyle, LineThickness</td>
</tr>
<tr>
<td>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Recognized Attributes</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>GraphFinal</td>
<td>Final data for the waterfall chart. Color applies to filled areas.</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td>GraphInitial</td>
<td>Initial data for the waterfall chart. Color applies to filled areas.</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td>GraphMissing</td>
<td>Properties for graph items representing missing values</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transparency</td>
</tr>
<tr>
<td>GraphOther</td>
<td>Other data for the graph. Color applies to filled areas.</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td>GraphOverflow</td>
<td>Overflow data for the graph. Color applies to filled areas. ContrastColor applies to</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>markers and lines.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td>GraphOutlier</td>
<td>Outlier data for the graph</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Recognized Attributes</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------------------------------------------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>GraphPrediction</td>
<td>Prediction lines</td>
<td>ContrastColor, Color, LineStyle, LineThickness, MarkerSize, MarkerSymbol</td>
</tr>
<tr>
<td>GraphPredictionLimits</td>
<td>Fills for prediction limits</td>
<td>ContrastColor, Color, MarkerSize, MarkerSymbol</td>
</tr>
<tr>
<td>GraphUnderflow</td>
<td>Underflow data for the graph. Color applies to filled areas. ContrastColor applies to markers and lines.</td>
<td>Color, ContrastColor, LineStyle, LineThickness, MarkerSize, MarkerSymbol, TextColor</td>
</tr>
<tr>
<td>GraphSelection</td>
<td>For interactive graphs, visual properties of selected item. Color for selected fill area, ContrastColor for selected marker or line.</td>
<td>ContrastColor, Color, MarkerSymbol, MarkerSize, LineStyle, LineThickness</td>
</tr>
<tr>
<td>ThreeColorAltRamp</td>
<td>Line contours, markers, and data labels with segmented range color response</td>
<td>StartColor, NeutralColor, EndColor</td>
</tr>
<tr>
<td>ThreeColorRamp</td>
<td>Gradient contours, surfaces, markers, and data labels with continuous color response</td>
<td>StartColor, NeutralColor, EndColor</td>
</tr>
<tr>
<td>TwoColorAltRamp</td>
<td>Line contours, markers, and data labels with segmented range color response</td>
<td>StartColor, EndColor</td>
</tr>
<tr>
<td>TwoColorRamp</td>
<td>Gradient contours, surfaces, markers, and data labels with continuous color response</td>
<td>StartColor, EndColor</td>
</tr>
</tbody>
</table>
### Table 11.15  Graphical Style Elements: Data Related (Grouped)

<table>
<thead>
<tr>
<th>Style Elements</th>
<th>Portion of Graph Affected</th>
<th>Recognized Attributes</th>
</tr>
</thead>
</table>
| GraphData1     | Primitives related to the first 7 grouped data items. Color applies to filled areas. ContrastColor applies to markers and lines. | Color  
ContrastColor  
FillPattern*  
LineStyle  
MarkerSymbol |
| GraphData2     |                          |                       |
| GraphData3     |                          |                       |
| GraphData4     |                          |                       |
| GraphData5     |                          |                       |
| GraphData6     |                          |                       |
| GraphData7     |                          |                       |
| GraphData8     | Primitives related to the 8th through 11th grouped data items. | Color  
ContrastColor  
FillPattern*  
LineStyle |
| GraphData9     |                          |                       |
| GraphData10    |                          |                       |
| GraphData11    |                          |                       |
| GraphData12    | Primitives related to the 12th grouped data item. | Color  
ContrastColor  
FillPattern* |
| GraphData13*   | Primitives related to the 13th through 15th grouped data items. | FillPattern |
| GraphData14    |                          |                       |
| GraphData15    |                          |                       |

* Style attribute FillPattern and style elements GraphData13–GraphData15 are available only with the JOURNAL2, JOURNAL3, and MONOCHROMEPRINTER styles.

### Table 11.16  Display Style Elements

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Portion of Graph Affected</th>
<th>Recognized Attributes</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraphAltBlock</td>
<td>Alternate fill color for block plots</td>
<td>Color</td>
<td>GraphColors(&quot;gablock&quot;)</td>
</tr>
<tr>
<td>GraphBand</td>
<td>Display options for confidence bands</td>
<td>DisplayOpts</td>
<td>&quot;Fill&quot;</td>
</tr>
<tr>
<td>GraphBar</td>
<td>Display options for bar charts</td>
<td>DisplayOpts</td>
<td>&quot;Fill outline&quot;</td>
</tr>
</tbody>
</table>
| GraphBox        | Display options for box plots | DisplayOpts  
CapStyle  
Connect | "Fill caps mean"  
"Median outliers"  
"Serif"  
"Mean" |
| GraphBlock      | Fill color for block plots | Color | GraphColors("gblock") |
Style Elements Affecting Device-Based Graphics

Device-based graphics are all SAS/GRAPH output where there is a user-specified or default device (DEVICE= option) that controls certain aspects of the graphical output. Supplied device drivers are stored in the Sashelp.Devices catalog. Examples of device drivers are SASPRTC, GIF, WIN, ACTIVEX, PDF, and SVG. Common SAS/GRAPH procedures that produce device-based graphics are GPLOT, GCHART, and GMAP. Most device-based graphics produce a GRSEG catalog entry as output and use the GOPTIONS statement to control the graphical environment.

For complete documentation on the style attributes that can be specified in each style element, see “Style Attributes Overview” in SAS Output Delivery System: Procedures Guide.

Note: These style elements affect device-based graphics only when the GSTYLE system option is in effect (this is the default for SAS 9.2). If the NOGSTYLE system option is specified, graphs do not use any style information. For more information about the GSTYLE system option, see SAS System Options: Reference.

Table 11.17  Device-Based Graph Style Elements: General Graph Appearance

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Portion of Graph Affected</th>
<th>Recognized Attributes</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>DropShadowStyle</td>
<td>Used with text types</td>
<td>Color</td>
<td></td>
</tr>
</tbody>
</table>

Table 11.17  Device-Based Graph Style Elements: General Graph Appearance

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Portion of Graph Affected</th>
<th>Recognized Attributes</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraphEllipse</td>
<td>Display options for confidence ellipses</td>
<td>DisplayOpts</td>
<td>&quot;Outline&quot;</td>
</tr>
<tr>
<td>GraphHistogram</td>
<td>Display options for histograms</td>
<td>DisplayOpts</td>
<td>&quot;Fill outline&quot;</td>
</tr>
<tr>
<td>GraphSkins</td>
<td>One or more display features</td>
<td>DataSkin</td>
<td>CRISP, GLOSS, MATTE, NONE, PRESSED</td>
</tr>
<tr>
<td>KpiSkin</td>
<td></td>
<td></td>
<td>BASIC, MODERN, NONE, ONYX, SATIN</td>
</tr>
<tr>
<td>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Recognized Attributes</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------</td>
<td>-----------------------</td>
<td></td>
</tr>
<tr>
<td>Graph</td>
<td>Graph size and outer border appearance</td>
<td>OutputWidth, OutputHeight, BorderColor, BorderWidth, CellPadding, CellSpacing</td>
<td></td>
</tr>
<tr>
<td>GraphAxisLines</td>
<td>X, Y, and Z axis lines</td>
<td>Color, LineStyle, LineThickness</td>
<td></td>
</tr>
<tr>
<td>GraphBackground</td>
<td>Background of the graph</td>
<td>Transparency, BackgroundColor, Gradient_Direction, StartColor, EndColor, BackgroundImage, Image, VerticalAlign, TextAlign</td>
<td></td>
</tr>
<tr>
<td>GraphBorderLines</td>
<td>Border around graph wall, legend border, borders to complete axis frame</td>
<td>Color, LineThickness, LineStyle</td>
<td></td>
</tr>
<tr>
<td>GraphCharts</td>
<td>All charts within the graph</td>
<td>Transparency, BackgroundColor, Gradient_Direction, StartColor, EndColor, BackgroundImage, Image, VerticalAlign, TextAlign</td>
<td></td>
</tr>
<tr>
<td>GraphDataText</td>
<td>Text font and color for point and line labels</td>
<td>Font or font-attributes, Color</td>
<td></td>
</tr>
<tr>
<td>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Recognized Attributes</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>GraphFloor</td>
<td>3D floor</td>
<td>BackgroundColor, Transparency, Gradient_Direction, StartColor, EndColor, BackgroundImage, Image, VerticalAlign, TextAlign</td>
<td></td>
</tr>
<tr>
<td>GraphFootnoteText</td>
<td>Text font and color for footnotes</td>
<td>Font or font-attributes*, Color</td>
<td></td>
</tr>
<tr>
<td>GraphGridLines</td>
<td>Horizontal and vertical grid lines drawn at major tick marks</td>
<td>Color, LineStyle, LineThickness, Transparency, displayopts</td>
<td></td>
</tr>
<tr>
<td>GraphLegendBackground</td>
<td>Background color of the legend</td>
<td>Color, FrameBorder, Transparency</td>
<td></td>
</tr>
<tr>
<td>GraphOutlines</td>
<td>Outline properties for fill areas such as bars, pie slices, and box plots.</td>
<td>Color, LineStyle, LineThickness</td>
<td></td>
</tr>
<tr>
<td>GraphTitle1Text</td>
<td>Text font and color for the first title</td>
<td>Font or font-attributes*, Color</td>
<td></td>
</tr>
<tr>
<td>GraphTitleText</td>
<td>Text font and color for titles subsequent to the first title</td>
<td>Font or font-attributes*, Color</td>
<td></td>
</tr>
<tr>
<td>GraphValueText</td>
<td>Text font and color for axis tick values and legend values</td>
<td>Font or font-attributes*, Color</td>
<td></td>
</tr>
</tbody>
</table>
### Table 11.18 Style Elements Affecting Device-Based Non-Grouped Graphical Data Representation

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Portion of Graph Affected</th>
<th>Default Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraphWalls</td>
<td>Vertical walls bounded by axes</td>
<td>Transparency, BackgroundColor, Gradient_Direction, StartColor, EndColor, BackgroundImage, Image</td>
</tr>
<tr>
<td>GraphCutLine</td>
<td>Cutline attributes for a dendogram</td>
<td>Color, LineStyle</td>
</tr>
<tr>
<td>GraphFinal</td>
<td>Final data for the waterfall chart. Color applies to filled areas.</td>
<td>Color, ContrastColor, LineStyle, LineThickness, MarkerSize, MarkerSymbol, TextColor</td>
</tr>
<tr>
<td>GraphInitial</td>
<td>Initial data for the waterfall chart. Color applies to filled areas.</td>
<td>Color, ContrastColor, LineStyle, LineThickness, MarkerSize, MarkerSymbol, TextColor</td>
</tr>
<tr>
<td>GraphOther</td>
<td>Other data for the graph. Color applies to filled areas.</td>
<td>Color, ContrastColor, LineStyle, LineThickness, MarkerSize, MarkerSymbol, TextColor</td>
</tr>
<tr>
<td>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Default Attributes</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>GraphOverflow</td>
<td>Overflow data for the graph. Color applies to filled areas. ContrastColor applies to markers and lines.</td>
<td>Color&lt;br&gt;ContrastColor&lt;br&gt;LineStyle&lt;br&gt;LineThickness&lt;br&gt;MarkerSize&lt;br&gt;MarkerSymbol&lt;br&gt;TextColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphUnderflow</td>
<td>Underflow data for the graph. Color applies to filled areas. ContrastColor applies to markers and lines.</td>
<td>Color&lt;br&gt;ContrastColor&lt;br&gt;LineStyle&lt;br&gt;LineThickness&lt;br&gt;MarkerSize&lt;br&gt;MarkerSymbol&lt;br&gt;TextColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ThreeColorAltRamp</td>
<td>Line contours, markers, and data labels with segmented range color response</td>
<td>StartColor&lt;br&gt;NeutralColor&lt;br&gt;EndColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ThreeColorRamp</td>
<td>Gradient contours, surfaces, markers, and data labels with continuous color response</td>
<td>StartColor&lt;br&gt;NeutralColor&lt;br&gt;EndColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TwoColorAltRamp</td>
<td>Line contours, markers, and data labels with segmented range color response</td>
<td>StartColor&lt;br&gt;EndColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TwoColorRamp</td>
<td>Gradient contours, surfaces, markers, and data labels with continuous color response</td>
<td>StartColor&lt;br&gt;EndColor</td>
</tr>
<tr>
<td>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Default Attributes</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>GraphData1</td>
<td>Primitives related to 1st grouped data items. Color applies to filled areas. ContrastColor applies to markers and lines.</td>
<td>BackGroundImage, ContrastColor, Color, EndColor, Gradient_Direction, Image, LineStyle, LineThickness, MarkerSize, MarkerSymbol, StartColor</td>
</tr>
<tr>
<td>GraphData2</td>
<td>Primitives related to 2nd grouped data items</td>
<td>BackGroundImage, ContrastColor, Color, EndColor, Gradient_Direction, Image, LineStyle, LineThickness, MarkerSize, MarkerSymbol, StartColor</td>
</tr>
<tr>
<td>GraphData3</td>
<td>Primitives related to 3rd grouped data items</td>
<td>BackGroundImage, ContrastColor, Color, EndColor, Gradient_Direction, Image, LineStyle, LineThickness, MarkerSize, MarkerSymbol, StartColor</td>
</tr>
<tr>
<td>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Default Attributes</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------</td>
<td>-------------------</td>
</tr>
</tbody>
</table>
| GraphData4    | Primitives related to 4th grouped data items | BackGroundImage  
ContrastColor  
Color  
EndColor  
Gradient_Direction  
Image  
LineStyle  
LineThickness  
MarkerSize  
MarkerSymbol  
StartColor |
| GraphData5    | Primitives related to 5th grouped data items | BackGroundImage  
ContrastColor  
Color  
EndColor  
Gradient_Direction  
Image  
LineStyle  
LineThickness  
MarkerSize  
MarkerSymbol  
StartColor |
| GraphData6    | Primitives related to 6th grouped data items | BackGroundImage  
ContrastColor  
Color  
EndColor  
Gradient_Direction  
Image  
LineStyle  
LineThickness  
MarkerSize  
MarkerSymbol  
StartColor |
<table>
<thead>
<tr>
<th>Style Element</th>
<th>Portion of Graph Affected</th>
<th>Default Attributes</th>
</tr>
</thead>
</table>
| GraphData7    | Primitives related to 7th grouped data items | BackGroundImage  
ContrastColor  
Color  
EndColor  
Gradient_Direction  
Image  
LineStyle  
LineThickness  
MarkerSize  
MarkerSymbol  
StartColor |
| GraphData8    | Primitives related to 8th grouped data items | BackGroundImage  
ContrastColor  
Color  
EndColor  
Gradient_Direction  
Image  
LineStyle  
LineThickness  
MarkerSize  
MarkerSymbol  
StartColor |
| GraphData9    | Primitives related to 9th grouped data items | BackGroundImage  
ContrastColor  
Color  
EndColor  
Gradient_Direction  
Image  
LineStyle  
LineThickness  
MarkerSize  
MarkerSymbol  
StartColor |
<table>
<thead>
<tr>
<th>Style Element</th>
<th>Portion of Graph Affected</th>
<th>Default Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraphData10</td>
<td>Primitives related to 10th grouped data items</td>
<td>BackGroundImage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EndColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gradient_Direction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Image</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StartColor</td>
</tr>
<tr>
<td>GraphData11</td>
<td>Primitives related to 11th grouped data items</td>
<td>BackGroundImage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EndColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gradient_Direction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Image</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StartColor</td>
</tr>
<tr>
<td>GraphData12</td>
<td>Primitives related to 12th grouped data items</td>
<td>BackGroundImage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EndColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gradient_Direction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Image</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StartColor</td>
</tr>
</tbody>
</table>
Overview

Style attributes influence the characteristics of individual cells, tables, documents, graphs, and HTML frames. Style attributes exist within style elements and are specified by the STYLE statement or the CLASS statement. The default value for an attribute depends on the style that is in use. For information about styles, style elements, and style attributes, see “Understanding Styles, Style Elements, and Style Attributes” in SAS Output Delivery System: Procedures Guide. For information about using style attributes with ODS Statistical Graphics, see the chapter on controlling the appearance of your graphics in SAS Graph Template Language: User’s Guide.

Style attributes can be supplied by SAS or user-defined. Style attributes can be referenced with a style reference. See “Understanding Style References” in SAS Output Delivery System: Procedures Guide and “style-reference” on page 1041 for more information.

The implementation of an attribute depends on the ODS destination that formats the output. When creating HTML output, the implementation of an attribute depends on the browser that is used. For information about viewing the attributes in a style, see “Viewing the Contents of a Style” in SAS Output Delivery System: Procedures Guide.

For a list of the values that style attributes can specify, see “Style Attribute Values” in SAS Output Delivery System: Procedures Guide. For a list of style elements that you can specify style attributes in, see Chapter 11, “Style Elements,” on page 965.

Style Attributes Tables

For detailed information about these style attributes and their aliases, see “Detailed Information for All Style Attributes” in SAS Output Delivery System: Procedures Guide.
Table 12.1  Table of General Style Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Task</th>
<th>Destinations</th>
<th>Affected Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>**“ABSTRACT=ON</td>
<td>OFF” (p. 1004)**</td>
<td>Specify whether styles used in an HTML document are used in CSS style files</td>
<td>Markup family</td>
</tr>
<tr>
<td><strong>“ACTIVE LINKCOLOR=color” (p. 1004)</strong></td>
<td>Specify the color that a link in an HTML document changes to after you click it, but before the browser opens that file</td>
<td>Markup family</td>
<td>HTML documents</td>
</tr>
<tr>
<td>**“ASIS=ON</td>
<td>OFF” (p. 1004)**</td>
<td>Specify how to handle leading spaces and line breaks in an HTML document</td>
<td>Markup family, printer family, and RTF</td>
</tr>
<tr>
<td><strong>“BACKGROUND COLOR=color” (p. 1005)</strong></td>
<td>Specify the color of the background of tables, table cells, or graphs</td>
<td>Markup family, printer family, and RTF</td>
<td>Table cells, tables, graphs</td>
</tr>
<tr>
<td><strong>“BACKGROUND IMAGE=”string” (p. 1005)</strong></td>
<td>Specify an image to use as the background</td>
<td>Markup family, PCL, PS, and TAGSETS.RTF. For TAGSETS.RTF, only applies to the document.</td>
<td>Table cells, tables, graphs. For TAGSETS.RTF, only applies to the document.</td>
</tr>
<tr>
<td><strong>“BACKGROUND POSITION=position” (p. 1005)</strong></td>
<td>Specify the position of the background of the tables, table cells, or graphs</td>
<td>Markup family, printer family, and RTF</td>
<td>Tables, graphs, and HTML documents</td>
</tr>
<tr>
<td><strong>“BACKGROUND REPEAT=option” (p. 1006)</strong></td>
<td>Specify whether an image is repeated horizontally, vertically, both, or not repeated</td>
<td>Markup family</td>
<td>Individual tables or table cells, graphs</td>
</tr>
<tr>
<td>**“BODY SCROLLBAR=YES</td>
<td>NO</td>
<td>AUTO” (p. 1006)**</td>
<td>Specify whether to put a scroll bar in the frame that references the body file</td>
</tr>
<tr>
<td>**“BODYSIZE=dimension</td>
<td>dimension%</td>
<td><em>” (p. 1007)</em>*</td>
<td>Specify the width of the frame that displays the body file in the HTML frame file</td>
</tr>
<tr>
<td><strong>“BORDER BOTTOM COLOR=color” (p. 1007)</strong></td>
<td>Specify the color of the bottom border of the table</td>
<td>Markup family, printer family, RTF, and Measured RTF</td>
<td>Bottom border of a table or table cell</td>
</tr>
<tr>
<td><strong>“BORDER BOTTOM STYLE=line-style” (p. 1007)</strong></td>
<td>Specify the line style of the bottom border of the selected cell</td>
<td>Markup family, RTF, and Measured RTF</td>
<td>Bottom border of a table or table cell</td>
</tr>
<tr>
<td>Attribute</td>
<td>Task</td>
<td>Destinations</td>
<td>Affected Items</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>“BORDERBOTTOMWIDTH=dimension” (p. 1008)</td>
<td>Specify the width of the bottom border of the table</td>
<td>Markup family, printer family, RTF, and Measured RTF</td>
<td>Bottom border of a table or table cell</td>
</tr>
<tr>
<td>“BORDERCOLLAPSE=COLLAPSE</td>
<td>SEPARATE” (p. 1008)</td>
<td>Specify whether the border is collapsed or separated</td>
<td>Markup family, printer family, RTF, and Measured RTF</td>
</tr>
<tr>
<td>“BORDERCOLOR=color” (p. 1008)</td>
<td>Specify the color of the border in a table or table cell if the border is just one color</td>
<td>Markup family, printer family, RTF, and Measured RTF</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>“BORDERCOLORDARK=color” (p. 1008)</td>
<td>Specify the darker color to use in a border that uses two colors to create a three-dimensional effect</td>
<td>Markup family and printer family</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>“BORDERCOLORLIGHT=color” (p. 1009)</td>
<td>Specify the lighter color to use in a border that uses two colors to create a three-dimensional effect</td>
<td>Markup family and printer family</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>“BORDERLEFTCOLOR=color” (p. 1009)</td>
<td>Specify the color of the left border of a table</td>
<td>Markup family, printer family, RTF, and Measured RTF</td>
<td>Left border of a table or table cell</td>
</tr>
<tr>
<td>“BORDERLEFTSTYLE=line-style” (p. 1009)</td>
<td>Specify the line style of the left border of the specified table cell</td>
<td>Markup family, RTF, and Measured RTF</td>
<td>Left border of the specified table cell</td>
</tr>
<tr>
<td>“BORDERLEFTWIDTH=dimension” (p. 1009)</td>
<td>Specify the width of the left border of the table</td>
<td>Markup family, printer family, RTF, and Measured RTF</td>
<td>Left border of a table or table cell</td>
</tr>
<tr>
<td>“BORDERRIGHTCOLOR=color” (p. 1010)</td>
<td>Specify the color of the right border of the table</td>
<td>Markup family, printer family, RTF, and Measured RTF</td>
<td>Right border of a table or table cell</td>
</tr>
<tr>
<td>“BORDERRIGHTSTYLE=line-style” (p. 1010)</td>
<td>Specify the line style of the right border of the selected cell</td>
<td>Markup family, RTF, and Measured RTF</td>
<td>Right border of the selected cell</td>
</tr>
<tr>
<td>“BORDERRIGHTWIDTH=dimension” (p. 1010)</td>
<td>Specify the width of the right border of the table</td>
<td>Markup family, printer family, RTF, and Measured RTF</td>
<td>Right border of a table</td>
</tr>
<tr>
<td>“BORDERSPACING=dimension” (p. 1010)</td>
<td>Specify the thickness of the spacing between cells in a table</td>
<td>Markup family, RTF, and printer family</td>
<td>Tables</td>
</tr>
<tr>
<td>“BORDERTOPCOLOR=color” (p. 1011)</td>
<td>Specify the color of the top border of the table</td>
<td>Markup family, printer family, RTF, and Measured RTF</td>
<td>Top border of a table or table cell</td>
</tr>
<tr>
<td>Attribute</td>
<td>Task</td>
<td>Destinations</td>
<td>Affected Items</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>“BORDERTOPSTYLE=\line-style” (p. 1011)</td>
<td>Specify the line style of the top border of the specified table cell</td>
<td>Markup family, RTF, and Measured RTF</td>
<td>Top border of the specified table cell</td>
</tr>
<tr>
<td>“BORDERTOPWIDTH=dimension” (p. 1012)</td>
<td>Specify the width of the top border of the table</td>
<td>Markup family, printer family, RTF, and Measured RTF</td>
<td>Top border of a table</td>
</tr>
<tr>
<td>“BORDERWIDTH=dimension” (p. 1012)</td>
<td>Specify the width of the border of the table</td>
<td>Markup family, RTF, and printer family</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>“CELLPADDING=dimension</td>
<td>dimension%” (p. 1012)</td>
<td>Specify the amount of white space on each of the four sides of the content in a table cell</td>
<td>Markup family, RTF, and printer family</td>
</tr>
<tr>
<td>“CLASS=&quot;string&quot;” (p. 1013)</td>
<td>Specify the name of the style sheet class to use in an HTML document for the table or table cell</td>
<td>Markup family</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>“COLOR=color” (p. 1013)</td>
<td>Specify the color of the foreground in tables, table cells, or graphs, which is primarily the color of text</td>
<td>Markup family, RTF, and printer family</td>
<td>Individual tables or table cells, graphs</td>
</tr>
<tr>
<td>“CONTENTPOSITION=position” (p. 1014)</td>
<td>Specify the position, within the frame file, of the frames that display the contents and the page files</td>
<td>Markup family</td>
<td>Individual frames in HTML output</td>
</tr>
<tr>
<td>“CONTENTSCROLLBAR=YES</td>
<td>NO</td>
<td>AUTO” (p. 1014)</td>
<td>Specify whether to put a scroll bar in the frames in the frame file that display the contents and the page files</td>
</tr>
<tr>
<td>“CONTENTSIZE=dimension</td>
<td>dimension %</td>
<td>*” (p. 1015)</td>
<td>Specify the width of the frames in the frame file that display the contents and the page files</td>
</tr>
<tr>
<td>“CONTENTTYPE=&quot;string&quot;” (p. 1015)</td>
<td>Specify the value of the content type for pages in an HTML document that is sent directly to a web server rather than to a file</td>
<td>Markup family</td>
<td>Individual frames in HTML output</td>
</tr>
<tr>
<td>“CONTRASTCOLOR=color” (p. 1015)</td>
<td>Specify the alternate colors for maps</td>
<td>Markup family, RTF, and printer family</td>
<td>Graphs</td>
</tr>
<tr>
<td>Attribute</td>
<td>Task</td>
<td>Destinations</td>
<td>Affected Items</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>&quot;DOCTYPE=&quot;string&quot;&quot; (p. 1017)</td>
<td>Specify the entire doctype declaration for the HTML document</td>
<td>Markup family</td>
<td>HTML documents</td>
</tr>
<tr>
<td>&quot;FILLRULEWIDTH=dimension&quot; (p. 1018)</td>
<td>Place a rule of the specified width into the space around the text (or entire cell if there is no text) in a table where white space would otherwise appear</td>
<td>Printer family</td>
<td>HTML documents</td>
</tr>
<tr>
<td>&quot;FLYOVER=&quot;string&quot;&quot; (p. 1018)</td>
<td>Specify the text to show in a data tip for the table cell</td>
<td>Markup family and PDF</td>
<td>Individual cells</td>
</tr>
<tr>
<td>&quot;FONT=font-definition&quot; (p. 1018)</td>
<td>Specify a font definition to use in tables, table cells, and graphs</td>
<td>Markup family, RTF, and printer family</td>
<td>Individual tables or table cells, graphs</td>
</tr>
<tr>
<td>&quot;FONTFAMILY=&quot;string-1&lt;…, string-n&gt;&quot;&quot; (p. 1019)</td>
<td>Specify the font to use in table cells and graphs</td>
<td>Markup family, RTF, and printer family</td>
<td>Individual tables or table cells, graphs</td>
</tr>
<tr>
<td>&quot;FONTSIZE=dimension</td>
<td>size&quot; (p. 1019)</td>
<td>Specify the size of the font for tables, table cells, and graphs</td>
<td>Markup family, RTF, and printer family</td>
</tr>
<tr>
<td>&quot;FONTSTYLE=ITALIC</td>
<td>ROMAN</td>
<td>SLANT&quot; (p. 1020)</td>
<td>Specify the style of the font for tables, table cells, and graphs</td>
</tr>
<tr>
<td>&quot;FONTWEIGHT=weight&quot; (p. 1020)</td>
<td>Specify the font weight of tables, table cells, and graphs</td>
<td>Markup family, RTF, and printer family</td>
<td>Individual tables or table cells, graphs</td>
</tr>
<tr>
<td>&quot;FONTWIDTH=relative-width&quot; (p. 1021)</td>
<td>Specify the font width of tables, table cells, and graphs compared to the width of the usual design of the table, table cell, or graph</td>
<td>Markup family, RTF, and printer family</td>
<td>Individual tables or table cells, graphs</td>
</tr>
<tr>
<td>&quot;FRAME=frame-type&quot; (p. 1021)</td>
<td>Specify the type of frame to use on a table</td>
<td>Markup family, RTF, and printer family</td>
<td>Tables</td>
</tr>
<tr>
<td>&quot;FRAMEBORDER=ON</td>
<td>OFF&quot; (p. 1022)</td>
<td>Specify whether to put a border around the frame for an HTML file that uses frames</td>
<td>Markup family</td>
</tr>
<tr>
<td>&quot;FRAMEBORDERWIDTH=dimension&quot; (p. 1022)</td>
<td>Specify the width of the border around the frames for an HTML file that uses frames</td>
<td>Markup family</td>
<td>Individual frames in HTML output</td>
</tr>
<tr>
<td>Attribute</td>
<td>Task</td>
<td>Destinations</td>
<td>Affected Items</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>“FRAMESPACING=dimension” (p. 1022)</td>
<td>Specify the width of the space between frames for HTML that uses frames</td>
<td>Markup family</td>
<td>Individual frames in HTML output</td>
</tr>
<tr>
<td>“HEIGHT=dimension” (p. 1022)</td>
<td>Specify the height of a table cell, graph, or graphics in an HTML document</td>
<td>Markup family, RTF, and printer family</td>
<td>Table cells, HTML documents, and graphs</td>
</tr>
<tr>
<td>“HREFTARGET=”target”” (p. 1023)</td>
<td>Specify the window or frame in which to open the target of the link</td>
<td>Markup family</td>
<td>Individual cells</td>
</tr>
<tr>
<td>“HTMLID=”string”” (p. 1024)</td>
<td>Specify an ID for the table or table cell</td>
<td>Markup family</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>“HTMLSTYLE=”string”” (p. 1024)</td>
<td>Specify individual attributes and values for a table or table cell in an HTML document</td>
<td>Markup family</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>“IMAGE=”string”” (p. 1024)</td>
<td>Specify the image to appear in a graph</td>
<td>Markup family, printer family, and RTF</td>
<td>Graphs</td>
</tr>
<tr>
<td>“LINKCOLOR=color” (p. 1025)</td>
<td>Specify the color for the links in an HTML document that have not yet been visited</td>
<td>Markup family, printer family, and RTF</td>
<td>HTML documents</td>
</tr>
<tr>
<td>“LISTENTRYANCHOR=ON</td>
<td>OFF” (p. 1025)</td>
<td>Specify whether to make the entry in the table of contents a link to the body file</td>
<td>Markup family</td>
</tr>
<tr>
<td>“LISTSTYLETYPE=bullet-type” (p. 1025)</td>
<td>Specify the string to use for the bullets in the contents file</td>
<td>Markup family</td>
<td>Individual frames in HTML output</td>
</tr>
<tr>
<td>“MARGINBOTTOM=dimension” (p. 1026)</td>
<td>Specify the bottom margin for the HTML document</td>
<td>Markup family, printer family, and RTF</td>
<td>HTML documents</td>
</tr>
<tr>
<td>“MARGINLEFT=dimension” (p. 1026)</td>
<td>Specify the left margin for the HTML document</td>
<td>Markup family, printer family, and RTF</td>
<td>HTML documents</td>
</tr>
<tr>
<td>“MARGINRIGHT=dimension” (p. 1027)</td>
<td>Specify the right margin for the HTML document</td>
<td>Markup family, printer family, and RTF</td>
<td>HTML documents</td>
</tr>
<tr>
<td>“MARGINTOP=dimension” (p. 1027)</td>
<td>Specify the top margin for the HTML document</td>
<td>Markup family, printer family, and RTF</td>
<td>HTML documents</td>
</tr>
<tr>
<td>Attribute</td>
<td>Task</td>
<td>Destinations</td>
<td>Affected Items</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>“NOBREAKSPACE=ON</td>
<td>OFF ” (p. 1028)</td>
<td>Specify how to handle space characters</td>
<td>Markup family, printer family, and RTF</td>
</tr>
<tr>
<td>“OVERHANGFACTOR=nonnegative-number” (p. 1028)</td>
<td>Specify an upper limit for extending the width of the column in an HTML document</td>
<td>Markup family and printer family</td>
<td>HTML documents</td>
</tr>
<tr>
<td>“PADDING=dimension</td>
<td>dimension%” (p. 1029)</td>
<td>Specify the amount of white space between the content of the table cell and the border</td>
<td>Markup family, RTF, and printer family</td>
</tr>
<tr>
<td>“PADDINGBOTTOM=dimension</td>
<td>dimension %” (p. 1029)</td>
<td>Specify the amount of white space on the bottom of the content of the table cell</td>
<td>Markup family, RTF, and printer family</td>
</tr>
<tr>
<td>“PADDINGLEFT=dimension</td>
<td>dimension%” (p. 1029)</td>
<td>Specify the amount of white space on the left side of the content of the table cell</td>
<td>Markup family, RTF, and printer family</td>
</tr>
<tr>
<td>“PADDINGRIGHT=dimension</td>
<td>dimension%” (p. 1029)</td>
<td>Specify the amount of white space on the right side of the content of the table cell</td>
<td>Markup family, RTF, and printer family</td>
</tr>
<tr>
<td>“PADDINGTOP=dimension</td>
<td>dimension%” (p. 1029)</td>
<td>Specify the amount of white space on the top of the content of the table cell</td>
<td>Markup family, RTF, and printer family</td>
</tr>
<tr>
<td>“PAGEBREAKHTML=&quot;string&quot;” (p. 1029)</td>
<td>Specify HTML to place at page breaks in an HTML document</td>
<td>Markup family</td>
<td>HTML documents</td>
</tr>
<tr>
<td>“POSTHTML=&quot;string”” (p. 1030)</td>
<td>Specify the HTML code to place after the table or table cell</td>
<td>Markup family</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>“POSTIMAGE=&quot;string”</td>
<td>fileref” (p. 1030)</td>
<td>Specify an image to place before the table or table cell</td>
<td>Markup family</td>
</tr>
<tr>
<td>“POSTTEXT=&quot;string”” (p. 1030)</td>
<td>Specify text to place after the table cell or table</td>
<td>Markup family, printer family, and RTF</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>“PREHTML=&quot;string”” (p. 1030)</td>
<td>Specify the HTML code to place before the table or table cell</td>
<td>Markup family</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>Attribute</td>
<td>Task</td>
<td>Destinations</td>
<td>Affected Items</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>&quot;PREIMAGE=&quot;string&quot;</td>
<td>fileref&quot; (p. 1031)</td>
<td>Specify an image to place before the table or table cell</td>
<td>Markup family, printer family, and RTF</td>
</tr>
<tr>
<td>&quot;PRETEXT=&quot;string&quot;&quot; (p. 1031)</td>
<td>Specify text to place before the table cell or table</td>
<td>Markup family, printer family, and RTF</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>&quot;PROTECTSPECIALCHARS=ON</td>
<td>OFF</td>
<td>AUTO&quot; (p. 1031)</td>
<td>Specify how less-than signs (&lt;), greater-than signs (&gt;), and ampersands (&amp;) are interpreted in table cells</td>
</tr>
<tr>
<td>&quot;RULES=rule-type&quot; (p. 1032)</td>
<td>Specify the types of rules to use in tables</td>
<td>Markup family, printer family, and RTF</td>
<td>Tables</td>
</tr>
<tr>
<td>&quot;STARTCOLOR=color&quot; (p. 1032)</td>
<td>Specify the start fill color for a graph</td>
<td>HTML</td>
<td>Graphs</td>
</tr>
<tr>
<td>&quot;TAGATTR=&quot;string&quot;&quot; (p. 1033)</td>
<td>Specify text to insert in the HTML</td>
<td>Markup family</td>
<td>Individual cells</td>
</tr>
<tr>
<td>&quot;TEXTALIGN=alignment&quot; (p. 1033)</td>
<td>Specify justification in tables, table cells, and graphs</td>
<td>Printer family and RTF</td>
<td>Individual tables or table cells, graphs</td>
</tr>
<tr>
<td>&quot;TEXTDECORATION=presentation–options&quot; (p. 1034)</td>
<td>Change the visual presentation of the text</td>
<td>Markup family, RTF, and printer family</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>&quot;TEXTINDENT=n&quot; (p. 1034)</td>
<td>Specify the number of spaces that the first line of output is indented</td>
<td>Markup family, RTF, and printer family</td>
<td>Individual tables or table cells</td>
</tr>
<tr>
<td>&quot;TEXTJUSTIFY=INTER_WORD</td>
<td>INTER_CHARACTER&quot; (p. 1035)</td>
<td>Specify if the words of the text are to be spaced evenly or if the characters are to be evenly justified</td>
<td>HTML, RTF, and TAGSETS.RTF</td>
</tr>
<tr>
<td>&quot;TRANSPARENCY=dimension&quot; (p. 1035)</td>
<td>Specify a transparency level for graphs</td>
<td>HTML</td>
<td>Graphs</td>
</tr>
<tr>
<td>&quot;URL=&quot;uniform-resource-locator&quot;&quot; (p. 1035)</td>
<td>Specify a URL to link to</td>
<td>Markup family, RTF, and printer family</td>
<td>Individual cells</td>
</tr>
<tr>
<td>&quot;VERTICALALIGN=BOTTOM</td>
<td>MIDDLE</td>
<td>TOP &quot; (p. 1035)</td>
<td>Specify vertical justification</td>
</tr>
</tbody>
</table>
Specify the color for links that have been visited in an HTML document

“WATERMARK=ON | OFF” (p. 1036)

Specify whether to make the image that is specified by BACKGROUNDIMAGE= into a "watermark "

“WHITESPACE=options” (p. 1036)

Specify how the browser handles multiple whitespace characters and line breaks.

“WIDTH=dimension ” (p. 1037)

Specify the width of a table cell, table, line, or a graph

Note: You can use the value _UNDEF_ for any style attribute. ODS treats an attribute that is set to _UNDEF_ as if its value had never been set, even in the parent or beyond.

Graphical style attributes can be used in graphical style elements for device-based graphics or template-based graphics (ODS graphics). Different style attributes are valid for different style elements. For a table of style elements and the style attributes that are valid in each one, see “Style Elements Affecting Template-Based Graphics” on page 975 and “Style Elements Affecting Device-Based Graphics” on page 983.

Device-based graphics are all SAS/GRAPH output where there is a user-specified or default device (DEVICE= option) that controls certain aspects of the graphical output. Supplied device drivers are stored in the Sashelp.Devices catalog. Examples of device drivers are SASPRTC, GIF, WIN, ACTIVEX, PDF, and SVG. Common SAS/GRAPH procedures that produce device-based graphics are GPlot, GCHART, and GMAP. Most device-based graphics produce a GRSEG catalog entry as output and use the GOPTIONS statement to control the graphical environment.

Template-based graphics include all SAS/GRAPH output where a compiled ODS template of type STATGRAPH is used to produce graphical output. Supplied templates are stored in Sashelp.Tmplmst. Device drivers and some global statements such as SYMBOL, PATTERN, AXIS, and LEGEND have no affect on this form of graphics. Common SAS/GRAPH procedures that produce template-based graphics are SGPLOT, SGPANEL, and SGRENDER, in addition to many SAS/STAT, SAS/ETS, and SAS/QC procedures. ODS graphics always produce output as image files and use the ODS GRAPHICS statement to control the graphical environment.
**Table 12.2** Table of Graphical Style Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Task</th>
<th>Graphics Environment</th>
<th>Affected Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>“BACKGROUNDIMAGE=”string”“ (p. 1005)</td>
<td>Specify an image file path</td>
<td>Device-based graphics</td>
<td>Image that can be stretched, but not positioned in graph, chart, walls, floor</td>
</tr>
<tr>
<td>“CAPSTYLE=line-shape” (p. 1012)</td>
<td>Specify the shape of the line at the end of a box whisker graph</td>
<td>Template-based graphics</td>
<td>Shape of line at end of box whisker</td>
</tr>
<tr>
<td>“COLOR=color” (p. 1013)</td>
<td>Specify the color of the foreground in tables, table cells, or graphs, which is primarily the color of text</td>
<td>All graphics environments</td>
<td>Background color of the graph, walls, or floor; color of text</td>
</tr>
<tr>
<td>“CONNECT=connect-line-type” (p. 1013)</td>
<td>Specify characteristics of a box plot connect line</td>
<td>Template-based graphics</td>
<td>Box plot connect line</td>
</tr>
<tr>
<td>“CONTRASTCOLOR=color” (p. 1015)</td>
<td>Specify the color a line or marker</td>
<td>Template-based graphics</td>
<td>Color of line or marker</td>
</tr>
<tr>
<td>“DATASKIN=CRISP</td>
<td>GLOSS</td>
<td>MATTE</td>
<td>NONE</td>
</tr>
<tr>
<td>“DISPLAYOPTS=“display-feature”“ (p. 1016)</td>
<td>Specify display features for graphs</td>
<td>Template-based graphics</td>
<td>Displayed features of box plots, ellipses, histograms, bands</td>
</tr>
<tr>
<td>“DROPSSHADOW=ON</td>
<td>OFF ” (p. 1017)</td>
<td>Specify whether the drop shadow color for text is displayed</td>
<td>Device-based graphics</td>
</tr>
<tr>
<td>“ENDCOLOR=color ” (p. 1017)</td>
<td>Specify the final color used with a two- or three-color ramp</td>
<td>All graphics environments</td>
<td>Contours, gradient legends</td>
</tr>
<tr>
<td>“FILLPATTERN=fillpattern-value” (p. 1017)</td>
<td>Specify the fill pattern to be displayed on a chart</td>
<td>Template-based graphics</td>
<td></td>
</tr>
<tr>
<td>“FONT=font-definition” (p. 1018)</td>
<td>Specify a font definition to use in tables, table cells, and graphs</td>
<td>All graphics environments</td>
<td>All text font attributes</td>
</tr>
<tr>
<td>“FONTFAMILY=“string-1&lt;…, string-n&gt;”“ (p. 1019)</td>
<td>Specify the font to use in table cells and graphs</td>
<td>All graphics environments</td>
<td>Font family</td>
</tr>
<tr>
<td>Attribute</td>
<td>Task</td>
<td>Graphics Environment</td>
<td>Affected Items</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>----------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>“FONTSIZE=dimension</td>
<td>size” (p. 1019)</td>
<td>Specify the size of the font for tables, table cells, and graphs</td>
<td>All graphics environments</td>
</tr>
<tr>
<td>“FONTSTYLE=ITALIC</td>
<td>ROMAN</td>
<td>SLANT” (p. 1020)</td>
<td>Specify the style of the font for tables, table cells, and graphs</td>
</tr>
<tr>
<td>“FONTSTYLE=ITALIC</td>
<td>ROMAN</td>
<td>SLANT” (p. 1020)</td>
<td>Specify the font weight of tables, table cells, and graphs</td>
</tr>
<tr>
<td>“FRAMEBORDER=ON</td>
<td>OFF” (p. 1022)</td>
<td>Specify whether there is a graph wall border</td>
<td>All graphics environments</td>
</tr>
<tr>
<td>“GRADIENT_DIRECTION=&quot;YAXIS&quot;</td>
<td>&quot;XAXIS &quot;” (p. 1022)</td>
<td>Specify the direction of the gradient</td>
<td>Device-based graphics</td>
</tr>
<tr>
<td>“IMAGE=&quot;string&quot;” (p. 1024)</td>
<td>Specify the path to an image</td>
<td>Device-based graphics</td>
<td>Image that can be positioned, but not stretched in graph, chart, walls, floor</td>
</tr>
<tr>
<td>“KPISKIN=BASIC</td>
<td>MODERN</td>
<td>NONE</td>
<td>ONYX</td>
</tr>
<tr>
<td>“LINESTYLE=pattern-number” (p. 1024)</td>
<td>Specify the pattern of a line</td>
<td>All graphics environments</td>
<td>Borders, axis lines, grid, reference</td>
</tr>
<tr>
<td>“LINETHICKNESS=dimension” (p. 1025)</td>
<td>Specify the thickness of a line</td>
<td>All graphics environments</td>
<td>Thickness of line</td>
</tr>
<tr>
<td>“MARKERSIZE=dimension” (p. 1027)</td>
<td>Specify a marker size</td>
<td>All graphics environments</td>
<td>Marker size</td>
</tr>
<tr>
<td>“MARKERSYMBOL=marker-symbol” (p. 1027)</td>
<td>Specify a marker symbol</td>
<td>All graphics environments</td>
<td>Marker used</td>
</tr>
<tr>
<td>“NEUTRALCOLOR=color” (p. 1028)</td>
<td>Specify the middle color of a three-color ramp</td>
<td>Template-based graphics</td>
<td>Contours, gradient legends</td>
</tr>
<tr>
<td>“OUTPUTHEIGHT=dimension” (p. 1028)</td>
<td>Specify the height of a graph</td>
<td>All graphics environments</td>
<td>Height of graph</td>
</tr>
<tr>
<td>Attribute</td>
<td>Task</td>
<td>Graphics Environment</td>
<td>Affected Items</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>----------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>OUTPUTWIDTH=dimension (p. 1028)</td>
<td>Specify the width of a graph</td>
<td>All graphics environments</td>
<td>Width of graph</td>
</tr>
<tr>
<td>STARTCOLOR=color (p. 1032)</td>
<td>Specify the start fill color for a graph</td>
<td>All graphics environments</td>
<td>Contours, gradient legends</td>
</tr>
<tr>
<td>TEXTALIGN=alignment (p. 1033)</td>
<td>Specify the alignment of an image</td>
<td>Device-based graphics</td>
<td>Image horizontal positioning</td>
</tr>
<tr>
<td>TICKDISPLAY=&quot;INSIDE&quot;</td>
<td>Specify the placement of all major and minor axis tick marks</td>
<td>Template-based graphics</td>
<td>Placement of all major and minor axis tick marks</td>
</tr>
<tr>
<td>TRANSPARENCY=dimension (p. 1035)</td>
<td>Specify the transparency of backgrounds, fills, lines, and markers</td>
<td>All graphics environments</td>
<td>Backgrounds, fills, lines, markers</td>
</tr>
<tr>
<td>VERTICALALIGN=BOTTOM</td>
<td>Specify vertical justification</td>
<td>Device-based graphics</td>
<td>Image vertical positioning</td>
</tr>
</tbody>
</table>

### Style Attributes Detailed Information

**ABSTRACT=ON | OFF**

specifies whether styles used in an HTML document are used in CSS style files.

**ON**
specifies that styles are used in CSS style files.

**OFF**
specifies that styles are not used in CSS style files.

**Restriction**
The ABSTRACT= attribute is valid only in markup family destinations.

**ACTIVE LINK COLOR=color**

specifies the color that a link in an HTML document changes to after you click it, but before the browser opens that file.

**Restriction**
The ACTIVELINKCOLOR= attribute is valid only in markup family destinations.

**See**
color style attribute value on page 1037

**ASIS=ON | OFF**

specifies how to handle leading spaces and line breaks in an HTML document.

**ON**
prints text with leading spaces and line breaks, in the same manner as the LISTING output.
OFF
trims leading spaces and ignores line breaks.

<table>
<thead>
<tr>
<th>Default</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restriction</td>
<td>The ASIS= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.</td>
</tr>
</tbody>
</table>

**BACKGROUNDCOLOR=**<i>color</i>
specifies the color of the background of the tables, table cells, or graphs.

<table>
<thead>
<tr>
<th>Alias</th>
<th>BACKGROUND=</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restriction</td>
<td>The BACKGROUNDCOLOR= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.</td>
</tr>
<tr>
<td>Interaction</td>
<td>The CBACK= option in the SAS/GRAPH GOPTIONS statement overrides the BACKGROUNDCOLOR= attribute.</td>
</tr>
<tr>
<td>Tip</td>
<td>Generally, the background color of the table cell overrides the background color of the table. You see the background color for the table only as the space between table cells (see “BORDERSPACING=dimension” on page 1010).</td>
</tr>
</tbody>
</table>

| See | color style attribute value on page 1037 |
| Examples | “Creating a Stand-Alone Style” in SAS Output Delivery System: Procedures Guide |
| | “Modifying the Default Style with the CLASS Statement” in SAS Output Delivery System: Procedures Guide |

**BACKGROUNDIMAGE=”string”**
specifies an image in a table, table cell, or graph to use as the background. Viewers can tile or stretch the image as the background for the HTML table or graph that the procedure creates. For graphs, the specified image is stretched.

<i>string</i>
is the name of a GIF, JPEG, or PNG file. Use a simple filename, a complete path, or a URL. However, the most versatile approach is to use a simple filename and to place all image files in the local directory.

| Restriction | The BACKGROUNDIMAGE= attribute is valid in markup family destinations, the PCL destination, and the PS destination. It is also valid in the TAGSETS.RTF destination, but the background image applies to the RTF document and not to a table or a table cell. |
| Interaction | The BACKGROUNDIMAGE= attribute is overridden by the IBACK= and IMAGESTYLE=FIT options in the SAS/GRAPH GOPTIONS statement. |
| See | string attribute value on page 1042 |

**BACKGROUNDPOSITION=**<i>position</i>
specifies the position of the background of the tables, table cells, or graphs.
position can be one of the following:

- BOTTOM
- BOTTOM_CENTER
- BOTTOM_LEFT
- BOTTOM_RIGHT
- CENTER
- CENTER_BOTTOM
- CENTER_CENTER
- CENTER_LEFT
- CENTER_RIGHT
- CENTER_TOP
- LEFT
- LEFT_BOTTOM
- LEFT_CENTER
- LEFT_TOP
- RIGHT
- RIGHT_BOTTOM
- RIGHT_CENTER
- RIGHT_TOP
- TOP
- TOP_CENTER
- TOP_LEFT
- TOP_RIGHT

Default: TOP_LEFT

BACKGROUNDREPEAT=option

specifies whether an image is repeated horizontally, vertically, both, or not repeated. option can be one of the following:

- NO_REPEAT
  specifies that the image is not repeated.
- REPEAT
  specifies that the image is repeated both horizontally and vertically.
- REPEAT_X
  specifies that the image is repeated horizontally.
- REPEAT_Y
  specifies that the image is repeated vertically.

Restriction: The BACKGROUNDREPEAT= attribute is valid in most markup family destinations.

BODYSCROLLBAR=YES | NO | AUTO

specifies whether to put a scroll bar in the frame that references the body file.
YES
places a scroll bar in the frame that references the body file.

NO
specifies not to put a scroll bar in the frame that references the body file.

AUTO
places a scroll bar in the frame that references the body file only if needed.

Restriction The BODYSCROLLBAR= attribute is valid only in markup family destinations.

Tip Typically, BODYSCROLLBAR= is set to AUTO.

\textbf{BODYSIZE=} \textit{dimension} \textbar{} \textit{dimension}\% \textbar{} *
specifies the width of the frame that displays the body file in the HTML frame file.

\textit{dimension}
is a nonnegative number or the width of the frame specified as a percentage of the entire display.

* specifies to use whatever space is left after displaying the content and page files as specified by the CONTENTSIZE= attribute.

Restriction The BODYSIZE= attribute is valid only in markup family destinations.

Tip If \textit{dimension} is a nonnegative number, then the unit of measure is pixels.

See dimension attribute value on page 1039
For information about the HTML files that ODS creates, see “HTML Links and References Produced by the HTML Destination ” on page 1101.

\textbf{BORDERBOTTOMCOLOR=} \textit{color}
specifies the color of the bottom border of a table or table cell.

Restriction The BORDERBOTTOMCOLOR= attribute is valid only in markup family destinations, printer family destinations, RTF destination, and the Measured RTF destination.

Tip You might also need to specify a BORDERBOTTOMWIDTH= attribute to override the style in the ODS destination.

See color style attribute value on page 1037

\textbf{BORDERBOTTOMSTYLE=} \textit{line-style}
specifies the line style of the bottom border of the specified table cell.

\textit{line-style}
can be one of the following:

- DASHED
- DOTTED
- DOUBLE
- GROOVE
Restriction  The BORDERBOTTOMSTYLE= attribute is valid only in markup family destinations, the RTF destination, and the Measured RTF destination.

Tip  You might also need to specify the BORDERBOTTOMWIDTH= attribute to override the style in the ODS destination.

**BORDERBOTTOMWIDTH=** _dimension_

specifies the width of the bottom border of a table or table cell.

Restriction  The BORDERBOTTOMWIDTH= attribute is valid only in markup family destinations, the RTF destination, printer family destinations, and the Measured RTF destination.

See  dimension attribute value on page 1039

**BORDERCOLLAPSE=** _COLLAPSE | SEPARATE_

specifies whether the border is collapsed or separated.

Default  SEPARATE

**BORDERCOLOR=** _color_

specifies the border color of a table or table cell. The color is applied to all four borders.

Restriction  The BORDERCOLOR= attribute is valid only in markup family destinations, the RTF destination, printer family destinations, and the Measured RTF destination.

See  color style attribute value on page 1037

**BORDERCOLORDARK=** _color_

in a table or table cell, specifies the darker color to use in a border that uses two colors to create a three-dimensional effect.

Restriction  The BORDERCOLORDARK= attribute is valid only in markup family destinations and printer family destinations.

Interaction  The BORDERCOLORDARK style attribute is ignored in HTML4 output because it is not part of the HTML4 standard. To create a color border in the HTML4 output, use the BORDERCOLOR= style attribute.

See  color style attribute value on page 1037

Example  “Defining a Table and Graph Style” in SAS Output Delivery System: Procedures Guide
BORDERCOLORLIGHT=\textit{color}  
in a table or table cell, specifies the lighter color to use in a border that uses two  
colors to create a three-dimensional effect.  

Restriction  
The BORDERCOLORLIGHT= attribute is valid only in markup  
family destinations and printer family destinations.  

Interaction  
The BORDERCOLORLIGHT style attribute is ignored in the creation  
of HTML4 output because it is not part of the HTML4 standard. To  
create a color border in HTML4 output, use the BORDERCOLOR=  
style attribute.  

See  
color style attribute value on page 1037  

Example  
“Defining a Table and Graph Style” in \textit{SAS Output Delivery System:  
Procedures Guide}  

BORDERLEFTCOLOR=\textit{color}  
specifies the color of the left border of the table.  

Restriction  
The BORDERLEFTCOLOR= attribute is valid only in markup family  
destinations, the RTF destination, printer family destinations, and the  
Measured RTF destination.  

Tip  
You might also need to specify the BORDERLEFTWIDTH= attribute  
to override the style in the ODS destination.  

See  
color style attribute value on page 1037  

BORDERLEFTSTYLE=\textit{line-style}  
specifies the line style of the left border of the specified table cell.  

\textit{line-style}  
can be one of the following:  

\begin{itemize}  
\item DASHED  
\item DOTTED  
\item DOUBLE  
\item GROOVE  
\item HIDDEN  
\item INSET  
\item OUTSET  
\item RIDGE  
\item SOLID  
\end{itemize}  

Restriction  
The BORDERLEFTSTYLE= attribute is valid only in markup family  
destinations, the RTF destination, and the Measured RTF destination.  

Tip  
You might also need to specify the BORDERLEFTWIDTH= attribute  
to override the style in the ODS destination.  

BORDERLEFTWIDTH=\textit{dimension}  
specifies the width of the left border of a table or table cell.
Restriction The BORDERLEFTWIDTH= attribute is valid only in markup family destinations, the RTF destination, printer family destinations, and the Measured RTF destination.

See dimension attribute value on page 1039

**BORDERRIGHTCOLOR=color**
specifies the color of the right border of a table or table cell.

Restriction The BORDERRIGHTCOLOR= attribute is valid only in markup family destinations, the RTF destination, printer family destinations, and the Measured RTF destination.

Tip You might also need to specify the BORDERRIGHTWIDTH= attribute to override the style in the ODS destination.

See color style attribute value on page 1037

**BORDERRIGHTSTYLE=line-style**
specifies the line style of the right border of the selected cell.

*line-style*
can be one of the following:

- DASHED
- DOTTED
- DOUBLE
- GROOVE
- HIDDEN
- INSET
- OUTSET
- RIDGE
- SOLID

Restriction The BORDERRIGHTSTYLE= attribute is valid only in markup family destinations, the RTF destination, and the Measured RTF destination.

Tip You might also need to specify the BORDERRIGHTWIDTH= attribute to override the style in the ODS destination.

**BORDERRIGHTWIDTH=dimension**
specifies the width of the right border of the table.

Restriction The BORDERRIGHTWIDTH= attribute is valid only in markup family destinations, printer family destinations, RTF destination, and the Measured RTF destination.

See dimension attribute value on page 1039

**BORDERSPACING=dimension**
specifies the vertical and horizontal thickness of the spacing between cells in a table.

Alias CELLSPACING=
Default 0

Restriction The BORDERSPACING= attribute is valid in markup family destinations other than HTML5, printer family destinations, and the RTF destination.

Interaction If BORDERWIDTH= is nonzero, and if the background color of the table cells contrasts with the background color of the table, then the color of the table cell spacing is determined by the table's background.

See dimension attribute value on page 1039

Examples “Creating a Stand-Alone Style” in SAS Output Delivery System: Procedures Guide

“Modifying the Default Style with the CLASS Statement” in SAS Output Delivery System: Procedures Guide

BORDERTOPCOLOR=color specifies the color of the top border of a table or table cell.

Restrictions The BORDERTOPCOLOR= attribute is valid only in markup family destinations, printer family destinations, RTF destination, and the Measured RTF destination.

To ensure that the top border color is created, specify the BORDERTOPWIDTH= attribute and the BORDERTOPCOLOR= attribute for the RTF destination. For the RTF destination, specify the BORDERTOPCOLOR= attribute in conjunction with the BORDERTOPWIDTH= attribute to ensure that the top border color is created.

Tip You might also need to specify the BORDERTOPWIDTH= attribute to override the style in the ODS destination.

See color style attribute value on page 1037

BORDERTOPSTYLE= line-style specifies the line style of the top border of the specified table cell.

*line-style* can be one of the following:

- DASHED
- DOTTED
- DOUBLE
- GROOVE
- HIDDEN
- INSET
- OUTSET
- RIDGE
- SOLID
Restrictions  The BORDERTOPSTYLE= attribute is valid only in markup family destinations, the RTF destination, and the Measured RTF destination.

For the RTF destination, specify the BORDERTOPSTYLE= attribute in conjunction with the BORDERTOPWIDTH= attribute to ensure that the style of the top border is the style that you specified.

Tip  You might also need to specify the BORDERTOPWIDTH= attribute to override the style in the ODS destination.

BORDERTOPWIDTH=dimension
 specifies the width of the top border of the table or table cell.

Restriction  The BORDERTOPWIDTH= attribute is valid only in markup family destinations, printer family destinations, RTF destination, and the Measured RTF destination.

See  dimension attribute value on page 1039

BORDERWIDTH=dimension
 specifies the width of the table borders. The value of BORDERWIDTH= is applied to all four borders.

Restriction  The BORDERWIDTH= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

Tip  Typically, when BORDERWIDTH=0, the ODS destination sets RULES=NONE (see the discussion about “RULES=rule-type” on page 1032) and FRAME=VOID (see the discussion about “FRAME=frame-type” on page 1021).

See  dimension attribute value on page 1039

Examples  “Creating a Stand-Alone Style” in SAS Output Delivery System: Procedures Guide

“Modifying the Default Style with the CLASS Statement” in SAS Output Delivery System: Procedures Guide

CAPSTYLE= line-shape
 specifies the shape of the line at the end of a box whisker. line-shape can be one of the following:

•  "BRACKET"
•  "LINE"
•  "NONE"
•  "SERIF"

Requirement  You must enclose line-shape in quotation marks.

CELLPADDING=dimension | dimension%
 specifies the amount of white space on each of the four sides of the content in a table cell.

dimension
 is a nonnegative number or the amount of white space on each of the four sides of the text in a table cell specified as a percentage of the table.
Restrictions

The CELLPADDING= attribute is valid in markup family destinations other than HTML5, printer family destinations, and the RTF destination.

CELLPADDING= is not valid in the HTML5 destination. All padding is done on the table cells.

See
dimension attribute value on page 1039

Example

“Modifying the Default Style with the CLASS Statement” in SAS Output Delivery System: Procedures Guide

CLASS="string"

specifies the name of the style sheet class to use in an HTML document for the table or table cell.

Alias

HTMLCLASS=

Restriction

The CLASS= attribute is valid only in markup family destinations.

See

string attribute value on page 1042

COLOR=color

specifies the color of the foreground in tables, table cells, or graphs, which is primarily the color of text.

Alias

FOREGROUND=

Restriction

The COLOR= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

Interaction

The COLOR= attribute is overridden by the CBACK= option in the SAS/GRAPH GOPTIONS statement.

Tip

In a table, the COLOR= attribute affects only the text that is specified with the PRETEXT=, POSTTEXT=, PREHTML=, and POSTHTML= attributes. To alter the font for the text that appears in the table, set the attribute for a table cell.

See
color style attribute value on page 1037

Examples

“Modifying the Default Style with the CLASS Statement” in SAS Output Delivery System: Procedures Guide

“Creating a Stand-Alone Style” in SAS Output Delivery System: Procedures Guide

CONNECT=connect-line-type

specifies the characteristics of a box plot connect line. connect-line-type can be one of the following:

- "MAX"
- "MEAN"
- "MEDIAN"
- "MIN"
- "Q1"
"Q3"

Requirement You must enclose connect-line-type in quotation marks.

CONTENTPOSITION=position
specifies the position, within the frame file, of the frames that display the contents and the page files. *position* can be one of the following:

**LEFT**
places the frames on the left.

**RIGHT**
places the frames on the right.

**TOP**
places the frames at the top.

**BOTTOM**
places the frames at the bottom.

Alias L

Alias R

Alias T

Alias B

Restriction The CONTENTPOSITION= attribute is valid only in markup family destinations.

See For information about the HTML files that ODS creates, see “HTML Links and References Produced by the HTML Destination ” on page 1101.

CONTENTSCROLLBAR=YES | NO | AUTO
specifies whether to put a scroll bar in the frames in the frame file that display the contents and the page files. (For information about the HTML files that ODS creates, see “HTML Links and References Produced by the HTML Destination ” on page 1101.)

**YES**
places a scroll bar in the frames in the frame file that display the contents and the page files.

**NO**
specifies not to put a scroll bar in the frames in the frame file that display the contents and the page files.

**AUTO**
specifies that the browser put a scroll bar on the table of contents frame only if the content in that panel is big enough to require scrolling.

Restriction The CONTENTSCROLLBAR= attribute is valid only in markup family destinations.

Tip Typically, CONTENTSCROLLBAR= is set to AUTO.
**CONTENTSIZE=**\( \text{dimension} \mid \text{dimension} \% \mid * \)  

Specifies the width of the frames in the frame file that display the contents and the page files.

- \( \text{dimension} \) is a nonnegative number or the width of the frames specified as a percentage of the entire display.
- \( * \) specifies to use whatever space is left after displaying the body file as specified by the BODYSIZE= attribute.

**Restriction**

The CONTENTSIZE= attribute is valid only in markup family destinations.

**Requirement**

\( \text{dimension} \% \) must be a positive number between 0 and 100.

**Tip**

If \( \text{dimension} \) is a nonnegative number, then the unit of measure is pixels.

**See**

dimension attribute value on page 1039

```
“BODYSIZE=\text{dimension} \mid \text{dimension}\% \mid *” on page 1007
```

For information about the HTML files that ODS creates, see “HTML Links and References Produced by the HTML Destination” on page 1101.

**CONTENTTYPE=**\( \text{string} \)

Specifies the value of the content type for pages in an HTML document that is sent directly to a web server rather than to a file.

- \( \text{string} \) is the content type for the pages.

**Requirement**

\( \text{string} \) must be enclosed in quotation marks.

**Tip**

The value of \( \text{string} \) is usually "text/html".

**See**

string attribute value on page 1042

**Alias**

HTMLCONTENTTYPE=

**Restriction**

The CONTENTTYPE= attribute is valid only in markup family destinations.

**CONTRASTCOLOR=**\( \text{color} \)

Specifies the alternate colors for maps. The alternate colors are applied to the blocks on region areas in block maps.

**Restriction**

The CONTRASTCOLOR= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**See**

color style attribute value on page 1037
DATASKIN=CRISP | GLOSS | MATTE | NONE | PRESSED | SHEEN

specifies the type of skin to apply to plots and charts (other than KPIs) to give them a raised, 3-D appearance.

The DATASKIN= style attribute is valid for the following plots and charts in the Graph Template Language:

- bar charts
- pie charts
- scatter plots
- waterfall charts

The DATASKIN= style attribute is valid for the following plots and charts in the SG procedures:

- bar charts
- scatter plots
- waterfall charts

<table>
<thead>
<tr>
<th>Table 12.3 DATASKIN Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
</tr>
<tr>
<td><img src="image" alt="NONE" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MATTE</th>
<th>PRESSSED</th>
<th>SHEEN</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="MATTE" /></td>
<td><img src="image" alt="PRESSSED" /></td>
<td><img src="image" alt="SHEEN" /></td>
</tr>
</tbody>
</table>

Restriction

In the first maintenance release of SAS 9.4 and later releases, the maximum number of skinned graphical elements is limited to 200 per plot in an overlay or prototype layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

DISPLAYOPTS="display-feature"

specifies one or more display features for ODS graphs. To specify multiple features, enclose the list of features in quotation marks (for example: displayopts="fill caps mean"). "display-feature" can be one of the following:

CAPS
displays caps at the ends of the whiskers.

Restriction  
CAPS can be used only for box plots.

CONNECT
displays the line connecting multiple boxes.

Restriction  
CONNECT can be used only for box plots.
FILL
displays filled boxes, bars, ellipses, and bands.

Restriction FILL can be used only for box plots, histograms, ellipses, and confidence bands.

MEAN
displays the mean symbol within a box.

Restriction MEAN can be used only for box plots.

MEDIAN
displays the median line within the box.

NOTCHES
displays notched boxes.

Restriction NOTCHES can be used only for box plots.

OUTLIERS
displays markers for the outliers.

Restriction OUTLIERS can be used only for box plots.

OUTLINE
displays outlined ellipses and bars.

Restriction OUTLINE can be used only for ellipses, bands, and histograms.

Requirement You must enclose "display-feature" in quotation marks.

DOCTYPE="string"
specifies the entire doctype declaration for the HTML document, including the opening "<!DOCTYPE" and the closing ">".

string is the doctype declaration.

Requirement string must be enclosed in quotation marks.

See string attribute value on page 1042

Alias HTMLDOCTYPE=

Restriction The DOCTYPE= attribute is valid only in markup family destinations.

DROPSHADOW=ON | OFF
specifies whether the drop shadow color for text is displayed.

ENDCOLOR="color"
specifies the final color used with a two- or three-color ramp.

See color style attribute value on page 1037

FILLPATTERN="fillpattern-value"
specifies the fill pattern to be displayed on the chart. The valid values are: S, E, L1, L2, L3, L4, L5, R1, R2, R3, R4, R5, X1, X2, X3, X4, and X5.
Restriction  The FILLPATTERN= attribute is valid for bar charts only.

Tip  To display these fill patterns on the bar chart through the style, you must also specify "fillpattern" as one of the DISPLAYOPTS in the GRAPHBAR style element.

See  For a table of style elements and the style attributes that are valid in each one, see “Style Elements Affecting Template-Based Graphics” on page 975 and “Style Elements Affecting Device-Based Graphics” on page 983.

FILLRULEWIDTH=dimension
places a rule of the specified width into the space around the text (or entire cell if there is no text) in a table where white space would otherwise appear.

Restriction  The FILLRULEWIDTH= attribute is valid only in printer family destinations.

Tip  If no text is specified, then FILLRULEWIDTH= fills the space around the text with hyphen marks. For example: --this-- or this ------.

See  dimension attribute value on page 1039

FLYOVER="string"
specifies the text to show in a data tip for the table cell.

string  is the text of the data tip.

Requirement  string must be enclosed in quotation marks.

See  string attribute value on page 1042

Restriction  The FLYOVER= attribute is valid only in markup family destinations and the PDF destination.

FONT=font-definition
specifies a font definition to use in tables, table cells, and graphs.

Restriction  The FONT= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

Tips  For a table, the FONT= attribute affects only the text that is specified with the PRETEXT=, POSTTEXT=, PREHTML=, and POSTHTML= attributes. To alter the font for the text that appears in the table, set the attribute for a table cell.
If the system does not recognize the font specified, then it refers to the system's default font. This attribute does not accept concatenated fonts. SAS Graph Styles can only specify one font.

**See**  
font-definition attribute value on page 1040

**Example**  
“Modifying the Default Style with the CLASS Statement” in SAS Output Delivery System: Procedures Guide

**FONTFAMILY=**"string-1<…, string-n>"

specifies the font to use in table cells and graphs. If you supply multiple fonts, then the destination device uses the first one that is installed on the system.

*string*

is the name of the font.

**Requirement**  
*string* must be enclosed in quotation marks.

**See**  
string attribute value on page 1042

**Alias**  
FONT_FACE=

**Restriction**  
The FONTFAMILY= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**Tips**  
For a table, the FONTFAMILY= attribute affects only the text that is specified with the PRETEXT=, POSTTEXT=, PREHTML=, and POSTHTML= attributes. To alter the font for the text that appears in the table, set the attribute for a table cell.

You cannot be sure what fonts are available to someone who is viewing the output in a browser or printing it on a high-resolution printer. Most devices support the following fonts: Times, Courier, Arial, Helvetica.

**Example**  
“Creating a Stand-Alone Style” in SAS Output Delivery System: Procedures Guide

**FONTSIZE=**\texttt{dimension} | \texttt{size}

specifies the size of the font for tables, table cells, and graphs.

*dimension*

is a nonnegative number.

**Alias**  
FONT_SIZE=

**Restriction**  
If you specify a dimension, then specify a unit of measure. Without a unit of measure, the number becomes a relative size.

**See**  
dimension attribute value on page 1039

*size*

The value of *size* is relative to all other font sizes in the HTML document.

**Range**  
1 to 7

**Restriction**  
The FONTSIZE= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.
Tip For a table, the FONTSIZE= attribute affects only the text that is specified with the PRETEXT=, POSTTEXT=, PREHTML=, and POSTHTML= attributes. To alter the font for the text that appears in the table, set the attribute for a table cell.

Example “Creating a Stand-Alone Style” in SAS Output Delivery System: Procedures Guide

FONTSTYLE=ITALIC | ROMAN | SLANT
specifies the style of the font for tables, table cells, and graphs. In many cases, italic and slant map to the same font.

Alias FONT_STYLE=

Restriction The FONTSTYLE= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

Tip For a table, the FONTSTYLE= attribute affects only the text that is specified with the PRETEXT=, POSTTEXT=, PREHTML=, and POSTHTML= attributes. To alter the font for the text that appears in the table, set the attribute for a table cell.

Examples “Creating a Stand-Alone Style” in SAS Output Delivery System: Procedures Guide

“Modifying the Default Style with the CLASS Statement” in SAS Output Delivery System: Procedures Guide

FONTWEIGHT=weight
specifies the font weight of tables, table cells, and graphs. weight is any of the following:

• MEDIUM
• BOLD
• DEMI_BOLD
• EXTRA_BOLD
• LIGHT
• DEMI_LIGHT
• EXTRA_LIGHT.

Alias FONT_WEIGHT=

Restrictions You cannot be sure what font weights are available to someone who is viewing the output in a browser or printing it on a high-resolution printer. Most devices support only MEDIUM and BOLD, and possibly LIGHT.

The FONTWEIGHT= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

Tip For a table, the FONTWEIGHT= attribute affects only the text that is specified with the PRETEXT=, POSTTEXT=, PREHTML=, and POSTHTML= attributes. To alter the font for the text that appears in the table, set the attribute for a table cell.
Examples

“Creating a Stand-Alone Style” in *SAS Output Delivery System: Procedures Guide*

**FONTWIDTH=**<br/>

Specifies the font width of tables, table cells, and graphs compared to the width of the usual design of the table, table cell, or graph. **relative-width** is any of the following:

- NORMAL
- COMPRESSED
- EXTRA_COMPRESSED
- NARROW
- WIDE
- EXPANDED

**Alias**

FONT_WIDTH=

**Restrictions**

Few fonts honor these values.

The FONTWIDTH= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**Tip**

For a table, the FONTWIDTH= attribute affects only the text that is specified with the PRETEXT=, POSTTEXT=, PREHTML=, and POSTHTML= attributes. To alter the font for the text that appears in the table, set the attribute for a table cell.

**Example**

“Creating a Stand-Alone Style” in *SAS Output Delivery System: Procedures Guide*

**FRAME=**<br/>

Specifies the type of frame to use on a table. This table shows the possible values for **frame-type** and their meanings:

<table>
<thead>
<tr>
<th>Value for frame-type</th>
<th>Frame Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABOVE</td>
<td>A border at the top</td>
</tr>
<tr>
<td>BELOW</td>
<td>A border at the bottom</td>
</tr>
<tr>
<td>BOX</td>
<td>Borders at the top, bottom, and both sides</td>
</tr>
<tr>
<td>HSIDES</td>
<td>Borders at the top and bottom</td>
</tr>
<tr>
<td>LHS</td>
<td>A border at the left side</td>
</tr>
<tr>
<td>RHS</td>
<td>A border at the right side</td>
</tr>
<tr>
<td>VOID</td>
<td>No borders</td>
</tr>
</tbody>
</table>
### Value for frame-type

<table>
<thead>
<tr>
<th>Value for frame-type</th>
<th>Frame Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSIDES</td>
<td>Borders at the left and right sides</td>
</tr>
</tbody>
</table>

**Restriction**
The FRAME= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**Example**
“Modifying the Default Style with the CLASS Statement” in *SAS Output Delivery System: Procedures Guide*

### FRAMEBORDER=ON | OFF
specifies whether to put a border around the frame for an HTML file that uses frames.

- **ON** places a border around the frame for an HTML file that uses frames.
- **OFF** specifies not to put a border around the frame for an HTML file that uses frames.

**Restriction** The FRAMEBORDER= attribute is valid only in markup family destinations.

### FRAMEBORDERWIDTH=dimension
specifies the width of the border around the frames for an HTML file that uses frames.

**Restriction** The FRAMEBORDERWIDTH= attribute is valid only in markup family destinations.

**See** dimension attribute value on page 1039

### FRAMESPACING=dimension
specifies the width of the space between frames for HTML that uses frames.

**Restriction** The FRAMESPACING= attribute is valid only in markup family destinations.

**See** dimension attribute value on page 1039

### GRADIENT_DIRECTION="YAXIS" | "XAXIS"
specifies the direction of the gradient.

- **"YAXIS"** specifies a vertical gradient.
- **"XAXIS"** specifies a horizontal gradient.

### HEIGHT=dimension
specifies the height of a table cell, graph, or graphics in an HTML document.

- **dimension** is a nonnegative number.

**See** dimension attribute value on page 1039
Aliases

<table>
<thead>
<tr>
<th>Alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CELLHEIGHT=</td>
<td></td>
</tr>
<tr>
<td>OUTPUTHEIGHT=</td>
<td></td>
</tr>
</tbody>
</table>

Restrictions

- The HEIGHT= option does not apply to output generated as a result of GRSEG (graph segment) output.
- The HEIGHT= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

Interaction

- The YPIXELS= option in the SAS/GRAPH GOPTIONS statement overrides the HEIGHT= attribute.

Tip

HTML automatically sets cell height appropriately. You will seldom need to specify this attribute in the HTML destination.

**HREFTARGET=**"*target*

specifies the window or frame in which to open the target of the link. *target* is one of these values:

- **_blank**
  - opens the target in a new, blank window. The window has no name.
  - **Restriction** Use lowercase letters to specify values for HREFTARGET.

- **_parent**
  - opens the target in the window from which the current window was opened.
  - **Restriction** Use lowercase letters to specify values for HREFTARGET.

- **_search**
  - opens the target in the browser's search pane.
  - **Restrictions** Only available in Internet Explorer 5.0 or later.
  - Use lowercase letters to specify values for HREFTARGET.

- **_self**
  - opens the target in the current window.
  - **Restriction** Use lowercase letters to specify values for HREFTARGET.

- **_top**
  - opens the target in the topmost window.
  - **Restriction** Use lowercase letters to specify values for HREFTARGET.

- **"name"**
  - opens the target in the specified window or the frame.

Default: **_self**

**Restrictions**

- The HREFTARGET= attribute is valid only in markup family destinations.
  - Use lowercase letters to specify values for HREFTARGET.

**Requirement**

*target* must be enclosed in quotation marks.
**HTMLID=**"string"

Specifies an ID for the table or table cell. The ID is for use by a Java Script.

**string**

Is the ID text.

**Requirement**  
String must be enclosed in quotation marks.

**See**  
String attribute value on page 1042

**Restriction**  
The HTMLID= attribute is valid only in markup family destinations.

**HTMLSTYLE=**"string"

Specifies individual attributes and values for a table or table cell in an HTML document.

**string**

Is the name of an attribute or value.

**Requirement**  
String must be enclosed in quotation marks.

**See**  
String attribute value on page 1042

**Restriction**  
The HTMLSTYLE= attribute is valid only in markup family destinations.

**IMAGE=**"string"

Specifies the image to appear in a graph. This image is positioned or tiled.

**string**

Is the name of the image.

**Requirement**  
String must be enclosed in quotation marks.

**See**  
String attribute value on page 1042

**Restriction**  
The IMAGE= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**Interaction**  
The BACK= and IMAGESTYLE=TILE options in the SAS/GRAPH GOPTIONS statement override the IMAGE= attribute.

**KPISKIN=BASIC | MODERN | NONE | ONYX | SATIN**

Specifies the type of skin to apply to KPI charts to give them a raised, 3-D appearance.

**LINESTYLE=**pattern-number

Specifies the pattern of a line. Valid pattern numbers range from 1 to 46. Not all pattern numbers have names. You must specify the line pattern by its number.

**pattern-number** can be one of the following:
LINETHICKNESS=\textit{dimension}\newline specifies the thickness of a line.

\textbf{See}  \textit{dimension} attribute value on page \textbf{1039}

\textbf{LINKCOLOR=\textit{color}}\newline specifies the color for the links in an HTML document that have not yet been visited.

\textbf{Restriction}  The \textbf{LINKCOLOR=} attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

\textbf{See}  \textit{color style} attribute value on page \textbf{1037}

\textbf{LISTENTRYANCHOR=ON | OFF}\newline in an HTML document, the \textbf{LISTENTRYANCHOR=} attribute specifies whether to make the entry in the table of contents a link to the body file.

\textbf{ON}\newline specifies to make this entry in the table of contents a link to the body file.

\textbf{OFF}\newline specifies not to make this entry in the table of contents a link to the body file.

\textbf{Restriction}  The \textbf{LISTENTRYANCHOR=} attribute is valid only in markup family destinations.

\textbf{LISTSTYLETYPE=\textit{bullet-type}}\newline specifies the type of bullet to use for lists and the contents file. ODS uses bullets in the contents file.

\textbf{bullet-type}\newline is one of the following:

\begin{verbatim}
<table>
<thead>
<tr>
<th>Pattern</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid</td>
<td>1</td>
</tr>
<tr>
<td>ShortDash</td>
<td>2</td>
</tr>
<tr>
<td>MediumDash</td>
<td>4</td>
</tr>
<tr>
<td>LongDash</td>
<td>5</td>
</tr>
<tr>
<td>MediumDashShortDash</td>
<td>8</td>
</tr>
<tr>
<td>DashDashDot</td>
<td>14</td>
</tr>
<tr>
<td>DashDotDot</td>
<td>15</td>
</tr>
<tr>
<td>Dash</td>
<td>20</td>
</tr>
<tr>
<td>LongDashShortDash</td>
<td>26</td>
</tr>
<tr>
<td>Dot</td>
<td>34</td>
</tr>
<tr>
<td>ThinDot</td>
<td>35</td>
</tr>
<tr>
<td>ShortDashDotDot</td>
<td>41</td>
</tr>
<tr>
<td>MediumDashDotDot</td>
<td>42</td>
</tr>
</tbody>
</table>
\end{verbatim}
Alias **BULLET**

See string attribute value on page 1042

**Restriction**
The LISTSTYLETYPE= attribute is valid only in markup family destinations.

**MARGINBOTTOM=dimension**
specifies the bottom margin for the HTML document.

Alias **BOTTOMMARGIN=**

**Restriction**
The MARGINBOTTOM= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**Tip**
If the orientation of a PDF document is changed after the PDF destination is opened and before the PDF destination is closed, any setting for margins is taken from the OPTIONS statement in place before the ODS PDF FILE= statement. If no OPTIONS statement is used to explicitly set the margins, the margin settings are retrieved from the SAS registry.

See dimension attribute value on page 1039

**MARGINLEFT=dimension**
specifies the left margin for the HTML document.
Alias LEFTMARGIN=

Restriction The MARGINLEFT= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

Tip If the orientation of a PDF document is changed after the PDF destination is opened and before the PDF destination is closed, any setting for margins is taken from the OPTIONS statement in place before the ODS PDF FILE= statement. If no OPTIONS statement is used to explicitly set the margins, the margin settings are retrieved from the SAS registry.

See dimension attribute value on page 1039

MARGINRIGHT=dimension
specifies the right margin for the HTML document.

Alias RIGHTMARGIN=

Restriction The MARGINRIGHT= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

Tip If the orientation of a PDF document is changed after the PDF destination is opened and before the PDF destination is closed, any setting for margins is taken from the OPTIONS statement in place before the ODS PDF FILE= statement. If no OPTIONS statement is used to explicitly set the margins, the margin settings are retrieved from the SAS registry.

See dimension attribute value on page 1039

MARGINTOP=dimension
specifies the top margin for the HTML document.

Alias TOPMARGIN=

Restriction The MARGINTOP= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

Tip If the orientation of a PDF document is changed after the PDF destination is opened and before the PDF destination is closed, any setting for margins is taken from the OPTIONS statement in place before the ODS PDF FILE= statement. If no OPTIONS statement is used to explicitly set the margins, the margin settings are retrieved from the SAS registry.

See dimension attribute value on page 1039

MARKERSIZE=dimension
specifies the marker size (both width and height).

See dimension attribute value on page 1039

MARKERSYMBOL=marker-symbol
specifies a marker symbol. marker-symbol can be one of the following:
**NEUTRALCOLOR=**<br>specifies the middle color in a three-color ramp.<br><br>See color style attribute value on page 1037

**NOBREAKSPACE=**<br>specifies how to handle space characters in table cells.<br><br>**ON**<br>does not let SAS break a line at a space character.<br><br>**OFF**<br>lets SAS break a line at a space character if appropriate.<br><br>*Restriction*<br>The NOBREAKSPACE= attribute is valid in markup family destinations, printer family destinations, and the RTF destination.

**OUTPUTHEIGHT=**<br>specifies the height of a graph.<br><br>See dimension attribute value on page 1039

**OUTPUTWIDTH=**<br>specifies the width of a graph.<br><br>See dimension attribute value on page 1039

**OVERHANGFACTOR=**<br>specifies an upper limit for extending the width of the column in an HTML document.<br><br>*Restriction*<br>The OVERHANGFACTOR= attribute is valid only in markup family and printer family destinations.<br><br>*Tips*<br>Typically, an overhang factor between 1 and 2 works well.<br><br>Usually, the HTML that is generated by ODS tries to ensure that the text in a column wraps when it reaches the requested column width. When the overhang factor is greater than 1, the text can extend beyond the specified width.
**PADDING=**`dimension | dimension%`

specifies the amount of white space between the content of the table cell and the border. The value of PADDING= applies to all four sides.

To change the padding of each side, use one or more of the following attributes:

- **PADDINGBOTTOM=** on page 1029
- **PADDINGLEFT=** on page 1029
- **PADDINGRIGHT=** on page 1029
- **PADDINGTOP=** on page 1029

**Restriction**
The PADDING= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**See**
dimension attribute value on page 1039

**PADDINGBOTTOM=**`dimension | dimension%`

specifies the amount of white space on the bottom of the content of the table cell.

**Default**
0

**Restriction**
The PADDINGBOTTOM= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**See**
dimension attribute value on page 1039

**PADDINGLEFT=**`dimension | dimension%`

specifies the amount of white space on the left side of the content of the table cell.

**Default**
0

**Restriction**
The PADDINGLEFT= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**See**
dimension attribute value on page 1039

**PADDINGRIGHT=**`dimension | dimension%`

specifies the amount of white space on the right side of the content of the table cell.

**Default**
0

**Restriction**
The PADDINGRIGHT= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**See**
dimension attribute value on page 1039

**PADDINGTOP=**`dimension | dimension%`

specifies the amount of white space on the top of the content of the table cell.

**Default**
0

**Restriction**
The PADDINGTOP= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**See**
dimension attribute value on page 1039

**PAGEBREAKHTML=**"string"

specifies HTML to place at page breaks in an HTML document.
**string**
is the HTML code used to place at page breaks.

**Requirement** *string* must be enclosed in quotation marks.

**See** *string attribute value on page 1042*

**Restriction** The PAGEBREAKHTML= attribute is valid only in markup family destinations.

**POSTHTML=/String**
specifies the HTML code to place after the table or table cell.

**string**
is the HTML code to place after a table or table cell.

**Requirement** *string* must be enclosed in quotation marks.

**See** *string attribute value on page 1042*

**Restriction** The POSTHTML= attribute is valid only in markup family destinations.

**Example**
“Modifying the Default Style with the CLASS Statement” in *SAS Output Delivery System: Procedures Guide*

**POSTIMAGE=/String | fileref**
specifies an image to place after the table or table cell.

**string**
names a GIF or JPEG file. Use a simple filename, a complete path, or a URL.

**Requirement** *string* must be enclosed in quotation marks.

**See** *string attribute value on page 1042*

**fileref**
is a reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

**See** "Statements" in *SAS Statements: Reference* for information about the FILENAME statement.

**Restriction** The POSTIMAGE= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**POSTTEXT=/String**
specifies text to place after the table cell or table.

**Restriction** The POSTTEXT= attribute is valid only for markup family destinations, printer family destinations, and the RTF destination.

**Requirement** *string* must be enclosed in quotation marks.

**See** *string attribute value on page 1042*

**PREHTML=/String**
specifies the HTML code to place before the table or table cell.
Restriction  The PREHTML= attribute is valid only for markup family destinations.

See  string attribute value on page 1042

**PREIMAGE=**"*string*" | *fileref*

specifies an image to place before the table or table cell.

*string*

names a GIF or JPEG file. Use a simple filename, a complete path, or a URL.

Restriction  When using the PREIMAGE= style attribute with the PRINTER destination, you must specify STARTPAGE=NO on the PRINTER family statement to display page numbers, times, dates, and titles. Without the STARTPAGE=NO option, preimages are treated like graphs and have no page numbers, times, dates, or titles displayed.

Requirement  Enclose *string* in quotation marks.

See  string attribute value on page 1042

*fileref*

is a reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref. (For information about the FILENAME statement, see "Statements" in *SAS Statements: Reference.*

Restriction  The PREIMAGE= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**PRETEXT=**"*string*"

specifies text to place before the table cell or table.

*string*

text that is placed before the table cell or table.

Requirement  Enclose *string* in quotation marks.

See  string attribute value on page 1042

Example  “Example 3: Customizing the Table of Contents” on page 600

Restriction  The PRETEXT= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**PROTECTSPECIALCHARS=**ON | OFF | AUTO

specifies how less-than signs (<), greater-than signs (>), and ampersands (&) are interpreted in table cells. In HTML and other markup languages, these characters indicate the beginning of a markup tag, the end of a markup tag, and the beginning of the name of a file or character entity.

ON  interprets special characters as the characters themselves. That is, when ON is in effect the characters are protected before they are passed to the HTML or other markup language destination so that the characters are not interpreted as part of the markup language. Using ON enables you to show markup language tags in the HTML document.
interprets special characters as markup language tags. That is, when OFF is in effect, the characters are passed to the HTML or other markup language destination without any protection so that the special characters are interpreted as part of the markup language.

**AUTO**
interprets any string that starts with a < and ends with a > as a markup language tag (ignoring spaces that immediately precede the <, spaces that immediately follow the >, and spaces at the beginning and end of the string). In any other string, AUTO protects the special characters from their markup language meaning.

**Restriction**
The PROTECTSPECIALCHARS= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**RULES=** *rule-type*
specifies the types of rules to use in tables. This table shows the possible values for the RULES= attribute and their meanings:

<table>
<thead>
<tr>
<th>Value of RULES= Attribute</th>
<th>Locations of Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Between all rows and columns</td>
</tr>
<tr>
<td>COLS</td>
<td>Between all columns</td>
</tr>
<tr>
<td>GROUPS</td>
<td>Between the table header and the table and between the table and the table footer, if there is one</td>
</tr>
<tr>
<td>NONE</td>
<td>No rules anywhere</td>
</tr>
<tr>
<td>ROWS</td>
<td>Between all rows</td>
</tr>
</tbody>
</table>

**Restriction**
The RULES= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**Example**
“Defining a Table and Graph Style” in *SAS Output Delivery System: Procedures Guide*

**STARTCOLOR=** *color*
specifies the start fill color for a graph. It is used to create a gradient effect.

**Note:** You can have either a start and end gradient effect or no gradient effect. If you specify a TRANSPARENCY level and you only specify the STARTCOLOR, then the end color is completely transparent gradationally to the specified start color.

**Restriction**
The STARTCOLOR= attribute is valid only for the HTML destination.

**See**
color style attribute value on page 1037
TAGATTR="string"
specifies text to insert into HTML.

string
is the text that is inserted into HTML tags.

Requirements
- string must be enclosed in quotation marks.
- string must be valid HTML for the context in which the style element is created.

Tip
Many style elements are created between <TD> and </TD> tags. To determine how a style element is created, look at the source for the output.

See
string attribute value on page 1042

Restriction
The TAGATTR= attribute is valid only in markup family destinations.

TEXTALIGN=alignment
specifies justification in tables, table cells, and graphs. In graphs, this option specifies the justification of the image specified with the IMAGE= statement. For example, this statement would produce a page number that is centered at the bottom of the page: style PageNo from TitleAndFooters / textalign=c verticalalign=b; This statement would produce a date in the body file that is left-justified at the top of the page: style BodyDate from Date / textalign=l; alignment can be one of the following:

CENTER
specifies center justification.
Alias C

DEC
specifies aligning the values by the decimal point.
Alias D
Restriction Decimal alignment is supported for the printer family and RTF destinations only.

LEFT
specifies left justification.
Alias L

RIGHT
specifies right justification.
Alias R
Restriction Not all contexts support RIGHT. If RIGHT is not supported, it is interpreted as CENTER.

Alias JUST=
Restriction The TEXTALIGN= attribute is valid in markup family destinations other than HTML5, printer family destinations, and the RTF
destination. For the HTML5 destination, you might be able to use MARGINRIGHT=0 instead.

**Tips**

For the printer family destinations and the MARKUP destination, use the style attribute TEXTALIGN= with the style attribute VERTICALALIGN= in the style element PAGENO to control the placement of page numbers.

For printer family destinations and the MARKUP destination, control the placement of dates by using the style attribute TEXTALIGN= with the style attribute VERTICALALIGN= in the BODYDATE or DATE style element.

**TEXTDECORATION=presentation–options**

changes the visual presentation of the text. *presentation–options* can be one of the following:

**BLINK**

specifies that the text's visual presentation alternates rapidly between visible and invisible.

**Restriction**

TEXTDECORATION=BLINK is valid only in the HTML and RTF destinations.

**LINE_THROUGH**

specifies that a line is drawn through the text.

**Restriction**

TEXTDECORATION=LINE_THROUGH is valid only in the HTML destination, the printer family, the measured RTF destination, and the RTF destination.

**OVERLINE**

specifies that a line is drawn above the text.

**Restriction**

TEXTDECORATION=OVERLINE is valid only in the HTML destination and the printer family destinations.

**UNDERLINE**

specifies that a line is drawn below the text.

**Restriction**

TEXTDECORATION=UNDERLINE is valid only in the HTML destination, the printer family destinations, the measured RTF destination, and the RTF destination.

**Tip**

TEXTDECORATION= can be used with inline formatting and the ODS PDF statement to enhance PDF files.

**Example**

“Formatting Cells Using the Textdecoration Style Attribute” in SAS Output Delivery System: Advanced Topics

**TEXTINDENT=n**

specifies the number of spaces that the first line of output is indented.

n specifies the number of spaces to indent the output.

**Alias**

INDENT=
**Default**
The default value for XML is 2. For all other ODS destinations, the default value is 0.

**Restriction**
The TEXTINDENT= attribute is valid only in the markup family destinations, the printer family destinations, and the RTF destination.

**TICKDISPLAY=**
"INSIDE" | "OUTSIDE" | "ACROSS"
specifies the placement of all major and minor axis tick marks.

**TEXTJUSTIFY=**
**INTER_WORD** | **INTER_CHARACTER**
specifies how to evenly distribute text.

**INTER WORD**
specifies that the words are evenly distributed across the page.

**INTER CHARACTER**
specifies that all characters are evenly distributed across a page.

**Tip**
Use the TEXTJUSTIFY= style attribute with the TEXTALIGN=J (alias JUST=) style attribute.

**TRANSPARENCY=**
*dimension*
specifies a transparency level for graphs. The values are 0.0 (opaque) to 1.0 (transparent).

**Restriction**
The TRANSPARENCY= attribute is valid only in the HTML destination.

**See**
dimension attribute value on page 1039

**URL=**
*uniform-resource-locator*
specifies a URL to link to from the current cell.

**Restriction**
The URL= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**Requirement**
*uniform-resource-locator* must be enclosed in quotation marks.

**VERTICALALIGN=**
**BOTTOM** | **MIDDLE** | **TOP**
specifies vertical justification for graphs and cells. In graphs, this option specifies the vertical justification of the image specified with IMAGE=. For example, this statement produces a page number that is centered at the bottom of the page: *style PageNo from TitleAndFooters / textalign=c verticalalign=b;*
This statement produces a date in the body file that is left-justified at the top of the page: *style BodyDate from Date / textalign=l verticalalign=t;*

**BOTTOM**
specifies bottom justification.

**Alias**
B

**MIDDLE**
specifies center justification.

**Alias**
M

**TOP**
specifies top justification.
**Alias** T

**Alias** VJUST=

**Restriction** The VERTICALALIGN= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

**Tips** For printer and markup family destinations, use the style attribute VERTICALALIGN= with the style attribute TEXTALIGN= in the style element PAGENO to control the placement of page numbers.

For printer and markup family destinations, control the placement of dates by using the style attribute VERTICALALIGN= with the style attribute TEXTALIGN= in the BODYDATE or DATE style element.

**VISITEDLINKCOLOR=color**

specifies the color for links that have been visited in an HTML document.

**Restriction** The VISITEDLINKCOLOR= attribute is valid only in markup family destinations.

**See** color style attribute value on page 1037

**WATERMARK=ON | OFF**

specifies whether to make the image that is specified by BACKGROUNDIMAGE= into a watermark. A watermark appears in a fixed position as the window is scrolled.

**ON** specifies to make the image that is specified by BACKGROUNDIMAGE= into a watermark.

**OFF** specifies not to make the image that is specified by BACKGROUNDIMAGE= into a watermark.

**Restriction** The WATERMARK= attribute is valid only in markup family destinations.

**Tip** You can apply a watermark to output generated using the ODS TAGSETS.RTF destination by specifying a background image file. Note that the image is applied to the RTF document and not to a table or a table cell.

**See** “BACKGROUNDIMAGE="string"" on page 1005

**WHITESPACE=options**

specifies how the browser handles multiple whitespace characters and line breaks. *options* can be one of the following:

**NORMAL** specifies that white spaces are compressed and text wraps normally.

**NOWRAP** specifies that white spaces are compressed and that text does not have line breaks.

**PRE** specifies that white spaces are left intact and that text does not have line breaks.
PRE_LINE specifies that white spaces are compressed, keeps line breaks that are in the text, and adds line breaks as needed.

PRE_WRAP specifies that white spaces are left intact and allows line breaking.

Default NORMAL

WIDTH=dimension specifies the width of a table cell, table, line, or a graph.

When used with graphs, the WIDTH= option must be specified as a pixel or percentage value. If a unit of measure is not specified with the dimension, then the value will be in pixels. If a unit of measure other than pixels or percentage is specified with the dimension, then the HEIGHT=dimension is not applied to the graph.

dimension is a nonnegative number.

See dimension attribute value on page 1039

Aliases CELLWIDTH=

OUTPUTWIDTH=

Restrictions The WIDTH= option does not apply to output generated as a result of GRSEG (graph segment) output.

The WIDTH= attribute is valid only in markup family destinations, printer family destinations, and the RTF destination.

Interaction The XPIXELS= option in the SAS/GRAPH GOPTIONS statement overrides the WIDTH= attribute.

Tips A column of cells has the width of the widest cell in the column.

Use WIDTH=100% to make the table or graph as wide as the window that it is open in.

Style Attribute Values

color is a string that identifies a color. A color is defined in the following ways:

• most of the color names that are supported by SAS/GRAPH. These names include the following:
  • a predefined SAS color (for example, blue or VIYG)
  • a red/green/blue (RGB) value (for example, CX0023FF)
  • a hue/light/saturation (HLS) value (for example, H14E162D)
  • a gray-scale value (for example, GRAYBB).
  • a red/green/blue transparency (RGBA) value (for example, a98FB9880)
• a cyan/magenta/yellow/black (CMYK) value (for example, FFFFFF00)

Note: RGBA color mode is not supported by Java devices. RGBA color mode is supported by ActiveX devices when the output is used in Microsoft applications.

• an RGB value with a leading number sign (#) rather than CX (for example, #0023FF).

• one of the colors that exist in the SAS session when the style is used:
  • DMSBLUE
  • DMSRED
  • DMSPINK
  • DMSGREEN
  • DMSCYAN
  • DMSYELLOW
  • DMSWHITE
  • DMSORANGE
  • DMSBLACK
  • DMSMAGENTA
  • DMSGRAY
  • DMSBROWN
  • SYSBACK
  • SYSSECB
  • SYSFORE

Note: Use these colors only when running SAS in the windowing environment.

• an English description of an HLS. Such descriptions use a combination of words to describe the lightness, the saturation, and the hue (in that order). Use the Color Naming System to form a color in the following ways:
  • combining a chromatic hue with a lightness, a saturation, or both
  • combining the achromatic hue gray with a lightness
  • combining the achromatic hue black or white without qualifiers

Use the words in the following table:

<table>
<thead>
<tr>
<th>Lightness</th>
<th>Saturation</th>
<th>Chromatic Hue</th>
<th>Achromatic Hue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very dark</td>
<td>Grayish</td>
<td>Purple</td>
<td></td>
</tr>
<tr>
<td>Dark</td>
<td>Moderate</td>
<td>Red</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>Strong</td>
<td>Orange</td>
<td>brown</td>
</tr>
</tbody>
</table>

Table 12.6  Hue/Light/Saturation (HLS) Values
<table>
<thead>
<tr>
<th>Lightness</th>
<th>Saturation</th>
<th>Chromatic Hue</th>
<th>Achromatic Hue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light</td>
<td>Vivid</td>
<td>Yellow</td>
<td>White</td>
</tr>
<tr>
<td>Very light</td>
<td>Green</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Black and white cannot be combined with a lightness value or a saturation value.
- Gray cannot be combined with a saturation value.

Combine these words to form a wide variety of colors. Here are examples:

- light vivid green
- dark vivid orange
- light yellow

*Note:* The Output Delivery System first tries to match a color with a SAS/GRAPH color. Thus, although brown and orange are interchangeable in the table, if you use them as unmodified hues, then they are different. The reason for this is that ODS interprets them as SAS colors, which are mapped to different colors.

You can also specify hues that are intermediate between two neighboring colors. To do so, combine one of these adjectives with one of its neighboring colors:

- reddish
- orangish
- brownish
- yellowish
- greenish
- bluish
- purplish
- bluish purple
- reddish orange
- yellowish green

**Tips**

For a list of some valid colors, see [Link to Valid Colors to use with cascading style sheets](#).

To see how color names map to hexadecimal values, submit the following REGISTRY procedure code:

```sas
proc registry list startat="COLORNAMES";
run;
```

**See** [RBG Color Codes, HLS Color Codes, and Gray-Scale Color codes in SAS/GRAPH: Reference](#) for information about SAS/GRAPH colors.

**dimension**

is a whole number, a percentage, or a nonnegative number followed by one of these units of measure:
Table 12.7 Units of Measure for Dimension

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>Centimeters</td>
</tr>
<tr>
<td>em</td>
<td>Standard typesetting measurement unit for width</td>
</tr>
<tr>
<td>ex</td>
<td>Standard typesetting measurement unit for height</td>
</tr>
<tr>
<td>in</td>
<td>Inches</td>
</tr>
<tr>
<td>mm</td>
<td>Millimeters</td>
</tr>
<tr>
<td>pt</td>
<td>A printer's point</td>
</tr>
</tbody>
</table>

Default: For the PRINTER destination, units of 1/150 of an inch

### font-definition

is the name of a font, the font size, and font keywords. A font definition has this general format:

```
("font-face-1 <… ,font-face-n">", font-size, keyword-list)
```

"font-face"

specifies the name of the font.

ODS styles can now use new TrueType fonts. All Universal Printers and many SAS/GRAPH devices use the FreeType library to render TrueType fonts for output in all of the operating environments that SAS software supports. In addition, by default, many SAS/GRAPH device drivers and all Universal Printers generate output using ODS styles, and these ODS styles use TrueType fonts. In addition to SAS Monospace and SAS Monospace Bold, 21 new TrueType fonts are made available when you install SAS:

- five Latin fonts compatible with Microsoft
- eight multilingual Unicode fonts
- eight monolingual Asian fonts

For more information about the TrueType fonts, see the section "Printing with SAS" in SAS Language Reference: Concepts.

### Restriction

You must enclose multiple font-face in quotation marks. If you specify only one font and if its name does not include a space character, then omit the quotation marks.

### Tip

If you specify more than one font, then the destination device uses the first one that is installed on the system.

### font-size

specifies the size of the font. font-size is a dimension or a number without units of measure. If you specify a dimension, then specify a unit of measure. Without a unit of measure the number becomes a size that is relative to all other font sizes in the HTML document. For more information, see dimension attribute value on page 1039.
specifies the font weight, font style, and font width. Include one value for each, in any order. This table shows the keywords to use:

Table 12.8 Font Keywords

<table>
<thead>
<tr>
<th>Keywords for Font Weight</th>
<th>Keywords for Font Style</th>
<th>Keywords for Font Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEDIUM</td>
<td>ITALIC</td>
<td>NORMAL*</td>
</tr>
<tr>
<td>BOLD</td>
<td>ROMAN</td>
<td>COMPRESSED*</td>
</tr>
<tr>
<td>DEMI_BOLD*</td>
<td>SLANT</td>
<td>EXTRA_COMPRESSED*</td>
</tr>
<tr>
<td>EXTRA_BOLD*</td>
<td></td>
<td>NARROW*</td>
</tr>
<tr>
<td>LIGHT</td>
<td></td>
<td>WIDE*</td>
</tr>
<tr>
<td>DEMI_LIGHT*</td>
<td></td>
<td>EXPANDED*</td>
</tr>
<tr>
<td>EXTRA_LIGHT*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Few fonts honor these values.


format
is a SAS format or a user-defined format.

integer | integer-list | integer-column-list
 specifies a column variable that contains integer values, or a dynamic variable that refers to such a column variable.

integer
specifies a single integer.

integer-list
specifies a sequence of integer values, or a column variable that contains integer values, or a dynamic variable that refers to such a column variable or to a string.

integer-column-list
specifies a sequence of column variables, or a column variable that contains column variables, or a dynamic variable that refers to such a column variable, or a dynamic variable that refers to a string containing a list of column variables. Values within the columns must be integers.

style-reference
is a reference to an attribute that is defined in the current style or in the parent style (or beyond). The value used is the name of the style element followed by the name of an attribute, in parentheses, within that element. Style references have the following form:

style-attribute=target-style-element("target-style-attribute")

style-attribute
specifies the name of the style attribute.
target-style-element
specifies the name of the style element that contains the style attribute that you want to reference.

target-style-attribute
specifies the style attribute with the value that you want to use.

Requirement  You must enclose target-style-attribute in quotation marks if it is a user-supplied style attribute.

See
“Understanding Style References” in SAS Output Delivery System: Procedures Guide

Example

"string"
is a quoted character string.

user-defined-format
specifies a format created with the FORMAT procedure.

Restriction  user-defined-format can only be specified for data cells.
Appendices

Appendix 1
   Output Object Table Names ........................................... 1045

Appendix 2
   Example Programs ......................................................... 1067

Appendix 3
   ODS and the HTML Destination ........................................ 1101

Appendix 4
   ODS HTML Statements for Running Examples in
   Different Operating Environments .................................. 1113

Appendix 5
   PowerPoint Transition Options and the Effect
   Options Supported ....................................................... 1115
Appendix 1
Output Object Table Names

ODS Table Names and the Base SAS Procedures That Produce Them

This table lists the output object table names that Base SAS procedures produce. The table provides the name of each table, a description of what the table contains, and the option, if any, that creates the output object table.

Table A1.1  ODS Table Names Produced by the CALENDAR Procedure

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calendar</td>
<td>Calendar</td>
</tr>
</tbody>
</table>
### Table A1.2 ODS Table Names Produced by the CATALOG Procedure

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog_Random</td>
<td>Table generated when the catalog is in a random-access data library</td>
</tr>
<tr>
<td>Catalog_Sequential</td>
<td>Table generated when the catalog is in a sequential-access data library</td>
</tr>
</tbody>
</table>

### Table A1.3 ODS Table Names Produced by the CHART Procedure

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block</td>
<td>Block chart</td>
</tr>
<tr>
<td>Hbar</td>
<td>Horizontal bar chart</td>
</tr>
<tr>
<td>Pie</td>
<td>Pie chart</td>
</tr>
<tr>
<td>Star</td>
<td>Star chart</td>
</tr>
<tr>
<td>Vbar</td>
<td>Vertical bar chart</td>
</tr>
</tbody>
</table>

### Table A1.4 ODS Table Names Produced by the COMPARE Procedure

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompareDatasets</td>
<td>Information about the data set or data sets</td>
<td>Omit NOSUMMARY or NOVALUE options</td>
</tr>
<tr>
<td>CompareDetails (Comparison results for observations)</td>
<td>List of observations that the base data set and the compare data set do not have in common</td>
<td>PRINTALL</td>
</tr>
<tr>
<td>CompareDifferences</td>
<td>Report of variable value differences</td>
<td>Omit NOVALUES option</td>
</tr>
<tr>
<td>CompareSummary</td>
<td>Summary report of observations, values, and variables of unequal values</td>
<td></td>
</tr>
</tbody>
</table>
### Table Name | Description | Option
---|---|---
CompareVariables | List of differences in variable types or attributes between the base data set and the compare data set | Omit NOSUMMARY option unless the variables are identical

**ODS Tables Created by the ID Statement**

| Table Name | Description | Option |
---|---|---|
CompareDetails | List of notes and warnings concerning duplicate ID variable values if duplicate ID variable values exist in either of the data sets | | 

### Table A1.5  ODS Table Names Produced by the CORR Procedure

| Table Name | Description | Option |
---|---|---|
Cov | Covariances | COV |
CronbachAlpha | Coefficient alpha | ALPHA |
CronbachAlphaDel | Coefficient alpha with deleted variable | ALPHA |
Csscp | Corrected sums of squares and crossproducts | CSSCP |
FisherPearsonCorr | Pearson correlation statistics using Fisher’s z transformation | FISHER |
FisherSpearmanCorr | Spearman correlation statistics using Fisher’s z transformation | FISHER SPEARMAN |
HoeffdingCorr | Hoeffding’s D statistics | HOEFFDING |
KendallCorr | Kendall’s tau-b coefficients | KENDALL |
PearsonCorr | Pearson correlations | PEARSON |
PolychoricCorr | Polychoric correlations | POLYCHORIC |
PolyserialCorr | Polyserial correlations | POLYSERIAL |
SimpleStats | Simple descriptive statistics | |
### Table Name | Description | Option
--- | --- | ---
SpearmanCorr | Spearman correlations | SPEARMAN
Sscp | Sums of squares and crossproducts | SSCP
VarInformation | Variable information |

**ODS Tables Created by the PARTIAL Statement**

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>FisherPearsonPartialCorr</td>
<td>Pearson partial correlation statistics using Fisher’s $z$ transformation</td>
<td>FISHER</td>
</tr>
<tr>
<td>FisherSpearmanPartialCorr</td>
<td>Spearman partial correlation statistics using Fisher’s $z$ transformation</td>
<td>FISHER SPEARMAN</td>
</tr>
<tr>
<td>PartialCsscp</td>
<td>Partial corrected sums of squares and crossproducts</td>
<td>CSSCP</td>
</tr>
<tr>
<td>PartialCov</td>
<td>Partial covariances</td>
<td>COV</td>
</tr>
<tr>
<td>PartialKendallCorr</td>
<td>Partial Kendall tau-b coefficients</td>
<td>KENDALL</td>
</tr>
<tr>
<td>PartialPearsonCorr</td>
<td>Partial Pearson correlations</td>
<td></td>
</tr>
<tr>
<td>PartialSpearmanCorr</td>
<td>Partial Spearman correlations</td>
<td>SPEARMAN</td>
</tr>
</tbody>
</table>

### Table A1.6  ODS Table Names Produced by the DATASETS and CONTENTS Procedures

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory</td>
<td>General library information</td>
<td>Omit NOLIST option</td>
</tr>
<tr>
<td>Members</td>
<td>Library member information</td>
<td>Omit NOLIST option</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
<td>Option</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Attributes</td>
<td>Data set attributes</td>
<td>Omit SHORT option</td>
</tr>
<tr>
<td>Directory</td>
<td>General library information</td>
<td>DATA=&lt;libref:<em>ALL</em> or the DIRECTORY option</td>
</tr>
<tr>
<td>EngineHost</td>
<td>Engine and operating environment information</td>
<td>Omit SHORT option</td>
</tr>
<tr>
<td>IntegrityConstraints</td>
<td>List of integrity constraints</td>
<td>Omit SHORT option and data has integrity constraints</td>
</tr>
<tr>
<td>IntegrityConstraintsShort</td>
<td>Concise listing of integrity constraints</td>
<td>SHORT option specified and data has integrity constraints</td>
</tr>
<tr>
<td>Indexes</td>
<td>List of indexes</td>
<td>Omit SHORT option and data set is indexed</td>
</tr>
<tr>
<td>IndexesShort</td>
<td>Concise list of indexes</td>
<td>SHORT option specified and data set is indexed</td>
</tr>
<tr>
<td>Members</td>
<td>Library member information</td>
<td>DATA=&lt;libref:<em>ALL</em> or the DIRECTORY option</td>
</tr>
<tr>
<td>Position</td>
<td>List of variables by logical position in the data set</td>
<td>Omit SHORT option and specify the VARNUM option</td>
</tr>
<tr>
<td>PositionShort</td>
<td>Concise list of variables by logical position in the data set</td>
<td>SHORT and VARNUM options</td>
</tr>
<tr>
<td>Sortedby</td>
<td>Sort information</td>
<td>Omit SHORT option and data set is sorted</td>
</tr>
<tr>
<td>SortedbyShort</td>
<td>Concise sort information</td>
<td>SHORT option and data set is sorted</td>
</tr>
<tr>
<td>Variables</td>
<td>List of variables in alphabetical order</td>
<td>Omit SHORT option</td>
</tr>
<tr>
<td>VariablesShort</td>
<td>Concise listing of variables in alphabetical order</td>
<td>SHORT</td>
</tr>
</tbody>
</table>

* For PROC DATASETS, if both the NOLIST option and either the DIRECTORY option or DATA=<libref:_ALL_ are specified, then the NOLIST option is ignored.
### Table A1.8 ODS Table Names Produced by the FREQ Procedure

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>BarnardsTest</td>
<td>Barnard’s exact test</td>
<td>EXACT</td>
<td>BARNARD</td>
</tr>
<tr>
<td>BinomialCLs</td>
<td>Binomial confidence limits</td>
<td>TABLES</td>
<td>BINOMIAL(AC</td>
</tr>
<tr>
<td>BinomialEquiv</td>
<td>Binomial equivalence analysis</td>
<td>TABLES</td>
<td>BINOMIAL(EQUIV)</td>
</tr>
<tr>
<td>BinomialEquivLimits</td>
<td>Binomial equivalence limits</td>
<td>TABLES</td>
<td>BINOMIAL(EQUIV)</td>
</tr>
<tr>
<td>BinomialEquivTest</td>
<td>Binomial equivalence test</td>
<td>TABLES</td>
<td>BINOMIAL(EQUIV)</td>
</tr>
<tr>
<td>BinomialNoninf</td>
<td>Binomial noninferiority test</td>
<td>TABLES</td>
<td>BINOMIAL(NONINF)</td>
</tr>
<tr>
<td>Binomial</td>
<td>Binomial proportion</td>
<td>TABLES</td>
<td>BINOMIAL</td>
</tr>
<tr>
<td>BinomialTest</td>
<td>Binomial proportion test</td>
<td>TABLES</td>
<td>BINOMIAL</td>
</tr>
<tr>
<td>BinomialSup</td>
<td>Binomial superiority test</td>
<td>TABLES</td>
<td>BINOMIAL(SUP)</td>
</tr>
<tr>
<td>BreslowDayTest</td>
<td>Breslow-Day test</td>
<td>TABLES</td>
<td>CMH (h x 2 x 2 table)</td>
</tr>
<tr>
<td>CMH</td>
<td>Cochran-Mantel-Haenszel statistics</td>
<td>TABLES</td>
<td>CMH</td>
</tr>
<tr>
<td>ChiSq</td>
<td>Chi-square tests</td>
<td>TABLES</td>
<td>CHISQ</td>
</tr>
<tr>
<td>CochransQ</td>
<td>Cochran’s</td>
<td>TABLES</td>
<td>AGREE (h x 2 x 2 table)</td>
</tr>
<tr>
<td>ColScores</td>
<td>Column scores</td>
<td>TABLES</td>
<td>SCOROUT</td>
</tr>
<tr>
<td>CommonOddsRatioC1</td>
<td>Exact confidence limits for the common odds ratio</td>
<td>EXACT</td>
<td>COMOR (h x 2 x 2 table)</td>
</tr>
<tr>
<td>CommonOddsRatioTest</td>
<td>Common odds ratio exact test</td>
<td>EXACT</td>
<td>COMOR (h x 2 x 2 table)</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
<td>Statement</td>
<td>Option</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------</td>
<td>-----------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>CommonPdiff</td>
<td>Common proportion difference</td>
<td>TABLES</td>
<td>RISKDIFF(COMMON)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(h \times 2 \times 2$ table)</td>
</tr>
<tr>
<td>CommonRelRisks</td>
<td>Common relative risks</td>
<td>TABLES</td>
<td>CMH</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(h \times 2 \times 2$ table)</td>
</tr>
<tr>
<td>CrossList</td>
<td>Crosstabulation table in column format</td>
<td>TABLES</td>
<td>CROSSLIST</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(n$-way table, $n &gt; 1$)</td>
</tr>
<tr>
<td>CrossTabFreqs</td>
<td>Crosstabulation table</td>
<td>TABLES</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(n$-way table, $n &gt; 1$)</td>
</tr>
<tr>
<td>EqualKappaTest</td>
<td>Test for equal simple kappas</td>
<td>TABLES</td>
<td>AGREE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(h \times 2 \times 2$ table)</td>
</tr>
<tr>
<td>EqualKappaTests</td>
<td>Tests for equal kappas</td>
<td>TABLES</td>
<td>AGREE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(h \times r \times r, r &gt; 2$)</td>
</tr>
<tr>
<td>EqualOddsRatios</td>
<td>Tests for equal odds ratios</td>
<td>EXACT</td>
<td>EQOR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(h \times 2 \times 2$ table)</td>
</tr>
<tr>
<td>GailSimon</td>
<td>Gail-Simon test</td>
<td>TABLES</td>
<td>GAILSIMON</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$(h \times 2 \times 2$ table)</td>
</tr>
<tr>
<td>FishersExact</td>
<td>Fisher’s exact test</td>
<td>EXACT</td>
<td>FISHER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FISHER or EXACT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CHISQ $(2 \times 2$ table)</td>
</tr>
<tr>
<td>FishersExactMC</td>
<td>Monte Carlo estimates for Fisher’s exact test</td>
<td>EXACT</td>
<td>FISHER / MC</td>
</tr>
<tr>
<td>Gamma</td>
<td>Gamma</td>
<td>TEST</td>
<td>GAMMA</td>
</tr>
<tr>
<td>GammaTest</td>
<td>Gamma test</td>
<td>TEST</td>
<td>GAMMA</td>
</tr>
<tr>
<td>JTTTest</td>
<td>Jonckheere-Terpstra test</td>
<td>TABLES</td>
<td>JT</td>
</tr>
<tr>
<td>JTTTestMC</td>
<td>Monte Carlo estimates for Jonckheere-Terpstra</td>
<td>EXACT</td>
<td>JT / MC</td>
</tr>
<tr>
<td></td>
<td>exact test</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kappa</td>
<td>Simple kappa coefficient</td>
<td>TEST</td>
<td>KAPPA</td>
</tr>
<tr>
<td></td>
<td>or EXACT</td>
<td></td>
<td>KAPPA</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
<td>Statement</td>
<td>Option</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------</td>
<td>------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>KappaMC</td>
<td>Monte Carlo exact test for simple kappa coefficient</td>
<td>EXACT</td>
<td>KAPPA / MC</td>
</tr>
<tr>
<td>KappaStatistics</td>
<td>Kappa statistics</td>
<td>TABLES</td>
<td>AGREE, no TEST or EXACT (r × r table, &gt; 2)</td>
</tr>
<tr>
<td>KappaTest</td>
<td>Simple kappa test</td>
<td>TEST or EXACT</td>
<td>KAPPA KAPPA</td>
</tr>
<tr>
<td>KappaWeights</td>
<td>Kappa weights</td>
<td>TABLES</td>
<td>AGREE(PRINTKW TS)</td>
</tr>
<tr>
<td>List</td>
<td>List format multiway table</td>
<td>TABLES</td>
<td>LIST</td>
</tr>
<tr>
<td>LRChiSq</td>
<td>Likelihood ratio chi-square exact test</td>
<td>EXACT</td>
<td>LRCHI</td>
</tr>
<tr>
<td>LRChiSqMC</td>
<td>Monte Carlo exact test for likelihood ratio chi-square</td>
<td>EXACT</td>
<td>LRCHI / MC</td>
</tr>
<tr>
<td>MantelFleiss</td>
<td>Mantel-Fleiss criterion</td>
<td>TABLES</td>
<td>CMH(MF) (h × 2 × 2 table)</td>
</tr>
<tr>
<td>McNemarsTest</td>
<td>McNemar’s test</td>
<td>TABLES</td>
<td>AGREE (2 × 2 table)</td>
</tr>
<tr>
<td>Measures</td>
<td>Measures of association</td>
<td>TABLES</td>
<td>MEASURES</td>
</tr>
<tr>
<td>MHChiSq</td>
<td>Mantel-Haenszel chi-square exact test</td>
<td>EXACT</td>
<td>MHCHI</td>
</tr>
<tr>
<td>MHChiSqMC</td>
<td>Monte Carlo exact test for Mantel-Haenszel chi-square</td>
<td>EXACT</td>
<td>MHCHI / MC</td>
</tr>
<tr>
<td>NLevels</td>
<td>Number of variable levels</td>
<td>PROC</td>
<td>NLEVELS</td>
</tr>
<tr>
<td>OddsRatioCLs</td>
<td>Odds ratio confidence limits</td>
<td>TABLES</td>
<td>OR(CL=) (2 × 2 table)</td>
</tr>
<tr>
<td>OddsRatioExactCL</td>
<td>Exact confidence limits for the odds ratio</td>
<td>EXACT</td>
<td>OR (2 × 2 table)</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
<td>Statement</td>
<td>Option</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------</td>
<td>-------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>OneWayChiSq</td>
<td>One-way chi-square test</td>
<td>TABLES</td>
<td>CHISQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(one-way table)</td>
</tr>
<tr>
<td>OneWayChiSqMC</td>
<td>Monte Carlo exact test for one-way chi-square</td>
<td>EXACT</td>
<td>CHISQ / MC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(one-way table)</td>
</tr>
<tr>
<td>OneWayFreqs</td>
<td>One-way frequencies</td>
<td>PROC or TABLES</td>
<td>(no TABLES stmt)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(one-way table)</td>
</tr>
<tr>
<td>OneWayLRChiSq</td>
<td>One-way likelihood ratio chi-square test</td>
<td>TABLES</td>
<td>CHISQ(LRCHI)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(one-way table)</td>
</tr>
<tr>
<td>OverallKappa</td>
<td>Overall simple kappa</td>
<td>TABLES</td>
<td>AGREE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>($h \times 2 \times 2$ table)</td>
</tr>
<tr>
<td>OverallKappas</td>
<td>Overall kappa coefficients</td>
<td>TABLES</td>
<td>AGREE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>($h \times r \times r$, $&gt; 2$)</td>
</tr>
<tr>
<td>PdiffCLs</td>
<td>Proportion difference confidence limits</td>
<td>TABLES</td>
<td>RISKDIFF(CL=)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2 $\times 2$ table)</td>
</tr>
<tr>
<td>PdiffEquiv</td>
<td>Equivalence analysis for the proportion difference</td>
<td>TABLES</td>
<td>RISKDIFF(EQUIV)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2 $\times 2$ table)</td>
</tr>
<tr>
<td>PdiffEquivLimits</td>
<td>Equivalence limits for the proportion difference</td>
<td>TABLES</td>
<td>RISKDIFF(EQUIV)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2 $\times 2$ table)</td>
</tr>
<tr>
<td>PdiffEquivTest</td>
<td>Equivalence test for the proportion difference</td>
<td>TABLES</td>
<td>RISKDIFF(EQUIV)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2 $\times 2$ table)</td>
</tr>
<tr>
<td>PdiffNoninf</td>
<td>Noninferiority test for the proportion difference</td>
<td>TABLES</td>
<td>RISKDIFF(NONINF)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2 $\times 2$ table)</td>
</tr>
<tr>
<td>PdiffSup</td>
<td>Superiority test for the proportion difference</td>
<td>TABLES</td>
<td>RISKDIFF(SUP)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2 $\times 2$ table)</td>
</tr>
<tr>
<td>PdiffTest</td>
<td>Proportion difference test</td>
<td>TABLES</td>
<td>RISKDIFF(EQUAL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(2 $\times 2$ table)</td>
</tr>
<tr>
<td>PearsonChiSq</td>
<td>Pearson chi-square exact test</td>
<td>EXACT</td>
<td>PCHI</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
<td>Statement</td>
<td>Option</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------</td>
<td>-----------</td>
<td>------------------</td>
</tr>
<tr>
<td>PearsonChiSqMC</td>
<td>Monte Carlo exact test for Pearson chi-square</td>
<td>EXACT</td>
<td>PCHI / MC</td>
</tr>
<tr>
<td>PearsonCorr</td>
<td>Pearson correlation</td>
<td>TEST or EXACT</td>
<td>PCORR</td>
</tr>
<tr>
<td>PearsonCorrMC</td>
<td>Monte Carlo exact test for Pearson correlation</td>
<td>EXACT</td>
<td>PCORR / MC</td>
</tr>
<tr>
<td>PearsonCorrTest</td>
<td>Pearson correlation test</td>
<td>TEST or EXACT</td>
<td>PCORR</td>
</tr>
<tr>
<td>PlCorr</td>
<td>Polychoric correlation</td>
<td>TEST</td>
<td>PLCORR</td>
</tr>
<tr>
<td>PlCorrTest</td>
<td>Polychoric correlation test</td>
<td>TEST</td>
<td>PLCORR</td>
</tr>
<tr>
<td>RelativeRisks</td>
<td>Relative risk estimates</td>
<td>TABLES</td>
<td>RELRISK or MEASURES (2 × 2 table)</td>
</tr>
<tr>
<td>RelRisk1ExactCL</td>
<td>Exact confidence limits for column 1 relative risk</td>
<td>EXACT</td>
<td>RELRISK (2 × 2 table)</td>
</tr>
<tr>
<td>RelRisk2ExactCL</td>
<td>Exact confidence limits for column 2 relative risk</td>
<td>EXACT</td>
<td>RELRISK (2 × 2 table)</td>
</tr>
<tr>
<td>RiskDiffCol1</td>
<td>Column 1 risk estimates</td>
<td>TABLES</td>
<td>RISKDIFF (2 × 2 table)</td>
</tr>
<tr>
<td>RiskDiffCol2</td>
<td>Column 2 risk estimates</td>
<td>TABLES</td>
<td>RISKDIFF (2 × 2 table)</td>
</tr>
<tr>
<td>RowScores</td>
<td>Row scores</td>
<td>TABLES</td>
<td>SCOROUT</td>
</tr>
<tr>
<td>SomersDCR</td>
<td>Somers’ $D(C</td>
<td>R)$</td>
<td>TEST or EXACT</td>
</tr>
<tr>
<td>SomersDCRMC</td>
<td>Monte Carlo exact test for Somers’ $D(C</td>
<td>R)$</td>
<td>EXACT</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
<td>Statement</td>
<td>Option</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------</td>
<td>-----------</td>
<td>------------</td>
</tr>
<tr>
<td>SomersDCRTTest</td>
<td>Somers’ $D(C</td>
<td>R)$ test</td>
<td>TEST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or EXACT</td>
<td>SMDCR</td>
</tr>
<tr>
<td>SomersDRC</td>
<td>Somers’ $D(R</td>
<td>C)$</td>
<td>TEST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or EXACT</td>
<td>SMDRC</td>
</tr>
<tr>
<td>SomersDRCMC</td>
<td>Monte Carlo exact test for Somers’ $D(R</td>
<td>C)$</td>
<td>EXACT</td>
</tr>
<tr>
<td>SomersDRCTest</td>
<td>Somers’ $D(R</td>
<td>C)$ test</td>
<td>TEST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or EXACT</td>
<td>SMDRC</td>
</tr>
<tr>
<td>SpearmanCorr</td>
<td>Spearman correlation</td>
<td>TEST</td>
<td>SCORR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or EXACT</td>
<td>SCORR</td>
</tr>
<tr>
<td>SpearmanCorrMC</td>
<td>Monte Carlo exact test for Spearman correlation</td>
<td>EXACT</td>
<td>SCORR / MC</td>
</tr>
<tr>
<td>SpearmanCorrTest</td>
<td>Spearman correlation test</td>
<td>TEST</td>
<td>SCORR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or EXACT</td>
<td>SCORR</td>
</tr>
<tr>
<td>SymmetryTest</td>
<td>Test of symmetry</td>
<td>TABLES</td>
<td>AGREE</td>
</tr>
<tr>
<td>TauB</td>
<td>Kendall’s tau-</td>
<td>TEST</td>
<td>KENTB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or EXACT</td>
<td>KENTB</td>
</tr>
<tr>
<td>TauBMC</td>
<td>Monte Carlo exact test for Kendall’s tau-</td>
<td>EXACT</td>
<td>KENTB / MC</td>
</tr>
<tr>
<td>TauBTest</td>
<td>Kendall’s tau- test</td>
<td>TEST</td>
<td>KENTB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or EXACT</td>
<td>KENTB</td>
</tr>
<tr>
<td>TauC</td>
<td>Stuart’s tau-</td>
<td>TEST</td>
<td>STUTC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or EXACT</td>
<td>STUTC</td>
</tr>
<tr>
<td>TauCMC</td>
<td>Monte Carlo exact test for Stuart’s tau-</td>
<td>EXACT</td>
<td>STUTC / MC</td>
</tr>
<tr>
<td>TauCTest</td>
<td>Stuart’s tau- test</td>
<td>TEST</td>
<td>STUTC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or EXACT</td>
<td>STUTC</td>
</tr>
<tr>
<td>TrendTest</td>
<td>Cochran-Armitage trend test</td>
<td>TABLES</td>
<td>TREND</td>
</tr>
<tr>
<td>TrendTestMC</td>
<td>Monte Carlo exact test for trend</td>
<td>EXACT</td>
<td>TREND / MC</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
<td>Statement</td>
<td>Option</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>WtKappa</td>
<td>Weighted kappa coefficient</td>
<td>TEST or EXACT</td>
<td>WTKAP</td>
</tr>
<tr>
<td>WtKappaMC</td>
<td>Monte Carlo exact test for weighted kappa coefficient</td>
<td>EXACT</td>
<td>WTKAP / MC</td>
</tr>
<tr>
<td>WtKappaTest</td>
<td>Weighted kappa test</td>
<td>TEST or EXACT</td>
<td>WTKAP</td>
</tr>
</tbody>
</table>

**Table A1.9**  
*ODS Table Names Produced by the MEANS and SUMMARY Procedures*

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Summary</td>
<td>Summary of descriptive statistics for variables across all observations and within groups of observations</td>
</tr>
</tbody>
</table>

**Table A1.10**  
*ODS Table Names Produced by the PLOT Procedure*

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plot</td>
<td>Single plot graph</td>
<td></td>
</tr>
<tr>
<td>Overlaid</td>
<td>Two or more plots on a single set of axes</td>
<td>OVERLAY</td>
</tr>
</tbody>
</table>

**Table A1.11**  
*ODS Table Names Produced by the REPORT Procedure*

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>Detail report, summary report, or combination of both detail and summary information report</td>
</tr>
</tbody>
</table>
Table A1.12  ODS Table Names Produced by the SQL Procedure

For detailed information, see the SAS SQL Procedure User’s Guide.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL_Results</td>
<td>SAS data file or a SAS data view</td>
</tr>
</tbody>
</table>

Table A1.13  ODS Table Names Produced by the TABULATE Procedure

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Descriptive statistics in tabular format that use some or all of the variables in a data set</td>
</tr>
</tbody>
</table>

Table A1.14  ODS Table Names Produced by the TIMEPLOT Procedure

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plot</td>
<td>Single plot graph</td>
<td>Omit the OVERLAY option</td>
</tr>
<tr>
<td>OverlaidPlot</td>
<td>Two or more plots on a single set of axes</td>
<td>OVERLAY</td>
</tr>
</tbody>
</table>

Table A1.15  ODS Table Names Produced by the UNIVARIATE Procedure

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>BasicIntervals</td>
<td>Confidence intervals for mean, standard deviation, variance</td>
<td>CIBASIC</td>
</tr>
<tr>
<td>BasicMeasures</td>
<td>Measures of location and variability</td>
<td>Default</td>
</tr>
<tr>
<td>ExtremeObs</td>
<td>Extreme observations</td>
<td>Default</td>
</tr>
<tr>
<td>ExtremeValues</td>
<td>Extreme values</td>
<td>NEXTRAVAL=</td>
</tr>
<tr>
<td>Table Name</td>
<td>Description</td>
<td>Option</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Frequencies</td>
<td>Frequencies</td>
<td>FREQ</td>
</tr>
<tr>
<td>LocationCounts</td>
<td>Counts used for sign test and signed rank test</td>
<td>LOCCOUNT</td>
</tr>
<tr>
<td>Missing Values</td>
<td>Missing values</td>
<td>Default, if missing values exist</td>
</tr>
<tr>
<td>Modes</td>
<td>Modes</td>
<td>MODES</td>
</tr>
<tr>
<td>Moments</td>
<td>Sample moments</td>
<td>Default</td>
</tr>
<tr>
<td>Plots</td>
<td>Line printer plots</td>
<td>PLOTS</td>
</tr>
<tr>
<td>Quantiles</td>
<td>Quantiles</td>
<td>Default</td>
</tr>
<tr>
<td>RobustScale</td>
<td>Robust measures of scale</td>
<td>ROBUSTSCALE</td>
</tr>
<tr>
<td>SSPlots</td>
<td>Line printer side-by-side box plot</td>
<td>PLOTS with BY statement</td>
</tr>
<tr>
<td>TestsForLocation</td>
<td>Tests for location</td>
<td>Default</td>
</tr>
<tr>
<td>TestsForNormality</td>
<td>Tests for normality</td>
<td>NORMALTEST</td>
</tr>
<tr>
<td>TrimmedMeans</td>
<td>Trimmed means</td>
<td>TRIMMED=</td>
</tr>
<tr>
<td>WinsorizedMeans</td>
<td>Winsorized means</td>
<td>WINSORIZED=</td>
</tr>
</tbody>
</table>

**ODS Tables Created by the HISTOGRAM Statement**

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bins</td>
<td>Histogram bins</td>
<td>MIDPERCENTS secondary option</td>
</tr>
<tr>
<td>FitQuantiles</td>
<td>Quantiles of fitted distribution</td>
<td>Any distribution option</td>
</tr>
<tr>
<td>GoodnessOfFit</td>
<td>Goodness-of-fit tests for fitted distribution</td>
<td>Any distribution option</td>
</tr>
<tr>
<td>HistogramBins</td>
<td>Histogram bins</td>
<td>MIDPERCENTS option</td>
</tr>
<tr>
<td>ParameterEstimates</td>
<td>Parameter estimates for fitted distribution</td>
<td>Any distribution option</td>
</tr>
</tbody>
</table>
### ODS Table Names and the Base SAS High-Performance Procedures That Produce Them

For detailed information, see the procedures in *Base SAS Procedures Guide: High-Performance Procedures*

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPBIN</td>
<td><em>ODS Tables Produced by PROC HPBIN</em></td>
</tr>
<tr>
<td>HPCORR</td>
<td><em>ODS Tables Produced by PROC HPCORR</em></td>
</tr>
<tr>
<td>HPIMPUTE</td>
<td><em>ODS Tables Produced by PROC HPIMPUTE</em></td>
</tr>
<tr>
<td>HPSAMPLE</td>
<td><em>ODS Tables Produced by PROC HPSAMPLE</em></td>
</tr>
</tbody>
</table>

### ODS Table Names and the Base SAS Statistical Procedures That Produce Them

For detailed information, see the procedures in *Base SAS Procedures Guide: Statistical Procedures*

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR</td>
<td><em>ODS Tables Produced by PROC CORR</em></td>
</tr>
<tr>
<td>FREQ</td>
<td><em>ODS Tables Produced by PROC FREQ</em></td>
</tr>
<tr>
<td>UNIVARIATE</td>
<td><em>ODS Tables Produced by PROC UNIVARIATE</em></td>
</tr>
</tbody>
</table>

### ODS Table Names and the SAS/STAT Procedures That Produce Them

For detailed information, see the procedures in *SAS/STAT User's Guide*

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACELUS</td>
<td><em>ODS Tables Produced by PROC ACELUS</em></td>
</tr>
</tbody>
</table>
For detailed information, see the procedures in SAS/STAT User’s Guide

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADAPTIVEREG</td>
<td>ODS Tables Produced by PROC ADAPTIVEREG</td>
</tr>
<tr>
<td>ANOVA</td>
<td>ODS Tables Produced by PROC ANOVA</td>
</tr>
<tr>
<td>BCHOICE</td>
<td>ODS Tables Produced by PROC BCHOICE</td>
</tr>
<tr>
<td>CALIS</td>
<td>ODS Tables Produced by PROC CALIS</td>
</tr>
<tr>
<td>CANCORR</td>
<td>ODS Tables Produced by PROC CANCORR</td>
</tr>
<tr>
<td>CANDISC</td>
<td>ODS Tables Produced by PROC CANDISC</td>
</tr>
<tr>
<td>CATMOD</td>
<td>ODS Tables Produced by PROC CATMOD</td>
</tr>
<tr>
<td>CLUSTER</td>
<td>ODS Tables Produced by PROC CLUSTER</td>
</tr>
<tr>
<td>CORRESP</td>
<td>ODS Tables Produced by PROC CORRESP</td>
</tr>
<tr>
<td>DISCRIM</td>
<td>ODS Tables Produced by PROC DISCRIM</td>
</tr>
<tr>
<td>FACTOR</td>
<td>ODS Tables Produced by PROC FACTOR</td>
</tr>
<tr>
<td>FASTCLUS</td>
<td>ODS Tables Produced by PROC FASTCLUS</td>
</tr>
<tr>
<td>FMM</td>
<td>ODS Tables Produced by PROC FMM</td>
</tr>
<tr>
<td>FREQ</td>
<td>ODS Tables Produced by PROC FREQ</td>
</tr>
<tr>
<td>GAM</td>
<td>ODS Tables Produced by PROC GAM</td>
</tr>
<tr>
<td>GEE</td>
<td>ODS Tables Produced by PROC GEE</td>
</tr>
<tr>
<td>GLMMOD</td>
<td>ODS Tables Produced by PROC GLMMOD</td>
</tr>
<tr>
<td>GLMSELECT</td>
<td>ODS Tables Produced by PROC GLMSELECT</td>
</tr>
<tr>
<td>HPMIXED</td>
<td>ODS Tables Produced by PROC HPMIXED</td>
</tr>
<tr>
<td>ICLIFETEST</td>
<td>ODS Tables Produced by PROC ICLIFETEST</td>
</tr>
<tr>
<td>ICPHREG</td>
<td>ODS Tables Produced by PROC ICPHREG</td>
</tr>
<tr>
<td>INBREED</td>
<td>ODS Tables Produced by PROC INBREED</td>
</tr>
<tr>
<td>IRT</td>
<td>ODS Tables Produced by PROC IRT</td>
</tr>
<tr>
<td>KDE</td>
<td>ODS Tables Produced by PROC KDE</td>
</tr>
</tbody>
</table>
### ODS Table Names and the SAS/STAT Procedures That Produce Them

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>KRIGE2D</td>
<td>ODS Tables Produced by PROC KRIGE2D</td>
</tr>
<tr>
<td>LATTICE</td>
<td>ODS Tables Produced by PROC LATTICE</td>
</tr>
<tr>
<td>LIFEREG</td>
<td>ODS Tables Produced by PROC LIFEREG</td>
</tr>
<tr>
<td>LIFETEST</td>
<td>ODS Tables Produced by PROC LIFETEST</td>
</tr>
<tr>
<td>LOESS</td>
<td>ODS Tables Produced by PROC LOESS</td>
</tr>
<tr>
<td>LOGISTIC</td>
<td>ODS Tables Produced by PROC LOGISTIC</td>
</tr>
<tr>
<td>MCMC</td>
<td>ODS Tables Produced by PROC MCMC</td>
</tr>
<tr>
<td>MDS</td>
<td>ODS Tables Produced by PROC MDS</td>
</tr>
<tr>
<td>MI</td>
<td>ODS Tables Produced by PROC MI</td>
</tr>
<tr>
<td>MIANALYZE</td>
<td>ODS Tables Produced by PROC MIANALYZE</td>
</tr>
<tr>
<td>MIXED</td>
<td>ODS Tables Produced by PROC MIXED</td>
</tr>
<tr>
<td>MODECLUS</td>
<td>ODS Tables Produced by PROC MODECLUS</td>
</tr>
<tr>
<td>MULTTEST</td>
<td>ODS Tables Produced by PROC MULTTEST</td>
</tr>
<tr>
<td>NESTED</td>
<td>ODS Tables Produced by PROC NESTED</td>
</tr>
<tr>
<td>NLIN</td>
<td>ODS Tables Produced by PROC NLIN</td>
</tr>
<tr>
<td>NLMIXED</td>
<td>ODS Tables Produced by PROC NLMIXED</td>
</tr>
<tr>
<td>NPAR1WAY</td>
<td>ODS Tables Produced by PROC NPAR1WAY</td>
</tr>
<tr>
<td>ORTHOREG</td>
<td>ODS Tables Produced by PROC ORTHOREG</td>
</tr>
<tr>
<td>PHREG</td>
<td>ODS Tables Produced by PROC PHREG</td>
</tr>
<tr>
<td>PLAN</td>
<td>ODS Tables Produced by PROC PLAN</td>
</tr>
<tr>
<td>PLM</td>
<td>ODS Tables Produced by PROC PLM</td>
</tr>
<tr>
<td>PLS</td>
<td>ODS Tables Produced by PROC PLS</td>
</tr>
<tr>
<td>POWER</td>
<td>ODS Tables Produced by PROC POWER</td>
</tr>
<tr>
<td>PRINCOMP</td>
<td>ODS Tables Produced by PROC PRINCOMP</td>
</tr>
</tbody>
</table>
## Output Object Table Names

For detailed information, see the procedures in [SAS/STAT User’s Guide](https://的支持页面链接).

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINQUAL</td>
<td>ODS Tables Produced by PROC PRINQUAL</td>
</tr>
<tr>
<td>PROBIT</td>
<td>ODS Tables Produced by PROC PROBIT</td>
</tr>
<tr>
<td>QUANTLIFE</td>
<td>ODS Tables Produced by PROC QUANTLIFE</td>
</tr>
<tr>
<td>QUANTREG</td>
<td>ODS Tables Produced by PROC QUANTREG</td>
</tr>
<tr>
<td>QUANTSELECT</td>
<td>ODS Tables Produced by PROC QUANTSELECT</td>
</tr>
<tr>
<td>REG</td>
<td>ODS Tables Produced by PROC REG</td>
</tr>
<tr>
<td>ROBUSTREG</td>
<td>ODS Tables Produced by PROC ROBUSTREG</td>
</tr>
<tr>
<td>RSREG</td>
<td>ODS Tables Produced by PROC RSREG</td>
</tr>
<tr>
<td>SEQDESIGN</td>
<td>ODS Tables Produced by PROC SEQDESIGN</td>
</tr>
<tr>
<td>SEQTEST</td>
<td>ODS Tables Produced by PROC SEQTEST</td>
</tr>
<tr>
<td>SIM2D</td>
<td>ODS Tables Produced by PROC SIM2D</td>
</tr>
<tr>
<td>SPP</td>
<td>ODS Tables Produced by PROC SPP</td>
</tr>
<tr>
<td>STDIZE</td>
<td>ODS Tables Produced by PROC STDIZE</td>
</tr>
<tr>
<td>STDRATE</td>
<td>ODS Tables Produced by PROC STDRATE</td>
</tr>
<tr>
<td>STEPDISC</td>
<td>ODS Tables Produced by PROC STEPDISC</td>
</tr>
<tr>
<td>SURVEYFREQ</td>
<td>ODS Tables Produced by PROC SURVEYFREQ</td>
</tr>
<tr>
<td>SURVEYIMPUTE</td>
<td>ODS Tables Produced by PROC SURVEYIMPUTE</td>
</tr>
<tr>
<td>SURVEYLOGISTIC</td>
<td>ODS Tables Produced by PROC SURVEYLOGISTIC</td>
</tr>
<tr>
<td>SURVEYMEANS</td>
<td>ODS Tables Produced by PROC SURVEYMEANS</td>
</tr>
<tr>
<td>SURVEYPHREG</td>
<td>ODS Tables Produced by PROC SURVEYPHREG</td>
</tr>
<tr>
<td>SURVEYREG</td>
<td>ODS Tables Produced by PROC SURVEYREG</td>
</tr>
<tr>
<td>SURVEYSELECT</td>
<td>ODS Tables Produced by PROC SURVEYSELECT</td>
</tr>
<tr>
<td>TPSPLINE</td>
<td>ODS Tables Produced by PROC TPSPLINE</td>
</tr>
<tr>
<td>TRANSREG</td>
<td>ODS Tables Produced by PROC TRANSREG</td>
</tr>
<tr>
<td>Procedure</td>
<td>ODS Table Names</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>TREE</td>
<td>ODS Tables Produced by PROC TREE</td>
</tr>
<tr>
<td>TTEST</td>
<td>ODS Tables Produced by PROC TTEST</td>
</tr>
<tr>
<td>VARCLUS</td>
<td>ODS Tables Produced by PROC VARCLUS</td>
</tr>
<tr>
<td>VARCOMP</td>
<td>ODS Tables Produced by PROC VARCOMP</td>
</tr>
<tr>
<td>VARIOGRAM</td>
<td>ODS Tables Produced by PROC VARIOGRAM</td>
</tr>
<tr>
<td>HPCANDISC</td>
<td>ODS Tables Produced by PROC HPCANDISC</td>
</tr>
<tr>
<td>HPFMM</td>
<td>ODS Tables Produced by PROC HPFMM</td>
</tr>
<tr>
<td>HPGENSELECT</td>
<td>ODS Tables Produced by PROC HPGENSELECT</td>
</tr>
<tr>
<td>HPLMIXED</td>
<td>ODS Tables Produced by PROC HPLMIXED</td>
</tr>
<tr>
<td>HPLOGISTIC</td>
<td>ODS Tables Produced by PROC HPLOGISTIC</td>
</tr>
<tr>
<td>HPNLMOD</td>
<td>ODS Tables Produced by PROC HPNLMOD</td>
</tr>
<tr>
<td>HPPLS</td>
<td>ODS Tables Produced by PROC HPPLS</td>
</tr>
<tr>
<td>HPPRINCOMP</td>
<td>ODS Tables Produced by PROC HPPRINCOMP</td>
</tr>
<tr>
<td>HPQUANTSELECT</td>
<td>ODS Tables Produced by PROC HPQUANTSELECT</td>
</tr>
<tr>
<td>HPREG</td>
<td>ODS Tables Produced by PROC HPREG</td>
</tr>
</tbody>
</table>
ODS Table Names and the SAS/ETS Procedures That Produce Them

This table lists the output object table names that SAS/ETS procedures produce. You must license SAS/ETS software in order to produce these output objects. The table provides the name of each table, a description of what the table contains, and the option, if any, that creates the output object table. For more information about SAS/ETS procedures, see *SAS/ETS User’s Guide*.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARIMA</td>
<td><em>ODS Tables Produced by PROC ARIMA</em></td>
</tr>
<tr>
<td>AUTOREG</td>
<td><em>ODS Tables Produced by PROC AUTOREG</em></td>
</tr>
<tr>
<td>COPULA</td>
<td><em>ODS Tables Produced by PROC COPULA</em></td>
</tr>
<tr>
<td>COUNTREG</td>
<td><em>ODS Tables Produced by PROC COUNTREG</em></td>
</tr>
<tr>
<td>ENTROPY</td>
<td><em>ODS Tables Produced by PROC ENTROPY</em></td>
</tr>
<tr>
<td>ESM</td>
<td><em>ODS Tables Produced by PROC ESM</em></td>
</tr>
<tr>
<td>HPCOUNTREG</td>
<td><em>ODS Tables Produced by PROC HPCOUNTREG</em></td>
</tr>
<tr>
<td>HPPANEL</td>
<td><em>ODS Tables Produced by PROC HPPANEL</em></td>
</tr>
<tr>
<td>HPQLIM</td>
<td><em>ODS Tables Produced by PROC HPQLIM</em></td>
</tr>
<tr>
<td>LOAN</td>
<td><em>ODS Tables Produced by PROC LOAN</em></td>
</tr>
<tr>
<td>PANEL</td>
<td><em>ODS Tables Produced by PROC PANEL</em></td>
</tr>
<tr>
<td>QLIM</td>
<td><em>ODS Tables Produced by PROC QLIM</em></td>
</tr>
<tr>
<td>SIMILARITY</td>
<td><em>ODS Tables Produced by PROC SIMILARITY</em></td>
</tr>
<tr>
<td>SIMLIN</td>
<td><em>ODS Tables Produced by PROC SIMLIN</em></td>
</tr>
<tr>
<td>SPECTRA</td>
<td><em>ODS Tables Produced by PROC SPECTRA</em></td>
</tr>
<tr>
<td>SSM</td>
<td><em>ODS Tables Produced by PROC SSM</em></td>
</tr>
<tr>
<td>STATESPACE</td>
<td><em>ODS Tables Produced by PROC STATESPACE</em></td>
</tr>
<tr>
<td>SYSLIN</td>
<td><em>ODS Tables Produced by PROC SYSLIN</em></td>
</tr>
</tbody>
</table>
For detailed information, see the procedures in SAS/ETS User's Guide

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIMEDATA</td>
<td>ODS Tables Produced by PROC TIMEDATA</td>
</tr>
<tr>
<td>TIMESERIES</td>
<td>ODS Tables Produced by PROC TIMESERIES</td>
</tr>
<tr>
<td>TSCSREG</td>
<td>ODS Tables Produced by PROC TSCSREG</td>
</tr>
<tr>
<td>UCM</td>
<td>ODS Tables Produced by PROC UCM</td>
</tr>
<tr>
<td>VARMAX</td>
<td>ODS Tables Produced by PROC VARMAX</td>
</tr>
<tr>
<td>X11 Procedure</td>
<td>ODS Tables Produced by PROC X11</td>
</tr>
</tbody>
</table>

ODS Table Names and the SAS/ETS High-Performance Procedures That Produce Them

For detailed information, see the procedures in SAS/ETS User's Guide: High-Performance Procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPCOUNTREG</td>
<td>ODS Tables Produced by PROC HPCOUNTREG</td>
</tr>
<tr>
<td>HPPANEL</td>
<td>ODS Tables Produced by PROC HPPANEL</td>
</tr>
<tr>
<td>HPQLIM</td>
<td>ODS Tables Produced by PROC HPQLIM</td>
</tr>
</tbody>
</table>

ODS Table Names and the SAS/QC Procedures That Produce Them

For detailed information, see the procedures in SAS/QC User's Guide

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVPMODEL</td>
<td>ODS Tables Produced by PROC MVPMODEL</td>
</tr>
<tr>
<td>RELIABILITY</td>
<td>ODS Tables Produced by PROC RELIABILITY</td>
</tr>
</tbody>
</table>
Appendix 2
Example Programs

Examples from the Gallery of ODS Samples ........................................ 1068
  EPUB Output ................................................................. 1068
  Excel Output ............................................................... 1068
  HTML Output .............................................................. 1068
  PDF Output ................................................................. 1070
  PostScript Output .......................................................... 1072
  PowerPoint Output .......................................................... 1073
  RTF Output ................................................................. 1075
Creating the $CNTRY Format ......................................................... 1079
Creating the Charity Data Set ...................................................... 1079
Creating the DVFMT. and USETYPE. Formats ................................... 1082
Creating the DistrData Data Set ................................................... 1082
Creating the Univ ODS Document .................................................. 1082
Creating the Employee_Data Data Set ........................................... 1083
Creating the Energy Data Set ....................................................... 1085
Creating the Exprev Data Set ..................................................... 1085
Creating the Gov Data Set ......................................................... 1086
Creating the Grain_Production Data Set ......................................... 1087
Creating the One Data Set ........................................................ 1087
Creating the Iron Data Set ......................................................... 1088
Creating the Model Data Set ....................................................... 1088
Creating the Neuralgia Data Set ................................................... 1089
Creating the Plants Data Set ....................................................... 1090
Creating the Plant_Stats Data Set ............................................... 1090
Creating the StatePop Data Set .................................................... 1090
Creating the Stats and Stats2 Data Sets ......................................... 1092
Creating the Table1 Table Template .............................................. 1092
Programs That Illustrate Inheritance ............................................. 1093
  Overview ............................................................................. 1093
  Using the FROM option ...................................................... 1093
Examples from the Gallery of ODS Samples

**EPUB Output**

```sas
ods html close;
data drugtest;
  input Drug $ PreTreatment PostTreatment @@;
datalines;
A 11 6 A 8 0 A 5 2 A 14 8 A 19 11
A 6 4 A 10 13 A 6 1 A 11 8 A 3 0
D 6 0 D 6 2 D 7 3 D 8 1 D 18 18
D 8 4 D 19 14 D 8 9 D 5 1 D 15 9
F 16 13 F 13 10 F 11 18 F 9 5 F 21 23
F 16 12 F 12 5 F 12 16 F 7 1 F 12 20
;
ods epub file="glm.epub" title="My First ODS EPUB E-book" options(creator="SAS Programmer" description="My First ODS EPUB Book" subject="PROC GLM" type="ODS EPUB book");
ods graphics on;
proc glm data=DrugTest;
  class Drug;
  model PostTreatment = Drug|PreTreatment;
run;
quit;
ods epub close;
```

**Excel Output**

```sas
ods html close;
title "Custom Excel Output";
proc sort data=sashelp.cars out=cars;
  by make;
run;
ods tagsets.excelxp file="bylines.xls" style=htmlBlue
  options( suppress_bylines='yes' Embedded_Titles="yes" Sheet_Label="By" Frozen_Headers="yes");
proc print data=cars;
  var make model msrp invoice;
  by make;
run;
ods tagsets.excelxp close;
```

**HTML Output**

```sas
options nocenter;
```
```ods html style=barrettsblue;
title;
data;
  input region $ citysize $ pop product $ saletype $
         quantity amount;
datalines;
Brazil S 25000 A100 R 150   3750.00
Canada S 37000 A100 R 200   5000.00
France S 48000 A100 R 410  10250.00
Mexico S 32000 A100 R 180   4500.00
Brazil M 125000 A100 R 350  8750.00
Canada M 237000 A100 R 600  15000.00
France M 348000 A100 R 710  17750.00
Mexico M 432000 A100 R 780  19500.00
Canada L 837000 A100 R 800  20000.00
France L 748000 A100 R 760  19000.00
Mexico L 932000 A100 R 180   3600.00
Brazil S 25000 A200 R 157   3925.00
Canada S 37000 A200 R 208   5200.00
France S 48000 A200 R 419  10475.00
Mexico S 32000 A200 R 186   4650.00
Brazil S 25000 A300 R 157   3925.00
Canada L 837000 A300 R 208  5200.00
France S 48000 A300 R 419  10475.00
Mexico S 32000 A300 R 186   4650.00
Brazil M 125000 A300 R 351  8725.00
Canada M 237000 A300 R 610  15250.00
```

*Examples from the Gallery of ODS Samples*
France M 348000 A300 R 714 17850.00
Mexico M 432000 A300 R 785 19625.00
Canada L 837000 A300 R 806 20150.00
France L 748000 A300 R 768 19200.00
Mexico L 932000 A300 R 880 22000.00
Brazil S 25000 A300 W 157 3140.00
Canada S 37000 A300 W 208 4160.00
Mexico S 32000 A300 W 186 3720.00
Brazil M 125000 A300 W 351 7020.00
Canada M 237000 A300 W 610 12200.00
France M 348000 A300 W 714 14280.00
Mexico M 432000 A300 W 785 15700.00
Brazil L 625000 A300 W 757 15140.00
Canada L 837000 A300 W 806 16120.00
France L 748000 A300 W 768 15360.00
Mexico L 932000 A300 W 880 17600.00

; proc format;
  value $salefmt 'R'='Retail'
    'W'='Wholesale';
proc tabulate style={foreground=green background=white};
  class region citysize saletype     / style={foreground=black};
  classlev region citysize saletype / style={foreground=red};
  var quantity amount                / style={foreground=black};
  keyword all sum                    / style={foreground=purple};
  format saletype $salefmt.;
  label region="Region" citysize="Citysize" saletype="Saletype";
  label quantity="Quantity" amount="Amount";
  keylabel all="Total";
  table all={label = "All Products" style={foreground=orange font_weight=bold}},
             (region all )*(citysize all*{style={foreground=CX002288 font_weight=bold}}),
             (saletype all)*{quantity*f=COMMA6. amount*f=dollar10.} / 
               style={background=red} misstext={label="Missing" 
                 style={foreground=brown font_weight=bold}}
             box={label="Region by Citysize by Saletype" 
                 style={foreground=brown background=cxebdded}};
run;
ods html close;

PDF Output

options center nodate;
ods pdf body="b.pdf" style=barrettsblue;
title 'TABULATE With Custom ODS Styles';
data tabulate;
  input dept acct qtr mon expense @@;
datalines;
1 1345 1 1 12980 1 1674 1 3 13135 3 4138 1 1 29930
1 1345 1 1 9475 1 1674 1 3 21672 3 4138 1 2 22530
1 1345 1 1 15633 1 1674 1 3 3847 3 4138 1 2 16446
1 1345 1 2 14009 1 1674 1 3 2808 3 4138 1 2 27135
1 1345 1 2 10226 1 1674 1 3 4633 3 4138 1 3 24399
1 1345 1 2 16872 2 2134 1 1 34520 3 4138 1 1 17811
1 1345 1 2 16872 2 2134 1 1 34520 3 4138 1 1 17811
proc format;
  value qtrfmt 1 = 'FIRST QUARTER'
                   2 = 'SECOND QUARTER'
                   3 = 'THIRD QUARTER'
                   4 = 'FOURTH QUARTER';

  value monfmt 1 = 'January'
                2 = 'February'
                3 = 'March'
                4 = 'April'
                5 = 'May'
                6 = 'June'
                7 = 'July'
                8 = 'August'
                9 = 'September'
               10 = 'October'
               11 = 'November'
               12 = 'December';

  value dept   1 = 'Accounting'
                2 = 'Human Resources'
                3 = 'Systems';

proc tabulate format=dollar11.2;
  class mon qtr acct dept;
  classlev mon qtr acct dept / style={fontstyle=italic color=yellow};
  var   expense;
  format qtr qtrfmt.;
  format mon monfmt.;
  format dept dept.;
  label expense = "Expenses" dept = "Department";
  table dept {all="All Departments"}
PostScript Output

data grocery;
  input Sector $ Manager $ Department $ Sales @@;
datalines;
  se 1 np1 50  se 1 pl 100  se 1 np2 120  se 1 p2 80
  se 2 np1 40  se 2 pl 300  se 2 np2 220  se 2 p2 70
  nw 3 np1 60  nw 3 pl 600  nw 3 np2 420  nw 3 p2 30
  nw 4 np1 45  nw 4 pl 250  nw 4 np2 230  nw 4 p2 73
  nw 9 np1 45  nw 9 pl 205  nw 9 np2 420  nw 9 p2 76
  sw 5 np1 53  sw 5 pl 130  sw 5 np2 120  sw 5 p2 50
  sw 6 np1 40  sw 6 pl 350  sw 6 np2 225  sw 6 p2 80
  ne 7 np1 90  ne 7 pl 190  ne 7 np2 420  ne 7 p2 86
  ne 8 np1 200  ne 8 pl 300  ne 8 np2 420  ne 8 p2 125;
proc format;
  value $sctrfmt 'se' = 'Southeast'
    'ne' = 'Northeast'
    'nw' = 'Northwest'
    'sw' = 'Southwest';

  value $mgrfmt '1' = 'Malik'
    '2' = 'Chang'
    '3' = 'Reveiz'
    '4' = 'Brown'
    '5' = 'Taylor'
    '6' = 'Adams'
    '7' = 'Alomar'
    '8' = 'Andrews'
    '9' = 'Pelfrey';

  value $deptfmt 'np1' = 'Paper'
    'np2' = 'Canned'
    'p1' = 'Meat/Dairy'
    'p2' = 'Produce';
run;

title 'Sales for Malik and Chang';
libname proclib 'SAS-library';
options nodate pageno=1 fmtsearch=(proclib);
ods ps file='sales-ps-file.ps';
ods pdf close;
proc report data=grocery nowd headline headskip
  style(report)=[cellspacing=5 borderwidth=10 bordercolor=blue]
  style(column)=[foreground=moderate brown fontweight=bold
    fontface=helvetica fontsize=4]
  style(lines)=[foreground=white background=black
    fontstyle=italic fontweight=bold fontsize=5]
  style(summary)=[foreground=white background=cxaeadd9]
column manager department sales;
define manager / order
  order=formatted
  format=$mgrfmt.
  'Manager'style(header)=[foreground=cyan
  background=black];

define department / order
  order=internal
  format=$deptfmt.
  'Department'style(column)=[fontstyle=italic];

break after manager / summarize;
compute after manager
  / style=[fontstyle=roman fontsize=3 fontweight=bold
  background=white foreground=black];
  line 'Subtotal for ' manager $mgrfmt. 'is '
  sales.sum dollar7.2 '.';
endcomp;
compute sales;
if sales.sum>100 and _break_=' ' then
  call define(_col_, "style",
    "style=[background=#CCFF00
    fontface=helvetica
    fontweight=bold]");
endcomp;
compute after;
  line 'Total for all departments: '
  sales.sum dollar7.2 '.';
endcomp;
where sector='se';
run;

ods ps close;

PowerPoint Output

ods html close;
title1 'PowerPoint - Various Layouts and Styles';
footnote 'The PowerPoint Destination';

proc template;
  define style styles.test;
    parent= styles.powerpointlight;
    class body /
      backgroundimage="radial-gradient(40%, lightblue 40%,
      yellow 30%, blue)";
    style graphbackground / image='c:\Public\foldedblends.bmp';
  end;
run;
ods escapechar = "^";
ods PowerPoint file="powerptOptions.ppt" layout=titleslide
The ODS Destination for PowerPoint: 9.4 - The Power to Know

What Output is Produced by the ODS Destination for PowerPoint?

New features include:
- Light and dark styles
- Gradients: Linear, Angles, Opacity, Radial
- Template layout: Titleslide, TitleandContent, TwoContent
- Graphics support
- Layout Support
- Images

Column Layout with Proc and Graphics
proc means data=sashelp.class min max;
run;

goptions hsize=3in vsize=3in dev=png;
pattern colors="#a78d84";

proc gchart data=sashelp.class;
vbar age / name='pptall0'
ctext="#fba16c"
coutline="red";
run;
quit;

ods powerpoint close;

RTF Output

Proc Format;
  Value Govtfmt -3='Council Manager'
                   0='Commission'
                   3='Mayor Council'
                .N='Not Applicable'
                .='   ?';
  Value Robfmt   1='100 or Less'
                   2='101-200'
                   3='201-300'
                   4='Over 300'
                .N='Not Known'
                .='   ?';
  Value Colfg   1='yellow'
                   2='red'
                   3='blue'
                   4='purple'
                .N='green'
                .='black'
          other='black';
  Value Rowfg   -3='red'
                   0='purple'
                   3='blue'
                .N='green'
                .='black'
          other='black';
run;

data gov;
  Label Citygovt='City Government Form'
       Robgrp='Number of Meetings Scheduled';
  Input Citygovt Robgrp Weight; Missing N;
  Format Citygovt Govtfmt. Robgrp Robfmt.;
  LOOP: OUTPUT; WEIGHT=WEIGHT-1; IF WEIGHT>0 THEN GOTO LOOP;
  DROP WEIGHT;
datalines;
0 1 6
0 3 3
Appendix 2 • Example Programs

ods path (prepend) work.templat(update);
ods noproctitle;
proc template;
  define style white;
    parent=styles.htmlblue;
    style body /
      backgroundcolor=white;
    style systemtitle /
      backgroundcolor=white
      fontsize=6
      fontweight=bold
      fontstyle=italic;
    style systemfooter /
      backgroundcolor=white
      fontsize=2
      fontstyle=italic;
    style proctitle /
      backgroundcolor=white
      color=#6078bf
      fontweight=bold
      fontstyle=italic;
  end;

proc template;
  define style white;
    parent=styles.htmlblue;
    style body /
      backgroundcolor=white;
    style systemtitle /
      backgroundcolor=white
      fontsize=6
      fontweight=bold
      fontstyle=italic;
    style systemfooter /
      backgroundcolor=white
      fontsize=2
      fontstyle=italic;
    style proctitle /
      backgroundcolor=white
      color=#6078bf
      fontweight=bold
      fontstyle=italic;
end;
fontstyle=italic;
end;
define crosstabs Base.Freq.CrossTabFreqs;
notes " Crosstabulation table " ;
style=table { backgroundcolor = #BFCFFF } ;
cell_style = data { backgroundcolor = #FFFFF0 } ;
row_var_style = rowheader { backgroundcolor = #BFCFFF color = rowfg. } ;
col_var_style = header { backgroundcolor = #BFCFFF color = colfg. } ;
row_total_style = data { backgroundcolor = #F0F0F0 } ;
col_total_style = data { backgroundcolor = #F0F0F0 } ;
grand_total_style = datastrong { backgroundcolor = #F0F0F0 } ;
legend_style = header { backgroundcolor = #BFCFFF color = #6078bf fontstyle = italic } ;
rows_header = RowsHeader cols_header = ColsHeader;
label = " Frequency Counts and Percentages " ;
define header TableOf;
text " Table of " _ROW_LABEL_ " by " _COL_LABEL_ / _ROW_LABEL_ ^= '' 
& _COL_LABEL_ ^= '' ;
text " Table of " _ROW_LABEL_ " by " _COL_NAME_ / _ROW_LABEL_ ^= '' ;
text " Table of " _ROW_NAME_ " by " _COL_LABEL_ / _COL_LABEL_ ^= '' ;
text " Table of " _ROW_NAME_ " by " _COL_NAME_ ;
sty le = header { backgroundcolor = #BFCFFF color = #6078bf fontstyle = italic } ;
end ;
define header RowsHeader;
text _ROW_LABEL_ / _ROW_LABEL_ ^= '' ;
text _ROW_NAME_ ;
sty le = header { backgroundcolor = #BFCFFF color = #6078bf fontstyle = italic } ;
space = 0 ;
end ;
define header ColsHeader;
text _COL_LABEL_ / _COL_LABEL_ ^= '' ;
text _COL_NAME_ ;
sty le = header { backgroundcolor = #BFCFFF color = #6078bf fontstyle = italic } ;
space = 1 ;
end ;
define header ControllingFor ;
dynamic StratNum StrataVariableNames StrataVariableLabels ;
text " Controlling for " StrataVariableNames / StratNum > 0 ;
sty le = header ;
end ;
define footer Missing ;
dynamic FMissing ;
text " Frequency Missing = " FMissing - 12.99 / FMissing ^= 0 ;
sty le = header { backgroundcolor = #BFCFFF color = #6078bf fontstyle = italic } ;
space = 1 ;
end ;
define footer NoObs ;
dynamic SampleSize ;
text " Effective Sample Size = 0 " / SampleSize = 0 ;
space = 1 ;
sty le = header ;

define cellvalue Frequency;
  header="";
  label="#Frequency Count#";
  format=BEST7.; data_format_override=on; print=on;
  cellstyle _val_ < 10 as datastrong {color=green},
       _val_ > 40 & _val_ < 50 as datastrong {color=orange},
       _val_ >= 50 as datastrong {color=red};
end;

define cellvalue Expected;
  header="";
  label="#Expected Frequency#";
  format=BEST6. data_format_override=on print=on;
end;

define cellvalue Deviation;
  header="";
  label="#Deviation from Expected Frequency#";
  format=BEST6. data_format_override=on print=on;
end;

define cellvalue CellChiSquare;
  header="";
  label="#Cell Chi-Square#";
  format=BEST6. print=on;
end;

define cellvalue TotalPercent;
  header="";
  label="#Percent of Total Frequency#";
  format=6.2 print=on;
end;

define cellvalue Percent;
  header="";
  label="#Percent of Two-Way Table Frequency#";
  format=6.2 print=on;
end;

define cellvalue RowPercent;
  header="";
  label="#Percent of Row Frequency#";
  format=6.2 print=on;
end;

define cellvalue ColPercent;
  header="";
  label="#Percent of Column Frequency#";
  format=6.2 print=on;
end;

define cellvalue CumColPercent;
  header="";
  label="#Cumulative Percent of Column Frequency#";
  format=6.2 print=on;
end;

cellvalue
  Frequency Expected Deviation
  CellChiSquare TotalPercent Percent
  RowPercent ColPercent CumColPercent;
header TableOf ControllingFor;
footer NoObs Missing;
end;

options nodate;
ods rtf file='MyCrosstabsTable.rtf' style=white;

title "City Government Form by Number of Meetings Scheduled";
ods noproctitle;

proc freq;
  tables citygovt*robgrp / missprint;
run;

ods rtf close;

Creating the $CNTRY Format

proc format;
  value $cntry 'BRZ'='Brazil'
    'CHN'='China'
    'IND'='India'
    'INS'='Indonesia'
    'USA'='United States';
run;

Creating the Charity Data Set

proc format;
  value yrFmt . = " All";
  value $schFmt " " = "All    ";
run;

data Charity;
  input School $ 1-7 Year 9-12 Name $ 14-20 moneyRaised 22-26
    hoursVolunteered 28-29;
  format moneyRaised dollar8.2;
  format hoursVolunteered f3.0;
  format Year yrFmt.;
  format School $schFmt.;
  label School = "Schools";
  label Year = "Years";
  retain yearmin yearmax;
  yearmin=min(yearmin,year);
  yearmax=max(yearmax,year);
  call symput('first_year',put(yearmin,4.));
  call symput('last_year', put(yearmax,4.));
datalines;
Monroe 1992 Allison 31.65 19
Monroe 1992 Barry 23.76 16
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Monroe 1992</td>
<td>Candace</td>
<td>21.11</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Danny</td>
<td>6.89</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Edward</td>
<td>53.76</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Fiona</td>
<td>48.55</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Gert</td>
<td>24.00</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Harold</td>
<td>27.55</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Ima</td>
<td>5.98</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Jack</td>
<td>20.00</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Katie</td>
<td>22.11</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Lisa</td>
<td>18.34</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Tonya</td>
<td>55.16</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Max</td>
<td>26.77</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Ned</td>
<td>28.43</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Monroe 1992</td>
<td>Opal</td>
<td>32.66</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Monroe 1993</td>
<td>Patsy</td>
<td>18.33</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Monroe 1993</td>
<td>Quentin</td>
<td>16.89</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Monroe 1993</td>
<td>Randall</td>
<td>12.98</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Monroe 1993</td>
<td>Sam</td>
<td>15.88</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Monroe 1993</td>
<td>Tyra</td>
<td>21.88</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Monroe 1993</td>
<td>Myrtle</td>
<td>47.33</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Monroe 1993</td>
<td>Frank</td>
<td>41.11</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Monroe 1993</td>
<td>Cameron</td>
<td>65.44</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Monroe 1993</td>
<td>Vern</td>
<td>17.89</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Monroe 1993</td>
<td>Wendell</td>
<td>23.00</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Monroe 1993</td>
<td>Bob</td>
<td>26.88</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Monroe 1993</td>
<td>Leah</td>
<td>28.99</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Becky</td>
<td>30.33</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Sally</td>
<td>35.75</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Edgar</td>
<td>27.11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Dawson</td>
<td>17.24</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Lou</td>
<td>5.12</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Damien</td>
<td>18.74</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Mona</td>
<td>27.43</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Della</td>
<td>56.78</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Monique</td>
<td>29.88</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Carl</td>
<td>31.12</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Reba</td>
<td>35.16</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Dax</td>
<td>27.65</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Gary</td>
<td>23.11</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Suzie</td>
<td>26.65</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Benito</td>
<td>47.44</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Thomas</td>
<td>21.99</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Annie</td>
<td>24.99</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Paul</td>
<td>27.98</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Alex</td>
<td>24.00</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Lauren</td>
<td>15.00</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Julia</td>
<td>12.98</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Keith</td>
<td>11.89</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Jackie</td>
<td>26.88</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Pablo</td>
<td>13.98</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>L.T.</td>
<td>56.87</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Willard</td>
<td>78.65</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Kathy</td>
<td>32.88</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Monroe 1994</td>
<td>Abby</td>
<td>35.88</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Kennedy 1992</td>
<td>Arturo</td>
<td>34.98</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Kennedy 1992</td>
<td>Grace</td>
<td>27.55</td>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>
Kennedy 1992 Winston 23.88 22
Kennedy 1992 Vince 12.88 21
Kennedy 1992 Claude 15.62 5
Kennedy 1992 Mary 28.99 34
Kennedy 1992 Abner 25.89 22
Kennedy 1992 Jay 35.89 35
Kennedy 1992 Alicia 28.77 26
Kennedy 1992 Freddy 29.00 27
Kennedy 1992 Eloise 31.67 25
Kennedy 1992 Jenny 43.89 22
Kennedy 1992 Thelma 52.63 21
Kennedy 1992 Tina 19.67 21
Kennedy 1992 Eric 24.89 12
Kennedy 1993 Bubba 37.88 12
Kennedy 1993 G.L. 25.89 21
Kennedy 1993 Bert 28.89 21
Kennedy 1993 Clay 26.44 21
Kennedy 1993 Leann 27.17 17
Kennedy 1993 Georgia 38.90 11
Kennedy 1993 Bill 42.23 25
Kennedy 1993 Holly 18.67 27
Kennedy 1993 Benny 19.09 25
Kennedy 1993 Cammie 28.77 28
Kennedy 1993 Amy 27.08 31
Kennedy 1993 Doris 22.22 24
Kennedy 1993 Robbie 19.80 24
Kennedy 1993 Ted 27.07 25
Kennedy 1993 Sarah 24.44 12
Kennedy 1993 Megan 28.89 11
Kennedy 1993 Jeff 31.11 12
Kennedy 1993 Taz 30.55 11
Kennedy 1993 George 27.56 11
Kennedy 1993 Heather 38.67 15
Kennedy 1994 Nancy 29.90 26
Kennedy 1994 Rusty 30.55 28
Kennedy 1994 Mimi 37.67 22
Kennedy 1994 J.C. 23.33 27
Kennedy 1994 Clark 27.90 25
Kennedy 1994 Rudy 27.78 23
Kennedy 1994 Samuel 34.44 18
Kennedy 1994 Forrest 28.89 26
Kennedy 1994 Luther 72.22 24
Kennedy 1994 Trey 6.78 18
Kennedy 1994 Albert 23.33 19
Kennedy 1994 Che-Min 26.66 33
Kennedy 1994 Preston 32.22 23
Kennedy 1994 Larry 40.00 26
Kennedy 1994 Anton 35.99 28
Kennedy 1994 Sid 27.45 25
Kennedy 1994 Will 28.88 21
Kennedy 1994 Morty 34.44 25
;
run;
Creating the DIVFMT. and USETYPE. Formats

```sas
proc format;
  value divfmt 1='New England'
                    2='Middle Atlantic'
                    3='Mountain'
                    4='Pacific';
  value usetype 1='Residential Customers'
                    2='Business Customers';
run;
```

Creating the DistrData Data Set

```sas
data distrdata;
  drop n;
  label Normal_x='Normal Random Variable'
           Exponential_x='Exponential Random Variable';
  do n=1 to 100;
    Normal_x=10*rannor(53124)+50;
    Exponential_x=ranexp(18746363);
    output;
  end;
run;
```

Creating the Univ ODS Document

```sas
ods document name=univ;

title '100 Obs Sampled from a Normal Distribution';
proc univariate data=distrdata noprint;
  var Normal_x;

    histogram Normal_x /normal(noprint) cbarline=grey name='normal';
run;

title '100 Obs Sampled from an Exponential Distribution';
proc univariate data=distrdata noprint;
  var Exponential_x;

    histogram /exp(fill l=3) cfill=yellow midpoints=.05 to 5.55 by .25
             name='exp';
run;

ods document close;
title;
```
### Creating the Employee_Data Data Set

```sas
options source pagesize=60 linesize=80 nodate;

data employee_data;
  input IdNumber $ 1-4 LastName $ 9-19 FirstName $ 20-29
          City $ 30-42 State $ 43-44 /
          Gender $ 1 JobCode $ 9-11 Salary 20-29 @30 Birth date9.
          @43 Hired date9. HomePhone $ 54-65;
  format birth hired date9.;

datalines;
1919 Adams Gerald Stamford CT
1653 Alexander Susan Bridgeport CT
1400 Apple Troy New York NY
1350 Arthur Barbara New York NY
1401 Avery Jerry Paterson NJ
1499 Barefoot Joseph Princeton NJ
1101 Baucom Walter New York NY
1333 Blair Justin Stamford CT
1402 Blalock Ralph New York NY
1428 Brady Christine Stamford CT
1479 Bostic Marie New York NY
1403 Bowden Earl Bridgeport CT
1739 Boyce Jonathan New York NY
1658 Bradley Jeremy New York NY
1574 Cahill Marshall New York NY
1404 Carter Donald New York NY
```

The above code creates a SAS data set named `employee_data` with variables for employee identification number, last name, first name, city, state, gender, job code, salary, birth date, hired date, and home phone number.
<table>
<thead>
<tr>
<th>#</th>
<th>First Name</th>
<th>Last Name</th>
<th>Gender</th>
<th>Date of Birth</th>
<th>Social Security Number</th>
<th>Telephone</th>
<th>City</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1437</td>
<td>Carter</td>
<td>Dorothy</td>
<td>M</td>
<td>27 FEB 41</td>
<td>PT2 91376</td>
<td>718/384-2946</td>
<td>Bridgeport</td>
<td>CT</td>
</tr>
<tr>
<td>1639</td>
<td>Carter</td>
<td>Karen</td>
<td>F</td>
<td>31 JAN 72</td>
<td>A3 40260</td>
<td>203/781-8839</td>
<td>Stamford</td>
<td>CT</td>
</tr>
<tr>
<td>1269</td>
<td>Caston</td>
<td>Franklin</td>
<td>M</td>
<td>29 JUN 45</td>
<td>A3 39675</td>
<td>212/588-5634</td>
<td>Stamford</td>
<td>CT</td>
</tr>
<tr>
<td>1065</td>
<td>Chapman</td>
<td>Neil</td>
<td>M</td>
<td>10 JAN 75</td>
<td>ME2 35090</td>
<td>718/384-5618</td>
<td>New York</td>
<td>NY</td>
</tr>
<tr>
<td>1876</td>
<td>Chin</td>
<td>Jack</td>
<td>M</td>
<td>29 JAN 32</td>
<td>TA3 18056</td>
<td>201/732-2323</td>
<td>New York</td>
<td>NY</td>
</tr>
<tr>
<td>1037</td>
<td>Cook</td>
<td>Brenda</td>
<td>F</td>
<td>16 SEP 80</td>
<td>A3 28558</td>
<td>203/781-8868</td>
<td>Stamford</td>
<td>CT</td>
</tr>
<tr>
<td>1129</td>
<td>Delgado</td>
<td>Maria</td>
<td>F</td>
<td>24 DEC 76</td>
<td>TA2 33462</td>
<td>203/781-1528</td>
<td>Stamford</td>
<td>CT</td>
</tr>
<tr>
<td>1983</td>
<td>Dunlap</td>
<td>Donna</td>
<td>F</td>
<td>21 NOV 75</td>
<td>TA3 39223</td>
<td>203/781-2229</td>
<td>New York</td>
<td>NY</td>
</tr>
<tr>
<td>1438</td>
<td>Donaldson</td>
<td>Karen</td>
<td>M</td>
<td>18 MAR 53</td>
<td>TA2 39923</td>
<td>203/781-2229</td>
<td>Stamford</td>
<td>CT</td>
</tr>
<tr>
<td>1256</td>
<td>Edgerton</td>
<td>Joshua</td>
<td>M</td>
<td>16 JUL 78</td>
<td>TA3 39771</td>
<td>212/588-5613</td>
<td>New York</td>
<td>NY</td>
</tr>
<tr>
<td>1905</td>
<td>Fernandez</td>
<td>Katrina</td>
<td>M</td>
<td>16 AUG 80</td>
<td>NA2 51081</td>
<td>212/588-1239</td>
<td>Bridgeport</td>
<td>CT</td>
</tr>
<tr>
<td>1124</td>
<td>Fields</td>
<td>Diana</td>
<td>F</td>
<td>19 OCT 69</td>
<td>ME2 35185</td>
<td>203/675-2962</td>
<td>White Plains</td>
<td>NY</td>
</tr>
<tr>
<td>1422</td>
<td>Fletcher</td>
<td>Marie</td>
<td>F</td>
<td>04 OCT 78</td>
<td>PA1 23177</td>
<td>914/455-2998</td>
<td>Princeton</td>
<td>NJ</td>
</tr>
<tr>
<td>1616</td>
<td>Flowers</td>
<td>Annette</td>
<td>F</td>
<td>09 APR 79</td>
<td>PA1 22454</td>
<td>212/812-0902</td>
<td>New York</td>
<td>NY</td>
</tr>
<tr>
<td>1389</td>
<td>Gordon</td>
<td>Levi</td>
<td>M</td>
<td>20 APR 79</td>
<td>A3 34137</td>
<td>718/384-3329</td>
<td>New York</td>
<td>NY</td>
</tr>
<tr>
<td>1905</td>
<td>Graham</td>
<td>Alvin</td>
<td>M</td>
<td>20 FEB 75</td>
<td>ME2 35185</td>
<td>203/675-6363</td>
<td>Bridgeport</td>
<td>CT</td>
</tr>
<tr>
<td>1407</td>
<td>Grant</td>
<td>Daniel</td>
<td>M</td>
<td>21 MAR 78</td>
<td>PT1 68096</td>
<td>914/468-1616</td>
<td>Mt. Vernon</td>
<td>NY</td>
</tr>
<tr>
<td>1114</td>
<td>Green</td>
<td>Janice</td>
<td>F</td>
<td>26 MAR 57</td>
<td>PT1 68096</td>
<td>914/468-1616</td>
<td>New York</td>
<td>NY</td>
</tr>
</tbody>
</table>

**Appendix 2 • Example Programs**
Creating the Energy Data Set

data energy;
  length State $2;
  input Region Division state $ Type Expenditures @@;
  datalines;
1 1 ME 1 708 1 1 ME 2 379 1 1 NH 1 597 1 1 NH 2 301
1 1 VT 1 353 1 1 VT 2 188 1 1 MA 1 3264 1 1 MA 2 2498
1 1 RI 1 531 1 1 RI 2 358 1 1 CT 1 2024 1 1 CT 2 1405
1 2 NY 1 8786 1 2 NY 2 7825 1 2 NJ 1 4115 1 2 NJ 2 3558
1 2 PA 1 6478 1 2 PA 2 3695 4 3 MT 1 322 4 3 MT 2 232
4 3 ID 1 392 4 3 ID 2 298 4 3 WY 1 194 4 3 WY 2 184
4 3 CO 1 1215 4 3 CO 2 1173 4 3 NM 1 545 4 3 NM 2 578
4 3 AZ 1 1694 4 3 AZ 2 1448 4 3 UT 1 621 4 3 UT 2 438
4 3 NV 1 493 4 3 NV 2 378 4 3 WA 1 1680 4 3 WA 2 1122
4 4 OR 1 1014 4 4 OR 2 756 4 4 CA 1 10643 4 4 CA 2 10114
4 4 AK 1 349 4 4 AK 2 329 4 4 HI 1 273 4 4 HI 2 298
; run;

Creating the Exprev Data Set

data exprev;
  input Country $ 1-24 Emp_ID $ 25-32 Order_Date $ Ship_Date $ Sale_Type $ & Quantity Price Cost;
  datalines;
Antarctica 99999999 1/1/05 1/7/05 Internet 2 92.60 20.70
Puerto Rico 99999999 1/1/05 1/5/05 Catalog 14 51.20 12.10
Virgin Islands (U.S.) 99999999 1/1/05 1/4/05 In Store 25 31.10 15.65
Aruba 99999999 1/1/05 1/4/05 Catalog 30 123.70 59.00
Bahamas 99999999 1/1/05 1/4/05 Catalog 8 113.40 28.45
Bermuda 99999999 1/1/05 1/4/05 Catalog 7 41.00 9.25
Belize 120458 1/2/05 1/2/05 In Store 20 146.40 36.70
British Virgin Islands 99999999 1/2/05 1/5/05 Catalog 11 40.20 20.20
Canada 99999999 1/2/05 1/5/05 Catalog 100 11.80 5.00
Cayman Islands 120458 1/2/05 1/2/05 In Store 20 71.00 32.30
Costa Rica 99999999 1/2/05 1/6/05 Internet 31 53.00 26.60
Cuba 121044 1/2/05 1/2/05 Internet 12 42.40 19.35
Dominican Republic 121040 1/2/05 1/2/05 Internet 13 48.00 23.95
El Salvador 99999999 1/2/05 1/6/05 Catalog 21 266.40 66.70
Guatemala 120931 1/2/05 1/2/05 In Store 13 144.40 65.70
Haiti 121059 1/2/05 1/2/05 Internet 5 47.90 23.45
Honduras 120455 1/2/05 1/2/05 Internet 20 66.40 30.25
Jamaica 99999999 1/2/05 1/4/05 In Store 23 169.80 38.70
Mexico 120127 1/2/05 1/2/05 In Store 30 211.80 33.65
Montserrat 120127 1/2/05 1/2/05 In Store 19 184.20 36.90
<table>
<thead>
<tr>
<th>Country</th>
<th>Code</th>
<th>Date</th>
<th>Date</th>
<th>Method</th>
<th>Count</th>
<th>Cost</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nicaragua</td>
<td>120932</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Internet</td>
<td>16</td>
<td>122.00</td>
<td>28.75</td>
</tr>
<tr>
<td>Panama</td>
<td>99999999</td>
<td>1/2/05</td>
<td>1/6/05</td>
<td>Internet</td>
<td>20</td>
<td>88.20</td>
<td>38.40</td>
</tr>
<tr>
<td>Saint Kitts/Nevis</td>
<td>99999999</td>
<td>1/2/05</td>
<td>1/6/05</td>
<td>Internet</td>
<td>20</td>
<td>41.40</td>
<td>18.00</td>
</tr>
<tr>
<td>St. Helena</td>
<td>120360</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Internet</td>
<td>19</td>
<td>94.70</td>
<td>47.45</td>
</tr>
<tr>
<td>St. Pierre/Miquelon</td>
<td>120842</td>
<td>1/2/05</td>
<td>1/16/05</td>
<td>Internet</td>
<td>16</td>
<td>103.80</td>
<td>47.25</td>
</tr>
<tr>
<td>Turks/Caicos Islands</td>
<td>120372</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Internet</td>
<td>10</td>
<td>57.70</td>
<td>28.95</td>
</tr>
<tr>
<td>United States</td>
<td>120372</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Internet</td>
<td>20</td>
<td>88.20</td>
<td>38.40</td>
</tr>
<tr>
<td>Anguilla</td>
<td>99999999</td>
<td>1/2/05</td>
<td>1/6/05</td>
<td>In Store</td>
<td>15</td>
<td>233.50</td>
<td>22.25</td>
</tr>
<tr>
<td>Antigua/Barbuda</td>
<td>120458</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>In Store</td>
<td>31</td>
<td>99.60</td>
<td>45.35</td>
</tr>
<tr>
<td>Argentina</td>
<td>99999999</td>
<td>1/2/05</td>
<td>1/6/05</td>
<td>In Store</td>
<td>42</td>
<td>408.80</td>
<td>87.15</td>
</tr>
<tr>
<td>Barbados</td>
<td>99999999</td>
<td>1/2/05</td>
<td>1/6/05</td>
<td>In Store</td>
<td>26</td>
<td>94.80</td>
<td>42.60</td>
</tr>
<tr>
<td>Bolivia</td>
<td>120127</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>In Store</td>
<td>26</td>
<td>66.00</td>
<td>16.60</td>
</tr>
<tr>
<td>Brazil</td>
<td>120127</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Catalog</td>
<td>12</td>
<td>73.40</td>
<td>18.45</td>
</tr>
<tr>
<td>Chile</td>
<td>120447</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>In Store</td>
<td>20</td>
<td>19.10</td>
<td>8.75</td>
</tr>
<tr>
<td>Colombia</td>
<td>121059</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Internet</td>
<td>28</td>
<td>361.40</td>
<td>90.45</td>
</tr>
<tr>
<td>Dominica</td>
<td>121043</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Internet</td>
<td>35</td>
<td>121.30</td>
<td>57.80</td>
</tr>
<tr>
<td>Ecuador</td>
<td>121042</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>In Store</td>
<td>11</td>
<td>100.90</td>
<td>50.55</td>
</tr>
<tr>
<td>Falkland Islands</td>
<td>120932</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>In Store</td>
<td>15</td>
<td>61.40</td>
<td>30.80</td>
</tr>
<tr>
<td>French Guiana</td>
<td>120935</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Catalog</td>
<td>15</td>
<td>96.40</td>
<td>43.85</td>
</tr>
<tr>
<td>Grenada</td>
<td>120931</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Catalog</td>
<td>19</td>
<td>56.30</td>
<td>25.05</td>
</tr>
<tr>
<td>Guadeloupe</td>
<td>120445</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Internet</td>
<td>21</td>
<td>231.60</td>
<td>48.70</td>
</tr>
<tr>
<td>Guyana</td>
<td>120455</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>In Store</td>
<td>25</td>
<td>132.80</td>
<td>30.25</td>
</tr>
<tr>
<td>Martinique</td>
<td>120841</td>
<td>1/2/05</td>
<td>1/3/05</td>
<td>In Store</td>
<td>16</td>
<td>56.30</td>
<td>31.05</td>
</tr>
<tr>
<td>Netherlands Antilles</td>
<td>99999999</td>
<td>1/2/05</td>
<td>1/6/05</td>
<td>In Store</td>
<td>31</td>
<td>41.80</td>
<td>19.45</td>
</tr>
<tr>
<td>Paraguay</td>
<td>120603</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Catalog</td>
<td>17</td>
<td>117.60</td>
<td>58.90</td>
</tr>
<tr>
<td>Peru</td>
<td>120845</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Catalog</td>
<td>12</td>
<td>93.80</td>
<td>41.75</td>
</tr>
<tr>
<td>St. Lucia</td>
<td>120845</td>
<td>1/2/05</td>
<td>1/2/05</td>
<td>Internet</td>
<td>19</td>
<td>64.30</td>
<td>28.65</td>
</tr>
<tr>
<td>Suriname</td>
<td>120538</td>
<td>1/3/05</td>
<td>1/3/05</td>
<td>Internet</td>
<td>22</td>
<td>110.80</td>
<td>29.35</td>
</tr>
</tbody>
</table>

Creating the Gov Data Set

```plaintext
data gov;
  label Citygovt='City Government Form';
  label Robgrp='Number of Citizens Robbed';
  input Citygovt Robgrp Weight;
  format Citygovt Govtfmt. Robgrp Robfmt.;
  loop: output;
  weight=weight-1;
  if weight>0 then goto loop;
  drop weight;
datalines;
0 1 6
0 3 3
0 2 7
0 4 5
N N 10
-3 1 47
-3 3 49
-3 2 63
-3 4 52
. 2 1
3 1 31
3 2 37
run;
```

Appendix 2 • Example Programs
Creating the Grain_Production Data Set

data grain_production;
  length Country $ 3 Type $ 5;
  input Year country $ type $ Kilotons;
datalines;
1995 BRZ  Wheat  1516
1995 BRZ  Rice   11236
1995 BRZ  Corn   36276
1995 CHN  Wheat  102207
1995 CHN  Rice   185226
1995 CHN  Corn   112331
1995 IND  Wheat  63007
1995 IND  Rice   122372
1995 IND  Corn   9800
1995 INS  Wheat  .
1995 INS  Rice   49860
1995 INS  Corn   8223
1995 USA  Wheat  59494
1995 USA  Rice   7888
1995 USA  Corn   187300
1996 BRZ  Wheat  3302
1996 BRZ  Rice   10035
1996 BRZ  Corn   31975
1996 CHN  Wheat  109000
1996 CHN  Rice   190100
1996 CHN  Corn   119350
1996 IND  Wheat  62620
1996 IND  Rice   120012
1996 IND  Corn   8660
1996 INS  Wheat  .
1996 INS  Rice   51165
1996 INS  Corn   8925
1996 USA  Wheat  62099
1996 USA  Rice   7771
1996 USA  Corn   236064
;
run;

Creating the One Data Set

data one;
  input year import doprod stock consum;
datalines;
49 15.9 149.3 4.2 108.1
Creating the Iron Data Set

The data set Iron contains data from Draper and Smith.

data iron;
  input Fe Loss @@;
datalines;
0.01 127.6   0.48 124.0   0.71 110.8   0.95 103.9
1.19 101.5   0.01 130.1   0.48 122.0   1.44  92.3
0.71 113.1   1.96  83.7   0.01 128.0   1.44  91.4
1.96  86.2
;
run;


Creating the Model Data Set

data model;
  input year 1-2 a 3-9 .3 b 10-17 .3 r4 18-24 .3 r8 25-31 .3
c 32-38 .3 d 39-45 .3 e 46-51 .3 r23 52-58 .3
r24 59-64 .3 r29 65-70 .3 r33 71-77 .3
;
datalines;
60 994534 53552371656049 9362944261250 8921423631971140299106045 8780 335066
611253576 5580643177015110671424650930 9933453874651217360151507 36871 49192
621318885 621448018932921075688469573610686654502881317293178014 66671 566079
631507969 6661251210462615330885117013116736951628215791487979106485 -4568
641811051 731945021737841454106554095914677245822921945534206255145948 -10940
652532026 816707123363201962785640926221155676314091906268218759195733 -145568
661845213 88903932680634222339564930721533118605504173294828832275400 132143

Creating the Neuralgia Data Set

Data Neuralgia;
  length Discomfort $ 2;
  input Treatment $ Gender $ Age Duration Discomfort $ @@;
  label Treatment='Treatment Regimen'
     Gender='Gender of Patient'
     Age='Age of Patient'
     Discomfort='Amount of Discomfort Experienced';
datalines;
A M 71 17 P A F 63 27 PF A F 69 18 P
P F 68 1 PF B M 74 16 PF P F 67 30 PF
P M 66 26 P B F 67 28 PF B F 77 16 PF
A F 71 12 PF B P 72 50 PF B F 76 9 P
B F 66 12 PF A M 62 42 PF P F 64 1 P
A F 64 17 PF P M 74 4 PF A F 72 25 PF
P M 70 1 P B M 66 19 PF B M 59 29 PF
A F 64 30 PF A M 70 28 PF A M 69 1 PF
B F 78 1 PF P M 83 1 P B F 69 42 PF
B M 75 30 P P M 77 29 P P P 79 20 P
A M 70 12 PF A P 69 12 PF B F 65 14 PF
B M 70 1 PF B M 67 23 PF A M 76 25 P
P M 78 12 P B M 77 1 P B F 69 24 PF
P M 66 4 P P F 65 29 PF P M 60 26 P
A M 78 15 P B M 75 21 P A F 67 11 PF
P F 72 27 PF P P P 70 13 P A M 75 6 P
B F 65 7 PF P P 68 27 P P M 68 11 P
P M 67 17 P B M 70 22 PF A M 65 15 PF
P F 67 1 P A M 67 10 PF P F 72 11 P
A F 74 1 PF B M 80 21 P A F 69 3 PF
run;
Creating the Plants Data Set

data plants;
    input type $ @;
    do block=1 to 3;
        input stemleng @;
        output;
    end;
    datalines;
    clarion  32.7 32.3 31.5
    clinton  32.1 29.7 29.1
    knox     35.7 35.9 33.1
    o'neill  36.0 34.2 31.2
    compost  31.8 28.0 29.2
    wabash   38.2 37.8 31.9
    webster  32.5 31.1 29.7
    ;
run;

Creating the Plant_Stats Data Set

data plant_stats;
    do month = 1 to 12;
        age  = 2 + 0.3*rannor(345467);
        age2 = 3 + 0.3*rannor(345467);
        age3 = 4 + 0.4*rannor(345467);
        output;
    end;
run;

Creating the StatePop Data Set

data statepop;
    input State $ CityPop_80 CityPop_90
        NonCityPop_80 NonCityPop_90 Region;
    format region 1.;
    label citypop_80= '1980 metropolitan pop in millions'
        noncitypop_80='1980 nonmetropolitan pop in millions'
        citypop_90='1990 metropolitan pop in millions'
        noncitypop_90='1990 nonmetropolitan pop in million'
        region='Geographic region';
    datalines;
    ME    .405    .443    .721    .785  1
    NH    .535    .659    .386    .450  1
    VT    .133    .152    .378    .411  1

Appendix 2 • Example Programs
Creating the StatePop Data Set

```r
MA  5.530  5.788  .207  .229  1
RI  0.886  0.938  0.061  0.065  1
CT  2.982  3.148  0.126  0.140  1
NY  16.144 16.515  1.414  1.475  1
NJ  7.365  7.730  0A     0A   1
PA 10.067 10.083  1.798  1.799  1
DE  4.96   5.53  0.098  0.113  2
MD  3.920  4.439  0.297  0.343  2
DC  3.966  4.773  1.381  1.414  2
VA  3.749  4.376  1.155  1.045  2
SC  2.114  2.423  1.006  1.064  2
GA  3.507  4.352  1.956  2.127  2
FL  9.039 12.023  0.708  0.915  2
KY  1.735  1.780  1.925  1.906  2
TN  3.045  3.298  1.546  1.579  2
AL  2.560  2.710  1.334  1.331  2
MS  0.716  0.776  1.805  1.798  2
AR  0.963  1.040  1.323  1.311  2
LA  3.125  3.160  1.082  1.060  2
OK  1.724  1.870  1.301  1.276  2
TX 11.539 14.166  2.686  2.821  2
OH  8.791  8.826  2.007  2.021  3
IN  3.885  3.962  1.605  1.582  3
IL  9.461  9.574  1.967  1.857  3
MI  7.719  7.698  1.543  1.598  3
WI  3.176  3.331  1.530  1.561  3
MN  2.674  3.011  1.402  1.364  3
IA  1.198  1.200  1.716  1.577  3
MO  3.314  3.491  1.603  1.626  3
ND  0.234  0.257  0.418  0.381  3
SD  0.194  0.221  0.497  0.475  3
NE  0.728  0.787  0.842  0.791  3
KS  1.184  1.333  1.180  1.145  3
MT  0.189  0.191  0.598  0.608  4
ID  0.257  0.296  0.687  0.711  4
WY  0.141  0.134  0.329  0.319  4
CO  2.326  2.686  0.563  0.608  4
NM  0.675  0.842  0.628  0.673  4
AZ  2.264  3.106  0.453  0.559  4
UT  1.128  1.336  0.333  0.387  4
NV  0.666  1.014  0.135  0.183  4
WA  3.366  4.036  0.776  0.830  4
OR  1.799  1.985  0.834  0.858  4
CA 22.907 28.799  0.760  0.961  4
AK  0.174  0.226  0.227  0.324  4
HI  0.763  0.836  0.202  0.272  4

; run;
```
Creating the Stats and Stats2 Data Sets

``` SAS
data Stats;
  input Price Quantity City $;
  datalines;
  3750  150 Brazil
  5000  200 Canada
  10250 410 France
;  
data Stats2;
  input Price Quantity City $;
  datalines;
  3750  150 Brazil
  5000  200 Canada
  10250 410 France
;  
run;
```

Creating the Table1 Table Template

``` SAS
proc template;
  define table table1;
    mvar sysdate9;
    dynamic colhd;
    classlevels=on;

    define column char_var;
      generic=on;
      blank_dups=on;
      header=colhd;
      style=cellcontents;
    end;

    define column num_var;
      generic=on;
      header=colhd;
      style=cellcontents;
    end;

    define footer table_footer;
      text 'Prepared on ' sysdate9;
    end;

  end;
run;
```
Overview

The programs in this section show the PROC TEMPLATE steps that were used in “Understanding Styles, Style Elements, and Style Attributes” in *SAS Output Delivery System: Procedures Guide* to illustrate inheritance in style templates. These programs also show the SAS code that uses the style definitions.

Using the FROM option

This program generates the HTML output in the section “Using the FROM Option” in *SAS Output Delivery System: Procedures Guide*.

- This version of the code uses the FROM option in the STYLE statement to create the Colors style element in the Concepts.Style2 style template.

```sas
ods path sashelp.tmplmst(read) sasuser.templat(update);
title;
options nodate pageno=1 linesize=72 pagesize=60;
data test;
  input country $ 1-13 grain $ 15-18 kilotons;
datalines;
Brazil     Rice   10035
China       Rice   190100
India       Rice   120012
Indonesia   Rice   51165
United States Rice   7771
;
proc template;
  define table mytable;
    column x y z w;
    define x;
      style=celldatasimple;
      dataname=country;
      header='Country';
    end;
    define y;
      style=celldataemphasis;
      dataname=grain;
      header='Grain';
    end;
    define z;
      style=celldatalarge;
      dataname=kilotons;
      header='Kilotons';
    end;
    define w;
      style=celldatasmall;
      dataname=kilotons;
      header='Kilotons';
  end;
```
end;
end;
run;

proc template;
/* to ensure a fresh start with the styles */
delete concepts.style1;
delete concepts.style2;
run;

proc template;
define style concepts.style1;
style colors /
'default'=white
'fancy'=very light vivid blue
'medium'=red ;
style celldatasimple /
fontfamily=arial
backgroundcolor=colors('fancy')
color=colors('default');
style celldataemphasis from celldatasimple /
color=colors('medium')
fontstyle=italic;
style celldatalarge from celldataemphasis /
fontweight=bold
fontsize=3;
end;
run;

proc template;
define style concepts.style2;
parent=concepts.style1;
style colors from colors/
'dark'=dark blue;
style celldataemphasis from celldataemphasis /
bgcolor=white;
style celldatasmall from celldatalarge /
fontsize=5
color=colors('dark')
backgroundcolor=colors('medium');
end;
run;
ods html body='display1-body.htm'
style=concepts.style2;
data _null_; 
set test;
file print ods=(template='mytable');
put _ods_; 
run;
ods html close;

• This version of the code does not use the FROM option in the STYLE statement to create the Colors style element in the Concepts.Style2 style template.

ods path sashelp.tmplmst(read) sasuser.templat(update);
title;
options nodate pageno=1 linesize=72 pagesize=60;
data test;
   input country $ 1-13 grain $ 15-18 kilotons;
   datalines;
Brazil    Rice   10035
China     Rice   190100
India     Rice   120012
Indonesia Rice   51165
United States Rice 7771
;
proc template;
   define table mytable;
      column x y z w;
      define x;
          style=celldatasimple;
          dataname=country;
          header=’Country’;
      end;
      define y;
          style=celldataemphasis;
          dataname=grain;
          header=’Grain’;
      end;
      define z;
          style=celldatalarge;
          dataname=kilotons;
          header=’Kilotons’;
      end;
      define w;
          style=celldatasmall;
          dataname=kilotons;
          header=’Kilotons’;
      end;
   end;
run;

proc template;
   /* to ensure a fresh start with the styles */
   delete concepts.style1;
   delete concepts.style2;
run;

proc template;
   define style concepts.style1;
   style colors /
     'default’=white
     'fancy’=very light vivid blue
     'medium’=red
     ;
   style celldatasimple /
     fontfamily=arial
     backgroundcolor=colors(’fancy’)
     color=colors(’default’);
   style celldataemphasis from celldatasimple /
     color=colors(’medium’)
     fontstyle=italic;
   style celldatalarge from celldataemphasis /
Inheritance Compatibility Across SAS Versions

This program generates the HTML output in the section “Inheritance Compatibility across Versions” in SAS Output Delivery System: Procedures Guide.

- This version of the code uses SAS 9.2 names for style attributes supplied by SAS.

```sas
ods path sashelp.tmplmst(read) sasuser.templat(update);
title;
options nodate pageno=1 linesize=72 pagesize=60;
data test;
  input country $ 1-13 grain $ 15-18 kilotons;
datalines;
Brazil        Rice   10035
China         Rice   190100
India         Rice   120012
Indonesia     Rice   51165
United States Rice  7771
;
proc template;
define table mytable;
column x y z w;
define x;
  style=celldatasmall;
datename=country;
header='Country';
end;
define y;
```

Programs That Illustrate Inheritance

```
style=celldataemphasis;
datename=grain;
header='Grain';
end;
define z;
  style=celldatalarge;
datename=kilotons;
header='Kilotons';
end;
define w;
  style=celldatasmall;
datename=kilotons;
header='Kilotons';
end;
end;
run;

proc template;
  /* to ensure a fresh start with the styles */
delete concepts.style1;
delete concepts.style2;
run;

proc template;
  define style concepts.style1;
    parent=concepts.style1;
    style celldatasimple /
      fontfamily=arial
      backgroundcolor=very light vivid blue
      color=white;
    style celldataemphasis from celldatasimple /
      color=red
      fontstyle=italic;
    style celldatalarge from celldataemphasis /
      fontweight=bold
      fontsize=5;
  end;
run;

proc template;
  define style concepts.style2;
    parent=concepts.style1;
    style celldataemphasis from celldataemphasis /
      backgroundcolor=yellow;
    style celldatasmall from celldatalarge /
      fontsize=2;
  end;
ods html body='display1-body.htm'
  style=concepts.style2;
data _null_; set test;
  file print ods=(template='mytable');
  put _ods_; run;
ods html close;
```
This version of the code uses SAS 9.1 names for style attributes that are supplied by SAS.

```sas
ods path sashelp.tmplmst(read) sasuser.templat(update);
title;
options nodate pageno=1 linesize=72 pagesize=60;
data test;
  input country $ 1-13 grain $ 15-18 kilotons;
datalines;
Brazil  Rice  10035
China   Rice  190100
India   Rice  120012
Indonesia Rice  51165
United States Rice  7771
;
proc template;
  define table mytable;
    column x y z w;
    define x;
      style=celldatasimple;
      dataname=country;
      header='Country';
    end;
    define y;
      style=celldataemphasis;
      dataname=grain;
      header='Grain';
    end;
    define z;
      style=celldatalarge;
      dataname=kilotons;
      header='Kilotons';
    end;
    define w;
      style=celldatasmall;
      dataname=kilotons;
      header='Kilotons';
    end;
  end;
run;
proc template;
  /* to ensure a fresh start with the styles */
  delete concepts.style1;
  delete concepts.style2;
run;
proc template;
  define style concepts.style1;
    style celldatasimple /
      fontface=arial
      background=very light vivid blue
      foreground=white;
    style celldataemphasis from celldatasimple /
      foreground=red
  end;
```
Programs That Illustrate Inheritance

```
fontstyle=italic;
style celldatalarge from celldataemphasis /
  fontweight=bold
  fontsize=5;
end;
run;

proc template;
  define style concepts.style2;
    parent=concepts.style1;
    style celldataemphasis from celldataemphasis /
      background=yellow;
    style celldatasmall from celldatalarge /
      fontsize=2;
  end;
ods html body='display1-body.htm'
  style=concepts.style2;
data _null_; run;
ods html close;
```
Appendix 3

ODS and the HTML Destination

---

**HTML Links and References Produced by the HTML Destination**

- What Are Links and References? ............................................. 1101
- Implementing HTML Links and References ............................. 1101
- How ODS Constructs Links and References ............................. 1103

**Files Produced by the HTML Destination**

- Overview .............................................................................. 1106
- The Body File ....................................................................... 1106
- The Contents File .................................................................. 1109
- The Page File ........................................................................ 1109
- The Frame File ....................................................................... 1109

---

**HTML Links and References Produced by the HTML Destination**

**What Are Links and References?**

An HTML link is a place in a document that enables you to jump to another specific place in the same document or in another document. A browser typically highlights the text that is between the tags that begin and end the link. When you click on the highlighted text, the browser displays the text at the link target. The browser might then display the contents of the target in the active window, or it might open another browser window that displays the contents of the target.

An HTML reference names a file for the browser to display. When a browser reads a reference, it displays the referenced file as if it were part of the file that it is displaying. You cannot tell by looking at the browser's display that some of the material is in the file that you are actually viewing and that some is referenced.

When you use ODS, the software automatically creates the links and references that you need. However, you can customize these links to some extent. If you want to do so, then you need to understand how HTML implements links and references.

**Implementing HTML Links and References**

*Note:* This simplified discussion of HTML links and references is designed to provide information that will help you understand what ODS does when it builds links and
Each link in HTML is implemented with a combination of two sets of `<A>` (anchor) tags. The anchor tag that is the starting point of the link has an `HREF` attribute that identifies the anchor tag to link to. The other anchor tag is the target of the link, and has a `NAME` attribute. This `NAME` attribute is what the `HREF` attribute in the first anchor tag points to. The value of each `NAME` attribute in a file must be unique so that each value of `HREF` points to a single, unambiguous location. The following figure illustrates linking within a file. The browser highlights the word link. When you click on link, the browser positions the target right here in the active window.

*Figure A3.1 Linking within a File*

The following features are important at the starting point of this link:

- The `<A>` and `</A>` tags surround the text that the browser highlights.
- The `HREF` attribute points to the link's target. The target is an anchor tag whose `NAME` attribute matches the text that follows the pound sign in the `HREF` attribute. Because no text precedes the pound sign (`#`), the browser knows that the target is in the same file as the anchor.

When a link points to a target outside the file that is being displayed, the `HREF` attribute must include the path to that file. The path can be the path within the file system or the uniform resource locator (URL) of the file. The following figure illustrates a link from one file to another file that is specified with a URL. The browser highlights the word link. When you click on link, the browser positions the target right here in the active window or opens another window that displays the target.
The following features are important at the starting point (the anchor) of the link:

- The `<A>` and `</A>` tags surround the text that the browser highlights.
- The **HREF** attribute points to the link's target. The text that precedes the pound sign (`#`) identifies the file that contains the target.

ODS provides features that enable you to customize the text that precedes the pound sign and the text that follows the pound sign. For information about how to do this, see the discussions of file-specification, **ANCHOR=**, **BASE=**, **PATH=**, and **GPATH=** in the "ODS HTML Statement " on page 379 as well as "How ODS Constructs Links and References" on page 1103.

HTML implements references in much the same way as it implements links. The main difference is that a link points to a particular location within a file and that a reference points to the file itself. HTML uses the **SRC** attribute to identify a file to reference. The value of the **SRC** attribute is constructed the same way that the value of the **HREF** attribute is constructed except that there is no pound sign and no text following it.

### How ODS Constructs Links and References

Several options in the ODS HTML statement affect how ODS constructs the links and references that point from the frame to the table of contents, table of pages, and body file and from the table of contents or table of pages to the body file. Links are made as **HREF** attributes on `<A>` (anchor) tags inside the HTML files. Each **HREF** attribute points to the **NAME** attribute on another `<A>` tag. The **HREF** must identify both the file that contains
the target and the name of the anchor within that file. The value of HREF must be a valid
target in a valid URL. It uses the following form:

```html
<A href="URL#anchor-name"/>
```

ODS constructs the value of an HREF attribute based on information that you provide in
the ODS HTML statement.

**Note:** HTML references to files use other tags, but the logic for creating the string that
identifies the file is the same as the logic for creating an HREF attribute. For more
information, see “How ODS Constructs Links and References” on page 1103.

The URL in an HREF attribute includes information from three options in the ODS
HTML statement:

- the BASE option
- the GPATH= or the PATH= option
- the BODY=, the CONTENTS=, or the PAGE= option

1. If you specify BASE=, then the value of that option is the first part of the URL for
every HREF attribute that ODS writes.

2. If you specify GPATH= or PATH=, then the next part of the URL in an HREF
attribute comes from that option.

If the file that you are linking to is a high-resolution graphic, then ODS uses
information from the GPATH= option as the next part of the HREF. For information
about these options, see the discussion of GPATH= and the discussion of PATH= in
the “ODS HTML Statement” on page 379. The following table shows how ODS
uses information from the GPATH= option in the URL in HREF attributes:

<table>
<thead>
<tr>
<th>file-specification in GPATH=</th>
<th>URL= Suboption</th>
<th>Information ODS Uses in the Second Part of the URL in the HREF attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>An external-file or libref.catalog</td>
<td>Not specified</td>
<td>The name of the file</td>
</tr>
<tr>
<td>An external-file or libref.catalog</td>
<td>Specified, but not NONE</td>
<td>The value of the URL= suboption</td>
</tr>
<tr>
<td>An external-file or libref.catalog</td>
<td>NONE</td>
<td>No information from GPATH=</td>
</tr>
<tr>
<td>A fileref</td>
<td>Specified or not specified</td>
<td>No information from GPATH=</td>
</tr>
</tbody>
</table>

* If you do not specify GPATH=, then ODS uses the value of PATH= to create this part of the HREF.

If the file that you are linking to is not a high-resolution graphic, then ODS uses
information from the PATH= option as the next part of the HREF. The following
table shows how ODS uses information from the PATH= option in the URL in HREF
attributes:
Table A3.2  Building an HREF Attribute from the PATH= Option

<table>
<thead>
<tr>
<th>file-specification</th>
<th>URL= Suboption</th>
<th>Information Used in the Second Part of the URL in the HREF Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>external-file or</td>
<td>Not specified</td>
<td>The name of the file</td>
</tr>
<tr>
<td>libref.catalog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>external-file or</td>
<td>Specified, but not NONE</td>
<td>The value of the URL= suboption</td>
</tr>
<tr>
<td>libref.catalog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>external-file or</td>
<td>NONE</td>
<td>No information from PATH=</td>
</tr>
<tr>
<td>libref.catalog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fileref</td>
<td>Specified or not specified</td>
<td>No information from PATH=</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: If you use a fileref as the file specification in the BODY=, CONTENTS=, or PAGE= option in the ODS HTML statement, and you do not use the URL= suboption in that option, then ODS does not use information from GPATH= or PATH= when it creates the complete URL for any corresponding HREF attributes.

3. The last part of the URL that is used in an HREF attribute is, by default, the name of the file that contains the target. ODS determines the name of the file from the file-specification that you use in the BODY=, CONTENTS=, or PAGE= option. (ODS does not create links or references to frame files.) For more information about these options, see “ODS MARKUP Statement” on page 488.

If you specify the URL= suboption in one of these options, then ODS uses the string that you specify instead of the filename.

Note: If you use a fileref as the file specification and do not use the URL= suboption, then ODS does not use information from GPATH= or PATH= when it creates the complete URL for the HREF attribute.

The anchor-name comes from the value of the ANCHOR= option.

The following figure illustrates the creation of the HREF:

Figure A3.3  Creating the Value of an HREF Attribute
Files Produced by the HTML Destination

Overview

The HTML destination can produce four types of files: body, contents, frame, and page files. You create these files with options in the ODS HTML statement (see “ODS HTML Statement” on page 379 for details).

The Body File

The body file contains HTML output that is generated from the output objects that your SAS job creates. The style and the table template that the job uses determine the appearance and content of the tables and the cells within them.

Typically, when you route an output object that does not contain graphics to the HTML destination, ODS places the results within `<TABLE>` tags, generating them as one or more HTML tables.

Graphics output is produced according to the SAS code that generates it. Instead of using `<TABLE>` tags, the body file contains an `<IMG>` (image) tag that references the graphic. When you view the body file in a browser, you cannot tell that the graphic is not part of the body file because the `<IMG>` tag displays it in the browser.

Note: Very few procedures produce output objects that are neither tabular nor graphics. In these cases, the output is not tagged as an HTML table.

Titles and footnotes in the body file are generated as HTML tables of their own near the top and bottom of each page of HTML output.

Note: For graphics output, titles and footnotes are, by default, part of the graphics file. You can use the NOGTITLE and NOGFOOTNOTE options to place them in the body file instead. See the discussion of GTITLE and GFOOTNOTE in “ODS HTML Statement” on page 379 for more information.

All `<TABLE>` tags and all `<IMG>` tags are potential targets for links or references. For more information, see “How ODS Constructs Links and References” on page 1103. Therefore, ODS must provide an `<A>` tag with a NAME attribute close to each `<TABLE>` and `<IMG>` tag for links and references to point to. The NAME attribute on the anchor tag becomes the final part of any reference or link to the table. ODS inserts anchor tags in its HTML output as follows:

- ODS places an anchor tag near the top of each page, before all tables on the page (including the table that holds the titles) and before all images. This anchor is the target for links to the first table (excluding any titles) or to the first image on the page.

  Note: Each procedure or DATA step starts a new page. In addition, ODS produces a new page of output whenever the SAS program explicitly asks for a new page. For example, if you use the page dimension in PROC TABULATE, then you create a page for each value of the variable that defines the pages. In this context, the word "page" has nothing to do with the PAGESIZE= setting in your SAS session.

- ODS places an anchor tag slightly before each `<TABLE>` tag, provided that the table contains results (not titles or footnotes) and that it is not the first table or image on the page.
• ODS places an anchor tag slightly before each `<IMG>` tag, provided that it is not the first table or image on a page.

The following figure illustrates the placement of anchor tags from a SAS job that executes two procedures. The first procedure creates two HTML tables of results on a single page. The page also includes an HTML table for the title and one for the footnote. Solid arrows indicate which `<A>` tag ODS uses as a target for each table. The second procedure creates a GIF file. The titles for this procedure are part of the GIF file (the default behavior). Again, the solid arrow indicates which anchor tag ODS uses as a target when it creates a link to the image. The dashed arrow points to the file that the `<IMG>` tag references.
Figure A3.4 Placement of <A> (anchor) Tags in HTML Output
For a view of this same file through a browser, see Figure A3.6 on page 1112.

**The Contents File**

The contents file contains a link to the body file for each HTML table that ODS creates from procedure or DATA step results. The targets for these links are the values of the **NAME** attributes on the anchor tags that are in the body file. For more information, see “The Body File” on page 1106. For example, an anchor tag that links to the second HTML table of results in Figure A3.4 on page 1108 looks like this:

```html
<A href="pop-body.htm#IDX1">
```

In this anchor tag,

- pop-body.htm identifies the file that contains the target.
- #IDX1 provides the name of the target.

You can view the contents file directly in the browser, or, if you make a frame file, you can see the contents file as part of the frame file. For more information, see “The Frame File” on page 1109.

**The Page File**

The page file contains a link to the body file for each page of HTML output that ODS creates from procedure or DATA step results. The targets for these links are the values of the **NAME** attributes on the anchor tags that are in the body file. For more information, see “The Body File” on page 1106. For example, an anchor tag that links to the second page of results in Figure A3.4 on page 1108 looks like this:

```html
<A href="pop-body.htm#IDX2">
```

In this anchor tag,

- pop-body.htm identifies the file that contains the target.
- #IDX2 provides the name of the target.

You can view the page file directly in the browser, or, if you make a frame file, you can see the page file as part of the frame file. For more information, see “The Frame File” on page 1109.

**The Frame File**

The frame file provides a simultaneous view of the body file and the contents file, the page file, or both. The following figure illustrates how a frame that references both the contents and page files looks (in part) to an ASCII editor. The **SRC** attribute identifies a file to display in the browser. ODS constructs the value for the **SRC** attribute the same way it constructs the value for an **HREF** attribute in a page or contents file. For more information, see Figure A3.5 on page 1110.
options nodate pageno=1 linesize=80 pagesize=72;
data statepop;
  input State $ CityPop_80 CityPop_90 NonCityPop_80 NonCityPop_90 Region @@;
  label citypop_80= '1980 metropolitan pop in millions'
     noncitypop_80='1980 nonmetropolitan pop in millions'
     citypop_90=   '1990 metropolitan pop in millions'
     noncitypop_90='1990 nonmetropolitan pop in million'
     region='Geographic region';
datalines;
ME  .405  .443  .721  .785  1 NH  .535  .659  .386  .450  1
VT  .133  .152  .378  .411  1 MA  5.530 5.788 .207 .229  1
RI  .886  .938  .061  .065  1 CT  2.982 3.148 .126 .140  1
PA 10.067 10.083 1.798 1.799 1 DE .496  .553  .098  .113  2
MD  3.920 4.439 .297  .343  2 DC .638  .607  .     .   2
VA  3.966 4.773 1.381 1.414 2 WV .796  .748 1.155 1.045  2
NC  3.749 4.376 2.131 2.253 2 SC  2.114 2.423 1.006 1.064  2
KY  1.735 1.780 1.925 1.906 2 TN  3.045 3.298 1.546 1.579  2
AL  2.560 2.710 1.334 1.331 2 MS .716 .776 1.805 1.798  2
AR  .963 1.040 1.323 1.311 2 LA  3.125 3.160 1.082 1.060  2
OK  1.724 1.870 1.301 1.276 2 TX 11.539 14.166 2.686 2.821  2
OH  8.791 8.826 2.007 2.021 3 IN  3.885 3.962 1.605 1.582  3
IL  9.461 9.574 1.967 1.857 3 WI  7.719 7.698 1.543 1.598  3
WI  3.176 3.331 1.530 1.561 3 MN  2.674 3.011 1.402 1.364  3
IA  1.198 1.200 1.716 1.577 3 MO  3.314 3.491 1.603 1.626  3
ND  .234  .257  .418  .381  3 SD  .194  .221  .497  .475  3
NE  .728  .787  .842  .791  3 KS  1.184 1.333 1.180 1.145  3
MT  .189  .191  .598  .608  4 ID  .257  .296  .687  .711  4
WY  .141  .134  .329  .319  4 CO  2.326 2.686  .563  .608  4
NM  .675  .842  .628  .673  4 AZ  2.264 3.106 .453  .559  4
UT  1.128 1.336 1.333 1.387 4 NV  .666 1.014 .135 .183  4
WA  3.366 4.036 .776 .830  4 OR  1.799 1.985 .834 .858  4
CA 22.907 28.799 .760 .961  4 AK  .174 .226 .227 .324  4
HI  .763  .836  .202  .272  4
;
run;

data statepop2 (drop=tempvar);
  length state 4;
  set statepop (rename={state=tempvar});
  where tempvar in('AZ', 'NM', 'TX', 'OK');
  state=stfips(tempvar);
run;

ods html body='pop-body.htm'
  contents='pop-contents.htm'
  page='pop-page.htm'
  frame='pop-frame.htm'
  path='../ods'
  (url=none);
ods select basicmeasures testsforlocation;
proc univariate data=statepop2 mu0=3.5;
  var citypop_90;
  title 'United States Census of Population and Housing';
  footnote 'Data from 1990';
run;
ods listing close;
goptions reset=global gunit=pct cback=white
  colors=(black blue green red)
  ftext=swiss ftitle=swissb htitle=6 htext=4;
data states;
  set maps.us;
  where state in(04, 35, 40, 48);
run;
goptions target=gif transparency noborder;
title '1990 Metropolitan Population';
title2 f=swissb '(Arizona, New Mexico, Texas, and Oklahoma)';
proc gmap map=states data=statepop2;
  format citypop_90 comma9.;
  id state;
  prism citypop_90 / discrete;
run;
quit;
ods html close;
ods listing;
Figure A3.6  Browser View of HTML Frame File
Using a z/OS UNIX System Services HFS Directory for HTML Output

/* Specify the files to create for the HTML output. */
/* The PATH= option specifies the location for all */
/* the HTML files. The URL= suboption prevents */
/* information from PATH= from appearing in the */
/* links and references that ODS creates. The URLs */
/* will be the same as the file specifications. */
ods html body='odsexample-body.htm'
    contents='odsexample-contents.htm'
    page='odsexample-page.htm'
    frame='odsexample-frame.htm'
    path='~'(url=none);

Using a z/OS PDSE for EBCDIC HTML Output

/* Allocate a PDSE for the HTML Output. */
filename pdsehtml '.example.htm'
    dsntype=library dsorg=po
    disp=(new, catlg, delete);

/* Specify the files to create for the HTML output. */
/* These files are PDSE members. */
/* The PATH= option specifies the location for all */
/* the HTML files. The URL= suboption prevents */
/* information from PATH= from appearing in the */
/* links and references that ODS creates. The URLs */
/* will be the same as the file specifications. */
Using a z/OS PDSE for ASCII HTML Output

/* Allocate a PDSE for the HTML Output. */
filename pdsehtml '.example.htm'
dsntype=library dsorg=po
disp=(new, catlg, delete);

/* Specify the files to create for the HTML output. */
/* These files are PDSE members. */
/* The URL= suboption in the HTML-file */
/* specifications provides a URL that will be valid */
/* after the PDSE members have been moved to an */
/* ASCII file system. When the files are */
/* transferred, they must retain their member names */
/* and have the ".htm" extension added in order for */
/* these URLs to be correct. */
/* The PATH= option specifies the location for all */
/* the HTML files. The URL= suboption in the PATH= */
/* option prevents information from PATH= from */
/* appearing in the links and references that ODS */
/* creates because it will not be a valid URL for */
/* the ASCII file system. */
/* The TRANTAB= option creates ASCII HTML that */
/* you can send to an ASCII-based Web server. */

ods html body='odsexb' (url='odsexb.htm')
  contents='odsexc' (url='odsexc.htm')
  page='odsexp' (url='odsexp.htm')
  frame='odsexf'
  path='.example.htm' (url=none)
  trantab=ascii;

Note: Use a binary transfer to move the files to the Web server.
Appendix 5
PowerPoint Transition Options and the Effect Options Supported

The following table lists the ODS destination for PowerPoint slide transitions (OPTIONS (TRANSITION=) and effects (OPTIONS (EFFECT_OPTIONS=) ) supported for each slide transition. See OPTIONS on page 628 for information.

Table A5.1  PowerPoint Transitions and Effects Supported for Each Slide Transition

<table>
<thead>
<tr>
<th>Transition</th>
<th>Microsoft PowerPoint 2013 Support</th>
<th>Effect Options</th>
<th>Effect Options Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>airplane</td>
<td>y</td>
<td>right</td>
<td>right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>left</td>
<td></td>
</tr>
<tr>
<td>blinds</td>
<td>n</td>
<td>vertical</td>
<td>vertical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>horizontal</td>
<td></td>
</tr>
<tr>
<td>box</td>
<td>n</td>
<td>from_right</td>
<td>from_right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_bottom</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_right</td>
<td></td>
</tr>
<tr>
<td>checkerboard</td>
<td>n</td>
<td>from_left</td>
<td>from_left</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_right</td>
<td></td>
</tr>
<tr>
<td>clock</td>
<td>n</td>
<td>clockwise</td>
<td>clockwise</td>
</tr>
<tr>
<td></td>
<td></td>
<td>counterclockwise</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>wedge</td>
<td></td>
</tr>
<tr>
<td>comb</td>
<td>n</td>
<td>vertical</td>
<td></td>
</tr>
<tr>
<td>Transition</td>
<td>Microsoft PowerPoint 2013 Support</td>
<td>Effect Options</td>
<td>Effect Options Default</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------</td>
<td>----------------</td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>horizontal</td>
<td>horizontal</td>
</tr>
<tr>
<td>conveyor</td>
<td>n</td>
<td>from_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_right</td>
<td>from_right</td>
</tr>
<tr>
<td>cover</td>
<td>n</td>
<td>from_right</td>
<td>from_right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_bottom</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top_right</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_bottom_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_bottom_right</td>
<td></td>
</tr>
<tr>
<td>crush</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cube</td>
<td>n</td>
<td>from_right</td>
<td>from_right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_bottom</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_left</td>
<td></td>
</tr>
<tr>
<td>curtains</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cut</td>
<td>n</td>
<td>through_black</td>
<td></td>
</tr>
<tr>
<td>dissolve</td>
<td>n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>doors</td>
<td>n</td>
<td>vertical</td>
<td>vertical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>horizontal</td>
<td></td>
</tr>
<tr>
<td>drape</td>
<td>y</td>
<td>left</td>
<td>left</td>
</tr>
<tr>
<td></td>
<td></td>
<td>right</td>
<td></td>
</tr>
<tr>
<td>fade</td>
<td></td>
<td>smoothly</td>
<td>smoothly</td>
</tr>
<tr>
<td></td>
<td></td>
<td>through_black</td>
<td></td>
</tr>
</tbody>
</table>
## PowerPoint Transition Options and the Effect Options Supported

<table>
<thead>
<tr>
<th>Transition</th>
<th>Microsoft PowerPoint 2013 Support</th>
<th>Effect Options</th>
<th>Effect Options Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>fall_over</td>
<td>y</td>
<td>left</td>
<td>left</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>right</td>
</tr>
<tr>
<td>ferris_wheel</td>
<td>n</td>
<td>from_right</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_left</td>
<td></td>
</tr>
<tr>
<td>flash</td>
<td>n</td>
<td>left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>right</td>
<td>right</td>
</tr>
<tr>
<td>flip</td>
<td>n</td>
<td>left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>right</td>
<td>right</td>
</tr>
<tr>
<td>fly_through</td>
<td>n</td>
<td>in</td>
<td>in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>in_with_bounce</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>out</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>out_with_bounce</td>
<td></td>
</tr>
<tr>
<td>fracture</td>
<td>y</td>
<td>from_right</td>
<td>from_right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_left</td>
<td></td>
</tr>
<tr>
<td>gallery</td>
<td>n</td>
<td>hexagons_from_left</td>
<td>hexagons_from_left</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hexagons_from_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>hexagons_from_right</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>hexagons_from_top</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>hexagons_from_bottom</td>
<td></td>
</tr>
<tr>
<td>glitter</td>
<td>n</td>
<td>diamonds_from_left</td>
<td>diamonds_from_left</td>
</tr>
<tr>
<td></td>
<td></td>
<td>diamonds_from_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>diamonds_from_right</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>diamonds_from_top</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>diamonds_from_bottom</td>
<td></td>
</tr>
<tr>
<td>honeycomb</td>
<td>n</td>
<td>from_right</td>
<td>from_right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_bottom</td>
<td></td>
</tr>
<tr>
<td>Transition</td>
<td>Microsoft PowerPoint 2013 Support</td>
<td>Effect Options</td>
<td>Effect Options Default</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------</td>
<td>----------------</td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_left</td>
<td></td>
</tr>
<tr>
<td>origami</td>
<td>y</td>
<td>right</td>
<td>right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>left</td>
<td></td>
</tr>
<tr>
<td>page_curl</td>
<td>y</td>
<td>double_left</td>
<td>double_left</td>
</tr>
<tr>
<td></td>
<td></td>
<td>double_right</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>single_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>single_right</td>
<td></td>
</tr>
<tr>
<td>pan</td>
<td>n</td>
<td>from_bottom</td>
<td>from_bottom</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_right</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top</td>
<td></td>
</tr>
<tr>
<td>peel_off</td>
<td>y</td>
<td>left</td>
<td>left</td>
</tr>
<tr>
<td></td>
<td></td>
<td>right</td>
<td></td>
</tr>
<tr>
<td>prestige</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>push</td>
<td>n</td>
<td>from_bottom</td>
<td>from_bottom</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_right</td>
<td></td>
</tr>
<tr>
<td>random</td>
<td>n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>randombars</td>
<td>n</td>
<td>vertical</td>
<td>vertical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>horizontal</td>
<td></td>
</tr>
<tr>
<td>reveal</td>
<td>n</td>
<td>smoothly_from_right</td>
<td>smoothly_from_right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>smoothly_from_left</td>
<td></td>
</tr>
<tr>
<td>Transition</td>
<td>Microsoft PowerPoint 2013 Support</td>
<td>Effect Options</td>
<td>Effect Options Default</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------</td>
<td>----------------</td>
<td>------------------------</td>
</tr>
<tr>
<td></td>
<td>smoothly_through_black_from_right</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>smoothly_through_black_from_left</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ripple</td>
<td>n</td>
<td>center</td>
<td>center</td>
</tr>
<tr>
<td></td>
<td>from_bottom_left</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>from_bottom_right</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>from_top_left</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>from_top_right</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rotate</td>
<td>n</td>
<td>from_right</td>
<td>from_right</td>
</tr>
<tr>
<td></td>
<td>from_left</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>from_bottom</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>from_top</td>
<td></td>
<td></td>
</tr>
<tr>
<td>shape</td>
<td>n</td>
<td>circle</td>
<td>circle</td>
</tr>
<tr>
<td></td>
<td>diamond</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>plus</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>in</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>out</td>
<td></td>
<td></td>
</tr>
<tr>
<td>shred</td>
<td>n</td>
<td>strips_in</td>
<td>strips_in</td>
</tr>
<tr>
<td></td>
<td>strips_out</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>particle_in</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>particle_out</td>
<td></td>
<td></td>
</tr>
<tr>
<td>split</td>
<td>n</td>
<td>vertical_in</td>
<td>vertical_out</td>
</tr>
<tr>
<td></td>
<td>vertical_out</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>horizontal_in</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>horizontal_out</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transition</td>
<td>Microsoft PowerPoint 2013 Support</td>
<td>Effect Options</td>
<td>Effect Options Default</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------</td>
<td>----------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>switch</td>
<td>n</td>
<td>left</td>
<td>right</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>right</td>
</tr>
<tr>
<td>uncover</td>
<td>n</td>
<td>from_bottom</td>
<td>from_right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_right</td>
<td>from_right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top_right</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_bottom_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_bottom_right</td>
<td></td>
</tr>
<tr>
<td>vortex</td>
<td>n</td>
<td>from_left</td>
<td>from_left</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_right</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_bottom</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top</td>
<td></td>
</tr>
<tr>
<td>wind</td>
<td>y</td>
<td>right</td>
<td>right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>left</td>
<td></td>
</tr>
<tr>
<td>window</td>
<td>n</td>
<td>vertical</td>
<td>vertical</td>
</tr>
<tr>
<td></td>
<td></td>
<td>horizontal</td>
<td></td>
</tr>
<tr>
<td>wipe</td>
<td>n</td>
<td>from_bottom</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_right</td>
<td>from_right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top_right</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_top_left</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_bottom_left</td>
<td></td>
</tr>
<tr>
<td>Transition</td>
<td>Microsoft PowerPoint 2013 Support</td>
<td>Effect Options</td>
<td>Effect Options Default</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------</td>
<td>----------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>from bottom right</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>zoom</td>
<td>n</td>
<td>in</td>
<td>in</td>
</tr>
<tr>
<td>out</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>zoom_and_rotate</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* A ‘y’ in this column means that the transition is supported only in PowerPoint 2013.
Here is the recommended reading list for this title. For a complete list of SAS publications, go to http://support.sas.com/publishing/index.html.

- *SAS Graph Template Language: Reference*
- *SAS ODS Graphics: Procedures Guide*
- *SAS Graph Template Language: User's Guide*
- *Getting Started with the SAS Output Delivery System*
- *SAS Output Delivery System: Advanced Topics*
- *SAS Output Delivery System: Procedures Guide*
- *Base SAS Procedures Guide*
- *SAS Language Reference: Concepts*
- *SAS Data Set Options: Reference*
- *SAS Functions and CALL Routines: Reference*
- *SAS Statements: Reference*
- *SAS System Options: Reference*
- *Step-by-Step Programming with Base SAS*
- *SAS 9 ODS CSS Tip Sheet*
- *SAS 9 ODS EPUB Tip Sheet*
- *SAS 9 ODS Layout Tip Sheet*
- *SAS 9 ODS EXCELXP Tip Sheet*
- *SAS 9 Report Writing Interface Tip Sheet*
- *SAS 9 ODS List and Text Block Tip Sheet*

SAS offers instructor-led training and self-paced e-learning courses to help you get started with the SAS Output Delivery System and learn advanced techniques for the SAS Output Delivery System. For more information about the courses available, see http://support.sas.com/learn/.

The recommended reading list from *SAS Press* includes:

- *Carpenter's Guide to Innovative SAS Techniques*
• Getting Started with the Graph Template Language in SAS: Examples, Tips, and Techniques for Creating Custom Graphs
• The Little SAS Book: A Primer
• The Little SAS Book for Enterprise Guide
• ODS Techniques: Tips for Enhancing Your SAS Output
• Output Delivery System: The Basics and Beyond
• PROC DOCUMENT by Example Using SAS
• PROC TEMPLATE Made Easy: A Guide for SAS Users
• Statistical Graphics Procedures by Example: Effective Graphs Using SAS
• Statistical Graphics in SAS: An Introduction to the Graph Template Language and the Statistical Graphics Procedures

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

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

the level of access that a user has to an item store. The possible access modes are read, write, and update. See also item store.

**ActiveX**

a technology developed by Microsoft that is used to add interactivity to web pages.

**ActiveX control**

a type of web application that is developed specifically for the Windows operating environment. ActiveX controls can provide web users with interactive capabilities.

**after-note**

in ODS, a note that is displayed after an output object each time the output object is displayed. The text is assigned to an output object by the procedure that produced the object. See also output object, ODS document, before-note.

**aggregate storage location**

a location in an operating system that can contain a group of distinct files. Depending on the operating system, the location could be a directory, folder, or partitioned data set.

**aliasing**

a visual effect in computer-generated images that produces several types of rendering problems, such as jagged edges along straight lines or polygon boundaries. Aliasing can occur when you try to render an object smaller than pixel size or a very narrow object. In a complex scene, fine details are sometimes lost or distorted beyond recognition due to aliasing. See also anti-aliasing.

**annotation**

a label, marker, or note that is not obtained from the data but is placed on a graph independently. Such annotations might or might not be linked to data values in the plot.

**anti-aliasing**

a rendering technique for improving the appearance of text and curved lines in a graph by blurring the jagged edges normally present. The degree of improvement is relative to the nature of the graphical content (for example, vertical and horizontal lines do not benefit from anti-aliasing). Extra processing is required to perform anti-aliasing. See also aliasing.
**before-note**
in ODS, a note that is displayed before an output object each time the output object is displayed. The text is assigned to the output object by the procedure that produced the object. See also after-note, output object, ODS document.

**cellvalue**
one of the possible values that PROC FREQ can produce for a crosstabulation table. Cellvalues are defined by the DEFINE CELLVALUE statement in a crosstabulation table template.

**column attribute**
a formatting property that controls aspects of a column, such as the appearance of the cells contents, presentation of data panels, and customization of column headings. Column attributes have a reserved name and value defined in ODS.

**crosstab (crosstabulation table)**
a two-dimensional table that shows frequency distributions or other aggregate statistics for the intersections of two or more category data items. In a crosstab, categories are displayed on both the columns and rows, and each cell value represents the data result from the intersection of the categories on the specific row and column. See also frequency table.

**crosstabulation table**
See crosstab.

**data component**
a form, similar to a SAS data set, that contains the results (numbers and characters) of a DATA step or PROC step that supports ODS.

**destination**
See ODS destination.

**device-based graphic**
a graph created with SAS/GRAFH software for which a user-specified or default device (DEVICE= option) controls certain aspects of the graphical output.

**dictionary variable**
a type of memory variable that consists of an array that contains a list of numbers or text strings that can be identified by a key. A dictionary variable has, as part of its name, a preceding '$' symbol and a subscript that contains a text string. The text string within the subscript is called a key. For example, the following dictionary variable identifies the entry in the $MyDictionary variable that contains the text-string 'dog': $MyDictionary['dog']. See also ODS event, list variable, memory variable, scalar variable.

**DOCUMENT destination**
a SAS Output Delivery System (ODS) destination that produces a hierarchy of output objects. The DOCUMENT destination enables users to render multiple ODS output formats without rerunning a PROC step or DATA step, and it gives users more control over the structure of the output. See also ODS destination.

**exclusion list**
a list that tells ODS which output objects to exclude from a specified ODS destination.
**footer attribute**

a formatting property that controls aspects of a footer, such as the appearance of the footer contents and the placement of the footer. The footer attribute has a reserved name and value defined in ODS. *See also header attribute.*

**frequency table**

a table that lists each of the distinct values that a variable has within all of the observations in a SAS data set. For each value, the table also lists the number of observations in which the variable has that value.

**graph segment**

in ODS, a file type or output object that contains a graph. Graphs are created in some SAS procedures, including those in SAS/GRAPH. The graph output object is referenced as a GRSEG. *See also output object.*

**graphics template**

*See ODS template.*

**header attribute**

a formatting property that controls aspects of a header, such as the appearance of the header contents and the placement of the header. The header attribute has a reserved name and value defined in ODS. *See also footer attribute.*

**HTML**

*See HyperText Markup Language.*

**HyperText Markup Language (HTML)**

a coding system in which the codes indicate the layout and style of the text in a text file. Other HTML codes enable you to embed electronic objects such as images, sounds, video streams, and applets (small software applications) into HTML documents. All web browsers can process HTML documents.

**inline formatting**

a feature of the Output Delivery System (ODS) that allows you to insert simple formatting text into ODS output by using the ODS ESCAPECHAR statement.

**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, which is similar to the directory structures that are used by UNIX System Services or by Windows. For example, a particular value might be stored and located using a directory path (root_dir/sub_dir/value). The SAS Registry is an example of an item store. *See also template store.*

**list variable**

a type of memory variable that consists of an array that contains a list of numbers or text strings that are indexed. A list variable has, as part of its name, a preceding 's' symbol and a subscript that is empty or contains a number or numeric variable. The number within the subscript is called an index. For example, the list variable $Mylist[2]$ identifies the second entry in the list variable $Mylist$. In this case, the index is 2. *See also dictionary variable, ODS event, memory variable, scalar variable.*

**LISTING destination**

an ODS destination that produces traditional SAS output (monospace format). *See also ODS destination.*
LISTING output
SAS procedure output that is in a monospace font. All text in listing output has the same font size, and no special font styles are applied to it.

marker
a symbol such as a diamond, a circle, or a triangle that is used to indicate the location of, or annotate, a data point in a plot or graph.

markup family
See ODS markup family.

markup language
a set of codes that are embedded in text in order to define layout and certain content.

measured RTF
a tagset that enables users to specify how and where page breaks occur in RTF documents and when to place titles and footnotes into the body of a page.

memory variable
within an ODS event, an area of memory that contains numeric data, character data, or lists of numeric or character data. A memory variable can be classified as a dictionary variable if it is created with a subscript that contains a key, or a list variable if it is created with a subscript that is empty or contains an index. If you do not specify a key or an index, then the memory variable is a numeric or character scalar variable, depending on the variable's value. See also dictionary variable, list variable, scalar variable, ODS event.

ODS
See Output Delivery System.

ODS destination (destination)
a designation that the Output Delivery System uses to generate a specific type of output. Types of ODS destinations include but are not limited to HTML, XML, listing, PostScript, RTF, and SAS data sets.

ODS document
a hierarchy of output objects created by the DOCUMENT procedure. These objects are in an unformatted form and are placed in a SAS item store. See also item store, output object.

ODS document path
 the location of an entry within an ODS document. See also ODS document, ODS entry.

ODS entry
an item in an ODS document. An ODS entry can be either a link, an output object, a file, or a partitioned data set.

ODS event
within a tagset definition, an action that causes output to be generated. Events are usually triggered by SAS but can also be triggered by other events.

ODS Graphics
an extension to ODS that is used to create analytical graphs using the Graph Template Language.
ODS markup family (markup family)
a group of ODS statements that produce SAS output that is formatted using a markup language such as HTML (HyperText Markup Language), XML (Extensible Markup Language), and LaTeX. SAS supplies many markup languages for you to use, ranging from DOCBOOK to TROFF. You can specify a markup language that SAS supplies, or you can create one of your own and store it as a user-defined markup language. See also ODS destination, ODS printer family.

ODS output
formatted output that is generated by any of the ODS destinations. For example, the OUTPUT destination produces SAS data sets, the LISTING destination produces listing output, and the HTML destination produces output that is formatted in Hypertext Markup Language.

ODS package
a container for information or digital content that is generated or collected for delivery to a consumer. ODS packages allow ODS destinations to use the SAS Publishing Framework.

ODS printer family (printer family)
a group of ODS statements that produce output in a format such as PostScript (PS), PDF, or PCL that is suitable for printing on a high-resolution printer.

ODS style (style)
a combination of colors, fonts, lines, marker symbols, and so on that provide a specific appearance for SAS output. A style is defined in ODS by a style template.

ODS table
a table that is created with an IMSTAT procedure statement with the SAVE= option. The contents of the table resemble ODS output and are commonly used with the STORE statement. This enables the ODS output from one statement to be used as input for another statement.

ODS template (graphics template)
a description of how output should appear when it is formatted. ODS templates are stored as compiled entries in a template store, also known as an item store. Common template types include STATGRAPH, STYLE, CROSSTABS, TAGSET, and TABLE.

Output Delivery System (ODS)
a component of SAS software that can produce output in a variety of formats such as markup languages (HTML, XML), PDF, listing, RTF, PostScript, and SAS data sets.

output object
a programming object that contains the data that is generated by a DATA step or a PROC step and which can also contain a table definition that provides information about how to format that data.

printer family
See ODS printer family.

Publishing Framework
a component of SAS Integration Technologies that enables both users and applications to publish SAS files (including data sets, catalogs, and database views), and other digital content to a variety of destinations. The Publishing Framework also
provides tools that enable both users and applications to receive and process published information.

**replay**

in ODS, the regeneration of output by the DOCUMENT procedure, in the same or different format, without rerunning analyses or data queries.

**root file location**

the top level of a file location in an ODS document. A root file location is not contained within another file location and does not have a name assigned to it. A root file location is similar to the root directory of a Windows operating environment. See also ODS document.

**SASEDOC engine**

a SAS engine that associates a SAS libref (library reference) with one or more ODS output objects that are stored in an ODS document.

**scalar variable**

a type of memory variable that contains one-dimensional numeric or character data. Once created, scalar variables are globally available in all events. See also dictionary variable, list variable, memory variable.

**stream variable**

within an ODS event, a temporary item store that contains output. While the stream variable is open, all output is directed to it until it is closed. See also item store, ODS event, tagset.

**style**

See ODS style.

**style attribute**

a visual property, such as color, font properties, and line characteristics, that is defined in ODS with a reserved name and value. Style attributes are collectively referenced by a style element within a style template.

**style element**

a named collection of style attributes that affects specific parts of ODS output. For example, a style element might specify the color and font properties of title text or other text in in a table or graph. See also table element, style attribute.

**style element inheritance**

the concept that a child style element receives all of the style attributes that are specified in its parent style element, unless the child style element overrides those attributes.

**style template**

a type of ODS template that defines the visual aspects (colors, fonts, lines, markers, and so on) of SAS output for a specific style. A style template defines style elements, and each style element consists of style attributes.

**table attribute**

a formatting property such as layout of headers, line spacing, and layout of rows and columns, that has a reserved name and value defined in ODS. See also table definition, table element.
**table definition**
a set of instructions that describe how to format output in the Output Delivery System (ODS).

**table element**
a collection of table attributes that each pertain to a particular column, header, or footer in a table in ODS output.

**table template**
a template that describes how to display 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 that uses it. Each table template contains or references table elements.

**tagset**
a template that defines how to create a type of markup language output from a SAS format. Tagsets produce markup output such as Hypertext Markup Language (HTML), Extensible Markup Language (XML), and LaTeX. See also markup language.

**tagset definition**
a template that specifies instructions for creating a markup language for your SAS output. The resulting output contains embedded instructions in order to define layout and some content. Each tagset definition contains event definitions and event attributes that control the generation of the output. SAS provides tagset definitions for a variety of markup languages. You can use the TEMPLATE procedure to modify any of these SAS tagsets or to create your own tagsets. See also ODS markup family.

**template store**
an item store that contains definitions that were created by the TEMPLATE procedure. Definitions that SAS provides are in the item store Sashelp.Tmplmst. You can store definitions that you create in any template store to which you have Write access. See also item store.
Index

Special Characters

_NULL_ option
  ODS DOCUMENT statement 209
_ODS_ option
  PUT statement 56, 865

A
  absolute layout statement 484, 903
  ABSTRACT= option
    ODS PACKAGE statement 557
  ABSTRACT= style attribute 1004
  accessibility
    ODS EPU5 statement 213, 249
    ODS EPU52 statement 225
    ODS EPU53 statement 249
    ODS, creating output 19
    Output Delivery System 19
  ACKGROUNDPOSITION= style attribute 1005
  Acrobat Distiller 703, 728
  actions
    ODS CHTML statement 172
    ODS CSVALL statement 196
    ODS DOCUMENT statement 208
    ODS HTML3 statement 355
    ODS LISTING statement 485
    ODS MARKUP statement 488
    ODS OUTPUT statement 534
    ODS PHTML statement 666
    ODS PRINTER statement 693
    ODS RTF statement 734
    ODS Tagset statement 774
  ACTIVELINKCOLOR= style attribute 1004
  ActiveX devices
    CODEBASE file path 118
  ALL option
    ODS EXCLUDE statement 322
  ANCHOR option
    ODS CHTML statement 173
    ODS EPU5 statement 216
  anchor tags
    base name for 492, 735, 806
    root name for 578, 693, 720
  ANCHOR= option
    ODS MARKUP statement 492
    ODS PDF statement 578
    ODS PRINTER statement 693
    ODS PS statement 720
    ODS RTF statement 735
  ANTIALIAS option
    ODS GRAPHICS statement 330
  ANTIALIAS= option
    ODS GRAPHICS statement 330
  ANTIALIASMAX= option
    ODS GRAPHICS statement 331
  APPEND option
    ODS PATH statement 564
  appending HTML files 146, 416
  applets
    viewing HTML output 114, 384
  ARCHIVE option
    ODS CHTML statement 174
    ODS HTML statement 114, 384
    ODS HTML3 statement 356
    ODS PHTML statement 667
    ODS Tagset statement 777
  ARCHIVE= option
    ODS MARKUP statement 492
  arguments
    ODS OUTPUT statement 534
  ARIMA procedure
    ODS table names 1064
  ASIS= style attribute 1004
  ATTRIBUTES option
    ODS CHTML statement 174
    ODS HTML statement 115, 384
    ODS HTML3 statement 357
ODS PHTML statement 668
ODS Tagset statement 778
ATTRIBUTES= option
ODS MARKUP statement 493
ATTRPRIORITY= option
ODS GRAPHICS statement 331
AUDIO function 275
AUTHOR option
ODS EXCEL statement 295
ODS POWERPOINT statement 626
ODS MARKUP statement 493
ODS PDF statement 578
ODS PRINTER statement 694
ODS PS statement 721
ODS RTF statement 736

B
BACKGROUND= style attribute 1005
BACKGROUNDCOLOR= style attribute 1005
BACKGROUNDIMAGE= style attribute 1005
BACKGROUNDREPEAT= Style Attribute 633, 1006
BASE option
ODS CHTML statement 175
ODS HTML statement 115, 385
ODS HTML3 statement 357
ODS HTML5 statement 429
ODS PHTML statement 668
ODS POWERPOINT statement 626
ODS TAGSETS.RTF statement 806
base text 579, 694, 721, 736, 806
HTML output 115, 385
BASE= option
ODS MARKUP statement 494
ODS PDF statement 579
ODS PRINTER statement 694
ODS PS statement 721
ODS RTF statement 736
batch jobs
ODS GRAPHICS statement for 346
benefits of ODS 3
body files 1106
creating 128, 185, 201, 369, 398, 440, 506, 571, 585, 680, 701, 727, 790
separate file per page of output 142, 412
BODY option
ODS CHTML statement 175
ODS CSVALL statement 197
ODS HTML3 statement 358
ODS HTML5 statement 429
ODS PHTML statement 669
ODS Tagset statement 779
BODY= option
ODS destination statements 14
ODS HTML statement 115, 385
ODS MARKUP statement 494
ODS RTF statement 741
suboptions 494
BODYSIZE= style attribute 1006
BODYTITLE= option
ODS RTF statement 736
BODYTITLE_AUX option
ODS RTF statement 737
BOLD function 276
BOOKMARK= option
ODS PDF statement 578
ODS PRINTER statement 693
ODS PS statement 720
BOOKMARKGEN= option
ODS PDF statement 580
ODS PRINTER statement 695
ODS PS statement 722
BOOKMARKLIST= option
ODS PDF statement 579
ODS PRINTER statement 695
ODS PS statement 721
bookmarks for PDF files 579, 580, 695, 721, 722
BORDER option
ODS GRAPHICS statement 332
BORDER= option
ODS GRAPHICS statement 332
BORDERBOTTOMCOLOR= style attribute 1007
BORDERBOTTOMSTYLE= style attribute 1007
BORDERBOTTOMWIDTH= style attribute 1008
BORDERCOLLAPSE= style attribute 1008
BORDERCOLOR= style attribute 1008
BORDERCORDARK= style attribute 1008
BORDERCOLORLIGHT= style attribute 1009
BORDERLEFTCOLOR= style attribute 1009
BORDERLEFTSTYLE= style attribute 1009
BORDERLEFTWIDTH= style attribute 1009
BORDERRIGHTCOLOR= style attribute 1010
BORDERRIGHTSTYLE= style attribute 1010
BORDERRIGHTWIDTH= style attribute 1010
BORDERSPACING= style attribute 1010
BORDERTOPCOLOR= style attribute 1010
BORDERTOPSTYLE= style attribute 1011
BORDERTOPWIDTH= style attribute 1012
BOX_SIZING option
  ODS EPUB statement 216
  ODS EPUB3 statement 236
  ODS EXCEL statement 296
  ODS HTML statement 117, 386
  ODS HTML5 statement 430
  ODS PHTML statement 670
  ODS RTF statement 737, 807
  buffers
    number of columns in 58, 867
BULLET= style attribute 1025
BYGROUP argument
  ODS EPUB statement 221
  ODS EUPB3 option
  ODS EPUB3 statement 242
BYLINE= option
  ODS GRAPHICS statement 332

C
CALENDAR procedure
  ODS table names 1045
CAPSTYLE= style attribute 1012
cascading style sheets
  applying to ODS output 527
  multiple, in one HTML document 526
CAT= option
  ODS DOCUMENT statement 209
CATALOG option
  ODS DOCUMENT statement 209
CATALOG procedure
  ODS table names 1045
catalogs
  copying GSREGs to 209
CATEGORY option
  ODS EXCEL statement 296
  ODS POWERPOINT statement 626
CELLHEIGHT= style attribute 1022
CELLPADDING= style attribute 1012
CELLSPACING= style attribute 1010
CELLWIDTH= style attribute 1037
character sets
  META declaration for HTML output
    117, 176, 359, 387, 431, 495, 670, 780
CHARSET option
  ODS CHTML statement 176
  ODS HTML statement 117, 387
  ODS HTML3 statement 359
  ODS HTML5 statement 431
  ODS PHTML statement 670
  ODS Tagset statement 780
CHARSET= option
  ODS MARKUP statement 495
CHART procedure
  ODS table names 1046
  CHTML destination 170
  CHTML tagset 775
CLASS= style attribute 1013
CLEAR action
  ODS Output statement 534
CLEAR option
  ODS PACKAGE statement 557
CLOSE action
  ODS CSVALL statement 196
  ODS DOCUMENT statement 208
  ODS LISTING statement 485
  ODS MARKUP statement 488, 491
  ODS OUTPUT statement 534
  ODS PCL statement 566
  ODS PDF statement 577
  ODS PRINTER statement 693
  ODS PS statement 719
  ODS RTF statement 734
CLOSE option
  ODS CHTML statement 172
  ODS EPUB statement 215
  ODS EPUB3 statement 235
  ODS EXCEL statement 294
  ODS HTML statement 112, 382
  ODS HTML3 statement 355
  ODS HTML5 statement 427
  ODS PHTML statement 666
  ODS POWERPOINT statement 625
  ODS Tagset statement 774
  ODS TAGSETS.RTF statement 805
CODE option
  ODS CHTML statement 177
  ODS HTML statement 117, 387
  ODS HTML3 statement 359
  ODS HTML5 statement 431
  ODS PHTML statement 670
  ODS Tagset statement 780
CODE= option
ODS MARKUP statement 495
CODEBASE option
ODS CHTML statement 177
ODS HTML statement 118, 388
ODS HTML3 statement 360
ODS PHTML statement 671
ODS Tagset statement 781
CODEBASE= option
ODS MARKUP statement 496
COLOR= option
ODS PCL statement 567
ODS PDF statement 580
ODS PRINTER statement 696
ODS PS statement 722
COLOR= style attribute 1013
colors
ODS PDF statement 580
ODS PRINTER statement 696
ODS PS statement 722
column attributes
values from data component 66, 162
column definitions
for multiple variables 62, 67, 158, 163
column pointer controls
ODS 56, 866
columns
assigning attributes to 78
for data components 60, 156
formats for 66, 162
justification 207
labels for 63, 67, 159, 163
number in buffers 58, 867
number in data components 58, 867
ODS PDF statement 581
ODS PRINTER statement 696
ODS PS statement 723
ODS RTF statement 737
specifying 65, 161
COLUMNS option
ODS TAGSETS.RTF statement 807
COLUMNS= option
ODS PCL statement 568
ODS PDF statement 581
ODS PRINTER statement 696
ODS PS statement 723
ODS RTF statement 737
COLUMNS= suboption
FILE PRINT ODS statement 60, 156
comma-delimited output 195
COMMENTS option
ODS EXCEL statement 296
ODS POWERPOINT statement 626
COMPARE procedure
ODS table names 1046
COMPRESS= option
ODS PRINTER statement 581, 697
compression
PDF files 581, 697
CONNECT= style attribute 1013
CONTENTPOSITION= style attribute 1014
contents file 1109
CONTENTS option
ODS CHTML statement 178
ODS HTML statement 119, 388
ODS HTML3 statement 361
ODS HTML5 statement 432
ODS PHTML statement 672
ODS RTF statement 738
ODS Tagset statement 782
CONTENTS procedure
ODS table names 1048
CONTENTS= option
ODS MARKUP statement 497
ODS PDF statement 582
ODS PRINTER statement 697
CONTENTSCROLLBAR= style attribute 1014
CONTENTS= option
ODS table names 1047
CSSSTYLE option
ODS EPUB statement 216
ODS EPUB3 statement 237
ODS EXCEL statement 296
ODS HTML statement 120, 390, 783
ODS HTML3 statement 362
ODS HTML5 statement 296, 433
ODS MARKUP statement 498
ODS PCL statement 568
ODS PDF statement 582
ODS PHTML statement 673
ODS PRINTER statement 697
ODS PS statement 723
ODS RTF statement 738
CSV tagset 775
CSVALL destination 195
CSVALL tagset 775
CSVBYLINE tagset 775
customized output
for output objects 39

D
DAGGER function 276
data components
binding to table templates 59, 155
column attribute values from 66, 162
columns for 60, 156
number of columns in 58, 867
DATA option
  ODS LISTING statement 485
data panels 485
data sets
  combined output data sets 538
  creating with/without MATCH_ALL
      option 545
  from output objects 534
  from similar output objects 542
  merging dissimilar output objects into
      537
DATA step
  column definitions for multiple
      variables 62, 67, 158, 163
  ODS and 53
  ODS enhanced features in 55
  ODS examples 69
  ODS reports with 54
DATA step statements
  ODS 103
  DATAPANEL= option
  ODS LISTING statement 485
DATASETS procedure
  ODS table names 1048
DATASKIN= style attribute 1016
DATASKINMAX= option
  ODS GRAPHICS statement 333
DATE function 276
decimal point
  in numeric columns 207
default devices 486, 499, 739
DESCRIPTION= option
  ODS PACKAGE statement 557
destination-independent input 34
DEVICD option
  ODS TAGSETS.RTF statement 807
DEVICE option
  ODS EPU BB statement 218
  ODS EPU BB3 statement 238
  ODS HTML statement 121, 391
  ODS HTML5 statement 434
DEVICE=
  ODS MARKUP statement 499
  ODS RTF statement 486, 739
DIR= option
  ODS DOCUMENT statement 209
DISCRETEMAX= option
  ODS GRAPHICS statement 333
DISPLAYOPTS= style attribute 1016
DOC_SEQNO= option
  LIBNAME statement, SASEDOC 166
DOCTYPE= style attribute 1017
DOCUMENT destination 34, 208
closing 208
excluding output objects 208
selecting output objects 209
writing selection/exclusion lists to log
      209
DOM option
  ODS CHTML statement 179
  ODS CSVALL statement 198
  ODS EPUB statement 218
  ODS EPUB3 statement 238
  ODS HTML statement 122, 297, 392,
      434, 784
  ODS HTML3 statement 363
  ODS MARKUP statement 500
  ODS PCL statement 569
  ODS PDF statement 583
  ODS PHT ML statement 674
  ODS PRINTER statement 699
  ODS PS statement 725
  ODS RTF statement 740
  ODS TRACE statement 855
double trailing @
  PUT _ODS_ statement 56, 865
DPI option
  ODS EXCELS statement 297, 298
  ODS POWERPOINT statement 627
DPI= option
  ODS EPU BB statement 221
  ODS EPU BB3 statement 242
  ODS LISTING statement 488
  ODS PCL statement 569
  ODS PDF statement 583
  ODS PRINTER statement 699
  ODS PS statement 725
  ODS RTF statement 742
DRILLTARGET= option
  ODS GRAPHICS statement 334
DROPSHADOW= style attribute 1017
DTDs
  creating, with XML files 521
dynamic attributes
  default values for 62, 158
dynamic graphics output
  attributes between tags 493
  parameters between tags 509
DYNAMIC= attribute suboption
  FILE PRINT ODS statement 66, 162
DYNAMIC= suboption
  FILE PRINT ODS statement 62, 158

E
EMPHASIS function 277
ENCODING option
  ODS CHTML statement 180
  ODS CSVALL statement 198
  ODS HTML statement 122, 392
  ODS HTML3 statement 363
ODS HTML5 statement  435
ODS PHTML statement  675
ODS Tagset statement  785
ODS TAGSETS.RTF statement  808
ENCODING= option
ODS MARKUP statement  500
ODS RTF statement  740
ENDCOLOR= style attribute  1017
EPUB  211
EPUB destination  35
EPUB2  213
EPUB3  233
escape characters
  for inline formatting  269
EVENT option
  ODS CHTML statement  180
  ODS CSVALL statement  198
  ODS EPUB statement  218
  ODS HTML statement  122, 392
  ODS HTML3 statement  364
  ODS PHTML statement  675
  ODS Tagset statement  785
EVENT_MAP tagset  797
EVENT= option
  ODS MARKUP statement  501
examples
  operating environments for  1113
  programs for  1083
EXCEL  292
CSS Stylesheet  317
style attribute  315
SVG Mode  313
EXCLUDE action
  ODS CSVALL statement  197
  ODS DOCUMENT statement  208
  ODS LISTING statement  485
  ODS MARKUP statement  491
  ODS PCL statement  566
  ODS PDF statement  577
  ODS PRINTER statement  693
  ODS PS statement  719
  ODS RTF statement  735
EXCLUDE option
  ODS CHTML statement  172
  ODS EPUB statement  215
  ODS EPUB3 statement  235
  ODS EXCEL statement  294
  ODS HTML statement  113, 382
  ODS HTML3 statement  355
  ODS HTML5 statement  427
  ODS PHTML statement  666
  ODS POWERPOINT statement  625
  ODS Tagset statement  774
  ODS TAGSETS.RTF statement  805
EXCLUDED option
  ODS TRACE statement  855
exclusion lists  39, 324, 762
destinations for output objects  38
OUTPUT destination  534
writing to log  771

F
FILE  59, 155
FILE option  219, 626
  ODS EPUB3 statement  241
  ODS EXCEL statement  297
  ODS TAGSETS.RTF statement  808
FILE PRINT ODS statement  55
restrictions  67, 163
syntax  59, 155
  without ODS suboptions  69, 165
FILE statement for ODS  59, 155
FILE= option
  ODS destination statements  14
  ODS HTML statement  115, 385
  ODS LISTING statement  486
  ODS MARKUP statement  494
  ODS PCL statement  569
  ODS PDF statement  583
  ODS PRINTER statement  699
  ODS PS statement  725
  ODS RTF statement  741
FILLPATTERN= style attribute  1017
FILLRULEWIDTH= style attribute  1018
FLYOVER= style attribute  1018
FONT_FACE= style attribute  1019
FONT= style attribute  1018
FONTFAMILY= style attribute  1019
FONTSIZE= style attribute  1019
FONTSTYLE= style attribute  1020
FONTWEIGHT= style attribute  1020
FONTWIDTH= style attribute  1021
footnotes
  in graphics output  503
  Printer output  700
  RTF output  741
FOREGROUNDCOLOR= style attribute  1013
FORMAT= attribute suboption
  FILE PRINT ODS statement  66, 162
  formats
    for columns  66, 162
  frame files  1109
FRAME option
  ODS CHTML statement  181
  ODS HTML statement  124, 393
  ODS HTML3 statement  365
  ODS HTML5 statement  435
  ODS PHTML statement  676
  ODS Tagset statement  786
FRAME= option
ODS MARKUP statement 502
FRAME= style attribute 1021
FRAMEBORDER= style attribute 1022
FRAMEBORDERWIDTH= style attribute 1022
FRAMESpacing= style attribute 1022
FREQ procedure

ODS table names 1049

G
GENERIC= attribute suboption
FILE PRINT ODS statement 67, 163
GENERIC= suboption
FILE PRINT ODS statement 62, 158
GFOOTNOTE option
ODS CHTML statement 182
ODS CSVALL statement 199
ODS EPUB statement 220
ODS EPUB3 statement 241
ODS Excel statement 298
ODS HTML statement 125, 395
ODS HTML3 statement 366
ODS HTML5 statement 437
ODS MARKUP statement 503
ODS PCL statement 570
ODS PDF statement 584
ODS PHTML statement 677
ODS POWERPOINT statement 626
ODS PS statement 726
ODS Tagset statement 787
ODS TAGSETS.RTF statement 809
GFOOTNOTE= option
ODS PRINTER statement 700
ODS RTF statement 741
global statements
category descriptions 106
ODS 103

GPATH option
ODS CHTML statement 183
ODS HTML statement 126, 395
ODS HTML3 statement 367
ODS HTML5 statement 437
ODS PHTML statement 678
ODS Tagset statement 788
GPATH= option
ODS LISTING statement 487
ODS MARKUP statement 504
GRADIENT_DIRECTION= style attribute 1022
graph segments (GRSEGs)
copying to catalogs 209
graphics
ODS RTF statement and 748
ODS TAGSET.RTF statement and 822
smoothing 330
template-based 328
graphics options
enabling for ODS 861, 862
ODS settings 860
graphics output
footnotes in 503
location for 487, 504
titles in 126, 184, 200, 368, 396, 438, 505, 679, 789
gredded layout statement 479, 899
GROUPMAX= option
ODS GRAPHICS statement 334
GRSEGs
copying to catalogs 209
GTITLE option
ODS CHTML statement 184
ODS CSVALL statement 200
ODS EPUB statement 220
ODS EPUB3 statement 242
ODS EXCEL statement 298
ODS HTML statement 126, 396
ODS HTML3 statement 368
ODS HTML5 statement 438
ODS MARKUP statement 505
ODS PCL statement 570
ODS PDF statement 584
ODS PHTML statement 679
ODS POWERPOINT statement 627
ODS PS statement 726
ODS Tagset statement 789
ODS TAGSETS.RTF statement 809
GTITLE= option
ODS PRINTER statement 700
ODS RTF statement 741

H
HEAD tags 127, 184, 368, 396, 438, 505, 679, 789
HEADTEXT option
ODS CHTML statement 184
ODS HTML statement 127, 396
ODS HTML3 statement 368
ODS HTML5 statement 438
ODS PHTML statement 679
ODS Tagset statement 789
HEADTEXT= option
ODS MARKUP statement 505
HEIGHT= option
ODS GRAPHICS statement 334
HEIGHT= style attribute 1022
HOST option
ODS PRINTER statement 700
HREFTARGET= style attribute 1023
HTML destination 35
body files 1106
contents file 1109
files produced by 1106
frame files 1109
links produced by 1101
output for 711
page files 1109
references produced by 1101

HTML files
appending to 146, 416
location of 132, 401
HTML links 1101
definition 1101
implementing 1101
ODS construction of 1103

HTML output
3.2 352
applet for viewing 114, 384
base text 115, 385
character set for META declaration 117, 176, 359, 387, 431, 495, 670, 780
creating 14
inserting text 136, 405
record separator 133, 402
sample 8
separate body file per page of output 142, 412
simple form 663
HTML references 1101
definition 1101
implementing 1101
ODS construction of 1103
HTML tagset 110, 379
HTML version setting 44
HTML3 destination 352
HTML4 tagset 776
HTML5 425
SVG Mode 453
HTMLCLASS= style attribute 1013
HTMLCONENTTYPE= style attribute 1015
HTMLDOCTYPE= style attribute 1017
HTMLID= style attribute 1024
HTMLSTYLE= style attribute 1024

I
ID option
ODS CHTML statement 185
ODS CSVALL statement 200
ODS EPUB statement 220
ODS EPUB3 statement 242
ODS EXCEL statement 299
ODS HTML statement 127, 397
ODS HTML3 statement 368
ODS HTML5 statement 439
ODS PHTML statement 680
ODS Tagset statement 789
ODS TAGSETS.RTF statement 809
ID= option
ODS MARKUP statement 505
ODS PCL statement 571
ODS PDF statement 585
ODS PRINTER statement 701
ODS PS statement 726
ODS RTF statement 742
image filenames 336
image files
resetting index counter 339
image format 338
IMAGE function 277
IMAGE_DPI option
ODS EPUB statement 221
ODS EPUB3 statement 242
ODS EXCEL statement 297, 298
ODS POWERPOINT statement 627
ODS TAGSETS.RTF statement 810
IMAGE_DPI= option
ODS LISTING statement 488
ODS MARKUP statement 506
ODS RTF statement 742
IMAGE= style attribute 1024
IMAGEMAP option
ODS GRAPHICS statement 335
IMAGEMAP= option
ODS GRAPHICS statement 335
IMAGENAME= option
ODS GRAPHICS statement 336
INDEX= style attribute 1034
index counter
resetting 339
inheritance
example programs 1093
style elements and 965
inline formatting 270, 273
escape characters for 269
nested 271
Unicode symbols 272
inline formatting functions 274
inline style attributes
nesting and 271
integrating output with applications 13
ITALIC function 278

J
Java devices
CODEBASE file path 118
JUST= style attribute 1033
justification
numeric columns 207
K
- KEEPN option
  - ODS RTF statement 742
- KEYWORDS option
  - ODS EXCEL statement 299
  - ODS POWERPOINT statement 627
  - ODS PDF statement 585
  - ODS PRINTER statement 701
  - ODS PS statement 727
  - KPISKIN= style attribute 1024

L
- LABEL= attribute suboption
  - FILE PRINT ODS statement 67, 163
  - FILE PRINT ODS statement 63, 159
- LABELMAX= option
  - ODS GRAPHICS statement 336
  - ODS GRAPHICS statement 336
- LABELPLACEMENT= option
  - ODS GRAPHICS statement 336
- LAYOUT ABSOLUTE statement
  - LAYOUT ABSOLUTE statement 454, 873
- LAYOUT destination
  - LAYOUT END 484, 884
  - LAYOUT END statement 484, 903
  - LAYOUT GRIDDED 468, 888
- LAYOUT GRIDDED statement
  - LAYOUT GRIDDED statement 468, 888
- LAYOUT option
  - ODS POWERPOINT statement 627
  - LEADERS function 278
  - LEFTMARGIN= style attribute 1026
  - LEGENDAREAMAX= option
    - ODS GRAPHICS statement 337
- LIBNAME 165
- LIBNAME statement for SASEDOC 165
- LIBNAME statement, SASEDOC 165
  - librefs
    - assigning to ODS documents 167
    - associating with output objects 165, 166
- line pointer controls
  - ODS 58, 867
  - LINKCOLOR= style attribute 1025
  - LINKHEIGHT= style attribute 1025
  - LINESTYLED= style attribute 1025
  - LINKTEXT= style attribute 1025
- LISTENTRYANCHOR= style attribute 1025
- LISTING destination 35
  - closing 485
  - excluding output objects 485
  - managing 484
  - opening 484
  - selecting output objects 485
  - writing selection/exclusion lists to log 485
  - writing trace records to 856
- LISTING option
  - ODS TRACE statement 856
- LISTING output
  - creating 15
  - LISTSTYLETITLE= style attribute 1025
  - LOESSMAXOBS= option
    - ODS GRAPHICS statement 337
    - LOESSOBSMAX= option
      - ODS GRAPHICS statement 337
  - log
    - output object records 855
    - writing selection/exclusion lists to 771
- M
  - MARGINBOTTOM= style attribute 1026
  - MARGINLEFT= style attribute 1026
  - MARGINRIGHT= style attribute 1027
  - MARGINTOP= style attribute 1027
  - MARKERSIZE= style attribute 1027
  - MARKERSYMBOL= style attribute 1027
  - MARKUP destination
    - closing 488, 491, 517
    - excluding output objects 491
    - opening 517
    - selecting output objects 491
    - markup files
      - location of 510
    - markup languages 488
  - MATCH_ALL option
  - MathML
    - ODS EPUB3 statement 249
    - MATHML function 279
    - MAXLEGENDAREA= option
      - ODS GRAPHICS statement 337
  - MEANS procedure
    - ODS table names 1049
    - MEAS_EVENT_MAP tagsets.rtf 820
    - MEAS_SHORT_MAP tagsets.rtf 820
    - MEAS_TEXT_MAP tagsets.rtf 821
    - META declaration
      - character set for 117, 176, 359, 387, 431, 495, 670, 780
META tags  128, 185, 369, 397, 439, 506, 680, 790
metadata  743, 810
author  578, 694, 721, 736, 806
string of keywords  585, 701, 727
subject  589, 707
title  589, 707, 747
METATEXT option
  ODS CHTML statement  185
  ODS HTML statement  128, 397
  ODS HTML3 statement  369
  ODS HTML5 statement  439
  ODS PHTML statement  680
  ODS Tagset statement  790
METATEXT= option
  ODS MARKUP statement  506

N
N= option
  FILE PRINT ODS statement  59, 155
NAME= option
  ODS DOCUMENT statement  210
NAMED_DEST= option
  ODS PDF statement  578
  ODS PRINTER statement  693
  ODS PS statement  720
NAMEDHTML tagset  798
NAMEVALUE== option
  ODS PACKAGE statement  558
NBSPACE function  279
nested inline formatting  271
nesting
  inline style attributes and  271
NEUTRALCOLOR= style attribute  1028
NEWCHAPTER option  221
  ODS EPUB3 statement  242
NEWFILE option
  ODS CHTML statement  185
  ODS CSVALL statement  201
  ODS HTML statement  128, 398
  ODS HTML3 statement  369
  ODS HTML5 statement  440
  ODS PHTML statement  680
  ODS Tagset statement  790
NEWFILE= option
  ODS MARKUP statement  506
  ODS PCL statement  571
  ODS PDF statement  585
  ODS PRINTER statement  701
  ODS PS statement  727
  ODS RTF statement  742
  ODS TAGSETS-RTF statement  810
NEWLINE function  279
NOANTIALIAS option
  ODS GRAPHICS statement  330
NOBORDER option
  ODS GRAPHICS statement  332
NOBREAKSPACE= style attribute  1028
NOGFOOTNOTE option
  ODS Excel statement  298
NOGTITLE option
  ODS Excel statement  298
NOIMAGEMAP option
  ODS GRAPHICS statement  335
NONE argument
  ODS EPUB statement  221
NONE option
  ODS EPUB3 statement  242
  ODS EXCLUDE statement  322
NOSCALE option
  ODS GRAPHICS statement  341
NOSCALEMARKERS option
  ODS GRAPHICS statement  342
NOSUBPIXEL option
  ODS GRAPHICS statement  344
NOTEREF function  280
NOTOC option
  ODS PDF statement  586
  ODS PRINTER statement  702
NOW argument
  ODS EPUB statement  221
NOW option
  ODS EPUB3 statement  243
NOWARN option
  ODS EXCLUDE statement  323
numeric columns
  justification of  207

O
OBJECT= suboption
  FILE PRINT ODS statement  63, 159
OBJECTLABEL= suboption
  FILE PRINT ODS statement  64, 160
OD NOVERIFY
  ODS VERIFY statement  864
ODS _ALL_ CLOSE statement  169
ODS (Output Delivery System)  3
  DATA step and  53
  DATA step examples  69
  how it works  29
  multiple output formats  16
  processing  29
  quick start  13
  registry and  40
  reports with DATA step  54
  samples  4
ODS argument
  FILE PRINT ODS statement  59, 155
ODS block  13
ODS CHTML statement  170
actions 172
options 173
ODS column pointer controls 56, 866
ODS CSVALL statement 195
actions 196
options 197
ODS DECIMAL_ALIGNMENT statement 207
ODS destinations
categories of 33
changing default settings 45, 49
changing default value 46, 48
closing 169
default devices 486, 499, 739
destination-independent input 34
excluding output objects 321
exclusion lists 39, 324, 762
SAS formatted destinations 34
selecting output objects for 758
selection lists 39, 324, 762
specifying multiple 517
system resources and 37
tagset keywords as 517
tagset names as 525
third-party formatted destinations 35
two-level tagset names as 518
ODS DOCUMENT statement 208
ODS documents
librefs for 167
ODS EPUB statement 211
ODS EPUB2 statement 213
ODS EPUB3 statement 233
ODS ESCAPECHAR statement 269
ODS ESCAPECHAR= statement
inline formatting functions for 274
ODS EXCEL statement 292
ODS EXCLUDE statement 321
ODS GRAPHICS statement 328
for batch jobs 346
ODS HTML statement 110, 379
ODS HTML3 statement 352
actions 355
options 355
ODS HTML5 statement 425
HTML5 425
ODS LAYOUT 871
ODS LAYOUT overview 873
ODS line pointer controls 58, 867
ODS LISTING statement 484
actions 485
default devices 486
options 485
ODS MARKUP statement 488
actions 488, 491
creating XML files 518
creating XML files and DTD 521
default devices 499
details 517
examples 488
multiple markup output 204, 523
options 492
tagset names as ODS destinations 525
ODS NODECIMAL_ALIGNMENT statement 533
ODS output
See also SAS Studio
assigning attributes to columns 78
DATA step enhanced features 55
formatting variables 59, 155
in SAS Studio 93
inserting text into 847
listing variables to include 517
multiple formats 208
selected variables in 73
tracking in Results window 731
ODS OUTPUT statement 534
actions 534
arguments 534
creating data sets 538, 542, 545
evaluations 534
merging output objects into data set 537
ODS PACKAGE statement 554
options 557
ODS PATH statement 563
ODS PCL statement 565
multiple instances of same destination 571
options 567
ODS PDF statement 575
multiple instances of same destination 585
opening multiple instances of same
destination 594
opening/closing PDF destination 590
options 578
ODS PHTML statement 663
actions 666
options 666
ODS POWERPOINT statement 623
ODS PREFERENCES statement 663
ODS PRINTER statement 690
actions 693
details 708
host information 710
multiple instances of same destination 710
opening/closing PRINTER destination 708
options 693
output for HTML destination 711
output for PRINTER destination 711
printing output directly to printers 708
Windows and 709
without actions or options 692
ODS PROCLABEL statement 716
ODS PROCTITLE statement 716
ODS PS statement 717
multiplied instances of same destination 726
options 720
ODS PUT statement 55, 864
ODS Region
absolute layout 465, 884
ODS Region statement, absolute 465, 884
ODS RESULTS statement 731
ODS RTF statement 732
actions 734
default devices 739
graphics and 748
opening/closing RTF destination 748
options 735
RTF output 748
ODS SELECT statement 758
ODS SHOW statement 771
ODS statements
category descriptions 106
definition of 103
global statements 103
Output Control statements 104
procedure statements 104
SAS formatted statements 104
third-party formatted statements 104
ODS table names
ARIMA procedure 1064
Base SAS procedures 1045
CALENDAR procedure 1045
CATALOG procedure 1045
CHART procedure 1046
COMPARE procedure 1046
CONTENTS procedure 1048
CORR procedure 1047
DATASETS procedure 1048
FREQ procedure 1049
MEANS procedure 1049
PLOT procedure 1056
REPORT procedure 1056
SAS/ETS procedures 1064
SQL procedure 1056
SUMMARY procedure 1049
TABULATE procedure 1057
TIMEPLOT procedure 1057
UNIVARIATE procedure 1057
ODS Tagset statement
actions 774
arguments 775
options 776
ODS TAGSET statement 772
ODS TAGSET.RTF statement
graphics and 822
ODS TAGSETS.RTF 819
ODS TAGSETS.RTF statement 803
ODS TAGSETS.RTF suboptions 811
ODS TEXT= statement 847
ODS TRACE statement 854
contents of trace record 856
equation 857
LABEL= option 535
ODS USEGOPT statement 860
ODS VERIFY statement 863
ODSDEST= system option
restoring defaults 907
ODSGRAPHICS= system option
restoring defaults 908
ODSTYLE= system option
restoring defaults 909
OFF option
ODS TRACE statement 855
ODS VERIFY statement 864
ON option
ODS TRACE statement 855
ODS VERIFY statement 864
OPERATOR option
ODS TAGSETS.RTF statement 810
OPERATOR= option
ODS RTF statement 743
OPTIONS option
ODS CHTML statement 173
ODS CSV ALL statement 197
ODS EPUB3 statement 243
ODS EXCEL statement 299
ODS HTML3 statement 355
ODS LISTING statement 485
ODS MARKUP statement 491, 492
ODS PHTML statement 666
ODS RTF statement 735
OPTIONS options
ODS CHTML statement 186
ODS CSV ALL statement 201
ODS HTML3 statement 129, 399
ODS HTML3 statement 370
ODS MARKUP statement 507
ODS PHTML statement 681
ODS POWERPOINT statement 628
ODS Tagset statement 791
ODS TAGSETS.RTF statement 811
OPTIONS options
ODS EPUB statement 221
output
integrate with applications 13
OUTPUT argument
ODS EPUB statement 221
ODS RTF statement 743
OUTPUT destination 35
Index 1145

closing 534
exclusion lists 534
selection lists 534
output objects
creating 59, 155
customized output for 39
data sets from 534, 542
determining destinations for 38, 534
excluding from LISTING destination 485
excluding from ODS destinations 321
hierarchy of 208
labels for 64, 160
librefs 165
merging dissimilar objects into data set 537
names for 63, 159
records in log 855
selecting for LISTING destination 485
selecting for ODS destinations 758
sequence number of 166
specifying 856
tracing 855
OUTPUT option
ODS EPUB3 statement 243
ODS TRACE statement 855
output pointer
number of lines for 59, 155
output strings
interpreting 273
OUTPUTFMT= option
ODS GRAPHICS statement 338
OUTPUTHEIGHT= style attribute 1022, 1028
OUTPUTWIDTH= style attribute 1028, 1037
overflow-control option
FILE PRINT ODS statement 59, 155
OVERHANGFACTOR= style attribute 1028

P
package objects
adding to 554
closing 554
opening 554
publishing 554
PACKAGE option
MARKUP statement 508
ODS CHTML statement 187
ODS CSV ALL statement 202
ODS HTML statement 130, 399
ODS HTML3 statement 373
ODS HTML5 statement 444
ODS MARKUP statement 509
ODS PHTML statement 683
ODS Tagset statement 793
PARAMETERS option
ODS CHTML statement 188
ODS CSV ALL statement 202
ODS HTML statement 132, 401
ODS HTML3 statement 373
ODS HTML5 statement 444
ODS MARKUP statement 509
ODS PHTML statement 684
ODS Tagset statement 793
PCL 565

ODS PRINTER statement 702
ODS RTF statement 743
ODS TAGSETS.RTF statement 816
PADDING= style attribute 1029
PADDINGBOTTOM= style attribute 1029
PADDINGLEFT= style attribute 1029
PADDINGRIGHT= style attribute 1029
PADDINGTOP= style attribute 1029
PAGE argument
ODS EPUB statement 221
page breaks 572, 587, 705, 729
RTF output 745
splitting tables at 742
page files 1109
PAGE option
ODS CHTML statement 187
ODS EPUB3 statement 243
ODS HTML statement 130, 399
ODS HTML3 statement 371
ODS HTML5 statement 443
ODS LISTING statement 485
ODS MARKUP statement 508
ODS PHTML statement 682
ODS Tagset statement 791
PAGEBREAKHTML= style attribute 1029
PAGEOF function 280
PAGEPANELS option
ODS TAGSETS.RTF statement 816
PARAMETERS= option
ODS CHTML statement 188
ODS CSV ALL statement 202
ODS HTML statement 132, 401
ODS HTML3 statement 373
ODS HTML5 statement 444
ODS MARKUP statement 509
ODS PHTML statement 684
ODS Tagset statement 793
PCL 565
PCL destination
  - closing 574
  - opening 574
PCL option
  - ODS PRINTER statement 702
PCL output 702
PDF 575
PDF destination
  - closing 590
  - opening 590
  - opening multiple instances 594
PDF files
  - adding notes 587, 703, 728
  - compressing 581, 697
  - list of bookmarks 579, 580, 695, 721, 722
PDF function 280
PDF option
  - ODS PRINTER statement 703
PDF output 703
  - inserting text 589
  - sample 9
PDFMARK option
  - ODS PRINTER statement 703
  - ODS PS statement 728
PDFNOTE option
  - ODS PDF statement 587
  - ODS PRINTER statement 703
  - ODS PS statement 728
PDFTOC= option
  - ODS PDF statement 587
  - ODS PRINTER statement 704
PHTML destination 663
PHTML output 663
PHTML tagset 776
PLOT procedure
  - ODS table names 1056
  - pointers
    - past end of line 58, 867
POP option
  - about 351
    - ODS GRAPHICS statement 338
POSTHTML= style attribute 1030
POSTIMAGE= style attribute 1030
PostScript files
  - tags for Acrobat Distiller 703, 728
PostScript output 705
  - sample 10
POSTSCRIPT option
  - ODS PRINTER statement 705
POSTTEXT= style attribute 1030
PowerPoint 623
PowerPoint destination 36
PREFERENCES destination 663
PREHTML= style attribute 1030
PREIMAGE= style attribute 1031
PREPAGE= option
  - ODS RTF statement 744
  - ODS TAGSETS.RTF statement 817
PREPEND option
  - ODS PATH statement 564
PRETEXT= style attribute 1031
PRINT argument
  - FILE PRINT ODS statement 59, 155
PRINTER 690
PRINTER destination 36
  - closing 693, 708
  - excluding output objects 693
  - opening 708
  - output for 711
  - selecting output objects 566, 577, 693, 720
  - writing selection/exclusion lists to log 693
printer drivers
  - ODS PRINTER statement 700
Printer output
  - footnotes 700
  - titles 700
PRINTER output
  - inserting text 707
PRINTER= option
  - ODS PRINTER statement 704
PROC argument
  - ODS EPUB statement 221
PROC option
  - ODS EPUB3 statement 243
procedure statements 104
  - procedures
    - creating data sets from output objects 542
    - ODS table names, Base SAS 1045
    - ODS table names, SAS/ETS 1064
    - style templates with 919
    - title in output 716
PROTECTSPECIALCHARS= style attribute 1031
PRT= option
  - ODS PRINTER statement 704
PS 717
PS destination
  - closing 731
  - opening 731
PS option
  - ODS PRINTER statement 705
PS output
  - inserting text 730
PUSH option
  - about 351
    - ODS GRAPHICS statement 338
PUT 55, 864
PUT statement
Index 1147

**R**
- RAW function 281
- RECORD_SEPARATOR option
  - ODS CHTML statement 190
  - ODS CSVALL statement 203
  - ODS HTML3 statement 373
  - ODS HTML5 statement 446
  - ODS PHTML statement 685
  - ODS Tagset statement 794
  - ODS TAGSETS.RTF statement 817
- RECORD_SEPARATOR= option
  - ODS HTML statement 133, 402
  - ODS MARKUP statement 511
  - ODS RTF statement 744
- RECSEP= option
  - ODS HTML statement 133, 402
  - ODS MARKUP statement 511
- references
  - See HTML references
- REGION statement 479, 899
- registry
  - changing default HTML version setting 44
  - changing ODS destination default settings 45
  - changing ODS EPUB Encoding settings 49
  - changing the default EPUB version 48
  - changing the default printer value 46
- REMOVE option
  - ODS PATH statement 565
- REPORT procedure
  - ODS table names 1056
  - RESEP= option
  - ODS RTF statement 744
- RESET option
  - ODS GRAPHICS statement 339
  - RESET= option
  - ODS GRAPHICS statement 339
- Results window
  - tracking ODS output 731
- RIGHTMARGIN= style attribute 1027
- RS= option
  - ODS HTML statement 133, 402
  - ODS MARKUP statement 511
  - ODS RTF statement 744
- RTF destination 37, 732
  - closing 734, 748, 803
  - excluding output objects 735
  - managing 803
  - opening 748, 803
  - selecting output objects 735
- writing selection/exclusion lists to log 735
- RTF files
  - location of 743, 816
  - record separator 744
  - style definitions 746
  - time and date of SAS program 745
- RTF markup files
  - style definitions 819
- RTF output 732, 748, 821
  - footnotes 741
  - graphics 748
  - image resolution 742
  - inserting text 747
  - page breaks 745
  - sample 12
  - splitting tables at page breaks 742
  - titles 741
  - translation tables 747
  - RULES= style attribute 1032

**S**
- SAS Explorer window
  - list of available styles 801
  - SAS formatted destinations 33, 34
  - SAS formatted statements 104
  - SAS Graphics Editor (SGE)
    - ODS HTML statement 133, 403
    - ODS HTML5 statement 446
  - SAS Studio
    - about 93
    - customized output 96
    - default output 94
    - output environment 94
  - SAS/ETS procedures
    - ODS table names 1064
  - SASDATE option
    - ODS EXCEL statement 311
    - ODS POWERPOINT statement 635
    - ODS RTF statement 745
  - SASEDOC argument
    - LIBNAME statement 166
  - SASEDOC engine
    - LIBNAME statement with 165
  - SCALE option
    - ODS GRAPHICS statement 341
  - SCALE= option
    - ODS GRAPHICS statement 341
  - SCALEMARKERS= option
    - ODS GRAPHICS statement 342
  - SELECT action
    - ODS CSVALL statement 197
    - ODS DOCUMENT statement 209
    - ODS LISTING statement 485
    - ODS MARKUP statement 491
ODS PCL statement 566
ODS PDF statement 577
ODS PRINTER statement 693
ODS PS statement 720
ODS RTF statement 735

SELECT option
ODS CHTML statement 172
ODS EPUB statement 215
ODS EPUB3 statement 235
ODS EXCEL statement 295
ODS HTML statement 113, 382
ODS HTML3 statement 355
ODS HTML5 statement 428
ODS PHTML statement 666
ODS POWERPOINT statement 625
ODS Tagset statement 774
ODS TAGSETS.RTF statement 805

destinations for output objects 38
multiple procedure steps with 762
OUTPUT destination 534
writing to log 771

SGE
ODS HTML statement 133, 403
ODS HTML5 statement 446

SGE= option
ODS LISTING statement 488

SHORT_MAP tagset 798

SHOW action
ODS CSVALL statement 197
ODS DOCUMENT statement 209
ODS LISTING statement 485
ODS MARKUP statement 491
ODS OUTPUT statement 534
ODS PCL statement 567
ODS PDF statement 577
ODS PRINTER statement 693
ODS PS statement 720
ODS RTF statement 735

SHOW argument
ODS OUTPUT statement 537

SHOW option
ODS CHTML statement 172
ODS EPUB statement 216
ODS EPUB3 statement 236
ODS EXCEL statement 295
ODS GRAPHICS statement 342
ODS HTML statement 113, 382
ODS HTML3 statement 355
ODS HTML5 statement 428
ODS PHTML statement 666
ODS POWERPOINT statement 625
ODS Tagset statement 774
ODS TAGSETS.RTF statement 806

smoothing graphics 330
SQL procedure
ODS table names 1056
STACKDEPTHMAX= option
ODS GRAPHICS statement 343
STARTCOLOR= style attribute 1032
STARTPAGE option
ODS TAGSETS.RTF statement 818
STARTPAGE= option
ODS PCL statement 572
ODS PDF statement 587
ODS PRINTER statement 705
ODS PS statement 729
ODS RTF statement 745

statement 55, 864
EPUB 211
EPUB2 213
EPUB3 233
FILE 59, 155
GRIDDED LAYOUT 484, 903
LAYOUT ABSOLUTE 454, 484, 873, 903
LAYOUT GRIDDED 468, 888
LIBNAME 165
PCL 565
PDF 575
PRINTER 690
PS 717

statements
REGION 479, 899

STATIC option
ODS GRAPHICS statement 338

STATUS option
ODS EXCEL statement 311
ODS POWERPOINT statement 635

STRONG function 281

style attribute
tagattr 315

style attributes 37
color 1037
data values 993
definition 913
dimension 1039
font definition 1040
format 1041
reference 1041
table of 993
values of 993

style definitions
RTF files 746, 819

style elements
definition 913
inherances of 965

TABULATE procedure 913

STYLE function 281

STYLE option
ODS CHTML statement 190
ODS EXCEL statement 311
ODS HTML statement 134, 403
ODS HTML3 statement 374
ODS PHTML statement 685
ODS Tagset statement 795
ODS TAGSETS.RTF statement 819
style overrides
  ODS EPUB statement 224
style templates
definition of 913
  ODS EPUB3 statement 247
  ODS MARKUP statement 511
  ODS PCL statement 573
  ODS PDF statement 588
  ODS POWERPOINT statement 635
  ODS PRINTER statement 706
  ODS PS statement 730
  procedures with 919
  verifying values 863
  viewing contents of 923
STYLE_DISPLAY tagset 798
STYLE_POPUP tagset 798
STYLE= option
  ODS LAYOUT ABSOLUTE statement 455, 474, 875, 894
  ODS LAYOUT REGION, Absolute statement 466, 886
  ODS LAYOUT REGION, Gridded statement 483, 902
  ODS MARKUP statement 511
  ODS PCL statement 573
  ODS PDF statement 588
  ODS PRINTER statement 706
  ODS PS statement 730
  ODS RTF statement 746
STYLESHEET option
  ODS CHTML statement 191
  ODS EPUB statement 224
  ODS EPUB3 statement 247
  ODS HTML statement 134, 404
  ODS HTML3 statement 375
  ODS HTML5 statement 447
  ODS HPTML statement 686
  ODS Tagset statement 795
STYLESHEET= option
  ODS MARKUP statement 512
SUB function 282
SUBJECT= option
  ODS PDF statement 589
  ODS PRINTER statement 707
SUBPIXEL option
  ODS GRAPHICS statement 344
SUMMARY procedure
  ODS table names 1049
SUPER function 282
T
  TABLE argument
    ODS EPUB statement 221
tables
  table of contents
    ODS PDF statement 582, 586
    ODS PRINTER statement 697, 702
  table templates
    binding data components to 59, 155
    specifying 64, 160
    user-defined templates 85
    verifying values 863
  TABLEROWS option
    ODS TAGSETS.RTF statement 819
tag attributes
  for dynamic graphics 493
  TAGATTR= style attribute 1033
  TAGSET.RTF output graphics 822
  TAGSET= option
    ODS MARKUP statement 513
tagsets
  36
    CHTML 775
    CSV 775
    CSVALL 775
    CSVBYLINE 775
    EVENT_MAP 797
    HTML 110, 379
    HTML4 776
  keyword values for 513
  keywords as ODS destinations 517
  list of 30
  NAMEDHTML 798
  names as ODS destinations 525
  PHTML 776
  SHORT_MAP 798
  statement 772
  STYLE_DISPLAY 798
  STYLE_POPUP 798
  TEXT_MAP 798
two-level names as ODS destinations 518
  user-defined 776
  XML 775
tagsets.rtf
  MEAS_EVENT_MAP 820
  MEAS_SHORT_MAP 820
MEAS_TEXT_MAP 821
TEMPLATE procedure
  list of available styles 801
  locations for definitions 563
  search order for definitions 563
  user-defined table template template 85
TEMPLATE= suboption
  FILE PRINT ODS statement 64, 160
text
  inserting into ODS output 847
TEXT option
  ODS CHTML statement 192
  ODS EXCEL statement 311
  ODS HTML statement 136, 405
  ODS HTML3 statement 376
  ODS HTML5 statement 449
  ODS MARKUP statement 514
  ODS PHTML statement 687
  ODS Tagset statement 797
  ODS TAGSETS.RTF statement 820
TEXT_MAP tagset 798
TEXT= option
  ODS MARKUP statement 501
  ODS PCL statement 573
  ODS PDF statement 589
  ODS PRINTER statement 707
  ODS PS statement 730
  ODS RTF statement 747
TEXTALIGN= style attribute 1034
TEXTDECORATION= style attribute 1034
TEXTINDENT= style attribute 1034
TEXTJUSTIFY= style attribute 1035
third-party formatted destinations 35
definition 33
  formatting control and 37
third-party formatted statements 104
THISPAGE function 283
TICKDISPLAY= style attribute 1035
TIMEPLOT procedure
  ODS table names 1057
TIPMAX= option
  ODS GRAPHICS statement 345
TITLE option
  ODS EPUB statement 225
  ODS EPUB3 statement 247
  ODS EXCEL statement 312
  ODS POWERPOINT statement 635
TITLE= option
  ODS PDF statement 589
  ODS PRINTER statement 707
  ODS RTF statement 747
titles
  in file metadata 589, 707, 747
  in graphics output 126, 184, 200, 368, 396, 438, 505, 679, 789
  Printer output 700
  procedure titles in output 716
  RTF output 741
  TOC_DATA= option
    ODS RTF statement 747
  TOCENTRYINDENT function 283
  TOCENTRYPAGE function 283
  TOPMARGIN= style attribute 1027
  trace records 855, 856
  trailing @
    PUT _ODS_ statement 56, 865
translation tables
  ODS MARKUP statement 514
  RTF output 747
TRANSPARENCY= style attribute 1035
TRANTAB option
  ODS CHTML statement 193, 204
  ODS HTML statement 136, 405
  ODS HTML3 statement 376
  ODS HTML5 statement 449
  ODS PHTML statement 688
  ODS Tagset statement 797
  ODS TAGSETS.RTF statement 820
TRANTAB= option
  ODS MARKUP statement 514
  ODS RTF statement 747
TYPE= option
  ODS MARKUP statement 513
U
UNICODE function 283
Unicode symbols 272
UNIFORM option
  ODS PCL statement 574
  ODS PDF statement 589
  ODS PRINTER statement 707
  ODS PS statement 730
  ODS TAGSETS.RTF statement 820
UNIVARIATE procedure
  ODS table names 1057
UNIX
  printing output directly to printer 709
URL= option
  ODS LISTING statement 487
  URL= style attribute 1035
user-defined tagsets 776
V
VARIABLES= suboption
  FILE PRINT ODS statement 65, 161
VERTICALALIGN= style attribute 1035
VIDEO function 284
VISITEDLINKCOLOR= style attribute 1036
VJUST= style attribute 1035

W
WATERMARK= style attribute 1036
WHERE option
  ODS EXCLUDE statement 323
WHITESPACE= style attribute 1036
WIDTH= option
  ODS GRAPHICS statement 345
WIDTH= style attribute 1037
Windows
  ODS PRINTER statement with 709
    printing output directly to printer 709
WORK option 635

ODS EPUB 225
ODS EPUB3 247
ODS EXCEL statement 312

X
XML files
  creating 518
    creating, with DTD 521
XML tagset 775

Z
z/OS
  printing output directly to printer 709