Contents

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

PART 1 Introduction 1

Chapter 1 • Getting Started with the Output Delivery System . . . 3
Welcome to the Output Delivery System ................................. 4
Gallery of ODS Samples ................................................... 4
A Quick Start to Using ODS ............................................... 13
Creating Accessible Output with the Output Delivery System ....... 20
Working with Output Defaults ............................................ 23
Where to Go from Here .................................................... 27

PART 2 Concepts 29

Chapter 2 • Output Delivery System: Basic Concepts ............... 31
Overview of How ODS Works ............................................. 31
Understanding Item Stores and Template Stores ..................... 34
Understanding ODS Destinations ........................................ 36
How ODS Determines the Destinations for an Output Object ....... 41
Selection and Exclusion Lists ............................................. 42
Customized Output for an Output Object ................................ 43
Customizing Titles and Footnotes .......................................... 44
Changing SAS Registry Settings for ODS ............................. 44

PART 3 Output Delivery System and the DATA Step 55

Chapter 3 • Using ODS with the DATA Step .......................... 57
Introduction .................................................................. 57
How ODS Works with the DATA Step .................................. 58
Syntax for ODS Enhanced Features in a DATA Step ................. 59
Dictionary .................................................................. 59
Examples .................................................................... 73

PART 4 SAS Studio and ODS 95

Chapter 4 • SAS Studio and ODS ......................................... 97
About SAS Studio ......................................................... 97
The SAS Studio Output Environment .................................... 98
Appendix 1 • Output Object Table Names ................................................................. 1141
  ODS Table Names Produced by Base SAS Procedures ...................................... 1141
  ODS Table Names Produced by Base SAS High-Performance Procedures .......... 1155
  ODS Table Names Produced by Base SAS Statistical Procedures ..................... 1155
  ODS Table Names Produced by SAS/STAT Procedures .................................. 1156
  ODS Table Names Produced by SAS/STAT High-Performance Procedures ........... 1160
  ODS Table Names Produced by SAS Enterprise Miner High-Performance Procedures .... 1161
  ODS Table Names Produced by SAS/ETS Procedures .................................. 1161
  ODS Table Names Produced by SAS/ETS High-Performance Procedures .......... 1163
  ODS Table Names Produced by SAS/QC Procedures ..................................... 1163
  ODS Table Names Produced by SAS/OR Procedures .................................. 1164
  ODS Table Names Produced by Graph Algorithms and Network Analysis Procedure .... 1165

Appendix 2 • Example Programs ............................................................................. 1167
  Examples from the Gallery of ODS Samples .................................................. 1168
  Creating the SENTRY Format ........................................................................ 1179
  Creating the Charity Data Set ........................................................................ 1179
  Creating the DIVFMT. and USETYPE. Formats .............................................. 1182
  Creating the DistrData Data Set .................................................................... 1182
  Creating the Univ ODS Document ................................................................ 1182
  Creating the Employee_Data Data Set .......................................................... 1183
  Creating the Energy Data Set ........................................................................ 1185
  Creating the Exprev Data Set ........................................................................ 1185
  Creating the Gov Data Set ............................................................................ 1186
  Creating the Grain_Production Data Set ........................................................ 1187
  Creating the One Data Set ............................................................................ 1187
  Creating the Iron Data Set ............................................................................ 1188
  Creating the Model Data Set .......................................................................... 1188
  Creating the Neuralgia Data Set ................................................................... 1189
  Creating the Plants Data Set .......................................................................... 1190
  Creating the Plant_Stats Data Set ................................................................. 1190
  Creating the StatePop Data Set .................................................................... 1190
  Creating the Stats and Stats2 Data Sets ......................................................... 1192
  Creating the Table1 Table Template ............................................................. 1192
  Programs That Illustrate Inheritance ............................................................. 1193

Appendix 3 • ODS and the HTML Destination ......................................................... 1201
  HTML Links and References Produced by the HTML Destination ................. 1201
  Files Produced by the HTML Destination ...................................................... 1206

Appendix 4 • ODS HTML Statements for Running Examples in Different Operating Environments ................................................................. 1213
  Using a z/OS UNIX System Services HFS Directory for HTML Output ............. 1213
  Using a z/OS PDSE for EBCDIC HTML Output ............................................. 1213
  Using a z/OS PDSE for ASCII HTML Output ................................................ 1214

Appendix 5 • PowerPoint Transition Options and the Effect Options Supported .... 1215
  PowerPoint Transition Options and the Effect Options Supported ................. 0
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:

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

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

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

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

MISSING character(s);

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

The BY statement is an example of this argument:

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

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

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

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

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

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

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

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

Here are examples of this form of multiple arguments:

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

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

Style Conventions

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

```
ERROR <message>;
```

UPPERCASE
identifies arguments that are literals.

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

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

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

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

  ```
  LINK label;
  ```

- nonliteral values that are assigned to an argument.

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

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

**Special Characters**

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

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

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

  ```
  MAPS=location-of-maps
  ```

- `< >` angle brackets identify optional arguments. A required argument is not enclosed in angle brackets.

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

  ```
  CAT (item-1 <, item-2, \ldots>)
  ```

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

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

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

- `\ldots` 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:

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

'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:

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

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

In this example, each statement ends with a semicolon:

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

**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:

\[
\begin{align*}
\text{infile } &\text{file-specification obs = 100;} \\
\text{libname } &\text{libref }'\text{SAS-library'}; \\
\end{align*}
\]
About This Book
What's New in the Output Delivery System

Overview

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

- In SAS 9.4M5, 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 (RWI) is now in production.
- In SAS 9.4M1, 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 Default to 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 SAS 9.4M1, new statement ODS EPUB3 was added. In SAS 9.4M3, the ODS EXCEL statement was added.
- In SAS 9.4M3, HTML5 supports video and audio when using the Report Writing Interface.
- New options have been added to ODS HTML and HTML5.
- Enhancements have been made to the documentation.
- ODS destinations support accessibility as follows:
  - Starting in SAS 9.4M4, the ODS HTML5 destination supports accessibility. The ACCESSIBLE_GRAPH option adds accessibility metadata to graphs that are created by ODS Graphics.
  - In SAS 9.4M5, the ODS HTML5 statement supports a new style element to enable you to control the appearance of the link-focus indicator in ODS output.
  - In SAS 9.4M5, image maps are supported with SVG output in the ODS HTML5 statement.
  - In SAS 9.4M5 the ODS PDF destination supports accessibility when you specify the new option ACCESSIBLE.
  - In SAS 9.4M5, new options have been added to the ODS EXCEL statement.
• In SAS 9.4M5, new options have been added to the ODS RTF statement.

• In SAS 9.4M5, the Encore style has been added. This style is used with Avenir Next for SAS fonts in some versions of SAS Studio. In SAS 9.4M6, the default style for HTML 5.0 output has changed from HTMLEncore to HTMLEncore.

• In SAS 9.4M6 the ODS destination for Word has been added. This destination creates Microsoft Word documents. The ODS destination for Word enables users to customize different aspects of the document, such as themes, tables of contents, and orientation.

Enhancements to the DOCUMENT Procedure

The new 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. For more information, see “STORE=template-store” in SAS Output Delivery System: Procedures Guide.

In SAS 9.4M5, the REPLAY statement has two new options:

• The new ANCESTORS= option specifies the number of parent directories to replay. For more information, see “ANCESTORS= n | ALL” in SAS Output Delivery System: Procedures Guide.

• The new DYNAMDATA= option specifies the name of the data set that contains dynamic values. For more information, see “DYNAMDATA=data-set-name” in SAS Output Delivery System: Procedures Guide.

New ODS Procedures

The following procedures are new of SAS 9.4:

• The ODSLIST procedure enables you to create lists that can be customized. For more information, see “ODSLIST Procedure” in SAS Output Delivery System: Procedures Guide.

• 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. For more information, see “ODSTABLE Procedure” in SAS Output Delivery System: Procedures Guide.

• The ODSTEXT procedure enables you to create lists and paragraphs for your output. In SAS 9.4M4, the new HEADING statement has been added to the ODSTEXT procedure. For more information, see “ODSTEXT Procedure” in SAS Output Delivery System: Procedures Guide.
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 SAS 9.4M3, the RWI has the following two new methods that support the ability to embed audio and video into HTML 5.0 output:

- AUDIO
- VIDEO

In SAS 9.4M4, the DESCRIPTION argument adds descriptive text to images and areas of a report. The DESCRIPTION argument has been added to the following methods:

- IMAGE
- LAYOUT_ABSOLUTE
- LAYOUT_GRIDDED
- REGION (absolute)
- REGION (gridded)
- TABLE_START

In SAS 9.4M5, the new TYPE argument has been added to the FORMAT_CELL method. The TYPE argument specifies that the cell class is a row header. This option helps make reports accessible for people with a wide range of abilities. For more information, see “TYPE:'H'" in SAS Output Delivery System: Advanced Topics.

In SAS 9.4M6, the new CAPTION argument has been added to the TABLE_START method. The CAPTION argument specifies a text string to add before each table. For more information, see “CAPTION: 'string'" in SAS Output Delivery System: Advanced Topics.

ODS and Cascading Style Sheets

In SAS 9.4M1, you can apply cascading style sheets to ODS output destinations HTML, PDF, and RTF. For more information, see “Introduction to Cascading Style Sheets” in SAS Output Delivery System: Advanced Topics.

New Default Styles for ODS Destinations

The default style for all printer family destinations has changed from Printer to Pearl. The following PDF output uses the new default style, Pearl:
The default HTML style for batch output on all hosts has changed from Default to HTMLBlue.

Beginning with SAS 9.4M6, the default style for HTML 5.0 output has changed from HTMLBlue to HTMLEncore.

For more information about changing these defaults, see “Working with Output Defaults” on page 23.

### ODS Statements Enhancements

In SAS 9.4M1, 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 SAS 9.4M1, 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 SAS 9.4M1, 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
- PRINTER family
- RTF

**ODS GRAPHICS Statement**

The ODS GRAPHICS statement now supports the following options:

- ATTRPRIORITY= on page 343
- BYLINE= on page 344
- DRILLTARGET= on page 345
- LOESSMAXOBS= on page 350

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 SAS 9.4M3, the ODS GRAPHICS statement supports the following options:

• PUSH and POP on page 352
• STACKDEPTHMAX= on page 357
• SHOW on page 356
• SUBPIXEL | NOSUBPIXEL on page 357

The following changes and enhancements were made to the ODS GRAPHICS statement in SAS 9.4M3:

• 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=.

Starting with SAS 9.4M4 the ODS GRAPHICS statement supports the following options:

• NBINSMAX= on page 350
• NXYBINSMAX= on page 350

Starting with SAS 9.4M5, the ODS GRAPHICS statement supports the following options:

• OBSMAX= on page 351
• TOTALCELLMAX= on page 359
• LINEPATTERNOBSMAX= on page 349

The following changes and enhancements were made to the ODS GRAPHICS statement in SAS 9.4M5:

• Vector graphics output can be generated for text plot back light, background fill, and background outline.

  For a table showing the cases for which vector graphics output can be generated, see Table 6.13 on page 364.
• Image maps are supported with vector graphics in HTML5. Image maps are enabled by specifying the IMAGEMAP option in the ODS GRAPHICS statement.
• If a BY statement is in effect for the data, then you can uniquely name image files based on BY iterations by inserting #BY substitution items into the image filename. You can specify BY variable values, BY variable names, or BY lines for the IMAGENAME= option in the ODS GRAPHICS statement.
ODS PDF Statement

The ODS PDF statement supports the following accessibility features.

- In SAS 9.4M4, the ODS PDF statement supported the option ACCESSIBLE as a preproduction feature. In SAS 9.4M5, the ODS PDF statement supports the option ACCESSIBLE as a production feature. This option enables accessibility features for PDF output.

- In SAS 9.4M5, ODS PDF supports linking (drilling down) to targets or destinations in PDF documents. For an example, see “Example 9: How to Use Drill-Down and Link to a Target in a PDF File” on page 645.

- In SAS 9.4M6, the method used to build and compress PDF files has been enhanced, resulting in smaller file sizes. This change affects all PDF files, including tagged PDF files.

- In SAS 9.4M6, you can use the new ACCESSIBLEPDF system option to enable the ACCESSIBLE option in the ODS PDF destination by default. For more information, see “ACCESSIBLEPDF System Option” in SAS System Options: Reference.

- Starting in SAS 9.4M6, the new ACCESSIBLE_IDENTIFIER option adds an identifier to the metadata of the PDF file confirming that the PDF produced by SAS meets the PDF Matterhorn Protocol.

- In SAS 9.4M6, alternate text can be added to images specified by the following style attributes:
  - IMAGE= on page 1118
  - PREIMAGE= on page 1125
  - POSTIMAGE= on page 1124

ODS RTF Statement

In SAS 9.4M5, the ODS RTF statement supports the new option NOTRKEEP. This option controls where tables split on a row. See, “ODS RTF Statement” on page 764.

ODS TAGSETS.RTF Statement

The ODS TAGSETS.RTF statement supports the following options and style attributes:

- BACKGROUNDIMAGE= on page 1099
- OPTIONS (WATERMARK=) on page 850
- OPTIONS (VSPACE=) on page 850
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 SAS 9.4M2, the ODS ESCAPECHAR statement supports the following functions that are used with the ODS EPUB and ODS EPUB3 statements:

- BOLD on page 280
- EMPHASIS on page 281
- ITALIC on page 282
- PDF on page 284
- STRONG on page 850

In SAS 9.4M3, 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 53.

**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 SAS 9.4M2, the ODS ESCAPECHAR statement supports the following functions that are used with the ODS EPUB and ODS EPUB3 statements:

- BOLD on page 280
- EMPHASIS on page 281
- ITALIC on page 282
- PDF on page 284
- STRONG on page 850

In SAS 9.4M3, the ODS ESCAPECHAR statement supports the following functions that are used with the ODS EPUB3 statements:

- AUDIO on page 279
- IMAGE on page 281
- MATHML on page 283
- NOTEREF on page 284
- VIDEO on page 288
In SAS 9.4M3, the ODS EPUB3 EVENTS= option supports the following new suboptions:

- BRANCH on page 242
- FIGCAPTION on page 243
- FIGURE on page 243
- IMAGE on page 244

In SAS 9.4M3, the default ODS EPUB destination is ODS EPUB3.

### ODS EXCEL

The ODS EXCEL statement was introduced in SAS 9.4M3.

In SAS 9.4M4, the ODS EXCEL statement supports the FLOW= suboption. For more information, see “FLOW="cell-names", "DATA", "HEADERS", "ROWHEADERS", "TABLES", "TEXT" “ on page 308.

In SAS 9.4M5, the ODS EXCEL statement supports the following options:

- SHEET_INTERVAL= on page 315
- PROTECT_WORKSHEET= on page 313

### ODS HTML5

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

In SAS 9.4M2, the ODS HTML5 statement supports the following new options:

- SHOW_GRAPH_STYLES=
- USE_CSS_RESET=

In SAS 9.4M3, 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 SAS 9.4M3, the ODS HTML5 statement supports the following new options:

- SGE

In SAS 9.4M5, the ODS HTML5 statement supports the following new suboption:

- OUTLINE= on page 467

Accessibility is supported by the ODS HTML5 statement as follows:

In SAS 9.4M4, the ACCESSIBLE_GRAPH option was added to the ODS HTML5 statement as a preproduction feature. In SAS 9.4M5, the ACCESSIBLE_GRAPH option is a production feature for the ODS HTML5 statement. The ACCESSIBLE_GRAPH option adds accessibility metadata to graphs that are created by ODS Graphics.

Users with visual impairments can access the graph using SAS Graphics Accelerator. The accelerator provides users with alternative presentations of SAS data visualizations that have been created with the metadata. For more information, see the software product page for SAS Graphics Accelerator.
For more information, see ACCESSIBLE_GRAPH.

- In SAS 9.4M5, ODS supports the style element FOCUSLINK, which enables you to control the appearance of the link-focus indicator in ODS output. You can use this style element to make the focus indicator more prominent in your ODS HTML5 output. The FOCUSLINK style element enables you to control the link-focus indicator outline color, line style, and line width.

You can create your own private ODS style that adds the FOCUSLINK style element to an existing ODS style. These can be defined using PROC Template or on a CSS style sheet.

For more information, see “ODS Style Elements” on page 1045.

- In SAS 9.4M5, image maps are supported with SVG output in HTML5. However, image maps are supported only when the HTML5 SVG mode is INLINE. Image maps enable data tips and drill-down generation.

- In SAS 9.4M6, alternate text can be added to images specified by the following style attributes:
  - IMAGE= on page 1118
  - PREIMAGE= on page 1125
  - POSTIMAGE= on page 1124

In SAS 9.4M6, the default style for HTML 5.0 output has changed from HTMLBlue to HTMLEncore.

**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). For more information, see:

- ODS LAYOUT ABSOLUTE Statement on page 474
- ODS LAYOUT GRIDDED Statement on page 489

In SAS 9.4M4, the DESCRIPTION= option specifies alternative text for layouts and layout regions.

The DESCRIPTION= option is new for the following statements:

- ODS LAYOUT ABSOLUTE
- ODS LAYOUT GRIDDED
- ODS REGION Statement, Absolute
- ODS REGION Statement, Gridded

**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 layout templates are supported. These gridded layouts correspond to the similarly named built-in PowerPoint layouts: TitleSlide, TitleAndContent, and TwoContent.

- In SAS 9.4M3, the ODS POWERPOINT statement enhances the LAYOUT=TwoContent to include the ADVANCE= suboption. 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 654.

• In SAS 9.4M3, 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 655.

In SAS 9.4M5, the ODS POWERPOINT statement adds the following options:

• BACKGROUNDIMAGE=’NONE’ on page 656
• STARTPAGE=NOW on page 662

**ODS WORD (Preproduction)**

Starting with SAS 9.4M6, the ODS WORD statement is used to create Microsoft Word documents. Documents can contain text, graphics, and tables.

• ODS WORD generates docx files. Docx files require minimal disk space and have fast upload times.

• ODS WORD supports Microsoft Word themes. Custom themes are also available by specifying the STYLE= option. Style definitions also allow you to customize document attributes like page numbers. See See Creating a Page X of Y Header on page 904 for an example of this.

• Tables of contents are customizable with options such as TOC_LEVEL= and TOC_TYPE.

• The ODS WORD statement honors the ORIENTATION= system option, which changes the orientation from portrait to landscape.

---

**Deprecated ODS Tagsets**

Starting in SAS 9.4M2, the following ODS tagsets have been deprecated:

• DOCBOOK
• HTMLCSS
• IMODE
• MVSHTML
• PYX
• SASREPORT family of tagsets
• TPL_STYLE_LIST
Starting in SAS 9.4M3, the following ODS tagsets have been deprecated:

- LATEX
- MLATEX
- TROFF

Documentation Enhancements

Starting in SAS 9.4M2, 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

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

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

The ODS Style Attributes chapter contains a table of all available style elements.
xxvi  Output Delivery System
Accessibility

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
Chapter 1
Getting Started with the Output Delivery System

Welcome to the Output Delivery System .............................................. 4

Gallery of ODS Samples ................................................................... 4
- Introduction to the ODS Samples ................................................. 4
- EPUB Output ................................................................................ 4
- Excel Output ................................................................................ 5
- Graphical Output .......................................................................... 6
- HTML Output ............................................................................... 8
- PDF Output ................................................................................ 9
- PostScript Output ......................................................................... 10
- PowerPoint Output ...................................................................... 11
- RTF Output ................................................................................ 12

A Quick Start to Using ODS .............................................................. 13
- The Purpose of These Examples .................................................... 13
- Creating HTML Output ................................................................. 14
- Creating LISTING Output .............................................................. 15
- Producing Output in Multiple Destinations at the Same Time .......... 16

Creating Accessible Output with the Output Delivery System .......... 20
- Overview .................................................................................... 20
- ODS EPUB and ODS EPUB3 Statements ......................................... 20
- ODS HTML5 Statement ................................................................. 20
- ODS HTML Statement ................................................................... 21
- ODS PDF Statement .................................................................... 21
- ODS PHTML Statement ................................................................. 22
- MSOFFICE2K tagset ..................................................................... 22
- Event Variables ........................................................................... 22
- Style Templates .......................................................................... 23
- Header Attributes ....................................................................... 23

Working with Output Defaults ......................................................... 23
- Overview .................................................................................... 23
- ODS Graphics ............................................................................ 23
- The Default Destination ............................................................... 24
- How to Restore 9.2 Behavior ......................................................... 24

Where to Go from Here ................................................................... 27
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 109.

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

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

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

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 1168.

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 215 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 1.3** Markup Destination Output Viewed with Excel

**Figure 1.4** Model Fit Plot Using Mytemplate and Sashelp.ClassFit
Figure 1.5  PROC SGSCATTER (SAS) with LISTING Style

Scatterplot Matrix for Iris Data

Sepal Length (mm)

Petal Length (mm)

Sepal Width (mm)

Petal Width (mm)

Iris Species  Setosa  Versicolor  Virginica
Figure 1.6  Custom Template Rendered with PROC SGRENDER (SAS) and a Custom Style

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.
### All Products

#### Region by Citysize by Saletype

<table>
<thead>
<tr>
<th>Region</th>
<th>Citysize</th>
<th>Retail</th>
<th>Wholesale</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Quantity</td>
<td>Amount</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sum</td>
<td>Sum</td>
</tr>
<tr>
<td>Brazil</td>
<td>L</td>
<td>Missing</td>
<td>Missing</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>1,066</td>
<td>$20,600</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>472</td>
<td>$11,800</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>1,538</td>
<td>$38,400</td>
</tr>
<tr>
<td>Canada</td>
<td>L</td>
<td>2,421</td>
<td>$80,525</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>1,825</td>
<td>$46,025</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>623</td>
<td>$15,576</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>4,869</td>
<td>$121,725</td>
</tr>
<tr>
<td>France</td>
<td>L</td>
<td>2,303</td>
<td>$57,575</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>2,149</td>
<td>$54,725</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>1,254</td>
<td>$31,150</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>5,706</td>
<td>$143,450</td>
</tr>
<tr>
<td>Mexico</td>
<td>L</td>
<td>2,655</td>
<td>$90,375</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>2,380</td>
<td>$59,000</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>561</td>
<td>$14,025</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>5,576</td>
<td>$135,400</td>
</tr>
<tr>
<td>Total</td>
<td>L</td>
<td>7,379</td>
<td>$194,475</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>7,400</td>
<td>$185,950</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>2,910</td>
<td>$72,550</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>17,689</td>
<td>$442,975</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 649 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.
A Quick Start to Using ODS

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.
    @33 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
M       ME1        28072     31JAN57      24DEC79    203/675-3434
1739    Boyce      Jonathan  New York     NY
M       PT1        66517     28DEC52      30JAN79    212/587-1247
1658    Bradley    Jeremy    New York     NY
M       SCP        17943     11APR55      03MAR80    212/587-3622
1428    Brady      Christine Stamford     CT
F       PT1        68767     07APR58      19NOV79    203/781-1212
1407    Grant      Daniel    Mt. Vernon   NY
M       PT1        68096     26MAR57      21MAR78    914/468-1616
1114    Green      Janice    New York     NY
F       TA2        32928     21SEP57      30JUN75    212/588-1092
;

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</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>03SEP1953</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>08OCT1968</td>
<td>201/812-5665</td>
</tr>
<tr>
<td>1101</td>
<td>18723</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

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 output is formatted in PDF and viewed in Adobe Acrobat Reader.

*Figure 1.16  PDF 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 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

Overview

The following functionality helps you create more accessible output. For complete information on creating accessible ODS output, see Creating Accessible SAS® 9.4 Output Using ODS and ODS Graphics.

Note: 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.

ODS EPUB and ODS EPUB3 Statements

The ODS EPUB 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 a wide range of abilities. For example, when reading an e-book created by the ODS EPUB and ODS EPUB3 destinations using iBooks on an iPad, users can adjust font size, color schemes, and magnification. They can also access the text using assistive technologies such as the Voiceover screen reader and refreshable braille displays. For complete documentation about the ODS EPUB statement, see the “ODS EPUB Statement” on page 215. For complete documentation about the ODS EPUB3 statement, see the “ODS EPUB3 Statement” on page 237.

ODS HTML5 Statement

The ODS HTML5 statement opens, manages, or closes the HTML5 destination, which produces HTML 5.0 output that contains embedded style sheets. ODS HTML5 provides the best accessibility of the ODS destinations. See, “ODS HTML5 Statement” on page 441.

It might be necessary to take steps within the HTML results created by ODS HTML5 to increase the accessibility of your files. Specifying page titles, using headings for sections and tables, generating tables of contents, and adding alternate text to images, videos, and other media, will increase the accessibility of your reports.

ODS HTML5 supports the following accessibility features.

- The HEADING statement in the ODSTEXT procedure adds heading tags such as <h1>, <h2>, and so on, to HTML 5.0. See “HEADING Statement” in SAS Output Delivery System: Procedures Guide and “The ODSTEXT Procedure” in SAS Output Delivery System: Procedures Guide.

- When used with the ODS Report Writing Interface, you can do the following:
  - add alternate text to images, or mark images as decorative (beginning with SAS 9.4M6)
  - add descriptions to layout regions
• add captions to tables (beginning with SAS 9.4M6)

For more information about creating accessible output with ODS, see Creating Accessible SAS® 9.4 Output Using ODS and ODS Graphics. For more information about the ODS Report Writing Interface, see SAS Output Delivery System: Advanced Topics.

• Starting with SAS 9.4M6, you can alternate text to images specified by the following style attributes:
  • IMAGE=
  • PREIMAGE=
  • POSTIMAGE=

ODS HTML5 options that can be used to produce more accessible output are as follows:

• ANCHOR= and CONTENTS= for creating a table of contents. If your output is long and contains many tables and/or graphs, it would be helpful to keyboard and screen reader users if you had a table of contents at the top of your report with links to each table or graph. The ANCHOR= option is specified in the ODS HTML5 statement and enables you to name the links that ODS generates. You can then create your own list of heading links using these names. ODS appends a number on the ANCHOR name when it generates each table.

  See, “Example 2: Creating a Custom Table of Contents with Anchors” on page 471.

• STYLE= for producing output in styles that are more accessible. Use HTMLBLUE, HIGHCONTRAST, and HIGHCONTRASTLARGE styles for accessible output.

• IMAGE_DPI= for producing accessible image resolution.

• SVG_MODE= INLINE for accessibility.

• ACCESSIBLE_GRAPH= for adding accessible metadata to graphs that are created by ODS Graphics. Users with a wide range of abilities will access the accessibility metadata using a companion product to ODS Graphics that will be released in the near future.

When producing SVG output using SVG_MODE, there is one limitation. 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 “ODS HTML5 Statement” on page 441 for details of each HTML5 option.

**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 114.

**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.

ODS PDF supports the following accessibility features:

• Supports option ACCESSIBLE, which enables accessibility features for PDF output. See ODS PDF Statement for details of each ODS PDF option.
• Starting in SAS 9.4M6, the new ACCESSIBLE_IDENTIFIER option adds an identifier to the metadata of the PDF file confirming that the PDF produced by SAS meets the PDF Matterhorn Protocol.

• In SAS 9.4M6, the method used to build and compress PDF files has been enhanced, resulting in smaller file sizes. This change impacts all PDF files including tagged PDF files.

• In SAS 9.4M6, you can use the new ACCESSIBLEPDF system option to create accessible PDF files. For more information, see "ACCESSIBLEPDF System Option" in SAS System Options: Reference.

• Starting with SAS 9.4M6, the ODS Report Writing Interface provides the following accessibility features for PDF output:
  • add alternate text to images, or mark images as decorative
  • add captions to tables

For more information about creating accessible output with ODS, see Creating Accessible SAS® 9.4 Output Using ODS and ODS Graphics. For more information about the ODS Report Writing Interface, see SAS Output Delivery System: Advanced Topics.

• Starting with SAS 9.4M6, you can alternate text to images specified by the following style attributes:
  • IMAGE= on page 1118
  • PREIMAGE= on page 1125
  • POSTIMAGE= on page 1124

**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 695.

**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 811.

**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 Templates

For information about the ODS Styles that SAS provides, see Chapter 10, “Style Templates,” on page 975.

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.

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 24.

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 affect only 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 961.

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 960.

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 31.

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 114 and Appendix 3, “ODS and the HTML Destination," on page 1201. 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 109. 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 57.

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.
Part 2

Concepts

Chapter 2

*Output Delivery System: Basic Concepts* .......................... 31
Chapter 2
Output Delivery System: Basic Concepts

Overview of How ODS Works .......................................................... 31
Components of SAS Output ......................................................... 31
Features of ODS ........................................................................... 33
Clearing the Results Window in the SAS Windowing Environment ........... 33
Understanding Item Stores and Template Stores ................................. 34
Understanding ODS Destinations .................................................... 36
Overview of ODS Destination Categories ....................................... 36
Definition of Destination-Independent Input ...................................... 37
The SAS Formatted Destinations .................................................... 37
The Third-Party Formatted Destinations ......................................... 38
Controlling the Formatting Features of Third-Party Formats ................. 40
ODS Destinations and System Resources ....................................... 41
How ODS Determines the Destinations for an Output Object ............... 41
Selection and Exclusion Lists ........................................................ 42
Customized Output for an Output Object ....................................... 43
Customizing Titles and Footnotes .................................................. 44
Changing SAS Registry Settings for ODS ....................................... 44
Overview of ODS and the SAS Registry .......................................... 44
Changing the SAS Windowing Environment Output Defaults ............... 45
Changing the ODS Default HTML Style ........................................ 47
Changing the Default HTML Version Setting ................................... 48
Changing ODS Destination Default Settings ................................... 49
Changing ODS HTML5 Destination Default Accessible Graph Value ....... 50
Changing ODS Printer Destination Default Printer Value ................... 51
Changing ODS EPUB Destination Default Value .............................. 52
Changing the ODS EPUB and EPUB3 Encoding Settings .................... 53

Overview of How ODS Works

Components of SAS Output

The PROC or DATA step supplies data and the name of the 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.
The following figure shows how SAS produces ODS output.

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

![Diagram showing the Overview of How ODS Works]

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

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

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

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

*Note: Additional tagsets can be found at http://support.sas.com/rnd/base/ods/odsmarkup/*.
**Features of ODS**

ODS delivers output in a variety of formats, and makes the formatted output easy to access.

Important features of ODS include the following:

- ODS combines data with table and graph templates to produce one or more output objects. These objects can be sent to any or all ODS destinations. The currently available ODS destinations can produce the following types of output:
  - 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 and graph 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. Most SAS procedures produce multiple output objects. You can easily create HTML output, an output data set, LISTING output, or printer output from any or all 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 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 program, but you do not need to maintain separate programs 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 in the SAS Windowing Environment**

The ODS HTML destination is turned on by default when you use the SAS windowing environment and SAS Studio. 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`
   
   or
   
   `CLEAR ALL`

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.

*Figure 2.2* Templates Window Showing Item Stores, Template Stores, and Items
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.TMPPLPROCGLM(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.

Troubleshooting

Template 'your-template' was unable to write to the template store!

Errors are generated when a user attempts to store templates in the SASUSER library while the library is in use by another SAS session or another process. By default, templates are stored in the SASUSER.TEMPLAT item store unless the ODS PATH is modified.

Errors might also occur when a user writes to the SASUSER location if the access mode for the template store has been changed to (read).

In addition, the following errors are generated if the RSASUSER option is set:

ERROR: A lock is not available for the SASUSER.TEMPLATE.ITEMSTOR, Lock held by another process.

ERROR: Template "xxxxxx" was unable to write to the template store.

To eliminate the error messages, use the ODS PATH statement to write to an alternate location. The example below adds a temporary item store in the Work directory where the templates will be written. In this example, the Work.Templat item store is added before the default path:

```ods path(prepend) work.templat(update);```
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 2.3 Destination Category Table

<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>
<tr>
<td></td>
<td>EPUB2</td>
<td>EPUB output that is compatible with various e-book readers and adaptable to the accessibility needs of users</td>
</tr>
<tr>
<td></td>
<td>EPUB3</td>
<td>EPUB output that supports multimedia, is compatible with various e-book readers, and is adaptable to the accessibility needs of users</td>
</tr>
<tr>
<td></td>
<td>Excel</td>
<td>Output written in Microsoft Excel format</td>
</tr>
<tr>
<td></td>
<td>HTML</td>
<td>HTML file for online viewing</td>
</tr>
<tr>
<td>Category</td>
<td>Destinations</td>
<td>Results</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HTML5</td>
<td>HTML file for online viewing with multimedia and accessibility support</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.

**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.
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) are free and open e-book standards 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.

The ODS EPUB2 and ODS EPUB3 statements generate the same type of output as ODS EPUB but offer more support for accessibility and multimedia.

Excel
Microsoft Excel is the name of a proprietary commercial software spreadsheet program developed by Microsoft. It is part of the Microsoft Office suite and runs on Microsoft Windows and Apple Mac OS X operating system. An Excel spreadsheet file, or workbook, consists of one or more worksheets. Worksheets can contain data in the form of numbers, currency, and percentages. The ODS EXCEL statement generates output with the .xlsx extension.
**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.

**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.

**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 509. 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 838 for specific information.
**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 838 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

Note: Although you can maintain a selection list for one destination and an exclusion list for another, it is easier to understand the results if you maintain the same types of lists for all the destinations where you route output.

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.
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 555. For a list of ODS output destinations and explanations of each, see “Understanding ODS Destinations” on page 36.

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 793, the “ODS EXCLUDE Statement” on page 332, and the “ODS TRACE Statement” on page 891.

---

**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 63.

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;
```

A few procedures do not produce ODS output. If you produce a trace record for one of these procedures, no template appears in the trace record. Most procedures use one or more templates to produce their output. Each 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 891.

You can use PROC TEMPLATE to modify templates. When a procedure or DATA step uses a table template, it uses the elements that are defined or referenced in its table template. In most cases, the only way to modify the appearance of your output is by modifying one or more templates, which can include table, graph, style, column, and other templates.

**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. Examples of these style options are BOLD, COLOR=, and FONT=. Because these options control only the presentation of the titles and footnotes, they have no effect on titles or footnotes that go to the LISTING or OUTPUT destination. 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 DATA Step Statements: Reference.

When used with graphics, 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. For information about adding titles and footnotes to ODS Graphics, see “Adding Titles, Footnotes, and Text Entries to Your Graph” in SAS Graph Template Language: User’s Guide.

For information about titles and footnotes rendered with and without using the graphics option USEGOPT, see “ODS USEGOPT Statement” on page 896.

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 in the tree. See “The SAS Registry” in SAS Language Reference: Concepts for details 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.
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 23.

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 Value 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 entering 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**.
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 HTML 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.
Figure 2.10  Changing the Default Registry Settings for ODS RTF Output Destination

Changing ODS HTML5 Destination Default Accessible Graph Value

The ACCESSIBLE_GRAPH option in the ODS HTML5 statement adds accessibility metadata to graphs that are created by ODS Graphics. See Creating Accessible SAS® 9.4 Output Using ODS and ODS Graphics for more information.

Users with a wide range of abilities access the accessibility metadata by using the SAS Graphics Accelerator, which is a browser extension that is used in conjunction with the ACCESSIBLE_GRAPH option in the ODS HTML5 destination statement. The accelerator creates an alternative presentation for graphs. For more information about the accelerator, including supported graph types, supported browsers, and installation instructions, see SAS Graphics Accelerator on the SAS Support site.

To change the Accessible Graph value of the ODS HTML5 destination in the SAS registry, perform the following steps:

1. Expand the ODS \Destinations \SCRIPT folders.
2. Select the html5 destination.
3. From the Contents of ‘HTML5’ pane, select Accessible Graph to change the default value.
4. Double-click Accessible Graph. The Edit String Value window appears.
5. Enter the data value of OFF or ON into the Value Data text box in the Edit String Value dialog box. Select OK.
Changing ODS Printer Destination Default Printer Value

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.

Figure 2.11 Changing the Default Printer Value 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 \(\Rightarrow\) 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 2.12*  Changing the Default PCL Printer Value in the Registry Editor Window

---

**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 \(\Rightarrow\) 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 the Default EPUB Version in the Registry Editor Window

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 2.14  Changing ODS EPUB3 Encoding in the SAS Registry
Part 3

Output Delivery System and the DATA Step

Chapter 3

Using ODS with the DATA Step .............................................. 57
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 though 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 107.

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 3.1  Steps to Producing Enhanced ODS Output with the DATA Step

<table>
<thead>
<tr>
<th>Steps</th>
<th>Tools</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify formatting 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. 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.
### Table 3.2  Where to Find More Information about How to Use ODS in the DATA Step

<table>
<thead>
<tr>
<th>Topic</th>
<th>Where to learn more</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODS formatting statements</td>
<td>Chapter 5, “Introduction to ODS Language Statements,” on page 107</td>
</tr>
<tr>
<td>ODS destinations</td>
<td>“Understanding ODS Destinations” on page 36</td>
</tr>
<tr>
<td>How ODS works</td>
<td>“Overview of How ODS Works” on page 31</td>
</tr>
</tbody>
</table>

### 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 DATA Step Statements: Reference.

**Syntax**

```sas
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 62

`variable`

identifies the variable to write.

Example “Example 4: Creating and Using a User-Defined Table Template” on page 89

_**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 63.

@ | @@

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 62

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 63.

**Example**  
“Example 4: Creating and Using a User-Defined Table Template” on page 89

+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 89

@ '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 89

**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 Global 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 63.

**See Also**

- Output Delivery System and the DATA Step on page 57
- Examples on page 73

**Statement**

- “FILE Statement for ODS” on page 63
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 DATA Step 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 73

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 64 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 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 64</td>
</tr>
<tr>
<td>Specify default values for dynamic-attribute values</td>
<td>DYNAMIC= on page 66</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 67</td>
</tr>
<tr>
<td>Specify a name for the output object that the DATA step produces</td>
<td>OBJECT= on page 67</td>
</tr>
<tr>
<td>Specify a label for the output object that the DATA step produces</td>
<td>OBJECTLABEL= on page 68</td>
</tr>
<tr>
<td>Specify the table template to use with the data component to produce the output object</td>
<td>TEMPLATE= on page 68</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.
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 89

`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 89

(*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 70.

### Table 3.4 Attribute Suboptions for the `COLUMNS=` Suboption

<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><code>DYNAMIC= on page 70</code></td>
</tr>
<tr>
<td>Specify a format for the current column</td>
<td><code>FORMAT= on page 70</code></td>
</tr>
<tr>
<td>Specify whether the DATA step uses this column definition for multiple variables</td>
<td><code>GENERIC= on page 71</code></td>
</tr>
<tr>
<td>Specify a label for a particular column</td>
<td><code>LABEL= on page 71</code></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:

\[ \text{dynamic-value-name} = \text{variable-name} \mid \text{constant} \]

*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*.

**LABEL=’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 82

**OBJECT= 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 67)  
- 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 82  

**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 66.  

**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 585 statement.  

**Example**  
“Example 4: Creating and Using a User-Defined Table Template” on page 89
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 77
“Example 3: Assigning Attributes to Columns in ODS Output” on page 82

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 70.

<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 70</td>
</tr>
<tr>
<td>Specify a format for the current column</td>
<td>FORMAT= on page 70</td>
</tr>
</tbody>
</table>
## Task

| Specify whether the DATA step uses this column definition for multiple variables | GENERIC= on page 71 |
| Specify a label for a particular column | LABEL= on page 71 |

### 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 77
- “Example 3: Assigning Attributes to Columns in ODS Output” on page 82

## 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 66

**DYNAMIC Statement**

**Example**  
“Example 4: Creating and Using a User-Defined Table Template” on page 89

### 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 89

---

**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 66

**Example**
“Example 4: Creating and Using a User-Defined Table Template” on page 89

---

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

**See**
LABEL= suboption on page 67

“Example 3: Assigning Attributes to Columns in ODS Output” on page 82

---

**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=
  - DLMSTR=
  - DSD
  - _FILE_=
  - FILEVAR=
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 1  DATA Step Using the GENERIC=ON Suboption**

```sas
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 2  DATA Step Using the GENERIC=ON Option**

```sas
data _null_;  
set top3list;   
file print ods = (   
template='means.topn'   
generic=on   
columns=(   
class=school   
class=year   
sum=moneyRaised_sum   
mean=moneyRaised_mean   
raised=moneyRaised_1   
raised=moneyRaised_2   
raised=moneyRaised_3   
name=name_1   
name=name_2   
```
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 57
• Examples on page 73

Statement

• “PUT Statement for ODS ” on page 59

Examples

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

Features:

FILE PRINT ODS statement
PUT _ODS_ statement

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

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
  1996 INS  Corn     8925
  1996 USA  Wheat    62099
  1996 USA  Rice     7771
  1996 USA  Corn     236064
; run;
Program Description

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

```sas
options obs=15;
```

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

```sas
title 'Leading Grain Producers';
```

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

```sas
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.

```sas
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.

```sas
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.

```sas
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.

```sas
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.
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 1213.
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.

```sas
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.

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

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

```sas
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.

```sas
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.

```sas
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.

```sas
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.

```sas
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.

```
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

**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>10035</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>8660</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Wheat</td>
<td>.</td>
</tr>
<tr>
<td>Indonesia</td>
<td>Rice</td>
<td>51166</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>210054</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: $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 1213.

Program

```
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
```
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.

```sas
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.

```sas
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.

```sas
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.

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;

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=
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

<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;
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;
```
data 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.

```plaintext
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.

```plaintext
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.

```plaintext
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.

\texttt{ods RTF close;}

**RTF Output**

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

![New Subscriber Telephone List](image_url)
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 HTML 5.0, PDF, and RTF output by default, and view the HTML 5.0 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 99. 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 99.

**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 100.

**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= file-reference 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 965 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 .......................... 107

Chapter 6

Dictionary of ODS Language Statements .......................... 109
Chapter 5
Introduction to ODS
Language Statements

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 DATA Step 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 36.

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 38.

Procedure Statements

Chapter 6
Dictionary of ODS Language Statements

ODS Statement Category Descriptions .................................................. 110
ODS Statements by Category .............................................................. 110

Dictionary .......................................................... 114
ODS HTML Statement .......................................................... 114
FILE Statement for ODS .................................................. 159
LIBNAME Statement, SASEDOC ........................................... 169
ODS_ALL_CLOSE Statement .................................................. 173
ODS_CHTML Statement .................................................. 174
ODS_CSVALL Statement .................................................. 200
ODS_DECIMAL_ALIGN Statement ........................................ 211
ODS_DOCUMENT Statement .................................................. 212
ODS_EPUB Statement ...................................................... 215
ODS_EPUB2 Statement ...................................................... 217
ODS_EPUB3 Statement ...................................................... 237
ODS_ESCAPECHAR Statement ............................................... 273
ODS_EXCEL Statement ...................................................... 296
ODS_EXCLUDE Statement ..................................................... 332
ODS_GRAPHICS Statement ..................................................... 340
ODS_HTML3 Statement ...................................................... 369
ODS_HTML Statement ......................................................... 396
ODS_HTML5 Statement ....................................................... 441
ODS_LAYOUT_ABSOLUTE Statement .................................... 474
ODS_REGION Statement, Absolute ........................................ 485
ODS_LAYOUT_GRIDDED Statement ....................................... 489
ODS_REGION Statement, Gridded .......................................... 499
ODS_LAYOUT_END Statement .................................................. 504
ODS_LISTING Statement ....................................................... 504
ODS_MARKUP Statement ........................................................ 509
ODS_NO_DECIMAL_ALIGN Statement .................................. 554
ODS_OUTPUT Statement ......................................................... 555
ODS_PACKAGE Statement ........................................................ 576
ODS_PATH Statement ........................................................... 585
ODS_PCL Statement .............................................................. 587
ODS_PDF Statement ............................................................... 597
ODS_POWERPOINT Statement .................................................. 649
ODS_PREFERENCES Statement .................................................. 694
ODS_PHTML Statement ............................................................ 695
ODS_PRINTER Statement ............................................................ 722
ODS_PROCLABEL Statement .................................................. 748
ODS_PROCTITLE Statement ..................................................... 748
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. 576)</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. 159)</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. 900)</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>Category</td>
<td>Language Elements</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ODS: Output Control</td>
<td>LIBNAME Statement, SASEDOC (p. 169)</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></td>
<td>ODS_ALL_CLOSE Statement (p. 173)</td>
<td>Closes all open ODS output destinations.</td>
</tr>
<tr>
<td></td>
<td>ODS DOCUMENT Statement (p. 212)</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>
</tr>
<tr>
<td></td>
<td>ODS ESCAPECHAR Statement (p. 273)</td>
<td>Defines a representative character to be used in output strings.</td>
</tr>
<tr>
<td></td>
<td>ODS EXCLUDE Statement (p. 332)</td>
<td>Specifies output objects to exclude from ODS destinations.</td>
</tr>
<tr>
<td></td>
<td>ODS GRAPHICS Statement (p. 340)</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>
</tr>
<tr>
<td></td>
<td>ODS LAYOUT ABSOLUTE Statement (p. 474)</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>
</tr>
<tr>
<td></td>
<td>ODS REGION Statement, Absolute (p. 485)</td>
<td>Creates a region container for absolute layouts. A region can contain output (such as text, tables, images, graphs) or nested layout containers.</td>
</tr>
<tr>
<td></td>
<td>ODS LAYOUT GRIDDED Statement (p. 489)</td>
<td>Enables you to arrange output dynamically in a two-dimensional grid structure.</td>
</tr>
<tr>
<td></td>
<td>ODS REGION Statement, Gridded (p. 499)</td>
<td>Creates a region container for gridded layouts. A region can contain output (such as text, tables, images, graphs) or nested layout containers.</td>
</tr>
<tr>
<td></td>
<td>ODS LAYOUT END Statement (p. 504)</td>
<td>Ends a gridded or absolute layout statement block.</td>
</tr>
<tr>
<td></td>
<td>ODS PATH Statement (p. 585)</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>
</tr>
<tr>
<td></td>
<td>ODS PREFERENCES Statement (p. 694)</td>
<td>Reverts the ODS settings back to start-up defaults.</td>
</tr>
<tr>
<td></td>
<td>ODS PROCLABEL Statement (p. 748)</td>
<td>Enables you to change a procedure label.</td>
</tr>
<tr>
<td></td>
<td>ODS PROCTITLE Statement (p. 748)</td>
<td>Determines whether to write the title that identifies the procedure that produces the results in the output.</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 RESULTS Statement (p. 763)</td>
<td>Tracks ODS output in the Results window.</td>
</tr>
<tr>
<td></td>
<td>ODS SELECT Statement (p. 793)</td>
<td>Specifies output objects for ODS destinations.</td>
</tr>
<tr>
<td></td>
<td>ODS SHOW Statement (p. 806)</td>
<td>Writes the specified selection or exclusion list to the SAS log.</td>
</tr>
<tr>
<td></td>
<td>ODS TEXT= Statement (p. 883)</td>
<td>Inserts text into your ODS output.</td>
</tr>
<tr>
<td></td>
<td>ODS TRACE Statement (p. 891)</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. 896)</td>
<td>Determines whether ODS uses traditional SAS/GRAPH option settings.</td>
</tr>
<tr>
<td></td>
<td>ODS VERIFY Statement (p. 899)</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</td>
<td>ODS DECIMAL_ALIGN Statement (p. 211)</td>
<td>Controls the justification of numeric columns when no justification is specified.</td>
</tr>
<tr>
<td>Formatted</td>
<td>ODS LISTING Statement (p. 504)</td>
<td>Opens, manages, or closes the LISTING destination.</td>
</tr>
<tr>
<td></td>
<td>ODS NO_DECIMAL_ALIGN Statement (p. 554)</td>
<td>Right-justifies numeric columns when no justification is specified.</td>
</tr>
<tr>
<td></td>
<td>ODS OUTPUT Statement (p. 555)</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</td>
<td>ODS HTML Statement (p. 114)</td>
<td>Opens, manages, or closes the HTML destination, which produces HTML 4.0 output in the SAS Windowing environment. This output can contain embedded style sheets.</td>
</tr>
<tr>
<td>Formatted</td>
<td>ODS CHTML Statement (p. 174)</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. 200)</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. 215)</td>
<td>Opens, manages, or closes the ODS EPUB destination, which generates EPUB e-books.</td>
</tr>
<tr>
<td></td>
<td>ODS EPUB2 Statement (p. 217)</td>
<td>Opens, manages, or closes the EPUB2 destination, which generates EPUB e-books using the EPUB2 standard.</td>
</tr>
<tr>
<td></td>
<td>ODS EPUB3 Statement (p. 237)</td>
<td>Opens, manages, or closes the EPUB3 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 EXCEL Statement</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>
<td></td>
</tr>
<tr>
<td>ODS HTML3 Statement</td>
<td>Opens, manages, or closes the HTML3 destination, which produces HTML 3.2 formatted output.</td>
<td></td>
</tr>
<tr>
<td>ODS HTML Statement</td>
<td>Opens, manages, or closes the HTML destination, which produces HTML 4.0 output in the SAS Windowing environment. This output can contain embedded style sheets.</td>
<td></td>
</tr>
<tr>
<td>ODS HTML5 Statement</td>
<td>Opens, manages, or closes the HTML5 destination, which produces HTML 5.0 output that contains embedded style sheets.</td>
<td></td>
</tr>
<tr>
<td>ODS MARKUP Statement</td>
<td>Opens, manages, or closes the MARKUP destination, which produces SAS output that is formatted using one of many different markup languages.</td>
<td></td>
</tr>
<tr>
<td>ODS PCL Statement</td>
<td>Opens, manages, or closes the PCL destination, which produces printable output for PCL (HP LaserJet) files.</td>
<td></td>
</tr>
<tr>
<td>ODS PDF Statement</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>
<td></td>
</tr>
<tr>
<td>ODS POWERPOINT Statement</td>
<td>Opens, manages, or closes the ODS destination for PowerPoint, which produces PowerPoint output.</td>
<td></td>
</tr>
<tr>
<td>ODS PHTML Statement</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.</td>
<td></td>
</tr>
<tr>
<td>ODS PRINTER Statement</td>
<td>Opens, manages, or closes the PRINTER destination, which produces printable output.</td>
<td></td>
</tr>
<tr>
<td>ODS PS Statement</td>
<td>Opens, manages, or closes the PS destination, which produces PostScript (PS) output.</td>
<td></td>
</tr>
<tr>
<td>ODS RTF Statement</td>
<td>Opens, manages, or closes the RTF destination, which produces output written in Rich Text Format for use with Microsoft Word.</td>
<td></td>
</tr>
<tr>
<td>ODS Tagset Statement</td>
<td>Opens, manages, or closes the specified tagset destination.</td>
<td></td>
</tr>
<tr>
<td>ODS TAGSETS.RTF Statement</td>
<td>Opens, manages, or closes the RTF destination, which produces measured output that is written in Rich Text Format.</td>
<td></td>
</tr>
<tr>
<td>ODS WORD Statement</td>
<td>Opens, manages, or closes the ODS destination for Word, which generates Microsoft Word output compatible with Microsoft Office 2010 and later versions.</td>
<td></td>
</tr>
</tbody>
</table>
Dictionary

ODS HTML Statement

Opens, manages, or closes the HTML destination, which produces HTML 4.0 output in the SAS Windowing environment. This output can contain embedded style sheets.

Valid in: Anywhere

Category: ODS: Third-Party Formatted

Default: The default style for markup family destinations is HTMLBlue. The default output for the HTML destination is HTML5 in the SAS Studio client for SAS Viya.

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:

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

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


Syntax

```
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 332

**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 793

**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 42.

See “ODS SHOW Statement” on page 806

**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. ODS assumes that if the anchor name ends in a number, then the number should be treated as a starting point for incrementing the 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.

Similarly, if you specify ANCHOR="TABULATE1", then ODS names the first anchor tabulate1. The second anchor is named tabulate2.
### 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:

```
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/GRAFH procedures and with the DEVICE=JAVA, JAVAMETA, or ACTIVEX options in the GOPTIONS statement.

**See**

[SAS/GRAFH: 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 Global 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 140.

**(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 141.

**(NO_TOP_MATTER)**

See: For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 141.

**(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.
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 44.

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 140.

(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 142.

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 Global 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 140.

(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 141.

(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 141.

(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 142.

(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 142.

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 Global 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 548

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 365.

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>Excel</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>PowerPoint</td>
<td>PNG</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 44.


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.
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.

(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 Global 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 140.

**(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 141.

**(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 141.

**(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 142.

**(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 142.

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 541

**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.

Requirements  You must enclose aggregate-file-storage-location in quotation marks.

The specified aggregate-file-storage-location must exist in the file system.

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 Global 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 617

IMAGE_DPI=

specifies the image resolution for graphical output.

Alias DPI=

Default 96

Restriction This option applies only 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.

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:
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.

Requirements: 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');

Requirements: _suboption(s)_ must be enclosed in parentheses.

Example: “Example: Using the DOC Suboption to Get ODS
TAGSETS.HTMLPANEL Information ” on page 837

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 576

Example: “Example 1: Creating an ODS Package” on page 580

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 Global Statements: Reference.

title ENTRY

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 140.

(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 141.

(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 141.

(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 142.

(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 142.

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 Global Statements: Reference.
libref.catalog
specifies a SAS catalog to write to.

See For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Global 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**

```sas
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.

**STYLESETTING='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 Global 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 140.

  **(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 141.

  **(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 141.

  **(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 142.

**(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 142.

**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 546

**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 336

**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 1109 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 543

(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.
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 23.

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**

```sas
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.
Output 6.1  Options Suboptions Available for HTML

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

```plaintext
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.

```plaintext
ods html file="test.html" options(pagebreak='no');
```

Print only two observations of the data set.

```plaintext
options obs=2;
```

Print the data set Sashelp.Class. The PROC PRINT statement prints the Sashelp.Class data set.

```plaintext
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.

```plaintext
proc print data=Sashelp.Class;
run;
```

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>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>
</tbody>
</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>
</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**

```plaintext
proc sort data=grain_production;
    by year country type;
run;

ods html body='grain-body.htm'
    contents='grain-contents.htm'
    frame='grain-frame.htm'
    page='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;
    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.

   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.
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
```
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 $cntry. kilotons comma12.;
  where year=1996;
  title 'Leading Grain-Producing Countries';
  footnote 'Measurements are in metric tons.';
run;
ods html close;
filename reports '../ods/grain-reports-body.htm' mod;
filename reports 'GrainReport.html' mod;
data _null_;
  file reports;
  put "<h2>The preceding output is from PROC PRINT.</h2>";
  put "I am going to try a variety of procedures.";
  put "Let me know which procedure you prefer.";
  put "This report uses the Blockprint style.</h2>";
run;
ods html body=reports (no_top_matter no_bottom_matter)
  anchor='report';
proc report data=grain_production nowindows;
  where year=1996;
```
column country type kilotons;
define country / group width=14 format=$c ENTRY. ;
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)
```

**Specify the style for formatting the HTML output.** The STYLE= option specifies that the style Blockprint be used.

```
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.

```
proc print data=grain_production;
  var country type kilotons;
  format country $c ENTRY. 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.
Example 4: Appending to HTML Files 153

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.

data _null_;
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.

```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 6.4  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 553 and "OPTIONS ( DOC=| <suboption(s)> )" on page 133.

```ods html options(pagebreak='no');

Create the graph. The SYMBOL statements and the GPLOT procedure create the graph.

```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 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>
See Also

- Appendix 3, “ODS and the HTML Destination,” on page 1201

Statements

- “ODS MARKUP Statement” on page 509
- “ODS Tagset Statement” on page 807
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 DATA Step Statements: Reference.

Syntax

FILE PRINT ODS <=(ODS-suboptions)> <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 73

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 160 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 160</td>
</tr>
<tr>
<td>Specify default values for dynamic-attribute values</td>
<td>DYNAMIC= on page 162</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 header specified in the COLUMNS= or VARIABLES= suboption</td>
<td>LABEL= on page 163</td>
</tr>
<tr>
<td>Specify a name for the output object that the DATA step produces</td>
<td>OBJECT= on page 163</td>
</tr>
<tr>
<td>Specify a label for the output object that the DATA step produces</td>
<td>OBJECTLABEL= on page 164</td>
</tr>
<tr>
<td>Specify the table template to use with the data component to produce the output object</td>
<td>TEMPLATE= on page 164</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-l<=variable-name-l<(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, \texttt{score1-score5}) to specify multiple column names.

Example “Example 4: Creating and Using a User-Defined Table Template” on page 89

\textit{variable-name} specifies a variable in the DATA step to place in the specified column.

Default If you omit \textit{variable-name}, then ODS looks for a DATA step variable named \textit{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, \texttt{score1-score5}) to specify a range of variable names.

Example “Example 4: Creating and Using a User-Defined Table Template” on page 89

\textit{(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 166.

\begin{table}[h]
\centering
\begin{tabular}{|c|l|}
\hline
\textbf{Task} & \textbf{Attribute Suboption} \\
\hline
Specify a value for the variable defined by the DYNAMIC statement in a table template & \texttt{DYNAMIC=} on page 166 \\
\hline
Specify a format for the current column & \texttt{FORMAT=} on page 166 \\
\hline
Specify whether the DATA step uses this column definition for multiple variables & \texttt{GENERIC=} on page 167 \\
\hline
Specify a label for a particular column & \texttt{LABEL=} on page 167 \\
\hline
\end{tabular}
\caption{Attribute Suboptions for the COLUMNS= Suboption}
\end{table}

Requirement You must enclose \textit{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 \textit{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.

<table>
<thead>
<tr>
<th>Restriction</th>
<th>ODS does not recognize the column names as a match unless you specify the (COLUMNS=(GENERIC=ON)) suboption.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction</td>
<td>If you do not specify a table template, the GENERIC= suboption is set to ON.</td>
</tr>
<tr>
<td>Tips</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

**LABEL=’column-label’**
specifies a label for any column that does not have a label specified in the COLUMNS= or VARIABLES= suboption.

<table>
<thead>
<tr>
<th>Default</th>
<th>If you use the LABEL= suboption, ODS uses the first of these labels that it finds:</th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>

| 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 82 |

**OBJECT= 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 163)
- 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 82

**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 162.

**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 585 statement.

**Example**

“Example 4: Creating and Using a User-Defined Table Template” on page 89
`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 77
- “Example 3: Assigning Attributes to Columns in ODS Output” on page 82

`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 166.

<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 166</td>
</tr>
<tr>
<td>Specify a format for the current column</td>
<td>FORMAT= on page 166</td>
</tr>
</tbody>
</table>
Task | Attribute Suboption
--- | ---
Specify whether the DATA step uses this column definition for multiple variables | GENERIC= on page 167
Specify a label for a particular column | LABEL= on page 167

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 77

“Example 3: Assigning Attributes to Columns in ODS Output” on page 82

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 162

DYNAMIC Statement

Example “Example 4: Creating and Using a User-Defined Table Template” on page 89

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 89

**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 162

Example
“Example 4: Creating and Using a User-Defined Table Template” on page 89

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

See LABEL= suboption on page 163

“Example 3: Assigning Attributes to Columns in ODS Output” on page 82

**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=
  - DLMSTR=
  - DSD
  - _FILE_=
  - FILEVAR=
• HEADER=
• PAD

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 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 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  
    mean=moneyRaised_mean  
    raised=moneyRaised_1  
    raised=moneyRaised_2  
    raised=moneyRaised_3  
    name=name_1  
    name=name_2

168  Chapter 6  • Dictionary of ODS Language Statements
name=name_3
school=school_1
school=school_2
school=school_3
year=year_1
year=year_2
year=year_3
}
);
put _ods_
run;

**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 57
- Examples on page 73

**Statement**

- “PUT Statement for ODS ” on page 900

---

**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.

<table>
<thead>
<tr>
<th>Valid in:</th>
<th>Anywhere except SAS Viya</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category:</td>
<td>ODS: Output Control</td>
</tr>
<tr>
<td>Restriction:</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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.

See  "Understanding Sequence Numbers " in *SAS Output Delivery System:
Procedures Guide*

**Additional LIBNAME Statement Arguments and Options**

For additional arguments and options that are valid for the LIBNAME statement, see the
LIBNAME statement in *SAS DATA Step 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 Sasuser library.

```sas
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 891.

```
proc glm  data=plant_stats;
class month;
model age age2 age3=month / nouni;
manova h=month /print;
run;
```

Create the output objects. The GLM procedure creates the output objects. The Plants 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 891.

```
proc glm  data=plants order=data;
class type block;
model stemleng=type block;
means type;
contrast 'compost vs others'  type -1 -1 -1 -1  6 -1 -1;
contrast 'river soils vs non' type -1 -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;
```

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.
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

Output 6.8 The Contents of Mylib

See Also

Procedures
• “The DOCUMENT Procedure” in SAS Output Delivery System: Procedures Guide

Statements
• “ODS DOCUMENT Statement” on page 212
• “ODS TRACE Statement” on page 891

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 except SAS Viya
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=**
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

\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

\textbf{Without Arguments} 
If you use the ODS CHTML statement without an action or options, then it opens the CHTML destination and creates CHTML output.

\textbf{Actions} 
The following actions are available for the ODS CHTML statement:

\texttt{CLOSE} 
closes the destination and any files that are associated with it.

\textit{Tip} When an ODS destination is closed, ODS does not send output to that destination. Closing an unneeded destination conserves system resources.

\texttt{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 332

\texttt{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 793

\texttt{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 42.

See
“ODS SHOW Statement” on page 806

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. ODS assumes that if the anchor name ends in a number, then the number should be treated as a starting point for incrementing the 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.

Similarly, if you specify ANCHOR="TABULATE1", then ODS names the first anchor tabulate1. The second anchor is named tabulate2.

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/GRAF 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 Global 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 197.

  (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 198.

  (NO_TOP_MATTER)
    See  For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 198.

  (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 199.

  (URL='Uniform-Resource-Locator')
    See  For complete documentation about the URL= suboption, see “(URL='Uniform-Resource-Locator')” on page 199.

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 538.

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.

Tip  When you use the ODS HTML statement with the FILE= option to specify a fully-qualified path with a directory and a file name, the best practice for generating HTML files is to separate the path from the file. To accomplish this, use a combination of the PATH= option, which specifies the directory, and the FILE= option, which names the actual file. For example:
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 197.

(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.
For complete documentation about the URL= suboption, see “(URL= ’Uniform-Resource-Locator’)” on page 199.

**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 Global 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 197.

(NOBOTTOM_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 198.

(NOTOP_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 198.

(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 199.
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 199.

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 Global 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 197.

**(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 198.

**(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 198.

**(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 199.

**(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.
For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator’)” on page 199.

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 541

**GFOOTNOTE | NOGFOOTNOTE**

controls the location where footnotes are printed in the graphics output.

**GFOOTNOTE**

writes footnotes that are created by SAS/GRAPH, the SG PLOT 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.

Requirements You must enclose aggregate-file-storage-location in quotation marks.

The specified aggregate-file-storage-location must exist in the file system.

fileref

is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.
If you specify a fileref in the GPATH= option, then ODS does not use information from the GPATH= option when it constructs links.

For information about the FILENAME statement, see “FILENAME Statement” in SAS Global 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.

You must enclose Uniform-Resource-Locator in quotation marks.

NONE
specifies that no information from the GPATH= option appears in the links or references.

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.

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
mark up 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 617

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:

\[ <\text{META} \text{ your-metatext-is-here}> \]

**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:

\[ \text{BODY} = '\text{REPORT.XML}' \]

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:

\[ \text{BODY} = '\text{MAY5.XML}' \]

**OPTIONS ( DOC= | <suboption(s)> )**

specifies tagset-specific suboptions and a named value.

\[ (\text{DOC=} '\text{HELP}' | '\text{QUICK}' | '\text{SETTINGS}' | '\text{CHANGELOG}') \]

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 837

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 576

Example  "Example 1: Creating an ODS Package" on page 580

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 Global 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 197.

(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 198.

(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 198.

(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 199.

(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 199.

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 Global Statements: Reference.
libref.catalog
  specifies a SAS catalog to write to.

See For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Global 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 Global 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 197.

(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 198.

(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 198.

(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 199.

(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 199.
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 546

TEXT=\text-string\n
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=\text-string \text-string

See  For information about events and event variables, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in \textit{SAS Output Delivery System: Procedures Guide}.

Example  “Example: Conditionally Excluding Output Objects and Sending Them to Different Output Destinations” on page 336

TRANTAB= \translation-table\n
specifies the translation table to use when transcoding a file for output.


\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 1109 in \texttt{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.

\begin{itemize}
  \item BODY=
  \item CONTENTS=
  \item PAGE=
  \item FRAME=
  \item STYLESHEET=
  \item TAGSET=
\end{itemize}

Requirements  You must enclose DYNAMIC in parentheses.

You must specify DYNAMIC next to the \textit{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.
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 543

(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 546

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 except SAS Viya  
**Category:** ODS: Third-Party Formatted  
**See:** "ODS Tagset Statement" on page 807

**Syntax**

```
ODS CSV ALL (<(ID=> identifier)> < action> );
ODS CSV ALL (<(ID=> identifier)> <option(s)> );
```

**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

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.

If CSV results are preferred without titles, notes and BY lines, see the ODS TAGSETS statement.

**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 332

**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 793

**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 42.

See “ODS SHOW Statement” on page 806
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 Global 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 538.

  **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.
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.

(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.
(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.

(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 617

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 837

**PACKAGE <(package-name)>**
specifies that the output from the destination be added to a package.

```sas
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.

```sas
ods csvall package(a);
```

**See** “ODS PACKAGE Statement” on page 576

**Example** “Example 1: Creating an ODS Package” on page 580

**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 Global Statements: Reference*.

*libref.catalog*
specifies a SAS catalog to write to.

**See** For information about the LIBNAME statement, see “LIBNAME Statement” in *SAS Global 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:

```plaintext
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:

```plaintext
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=`

**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.

If CSV results are preferred without titles, notes, and BY lines, see “ODS Tagset Statement” on page 807.

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

```plaintext
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;
```
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.

```r
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.

```r
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.

```r
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.

```r
proc print data=grain_production;
   title 'Leading Grain-Producing Countries';
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.

```r
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

<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>IRI</td>
<td>Wheat</td>
<td>1995</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>IRI</td>
<td>Rice</td>
<td>1995</td>
<td>49860</td>
</tr>
<tr>
<td>12</td>
<td>IRI</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>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.

**Output 6.10 CSVALL Output Viewed in Microsoft Excel**

<table>
<thead>
<tr>
<th></th>
<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>1</td>
<td></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>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Obs</td>
<td>Country</td>
<td>Type</td>
<td>Year</td>
<td>Kilotons</td>
<td></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>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>BRZ</td>
<td>Rice</td>
<td>1995</td>
<td>11236</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>BRZ</td>
<td>Corn</td>
<td>1995</td>
<td>36276</td>
<td></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>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>CHN</td>
<td>Rice</td>
<td>1995</td>
<td>185226</td>
<td></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>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>IND</td>
<td>Wheat</td>
<td>1995</td>
<td>63007</td>
<td></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>
<td></td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>IND</td>
<td>Corn</td>
<td>1995</td>
<td>9800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>INS</td>
<td>Wheat</td>
<td>1995</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>INS</td>
<td>Rice</td>
<td>1995</td>
<td>49860</td>
<td></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>
<td></td>
</tr>
<tr>
<td>16</td>
<td>13</td>
<td>USA</td>
<td>Wheat</td>
<td>1995</td>
<td>59494</td>
<td></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>
<td></td>
</tr>
<tr>
<td>18</td>
<td>15</td>
<td>USA</td>
<td>Corn</td>
<td>1995</td>
<td>137300</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**ODS DECIMAL_ALIGN Statement**

Controls the justification of numeric columns when no justification is specified.

- **Valid in:** Anywhere except SAS Viya
- **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 except SAS Viya

**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 "DOCUMENT Procedure" in SAS Output Delivery System: Procedures Guide.

**Syntax**

```plaintext
ODS DOCUMENT action;

ODS DOCUMENT
< NAME=<libref.> member-name <(access-option)> >
< DIR=(< PATH=path <(access-option)> < LABEL="label" > )> >
< CATALOG=permanent-catalog | _NULL_ > ;
```

**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.
See “ODS EXCLUDE Statement” on page 332

**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 793

**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 806

**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.
**NAME=** `<libref.>` `member-name` `<(access-option)>`

*libref*  
specifies the SAS library where the document is stored.

---

**Interaction**  
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 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.
**Default**  
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 except SAS Viya

**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 52.

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.

**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 53.

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 229
- “Example 2: Create an ODS EPUB E-book with Inline and Nonlinear Tables” on page 233
- “Example 3: ODS EPUB Book with a Customized TOC” on page 235

**Syntax**

```
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 52.

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 217 or “ODS EPUB3 Statement” on page 237.
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 52.

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 a wide range of abilities. 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 except SAS Viya
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 52.

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 53.

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 229
“Example 2: Create an ODS EPUB E-book with Inline and Nonlinear Tables” on page 233
“Example 3: ODS EPUB Book with a Customized TOC” on page 235

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.

Default NONE

Restriction A destination must be open for this action to take effect.

See “ODS EXCLUDE Statement” on page 332

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 793
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 42.

See
“ODS SHOW Statement” on page 806

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.

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.

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 44.

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 Global 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 548

**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 44.

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.
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.

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 44.

**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.


(ISBN='isbn')

Requirement Every EPUB e-book must be uniquely identifiable.
(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')

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 44.

See  For a complete discussion about styles, see Chapter 9, “Overview,” on page 965.

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 52.

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 a wide range of abilities. 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 52 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 52 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.

```sas
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.

```sas
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.

```sas
ods proclabel='Non-linear Table';
title 'Non-linear Table';
ods epub options(nonlinear="table");
```

**Print the data set Sashelp.Prdsal2.** The PROC PRINT statement prints the Sashelp.Prdsal2 data set.

```sas
proc print data=sashelp.prdsal2(obs=15);
    run;
```

**Close the EPUB destination.**

```sas
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 52 for additional information.

Program

```sas
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.

```
proc print data=sashelp.class;
  run;
```

Close the EPUB destination.

```
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 except SAS Viya
- **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 52.
  
  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 53.

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.

<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 332</td>
</tr>
</tbody>
</table>

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

| Default | ALL |
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 42.

See
“ODS SHOW Statement” on page 806

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.

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.

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 44.
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 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 Global 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 548

---

**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 44.

**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. 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.
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".

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;
```

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".

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;
```

**(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**
```
%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.
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 259

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 44.

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.

 \(\text{DEFEAT\_IBOOKS\_CACHING= 'YES' | 'NO' | 'OFF' | 'ON'}\)

defeats iBooks e-book reader's caching of e-books that have the same title.

 \(\text{OFF}\)
caches iBooks of the same title.

 Alias NO

 \(\text{ON}\)
defeats iBooks caching.

 Alias YES

 Default ON | YES

 \(\text{DESCRIPTION= 'text-string'}\)
specifies a description of the content of the e-book.

 \(\text{EMBEDDED\_FONTS= 'fonts'}\)
specifies a list of fonts to be embedded in the e-book.


 \(\text{HIDE\_CONTENTS= 'YES' | 'NO' | 'OFF' | 'ON'}\)
hides entries in a linear table of contents.

 \(\text{OFF}\)
does not hide the entries in a linear table of contents.

 Alias NO

 \(\text{ON}\)
hides the entries in a linear table of contents.

 Alias YES

 Default OFF | NO

 \(\text{HIDE\_LANDMARKS= 'YES' | 'NO' | 'OFF' | 'ON'}\)
hides entries in a linear table of landmarks.

 \(\text{YES}\)
hides entries in a linear table of landmarks.

 \(\text{NO}\)
does not hide the entries in a linear table of landmarks.

 \(\text{OFF}\)
does not hide the entries in a linear table of landmarks.

 Alias NO
ON
  hides the entries in a linear table of landmarks.

Alias  YES

Default  OFF | NO

(ISBN="isbn")
For more information, see http://en.wikipedia.org/wiki/International_Standard_Book_Number. If not specified, a Universally Unique Identifier (UUID) is generated for the e-book. For more information, see http://en.wikipedia.org/wiki/Universally_unique_identifier.

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 247.

(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.

<table>
<thead>
<tr>
<th>OFF</th>
<th>overrides user-specified fonts.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias NO</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ON</th>
<th>honors user-specified fonts.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias YES</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default</th>
<th>ON</th>
<th>YES</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Note</th>
<th>iBooks on iPad is the preferred e-book reader. Results can vary using other e-book readers.</th>
</tr>
</thead>
</table>

(PAGEBREAK= 'YES' | 'NO' | 'OFF' | 'ON') honors explicit page breaks.

<table>
<thead>
<tr>
<th>OFF</th>
<th>does not honor explicit page breaks.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias NO</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ON</th>
<th>honors explicit page breaks.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias YES</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AUTO</th>
<th>allows ODS and procedures to determine when page breaks occur. Honors explicit page breaks.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>AUTO</td>
</tr>
</tbody>
</table>

(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.

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 44.

See  For a complete discussion about styles, see Chapter 9, “Overview,” on page 965.

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 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 44.

**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.

```sas
%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 259 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 263.
• See “Example 5: ODS EPUB3 - Adding Video Using Inline Functions” on page 265.

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 267.
• See “Example 7: ODS EPUB3 - Adding Video Using RWI” on page 269.
• 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 a wide range of abilities. For example, when reading an e-book created by the ODS EPUB and ODS EPUB3 destinations using iBooks on an iPad, users can adjust font size, color schemes, and magnification. They can also access the text using assistive technologies such as the Voiceover screen reader and refreshable braille displays.

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.

\textit{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

\section*{Examples}

\subsection*{Example 1: ODS EPUB E-book with TOC and Landmark TOC}

\begin{itemize}
  \item **Features:** ODS EPUB3 Statement
    \begin{itemize}
      \item \texttt{FILE=} option
      \item \texttt{TITLE=} option
      \item \texttt{OPTIONS} linear navigation suboption
    \end{itemize}
  \item **Other features:** Data Set
    ARIMA procedure
\end{itemize}

\subsection*{Details}

The \texttt{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 \texttt{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.

\subsection*{Program}

\begin{verbatim}
title;
ods epub3 file="tocwithlandmarktoc.epub"
   title="PROC ARIMA TOC and Landmark TOC in EPUB3 e-book"
   options\{linear_nav="all"\};
\end{verbatim}
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

**Table of Contents**

*The Arima Procedure*
- Identification
- Descriptive Statistics
- Autocorrelation Check for White Noise
- Series Correlation Panel
- Estimation
- Conditional Least Squares Estimation
- Fit Statistics
- Correlations of Parameter Estimates
- Autocorrelation Check of Residuals
- Residual Correlation Panel
- Residual Normality Panel
- Model Filters
- Model for variable x
- Autoregressive Factors
- Forecast
- Forecasts for x
- Forecasts Only Plot

**Table of Landmarks**
- Start Reading

**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
- ODS TEXT 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;
```
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=

- ODSTEXT 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;
```
%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));
title;
ods escapechar='^';
ods epub3 file="figure.epub"
   title="Figure and Caption Test"
   style=mystyle;
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 "^\newlineHere 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")
title;
ods epub3 close;
proc template;
delete styles.mystyle;
run;

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.

```sas
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.
**Output 6.20** ODS EPUB3 - Adding Figure Captions to a Figure

Here is a figure with a top caption.

![Figure 1. Nonlinear table icon](image1.png)

Here is a figure with a bottom caption.

![Figure 2. High resolution (2x) nonlinear table icon](image2.png)

---

**Example 4: ODS EPUB3 - Adding Audio Using Inline Functions**

**Features:**
- ODS EPUB3 Statement
  - FILE= option
  - NEWCHAPTER= option
  - OPTIONS nonlinear suboption
  - EVENT= option
- 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";
proc odstext contents="";
```
Program Description

Open the ODS EPUB3 destination. Open the ODS EPUB3 destination, provide a file for the EPUB output, and create a new chapter.

ods epub3 file="AudioEscapechar.epub"
   newchapter=now
   options(nonlinear="chapter");

Set the ODS inline formatting escape character.

ods escapechar='^';

Add a title when the procedure runs that describes what is being added to the e-book.

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.

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.

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.

title;
ods epub3 close;

Output

The following output shows an image of the e-book created that has embedded audio.
You can embed Audio into your eBook
Use the Inline Formatting AUDIO Function

Example 5: ODS EPUB3 - Adding Video Using Inline Functions

Features:
- ODS EPUB3 Statement
  - FILE= option
  - NEWCHAPTER= option
  - TITLE= option
- ODS PROCLABEL statement
- ODSTEXT 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/
   marmot.jpg;float=left;
   margin=top=0;
```
No vacation is complete without home video.
If you enjoy wildlife, check out the marmot
\(^{\text{noteref large mountain-dwelling ground squirrel}}\)
we saw while driving the
\(^{\text{style [url='http://en.wikipedia.org/wiki/Trail_Ridge_Road']}\}
Trail Ridge Road).";
run;
ods epub3 close;

Program Description

Open the ODS EPUB3 destination and suppress the system title. Open the ODS
EPUB3 destination, provide a file for the EPUB output, create a title for the e-book, and
create a new chapter.

title;
ods epub3 file="VideoEscapechar.epub"
        title="Marmot Sighting eBook"
        newchapter=now;

Set the ODS inline formatting escape character.

ods escapechar='^';

Create a label for the chapter in the table contents

ods proclabel = "Marmot Sighting (Video)";

Add video to the e-book. Use the ODS ESCAPECHAR VIDEO function with PROC
ODSTEXT to add video to your e-book. In this case, we also added a cover image for
the video and specified the size of the video. We added a note where you can touch the
footnote marker and more information about the marmot will appear.

proc odstext contents="";
p "^{video multimedia/marmot.m4v?controls=controls;
poster=multimedia/
marmot.jpg;float=left;
margintop=0;
marginright=.5em;
width=240px;
height=135px}
No vacation is complete without home video.
If you enjoy wildlife, check out the marmot
\(^{\text{noteref large mountain-dwelling ground squirrel}}\)
we saw while driving the
\(^{\text{style [url='http://en.wikipedia.org/wiki/Trail_Ridge_Road']}\}
Trail Ridge Road).";
run;

Close the EPUB3 destination and clear the title.

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 ESCAPECAR 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 to add audio to your e-book.

Program

```plaintext
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.

```plaintext
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.

```plaintext
   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.

```
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.

```
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
```
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>
        <munderover>
            <mo>&#x02211;</mo>
            <mo>&#x02212;</mo>
        </munderover>
    </mrow>
    <mo>&#x02211;</mo>
</math>}
<mo>&#x02211;</mo>
```

<mrow>
<mi>n</mi>
<mo>=</mo>
<mn>1</mn>
</mrow>
<mrow>
<mn>5</mn>
</mrow>
</munderover>
<mrow>
<msup>
<mrow>
<mi>n</mi>
</mrow>
<mrow>
<mn>2</mn>
</mrow>
</msup>
</mrow>
</math>

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.

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 '!{mathml <math xmlns="http://www.w3.org/1998/Math/MathML">
    <mrow>
      <munderover>
        <mo>&#x02211;</mo>
        <mrow>
          <mi>n</mi>
          <mo>=</mo>
          <mn>1</mn>
        </mrow>
        <mrow>
          <mn>5</mn>
        </mrow>
      </munderover>
      <msup>
        <mi>n</mi>
        <mn>2</mn>
      </msup>
    </mrow>
  </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 except SAS Viya
Category: ODS: Output Control
Restrictions: Affects all open destinations except for the LISTING destination. SAS/GRAPH does not support inline formatting.
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 278.

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 \ 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 275.

**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 “Style Attributes: Detailed Information” 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 Global 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:

```
^\{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 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:

```
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 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 1045.
Using Unicode Symbols

ODS supports the ability to incorporate Unicode symbols such as Greek symbols into your output using the UNICODE inline formatting function. The UNICODE syntax is \{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 predefine 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 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 inline style function. You can still use existing inline style functions, ^{DAGGER} and ^{SIGMA}.

To find out what symbols are available to you for Windows, search for the 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:

```plaintext
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:

```plaintext
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:

```plaintext
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:

```plaintext
set $unicodeMap['<new function name>' ] '<unicode value>';
```

Save the file and compile the modified tagset using PROC TEMPLATE as follows:

```plaintext
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 44. For information about all the True Type fonts available, 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.

```sas
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*.

```sas
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 6.6 on page 278. 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 6.6  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 279</td>
<td>Filename Fileref Web Address</td>
</tr>
<tr>
<td>BOLD on page 280</td>
<td>Text</td>
</tr>
<tr>
<td>DAGGER on page 280</td>
<td>None</td>
</tr>
<tr>
<td>DATE on page 280</td>
<td>None</td>
</tr>
<tr>
<td>DEST on page 281</td>
<td>OUTPUT destination</td>
</tr>
<tr>
<td>EMPHASIS on page 281</td>
<td>Text</td>
</tr>
<tr>
<td>IMAGE on page 281</td>
<td>Filename Fileref</td>
</tr>
<tr>
<td>ITALIC on page 282</td>
<td>Text</td>
</tr>
<tr>
<td>Function Name</td>
<td>Argument</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>LASTPAGE</td>
<td>None</td>
</tr>
<tr>
<td>LEADERS</td>
<td>String</td>
</tr>
<tr>
<td>MATHML</td>
<td>MathML Presentation Markup</td>
</tr>
<tr>
<td>NBSPACE</td>
<td>Optional number</td>
</tr>
<tr>
<td>NEWLINE</td>
<td>Optional number</td>
</tr>
<tr>
<td>NOTEREF</td>
<td>Number</td>
</tr>
<tr>
<td>PAGEOF</td>
<td>None</td>
</tr>
<tr>
<td>PDF</td>
<td>Filename, Fileref</td>
</tr>
<tr>
<td>RAW</td>
<td>String</td>
</tr>
<tr>
<td>SIGMA</td>
<td>None</td>
</tr>
<tr>
<td>STRONG</td>
<td>None</td>
</tr>
<tr>
<td>STYLE</td>
<td>Style elements, style attributes, and STYLE= option formats</td>
</tr>
<tr>
<td>SUB</td>
<td>Arguments to subscript</td>
</tr>
<tr>
<td>SUPER</td>
<td>Arguments to superscript</td>
</tr>
<tr>
<td>THISPAGE</td>
<td>None</td>
</tr>
<tr>
<td>TOCENTRYINDENT</td>
<td>Length</td>
</tr>
<tr>
<td>TOCENTRYPAGE</td>
<td>None</td>
</tr>
<tr>
<td>UNICODEPAGE</td>
<td>Unicode value</td>
</tr>
<tr>
<td>VIDEO</td>
<td>Filename, Filerefer</td>
</tr>
</tbody>
</table>

**AUDIO Function**

```
^\{AUDIO filename | fileref | web-address; <options-list>\}
```

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
ods escapechar='^';
proc odstext contents="";
p "^audio SAS02_Orchestral30.mp3?controls=controls"/style={just=c};
run;

Example
“Example 4: ODS EPUB3 - Adding Audio Using Inline Functions” on page 263

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.

See
“Example 1: Basic Inline Formatting Functions ” on page 289

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.

<table>
<thead>
<tr>
<th>Tip</th>
<th>You can use this function only with the ODS EPUB3 destination.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>ods epub3 file=&quot;ImageeBook.epub&quot;</td>
</tr>
<tr>
<td></td>
<td>title=&quot;My Trip to Bhutan in eBook&quot;;</td>
</tr>
<tr>
<td></td>
<td>ods escapechar='^';</td>
</tr>
<tr>
<td></td>
<td>proc odstext contents=&quot;&quot;;</td>
</tr>
<tr>
<td></td>
<td>p &quot;Visit Bhutan&quot;/style={just=c};</td>
</tr>
<tr>
<td></td>
<td>p &quot;^[image Bhutan.jpg?width=75%;alt_desc=Bhutan]*&quot;</td>
</tr>
<tr>
<td></td>
<td>/style={just=c};</td>
</tr>
<tr>
<td></td>
<td>ods epub3 close;</td>
</tr>
</tbody>
</table>

Example

“Example 4: ODS EPUB3 - Adding Audio Using Inline Functions” on page 263

**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 293

**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:

```plaintext
PostText = "^{leaders . }^\{tocentrypage}\n```

**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**
```plaintext
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 271

---

**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 289

---

**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 289

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 289

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 265

PAGEOF Function  
\{PAGEOF\}  
s 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**

```plaintext
proc odstext;
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.

**Tip**
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 289

---

**SIGMA Function**

```
^ {SIGMA}
```

generates the Greek SIGMA sign $\sigma$.

**Tip**
The preferable way to produce a SIGMA sign is to use the UNICODE function.

**See**
“Example 1: Basic Inline Formatting Functions” on page 289

---

**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:

```plaintext
^ {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:

```plaintext
^ {style rowheader [color=red] my text};
```

or
\{'style rowheader my text\};

See Chapter 11, “Style Elements,” on page 1045

**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 1074.

**Example** “Example 2: Customizing Titles and Page Numbers” on page 293

**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 275.

**SUB Function**
\{'SUB <subscript-character>\}

**subscript-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 subscript-value must immediately follow the SUB function.

See “Example 1: Basic Inline Formatting Functions ” on page 289

**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 289

THISPAGE Function

\{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 293

TOCENTRYINDENT Function

\{TOCENTRYINDENT <len>\}

len

is the amount to be indented per level. Example code is:

PreText = "\{tocentryindent 2em\}"

Tip This function can be used only with the PRINTER destination.

TOCENTRYPAGE Function

\{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

\{UNICODE <unicode-value | 'unicode-value'X>\}

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 276.

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.

See “Example 1: Basic Inline Formatting Functions” on page 289
is syntax used with ODS Statistical Graphics. A hexadecimal value is enclosed in single or double quotation marks followed by an \text{x}. The \text{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 276.

<table>
<thead>
<tr>
<th>Tips</th>
<th>Thorndale Duospace WT J is the default font used in the inline style Unicode function for the PDF destination.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The \text{unicode-value} can be enclosed with single or double quotation marks.</td>
</tr>
</tbody>
</table>

| 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 |

\textbf{VIDEO Function}

\begin{verbatim}
^{{\text{VIDEO}} \text{filename} | \text{fileref}; <options-list>}
\end{verbatim}

includes a video in an e-book created by using the ODS EPUB3 destination.

- \text{filename} | \text{fileref}
  - ODS EPUB3 supports the M4V, MP4, and QuickTime video formats.
  - \text{filename} is the name and optional path of an external video file.
  - \text{fileref} is a file reference that has been assigned to an external file. Use the FILENAME statement to assign a fileref.

- \text{web-address}
  - is the web address of an external video file.

- \text{<options-list>}
  - The options specified in the \text{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.
    - Image 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**

```plaintext
ods epub3 file="VideoBook.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;
  margin-right=.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**

```plaintext
options nodate nonumber;
```
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, `\cf12` is an instruction that the RTF destination recognizes and can display. The PDF destination does not recognize this instruction.

```title4 'Example of \{raw \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.

```title6 "Example \{style [foreground=red] of Super, Alpha \{super \{unicode ALPHA\} \{style [foreground=green] Nested\}\} Formatting\} and Scoping";
```
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.

```plaintext
title7 "Example of SUB, ^{sub
  ^{style [foreground=red] red
  ^{style [foreground=green] green } and
  ^{style [foreground=blue] blue styles }}]} and
SIGMA ^{unicode SIGMA} Functions*;
```

Print the data set.

```plaintext
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.

```plaintext
ods _all_ close;
```

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

<table>
<thead>
<tr>
<th>Example of</th>
<th>Non-Breaking Spaces Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newline Function</td>
<td></td>
</tr>
<tr>
<td>Example of RAW RAW function</td>
<td></td>
</tr>
<tr>
<td>Example of UNICODE a function</td>
<td></td>
</tr>
<tr>
<td>Example of Super, Alpha a Nested Formatting and Scoping</td>
<td></td>
</tr>
<tr>
<td>Example of SUB, red green and blue styles and SIGMA a Functions</td>
<td></td>
</tr>
</tbody>
</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>
</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
- 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 destination to conserve resources. If the destination were left open, then ODS would produce both HTML and PDF output.
Open 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*

---

**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 except SAS Viya

**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 works only 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**

**ODS EXCEL** `<(<ID>= identifier)> < action> ;`

**ODS EXCEL** `<(<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='text-string'**
  - Specify the author of the Excel 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 Excel document

- **CLOSE**
  - Close the destination and the file that is associated with it.

- **COMMENTS='text-string'**
  - Add comments to the properties of the Excel document

- **CSSSTYLE='file-specification' <(media-type1 <...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='number'**
  - Specify the image resolution for the graphical output

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

- **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)
Specify 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 332

**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 793

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 42.

See  “ODS SHOW Statement” on page 806

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 enables 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 44.
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 DATA Step 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
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 548.

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.

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 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 617

**OPTIONS (< suboption(s)>)**

specifies destination-specific suboptions with space-delimited name=value pairs.

**suboption(s)** are the following:

**ABSOLUTE_COLUMN_WIDTH =number-list**

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. A value of 0 resets the default.

Valid units of measure:

- in inches
- cm centimeters
- em standard typesetting measurement unit for width
- ex standard typesetting measurement unit for height
- in inches
- mm millimeters
- pt a printer’s point
- px pixels

**Default** Character

**Tip**

Delimit multiple row values with commas.

**Example**

ods excel file='c:\file-path\footer.xlsx'
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). The number-list is a comma separated list of numbers.

Valid units of measure:

- in inches
cm  centimeters
em  standard typesetting measurement unit for width
ex  standard typesetting measurement unit for height
in  inches
mm  millimeters

pt  a printer’s point
px  pixels

Default  Points
Example  ods excel options(absolute_row_height='20');

(AUTOFILTER = 'ALL' | 'NONE' | 'range' | 'column')
turns on filtering for specified columns in the worksheet.

ALL  an autofilter is applied to all columns.

NONE  no autofiltering is applied.

range  specifies the range of columns to which filtering is applied. Acceptable
        values for the range are a numeric range ('3-5') or a column range ('A:C').

column  specifies the column to which filtering is applied. Columns can be written as
        a number or a letter.

Restriction  This option does not support the specification of multiple
        individual columns (for example, '2,4').

Default  NONE.
Example  ods excel file='c:\file-path\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
(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 OFF

Example ods excel options(blackandwhite='on');

(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='column' | 'range' | 'HEADER')
controls how column headings are repeated across pages.

HEADER
repeat any of the columns containing headers.

column
specifies that the header of the column specified is repeated on each page. A
column can be a number or a letter.
Example  The following example specifies that the header for column 1 is repeated across the printed page:
ods excel options(column_repeat='1');

\( range \)
specifies that the headers of the columns within the range specified are repeated on each page. A range can be numbers ('1-3') or columns ('A:C').

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.

<table>
<thead>
<tr>
<th>ON</th>
<th>embed footnotes in the worksheet.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OFF</th>
<th>do not embed footnotes in the worksheet.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>NO</td>
</tr>
</tbody>
</table>

| 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.

<table>
<thead>
<tr>
<th>ON</th>
<th>embedded footnotes appear only once at the bottom of the worksheet.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OFF</th>
<th>embedded footnotes appear at the bottom of the worksheet.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>NO</td>
</tr>
</tbody>
</table>

| 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.

<table>
<thead>
<tr>
<th>ON</th>
<th>embed titles in the worksheet.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OFF</th>
<th>do not embed titles in the worksheet.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>NO</td>
</tr>
</tbody>
</table>

| Default | NO |

Example ods excel file='c:\file-path\myxml.xlsx' options(embedded_titles='on');

Example “Example 1: Customizing Your Excel Output” on page 320
(EMBED_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 320

(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

Default OFF

Interaction If FITTOPAGE="OFF" and either PAGES_FITWIDTH="1" or PAGES_FITHEIGHT="1", FITTOPAGE is turned back on.

Example ods excel options(fittopage='on');

FLOW= <"cell-names", "DATA", "HEADERS", "ROWHEADERS", "TABLES", "TEXT">
specifies that a designated Worksheet area enables Wrap Text and disables newline character insertion. Excel wraps the text to the column width.

cell-names enables Wrap Text for a single cell such as “A12” or a cell range such as “C1:E4”.

DATA enables Wrap Text for table data cells.

HEADERS enables Wrap Text in table column headers.

Alias HEADER

ROWHEADERS enables Wrap Text for table row headers.
**TABLES**

enables Wrap Text for all parts of a table: HEADER, ROWHEADER, and DATA.

**TEXT**

makes ODS TEXT output work like titles, footnotes, PROC titles, and BY lines. The text is written into multiple merged cells and Wrap Text is enabled.

**Examples**

```r
ods excel file= "c:\file-path\test.xlsx" options(flow="A2");
```

```r
ods excel file= "c:\file-path\test.xlsx" options(flow="tables");
```

```r
ods excel file= "c:\file-path\test.xlsx" options(flow="text");
```

**(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**

```r
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 not scroll with the scroll bar.

**Alias** YES | TRUE

**OFF**

headers do scroll with the scroll bar.

**Alias** NO | FALSE

**number**

the number of the header row that does not scroll with the scroll bar.

**Default** OFF

**Example**

```r
ods excel options(frozen_headers='on');
```

**(FROZEN_ROWHEADERS= 'OFF' | 'ON' | 'number')**

specifies if the row headers are on the left scroll when the table data scrolls.
ON

the row headers on the left are frozen when the table data scrolls.

Alias YES

OFF

the row headers on the left scroll when the table data scrolls.

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-listcolumn' | 'range')
specifies the columns to hide.

number-list

specifies a comma-delimited list of column numbers

column

specifies the number or letter of a column

range

specifies a cell range. Cell ranges can be written as numbers (1-3) or letters (A:C).

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.

Examples 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= NO NOTES | NO)
suppresses messages from Excel.

NO NOTES
suppresses messages

NO
leaves messages

Default    NO

(Orientation= 'PORTRAIT' | 'LANDSCAPE'
orientates 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.

Interaction | If option PAGES_FITHEIGHT="1", the FITTOPAGE option is turned on.
Example | ods excel options(PAGES_FITHEIGHT='3');

(PAGES_FITWIDTH='number')
specifies the number of pages to fit the worksheet across when printing.

Interaction | If option PAGES_FITWIDTH="1", the FITTOPAGE option is turned on.
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 delimited by commas, or you can use a cell range. The following example indicates top left corner and bottom right corner:

print_area='a,2,g,20'
print_area='A2:G20'

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.

Tip | You can specify special formatting and Visual Basic for Applications (VBA) codes in the PRINT_HEADER= and PRINT_FOOTER= options. For more information, see Formatting and VBA Codes for Headers and Footers.
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.

Tip You can specify special formatting and Visual Basic for Applications (VBA) codes in the PRINT_HEADER= and PRINT_FOOTER= options. For more information, see Formatting and VBA Codes for Headers and Footers.

Example The following example specifies “My custom header” as the text for headers.
```
ods excel file='c:\file-path\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");

(PROTECT_WORKSHEET= ’OFF’ | ’ON’) protects the worksheet by making it read-only.

ON read-only access protects the worksheet.

Alias YES

OFF does not protect the worksheet with read-only access.

Alias NO

Default OFF

Example ods excel options(protect_worksheet="ON");

(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
(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 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.

Heights can be written in the following units of measurement: in (inches), cm (centimeters), mm (millimeters), px (pixels), ex (x heights), em (m widths), pt (points).

Default Points. 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='20mm,50mm,100mm,20mm,50mm,100mm,50mm');

(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 examples specifies that the headers for rows 1, 2, and 3 are repeated across the printed page:

```ods excel options(row_repeat='1:3');
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");```

**(SCREEN_RESOLUTION_SCALE_FACTOR='display-scale-factor')**
specifies the scale factor to optimize viewing of Excel spreadsheets on high-resolution displays.

**Default** 100%

**Tip**
Typical display scale factors are 125% or 150%

**(SHEET_INTERVAL= 'BYGROUP' | 'PAGE' | 'PROC' | 'NONE' | 'NOW' | 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.

**Interaction**
The option SHEET_INTERVAL='NONE' overrides the PAGEBREAK=YES option for PROC ODSTEXT and PROC ODSLIST. That is, if the ODS EXCEL option SHEET_INTERVAL='NONE', the PAGEBREAK=YES option used by either PROC ODSTEXT or PROC ODSLIST is ignored. The ODSLIST and ODSTEXT output are placed on the current worksheet.

**NOW**
creates a new worksheet. When used, the next output object starts on a new sheet. After SHEET_INTERVAL='NOW' is executed, the SHEET_INTERVAL option reverts to the previous setting.

**Interaction**
The option SHEET_INTERVAL='NONE' overrides the PAGEBREAK=NO option for PROC ODSTEXT and PROC ODSLIST. That is, SHEET_INTERVAL='NOW' creates a new worksheet when PROC ODSLIST or PROC ODSTEXT uses the PAGEBREAK=NO option.
ODS EXCEL OPTIONS (SHEET_INTERVAL=’now’);

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='C:\file-path\myExcel.xlsx' OPTIONS(SHEET_INTERVAL=’proc’);

Example "Example 1: Customizing Your Excel Output" on page 320

(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 320

(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 (A1).

Default 1,1 (A1)

Tip This option cannot be changed in the middle of a sheet.

Example ODS EXCEL OPTIONS(START_AT=’2,2’);

ods excel options (start_at=’B2’);

(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 320

(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='ON' | 'OFF')
  specifies whether procedure titles, titles, BY lines, and footnotes wrap across lines.

OFF
  specifies that procedure titles, titles, BY lines, and footnotes wrap across lines.

Alias  NO

ON
  specifies that procedure titles, titles, BY lines, and footnotes do not wrap across lines.

Alias  YES

Default  ON

Example  ods excel file="c:\file-path\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
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 box. The date field is updated whenever you open the worksheet. When the SASDATE option is used, instead of a date field, the ODS destination for Excel inserts the date and time that the worksheet was created. 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)
specifies 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.

Default 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

Overview
The ODS destination for Excel uses Microsoft Open Office XML Format for Office 2010 and later. The ODS EXCEL 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
- An Introduction to Creating Multi-Sheet Microsoft Excel Workbooks the Easy Way with SAS

Supported File Types
The following table provides descriptions of the supported file types for the ODS destination for Excel.

Table 6.7 Description of Supported File Types

<table>
<thead>
<tr>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>PNG (Portable Network Graphic)</td>
<td>Supports true color, gray-scale, and 8-bit images.</td>
</tr>
</tbody>
</table>

Securing ODS-Generated Excel Files
The ODS destination for Excel does not support password protection at the workbook level. However, you can use the “(PROTECT_WORKSHEET= 'OFF' | 'ON')” on page 313 option to protect the worksheet by making it read-only.
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 suppressed.
- The title created by the TITLE statement is embedded in the output.
- The worksheet labels are customized.

**Program**

```
ods excel file="c:\file-path\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 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;
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.

```sas
ods excel file="c:\file-path\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.

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.

ods excel close;
```
### Example 2: Customizing ODS Excel Output by Using the `TAGATTR='format'` Style Attribute

**Features:**
- ODS EXCEL Statement

**Other features:**
- PROC PRINT procedure
  - VAR with STYLE=TAGATTR
  - SUM with STYLE=TAGATTR
  - LABEL

#### Details

In the following example, the `TAGATTR='format'` style attribute enables 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='format'` style attribute.

#### Program

```plaintext
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='c:\file-path\tagattr.xlsx';

proc print data=prdsale(obs=15) noobs label split='*';
  id country region division;
  var prodtype product quarter month year;
  var predict actual /
    style(data)={tagattr='format:$#,##0_);[Red]($(#,##0)'};
  var difference /
    style(data)={tagattr='format:$#,##0_);[Red]($(#,##0) formula:RC[-1]-RC[-2]'};
  sum predict actual difference /
    style(data)={tagattr='format:$#,##0_);[Red]($(#,##0)'};
  label prodtype = 'Product*Type'
    predict = 'Predicted*Sales*For Area'
    actual = 'Actual*Sales*Amount';
run;

ods excel close;
ods html;

---

**Program Description**

**Close the HTML destination.**

ods html close;

**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.

data prdsale;
  set sashelp.prdsale;
  Difference = actual-predict;
run;

**Sort the data set.**

proc sort data=prdsale;
  by country region division year;
run;

**Open the ODS destination for Excel and name the output file.**

ods excel file='c:\file-path\tagattr.xlsx';

**Begin the PRINT procedure step.**

proc print data=prdsale(obs=15) noobs label split='*';
  id country region division;
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.

```plaintext
var prodtype product quarter month year;
var predict actual /
   style(data)={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].

```plaintext
var difference /
   style(data)={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.

```plaintext
sum predict actual difference /
   style(data)={tagattr='format:$#,##0_;[Red]\($#,##0\)'};
```

Specify the labels for the column headings.

```plaintext
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.

```plaintext
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.
Customize the Data in the Excel Workbook Using the TAGATTR= Style Attribute

Example 3: Customizing ODS Excel Output by Using the TAGATTR='type' Style Attribute

Features:
- ODS EXCEL Statement

Other features:
- PROC PRINT procedure
  - VAR with STYLE=TAGATTR

Details
In the following example, the TAGATTR= style attribute enables the user to customize the type and format of the data in the Excel workbook. In the example, the data is converted to a number in Excel and formatted using the Japanese Yen with the TAGATTR= style attribute.

Program
```bef
proc format;
  picture myyen
    low <= 0 = "0000.00" (prefix='-¥')
    other = "0000.00" (prefix='¥');
run;
data;
  input a1 a2 a3;
datalines;
-20   19 -12.34
100 -123 99.99
-123 555 21.00
999   1 13.25
;
run;
ods excel file="c:\file-path\test.xlsx" options(embedded_titles="yes");
title 'TAGATTR';
title2 'The data should be stored as numbers';
title3 'formatted with the Excel format.';
```

proc print data=_last_ noobs;
  var _numeric_/ 
    style(data)={tagattr='type:number format:¥#,##0.00_);[Red](¥#,##0.00)'};
    format _numeric_ myyen.;
run;
run;
ods excel close;

Program Description

Create a custom format for the data. This format will be used later in the program.

proc format;
  picture myyen
    low < 0 = "0000.00" (prefix='¥')
    other = "0000.00" (prefix='¥');
run;

Create the data set.

data;
  input a1 a2 a3;
  datalines;
-20   19 -12.34
100 -123  99.99
-123  555  21.00
999    1  13.25
;
run;

Open the ODS destination for Excel and name the output file. Create titles that are displayed in the output.

ods excel file="c:\file-path\test.xlsx" options(embedded_titles="yes");
  title 'TAGATTR';
  title2 'The data should be stored as numbers';
  title3 'formatted with the Excel format.';

Customize the output data type using the TAGATTR= style attribute. The TAGATTR= style attribute on the first VAR statement specifies that Excel stores the values as a number. An additional format is applied, which prints negative values in red and surrounded by parenthesis.

proc print data=_last_ noobs;
  var _numeric_/ 
    style(data)={tagattr='type:number format:¥#,##0.00_);[Red](¥#,##0.00)'};
    format _numeric_ myyen.;
run;
run;

Close the ODS destination for Excel.

ods excel close;
Excel Output

This output is created using the ODS destination for Excel. It uses the style attribute TAGATTR='type' to customize the data in the Excel workbook.

Example 4: 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.

```
#Expense .header {
    background-color:green
}

#Reports .header {
    background-color:red
}
```

Program

```
ods html close;
ods excel file="c:\file-path\excelAnchorCss.xlsx"
```
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.
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.
ods excel file="c:\file-path\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.
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.

```plaintext
ods excel close;
ods html;
```
### Excel Output

**Output 6.31**  Customized ODS Excel Output By Using a Style Sheet

<table>
<thead>
<tr>
<th></th>
<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>1</td>
<td>Obs</td>
<td>Name</td>
<td>Sex</td>
<td>Age</td>
<td>Height</td>
<td>Weight</td>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></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>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Obs</th>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>1</td>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69.0</td>
<td>112.5</td>
</tr>
<tr>
<td>24</td>
<td>2</td>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84.0</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98.0</td>
</tr>
<tr>
<td>26</td>
<td>4</td>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
<td>102.5</td>
</tr>
<tr>
<td>27</td>
<td>5</td>
<td>Henry</td>
<td>M</td>
<td>14</td>
<td>63.5</td>
<td>102.5</td>
</tr>
<tr>
<td>28</td>
<td>6</td>
<td>James</td>
<td>M</td>
<td>12</td>
<td>57.3</td>
<td>83.0</td>
</tr>
</tbody>
</table>
Example 5: Using BORDERBOTTOMCOLOR with Excel Output

Features:
- ODS EXCEL Statement options
  - FILE=

Other features:
- PROC REPORT

Details

The following example shows how the thickness of the border is interpreted by EXCEL when using the BORDERBOTTOMCOLOR style attribute.

Program

```plaintext
data temp;
  z = 0;
  a = 1;
  b = 2;
  c = 3;
  d = 4;
  e = 5;
  f = 6;
run;

ods excel file='c:\file-path\p.xlsx';

proc report data=temp;
  columns z a b c d e f ;
  define z / style(column)=[borderbottomcolor=red borderbottomwidth=0pt];
  define a / style(column)=[borderbottomcolor=red borderbottomwidth=1pt];
  define b / style(column)=[borderbottomcolor=red borderbottomwidth=2pt];
  define c / style(column)=[borderbottomcolor=red borderbottomwidth=3pt];
  define d / style(column)=[borderbottomcolor=red borderbottomwidth=4pt];
  define e / style(column)=[borderbottomcolor=red borderbottomwidth=5pt];
  define f / style(column)=[borderbottomcolor=red borderbottomwidth=6pt];
run;
ods excel close;
```

Program Description

Create a data set called temp. This data set maps the BORDERBOTTOMWIDTH value to the thickness of the border that it creates.

```plaintext
data temp;
  z = 0;
  a = 1;
  b = 2;
  c = 3;
  d = 4;
  e = 5;
  f = 6;
run;
```

Open the ODS destination for Excel.

```plaintext
ods excel file='c:\file-path\p.xlsx';
```
Use PROC REPORT to show the thickness of the bottom border.

```
proc report data=temp;
  columns z a b c d e f;
  define z / style(column)=[borderbottomcolor=red borderbottomwidth=0pt];
  define a / style(column)=[borderbottomcolor=red borderbottomwidth=1pt];
  define b / style(column)=[borderbottomcolor=red borderbottomwidth=2pt];
  define c / style(column)=[borderbottomcolor=red borderbottomwidth=3pt];
  define d / style(column)=[borderbottomcolor=red borderbottomwidth=4pt];
  define e / style(column)=[borderbottomcolor=red borderbottomwidth=5pt];
  define f / style(column)=[borderbottomcolor=red borderbottomwidth=6pt];
run;
```

Close the ODS destination for Excel.

```
ods excel close;
```

**Excel Output**

**Output 6.32 Using BORDERBOTTOMCOLOR with the ODS Destination for Excel**

![Excel Output Image]

---

**ODS EXCLUDE Statement**

Specifies output objects to exclude from ODS destinations.

**Valid in:** Anywhere except SAS Viya  
**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 42.

Each exclusion has the following form.

```
output-object <(PERSIST)>
```

**output-object**  
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:

  \texttt{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:

  \texttt{Univariate.City\_Pop\_90.TestsForLocation}

  Then the partial paths are as follows:

  \texttt{City\_Pop\_90.TestsForLocation}
  \texttt{TestsForLocation}

- a label that is enclosed in quotation marks.

  For example:

  \texttt{"The UNIVARIATE Procedure" "Tests for Location"}

- a label path. For example, the following is the label path for the output object:

  \texttt{"The UNIVARIATE Procedure"."CityPop\_90"."Tests For Location"}

  \textit{Note:} The trace record shows the label path only if you specify the \texttt{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:

  \texttt{"The UNIVARIATE Procedure"."CityPop\_90"."Tests For Location"}

  Then the partial label paths are as follows:

  \texttt{"CityPop\_90"."Tests For Location"}
  \texttt{"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, \texttt{TestsForLocation\#3} refers to the third output object that is named \texttt{TestsForLocation}.

See \textit{“ODS TRACE Statement” on page 891}.

\textbf{(PERSIST)}

keeps the \textit{output-object} that precedes the \texttt{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.

```ods exclude summary (nowarn);```

```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 36.

**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 555.
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:

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:

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.
_NAME_
  is the name of the output object.
_PATH_
  is the name 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>
<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 555. For a list of ODS output destinations and explanations of each, see “Understanding ODS Destinations” on page 36.

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 793, the “ODS EXCLUDE Statement” on page 332, and the “ODS TRACE Statement” on page 891.

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=BP Pressure;
   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.

```plaintext
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.

```plaintext
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.

```plaintext
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.

```plaintext
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.

```plaintext
ods html close;
```

Close the PDF destination. This ODS PDF statement closes the PDF destination and all the files that are associated with it.

```plaintext
ods pdf close;
```
Output 6.33  Partial HTML Output with Systolic Output Objects

Output 6.34  Partial PDF Output with Diastolic Output Objects

See Also

Statements

- “ODS SELECT Statement ” on page 793
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.

ATTRPRIORITY=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 graphical elements allowed per plot when a data skin is applied.

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 data label collision avoidance 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.

LINEPATTERNOBSMAX=n
specifies the maximum the number of vertices for a patterned line.

LOESSOBSMAX=n
specifies an upper limit for the number of observations that can be used with a loess plot.

NBINSMAX=n
specifies the maximum number of bins that are processed for histograms.

NXYBINSMAX=n
specifies the maximum number of bins that are processed for heat maps.

OBSMAX=n
specifies the maximum number of observations that are processed.

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 data tip areas allowed before data tips are disabled.
TOTALCELLMAX=$n$
specifies the maximum number of total cells in a graph where the number of cells is determined dynamically by classification variables.

WIDTH=$dimension$
specifies the width of any graph.

Without Arguments
If ODS Graphics is currently disabled, then specifying the ODS GRAPHICS statement without options enables it. If ODS Graphics is currently enabled, then specifying the ODS GRAPHICS statement leaves it 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.

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
smooths 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
smooths 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 disabled. 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 SAS 9.4M3, 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 SAS 9.4M3, 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.

Tip Use the ATTRPRIORITY=NONE option if you want groups to be distinguished by color, marker, and line changes for all styles that use color.

BORDER | NOBORDER | BORDER=ON | OFF
specifies whether to draw a border around each graph.

BORDER
draws a border around the graph.

NOBORDER
does not draw a border around the graph.
BORDER=ON | OFF
specifies whether to draw the graph with a border on the outermost layout.

ON
draws a border around the graph.

Alias YES

OFF
does not 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. The option specifies how the BY line is displayed 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 most graph templates:

```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 6.9 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>
<tr>
<td><em>BYLINE</em></td>
<td>This variable contains the text string that can be displayed as a title or footnote.</td>
</tr>
</tbody>
</table>
The variables in the table are set automatically only for analytical procedures that support ODS Graphics. For all other procedures, the variables are not set automatically (NULL). To determine whether the procedure you are using supports ODS Graphics, refer to the procedure documentation.

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

Restriction
  This option does not work with the ODS Graphics procedures such as SGPLOT and SGPANEL. To remove BY lines in those procedures, use the NOBYLINE SAS system option. Example: options nobyline;

DATASKINMAX=n
  specifies the maximum number of graphical elements allowed per plot when a data skin is applied.

  Note: This feature applies to SAS 9.4M1 and to later releases.

  n
    specifies a positive integer.

Default  200

DISCRETEMAX=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.

  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. In that case, use the DISCRETEMAX= option to increase the maximum number of discrete values that are allowed. Starting with SAS 9.4M5, the log message includes a suggested value for DISCRETEMAX=.

DRILLTARGET="_blank" | "_self" | "_parent" | "_top" | "frame-name"
  specifies the window that displays the drill-down output.

  Note: This option is supported only for HTML.
uses the new browser window to display the drilldown output.

Default  

Requirements  

"_blank"

opens a new browser window to display the drilldown output.

Requirements  

"_self"

opens the drill-down output in the same window.

Requirements  

"_parent"

opens the drill-down output in the parent frame.

Requirements  

"_top"

opens the drill-down output in the full body of the window.

Requirements  

"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.

Requirements  

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  

HEIGHT=dimension

specifies the height of a graph.

dimension  
is a nonnegative number followed by one of these units of measure:
Table 6.10  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 are used by default.

**Tip**
If only the HEIGHT= option is specified, then the default aspect ratio 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 one of the ODS HTML* destinations is used.

Prior to SAS 9.4M5, 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 the inline SVG mode (the default value), the tooltips and links are written in the SVG portion of the document. (If you are using SAS 9.4M5, see the following Note.)

**Interaction**

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.

**Note**

Starting with SAS 9.4M5, image maps are supported with SVG output using HTML5. However, image maps are supported only when the HTML5 SVG mode is INLINE (the default value).

---

**IMAGENAME=**"filename"

specifies the base image filename. If more than one image is generated, each is assigned a filename that consists of 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.

*Note:* Starting with SAS 9.4M5, if a BY statement is in effect, you can uniquely name image files based on BY groups. To do this, insert #BY text into the image name. For more information, see “Substituting BY Line Values in the Image Name” on page 361.

**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 360

---

**LABELMAX=**\(n\)

specifies the maximum number of labeled areas before data label collision avoidance is disabled. For example, if there are more than 50 points to be labeled and LABELMAX=50, then collision avoidance is turned off and the labels are all displayed at the top right of the data points.

\[n\]

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 SAS 9.4M3, LEGENDAREAMAX= replaces
MAXLEGENDAREA=. However, MAXLEGENDAREA= is supported as an
alias. It is recommended that you use LEGENDAREAMAX=.

n
specifies a positive integer.

Alias  MAXLEGENDAREA=

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.

LINEPATTERNOBSMAX=n
specifies the maximum the number of vertices for a patterned line. If the number of
vertices exceeds the specified limit, the plot is not drawn.

n
specifies a positive integer.

Note: This feature applies to SAS 9.4M5 and to later releases.
<table>
<thead>
<tr>
<th><strong>Default</strong></th>
<th><strong>10000</strong></th>
</tr>
</thead>
</table>

**Notes**  
If the number of vertices exceeds the specified limit, the plot is not drawn and a note is written to the SAS log. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the log message, or change the line pattern to SOLID.  
This option applies only to graphs created with the Graph Template Language and with the ODS Graphics procedures.

**LOESSOBSMAX=n**  
specifies an upper limit for the number of observations that can be used with a loess plot.  
*Note:* Starting with SAS 9.4M3, 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**

**NBINSMAX=n**  
specifies the maximum number of bins that are processed for histograms. This option affects both computed and parameterized histograms, when available.  
*Note:* This feature applies to SAS 9.4M4 and to later releases.  

\[ n \]  
specifies a positive integer.  

**Default**  
**10000**

**Interaction**  
If you specify the number of bins in the histogram plot statement, that option is honored regardless of the NBINSMAX= option.

**NXYBINSMAX=n**  
specifies the maximum number of bins that are processed for heat maps. This option affects both computed and parameterized heat maps. The option applies to the product of the X and Y bins.  
*Note:* This feature applies to SAS 9.4M4 and to later releases.  

\[ n \]  
specifies a positive integer.  

**Default**  
**100,000**

**Interaction**  
If you specify the number of bins in the heat map statement, that option is honored regardless of the NXYBINSMAX= option.
**OBSMAX=n**

specifies the maximum number of observations that are processed. If the number of observations in the data set exceeds the value specified for OBSMAX=, the procedure step terminates with a log message.

\[ n \]

specifies a positive integer.

**Alias** MAXOBS=

**Default** 2 million observations

**Interaction** There are other ways to control the number of observations: `CASDATALIMIT=` system option, `DATALIMIT=` option in the CAS LIBNAME statement, and `DATALIMIT=` data set option. If the CAS data transfer limit is set lower than OBSMAX=, then OBSMAX= has no effect.

**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 365.

**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 362

**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** 1000

**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.

_**Tip**_ You can use the TOTALCELLMAX= option to control the total maximum number of cells in the graph.
**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 SAS 9.4M3 and to later releases.

**PUSH**

pushes the current ODS GRAPHICS settings to a stack.

**POP**

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 368.

**Note**

Order of specification is important when using the PUSH and POP options. For more information, see “About PUSH and POP” on page 367.

**Tip**

Use the SHOW option to show the current ODS GRAPHICS settings.

**See**

“Temporarily Saving and Restoring ODS GRAPHICS Settings” on page 367

---

**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 342**

**ANTIALIASMAX**

resets the ANTIALIASMAX= option to its default.

See **ANTIALIASMAX= on page 343**

**ATTRPRIORITY**

resets the ATTRPRIORITY= option to its default.

See **ATTRPRIORITY= on page 343**

**BORDER**

resets the BORDER= option to its default.
See \texttt{BORDER=} on page 343

\textbf{BYLINE}
 resets the \texttt{BYLINE=} option to its default.

See \texttt{BYLINE=} on page 344

\textbf{DATASKINMAX}
 resets the \texttt{DATASKINMAX=} option to its default.

See \texttt{DATASKINMAX=} on page 345

\textbf{DISCRETEMAX}
 resets the \texttt{DISCRETEMAX=} option to its default.

See \texttt{DISCRETEMAX=} on page 345

\textbf{DRILLTARGET}
 resets the \texttt{DRILLTARGET=} option to its default.

See \texttt{DRILLTARGET=} on page 345

\textbf{GROUPMAX}
 resets the \texttt{GROUPMAX=} option to its default.

See \texttt{GROUPMAX=} on page 346

\textbf{HEIGHT}
 resets the \texttt{HEIGHT=} option to its default.

See \texttt{HEIGHT=} on page 346

\textbf{IMAGEMAP}
 resets the \texttt{IMAGEMAP=} option to its default.

Note: Not all output destinations support this feature.

See \texttt{IMAGEMAP=} on page 347

\textbf{IMAGENAME}
 resets the \texttt{IMAGENAME=} option to its default.

\textit{Note:} This feature applies to \texttt{SAS 9.4M3} and to later releases.

See \texttt{IMAGENAME=} on page 348

\textbf{INDEX \texttt{<(positive-integer)>}}
 resets 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. \texttt{positive-integer} determines the suffix for the next subsequent image, and increments with each new image. This feature applies to \texttt{SAS 9.4M3} and to later releases.

See “Resetting the Image Name” on page 360

\textbf{LABELMAX}
 resets the \texttt{LABELMAX=} option to its default.
LABELMAX= on page 348

LABELPLACEMENT
specifies the label-placement algorithm to use for positioning labels in the graphs.

LABELPLACEMENT= on page 349

LEGENDAREAMAX
resets the LEGENDAREAMAX= option to its default.

LEGENDAREAMAX= on page 349

LOESSOBSMAX
resets the LOESSOBSMAX= option to its default.

LOESSOBSMAX= on page 350

NBINSMAX
resets the NBINSMAX= option to its default.

NBINSMAX= on page 350

NXYBINSMAX
resets the NXYBINSMAX= option to its default.

NXYBINSMAX= on page 350

OUTPUTFMT
resets the OUTPUTFMT= option to its default.

Note: This feature applies to SAS 9.4M3 and to later releases.

OUTPUTFMT= on page 351

PANELCELLMAX
resets the PANELCELLMAX= option to its default.

PANELCELLMAX= on page 351

SCALE
resets the SCALE= option to its default.

SCALE= on page 355

SCALEMARKERS
resets the SCALEMARKERS= option to its default.

SCALEMARKERS= on page 355

STACKDEPTHMAX
resets the STACKDEPTHMAX= option to its default.

Note: This feature applies to SAS 9.4M3 and to later releases.

STACKDEPTHMAX= on page 357

SUBPIXEL
resets the SUBPIXEL option to its default.
Note: This feature applies to SAS 9.4M3 and to later releases.

See **SUBPIXEL on page 357**

TIPMAX

resets the TIPMAX= option to its default.

See **TIPMAX = on page 358**

WIDTH

resets the WIDTH= option to its default.

See **WIDTH= on page 359**

**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.

Alias **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.
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 367.

Note: This feature applies to SAS 9.4M3 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:

ODS Graphics Settings
--------------
Output format: STATIC
By line: NOBYLINE
Antialias: ON
Maximum Loess observations: 5000
Maximum stack depth: 1024
Stack depth: 0
MaxObs: 2000000
Maximum Histogram Bins: 10000
Maximum Heatmap Bins: 100000
Maximum Obs for Patterned Lines: 10000
Maximum Total Cells per BY-group: 2000

If you have specified the settings for one or more options, then SHOW includes those settings along with the defaults.

Order of specification is important when using the SHOW option. For example, the following statement shows the current settings and then sets the NOBORDER option.

ods graphics / show noborder;

However, the following statement sets the NOBORDER option and then shows the settings. The NOBORDER setting is shown in the log along with the other settings that are in effect.

ods graphics / noborder show;

The following statement resets all settings. It then sets the image width and shows the default settings along with the specified width.

ods graphics / reset=all width=5in show;

Here are the default values plus the image width, as displayed in the SAS log:
ODS Graphics Settings
----------------------
Output format: STATIC
By line: NOBYLINE
Antialias: ON
Maximum Loess observations: 5000
Image width: 5in
Maximum stack depth: 1024
Stack depth: 0
Maximum Histogram Bins: 10000
Maximum Heatmap Bins: 100000

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.
ods graphics / push reset=all show;
ods graphics / pop;

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 SAS 9.4M3 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:
ods graphics / stackdepthmax=0 reset=stackdepthmax;

You can use any of the following commands to reset the stack to its default maximum depth:
reset=stackdepthmax
reset=all
reset
stackdepthmax=1024

See “Managing the Stack Depth” on page 368

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 SAS 9.4M3 and to later releases.
SUBPIXEL
always uses subpixel rendering, when applicable, for rendering lines and bars.

NOSUBPIXEL
never 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
never 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 data tip 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 SAS 9.4M3, 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 SAS 9.4M3, the TIPMAX= option specifies the maximum number of data-tip 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 data-tip 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
TOTALCELLMAX=\textit{n}

specifies the maximum number of total cells in a graph where the number of cells is determined dynamically by classification variables.

\textit{Note:} This feature applies to SAS 9.4M5 and to later releases.

If the number of cells exceeds the specified limit, the graph is not drawn. This option is useful when multiple classification variables or large amounts of data result in a large number of cells.

\textit{n}

specifies a positive integer.

Default 2000

Notes If the number of cells exceeds the limit specified for TOTALCELLMAX=, the graph is not drawn and an error message is written to the SAS log. The error message includes a suggested value for TOTALCELLMAX=. You can use the TOTALCELLMAX= option to increase the maximum number of cells. Note, however, that the processing time to render the graph increases with an increase in the maximum number of cells.

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.

Tip You can use the PANELCELLMAX= option to control the maximum number of cells in a graph panel.

WIDTH=\textit{dimension}

specifies the width of any graph.

\textit{dimension}

is a nonnegative number followed by one of these units of measure:

\begin{table}[h]
\centering
\begin{tabular}{l l}
\hline
\textbf{cm} & Centimeters \\
\textbf{in} & Inches \\
\textbf{mm} & Millimeters \\
\textbf{pct or %} & Percentage \\
\textbf{pt} & Point size (72 points = 1 inch) \\
\textbf{px} & Pixels \\
\hline
\end{tabular}
\end{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:

```sas
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:

```sas
ods graphics off;
```

*Note:* ODS Graphics is ON by default for procedures SGPLOT, SGPANEL, SGSATTER, 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 ODS 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:

```sas
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:

```sas
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. This feature applies to SAS 9.4M3 and to later releases.

For a usage example, suppose that you are developing a paper or a presentation 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:

```sas
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:

```
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:

```
ods graphics / reset=index(101) imagename="MyName";
```

The next generated image is named MYNAME101.

**Substituting BY Line Values in the Image Name**

*Note:* This feature applies to SAS 9.4M5 and to later releases.

If a BY statement is in effect for the data, then you can uniquely name image files based on BY groups. You do this by inserting any of the #BY* substitution items into the image filename. You can specify BY lines, BY variable values, or BY variable names for IMAGENAME=. You can combine the #BY* substitution item with other text. To do this, insert the #BY* item in the specified text string at the position where you want the substitution text to appear.

As explained in the IMAGENAME= description, the file extension is automatically generated based on the OUTPUTFMT= option.

Here are descriptions of the #BY* substitution items:

**#BYLINE**

substitutes the entire BY line without leading or trailing blanks for #BYLINE in the filename string. The BY line uses the format *variable-name*=*value*. However, the “=” character in the BY line is replaced with an underscore character.

**Examples**

The following example specifies the BY line as the image filename:

```
imagename="#byline"
```

The following example specifies the BY line preceded by the text `sales`:.

```
imagename="sales_#byline"
```

**#BYVALn | #BYVAL(BY-variable-name)**

substitutes the current value of the specified BY variable for #BYVAL in the filename string. Specify the variable with one of these:

`n`

specifies a variable by its position in the BY statement. You must provide the position even if your BY statement contains only one variable. Otherwise, no substitution occurs.

`BY-variable-name`

specifies a variable from the BY statement by its name. `BY-variable-name` is not case sensitive.
Examples

The following example specifies the first variable in the BY statement. The value of that variable is used for the image filename.

```
imagename="#byval1"
```

The following example specifies the second variable in the BY statement preceded by the text `sales_`:

```
imagename="sales_#byval2"
```

The following example specifies the YEAR variable in the BY statement. The value of that variable is used for the image filename:

```
imagename="#byval(year)"
```

The following example specifies the YEAR variable in the BY statement followed with the text `_sales`:

```
imagename="#byval(year)_sales"
```

Note: When using this feature, consider any filename length restrictions that might apply to your platform host. The filename length can vary greatly depending on the input data.

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>
<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>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>
<tr>
<td>ODS PRINTER</td>
<td>OUTPUTFMT=PCL (for PCL output)</td>
</tr>
<tr>
<td>ODS RTF</td>
<td>OUTPUTFMT=EMF</td>
</tr>
<tr>
<td>ODS Measured RTF</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 SAS 9.4M2, 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 SAS 9.4M3, 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.

Starting with SAS 9.4M5, vector graphics output can or cannot be generated in the output formats as indicated in the following table:

<table>
<thead>
<tr>
<th>Case</th>
<th>EMF</th>
<th>PCL</th>
<th>PDF</th>
<th>PS</th>
<th>SVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surface plots</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Gradient fill contour plots</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>3-D bivariate histograms</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Continuous legends without data or fill transparency</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Continuous legends with data transparency</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Data skins</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Data skins with transparency</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Supported File Types for Output Destinations

The following table lists all of the supported file types for some ODS 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>Note: EPUB3 was added in SAS 9.4M1.</td>
</tr>
<tr>
<td></td>
<td>Note: Starting with SAS 9.4M3, 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, 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, EPSI, 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>
</tbody>
</table>
Description of Supported File Types

The following table provides descriptions of the supported file types for ODS output destinations.

Table 6.15  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>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>File Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>JFIF (JPEG File Interchange Format)</td>
<td>Supports JPEG image compression. JFIF 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>
<tr>
<td>PSL (PostScript)</td>
<td>PostScript</td>
</tr>
<tr>
<td>STATIC</td>
<td>Chooses the best image format for the current ODS destination.</td>
</tr>
<tr>
<td>SVG (Scalable Vector Graphics)</td>
<td>Is an XML language for describing two-dimensional vector graphics.</td>
</tr>
<tr>
<td>TIFF (Tagged Image File Format)</td>
<td>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).</td>
</tr>
<tr>
<td>XBM</td>
<td>X Window Bitmap</td>
</tr>
<tr>
<td>XPM</td>
<td>X Window Pixmap</td>
</tr>
</tbody>
</table>

**Temporarily Saving and Restoring ODS GRAPHICS Settings**

This feature applies to SAS 9.4M3 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 write 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 368. For each PUSH option, you can specify a
POP option. 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.

```sas
ods graphics / noborder push;
```

A subsequent POP option restores the pushed settings including NOBORDER.

However, the following statement pushes the current custom settings and then sets the
NOBORDER option.

```sas
ods graphics / push noborder;
```

Here, the subsequent POP option restores whatever border setting was in effect when the
PUSH option was specified.

**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 options 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 option is
ignored.

For example, the PUSH option is ignored in the following statement:

```sas
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 option is honored.

```sas
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 STACKDEPTMAX= 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 except SAS Viya
- **Category:** ODS: Third-Party Formatted
- **Default:** The default style for Markup family destinations is HTMLBlue.

**Syntax**

```ods HTML3</(\id=>identifier)><action>;```

```ods HTML3</<\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

**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

STYLE$HEET= '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 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 332

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 793

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 42.

See “ODS SHOW Statement” on page 806

**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. ODS assumes that if the anchor name ends in a number, then the number should be treated as a starting point for incrementing the 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.

Similarly, if you specify ANCHOR="TABULATE1", then ODS names the first anchor tabulate1. The second anchor is named tabulate2.

**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 Global 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 393.

**(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 394.

**(NO_TOP_MATTER)**

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 394.

**(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 395.

**(URL='Uniform-Resource-Locator')**

See For complete documentation about the URL= suboption, see “(URL='Uniform-Resource-Locator')” on page 395.

**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 538.

**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.

**Tip**

When you use the ODS HTML statement with the FILE= option to specify a fully-qualified path with a directory and a file name, the best practice for generating HTML files is to separate the path from the file. To accomplish this, use a combination of the PATH= option, which specifies the directory, and the FILE= option, which names the actual file. For example:
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 393.

(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.
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 Global 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 393.

**(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 394.

**(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 394.

**(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 395.
(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 395.

**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 Global 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 548

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 Global 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 393.

**(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 394.

**(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 394.

**(TITLE=’title-text’)**
is a file that indicates the title.

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 395.

(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 395.

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 541

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.

Requirements You must enclose aggregate-file-storage-location in quotation marks.
The specified aggregate-file-storage-location must exist in the file system.

**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 Global 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 617

**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:

```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.
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 837

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 576

Example “Example 1: Creating an ODS Package” on page 580

PAGE= 'file-specification' <(suboption(s))>
onopens 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 Global 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 393.

**(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 394.

**(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 394.

**(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 395.

(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 395.

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 Global Statements: Reference.

---

**libref.catalog**

specifies a SAS catalog to write to.

**See**

For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Global 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 Global 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 393.

(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 394.

(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 394.

(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.
For complete documentation about the TITLE= suboption, see “(TITLE='title-text’)” on page 395.

(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.

For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator’)” on page 395.

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.

For information about events and event variables, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in SAS Output Delivery System: Procedures Guide.

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 1109 in PROC TEMPLATE.

If you do not specify DYNAMIC, then ODS sets the value of HTMLCONTENTTYPE= for writing to a file.

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=
Requirements

You must enclose DYNAMIC in parentheses.

You must specify DYNAMIC next to the file-specification specified by the BODY=, CONTENTS=, PAGE=, FRAME=, or STYLESEET= 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 STYLESEET= 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 STYLESEET= 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 543

(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 546
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 48.

See Also

• Appendix 3, “ODS and the HTML Destination,” on page 1201
• “Changing SAS Registry Settings for ODS” on page 44

Statements

• “ODS MARKUP Statement” on page 509
• “ODS HTML Statement” on page 396

ODS HTML Statement

Opens, manages, or closes the HTML destination, which produces HTML 4.0 output in the SAS Windowing environment. This output can contain embedded style sheets.

Valid in: Anywhere
Category: ODS: Third-Party Formatted
Default: The default style for markup family destinations is HTMLBlue. The default output for the HTML destination is HTML5 in the SAS Studio client for SAS Viya.

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:

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

**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**  
Exclude output objects from the destination

**FRAME='file-specification'**  
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'**  
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'**  

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 332

**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 793

**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 42.

**See**  “ODS SHOW Statement” on page 806

**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. ODS assumes that if the anchor name ends in a number, then the number should be treated as a starting point for incrementing the 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.

Similarly, if you specify `ANCHOR="TABULATE1"`, then ODS names the first anchor `tabulate1`. The second anchor is named `tabulate2`.

**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:

```
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 Global 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.
For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 423.

(NO_BOTTOM_MATTER)

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

For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 423.

(NO_TOP_MATTER)

For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 424.

(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.

For complete documentation about the TITLE= suboption, see “(TITLE=’title-text’)” on page 424.

(URL= ’Uniform-Resource-Locator’)  

For complete documentation about the URL= suboption, see “(URL= ’Uniform-Resource-Locator’)” on page 424.

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 538.

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.

Tip

When you use the ODS HTML statement with the FILE= option to specify a fully-qualified path with a directory and a file name, the best practice for generating HTML files is to separate the path from the file. To accomplish this, use a combination of the PATH= option, which specifies the directory, and the FILE= option, which names the actual file. For example:

```plaintext
ods html path="c:\temp" file="myfile.html";
proc print data=sashelp.class;
run;
ods html close;
```

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 44.

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 423.

(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 424.
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 APPLETLLOC= 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 Global 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 423.

**(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 423.

**(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 424.

**(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 424.

**(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 Global 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.
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

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.

For complete documentation about ODS and Cascading Style Sheets, see *SAS Output Delivery System: Advanced Topics*

“Example 6: Applying a CSS File to ODS Output” on page 548

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 365.

**Table 6.16  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>Excel</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>PowerPoint</td>
<td>PNG</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 44.


**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 Global 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 423.

**(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 423.

**(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 424.

**(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 424.

**(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.
For complete documentation about the URL= suboption, see “(URL=’Uniform-Resource-Locator’)” on page 424.

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 541

<table>
<thead>
<tr>
<th>GFOOTNOTE</th>
<th>NOGFOOTNOTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>controls the location where footnotes are printed in the graphics output.</td>
<td></td>
</tr>
</tbody>
</table>

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.

Requirements
You must enclose aggregate-file-storage-location in quotation marks.

The specified aggregate-file-storage-location must exist in the file system.

fileref
is a file reference that has been assigned to an aggregate storage location. Use the FILENAME statement to assign a fileref.
If you specify a fileref in the GPATH= option, then ODS does not use information from the GPATH= option when it constructs links.

For information about the FILENAME statement, see “FILENAME Statement” in SAS Global 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.

You must enclose Uniform-Resource-Locator in quotation marks.

NONE

specifies that no information from the GPATH= option appears in the links or references.

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.

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 617

**IMAGE_DPI=**
specifies the image resolution for graphical output.

**Alias**
DPI=

**Default**
96

**Restriction**
This option applies only 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.

**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:**

```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:
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 837

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.
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.

- 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 Global 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 423.

- (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 423.
(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 424.

(TITLE="title-text")

does not insert 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 424.

(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 424.

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 Global Statements: Reference.

libref:catalog
specifies a SAS catalog to write to.

See For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Global 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:

\texttt{RECORD\_SEPARATOR= '0D0A'x}

\textit{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:

\texttt{RECORD\_SEPARATOR= '0D25'x}

\textbf{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.

\textit{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 \texttt{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.

\textbf{Aliases}
RECSEP=

\begin{verbatim}
RS=
\end{verbatim}

\texttt{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.

\texttt{See}
For details about using the ODS Graphics Editor to create SGE files, see \textit{SAS ODS Graphics Editor: User’s Guide}.

\texttt{Example}
ods html sge=on;
proc sgplot data=sashelp.class;
scatter x=weight y=height;
run;

\texttt{STYLE=} \texttt{style-template}
specifies the style template to use in writing the output files.

\textit{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.

\textbf{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.

```
STYLE= '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 Global 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.
For complete documentation about the DYNAMIC suboption, see “(DYNAMIC)” on page 423.

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

For complete documentation about the NO_BOTTOM_MATTER suboption, see “(NO_BOTTOM_MATTER)” on page 423.

(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.

For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 424.

(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.

For complete documentation about the TITLE= suboption, see “(TITLE=’title-text’)” on page 424.

(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.

For complete documentation about the URL= suboption, see “(URL=’Uniform-Resource-Locator’)” on page 424.

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 546

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)

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 336

**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 1109 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 543

(**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 546

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 23.

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.
These are the options supported by this tagset.

Sample usage:

```plaintext
ods html options(doc='Quick');
ods html options(header_dots='yes' summary_byvars='yes');
```

**Doc:** No default value.

- **Help:** Displays introductory text and options.
- **Quick:** Displays available options.

**header_data_associations:** Default Value 'no'

Associates data cells and header cells by adding an ID attribute to each header cell and listing the IDs of associated headers in a HEADERS attribute added to each data cell. (PROC REPORT only)

**header_dots:** Default Value 'no'

- Puts hidden dots before the text in all table headers

**summary_as_caption:** Default Value 'no'

- Causes a table caption to be created from the table summary

**summary_byvars:** Default Value 'no'

- Adds a list of by variable names to the table summary

**summary_byvals:** Default Value 'no'

- Adds the values of the by variables along with the names in the table summary

This works with summary byvars but not without.

**summary:** Default Value ''

- Text for the table summary

**summary_prefix:** Default Value ''

- Text to place at the beginning of table summary

**summary_suffix:** Default Value ''

- Text to place at the end of table summary

**page_break:** Default Value 'yes'

- If yes, the usual pagebreak style attribute will be used to create what becomes the page separator. Usually that is an HR line.
- If no, then no pagebreak will be output.
- If anything else, the value given will be output as the pagebreak.

**css_table:** Default Value 'no'

---

**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

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.

ods html file="test.html" options(pagebreak='no');

Print only two observations of the data set.

options obs=2;

Print the data set Sashelp.Class. The PROC PRINT statement prints the Sashelp.Class data set.

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.

proc print data=Sashelp.Class;
run;

Output

Specify the PAGEBREAK="NO" suboption if you want fewer pages of output.

Output 6.36 PAGEBREAK= Suboption Set to NO in HTML

<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>
</tbody>
</table>

The SAS System

<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>
</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**

```plaintext
proc sort data=grain_production;
  by year country type;
run;
ods html body='grain-body.htm'
  contents='grain-contents.htm'
  frame='grain-frame.htm'
  page='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;
  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.

proc report data=grain_production nowindows;
by year;
column country type kilotons;
declare country / group width=14 format=$cntry.;
declare type / group 'Type of Grain';
declare 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.
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
```r
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.

```plaintext
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.

```plaintext
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.

```plaintext
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.

```plaintext
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.

```plaintext
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.
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.

```plaintext
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.

```plaintext
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.

```plaintext
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).

```plaintext
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).

```plaintext
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.

```plaintext
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.

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 6.38  HTML Output with Appended HTML*

---

**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>8,660</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**

```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 553 and “OPTIONS (DOC= | <suboption(s)> )” on page 416.

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.

```plaintext
title;
proc print data=sashelp.class;
run;
quit;
ods html close;
```
Example 5: Removing the Horizontal Rule between Procedures 439

Output

Output 6.39  Default Output with Page Break

![Student Correlation Graph]

<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 6.40  Output with No Page Break

See Also

- Appendix 3, “ODS and the HTML Destination,” on page 1201

Statements

- “ODS MARKUP Statement” on page 509
- “ODS Tagset Statement” on page 807
ODS HTML5 Statement

Opens, manages, or closes the HTML5 destination, which produces HTML 5.0 output that contains embedded style sheets.

Valid in: Anywhere except SAS Viya
Category: ODS: Third-Party Formatted
Defaults: The default style is HTMLEncore.
SVG is the default device driver for the ODS HTML5 destination.
Restriction: The ODS HTML5 destination does not support table cell padding or spacing. Use CSS for ODS HTML5 to produce padding and spacing.
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:

```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 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 1: ODS Graphics SVG Graph in an HTML5 File” on page 469
“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

```sas
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
ACCESSIBLE GRAPH | NOACCESSIBLE GRAPH
Add accessibility metadata to graphs that are created by ODS Graphics
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

**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

**CONTENTS= 'file-specification' <(suboption(s))>**
Open the HTML5 destination and specify the file that contains a table of contents for the output

**CSSSTYLE='file-specification' <(media-type1 <...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

**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 ( 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 332

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 793

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 42.

See: “ODS SHOW Statement” on page 806

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. ODS assumes that if the anchor name ends in a number, then the number should be treated as a starting point for incrementing the 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.

Similarly, if you specify ANCHOR= "TABULATE1", then ODS names the first anchor tabulate1. The second anchor is named tabulate2.

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.
You must enclose "anchor-name" in quotation marks.

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.

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:

```
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.

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 Global 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 465.

(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 466.

(NO_TOP_MATTER)

See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 466.

(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 467.

(URL='Uniform-Resource-Locator')

See For complete documentation about the URL= suboption, see “(URL='Uniform-Resource-Locator’)” on page 467.

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 538.

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.

Tip
When you use the ODS HTML statement with the FILE= option to specify a fully-qualified path with a directory and a file name, the best practice for generating HTML files is to separate the path from the file. To accomplish this, use a combination of the PATH= option, which specifies the directory, and the FILE= option, which names the actual file. For example:

```plaintext
ods html path="c:\temp" file="myfile.html";
proc print data=sashelp.class;
run;
ods html close;
```

**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 44.

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.

**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 465.

(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 467.

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 Global 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 465.

(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 466.

(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 466.

(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 467.

(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 467.

CSSSTYLE='file-specification' <(media-type1 <...media-type-10>)>

specifies a cascading style sheet to apply to your output.

See For complete documentation about the CSSSTYLE= suboption, see “CSSSTYLE='file-specification' <(media-type1 <...media-type-10>)>” on page 469.

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 DATA Step 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.

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 548.

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 365.

**Table 6.17  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>Excel</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>PowerPoint</td>
<td>PNG</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 44.

**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*. 
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 Global 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 465.

  (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 466.

  (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.
For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 466.

(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.

For complete documentation about the TITLE= suboption, see “(TITLE="title-text")” on page 467.

(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.

For complete documentation about the URL= suboption, see “(URL='Uniform-Resource-Locator')” on page 467.

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 541

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.

Requirements

You must enclose aggregate-file-storage-location in quotation marks.

The specified aggregate-file-storage-location must exist in the file system.

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 Global 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 617

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:

```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:**

```sas
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. These values must be enclosed in parentheses.

BITMAP_MODE= | GRAPH_BITMAP_MODE= | STYLE_BITMAP_MODE=
specifies how all bit mapped images are inserted into the HTML5 document. These values must be enclosed in parentheses.

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).

```html
<object data="c:\Public\arrow.png"/>
```

**SVG_MODE= | GRAPH_SVG_MODE= | STYLE_SVG_MODE=**
specifies how all SVG images are inserted into the HTML5 document. These values must be enclosed in parentheses.

Use the GRAPH_SVG_MODE and the STYLE_SVG_MODE to insert your style images and your graph images differently.

```sas
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.

```html
<embed src="c:\Public\arrow.svg"/>
```

IMG
creates an HTML5 `<img>` tag that displays the image file.

```html
<img src="c:\Public\arrow.svg"/>
```

**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.

```html
<a href="c:\Public\arrow.svg" My Arrow SVG</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.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. These values must be enclosed in parentheses.

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=mscorlib 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 576

Example “Example 1: Creating an ODS Package” on page 580

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 Global 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 465.

(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 466.

(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 466.

(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 467.
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 467.

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.

'value'='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.
libref.catalog
 specifies a SAS catalog to write to.

See For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Global 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;
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 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 Global 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 465.

(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 466.

(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 466.
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 467.

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 467.

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 546

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.

See For information about events and event variables, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in SAS Output Delivery System: Procedures Guide.

specifies the translation table to use when transcoding a file for output.


Suboptions

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 1109 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=
### NO_BOTTOM_MATTER

Specify 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.

**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.
- 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.

**Tip**
- If you append text to an external file, you must use a FILENAME statement with the appropriate option for the operating environment.

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.

**Interactions**
- 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

\((\text{TITLE}='\text{title-text}')\)

inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar.

\text{title-text}

is the text in the metadata of a file that indicates the title.

Requirements

You must enclose \text{TITLE}= in parentheses.

You must enclose \text{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 543

\((\text{URL='Uniform-Resource-Locator'})\)

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 \text{URL}='\text{Uniform-Resource-Locator}' in parentheses.

You must enclose \text{Uniform-Resource-Locator} in quotation marks.

You must specify \text{URL}='\text{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 546

\((\text{OUTLINE}='\text{ON}' | '\text{OFF}')\)

controls whether table of contents metadata is included in the ODS HTML5 body file.
"OFF"
  specifies that no table of contents metadata is included in the ODS HTML5 body file. This suppresses the display of procedure titles. This is useful when creating headings with PROC ODSTEXT for accessibility.

Alias   "NO"

ON
  specifies that table of contents metadata is included in the ODS HTML5 body file.

Alias   "YES"

Default "OFF"

Example “Example 2: Creating a Custom Table of Contents with Anchors” on page 471

Details

Accessibility with HTML5
In the SAS 9.4M4, the ODS HTML5 destination together provides the best accessibility of the ODS destinations because there are many tags and attributes that allow for compliance with W3C Web Content Accessibility Guidelines (WCAG), version 2.0. For complete information about creating accessible output with ODS, see Creating Accessible SAS® 9.4 Output Using ODS and ODS Graphics.

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 97.

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 23.

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.

**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

**Examples**

*Example 1: 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.
Example 2: Creating a Custom Table of Contents with Anchors

Features:
- ODS HTML5 Statement

Options
- ANCHOR=
- CONTENTS=

Other features:
- PROC PRINT
- PROC SORT

Details
Options can be added to the ODS HTML5 statement to help with accessibility. To generate a Table of Contents that includes links, specify options ANCHOR= and CONTENTS=.

Program

data a;
  set sashelp.demographics (where=(region='AMR'));
  fmt pop = pop;
run;
proc sort;
  by pop;
run;
proc format;
  value population
  low - <1000000 = Small
  1000000 - <10000000 = Medium
  10000000 - <100000000 = Large
  100000000 - <1000000000 = Huge
;
run;
ods html5 (id=web) style=daisy path='file-path'
  file='file-name'
  {title='Create Links to Output Tables'}
options(outline='no')
  anchor='myreport';
proc odstext contents=''
  p "<h1>North and South American Countries by population</h1>";
  p "<h2><a href='#myreport1'>Small populations</a></h2>";
  p "<h2><a href='#myreport2'>Medium populations</a></h2>";
  p "<h2><a href='#myreport3'>Large populations</a></h2>";
  p "<h2><a href='#myreport4'>Huge populations</a></h2>";
run;
proc print data=a label;
  var name pop gni;
  id name;
  format fmtpop population.;
  by fmtpop;
run;
ods html5 close;

Program Description

Create your data set.

data a;
  set sashelp.demographics {where=(region='AMR')};
  fmtpop = pop;
run;

Sort your data set.

proc sort;
  by pop;
run;

Format your data. Establish the format for your data.

proc format;
  value population
    low - <1000000 = Small
    1000000 - <10000000 = Medium
    10000000 - <100000000 = Large
    100000000 - <1000000000 = Huge

Open the HTML5 destination. The ODS HTML5 destination statement sets the ODS environment. The STYLE= option specifies the style Daisy, which is recommended for accessible output. The (ID=WEB) option is required for use with SAS Studio only. The (TITLE=) suboption inserts into the metadata of a file the text string that you specify as the text to appear in the browser window title bar. It is recommended that you always specify (TITLE=) for accessible output. The OUTLINE=NO option specifies that no table of contents metadata is included in the ODS HTML5 body file. This enables
procedure titles to be suppressed in SAS Studio. If you do not specify OUTLINE=NO, procedure titles will show up as headings in the HTML outline that is read by screen readers.

```sas
ods html5 (id=web) style=daisy path='file-path'
    file='file-name'
    {titles='Create Links to Output Tables'}
    options(outline='no')
    anchor='myreport';
```

**Create a Table of Contents with links.** Four links are created that point to each table (#myreport1 through #myreport4). There are four links because the BY statement is used in PROC PRINT and the Fmtpop variable is formatted by the custom "population" format, which has four values. The CONTENTS= option is set to null for PROC ODSTEXT so that there is not an extra item in the table of contents generated by SAS.

```sas
proc odstext contents='';
    p "<h1>North and South American Countries by population</h1>";
    p "<h2><a href='#myreport1'>Small populations</a></h2>";
    p "<h2><a href='#myreport2'>Medium populations</a></h2>";
    p "<h2><a href='#myreport3'>Large populations</a></h2>";
    p "<h2><a href='#myreport4'>Huge populations</a></h2>";
run;
```

**Print the data set.** The ID statement is used so that the country names will be formatted as row headers.

```sas
proc print data=a label;
    var name pop gni;
    id name;
    format fmtpop population.;
    by fmtpop;
run;
```

**Close the HTML5 destination.** The ODS HTML5 CLOSE statement closes the HTML5 destination and all the files that are associated with it.

```sas
ods html5 close;
```

**HTML5 Output**

This output is created using the ODS HTML5 destination. It creates more accessible output by adding anchors to the table of contents.
Adding Anchors to the Table of Contents for Better Accessibility

Output 6.42  Adding Anchors to the Table of Contents for Better Accessibility

**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 except SAS Viya
- **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 485.

**Example:**
```plaintext
ods layout absolute y=1.25in x=1in width=6in;
ods region;
proc print data=sashelp.class;
run;
```
**Syntax**

**ODS LAYOUT ABSOLUTE**<option-1><option-2 ...>

**Summary of Optional Arguments**

- **DESCRIPTION**="text-string"
  
  Specify alternate text for your layout containers and regions. For accessibility, the text should convey the meaning of the container or region. The alternate text will be read by assistive technology such as a screen reader.

- **HEIGHT**=dimension
  
  Specify a vertical height of the layout. It is not recommended that you use pixels because of adverse dependencies on resolution that can differ between destinations.

- **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**

- **DESCRIPTION**="text-string"
  
  Specifies alternate text for layout containers and regions. This option helps to make reports accessible for people with a wide range of abilities. For accessibility, the text should convey the meaning of the container or region. The alternate text will be read by assistive technology such as a screen reader.

- **HEIGHT**=dimension
  
  Specifies the vertical height of the layout. 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
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:

- 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

```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.

Example

```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
Example 1: Absolute Layout Using the Region Container

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

```r
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 height=1.75in;
ods text="Approximately 5500 different sports and outdoor products
are offered at Orion Star. Products are sold in volumes that reflect
...";
```

Example 1: Absolute Layout Using the Region Container

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

```r
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 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.

Products are organized in a hierarchy consisting of three levels:

- Product Line
- Product Category
- Product Group

Where we generate our profit...

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

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

Our Values
To stay Customer focused, Swift and Agile, Innovative, and Trustworthy.
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=center 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.
The Orion Star Sports & Outdoors Company is a fictional international retail company that sells sports and outdoor products. The headquarters is based in the United States. Retail stores are situated in a number of other countries including Belgium, Holland, Germany, the United Kingdom, Denmark, France, Italy, Spain, and Australia.

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

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

Products are organized in a hierarchy consisting of three levels:

- Product Line
- Product Category
- Product Group

Where we generate our profit

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.

Create the fourth region, “What we sell”, using ODS TEXT. Add the star image and the text for the second item in the list.

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.

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.

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.
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.

```sas
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;
declare product_category / group;
declare 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.

```sas
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.

```sas
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.

```sas
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.

```sas
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.
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 6.43 ODS Absolute Layout – Company Information
See Also

ODS Statements

- “ODS REGION Statement, Absolute” on page 485
- “ODS LAYOUT END Statement” on page 504

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.

Valid in: Anywhere except SAS Viya

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).

Note: This ODS REGION statement is inserted 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.

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

`ODS REGION < option-1>< option-2 ...>`

Summary of Optional Arguments

- `DESCRIPTION="text-string"`  
  Specify alternate text for your layout containers and regions.

- `HEIGHT=dimension`  
  Specify a vertical height of the region.

- `STYLE=<style-element-name> <[style-attribute-specification(s)]>`  
  Specify 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

**DESCRIPTION=“text-string”**
specifies alternate text for layout containers and regions. This option helps to make reports accessible for people with a wide range of abilities. For accessibility, the text should convey the meaning of the container or region. The alternate text will be read by assistive technology such as a screen reader.

Valid in HTML5 and PDF

Alias DESC

**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=[bgcolor=yellow];
ods region style=[bgcolor=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=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

```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 474
- “**ODS LAYOUT END Statement**” on page 504
ODS LAYOUT GRIDDED Statement

Enables you to arrange output dynamically in a two-dimensional grid structure.

Valid in: Anywhere except SAS Viya
Category: ODS: Output Control
Requirement: The ODS LAYOUT GRIDDED statement must be used with the ODS LAYOUT END statement.

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 | EXPPLICIT | 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.
DESCRIPTION="text-string"
   Specify alternate text for your layout containers and regions.
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)]>
Specify 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:

- **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**

```plaintext
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);
```
**DESCRIPTION=**"text-string"

specifies alternate text for layout containers and regions. This option helps to make reports accessible for people with a wide range of abilities. For accessibility, the text should convey the meaning of the container or region. The alternate text will be read by assistive technology such as a screen reader.

Valid in HTML5 and PDF

Alias DESC

**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 gridded height=7in;
   proc print data=sashelp.class; run
ods layout end;
```

**ORDER_TYPE=** <ROW_MAJOR | COLUMN_MAJOR>

populates the grid by rows or columns.

**COLUMN_MAJOR**

specifies that the gridded layout first populates all regions in the first column before moving on to the next column.
ROW_MAJOR
specifies that the gridded layout first populates all regions in the row before moving on to the next row.

Default
ROW_MAJOR

Restrictions
The ORDER_TYPE= option is valid only in PRINTER destinations.

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.

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;
       var height weight;
   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=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;

\(Y=\text{dimension}\)

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
ods layout gridded y=10cm;
   proc print data=sashelp.class; run
ods layout end;

\section*{Details}

\textbf{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.

\textit{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
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='gttitle';
  title 'This is the PROCEDURE TITLE';
  footnote 'This is the PROCEDURE FOOTNOTE';
run;
quit;
```
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='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.

```
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 6.44  ODS Gridded Layout – Changing Titles and Footnotes**

![Diagram](image)

**See Also**

**ODS Statements**

- “ODS REGION Statement, Gridded” on page 499
- “ODS LAYOUT END Statement” on page 504

---

**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.

**Valid in:** Anywhere except SAS Viya
**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 the ODS destination for PowerPoint.

**Note:** This ODS REGION statement is inserted between the ODS LAYOUT GRIDDED statement and the ODS LAYOUT END statement. The ODS LAYOUT GRIDDED statement manages the ODS LAYOUT destination.

**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.

- **DESCRIPTION="text-string"**
  Specify alternate text for your layout containers and regions.

- **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)]>**
  Specify 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.

```plaintext
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.
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.
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;

DESCRIPTION="text-string"
specifies alternate text for layout containers and regions. This option helps to make reports accessible for people with a wide range of abilities. For accessibility, the text should convey the meaning of the container or region. The alternate text will be read by assistive technology such as a screen reader.

Valid in HTML5 and PDF

Alias DESC

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.
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

```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

```plaintext
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 489
- “ODS LAYOUT END Statement” on page 504

ODS LAYOUT END Statement

Ends a gridded or absolute layout statement block.

Valid in: Anywhere except SAS Viya

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:

```plaintext
ods layout gridded;
  ods region;
    proc print data=sashelp.class;
      run;
  ods layout end;
```

Syntax

```plaintext
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 except SAS Viya

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> ;

ODS LISTING <DATAPANEL=number | DATA | PAGE > <FILE=file-specification> ;

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 332

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 793

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 806
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.

**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 6.18** Default Devices for 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>Excel</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>
</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 DATA Step 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 DATA Step 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 576

`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.
The default style for the LISTING destination is Listing.

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 23.


**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 except SAS Viya
- **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.
SEE

**Restriction**
A destination must be open for this action to take effect.

See  
“ODS EXCLUDE Statement” on page 332

**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 793

**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 42.

See  
“ODS SHOW Statement” on page 806

**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. ODS assumes that if the anchor name ends in a number, then the number should be treated as a starting point for incrementing the 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.

Similarly, if you specify ANCHOR="TABULATE1", then ODS names the first anchor `tabulate1`. The second anchor is named `tabulate2`.

**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.

**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 Global Statements: Reference.

element.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 536.

(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 536.

(NO_TOP_MATTER)
See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 537.

(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 537.

(URL= 'Uniform-Resource-Locator')
See For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator’)” on page 537.
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 538.

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.

Tip  When you use the ODS HTML statement with the FILE= option to specify a fully-qualified path with a directory and a file name, the best practice for generating HTML files is to separate the path from the file. To accomplish this, use a combination of the PATH= option, which specifies the directory, and the FILE= option, which names the actual file. For example:

```sas
ods html path="c:\temp" file="myfile.html";
proc print data=sashelp.class;
run;
ods html close;
```

**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.
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 536.

(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'

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 Global 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 536.

(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 536.

(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 537.

(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 537.

(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 537.

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 Global 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 548

**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 365.

<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>

**Table 6.19 Default Devices for ODS Output Destinations**
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 44.

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.

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.
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 Global 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 536.

(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 536.

(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 537.

(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 537.

(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 537.

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 541

GFOOTNOTE | NOGFOOTNOTEcontrols 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.

**Requirements**

You must enclose *aggregate-file-storage-location* in quotation marks.

The specified *aggregate-file-storage-location* must exist in the file system.

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 Global 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 graphics in the location that is specified by the PATH= option.
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
mark up 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 617

**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: <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 837

**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 576

Example  
"Example 1: Creating an ODS Package" on page 580

**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 Global 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 536.

(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 536.

(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 537.

(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 537.

(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 537.

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.

Interaction

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 Global Statements: Reference.

libref.catalog
specifies a SAS catalog to write to.

See For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Global 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.

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 Global 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 536.

(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 536.

(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 537.

(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 537.

(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 537.

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 546

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*.

<table>
<thead>
<tr>
<th>Alias</th>
<th>TYPE=</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>If you do not specify a TAGSET= value, then the ODS MARKUP statement defaults to XML output.</td>
</tr>
</tbody>
</table>

| 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 538. |

<table>
<thead>
<tr>
<th>Tip</th>
<th>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:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>proc template;</td>
</tr>
<tr>
<td></td>
<td>list tagsets;</td>
</tr>
<tr>
<td></td>
<td>run;</td>
</tr>
<tr>
<td>See</td>
<td>For a list of valid tagsets and their descriptions, see “ODS Tagset Statement” on page 807.</td>
</tr>
<tr>
<td></td>
<td>For additional information about specifying tagsets, see “TEMPLATE Procedure: Creating Markup Language Tagsets” in <em>SAS Output Delivery System: Procedures Guide</em>.</td>
</tr>
</tbody>
</table>

| Examples | “Example 2: Creating an XML File and a DTD” on page 541 |
|          | “Example 3: Creating Multiple Markup Output” on page 543 |
|          | “Example 4: Specifying Tagset Names as ODS Destinations” on page 546 |

**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. |
|         | 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 336 |

**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 1109 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 543

(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 546

**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.

```sas
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;
```
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 6.45  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

```plaintext
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.

```plaintext
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.

```plaintext
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.

```plaintext
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.

```plaintext
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.
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**

```plaintext
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.

```plaintext
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.

```plaintext
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.

```plaintext
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.

```plaintext
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.

```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 6.47  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>9800</td>
</tr>
<tr>
<td>11</td>
<td>INS</td>
<td>Rice</td>
<td>1995</td>
<td>49860</td>
</tr>
<tr>
<td>12</td>
<td>INS</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>7888</td>
</tr>
<tr>
<td>15</td>
<td>USA</td>
<td>Corn</td>
<td>1995</td>
<td>187300</td>
</tr>
</tbody>
</table>

Leading Grain-Producing Countries
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 6.48 CSVALL Output Viewed in Microsoft Excel**

<table>
<thead>
<tr>
<th></th>
<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>1</td>
<td>Leading Grain-Producing Countries</td>
<td></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>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Obs</td>
<td>Country</td>
<td>Type</td>
<td>Year</td>
<td>Kilotons</td>
<td></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>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>BRZ</td>
<td>Rice</td>
<td>1995</td>
<td>11236</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>BRZ</td>
<td>Corn</td>
<td>1995</td>
<td>38276</td>
<td></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>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>CHN</td>
<td>Rice</td>
<td>1995</td>
<td>185226</td>
<td></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>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>IND</td>
<td>Wheat</td>
<td>1995</td>
<td>630007</td>
<td></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>
<td></td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>IND</td>
<td>Corn</td>
<td>1995</td>
<td>9800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>INS</td>
<td>Wheat</td>
<td>1995</td>
<td>4880</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>INS</td>
<td>Rice</td>
<td>1995</td>
<td>48807</td>
<td></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>
<td></td>
</tr>
<tr>
<td>16</td>
<td>13</td>
<td>USA</td>
<td>Wheat</td>
<td>1995</td>
<td>58494</td>
<td></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>
<td></td>
</tr>
<tr>
<td>18</td>
<td>15</td>
<td>USA</td>
<td>Corn</td>
<td>1995</td>
<td>187300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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=
STYLESHET= 
URL=suboption

**Other features:**
OPTIONS statement
PROC PRINT
TITLE statement
Example 5: Including Multiple Cascading Style Sheets in One HTML Document

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;
options 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;
options obs=15;

Create the HTML output and two style sheets. The ODS HTML statements open the HTML destination and creates HTML output. The STYLESHEET= 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;
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 6.49  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">
</head>
<body>

...}
.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 }

...>
</body>
</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.

```
.body {
  background-color: white;
}
```
color: black;
font-family: times, serif;
}
.header, .rowheader, .footer, .rowfooter, .data {
border: 1px black solid;
color: black;
padding: 5px;
font-family: times, serif;
}
.header, .rowheader, .footer, .rowfooter {
background-color: #a0a0a0;
}
.table {
background-color: #dddddd;
border-spacing: 0;
border: 1px black solid;
}
.proctitle {
font-family: helvetica, sans-serif;
font-size: x-large;
font-weight: normal;
}
@media screen {
.header, .rowheader, .footer, .rowfooter, {
color: white;
background-color: green;
}
.table {
background-color: yellow;
border-spacing: 0;
font-size: small
border: 1px black solid;
}
}
@media print {
.header, .rowheader, .footer, .rowfooter, {
color: white;
background-color: blue;
padding: 5px;
}
.data {
font-size: small;
}
}

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;
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 6.50**  HTML Output Using Both a Style Sheet with 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>V9</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></td>
<td>Compressed</td>
<td>NO</td>
</tr>
<tr>
<td>Data Set Type</td>
<td></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>WINDOWS_22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encoding</td>
<td>w-ascii ASCII (ANSI)</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\sasuser\drive\move\9200sa\devData\drive\move\sashelp\class.sas7bdat</td>
</tr>
<tr>
<td>Release Created</td>
<td>9.020.160</td>
</tr>
<tr>
<td>Host Created</td>
<td>WIN_PRO</td>
</tr>
</tbody>
</table>

**Alphabetical 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>
The white font, small font size, cell padding, and the blue background color all come from the Print media block. All other style information comes from the code outside of the media blocks. No information from the Screen media block is used.

Output 6.51  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>19</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>NO</td>
</tr>
<tr>
<td>Data Set Type</td>
<td></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>WINDOWS_32</td>
<td>Encoding</td>
<td>UTF-8: ASCII (ANSI)</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 Pages</td>
<td>1</td>
</tr>
<tr>
<td>File Name</td>
<td>C:SAS\v9.2\userlib\v920\sas_dvd\inst\v920\sashelp\class\sas7bed4.sas</td>
</tr>
<tr>
<td>Release Created</td>
<td>20040830</td>
</tr>
<tr>
<td>Host Created</td>
<td>WIN_PRO</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 6.52  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=
- PROC PRINT

Details
The following example prints to the SAS log the OPTIONS suboptions and a description of each available suboption.

Program
```sas
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 except SAS Viya
- **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 except SAS Viya
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 42. 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 891. 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 557.
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 558.

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:

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 41

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=
- ODS LISTING statement
  - CLOSE
- ODS OUTPUT statement

**Other features:**

- PROC FORMAT
- PROC PRINT
- PROC TABULATE
- KEEP= data set option

**Data set:** Energy
Details

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 1213.

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';
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 891.

ods output table=energyoutput(keep=region division type expenditures_sum);

Create HTML output.

ods html body='your_body_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;
  format region regfmt. division divfmt. type usetype.;
  title 'Energy Expenditures for Each Region';
  title2 '(millions of dollars)';
run;

Close the current body file and open a new file. The ODS HTML statement closes the original body file and opens a new one. The ODS HTML statement opens the HTML destination and creates HTML output. The output from PROC TABULATE is sent to the body file.

ods html path='..:/ods'(url=none)
  body='odsoutput-printbody.htm';

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.
Example 1: Creating a Combined Output Data Set

```
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 6.53  Combined Data Set**

<table>
<thead>
<tr>
<th>Region</th>
<th>Division</th>
<th>Type</th>
<th>Expenditures_Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Northeast</td>
<td>New England</td>
<td>Residential Customers</td>
<td>7477</td>
</tr>
<tr>
<td>Northeast</td>
<td>New England</td>
<td>Business Customers</td>
<td>5129</td>
</tr>
<tr>
<td>Northeast</td>
<td>Middle Atlantic</td>
<td>Residential Customers</td>
<td>15379</td>
</tr>
<tr>
<td>Northeast</td>
<td>Middle Atlantic</td>
<td>Business Customers</td>
<td>15078</td>
</tr>
<tr>
<td>West</td>
<td>Mountain</td>
<td>Residential Customers</td>
<td>5476</td>
</tr>
<tr>
<td>West</td>
<td>Mountain</td>
<td>Business Customers</td>
<td>4729</td>
</tr>
<tr>
<td>West</td>
<td>Pacific</td>
<td>Residential Customers</td>
<td>13959</td>
</tr>
<tr>
<td>West</td>
<td>Pacific</td>
<td>Business Customers</td>
<td>12619</td>
</tr>
</tbody>
</table>
The following output shows the two separate BY groups that are created by the TABULATE procedure.

**Output 6.54** Output Objects Created by PROC TABULATE

### Energy Expenditures for Each Region (millions of dollars)

**Region**: Northeast

<table>
<thead>
<tr>
<th>Division</th>
<th>Residential Customers</th>
<th>Business Customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>New England</td>
<td>$7,477</td>
<td>$5,129</td>
</tr>
<tr>
<td>Middle Atlantic</td>
<td>$15.379</td>
<td>$15.078</td>
</tr>
</tbody>
</table>

### Energy Expenditures for Each Region (millions of dollars)

**Region**: West

<table>
<thead>
<tr>
<th>Division</th>
<th>Residential Customers</th>
<th>Business Customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mountain</td>
<td>$5,476</td>
<td>$4,729</td>
</tr>
<tr>
<td>Pacific</td>
<td>$13.955</td>
<td>$12,519</td>
</tr>
</tbody>
</table>

**Example 2: Using Different Procedures to Create a Data Set from Similar Output Objects**

**Features**: ODS HTML statement options

- `BODY=`

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.
Program

```sas
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;
ods _all_ close;
ods html;
```

Program Description

**Specify the output objects to be sent to all 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 HTML destination is open by default, and its default list is SELECT ALL. The PERSIST option specifies that ParameterEstimates should remain in the overall selection list until the list is explicitly modified.

```sas
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.

```sas
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.

```sas
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 destination. (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 the HTML destination and any other open destinations.

    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.
Example 3: Creating Merged Data Sets

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 merging data sets with and without the MATCH_ALL option. When you use the MATCH_ALL option to merge data sets, a new data set for each output object is created.

The data sets that are printed are parameter estimates that PROC REG generates. The PROC REG procedure is part of SAS/STAT software.

Program 1: Creating a Merged Data Set Using the MATCH_ALL Option

```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 VARREMOVED 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
       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;

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 following 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.

<table>
<thead>
<tr>
<th>Obs</th>
<th>Model</th>
<th>Dependent</th>
<th>Step</th>
<th>VarEntered</th>
<th>NumberIn</th>
<th>PartialR2Square</th>
<th>ModelR2Square</th>
<th>Cp</th>
<th>FValue</th>
<th>ProbF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MODEL1</td>
<td>r33</td>
<td>1</td>
<td>d</td>
<td>1</td>
<td>0.9817</td>
<td>0.9817</td>
<td>379.485</td>
<td>68.52</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>2</td>
<td>MODEL1</td>
<td>r33</td>
<td>2</td>
<td>e</td>
<td>2</td>
<td>0.0712</td>
<td>0.9329</td>
<td>181.392</td>
<td>10.62</td>
<td>0.0086</td>
</tr>
<tr>
<td>3</td>
<td>MODEL1</td>
<td>r33</td>
<td>3</td>
<td>r24</td>
<td>3</td>
<td>0.0557</td>
<td>0.9886</td>
<td>20.903</td>
<td>44.17</td>
<td>&lt;0.001</td>
</tr>
</tbody>
</table>
The following 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.

<table>
<thead>
<tr>
<th>Obs</th>
<th>Model</th>
<th>Dependent</th>
<th>Step</th>
<th>VarRemoved</th>
<th>Parameter</th>
<th>PartialRSquare</th>
<th>ModelRSquare</th>
<th>Cp</th>
<th>FValue</th>
<th>ProbF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MODEL2</td>
<td>r33</td>
<td>1</td>
<td>r24</td>
<td>9</td>
<td>0.0000</td>
<td>0.9993</td>
<td>9.0522</td>
<td>0.05</td>
<td>0.8405</td>
</tr>
<tr>
<td>2</td>
<td>MODEL2</td>
<td>r33</td>
<td>2</td>
<td>r29</td>
<td>8</td>
<td>0.0000</td>
<td>0.9992</td>
<td>7.1193</td>
<td>0.10</td>
<td>0.7747</td>
</tr>
<tr>
<td>3</td>
<td>MODEL2</td>
<td>r33</td>
<td>3</td>
<td>d</td>
<td>7</td>
<td>0.0001</td>
<td>0.9991</td>
<td>5.4330</td>
<td>0.59</td>
<td>0.4845</td>
</tr>
</tbody>
</table>

The following 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.

<table>
<thead>
<tr>
<th>Obs</th>
<th>Model</th>
<th>Dependent</th>
<th>Step</th>
<th>VarEntered</th>
<th>Parameter</th>
<th>PartialRSquare</th>
<th>ModelRSquare</th>
<th>Cp</th>
<th>FValue</th>
<th>ProbF</th>
<th>VarRemoved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MODEL1</td>
<td>r33</td>
<td>1</td>
<td>d</td>
<td>1</td>
<td>0.8817</td>
<td>379.485</td>
<td>68.52</td>
<td>&lt;0.001</td>
<td>r24</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MODEL1</td>
<td>r33</td>
<td>2</td>
<td>e</td>
<td>2</td>
<td>0.0712</td>
<td>101.392</td>
<td>10.62</td>
<td>0.0065</td>
<td>r24</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MODEL1</td>
<td>r33</td>
<td>3</td>
<td>r24</td>
<td>3</td>
<td>0.0657</td>
<td>20.8003</td>
<td>44.17</td>
<td>&lt;0.001</td>
<td>r24</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MODEL2</td>
<td>r33</td>
<td>1</td>
<td></td>
<td>9</td>
<td>0.0000</td>
<td>9.0622</td>
<td>0.05</td>
<td>0.8495</td>
<td>r24</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MODEL2</td>
<td>r33</td>
<td>2</td>
<td></td>
<td>8</td>
<td>0.0000</td>
<td>7.1193</td>
<td>0.10</td>
<td>0.7747</td>
<td>r24</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>MODEL2</td>
<td>r33</td>
<td>3</td>
<td></td>
<td>7</td>
<td>0.0001</td>
<td>5.4330</td>
<td>0.59</td>
<td>0.4845</td>
<td>d</td>
<td></td>
</tr>
</tbody>
</table>

Program 2: Creating a Merged Data Set without Using the MATCH_ALL Option

```sas
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
    sl=.5 maxstep=3;
  model r33=a b r4 r8 c d e r23 r24 r29/ selection=backward
    sl=.05 maxstep=3;
run;
ods trace off;
ods output SelectionSummary=summary;
title1 'Without the MATCH_ALL Option, ODS Produces a Single Data Set With All Of the Columns';
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.

```sas
ods output SelectionSummary=summary;
title1 'Without the MATCH_ALL Option, ODS Produces a Single Data Set With All Of the 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
      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;
```

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.

```sas
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 6.56 Using the ODS OUTPUT Statement without the MATCH_ALL Option to Combine Data Sets**

<table>
<thead>
<tr>
<th>Obs</th>
<th>Model</th>
<th>Dependent</th>
<th>Step</th>
<th>VarEntered</th>
<th>NumberIn</th>
<th>PartialR$^2$</th>
<th>ModelR$^2$</th>
<th>Cp</th>
<th>FValue</th>
<th>ProbF</th>
<th>VarRemoved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MODEL1</td>
<td>r33</td>
<td>1</td>
<td>d</td>
<td>1</td>
<td>0.8017</td>
<td>0.8017</td>
<td>379.485</td>
<td>68.52</td>
<td>&lt;0001</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MODEL1</td>
<td>r33</td>
<td>2</td>
<td>e</td>
<td>2</td>
<td>0.0712</td>
<td>0.9329</td>
<td>181.382</td>
<td>10.62</td>
<td>0.0096</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>MODEL1</td>
<td>r33</td>
<td>3</td>
<td>r24</td>
<td>3</td>
<td>0.0557</td>
<td>0.9888</td>
<td>26.8903</td>
<td>44.17</td>
<td>&lt;0001</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>MODEL2</td>
<td>r33</td>
<td>1</td>
<td></td>
<td>9</td>
<td>0.0000</td>
<td>0.9993</td>
<td>9.0522</td>
<td>0.05</td>
<td>0.8405</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MODEL2</td>
<td>r33</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>0.0000</td>
<td>0.9992</td>
<td>7.1193</td>
<td>0.10</td>
<td>0.7747</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>MODEL2</td>
<td>r33</td>
<td>3</td>
<td>7</td>
<td>7</td>
<td>0.0001</td>
<td>0.9991</td>
<td>5.4330</td>
<td>0.59</td>
<td>0.4845</td>
<td></td>
</tr>
</tbody>
</table>

**Example 4: Creating an Output Data Set with Multiple BY Variables**

**Features:**
- ODS OUTPUT statement
- ODS TRACE statement

**Other features:**
- PROC PRINT
- PROC REG
- PROC SORT

**Details**

This example demonstrates how to create an output data set using multiple BY variables. The program uses PROC REG to perform a linear regression, and then captures the R-square value for multiple BY variables in a SAS data set.

The example is based on Wicklin (2017) but extends Wicklin’s example by using two BY variables.
Program

```sas
proc sort data=Sashelp.Cars out=Cars;
   where Type in ('SUV', 'Sedan');
   by Origin Type;
run;

ods trace on;
proc reg data=Cars plots=none;
   by Origin Type;
   model Horsepower = EngineSize Weight;
quit;
ods trace off;
ods output FitStatistics=Stats;
proc print data=Stats noobs;
   where Label2="R-Square";
   var Origin Type Label2 nValue2;
run;
```

Program Description

**Sort the data set.**

```sas
proc sort data=Sashelp.Cars out=Cars;
   where Type in ('SUV', 'Sedan');
   by Origin Type;
run;
```

**Create the PROC REG output and view the output tables that are created.** The REG procedure performs a linear regression using two BY variables, Origin and Type. The ODS TRACE statement writes a record of each output object created to the log. In the ODS TRACE output, you can see that the output object that contains the fit statistics is named FitStatistics.

```sas
ods trace on;
proc reg data=Cars plots=none;
   by Origin Type;
   model Horsepower = EngineSize Weight;
quit;
ods trace off;
```

**Create the output data set.** The ODS OUTPUT statement creates the data set named Stats that contains the fit statistics for each BY group.

```sas
ods output FitStatistics=Stats;
```

**Print the output data set.**

```sas
proc print data=Stats noobs;
   where Label2="R-Square";
```
**Partial ODS TRACE Output**

```plaintext
Output Added:
-----------------
Name:   NObs
Label:  Number of Observations
Template: Stat.Reg.NObs
Path:   Reg.ByGroup1.MODEL1.Fit.Horsepower.NObs

-----------------

Name:   ANOVA
Label:  Analysis of Variance
Template: Stat.REG.ANOVA
Path:   Reg.ByGroup1.MODEL1.Fit.Horsepower.ANOVA

-----------------

Name:   FitStatistics
Label:  Fit Statistics
Template: Stat.REG.FitStatistics
-----------------
```

**Data Set with Fit Statistics Shown for Each By Group**

<table>
<thead>
<tr>
<th>Origin</th>
<th>Type</th>
<th>Label2</th>
<th>nValue2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia</td>
<td>SUV</td>
<td>R-Square</td>
<td>0.760956</td>
</tr>
<tr>
<td>Asia</td>
<td>Sedan</td>
<td>R-Square</td>
<td>0.902094</td>
</tr>
<tr>
<td>Europe</td>
<td>SUV</td>
<td>R-Square</td>
<td>0.517596</td>
</tr>
<tr>
<td>Europe</td>
<td>Sedan</td>
<td>R-Square</td>
<td>0.754382</td>
</tr>
<tr>
<td>USA</td>
<td>SUV</td>
<td>R-Square</td>
<td>0.773578</td>
</tr>
<tr>
<td>USA</td>
<td>Sedan</td>
<td>R-Square</td>
<td>0.695844</td>
</tr>
</tbody>
</table>

**Example 5: Customizing BY Lines**

**Features:**
ODS PDF statement option

```plaintext
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 “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 "#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.

```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.

```plaintext
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.

```plaintext
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 6.57 Default Table of Contents

Step 2: Create a DATA Set

```plaintext
ods output properties=Props;
proc document name=reorder;
list / levels=all;
run;
quit;
```

```plaintext
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.

\[
\text{ods output properties=Props;}
\]

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.

\[
\text{proc document name=reorder;}
\text{list / levels=all;}
\text{run;}
\text{quit;}
\]

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.

\[
\text{ods listing;}
\text{proc print data=props;}
\text{run;}
\text{ods listing close;}
\]

Notice that some of the levels are directories (Dir) and some are tables (Table).

Output 6.58 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\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#1\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>7</td>
<td>\Tabulate#1\ByGroup2#1\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#1\Report#1</td>
<td>Dir</td>
</tr>
<tr>
<td>10</td>
<td>\Tabulate#1\ByGroup3#1\Report#1\Table#1</td>
<td>Table</td>
</tr>
</tbody>
</table>

Step 3: Customize the Table of Contents

\[
data _null_
\set props end=last;
\if type in("Table") then do;
\count+1;
\call symputx('path\i|trim(left(count)),path);
\end;
\call symputx('total',count);
\run;
\]

%macro order;
\proc document name=work.reorder;
%do i=1 %to &total;
\setlabel &path\i "#byval1";
\move &path\i to ";
%end;
ods pdf file="replayed.pdf";
   replay ;
run;
ods pdf close;

ods listing;
   list / levels=all;
run;
ods listing close;
quit;
%mend;

%order

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.

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.

%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.

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.

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 6.59  Customized BY Line in PDF Output

Output 6.60  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>*byval1</td>
<td>Before</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>*byval1</td>
<td>Before</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>*byval1</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%! \ByGroup!</td>
<td>Dir</td>
</tr>
<tr>
<td>2</td>
<td>\Tabulate%! \ByGroup! \Report!</td>
<td>Dir</td>
</tr>
<tr>
<td>3</td>
<td>\Tabulate%! \ByGroup! \Report!</td>
<td>Dir</td>
</tr>
<tr>
<td>4</td>
<td>\Tabulate%! \ByGroup! \Report!</td>
<td>Dir</td>
</tr>
<tr>
<td>5</td>
<td>\Tabulate%! \ByGroup! \Report!</td>
<td>Dir</td>
</tr>
<tr>
<td>6</td>
<td>\Tabulate%! \ByGroup! \Report!</td>
<td>Dir</td>
</tr>
<tr>
<td>7</td>
<td>\Tabulate%! \ByGroup! \Report!</td>
<td>Dir</td>
</tr>
<tr>
<td>8</td>
<td>Table</td>
<td>Table</td>
</tr>
<tr>
<td>9</td>
<td>Table</td>
<td>Table</td>
</tr>
<tr>
<td>10</td>
<td>Table</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 except SAS Viya
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:

```plaintext
ods package add file="test.sas" mimetype="text/plain" path="sas/";
```

**DATA=member-specification**

specifies the data set that you want to add to an ODS package. `member-specification` can be in the form `libname.membername` or `membername`.

**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:

```plaintext
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. `transport` 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 *SAS Publishing Framework: Developer’s Guide*.

**Example**

The following statement publishes an ODS package to the archive Test.spk:

```plaintext
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 SAS Publishing Framework: Developer's Guide.

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```
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" . . . 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;
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;
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;
```
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 6.61  Listing of Work.Archive and Details of HTM File

The SAS System

Listing of Work.Archive:

Order by: Insertion

Number of levels: All

<table>
<thead>
<tr>
<th>Obs</th>
<th>Path</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mypackage1</td>
<td>Dir</td>
</tr>
<tr>
<td>2</td>
<td>mypackage1'sashtml.htm'n1</td>
<td>File</td>
</tr>
<tr>
<td>3</td>
<td>mypackage1''gplot1.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>4</td>
<td>mypackage1''gplot2.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>5</td>
<td>mypackage1''gplot3.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>6</td>
<td>mypackage1''gplot4.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>7</td>
<td>mypackage1''gplot5.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>8</td>
<td>mypackage1''gplot6.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>9</td>
<td>mypackage1''gplot7.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>10</td>
<td>mypackage1''gplot8.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>11</td>
<td>mypackage1''gplot9.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>12</td>
<td>mypackage1''gplot10.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>13</td>
<td>mypackage1''gplot11.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>14</td>
<td>mypackage1''gplot12.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>15</td>
<td>mypackage1''gplot13.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>16</td>
<td>mypackage1''gplot14.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>17</td>
<td>mypackage1''gplot15.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>18</td>
<td>mypackage1''gplot16.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>19</td>
<td>mypackage1''gplot17.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>20</td>
<td>mypackage1''gplot18.gifn1</td>
<td>File</td>
</tr>
<tr>
<td>21</td>
<td>mypackage1''gplot19.gifn1</td>
<td>File</td>
</tr>
</tbody>
</table>

The SAS System

Listing of:\Work.Archive:myPackage1\'sashtml.htm\'n1

Order by: Insertion

Number of levels: 1

<table>
<thead>
<tr>
<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>
</table>

See Also

SAS Publishing Framework: Developer’s Guide
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 except SAS Viya
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

ODS PATH <(APPEND) | (PREPEND) | (REMOVE) > location(s);

ODS 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
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 except SAS Viya
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

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
Specify 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 332

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 793

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 42.

See “ODS SHOW Statement” on page 806

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.
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 Global 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/GRAF 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 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction</td>
<td>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.</td>
</tr>
<tr>
<td>See</td>
<td>For complete documentation about ODS and Cascading Style Sheets, see SAS Output Delivery System: Advanced Topics</td>
</tr>
<tr>
<td>Example</td>
<td>“Example 6: Applying a CSS File to ODS Output” on page 548</td>
</tr>
</tbody>
</table>

**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 DATA Step 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 740.

See For information about the FILENAME statement, see *SAS DATA Step 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 44.

---

**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.
For more information, see “Customizing Titles and Footnotes” on page 44.

**(<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.

  **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 576

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 631

“Example 6: Adding Text That Imitates a System Title” on page 634
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 638

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” in SAS Output Delivery System: Procedures
Guide.

Examples “Example 3: Customizing the Table of Contents” on page 623

“Example 6: Adding Text That Imitates a System Title” on page 634

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.
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 38

Statements

- “ODS PDF Statement” on page 597
- “ODS PRINTER Statement” on page 722
- “ODS PS Statement” on page 749
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.
Note: 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.
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.
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
ACCESSIBLE_IDENTIFIER <=ON | OFF | YES | NO> |
NOACCESSIBLE_IDENTIFIER
Add an identifier to the metadata of the PDF file confirming that the PDF produced by SAS meets the PDF Matterhorn Protocol
ACCESSIBLE <=ON | OFF | YES | NO> | NOACCESSIBLE
Add non-visual metadata to the PDF file that enables the file to be accessed by assistive technology such as a screen reader
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=
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
Specify 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 332

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 793

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 42.
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.

ACCESSIBLE <=ON | OFF | YES | NO> | NOACCESSIBLE
adds non-visual metadata to the PDF file that enables the file to be accessed by assistive technology such as a screen reader. When metadata is added, the file is often called a tagged PDF and follows the PDF/Universal Accessibility (PDF/UA) format.

Turning on the accessible feature after non-accessible content has been produced will not generate an accessible document.

Default NOACCESSIBLE

Interaction When the ACCESSIBLE= option is enabled, the PDFTOC= option is set to 0. The table of contents is fully expanded.

Tip You can also use the ACCESSIBLEPDF system option to enable the ACCESSIBLE option in the ODS PDF destination by default.
ACCESSIBLE_IDENTIFIER <=ON | OFF | YES | NO> | NOACCESSIBLE_IDENTIFIER

adds an identifier to the metadata of the PDF file confirming that the PDF produced by SAS meets the PDF Matterhorn Protocol. If the identifier is included, then a note is written to the SAS log. Use the ODS PDF ACCESSIBLE option or the ACCESSIBLEPDF system option to create accessible PDF output. For information about the ACCESSIBLEPDF system option, see “ACCESSIBLEPDF System Option” in SAS System Options: Reference.

Accessibility note: Using the ACCESSIBLE_IDENTIFIER option does not produce an accessible document. You must manually check your PDF for accessibility. The ACCESSIBLE_IDENTIFIER option is a method for you to add confirmation of accessibility to your document.

Note: This feature applies to SAS 9.4M6 and to later releases.

Default NOACCESSIBLE_IDENTIFIER, ACCESSIBLE_IDENTIFIER=NO

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: 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.

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

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
spefies 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 621

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 621

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.

Aliases  OFF

YES
specifies to generate bookmarks in PDF and PS files.

Aliases  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](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 621

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.

RequirementYou 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 Global Statements: Reference.

"URL"
is a URL to an external file.

RequirementYou 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.

DefaultIf 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.

RangeYou can specify up to ten different media types.

RequirementsYou must enclose media-type in parentheses.

You must specify media-type next to the file-specification specified by the CSSSTYLE= option.

TipIf 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.

RestrictionThe CSSSTYLE= option does not affect SAS/GRAPH output.

RequirementCSS 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

InteractionIf both the STYLE= option and the CSSSTYLE= option are specified in an ODS statement, the option specified last is the option that is used.
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 DATA Step 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 740.

See

For information about the FILENAME statement, see SAS DATA Step 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 44.

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 44.

(<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 631

“Example 6: Adding Text That Imitates a System Title” on page 634

“Example 7: Toggling Page Breaks” on page 638

“Example 8: Suppressing a Page Break” on page 642

---

**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 576

---

**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.
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

Interaction When the ACCESSIBLE= option is enabled, the PDFTOC= option is set to 0. The table of contents is fully expanded.

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 617

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 631

“Example 6: Adding Text That Imitates a System Title” on page 634

“Example 7: Toggling Page Breaks” on page 638
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 638

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 \( \Rightarrow \) DESTINATIONS \( \Rightarrow \) 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 \( \Rightarrow \) 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” in SAS Output Delivery System: Procedures Guide.

Examples “Example 3: Customizing the Table of Contents” on page 623

“Example 6: Adding Text That Imitates a System Title” on page 634

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 336

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

Overview of the PDF Statement
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.

Starting with SAS 9.4M6, the method used to build and compress PDF files has been
enhanced, resulting in smaller file sizes. This change impacts all PDF files including
tagged PDF files.

**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 6.20  PDF System Options and Associated PDF Document Properties

<table>
<thead>
<tr>
<th>Action</th>
<th>System Option</th>
<th>Document Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifies whether PDF documents can be edited</td>
<td>PDFACCESS</td>
<td>Editing the Document</td>
</tr>
<tr>
<td>Specifies whether PDF documents can be assembled</td>
<td>PDFASSEMBLY</td>
<td>Document Assembly</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>
</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>Table 6.21</th>
<th>PDFSECURITY= Option and PDFPASSWORD= Option Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDFPASSWORD=</td>
<td>PDFSECURITY=HIGH</td>
</tr>
<tr>
<td>Security Method</td>
<td>Password Security</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>Table 6.22</th>
<th>Default PDF Document Properties for PDFSECURITY= Option Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PDFSECURITY=NONE</strong></td>
<td><strong>PDFSECURITY=HIGH</strong></td>
</tr>
<tr>
<td>Printing</td>
<td>Allowed</td>
</tr>
<tr>
<td>Changing the Document</td>
<td>Allowed</td>
</tr>
<tr>
<td>Commenting</td>
<td>Allowed</td>
</tr>
<tr>
<td>Form Field Fill-in or Signing</td>
<td>Allowed</td>
</tr>
<tr>
<td>Document Assembly</td>
<td>Allowed</td>
</tr>
<tr>
<td>Content Copying</td>
<td>Allowed</td>
</tr>
<tr>
<td>Content Accessibility Enabled</td>
<td>Allowed</td>
</tr>
<tr>
<td>Page Extraction</td>
<td>Allowed</td>
</tr>
<tr>
<td>Encryption Level</td>
<td>None</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 6.62 PDF Output Default Bookmark Tree

- A printable table of contents is generated using the CONTENTS=YES option on the ODS PDF FILE= statement. When viewing the PDF file on a computer, the table of contents is clickable. The page count of the PDF file is not affected by the table of contents. 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.

Output 6.63 PDF Output Default Table of Contents Page

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
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

```plaintext
proc sort data=grain_production;
    by year country type;
run;
ods HTML close;
options 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;
title2;
proc tabulate data=grain_production format=comma12.;
class year country type;
var kilotons;
table year,
country*type,
```
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.

```sql
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.

```sql
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.

```sql
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.

```sql
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.

```sql
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.

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.

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 or print the output.

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.

ods html;
The default style for the ODS PDF, ODS PS, ODS PCL, and ODS PRINTER statements is Pearl.

Output 6.64 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>36,275</td>
</tr>
<tr>
<td></td>
<td>Rice</td>
<td>11,236</td>
</tr>
<tr>
<td></td>
<td>Wheat</td>
<td>1,515</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**

```sas
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;
```
PDF Output

Output 6.66  Printable Table of Contents for PDF Output

<table>
<thead>
<tr>
<th>Table of Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Freq Procedure</td>
</tr>
<tr>
<td>Table Origin *Tyre</td>
</tr>
<tr>
<td>Cross-Tabular Proc Table</td>
</tr>
<tr>
<td>The Print Procedure</td>
</tr>
<tr>
<td>Data Set SASHelp.CARS</td>
</tr>
</tbody>
</table>

**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 965.
- For a table of style attributes, see “Style Attributes Tables” on page 1074.
- For a table of style elements affecting pages, see “Style Elements Affecting Pages” on page 1047.
- For a table of style elements affecting the table of contents, see “Style Elements Affecting Tables of Contents” on page 1046.
Program

```
ods html close;
proc template;
  define style Styles.CustomTitle;
  parent=Styles.pearl;
    class ContentTitle from ContentTitle /
      pretext='My Customized Title';
  end;
run;

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.
Specify a title and set the SAS system options.

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.

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.

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.

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.

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.

proc template;
delete Styles.CustomTitle;
run;
PDF Output

Output 6.67  Customized Bookmark List for PDF Output

Output 6.68  Default Table of Contents

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 “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 6.69 Default Table of Contents**

![Default Table of Contents](image)

**Step 2: Create a DATA Set**

```ods output properties=Props;
ods listing;
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.

```latex
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 6.70  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>Tabulate1</td>
<td>Dir</td>
</tr>
<tr>
<td>2</td>
<td>Tabulate1\ByGroup1</td>
<td>Dir</td>
</tr>
<tr>
<td>3</td>
<td>Tabulate1\ByGroup1\Report1</td>
<td>Dir</td>
</tr>
<tr>
<td>4</td>
<td>Tabulate1\ByGroup1\Report1\Table1</td>
<td>Table</td>
</tr>
<tr>
<td>5</td>
<td>Tabulate1\ByGroup2</td>
<td>Dir</td>
</tr>
<tr>
<td>6</td>
<td>Tabulate1\ByGroup2\Report1</td>
<td>Dir</td>
</tr>
<tr>
<td>7</td>
<td>Tabulate1\ByGroup2\Report1\Table1</td>
<td>Table</td>
</tr>
<tr>
<td>8</td>
<td>Tabulate1\ByGroup3</td>
<td>Dir</td>
</tr>
<tr>
<td>9</td>
<td>Tabulate1\ByGroup3\Report1</td>
<td>Dir</td>
</tr>
<tr>
<td>10</td>
<td>Tabulate1\ByGroup3\Report1\Table1</td>
<td>Table</td>
</tr>
</tbody>
</table>

Step 3: Customize the Table of Contents

```latex
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;
```

```latex
%macro order;  
proc document name=work.reorder;  
%do i=1 %to &total;  
    setlabel &patht&i "#byval1";  
    move &patht&i to ^;  
%end;  
ods pdf file="replayed.pdf";  
replay ;  
run;  
ods pdf close;
```

```latex
ods listing;  
list / levels=all;  
run;  
ods listing close;  
quit;  
%mend;
```
%order

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.

```macro
%order
```

**Output 6.71** Customized BY Line in PDF Output

![Image showing a table with columns Asia, Europe, and USA]

**Output 6.72** 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>*byval1</td>
<td>Before</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>*byval1</td>
<td>Before</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>*byval1</td>
<td>Before</td>
<td></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
ods pdf file="file.pdf" notoc startpage=no nogtitle;
ods noproctitle;
```
Create the MEANS procedure output.

```sas
proc means data=sashelp.cars;
   class cylinders;
   var mpg_city mpg_highway;
run;
```

Create a scatter plot with ODS graphics.

```sas
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.

```sas
ods pdf close;
ods html;
```
Example 6: Adding Text That Imitates a System Title

**Features:**

ODS PDF statement option

FILE=
NOTOC
STARTPAGE=NO
STYLE=

---

### Eliminating Page Breaks

<table>
<thead>
<tr>
<th>Cylinders</th>
<th>N Obs</th>
<th>Variable</th>
<th>Label</th>
<th>N</th>
<th>Mean</th>
<th>Std Dev</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>MPG_City</td>
<td>MPG (City)</td>
<td>1</td>
<td>66.000000000</td>
<td>.</td>
<td>66.000000000</td>
<td>66.000000000</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>MPG_Highway</td>
<td>MPG (Highway)</td>
<td>1</td>
<td>66.000000000</td>
<td>.</td>
<td>66.000000000</td>
<td>66.000000000</td>
</tr>
<tr>
<td>4</td>
<td>136</td>
<td>MPG_City</td>
<td>MPG (City)</td>
<td>136</td>
<td>24.5411765</td>
<td>5.2039430</td>
<td>16.000000000</td>
<td>59.000000000</td>
</tr>
<tr>
<td></td>
<td>136</td>
<td>MPG_Highway</td>
<td>MPG (Highway)</td>
<td>136</td>
<td>31.6897059</td>
<td>4.6544333</td>
<td>23.000000000</td>
<td>51.000000000</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>MPG_City</td>
<td>MPG (City)</td>
<td>7</td>
<td>19.8571429</td>
<td>0.8997354</td>
<td>18.000000000</td>
<td>21.000000000</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>MPG_Highway</td>
<td>MPG (Highway)</td>
<td>7</td>
<td>26.8571429</td>
<td>1.0690450</td>
<td>25.000000000</td>
<td>28.000000000</td>
</tr>
<tr>
<td>6</td>
<td>190</td>
<td>MPG_City</td>
<td>MPG (City)</td>
<td>190</td>
<td>18.5157895</td>
<td>1.7630130</td>
<td>14.000000000</td>
<td>23.000000000</td>
</tr>
<tr>
<td></td>
<td>190</td>
<td>MPG_Highway</td>
<td>MPG (Highway)</td>
<td>190</td>
<td>25.5528318</td>
<td>3.1697760</td>
<td>17.000000000</td>
<td>32.000000000</td>
</tr>
<tr>
<td>8</td>
<td>87</td>
<td>MPG_City</td>
<td>MPG (City)</td>
<td>87</td>
<td>15.8725632</td>
<td>1.8912565</td>
<td>10.000000000</td>
<td>18.000000000</td>
</tr>
<tr>
<td></td>
<td>87</td>
<td>MPG_Highway</td>
<td>MPG (Highway)</td>
<td>87</td>
<td>21.8852675</td>
<td>3.5777909</td>
<td>12.000000000</td>
<td>28.000000000</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>MPG_City</td>
<td>MPG (City)</td>
<td>2</td>
<td>11.000000000</td>
<td>1.4142136</td>
<td>10.000000000</td>
<td>12.000000000</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>MPG_Highway</td>
<td>MPG (Highway)</td>
<td>2</td>
<td>16.000000000</td>
<td>4.9497475</td>
<td>13.000000000</td>
<td>20.000000000</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>MPG_City</td>
<td>MPG (City)</td>
<td>3</td>
<td>12.6666667</td>
<td>0.5773500</td>
<td>12.000000000</td>
<td>13.000000000</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>MPG_Highway</td>
<td>MPG (Highway)</td>
<td>3</td>
<td>19.000000000</td>
<td>0</td>
<td>19.000000000</td>
<td>19.000000000</td>
</tr>
</tbody>
</table>
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 965.
- For a table of style attributes, see “Style Attributes Tables” on page 1074.
- For a table of style elements affecting pages, see “Style Elements Affecting Pages” on page 1047.
- For a table of style elements affecting the table of contents, see “Style Elements Affecting Tables of Contents” on page 1046.

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;
```

Example 6: Adding Text That Imitates a System Title 635
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.

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;
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;

After your output is produced, you can remove the custom style. The DELETE statement in PROC TEMPLATE removes the custom style.

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 6.74  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 and SGPLOT image are on one page, and the second REPORT table and SGPLOT image are placed each on the next page. Each page has a different title and footnote.
Program

```sas
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;
ods 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.
ods html close;

Set the SAS system options.

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.

```
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.

```
title "Top of the First Page Title";
footnote "Bottom of the first page footnote";
```

Create the REPORT procedure output for the first page.

```
proc report data=sashelp.cars ;
  col origin enginestize 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.

```
title;
footnote;
```

Create the SGPLOT procedure output for the first page.

```
ods graphics on / reset noborder;
proc sgplot data=sashelp.cars;
  where mpg_highway gt 45;
  scatter x=enginestize 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.

```
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;
```
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 6.75  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

Other features:
- 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;

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.**
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.

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.

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.

ods pdf close;
ods html;
PDF Output

Output 6.76  Preventing a Page Break in PDF Output

Example 9: How to Use Drill-Down and Link to a Target in a PDF File

Features: ODS PDF statement option:
Details

Starting in SAS 9.4m5, ODS PDF supports linking (drilling down) to targets or destinations in PDF documents. For an explanation how to link to graphs generated using SAS/GRAPH, see “Adding Drill-Down Graphs in Your PDF File” in SAS/GRAPH: Reference.

Program

```sas
%let outpath=.;
data cars;
  length url $60;
  set sashelp.cars;
  url="/cars.pdf#" || strip(origin);
run;
proc sort data=cars;
  by origin;
run;
ods _all_ close;
goptions reset=all border vsize=350pt hsize=420pt;
ods pdf style=daisy anchor="#BYVAL" file="/cars.pdf";
proc gchart data=cars;
  pie type / sumvar=mpg_highway type=mean;
  by origin;
run;
quit;
ods _all_ close;
title "Mean Highway MPG and Vehicle Origin";
footnote "Click a bar for details";
ods pdf style=daisy file="/sales.pdf";
proc gchart data=cars;
  vbar origin / sumvar=mpg_highway type=mean url=url;
run;
quit;
ods pdf close;
goptions reset=all;
```

Program Description

Specify the data set.

```sas
%let outpath=.;
data cars;
```
length url $60;
set sashelp.cars;
url="/cars.pdf## || strip(origin);
run;

Sort the data.

proc sort data=cars;
  by origin;
run;

ods _all_ close;

Generate a PDF with target graphs.

goptions reset=all border vsize=350pt hsize=420pt;
ods pdf style=daisy anchor="#BYVAL" file="\cars.pdf";
proc gchart data=cars;
  pie type / sumvar=mpg_highway type=mean;
  by origin;
run;
quit;
ods _all_ close;

Define the target URLs for the drill-down graph The URL= option in the VBAR statement specifies the column in the plot data that provides the drill-down URLs for the bars. The data provides a unique URL for each origin, which, in this case, links to a named target in the target PDF file. In the resulting bar chart, each bar is a hotspot that is linked to its assigned URL.

title "Mean Highway MPG and Vehicle Origin";
footnote "Click a bar for details";
ods pdf style=daisy file="\sales.pdf";
proc gchart data=cars;
  vbar origin / sumvar=mpg_highway type=mean url=url;
run;
quit;
ods pdf close;
goptions reset=all;
PDF Drill-Down Output

The bar graph values shown in the image below contain URLs that link to more detailed pie graphs about the respective values.
The Asia value in the bar graph links to the Asia pie graph shown below.

See Also

Other ODS Features

- For complete documentation about styles, see “Understanding Styles, Style Elements, and Style Attributes” on page 965.
- For a table of style attributes, see “Style Attributes Tables” on page 1074.

Statements

- “ODS PCL Statement” on page 587
- “ODS PRINTER Statement” on page 722
- “ODS PS Statement” on page 749

ODS POWERPOINT Statement

Opens, manages, or closes the ODS destination for PowerPoint, which produces PowerPoint output.

Valid in: Anywhere except SAS Viya
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 works only 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 667
“Example 2: ODS PowerPoint Using Layouts, Styles, and Background Gradients” on page 669
“Example 3: Customizing the Background and Transition between the PowerPoint Slides” on page 677
“Example 4: PowerPoint Slides with Background Image Specified by Path” on page 679
“Example 5: PowerPoint Slides with Tile Background Image Specified by FILEREF” on page 681
“Example 7: PowerPoint Slides with Radial Gradient Filled Background” on page 684
“Example 8: Creating Transition Effects for PowerPoint Slides” on page 686

Syntax

```sas
ODS POWERPOINT (<ID=> identifier> <action>);
ODS POWERPOINT (<ID=> identifier> <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 POWERPOINT 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
STARTPAGE= YES | NO | NOW
Controls insertion of new slides
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 332

**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 793

**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 42.
See | “ODS SHOW Statement” on page 806

**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 44.

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.

Restriction The LAYOUT= option represses this option.

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

Restrictions 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.

The LAYOUT= option represses this option.

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

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

ods powerpoint file="test_output.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

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  ods powerpoint file="test_proc.pptx"
    layout=twocontent(advance=table);

Restriction  This option represses the GTITLE and GFOOTNOTE ODS
              PowerPoint options.

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= |
           BACKGROUNDCOLOR= | BACKGROUNDIMAGE= |
           BACKGROUNDREPEAT= | 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  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');

BACKGROUNDCOLOR=color
specifies the color of the background of the slides.

Tip  When present, the value of this option overrides the
      BACKGROUNDCOLOR attribute of the BODY style element in the style
for all the slides created after the option was used.

See  See “color” on page 1132.

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 677
BACKGROUNDIMAGE='file-specification' | 'LINEAR-GRADIENT' | 'RADIAL-GRADIENT' | 'NONE'
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 686

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.

Here 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.

Notes: The gradient must include at least two color-stops.

Example: `linear-gradient (top, black, white);`

\[ angle, \text{color-stop}[, \text{color-stop}]^{+} \]

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 666.

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.

Notes: The side and corner names do not require quotation marks.

Example: `linear-gradient (45deg, #98f5ff, #7eb6ff, #3a5fcd, #4d71a3, #3a66a7);`

```
angle, 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 #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.
Example  linear-gradient(180deg, purple 30%, yellow 75% );

Examples

“Example 6: PowerPoint Slides with Linear Gradient Filled Background ” on page 682

“Example 4: PowerPoint Slides with Background Image Specified by Path ” on page 679

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 686

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**

```plaintext
red
  blue 25%
  #dde6cf
  #d49e6c 30%
  rgb(23,55,94)
  rgb(85%,85%,85%) 33%
```

**NONE**

turns off the background image and returns to the default.

**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.

**Notes**

BACKGROUNDIMAGE= takes precedence over BACKGROUNDCOLOR.

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.
BACKGROUNDREPEAT=\texttt{option} specifies whether the image used for the slide background is repeated horizontally, vertically, both, or not repeated. \texttt{option} can be one of the following:

- \texttt{NO_REPEAT} specifies that the image is not repeated.
- \texttt{REPEAT} specifies that the image is repeated both horizontally and vertically.

Example ods powerpoint options(backgroundimage="backgrnd" backgroundrepeat="repeat");

DURATION= \texttt{\textquotesingle seconds\textquotesingle} 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.

Example ods powerpoint options(duration="1.5");

EFFECT_OPTION= \texttt{\textquotesingle options\textquotesingle} 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 1215 for a list of the options supported for the EFFECT_OPTION.

Default See Table A5.1 on page 1215.

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" 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 677

"Example 8: Creating Transition Effects for PowerPoint Slides" on page 686

SOUND= \texttt{\textquotesingle filename\textquotesingle} 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  
```
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 1215 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 1215 for a list of the transition names.

Example

```ods powerpoint file=\"test.pptx\"
options\{bgcolor="#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 677

\"Example 8: Creating Transition Effects for PowerPoint Slides\" on page 686

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.

**STARTPAGE= YES | NO | NOW**

controls the insertion of new slides. The first output in a layout template always starts on a new slide. When STARTPAGE=YES, each output in a layout triggers the insertion of a new slide. When STARTPAGE=NO, any following output goes on the same slide if there is room. When STARTPAGE=NOW, forced insertion of a new slide occurs.

**YES**

inserts a new slide at the start of each table or graph.

**Alias**

ON

**Restriction**

STARTPAGE=YES can be used to trigger a new slide between containers generated by ODS LAYOUT GRIDDED statements. It must be used between the end of one container and the start of another.

**NO**

instructs ODS not to insert any new slides at the start of each procedure or within certain procedures, even if the procedure code requests new slides. A new slide begins only when a slide is filled or when you specify STARTPAGE= NOW.

**Alias**

NEVER

**Restrictions**

The STARTPAGE= NO | NOW options cannot be used with predefined layouts.

If you use the LAYOUT= option and the STARTPAGE= NO | NOW options on the same ODS POWERPOINT statement, you get a warning message and the STARTPAGE option is ignored.

**Tip**

If you specify a predefined layout with the LAYOUT= option and STARTPAGE=NO is in effect, STARTPAGE=NO is suspended and STARTPAGE=YES is effective until the layout is terminated by using a LAYOUT=_NULL_ option. At that point, =NO is restored.

**NOW**

forces the immediate insertion of a new slide.

**Tip**

This option is useful primarily when the current value of the STARTPAGE= option is NO. Otherwise, each new procedure forces a new slide automatically.

**Default**

YES

**Tip**

Specifying STARTPAGE= NO prevents forced slide insertion. You can turn on forced slide insertion again by specifying STARTPAGE=YES. You can insert a new slide at any time by specifying STARTPAGE=NOW.

**Example**

“Example 9: Using STARTPAGE=NOW” on page 691
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.

\[\text{style-element-name} \mid [\text{style-attribute-name-1}=\text{style-attribute-value-1} \langle \text{style-attribute-name-2}=\text{style-attribute-value-2} \ldots \rangle]\]

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 965.

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.

**Supported File Types**

The following table provides descriptions of the supported file types for the ODS destination for PowerPoint.

*Table 6.23 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>GIF (Graphics Interchange Format)</td>
<td>Supports only color-mapped images. GIF is owned by CompuServe, Inc.</td>
</tr>
<tr>
<td>JPEG (Joint Photographic Experts Group)</td>
<td>A file format that is used for storing noninteractive images.</td>
</tr>
<tr>
<td>PNG (Portable Network Graphic)</td>
<td>Supports true color, gray-scale, and 8-bit images.</td>
</tr>
<tr>
<td>TIFF (Tagged Image File Format)</td>
<td>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).</td>
</tr>
</tbody>
</table>

**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. Here are the predefined gridded layouts.

- **TitleSlide**: 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 ODS TEXT. All TITLE statements are ignored. Footnotes specified after the first footnote are ignored.

- **TitleAndContent**: 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=_NULL_.

TwoContent
The TwoContent layout creates two-column output. This layout is similar to specifying the COLUMNS=2 option when you are using ODS LAYOUT. It places 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 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 6.14 on page 365.

**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 666.

Figure 6.1  Gradient Line Specified by an Angle

To see an example program, see “Example 6: PowerPoint Slides with Linear Gradient Filled Background” on page 682.

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 684.
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 \texttt{BACKGROUNDIMAGE=} on page 656.

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.

- \texttt{CSS Image Values and Replaced Content Module Level 3}
- \texttt{CSS3 Values and Units}
- \texttt{CSS Color Module Level 3}

\section*{Examples}

\textbf{Example 1: Creating Content for PowerPoint Slides}

\begin{description}
\item[Features:] ODS POWERPOINT statement
\item[CLOSE] PROC ODSLIST statement
\item[ITEM statement]
\item[P statement]
\item[STYLE= option]
\item[Other features:] FOOTNOTE statement
\item[ODS HTML CLOSE statement]
\item[TITLE statement]
\item[goptions]
\item[GMAP procedure]
\end{description}

\section*{Details}

The following example produces PowerPoint output.

\section*{Program}

\begin{verbatim}
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';
\end{verbatim}
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 and specify footnotes.**

```sas
title1 'PowerPoint Using Template Layout Twocontent with ODSLIST/GMAP';
footnote "The ODS Output Destination for PowerPoint";
```

**Create a PowerPoint file and two-column layout.**

```sas
ods powerpoint file="Layout2List.ppt" layout=twocontent
nogtitle nogfootnote;
```

**Create a list for the left column of the slide.**

```sas
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.**

```sas
options 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;
```
run;
quit;
ods _all_ close;

Output
A PowerPoint slide is produced using PROC ODSLIST and PROC GMAP using a two-column layout.

Output 6.77  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
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";
Example 2: ODS PowerPoint Using Layouts, Styles, and Background Gradients

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;
   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'
    ctext="#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.
Set the escape character. The escape character is used to add style options.

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.

ods powerpoint file="powerptOptions.ppt" style=styles.example;

Create the title slide. The LAYOUT= output specifies TITLESLIDE as the layout.

ods powerpoint layout=titleslide;

Create content for the title slide. The ODSTEXT procedure creates output for the title slide.

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.

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.

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=TWOCONTENT 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.

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;
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.

Output 6.78  PowerPoint Slide Using the TITLESLIDE Layout and Background Gradients

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

The ODS Destination for PowerPoint
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";
ods powerpoint file="test.pptx"
```
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(backgroundcolor="#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

**Other features:**
- TABULATE Procedure
- TITLE statement

**Details**

The following example produces PowerPoint output where the background of the worksheet has been customized using the `BACKGROUNDIMAGE=` suboption.

To access the image file used in this example, see User's Guide Examples ZIP file.

**Program**

```plaintext
dohtml 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 6.83**  *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.

To access the image file used in this example, see User's Guide Examples ZIP file.

Program
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.

ods html close;

Specify titles.

  title;
  title2 'Tiled Image Background';
  title3 'Specified By a Fileref';

Create a PowerPoint slide with a tiled background specified by using a fileref.

  filename backgrnd "tilepattern.gif";
  ods powerpoint options(backgroundimage="backgrnd"
    backgroundrepeat="repeat");

Print the data with the customized background image.

  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 6.84** 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)';
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.
**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**

```sas
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;
```

**Output 6.85**  
PowerPoint Slide with OPTIONS BACKGROUNDIMAGE= LINEAR_GRADIENT
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.

```plaintext
ods html close;
```

**Specify titles.**

```plaintext
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.

```plaintext
ods powerpoint options(backgroundimage=
    "radial-gradient(bottom left circle farthest-corner,
    #FFFF00, #00FFFF 33%, #FFFFFF )");
```

**Print the data with the customized background.**

```plaintext
proc print data=sashelp.class(obs=5);
run;
```

**Close the PowerPoint destination.**

```plaintext
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
```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
```

Chapter 6 • Dictionary of ODS Language Statements
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.

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
;
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: \( \log CP \)

Smoothing Model Component(s): \( \text{spline(Age)} \) \( \text{spline(BaseDeficit)} \)

<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>
Patterns of Diabetes

The GAM Procedure
Dependent Variable: logCP
Smoothing Model Component(s): spline(Age) spline(BaseDeficit)

Smoothing Model Analysis
Analysis of Deviance

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Sum of Squares</th>
<th>Chi-Square</th>
<th>Pr &gt; ChiSq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spline(Age)</td>
<td>3</td>
<td>0.150761</td>
<td>12.2605</td>
<td>0.0065</td>
</tr>
<tr>
<td>Spline(BaseDeficit)</td>
<td>3</td>
<td>0.081273</td>
<td>6.6095</td>
<td>0.0854</td>
</tr>
</tbody>
</table>

Output 6.88  Transition from Slide 2 to Slide 3

Output 6.89  Customized PowerPoint Slide 3
Example 9: Using STARTPAGE=NOW

Features:
ODS POWERPOINT statement
STARTPAGE= option

Other features:
ODS LAYOUT GRIDDED statement
PROC ODSTEXT
PROC SQL

Details
The following example shows you how to use the STARTPAGE=NOW option to insert a new slide at the start of the next procedure.

Program
ods _all_ close;
ods powerpoint file='test.ppt'
options (backgroundcolor='#93ACFF');

proc sql;
create table cars as
select Make,
    sum(MSRP) as TotMSRP format=dollar10.
from sashelp.cars
    group by Make;
quit;

title;
proc odstext;
p **;
p **;
p **;
p **;
p **;
p **;
p **;
p **;
p **;
p **;
p "Daily Report" / style={fontweight=bold fontsize=36pt};
p "Thursday, August 25, 2017" / style={fontweight=bold fontsize=20pt};
run;
ods powerpoint startpage=now;

ods layout gridded;
ods graphics / width=8in height=3in;
ods region;
proc sgplot data=cars;
vbar Make / response=TotMSRP
    nostatlabel name="bar" nooutline;
yaxis min=0 offsetmin=0 values=(0 to 1650000 by 150000) grid
    label="Total MSRP";
xaxis display=(noticks nolabel);
run;
ods layout end;

ods layout gridded columns=2;
ods region;
proc odslist;
  item;
    p "Car Makes" / style={fontweight=bold fontsize=10pt};
    list;
      item "Acura" / style={fontsize=10pt};
      item "Audi" / style={fontsize=10pt};
      item "BMW" / style={fontsize=10pt};
      item "Buick" / style={fontsize=10pt};
      item "ETC" / style={fontsize=10pt};
    end;
  end;
run;

ods region;
proc odslist;
  item;
    p "Car Makes" / style={fontweight=bold fontsize=10pt};
    list;
      item "Acura" / style={fontsize=10pt};
      item "Audi" / style={fontsize=10pt};
      item "BMW" / style={fontsize=10pt};
      item "Buick" / style={fontsize=10pt};
      item "ETC" / style={fontsize=10pt};
    end;
  end;
run;
ods layout end;
ods powerpoint close;

Program Description

Close all open destinations.
ods _all_ close;

Create the PowerPoint output.
ods powerpoint file='test.ppt'
  options (backgroundcolor='#93ACFF');

Use PROC SQL to format your table data.
proc sql;
  create table cars as
    select Make,
    sum(MSRP) as TotMSRP format=dollar10.
    from sashelp.cars
    group by Make;
quit;
Example 9: Using STARTPAGE=NOW

Output your text and the style to use for your text.

```plaintext
proc odstext;
   p "";
   p "";
   p "";
   p "";
   p "";
   p "";
   p "";
   p "Daily Report" / style={fontweight=bold fontsize=36pt};
   p "Thursday, August 25, 2017" / style={fontweight=bold fontsize=20pt};
run;
```

Start a new slide.

```plaintext
ods powerpoint startpage=now;
```

Place a graph and two lists on the new slide.

```plaintext
ods layout gridded;
   ods graphics / width=8in height=3in;
   ods region;
   proc sgplot data=cars;
      vbar Make / response=TotMSRP
         nostatlabel name="bar" nooutline;
      yaxis min=0 offsetmin=0 values=(0 to 1650000 by 150000) grid
         label="Total MSRP";
      xaxis display=(noticks nolabel);
   run;
   ods layout end;

ods layout gridded columns=2;
   ods region;
   proc odslist;
      item;
      p "Car Makes" / style={fontweight=bold fontsize=10pt};
      list;
      item "Acura" / style={fontsize=10pt};
      item "Audi" / style={fontsize=10pt};
      item "BMW" / style={fontsize=10pt};
      item "Buick" / style={fontsize=10pt};
      item "ETC" / style={fontsize=10pt};
   end;
   end;
   run;

ods region;
proc odslist;
   item;
   p "Car Makes" / style={fontweight=bold fontsize=10pt};
```
list;
    item "Acura" / style={fontsize=10pt};
    item "Audi" / style={fontsize=10pt};
    item "BMW" / style={fontsize=10pt};
    item "Buick" / style={fontsize=10pt};
    item "ETC" / style={fontsize=10pt};
    end;
  end;
run;
ods layout end;

Close the ODS destination for PowerPoint.
ods powerpoint close;

PowerPoint Output

Output 6.90 STARTPAGE=NOW title page

Daily Report
Thursday, August 25, 2017

Output 6.91 STARTPAGE=NOW Moves the Graph to the Second Page

ODS PREFERENCES Statement
Reverts the ODS settings back to start-up defaults.

Valid in: Anywhere except SAS Viya
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.

Valid in: Anywhere except SAS Viya
Category: ODS: Third-Party Formatted
Default: The default style for Markup family destinations is HTMLBlue.
Note: Class attributes are used only for the justification.

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
ODS PHTML Statement

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 332

**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 793

**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 42.

See: “ODS SHOW Statement” on page 806

**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. ODS assumes that if the anchor name ends in a number, then the number should be treated as a starting point for incrementing the 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.

Similarly, if you specify ANCHOR=’TABULATE1’, then ODS names the first anchor tabulate1. The second anchor is named tabulate2.

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.

**Interaction**
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.

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:

```sas
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 720.

- `(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 720.

- `(NO_TOP_MATTER)`
  **See**
  For complete documentation about the NO_TOP_MATTER suboption, see
  “(NO_TOP_MATTER)” on page 721.

- `(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 721.
For complete documentation about the URL= suboption, see “(URL= 'Uniform-Resource-Locator’ )” on page 722.

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

When you use the ODS HTML statement with the FILE= option to specify a fully-qualified path with a directory and a file name, the best practice for generating HTML files is to separate the path from the file. To accomplish this, use a combination of the PATH= option, which specifies the directory, and the FILE= option, which names the actual file. For example:

```ods html path="c:\temp" file="myfile.html";
proc print data=sashelp.class;
run;
ods html close;```

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 44.

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.

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 720.

(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 722.

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 Global 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 720.

(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 720.

(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 721.

(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 721.

(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 722.

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 Global 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 548

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 Global 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 720.

**(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 720.

**(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 721.

**(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 721.

**(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 722.

**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 541

**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.

Requirements
You must enclose aggregate-file-storage-location in quotation marks.

The specified aggregate-file-storage-location must exist in the file system.

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 Global 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=
onection, 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 617

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:

```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/GRAFPH this means that ODS creates a new file for each SAS/GRAFPH 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.
suboption(s)
specifies one or more suboptions that are valid for the specified tagset. Suboptions have the following format.

\[ \text{keyword} = \text{value} \]

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 837

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 576

Example "Example 1: Creating an ODS Package" on page 580

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 Global Statements: Reference.

element.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 720.

(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 720.

(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 721.

(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 721.

(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 722.

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 Global Statements: Reference.

libref.catalog
specifies a SAS catalog to write to.

See For information about the LIBNAME statement, see “LIBNAME Statement” in SAS Global 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 Global 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 720.

(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 720.

(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 721.

(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 721.

(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 722.

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 546
**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 336

**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 1109 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 543

(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 546

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 except SAS Viya

**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**

```sas
<(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 332

**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 793

**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 42.

See “ODS SHOW Statement” on page 806

**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.

<table>
<thead>
<tr>
<th>Alias</th>
<th>NAMED_DEST=</th>
<th>BOOKMARK=</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restriction</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Tips</td>
<td>You can change anchor names as often as you want by submitting the <code>ANCHOR=</code> 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.</td>
<td></td>
</tr>
</tbody>
</table>

**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: 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.

`BASE='http://www.your-company.com/local-url/'`
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 621

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 621

**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](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](#).
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.
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.

Example For information about the FILENAME statement, see “FILENAME Statement” in SAS Global 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 548

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 DATA Step 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 740.

See  For information about the FILENAME statement, see *SAS DATA Step 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 44.

**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 44.

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 740.

(<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 631

“Example 6: Adding Text That Imitates a System Title” on page 634

“Example 7: Toggling Page Breaks” on page 638

“Example 8: Suppressing a Page Break” on page 642

**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 576

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 740.

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 740.

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 740.

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
d-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

Interaction When the ACCESSIBLE= option is enabled, the PDFTOC= option is set to 0. The table of contents is fully expanded.

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 617

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 44.

**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 740.

**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 740.

**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.

<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><strong>Examples</strong></td>
<td>“Example 5: Combining a Table and Image on the Same Page” on page 631</td>
</tr>
<tr>
<td></td>
<td>“Example 6: Adding Text That Imitates a System Title” on page 634</td>
</tr>
<tr>
<td></td>
<td>“Example 7: Toggling Page Breaks” on page 638</td>
</tr>
<tr>
<td></td>
<td>“Example 8: Suppressing a Page Break” on page 642</td>
</tr>
</tbody>
</table>

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 638 |

YES

inserts a new page at the beginning of each procedure, and within certain procedures, as requested by the procedure code.

<table>
<thead>
<tr>
<th>Alias</th>
<th>ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>YES</td>
</tr>
</tbody>
</table>

**STYLE=style-template**

specifies the style template to use in writing the printer output.
### Subject

**Default**
If you do not specify a style template, then ODS uses the style template that is specified in the SAS registry subkey: `ODS DELEGATIONS PRINTER`. By default, this value is Perl 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” in SAS Output Delivery System: Procedures Guide.

**Examples**
“Example 3: Customizing the Table of Contents” on page 623

“Example 6: Adding Text That Imitates a System Title” on page 634

**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 336

**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.

```sas
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.

```sas
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 6.24  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:</td>
</tr>
<tr>
<td></td>
<td><code>filename your-fileref sysout=a dest=printer-name;</code></td>
</tr>
<tr>
<td></td>
<td><code>ods printer file=your-fileref;</code> Example:</td>
</tr>
<tr>
<td></td>
<td><code>filename local sysout=a dest=chpljj21;</code></td>
</tr>
<tr>
<td></td>
<td><code>ods printer file=local;</code></td>
</tr>
<tr>
<td>UNIX</td>
<td>Use the FILENAME statement with the PIPE command to associate a fileref with your <code>lpr</code> print command. Syntax:</td>
</tr>
<tr>
<td></td>
<td><code>filename your-fileref pipe 'lpr -P printer-name';</code></td>
</tr>
<tr>
<td></td>
<td><code>ods printer file=your-fileref;</code> Example:</td>
</tr>
<tr>
<td></td>
<td><code>filename local pipe 'lpr -P chpljj21';</code></td>
</tr>
<tr>
<td></td>
<td><code>ods printer file=local;</code></td>
</tr>
</tbody>
</table>
Windows

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.

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 736. 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 44 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 736 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:

options pdfsecurity=high pdfpw=(open=testpw);

The following code shows the PDF security option used with the PDF destination:
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 613.

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:**
- 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**

```sas
    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;
```
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.

ods html body='your_file.html';

create PostScript output. The ODS PRINTER 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;

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.

Output 6.92  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>Location</th>
<th>Variability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>3.468471</td>
</tr>
<tr>
<td>Median</td>
<td>2.114000</td>
</tr>
<tr>
<td>Mode</td>
<td>.</td>
</tr>
<tr>
<td></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>t</td>
<td>Pr &gt;</td>
</tr>
<tr>
<td>Sign</td>
<td>M</td>
<td>Pr &gt;=</td>
</tr>
<tr>
<td>Signed Rank</td>
<td>S</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 6.93  PostScript Output for the Variables CityPop_90 and CityPop_80
**ODS PROCLABEL Statement**

Enables you to change a procedure label.

**Valid in:** Anywhere except SAS Viya  
**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 634  
- “Example 3: Customizing the Table of Contents” on page 623

**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 except SAS Viya  
**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 631
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>
<thead>
<tr>
<th>Statement</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ODS PROCTITLE</strong></td>
<td><strong>ODS PROCTITLE=ON</strong></td>
</tr>
<tr>
<td></td>
<td><strong>ODS PTITLE</strong></td>
</tr>
<tr>
<td></td>
<td><strong>ODS PTITLE=ON</strong></td>
</tr>
<tr>
<td></td>
<td><strong>ODS PTITLE=YES</strong></td>
</tr>
<tr>
<td><strong>ODS NOPROCTITLE</strong></td>
<td><strong>ODS PROCTITLE=OFF</strong></td>
</tr>
<tr>
<td></td>
<td><strong>ODS NOPTITLE</strong></td>
</tr>
<tr>
<td></td>
<td><strong>ODS PTITLE=OFF</strong></td>
</tr>
<tr>
<td></td>
<td><strong>ODS PTITLE=NO</strong></td>
</tr>
</tbody>
</table>

**ODS PS Statement**

Opens, manages, or closes the PS destination, which produces PostScript (PS) output.

**Valid in:** Anywhere except SAS Viya

**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=`
  - 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.

Default | NONE
Restriction | A destination must be open for this action to take effect.
See “ODS EXCLUDE Statement” on page 332

**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 793

**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 42.

See “ODS SHOW Statement” on page 806

**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.

<table>
<thead>
<tr>
<th><strong>AUTHOR=</strong> 'author-text'</th>
<th>inserts the text string that you specify as the author into the metadata of a file.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>author-text</em> is the text in the metadata of an open file that indicates the author.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th><strong>BASE='base-text'</strong></th>
<th>specifies the text to use as the first part of all references that ODS creates in the output file.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>base-text</em> is the text that ODS uses as the first part of all references that ODS creates in the file.</td>
</tr>
</tbody>
</table>

Consider this specification: 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.

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

**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.

<table>
<thead>
<tr>
<th><strong>BOOKMARKLIST=</strong> HIDE</th>
<th>NONE</th>
<th>SHOW</th>
<th>specifies whether to generate and display the list of bookmarks for PDF and PS files.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>HIDE</strong> 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>NONE</strong> specifies not to generate a list of bookmarks for your PDF and PS files.</td>
</tr>
<tr>
<td><strong>Aliases</strong></td>
<td></td>
<td></td>
<td>NO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NOBOOKMARKLIST is an alias for BOOKMARKLIST=NONE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>SHOW</strong> generates a list of bookmarks for your PDF and PS files. The bookmarks are automatically displayed when you open the PDF and PS files.</td>
</tr>
<tr>
<td><strong>Aliases</strong></td>
<td></td>
<td></td>
<td>YES</td>
</tr>
</tbody>
</table>
**BOOKMARKLIST** is an alias for **BOOKMARKLIST=SHOW | YES | ON**.

**Example**

“Example 2: Creating a Printable Table of Contents” on page 621

**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 621

**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 Global 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 548
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 DATA Step 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 740.
See For information about the FILENAME statement, see *SAS DATA Step 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 44.

**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 44.

**(<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 \textit{MAY5. PS}. Additional body files are named \textit{MAY6. PS}, \textit{MAY7. PS}, and so on.

Example:
\texttt{FILE= 'MAY5. PS'}

\textbf{PACKAGE <package-name>}

specifies that the output from the destination be added to a package.

\textit{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 \textit{“ODS PACKAGE Statement”} on page \texttt{576}

\textbf{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 \texttt{URL=} style attribute.
- notes for items that are associated with the \texttt{FLYOVER=} style attribute. Notes are optional, and are based on the PDFNOTE option.
- author, keywords, subject, and title in the metadata of a file.

\textbf{Default}

Because using PDFMARK implies PostScript output, SAS automatically uses the PostScript driver that SAS supplies with this option.

\textbf{Restriction}

You cannot use the PRINTER= option with the PDFMARK option.

\textbf{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.

\textbf{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 \textit{“Opening and Closing the PRINTER Destination”} on page \texttt{740}.

\textbf{Tip}

Use this option only if you plan to distill the output. Otherwise, it uses excess resources and does not enhance the results.

\textbf{PDFNOTE | NOPDFNOTE}

controls whether notes are added to a PDF file for items that are associated with the \texttt{FLYOVER=} style attribute.

\textbf{PDFNOTE}

adds notes to a PDF file for items that are associated with the \texttt{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 631

“Example 6: Adding Text That Imitates a System Title” on page 634

“Example 7: Toggling Page Breaks” on page 638

“Example 8: Suppressing a Page Break” on page 642

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 638

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” in SAS Output Delivery System: Procedures Guide.

Examples

“Example 3: Customizing the Table of Contents” on page 623

“Example 6: Adding Text That Imitates a System Title” on page 634

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 336

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 38

**Statements**

- “ODS PCL Statement” on page 587
- “ODS PDF Statement” on page 597
- “ODS PRINTER Statement” on page 722

---

**ODS RESULTS Statement**

Tracks ODS output in the Results window.

**Valid in:** Anywhere except SAS Viya

**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.

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:

```
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 787 for details.

Tips: When producing large tables, use the ODS TAGSETS.RTF statement. For detailed information, see “ODS TAGSETS.RTF Statement ” on page 838.
Starting in SAS 9.4, cellx values are computed for all ODS destinations based on CSS standards. In the RTF destination, an attempt is made to honor the cellx value
of 1440 twips to 1 inch. The final cellx value takes into account the text width, the accumulated border widths and accumulated padding. An additional 2% font padding is added to the text measurement when an explicit dimension has not been provided.

## 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

NOTRKEEP
Control whether table rows can be split by a page break

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 332

**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 793

**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 42.

See  “ODS SHOW Statement” on page 806

**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.

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.

See  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.

See  BODYTITLE option

Example  “Example 2: Justifying Title and Footnotes When You Specify the BODYTITLE_AUX Option” on page 785

**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.

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 781

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 Global Statements: Reference.

"URL"

is a URL to an external file.
Requirement You must enclose *URL* in quotation marks.

(\textit{media-type-1}..<\textit{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 \textit{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 \textit{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 \textit{media-type} in parentheses.

You must specify \textit{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 548

**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 365.

Table 6.26  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>Excel</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>PowerPoint</td>
<td>PNG</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 44.


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.
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 DATA Step 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 780.

**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 617

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.

**NOTRKEEP**
specifies that the table row is not kept together and can be split by a page break.

**Default** TRKEEP

**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 576
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.

See “FILENAME Statement” in SAS Global Statements: Reference.

libref.catalog
specifies a SAS catalog to write to.

See “LIBNAME Statement” in SAS Global 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.
The `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” 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.

**Restriction** 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.

**Requirement** You must enclose a `text-string` in quotation marks.

**Tips** A better way to add text to your RTF output is to use the ODS TEXT= statement or the PROC ODSTEXT statement.

To add text and start a new page, use the ODS TEXT= statement and the ODS TAGSETS.RTF statement with the STARTPAGE=NOW option instead of using ODS TAGSETS.RTF TEXT= and STARTPAGE= options.

**Example** “Example 8: Using the ODS TEXT= statement to Add Text to RTF OUTPUT” on page 882

**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.

**Requirement** 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:

\texttt{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 781

\texttt{TRANTAB=\textit{translation-table}}

specifies the translation table for ODS to use when it transcodes a file for output.

See  For more information, see “\texttt{TRANTAB=} System Option” in \textit{SAS National Language Support (NLS): Reference Guide.}

\section*{Details}

\textbf{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:

\begin{itemize}
  \item close the open destination referred to in the ODS RTF statement
  \item close any files associated with the open RTF destination
  \item open a new instance of the RTF destination
\end{itemize}

If you use the FILE= option, you should explicitly close the destination yourself.

\textbf{Understanding How RTF Formats Output}
RTF produces output for Microsoft Word. 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 838.

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 6.27 Graphics Formats Supported by ODS RTF**

<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, 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**

```
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.;
   define type      / group 'Type of Grain';
   define kilotons  / format=comma12.;
   footnote 'Measurements are in metric tons.';
run;
options byline;
title2;
ods rtf notoc_data;
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 rtf toc_data;
proc print data=Grain_Production;
run;
ods rtf close;
ods html;
```

**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';```
```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, 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.

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 TOC_DATA option before the PRINT procedure, ODS inserts contents data, and an entry for PROC PRINT appears in the table of contents.

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 838.

**Program**

```plaintext
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=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";
```
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 is placed into cells.
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;
ods html close;
title 'Page Orientation';
```
title2 'Default';
ods rtf file="ChgOrientation.rtf";
proc print data=sashelp.class (obs=1); run;
title 'Page Orientation';
title2 'Landscape';
options orientation=landscape;
ods rtf;
proc print data=sashelp.class (obs=1); run;
ods rtf close;
ods html;

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.

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;

Add titles and footnotes to the output. Add a title for the overall file output and then titles that describe the changing orientation.

    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.

    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.

    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.

    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.

Example 4: RTF Using NOTRKEEP Option
The default behavior when a single cell does not fit on a page is for Microsoft Word to truncate the cell. When the NOTRKEEP option is specified, the cell splits at a page break and continues on the next page. The following example provides example code for specifying the NOTRKEEP option within an ODS RTF statement.

Program

```
data test;
  length string $ 4010;
  string=repeat('1',1000)||repeat('2',1000)||repeat('3',1000)||repeat('4',1000)||'END';
  output;
  string='ABC';
  output;
run;
ods rtf body="NOTRKEEP.rtf" NOTRKEEP ;

proc print data=test;
run;
ods rtf close;
```

Program Description

**Load the data set.**

```
data test;
  length string $ 4010;
  string=repeat('1',1000)||repeat('2',1000)||repeat('3',1000)||repeat('4',1000)||'END';
  output;
  string='ABC';
```
Specify the NOTRKEEP option in the ODS RTF statement to enable a page break.

```sql
ods rtf body="NOTRKEEP.rtf" NOTRKEEP ;
proc print data=test;
run;
```

Close the RTF destination.

```sql
ods rtf close;
```
RTF Output

The following ODS RTF output illustrates the use of the NOTRKEEP option. Microsoft Word uses the next page to display the full value of the STR variable.
The following ODS RTF output illustrates the default of the NOTRKEEP option (TRKEEP).

```
The SAS System

Data: STRING
  #  AEC
```

**ODS SELECT Statement**

Specifies output objects for ODS destinations.

- **Valid in:** Anywhere except SAS Viya
- **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 332
- **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 42. 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" "Tests For Location"

• 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.

See “ODS TRACE Statement” on page 891

(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

Requirement 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.

```plaintext
ods select summary (nowarn);
proc contents data=sashelp.class;
run;
```

### ODS-destination

specifies to which ODS destination's selection list to write, where ODS-destination can be any valid ODS destination except for the OUTPUT destination.

#### Default

If you omit ODS-destination, ODS writes to the overall selection list.

#### Restriction

You cannot write to the OUTPUT destination's selection list.

#### Tip

To set the selection list for the Output destination to something other than the default, see the “ODS OUTPUT Statement” on page 555.

#### See

“Understanding ODS Destinations” on page 36 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:

```plaintext
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:

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 name 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>
<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 555. For a list of ODS output destinations and explanations of each, see “Understanding ODS Destinations” on page 36.

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 793, the “ODS EXCLUDE Statement” on page 332, and the “ODS TRACE Statement” on page 891.

### 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=
  - FRAME=
  - PAGE=

**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**

```sas
ods html body='odspersist-body.htm'
   frame='odspersist-frame.htm'
   contents='odspersist-contents.htm'
   page='odspersist-page.htm';
ods show;
ods select ParameterEstimates
   "Type III Model ANOVA";
ods show;
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 select all;
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'
```
Example 1: Using a Selection List with Multiple Procedure Steps

```sas
loss='Weight Loss';
title 'Plot of Iron Versus Loss';
run;
quit;
ods html close;

Program Description

Create HTML output. The ODS HTML statement 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.

```sas
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 802.

```sas
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.

```sas
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 802.

```sas
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 802. 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 802.

    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.

    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 802.

    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 802. The QUIT statement ends the procedure. Because the list uses the argument ALL, ODS does not automatically modify it when the PROC step ends.

    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.

    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 802.

    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 802.

```sas
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 802.

```sas
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.

```sas
quit;
```

PROC PRINT does not produce any output that is named OverallANOVA. Therefore, no PROC PRINT output is sent to the ODS destinations.

```sas
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.

```sas
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).

```sas
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.

```sas
quit;
```

Close the HTML destination. This ODS HTML statement closes the HTML destination and all the files that are associated with it.

```sas
ods html close;
```
SAS Log

Output 6.94  The ODS SHOW Statement Writes the Current Selection List to the SAS Log.

```
%Log -(Untitled)
20 ods html body=ods_persist-body.htm'
21 frame=ods_persist-frame.htm'
22 contents=ods_persist-contents.htm'
23   pages=ods_persist-page.htm';ods show;ods select ParameterEstim
NOTE: Writing HTML Body file: ods_persist-body.htm
NOTE: Writing HTML Contents file: ods_persist-contents.htm
NOTE: Writing HTML Frame file: ods_persist-frame.htm
Current OVERALL select list is: ALL
14  'Type III Model ANOVA'; ods show; proc glm data=iron;
Current OVERALL select list is: ALL
1. ParameterEstimates
2. 'Type III Model ANOVA'
   model losse=Fe;
   title 'Parameter Estimates and Type III Model ANOVA';
   run;
Current OVERALL select list is: ALL
18  model losse=Fe;
19   title 'All Output Objects Selected';
20  run;
21 quit;
NOTE: PROCEDURE GLM used (Total process time):
   real time 0.07 seconds
   cpu time 0.04 seconds

21 ! ods select OverallANOVA(persist) "Fit Statistics"; proc glm data=iron;
22   model losse=Fe;
23   title 'OverallANOVA and Fitness Statistics';
24  run;
24 ! quit;
NOTE: PROCEDURE GLM used (Total process time):
   real time 1.13 seconds
   cpu time 0.18 seconds

24 ! ods show; proc glm data=iron;
Current OVERALL select list is:
1. OverallANOVA(PERSIST)
25  model losse=Fe;
26   title 'OverallANOVA';
27   title2 'Part of the Selection List Persisted';
28  run;
28 ! quit;
```

HTML Output

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;
```
Program Description

Create the BPressure data set.

```sas
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;
```

Add a title.

```sas
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 6.96** SAS Log Including Trace Record

```
log - (Unlabeled)
1 data BPressure;
2   length PatientID %24;  
3   input PatientID $ Systolic Diastolic $;  
NOTE: SAS went to a new line when INPUT statement reached past the end of a line.  
NOTE: DATA statement used (Total process time):  
   real time 0.25 seconds  
   cpu time 0.21 seconds  
4 datalines;  

NOTE: ODS sent a new line when INPUT statement reached past the end of a line.  
NOTE: The data set WORK.BPRESSURE has 32 observations and 3 variables.  
NOTE: DATA statement used (Total process time):  
   real time 0.25 seconds  
   cpu time 0.21 seconds  
5 run;  
6   title 'Systolic and Diastolic Blood Pressure';ods trace on / label excluded;  
7   ods select where(_path_ ? "Diastolic" and _name_='Moments') ; proc univariate  
8   data BPressure;  
NOTE: Writing HTML Body file: sashtml.htm  
9 var Systolic Diastolic;  
10 run;
```

Output Excluded:  
----------------
Name: Moments  
Label: Moments  
Template: base.univariate.Moments  
Path: Univariate.Systolic.Moments  
Label Path: 'The Univariate Procedure'.'Systolic'.'Moments'  

Output Excluded:  
----------------
Name: BasicMeasures  
Label: Basic Measures of Location and Variability  
Template: base.univariate.BasicMeasures  
Path: Univariate.Systolic.BasicMeasures  
Label Path: 'The Univariate Procedure'.'Systolic'.'Basic Measures of Location and Variability'  

Output Excluded:  
----------------
Name: TestForLocation  
Label: Tests For Location  
Template: base.univariate.Location  
Path: Univariate.Systolic.TestForLocation  
Label Path: 'The Univariate Procedure'.'Systolic'.'Tests For Location'  

Output Excluded:  
----------------
Name: Quantiles  
Label: Quantiles  
Template: base.univariate.Quantiles  
Path: Univariate.Systolic.Quantiles  
Label Path: 'The Univariate Procedure'.'Systolic'.'Quantiles'  
```
See Also

Statements

• “ODS EXCLUDE Statement” on page 332
• “ODS SHOW Statement” on page 806
• “ODS TRACE Statement” on page 891

ODS SHOW Statement

Writes the specified selection or exclusion list to the SAS log.

Valid in: Anywhere except SAS Viya
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 36. For information about selection and exclusion lists, see “Selection and Exclusion Lists” on page 42.
If you omit ODS-destination, ODS SHOW writes the overall selection or exclusion list.

See Also

Statements

• “ODS EXCLUDE Statement” on page 332
• “ODS SELECT Statement ” on page 793
• “ODS TRACE Statement” on page 891

ODS Tagset Statement

Opens, manages, or closes the specified tagset destination.

Valid in: Anywhere except SAS Viya
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 538.

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 509.

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 332

**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 793

**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 42.

See “ODS SHOW Statement” on page 806

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 833.

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 174

- CORE contains a table of Unicode values and mnemonics. For a detailed description of using this tagset, see “Using Unicode Symbols” on page 276.

- 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 200

- 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:

```ods tagsets.excelxp file='test.xml' options(doc='help');```

HTML4 produces HTML 4.0 embedded style sheets.

See “ODS HTML Statement ” on page 396

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:

```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 695

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 838.

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. ODS assumes that if the anchor name ends in a number, then the number should be treated as a starting point for incrementing the 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.

Similarly, if you specify ANCHOR="TABULATE1", then ODS names the first anchor `tabulate1`. The second anchor is named `tabulate2`.

**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:

```
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.yourcompany.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 Global 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 833.
(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 834.

(NO_TOP_MATTER)
See For complete documentation about the NO_TOP_MATTER suboption, see “(NO_TOP_MATTER)” on page 834.

(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 835.

(URL='Uniform-Resource-Locator')
See For complete documentation about the URL= suboption, see “(URL='Uniform-Resource-Locator’)” on page 835.

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 538.

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.

Tip When you use the ODS HTML statement with the FILE= option to specify a fully-qualified path with a directory and a file name, the best practice for generating HTML files is to separate the path from the file. To accomplish this, use a combination of the PATH= option, which specifies the directory, and the FILE= option, which names the actual file. For example:

```
ods html path="c:\temp" file="myfile.html";
proc print data=sashelp.class;
run;
ods html close;
```

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 Global 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 833.

  • (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 835.

**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 Global 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 833.

**/(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 834.

**/(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 834.

**(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 835.

**(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 835.

**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 Global 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 548
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.
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 833.

(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 834.

(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 834.

(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 835.

(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 835.

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 541
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.catalo(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.

Requirements
You must enclose aggregate-file-storage-location in quotation
marks.

The specified aggregate-file-storage-location must exist in the
file system.

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 Global Statements: Reference.

libref.catalo
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 617

**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:

```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.

**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= ) | sub-option(s)**

specifies ODS tagset-specific suboptions and a named value.

```plaintext
(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 837

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 Global 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 833.

(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 834.

(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 834.

(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 835.

(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 835.

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 Global Statements: Reference*.

*libref.catalog*

specifies a SAS catalog to write to.

**See** For information about the LIBNAME statement, see “LIBNAME Statement” in *SAS Global 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**

```plaintext
RECSEP=
```

**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 GET DESTINATIONS GET 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 Global 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 833.

- `(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 834.

(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 834.

(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 835.

(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 835.

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 546

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 336

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 833, 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.

**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.

**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 1109 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.

<table>
<thead>
<tr>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>The NO_BOTTOM_MATTER suboption and the ANCHOR= option</td>
</tr>
</tbody>
</table>

**(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.

<table>
<thead>
<tr>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>You must enclose TITLE= in parentheses.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tip</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Example 3: Creating Multiple Markup Output” on page 543</td>
</tr>
</tbody>
</table>

**(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.

<table>
<thead>
<tr>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>You must enclose URL= 'Uniform-Resource-Locator' in parentheses.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tips</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Example 5: Including Multiple Cascading Style Sheets in One HTML Document” on page 546</td>
</tr>
</tbody>
</table>
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.cthtml;
```
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.

**Output 6.97 Options Available for the HTMLPANEL Tagset**

See Also


**Statement**

- “ODS MARKUP Statement” on page 509

---

**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 except SAS Viya
- **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.

### Syntax

```sas
ODS TAGSETS.RTF <(<ID=> identifier)> action;
ODS TAGSETS.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

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

Restriction A destination must be open for this action to take effect.

See “ODS EXCLUDE Statement” on page 332
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 793

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 42.

See              “ODS SHOW Statement” on page 806

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.
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/*.

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
```

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

**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 365.

Table 6.29  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>Excel</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>PowerPoint</td>
<td>PNG</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 44.


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 DATA Step 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 780.

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 617

**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.

**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.
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= | TROWHDR= | 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 848. 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 858

(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 858

(DOC='QUICK' | 'HELP' | 'SETTINGS')
provides information about the 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.

Example: "Example: Using the DOC Suboption to Get ODS TAGSETS.HTMLPANEL Information" on page 837

(SECT='rtf_control_string' | 'OFF' | 'NONE')
inserts RTF control words into the section data specifications.

rtf_control_string
specifies RTF control words used to format the section data.

OFF
turns off the usage of RTF control words for the section data and resets the rtf_control_string to null.

Alias: NO

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 rtf_control_string, assign a different value or use the OFF or NO values.


(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.
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 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  “ODS Style Elements” on page 1045 for information about style elements

Example  “Example 3: Using the TABLES_OFF Suboption” on page 863

(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 858

(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 858.

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 858

(TROWD='rtf_control_string' | 'OFF')
inserts raw RTF specifications directly into header descriptions of the table row.

\textit{rtf\_control\_string}
\begin{itemize}
\item \textbf{OFF}
\begin{itemize}
\item RTF controls are no longer inserted.
\end{itemize}
\end{itemize}

\textbf{Alias} \textbf{NO}

\textbf{Requirement} You must enclose all values in quotation marks.

\textbf{Tip} If the RTF code inserted in the document is invalid, the code is either ignored or causes the document to be unusable.

\textbf{See} Rich Text Format (RTF) Specification, version 1.6 available on the MSDN home web page for information about RTF control words. Search for the RTF 1.6 document.

\textbf{Example} “Example 4: Column Heading Rotation Using the \texttt{TRHDR}, \texttt{TROWHDRCELL}, and \texttt{TROWD} Options” on page 865

(TRHDR='rtf\_control\_string' | 'OFF')
inserts raw table row RTF specifications directly into the header description of the table row.

\textit{rtf\_control\_string}
\begin{itemize}
\item \textbf{OFF}
\begin{itemize}
\item RTF controls are no longer inserted.
\end{itemize}
\end{itemize}

\textbf{Alias} \textbf{NO}

\textbf{Requirement} You must enclose all values in quotation marks.

\textbf{Tip} If the RTF code inserted in the document is invalid, the code is either ignored or causes the document to be unusable.

\textbf{See} Rich Text Format (RTF) Specification, version 1.6 available on the MSDN home web page for information about RTF control words. Search for the RTF 1.6 document.

\textbf{Example} “Example 4: Column Heading Rotation Using the \texttt{TRHDR}, \texttt{TROWHDRCELL}, and \texttt{TROWD} Options” on page 865
(TROWHDRCELL='text_string' | 'OFF')

inserts raw text into the table row cells. If the RTF Reader does not recognize this
\textit{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 \textit{text_string} as
RTF control words.

\textit{text_string}

any text specified.

\textbf{OFF}

inserts a null string. Text is no longer inserted.

\textbf{Alias} \textit{NO}

\textbf{Requirement} You must enclose all values in quotation marks.

\textbf{See} Rich Text Format (RTF) Specification, version 1.6 available on
the MSDN home web page for information about RTF control
words. Search for the RTF 1.6 document.

\textbf{Example} “Example 4: Column Heading Rotation Using the TRHDR,
TROWHDRCELL, and TROWD Options” on page 865

(VSPACE='ON' | 'OFF')

specifies whether to add or remove space before and after tables.

\textit{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.

\textbf{OFF}

specifies that no space is added before and after tables in the RTF file.

\textbf{Alias} \textit{NO}

\textbf{ON}

specifies that space is added before and after tables in the RTF file.

\textbf{Alias} \textit{YES}

\textbf{Default} \textit{YES or ON}

\textbf{Requirement} You must enclose all values in quotation marks.

\textbf{Example} \texttt{ods tagsets.rtf file='myrtf.rtf' OPTIONS(VSPACE='NO');}

\textbf{Example} “Example 7: Spacing Using Parskip Style Element and
VSPACE= Suboption” on page 879

(WATERMARK='text_string' | ' ')

inserts a watermark that is displayed diagonally across each page of the RTF
document.

\textit{text_string}

specifies the text string that appears diagonally across each page of the RTF
document.

\texttt{\textbackslash}

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 576

**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 867

**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.
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**


**libref.catalog**

specifies a SAS catalog to write to.

**See**

“LIBNAME Statement” in SAS Global 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`.

<table>
<thead>
<tr>
<th>Alias</th>
<th>NEVER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction</td>
<td>When you specify the <code>COLUMNS=</code> option, the <code>STARTPAGE=NO</code> option is not honored.</td>
</tr>
</tbody>
</table>

**NOW**

forces the immediate insertion of a new page.

<table>
<thead>
<tr>
<th>Tip</th>
<th>This option is useful primarily when the current value of the <code>STARTPAGE=</code> option is NO. Otherwise, each new procedure forces a new page automatically.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>YES</td>
</tr>
</tbody>
</table>

| 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.

<table>
<thead>
<tr>
<th>Note</th>
<th>If you are using SAS Studio, you do not need to specify the <code>STYLE=</code> option. You can go to Preferences ⇒ Results and change the style from the drop-down list for your selected destination.</th>
</tr>
</thead>
<tbody>
<tr>
<td>See</td>
<td>For a complete discussion of style templates, see “TEMPLATE Procedure” in <em>SAS Output Delivery System: Procedures Guide</em>.</td>
</tr>
<tr>
<td>Default</td>
<td>If you do not specify a style template, ODS uses the file that is specified in the SAS registry subkey: <code>ODS ⇒ DESTINATIONS ⇒ RTF</code>. By default, this value specifies <code>RTF</code> for traditional RTF and Measured RTF.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Note</th>
<th>Page breaks are not forced between panels.</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>n</code></td>
<td>is a positive integer.</td>
</tr>
<tr>
<td>Alias</td>
<td>0</td>
</tr>
<tr>
<td>-------</td>
<td>---</td>
</tr>
<tr>
<td>Default</td>
<td>Allow SAS to determine the number of rows per table.</td>
</tr>
<tr>
<td>Tip</td>
<td>0 or NONE returns to the default, which allows SAS to determine the number of rows per table.</td>
</tr>
<tr>
<td>Example</td>
<td>“Example 5: Paneling Using the TABLEROWS and PAGEPANELS Options” on page 867</td>
</tr>
</tbody>
</table>

**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.

**Tips**

A better way to add text to your RTF output is to use the ODS TEXT= statement or the PROC ODSTEXT statement.

To add text and start a new page, use the ODS TEXT= statement and the ODS TAGSETS.RTF statement with the STARTPAGE=NOW option instead of using ODS TAGSETS.RTF TEXT= and STARTPAGE= options.

**Example**

“Example 8: Using the ODS TEXT= statement to Add Text to RTF OUTPUT” on page 882

**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*. 
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 6.30  Supported Graphics Formats for Measured RTF

<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, JAVAIMG graphics drivers to generate graphics in your measured 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 TAGSET.RTF statement, the JAVAIMG 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

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');
opts nobyline;
Example 1: Creating a Table of Contents

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 options(toc_data='no');
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 tagsets.rtf options(toc_data='yes');
proc print data=Grain_Production;
run;
ods tagsets.rtf close;
ods html;
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.

```r
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.

```r
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 variable. The TITLE2 statement clears the second TITLE statement.

```r
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.

```r
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.

```r
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.
ods tagsets.rtf options(toc_data='yes');

Print the Grain_Production data set.
   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.
   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.

```sas
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.

```sas
proc print data=Sashelp.Class;
run;
```

Close the TAGSETS.RTF destination. If you do not close this destination, then you cannot view the output.

```sas
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")
  - Option: FILE=
  - Option: TEXT=

---

Log (Unlinked)

```sas
ods tagsets.rtf file='help.rtf' options (doc='help');
NOTE: Writing TAGSETS.RTF body file: help.rtf
---------------------------------------------------------------------
The RTF Tagset Help Text.
This Tagset/Destination helps with the creation of RTF files
for MS Word
---------------------------------------------------------------------
These are the options currently supported by this tagset.
Sample usage:
ODS TAGSETS.RTF OPTIONS(doc='Quick',
                  CONTENTS='yes');
ODS TAGSETS.RTF OPTIONS(SPECIFIC_OPTION='value');

Debug Level: No default value.
  Current value: 0
  Usage: OPTIONS(Debug Level=)
  Description:
    Determine what debugging information should be printed
to the log. The values expected are numeric and can be used to
take whatever action is needed. Used in tagsets being debugged,
but requires a local tagset to be modified.

Doc: No default value.
  Help: Displays introductory text and options.
  Quick: Displays available options.
  Settings: Displays Current settings.

CONTENTS: No default value
  Usage: OPTIONS(CONTENTS='yes')
  Description:
    Adds a table of contents page at the top of the file,
    followed by a page break. This must occur before any other output
    in order to have an effect.
    'yes' and 'on' have the same action.

SECT: No default value.
  Usage: OPTIONS(SECT='string')
  Description:
    Inserts RTF controls onto the section data specifcations.
    ODS tagsets.rtf OPTIONS(sect='string')
    The special string 'NONE' prevents any section data from
    being added.
    Assigning 'no' or 'off' resets the option to NULL.

TABLES_OFF: No default value.
```
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
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

```sas
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.

```sas
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.

```sas
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.

```sas
ods tagsets.rtf file='Mrotate.rtf'
OPTIONS (TRHDR='\trrh750'
TROWHDRCELL='\cltxbtlr'
TROWD='\rtlrow');
```

Print the Sashelp.Class data set.

```sas
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
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

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;
   output;
end;
run;
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”.

```plaintext
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'.

```plaintext
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.

```plaintext
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 6.98** RTF Output with Default Page Paneling - Page 1

---

**Default Paneling**

<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>
<tr>
<td>6</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>
<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>

<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>

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 6.99**  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>

### Paneling with TABLEROWS=5 and PAGEPANELS=2

<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

<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>6</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>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;
  end;
run;
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'

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>6</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>6</td>
<td>51</td>
</tr>
</tbody>
</table>

872 Chapter 6 • Dictionary of ODS Language Statements
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;
ods rtf file='RtfTab.rtf';
```
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.

```
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.

```
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.

```
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.
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.

title2 'My Favorite Cars';
title3 '(large data set)';

Produce a report. PROC TABULATE creates the table of cars and their attributes.

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.

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
carried to each page and that the word “Continued” appears at the bottom of each page of RTF output.

Output 6.100  Measured RTF Output

<table>
<thead>
<tr>
<th>DATSUN 260AX</th>
<th>RED</th>
<th>ALL THE WAY</th>
<th>POWER STEERING</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>VERY SLOW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SLOW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AVERAGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FAST</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VERY FAST</td>
</tr>
</tbody>
</table>

| SUN ROOF | VERY SLOW    |
|          | SLOW         |
|          | AVERAGE      |
|          | FAST         |
|          | VERY FAST    |

| AUTOMATIC | VERY SLOW    |
|           | SLOW         |
|           | AVERAGE      |
|           | FAST         |
|           | VERY FAST    |

<table>
<thead>
<tr>
<th>STANDARD OPTIONS</th>
<th>POWER STEERING</th>
</tr>
</thead>
<tbody>
<tr>
<td></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>

| SUN ROOF | VERY SLOW    |
|          | SLOW         |
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
page two of the output. Also note that page one does not indicate that more pages of output follow.

**Output 6.101 Traditional RTF Output**

```markdown
<table>
<thead>
<tr>
<th>DATSUN 280SX</th>
<th>RED</th>
<th>ALL THE WAY</th>
<th>POWER STEERING</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>VERY SLOW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SLOW</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>AVERAGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FAST</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VERY FAST</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

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;
          style parskip / fontsize = 40pt;
    end;

Program Description

**Specify the system options.** The NODATE option turns off the output of the date and time.

```
option nodate;
```

**Close the HTML destination.** The HTML destination is open by default.

```
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.

```
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.

```
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.

```
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 6.102  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 6.103** 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

**Example 8: Using the ODS TEXT= statement to Add Text to RTF OUTPUT**

**Features:**
- ODS TAGSETS.RTF statement:
  - Option: STARTPAGE=NOW
- ODS TEXT= statement

**Other features:**
- PROC PRINT

The following example shows how to use the ODS TEXT= statement to add text to your RTF output. It also shows how to use the STARTPAGE=NOW option to create a new page.

**Program**

```r
don_tagsets.rtf file="InsertNewPage.doc";
ods text=" This text appears on the first page";
don_tagsets.rtf startpage=now;
don_tagsets.rtf text="This text starts on the second page";
title;
proc print data=sashelp.class(obs=10) ;
run;
don _all_ close;
```

**Program Description**

- Open the ODS TAGSETS.RTF destination.
  ```r
don_tagsets.rtf file="InsertNewPage.doc";
```

- Use the ODS TEXT= statement to add text to your output.
  ```r
don_text=" This text appears on the first page";
```
Start a new page.

    ods tagsets.rtf startpage=now;

Add text and data output to the second page of RTF output. Use the ODS TEXT= statement to add text to the second page of your output. Then add a title before the data set is printed.

    ods text="This text starts on the second page";
    title;
    proc print data=sashelp.class(obs=10) ;
    run;

Close the RTF destination. If you do not close this destination, then you cannot view the output.

    ods _all_ close;

RTF Output

Add text to your RTF output using the ODS TEXT= statement. The first page of output includes text. A second page of output is added using the STARTPAGE=NOW option.

A second page of output is created by specifying the STARTPAGE=NOW option on the ODS TAGSETS.RTF statement. Text is added to the second page using the ODS TEXT= statement.
Valid in: Anywhere except SAS Viya
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**

```
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 templates 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 Exprev. The PRINT procedure prints the Exprev data set.

```proc print data=exprev;
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 6.104  HTML Output with Text Added

![January Orders](image)

<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>9.26</td>
<td>20.70</td>
</tr>
<tr>
<td>2</td>
<td>Puerto Rico</td>
<td>99999999</td>
<td>1/1/05</td>
<td>1/5/05</td>
<td>Catalog</td>
<td>14</td>
<td>51.2</td>
<td>121.10</td>
</tr>
<tr>
<td>3</td>
<td>Virgin Islands (U.S.)</td>
<td>99999999</td>
<td>1/1/05</td>
<td>1/4/05</td>
<td>In Store</td>
<td>25</td>
<td>31.1</td>
<td>15.65</td>
</tr>
<tr>
<td>4</td>
<td>Andorra</td>
<td>99999999</td>
<td>1/1/05</td>
<td>1/4/05</td>
<td>Catalog</td>
<td>30</td>
<td>125.7</td>
<td>30.00</td>
</tr>
<tr>
<td>5</td>
<td>Bahamas</td>
<td>99999999</td>
<td>1/1/05</td>
<td>1/4/05</td>
<td>Catalog</td>
<td>6</td>
<td>113.4</td>
<td>20.45</td>
</tr>
<tr>
<td>6</td>
<td>Bermuda</td>
<td>99999999</td>
<td>1/1/05</td>
<td>1/4/05</td>
<td>Catalog</td>
<td>7</td>
<td>41.0</td>
<td>9.25</td>
</tr>
<tr>
<td>7</td>
<td>Botswana</td>
<td>120492</td>
<td>1/2/05</td>
<td>1/10/05</td>
<td>In Store</td>
<td>2</td>
<td>46.4</td>
<td>36.70</td>
</tr>
<tr>
<td>8</td>
<td>British Virgin Islands</td>
<td>99999999</td>
<td>1/2/05</td>
<td>1/10/05</td>
<td>Catalog</td>
<td>11</td>
<td>40.2</td>
<td>20.20</td>
</tr>
<tr>
<td>9</td>
<td>Canada</td>
<td>99999999</td>
<td>1/2/05</td>
<td>1/6/05</td>
<td>Catalog</td>
<td>100</td>
<td>116</td>
<td>5.00</td>
</tr>
<tr>
<td>10</td>
<td>Cayman Islands</td>
<td>120494</td>
<td>1/2/05</td>
<td>1/10/05</td>
<td>In Store</td>
<td>30</td>
<td>71.0</td>
<td>32.30</td>
</tr>
</tbody>
</table>

For All Employees
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 965.
- For a table of style attributes, see “Style Attributes Tables” on page 1074.
- For a table of style elements affecting pages, see “Style Elements Affecting Pages” on page 1047.
- For a table of style elements affecting the table of contents, see “Style Elements Affecting Tables of Contents” on page 1046.

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;
runc:

ods text="My Custom PROC PRINT Output Title";

proc print data=sashelp.cars noobs;
  where mpg_highway gt 45;
runc:
```
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.

ods pdf close;
ods html;
proc template;
delete Styles.Mimictitle;
run;
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;
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;

After your output is produced, you can remove the custom style. The DELETE statement in PROC TEMPLATE removes the custom style.

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 6.107  PDF Output with Custom Procedure Title

![Image showing the PROC PRINT output with custom 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 except SAS Viya
- **Category:** ODS: Output Control
- **Default:** OFF
- **Example:** “Example 3: Creating Merged Data Sets” on page 565

**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 803

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 41. 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 893.

### 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:
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**

```plaintext
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.

```plaintext
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.

```plaintext
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.

```
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:
-------------
Name: Moments
Label: Moments
Template: base.univariate.Moments
Path: Univariate.CityPop_80.Moments
Label Path: "The Univariate Procedure"."CityPop_80"."Moments"
-------------
Output Added:
-------------
Name: BasicMeasures
Label: Basic Measures of Location and Variability
Template: base.univariate.Measures
Path: Univariate.CityPop_80.BasicMeasures
Label Path: "The Univariate Procedure"."CityPop_80"."Basic Measures of Location and Variability"
-------------
Output Added:
-------------
Name: TestsForLocation
Label: Tests For Location
Template: base.univariate.Location
Path: Univariate.CityPop_80.TestsForLocation
Label Path: "The Univariate Procedure"."CityPop_80"."Tests For Location"
-------------
Output Added:
-------------
Name: Quantiles
Label: Quantiles
Template: base.univariate.Quantiles
Path: Univariate.CityPop_80.Quantiles
Label Path: "The Univariate Procedure"."CityPop_80"."Quantiles"
-------------
Output Added:
-------------
Name: ExtremeObs
Label: Extreme Observations
Template: base.univariate.ExtObs
Path: Univariate.CityPop_80.ExtremeObs
Label Path: "The Univariate Procedure"."CityPop_80"."Extreme Observations"
-------------

See Also

Statements

- “ODS EXCLUDE Statement” on page 332
- “ODS SELECT Statement ” on page 793

ODS USEGOPT Statement

Determines whether ODS uses traditional SAS/GRAPH option settings.

Valid in: Anywhere except SAS Viya
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.
See: SAS/GRAPH: Reference

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

```plaintext
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.

```plaintext
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.

```plaintext
goptions reset=all htext=2 ftitle=script ftext=script;
```

**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.

```plaintext
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;
```

**Disable the graphics options.** The NOUSEGOPT statement suppresses the use of the HTEXT= and CTEXT= graphics option settings for your output.

```plaintext
goptions reset=all htext=2 ftitle=script ftext=script;
```

**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>999999999</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 Rico</td>
<td>999999999</td>
<td>1/1/05</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 except SAS Viya
- **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 585.
**Syntax**

**ODS VERIFY** `<ON | OFF | ERROR | WARN>` ;

**Optional Arguments**

**ON**  
prints the warning and sends output objects to open destinations.

Aliases  

<table>
<thead>
<tr>
<th>ODS VERIFY</th>
<th>YES</th>
</tr>
</thead>
</table>

**OFF**  
suppresses the warning.

Aliases  

<table>
<thead>
<tr>
<th>ODS NOVERIFY</th>
<th>NO</th>
</tr>
</thead>
</table>

**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 DATA Step 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 903

*variable*

identifies the variable to write.

Example “Example 4: Creating and Using a User-Defined Table Template” on page 89

_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 159.

@ | @@

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 903
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 159.

**Example**

“Example 4: Creating and Using a User-Defined Table Template” on page 89

+*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 89
@ '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 89

**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 Global 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 159.

**See Also**

- Output Delivery System and the DATA Step on page 57
ODS WORD Statement (Preproduction)

Opens, manages, or closes the ODS destination for Word, which generates Microsoft Word output compatible with Microsoft Office 2010 and later versions.

- **Valid in:** Anywhere except SAS Viya
- **Category:** ODS: Third-Party Formatted
- **Default:** The default style is Word.

**Syntax**

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

**Summary of Optional Arguments**

- **(ID-identifier)**
  - Run multiple instances of the same destination at the same time. Each instance can have different options.

- **ANCHOR='anchor-name'**
  - Specify a unique base name for the anchor tag that identifies each output object in the document

- **AUTHOR='text-string'**
  - Specify the author of the Word document

- **BASE='base-text'**
  - Specify the text to use as the first part of all links and references that ODS creates in the document

- **BOX_SIZING=(CONTENT_BOX | BORDER_BOX)**
  - Specify how to measure the width of cells

- **CATEGORY='text-string'**
  - Specify the category of the Word document

- **CLOSE**
  - Close the destination and the file that is associated with it.

- **COMMENTS='text-string'**
  - Add comments to the properties of the Word document

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

- **FILE='file-specification'**
  - Specify the file that contains the Word document 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='integer'**
Specify the image resolution for the graphical output

**KEYWORDS='text-string'**
Add keywords to the Word document properties

**OPTIONS (< suboption(s)>)**
Specify destination-specific suboptions

**SASDATE**
Insert the standard SAS date in the document in place of the default Word 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

**STARTPAGE= YES | NO | NOW**
Control the insertion of new pages

**STATUS='text-string'**
Specify the status of the document

**STYLE=style-definition**
Specify the style template to apply to your output

**TITLE='text-string'**
Specify the title for the document

**WORK='fileref' | 'directory-name'**
Specify an alternate directory for the temporary files

---

**Without Arguments**
The ODS destination for Word and creates Word output in Microsoft Open Office XML format for Office 2010 and later.

---

**Actions**
The following actions are available for the ODS WORD 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 332

**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 793
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 42.

**See**
“ODS SHOW Statement” on page 806

**Optional Arguments**

**ANCHOR='anchor-name'**
specifies a unique base name for the anchor tag that identifies each output object in the document. Each output object has an anchor tag to reference.

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

**Default**
IDX

**Restriction**
Each anchor name in a document must be unique.

**Requirement**
You must enclose anchor-name in quotation marks.

**Tips**
You can change anchor names as often as you want by specifying the ANCHOR= option in an ODS WORD 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 sections of your document. Because you can control where the anchor name changes, you know in advance what the anchor name will be at those points.

**AUTHOR='text-string'**
specifies the author of the Word document. This information can be seen in the document properties.

**BASE='base-text'**
specifies the text to use as the first part of all links and references that ODS creates in the document.

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

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

**Example**
BASE='http://www.your-company.com/local-url/'
In this example, ODS creates links that begin with the string http://www.your-company.com/local-url/.

**BOX_SIZING**(CONTENT_BOX | BORDER_BOX)

specifies how to measure the width of cells. This option overrides the default value of BOX_SIZING for the destination.

BOX_SIZING is defined by the WC3 specification, the CSS3 Module. For more information, refer to the CSS3 Box Model specification at [https://www.w3.org/TR/css-box-3/](https://www.w3.org/TR/css-box-3/).

Default: BORDER_BOX

**CATEGORY**='text-string'

specifies the category of the Word document. This information can be seen in the document properties.

**COMMENTS**='text-string'

adds comments to the properties of the Word document. This information can be seen in the document properties.

**FILE**='file-specification'

specifies the file that contains the Word document 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 Word is “sasword.docx”.

**GFOOTNOTE | NOGFOOTNOTE**

controls the location where footnotes are printed in the graphics output.

**GFOOTNOTE**

prints footnotes that are created by SAS/GRAPH, or the SG Procedures. 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 document. 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 SG Procedures.
**GTITLE | NOGTITLE**

controls the location where titles are printed in the graphics output.

**GTITLE**

prints the title that is created by SAS/GRAPH or the SG Procedures. 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**

This option applies only to SAS programs that produce one or more device-based graphics, or graphics created by the SG Procedures.

Titles that are displayed support most SAS/GRAPH TITLE options. The font must be valid for the document. Options that ODS cannot handle, such as text angle specifications, are ignored.

**See**

For details about the SAS /GRAPH TITLE statement, see “TITLE, FOOTNOTE, and NOTE Statements” in *SAS/GRAPH: Reference*.

**IMAGE_DPI=’integer’**

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 Word document. This information can be seen in the document properties.

**Note:** The KEYWORDS value is 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.
OPTIONS (<suboption(s)>)
specifies destination-specific suboptions with space-delimited name='value' pairs.

suboption(s) are the following:

BODY_TITLE= 'YES' | 'NO' | 'OFF' | 'ON'
specifies that SAS titles and footnotes are placed into the body of the document instead of the header and footer sections of the document.
Default OFF

CANT_SPLIT= 'YES' | 'NO' | 'OFF' | 'ON'
specifies whether all contents of a table row shall be rendered on the same page. If necessary, the start of the current row is moved to the start of a new page.
Default ON

CONTENTS= 'YES' | 'NO' | 'OFF' | 'ON'
generates a table of contents.
Default OFF

FOOTER_Y='text-string'
specifies the distance from the bottom edge of the page to the bottom edge of the footer.
Valid units of measure:
in inches
cm centimeters
mm millimeters
pt a printer’s point
Default pt
Requirement Numerics must be a 0 or a positive number. Negative numerics are not accepted.

HEADER_Y='text-string'
specifies the distance from the top edge of the page to the top edge of the header.
Valid units of measure:
in inches
cm centimeters
mm millimeters
pt a printer’s point
Default pt
Requirement Numerics must be a 0 or a positive number. Negative numerics are not accepted.

KEEP_LINES= 'YES' | 'NO' | 'OFF' | 'ON'
specifies that all lines of the current paragraph should be maintained on a single page whenever possible.
KEEP_NEXT= 'YES' | 'NO' | 'OFF' | 'ON'
specifies that the contents of the current paragraph are at least partly rendered on
the same page as the following paragraph whenever possible.

Default ON

STYLE_PREFIX= 'text-string'
specifies the prefix for Word styles generated from ODS styles.

Default 'sas'

STYLE_VISIBILITY= 'visible' | 'semihidden' | 'hidden'
specifies the visibility for Word styles generated from ODS styles.

Default semihidden

TOC_DATA= 'YES' | 'NO' | 'OFF' | 'ON'
specifies whether to capture the contents data in the document.

Default OFF

TOC_LEVEL= 'integer'
controls the depth of the table of contents in the document. A value of 0 means
that there is no limit to the depth of the table of contents.

Default 0

TOC_TYPE= 'LEGACY' | 'HEADINGS'
specifies whether the ODS destination for Word generates the legacy table of
contents automatically, or the user generates a custom table of contents manually
using nested headings.

Default LEGACY

SASDATE
inserts the standard SAS date in the document in place of the default Word date and
time field. Word does not update the date and time.

STARTPAGE= YES | NO | NOW
controls the insertion of new pages.

YES
inserts a new page when ODS or the procedure requests one.

Alias ON

NO
instructs ODS not to insert a new page, even if the procedure code requests a new
page. A new page is inserted only when a page is filled or when you specify
STARTPAGE= NOW.

Alias OFF

NOW
forces the immediate insertion of a new page. Once STARTPAGE=NOW
executes, the STARTPAGE= option reverts to the previously set STARTPAGE=
argument. For example, if you set STARTPAGE=NO to STARTPAGE=NOW, STARTPAGE= reverts to STARTPAGE=NO.

Default     YES

STATUS='text-string'
specifies the status of the document. This information can be seen in the document properties.

STYLE='style-definition'
specifies the style template to apply to your output.

style-definition
describes how to display the presentation aspects (color, font face, and font size) of your SAS output. A style template determines the overall appearance of the document. Each style template consists of style elements.

Default     Word

TITLE='text-string'
specifies the title for the 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 Word uses the SAS WORK library to hold 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
The ODS destination for Word uses Microsoft Open Office XML Format for Office 2010 and later.

'Portrait' is the default printing orientation. The orientation can be changed to landscape using the ORIENTATION= System Option.

SAS supports Microsoft Word 2010 and later. For more information, see the following resources:

- Microsoft Word Standards Support
- Base SAS Focus Area

ODS Word Style Template
ODS WORD uses the PROC TEMPLATE SCHEME statement to define its style, which is Styles.Word by default. To learn more about the SCHEME statement, see A Second Look at the ODS Destination for PowerPoint

ODS WORD allows the user to control the placement and appearance of dates and page numbers by using the BodyDate, SASDate and PageNo style elements. Users can define behaviors specific to Word with the CONTENT style attribute for the Date and PageNo style elements. To do this, set the CONTENT style attribute to a text string containing one or more Word field codes. For more information, see List of field codes in Word.
See “Example 1: Creating a Page X of Y Header” on page 912 for more information about the PAGE and NUMPAGE field codes.

**Supported File Types**

The following table provides descriptions of the supported file types for the ODS destination for Word.

*Table 6.31 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>GIF (Graphics Interchange Format)</td>
<td>Supports only color-mapped images. GIF is owned by CompuServe, Inc.</td>
</tr>
<tr>
<td>JPEG (Joint Photographic Experts Group)</td>
<td>A file format that is used for storing noninteractive images.</td>
</tr>
<tr>
<td>PNG (Portable Network Graphic)</td>
<td>Supports true color, gray-scale, and 8-bit images.</td>
</tr>
<tr>
<td>TIFF (Tagged Image File Format)</td>
<td>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).</td>
</tr>
</tbody>
</table>

**Examples**

**Example 1: Creating a Page X of Y Header**

**Features:**

- ODS WORD Statement options
  - PROC TEMPLATE
  - PageNo style element
  - CONTENT= style attribute

**Note:**

See List of field codes in Word for more information about field codes in Microsoft Word.

**Details**

The following example creates a Word document with a customized header. Instead of the default page number, you get Page X of Y behavior.
Program

proc template;
    define style styles.mystyle;
    parent = styles.word;
    class pageno /
        content = "Page {PAGE} of {NUMPAGES}";
    end;
run;
ods graphics on;
ods word file="pagexofy.docx" title="Page X of Y test" style=mystyle;
proc freq data=sashelp.class;
    tables age*sex/plots=freq;
run;
ods word close;

Program Description

Create a style for the ODS destination for Word. Control the appearance of the page numbers using the PageNo style element and the CONTENT= style attribute containing the Microsoft Word field codes {PAGE} and {NUMPAGES}.

proc template;
    define style styles.mystyle;
    parent = styles.word;
    class pageno /
        content = "Page {PAGE} of {NUMPAGES}";
    end;
run;

Enable ODS Graphics.

ods graphics on;

Specify the ODS destination for Word and the style, mystyle, that was defined earlier in the program.

ods word file="pagexofy.docx" title="Page X of Y test" style=mystyle;

Create a crosstabulation frequency table.

proc freq data=sashelp.class;
    tables age*sex/plots=freq;
run;

Close the ODS destination for Word.

ods word close;
### Example 2: Customizing the Table of Contents

**Features:**
- TOC_LEVEL=
- TOC_DATA=
- CONTENTS=

**Other features:**
- ORIENTATION= system option

**Details**

The following example creates a Word document with a table that generates a table of contents.

**Program**

```sas
ods graphics on;
data plants;
  input type $ @;
```
do block=1 to 3;
    input stemleng @;
    output;
end;
cards;
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;
ods word file="toc_level_landscape.docx"
options(contents='on' toc_data='on' toc_level='2')
title="toc_level = '2'";
title "toc_level = '2'";
options orientation=landscape;
proc glm order=data data = plants;
class type block;
model stemleng=type block / solution;
means type / waller regwq;
*type-order------------clrn-cltn-knox-onel-cpst-wbsh-wstr;
contrast 'compost vs others' type -1 -1 -1 -1 6 -1 -1;
contrast 'river soils vs.non' type -1 -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;
run;
quit;
options orientation=portrait;
proc glm order=data data = plants;
class type block;
model stemleng=type block / solution;
means type / waller regwq;
*type-order------------clrn-cltn-knox-onel-cpst-wbsh-wstr;
contrast 'compost vs others' type -1 -1 -1 -1 6 -1 -1;
contrast 'river soils vs.non' type -1 -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;
run;
quit;

Program Description
Turn on ODS Graphics.
ods graphics on;

Read in data.
data plants;
  input type $ @;
  do block=1 to 3;
    input stemleng @;
    output;
  end;
cards;
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;

Open the ODS destination for Word.
ods word file="toc_level_landscape.docx"

Specify the TOC_LEVEL option. The table of contents does not display by default. From the table of contents page, use Ctrl+A to select all content in the Word document, and then press F9 to display the table of contents.
options(contents='on' toc_data='on' toc_level='2')
title="toc_level = '2'";

Invoke the GLM procedure and change page orientation for the output to landscape. The default orientation is portrait.
title "toc_level = '2'";
options orientation=landscape;
proc glm order=data data = plants;
class type block;
model stemleng=type block / solution;
means type / waller regwq;
* -type-order------------clrn-cltn-knox-onel-cpst-wbsh-wstr;
contrast 'compost vs others' type -1 -1 -1 -1  6 -1 -1;
contrast 'river soils vs.non' type -1 -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;
run;
quit;
Invoke the GLM procedure again and change the page orientation for the output back to portrait.

```sas
options orientation=portrait;
proc glm order=data data = plants;
   class type block;
   model stemleng=type block / solution;
   means type / waller regwq;
* -type-order------------clrn-cltn-knox-onel-cpst-wbsh-wstr;
contrast 'compost vs others' type -1 -1 -1 -1 6 -1 -1;
contrast 'river soils vs non' type -1 -1 -1 -1 0 5 -1,
   type 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 1;
contrast 'knox vs oneill' type 0 0 1 -1 0 0 0;
run;
quit;
```

Close the ODS destination for Word.
```
ods word close;
```

**Word Output**

This output is created using the ODS destination for Word. It uses the table of contents options to generate a table of contents for the GLM output. There are 18 pages of output associated with this program. Only excerpts of the output are shown here.
Table of Contents

The GLM Procedure............................................................................................................1
Data......................................................................................................................................1
Analysis of Variance...........................................................................................................2
The GLM Procedure...........................................................................................................4
Data......................................................................................................................................4
Analysis of Variance...........................................................................................................5
Example 2: Customizing the Table of Contents

The GLM Procedure

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>8</td>
<td>142.1885714</td>
<td>17.755714</td>
<td>10.60</td>
<td>0.0003</td>
</tr>
<tr>
<td>Error</td>
<td>12</td>
<td>19.74285714</td>
<td>1.64528571</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>20</td>
<td>161.9354286</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R-Square</th>
<th>Coeff Var</th>
<th>Root MSE</th>
<th>stemlen Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.878079</td>
<td>3.939745</td>
<td>1.282669</td>
<td>32.55714</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Type I SS</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
</table>
Part 6

Arranging Tabular and Graphic Information

Chapter 7

Arranging Output with ODS LAYOUT ........................................... 923
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 925
- “ODS REGION Statement, Absolute” on page 936
- “ODS LAYOUT END Statement” on page 956

See “Example: Absolute Layout Using the Region Container” on page 930 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 940
- “ODS REGION Statement, Gridded” on page 951
- “ODS LAYOUT END Statement” on page 956

See “Example: Gridded Layout Titles in a Region” on page 949 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 except SAS Viya
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 936.

Example:
```plaintext
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

- DESCRIPTION="text-string"
  Specify alternate text for your layout containers and regions.
- 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

DESCRIPTION="text-string"
specifies alternate text for layout containers and regions. This option helps to make reports accessible for people with a wide range of abilities. For accessibility, the text should convey the meaning of the container or region. The alternate text will be read by assistive technology such as a screen reader.

Valid in: HTML5 and PDF
Alias: DESC
HEIGHT=\textit{dimension}

specifies the vertical height 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:

\begin{itemize}
  \item cm centimeters
  \item em standard typesetting measurement unit for width
  \item ex standard typesetting measurement unit for height
  \item in inches
  \item mm millimeters
  \item pct a percentage. You can also use the ‘\%’ symbol.
  \item pt a printer’s point
  \item px pixels
\end{itemize}

Default

If omitted, the height of the layout defaults to the maximum vertical space needed to display all of the regions.

Example

\begin{verbatim}
ods layout absolute height=5in;
  proc print data=sashelp.class;
  run;
ods layout end;
\end{verbatim}

STYLE=\texttt{<style-element-name> \texttt{<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

\begin{verbatim}
ods layout absolute style=[backgroundcolor=yellow];
  proc print data=sashelp.class;run;
ods layout end;
\end{verbatim}

WIDTH=\textit{dimension}

specifies the horizontal width 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:

\begin{itemize}
  \item cm centimeters
  \item em standard typesetting measurement unit for width
  \item ex standard typesetting measurement unit for height
  \item in inches
  \item mm millimeters
  \item pct a percentage. You can also use the ‘\%’ symbol.
\end{itemize}


<table>
<thead>
<tr>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pt</strong></td>
<td>a printer’s point</td>
</tr>
<tr>
<td><strong>px</strong></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><strong>cm</strong></td>
<td>centimeters</td>
</tr>
<tr>
<td><strong>em</strong></td>
<td>standard typesetting measurement unit for width</td>
</tr>
<tr>
<td><strong>ex</strong></td>
<td>standard typesetting measurement unit for height</td>
</tr>
<tr>
<td><strong>in</strong></td>
<td>inches</td>
</tr>
<tr>
<td><strong>mm</strong></td>
<td>millimeters</td>
</tr>
<tr>
<td><strong>pct</strong></td>
<td>a percentage. You can also use the ‘%’ symbol.</td>
</tr>
<tr>
<td><strong>pt</strong></td>
<td>a printer’s point</td>
</tr>
<tr>
<td><strong>px</strong></td>
<td>pixels</td>
</tr>
</tbody>
</table>

**Default**

0

**Example**

```
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:

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cm</strong></td>
<td>centimeters</td>
</tr>
<tr>
<td><strong>em</strong></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 Y argument defaults to the current vertical position on the page.

Example
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 LAYOUTABSOLUTE 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**

```r
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.

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

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

Products are organized in a hierarchy consisting of three levels:

- Product Line
- Product Category
- Product Group

Where we generate our profit...

Our Mission

To deliver the best quality sporting equipment, accessories, and outdoor equipment for all seasons at the
most affordable prices.)*;

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.

\begin{verbatim}
ods pdf file="OrionstarCoInfo.pdf" notoc nogtitle nogfootnote;
\end{verbatim}

Set the layout to absolute layout.

\begin{verbatim}
ods layout absolute;
\end{verbatim}

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.

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

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.

\begin{verbatim}
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.\};
\end{verbatim}

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.

\begin{verbatim}
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.\};
\end{verbatim}

Create the fourth region, “What we sell”, using ODS TEXT. Add the star image and the text for the second item in the list.

\begin{verbatim}
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...}\};
\end{verbatim}

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.

\begin{verbatim}
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\};
\end{verbatim}
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]Our Vision: To achieve the highest profits
   with the best products and customer satisfaction.}";
```
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 [font_style=italic vjust=center font_size=10pt just=center]To transform the way the world purchases sporting and outdoor equipment.}";

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 [font_style=italic vjust=center font_size=10pt just=center]To stay Customer focused, Swift and Agile, Innovative, and Trustworthy.}";

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

See Also

ODS Statements
- “ODS REGION Statement, Absolute” on page 936
- “ODS LAYOUT END Statement” on page 956

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.

Valid in:  Anywhere except SAS Viya
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).

Note: This ODS REGION statement is inserted 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.

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

ODS REGION < option-1>< option-2 ...>

Summary of Optional Arguments

DESCRIPTION="text-string"
Specify alternate text for your layout containers and regions.

HEIGHT=dimension
Specify a vertical height of the region.

STYLE=<style-element-name> <[style-attribute-specification(s)]>
Specify 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

DESCRIPTION="text-string"
specifies alternate text for layout containers and regions. This option helps to make reports accessible for people with a wide range of abilities. For accessibility, the text should convey the meaning of the container or region. The alternate text will be read by assistive technology such as a screen reader.

Valid in HTML5 and PDF

Alias DESC

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:

<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 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**

```plaintext
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**

```plaintext
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:

<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>
</tbody>
</table>
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=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 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**

```sas
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 925
- “ODS LAYOUT END Statement” on page 956

**ODS LAYOUT GRIDDED Statement**

Enables you to arrange output dynamically in a two-dimensional grid structure.

**Valid in:** Anywhere except SAS Viya

**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.
- **DESCRIPTION=** "*text-string*"
  Specify alternate text for your layout containers and regions.
- **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)]>
  Specify 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:

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
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
ods layout gridded columns=2 column_widths=(2in 2in);
DESCRIPTION="text-string"

specifies alternate text for layout containers and regions. This option helps to make reports accessible for people with a wide range of abilities. For accessibility, the text should convey the meaning of the container or region. The alternate text will be read by assistive technology such as a screen reader.

Valid in HTML5 and PDF

Alias DESC

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 gridded height=7in;
proc print data=sashelp.class; run
ods layout end;

ORDER_TYPE= <ROW_MAJOR | COLUMN_MAJOR>

populates the grid by rows or columns.

COLUMN_MAJOR

specifies that the gridded layout first populates all regions in the first column before moving on to the next column.

ROW_MAJOR

specifies that the gridded layout first populates all regions in the row before moving on to the next row.

Default ROW_MAJOR

Restrictions The ORDER_TYPE= option is valid only in PRINTER destinations.

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.
Examples

This code populates row1 column1, row1 column2, row2 column1, and row2 column2:

```plaintext
ods layout gridded columns=2 rows=2 order_type=row_major;
```

This example populates row1 column1, row2 column1, row1 column2, and row2 column2:

```plaintext
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**

```plaintext
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**

```plaintext
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.

```plaintext
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**

```plaintext
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=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;
```

**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.

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ods layout gridded y=10cm;</td>
</tr>
<tr>
<td>proc print data=sashelp.class; run</td>
</tr>
<tr>
<td>ods layout end;</td>
</tr>
</tbody>
</table>

**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.

```ods
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
don layout gridded;
don 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.

```ods
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.

```ods
layout end;
don pdf close;
```

The following output shows how tables, titles, footnotes, and charts are placed on a page using gridded layout:
**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.

**Valid in:** Anywhere except SAS Viya

---

**See Also**

**ODS Statements**
- “ODS REGION Statement, Gridded” on page 951
- “ODS LAYOUT END Statement” on page 956
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 the ODS destination for PowerPoint.

Note: This ODS REGION statement is inserted between the ODS LAYOUT GRIDDED statement and the ODS LAYOUT END statement. The ODS LAYOUT GRIDDED statement manages the ODS LAYOUT destination.

Tip: Regions can have a fixed size or can be dynamically sized.

Syntax

ODS REGION (<option-1><option-2 ...>);

Summary of Optional Arguments

COLUMNSPAN=number
Specify the number of grid columns that the region occupies.

COLUMNS=number
Specify the current grid column position in the gridded layout.

DESCRIPTION="text-string"
Specify alternate text for your layout containers and regions.

HEIGHT=dimension
Specify the vertical height of the region.

ROWSPAN=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)]>
Specify 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.

    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.
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.
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;

DESCRIPTION="text-string"
specifies alternate text for layout containers and regions. This option helps to make reports accessible for people with a wide range of abilities. For accessibility, the text should convey the meaning of the container or region. The alternate text will be read by assistive technology such as a screen reader.

Valid in HTML5 and PDF

Alias DESC

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.

ROWSPAN=number

specifies the number of grid rows that the region occupies. The ROWSPAN= option enables you to combine adjacent grid rows in gridded layout.
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.
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:

<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 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 940
- “ODS LAYOUT END Statement” on page 956

ODS LAYOUT END Statement
Ends a gridded or absolute layout statement block.

Valid in: Anywhere except SAS Viya
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 23

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 23.

**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.

```plaintext
options odsgraphics=off odsdest=listing odsstyle=default;
```

**See Also**

“Working with Output Defaults” on page 23

---

**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 975.

To create your own styles or to modify styles, see “Overview: ODS Style Templates” in SAS Output Delivery System: Procedures Guide.

**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.

```
options odsstyle=Sapphire;
odc html close;
odc html;
```

See Also

“Working with Output Defaults” on page 23
Part 8

ODS Styles Reference

Chapter 9
  Overview ................................................................. 965

Chapter 10
  Style Templates ..................................................... 975

Chapter 11
  Style Elements ..................................................... 1045

Chapter 12
  Style Attributes .................................................... 1073
Chapter 9
Overview

Understanding Styles, Style Elements, and Style Attributes .......................... 965
Using Styles with Base SAS Procedures ......................................................... 971

Understanding Styles, Style Elements, and Style Attributes

The appearance of SAS output is controlled by ODS style templates (ODS styles). ODS styles are produced from compiled STYLE templates written in PROC TEMPLATE style syntax. An ODS style template is a collection of style elements that provides specific visual attributes for your SAS output.

• 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 using the LIST statement in PROC TEMPLATE:

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

For complete information about viewing ODS styles, see “Viewing ODS Styles Supplied by SAS” on page 975.
By default, HTML 4 output uses the HTMLBlue style template and HTML 5 output uses the HTMLEncore style template. To help you become familiar with styles, style elements, and style attributes, look at the relationship between them.

You can use the `SOURCE` statement in PROC TEMPLATE to display the structure of a style template. The following code prints the structure of the HTMLBlue style template to the SAS log:

```sas
proc template;
   source styles.HTMLBlue;
run;
```

The following figure illustrates the structure of a style. The figure shows the relationship between the style, the style elements, and the style attributes.
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.
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 975.

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 1045.

3 BORDERCOLOR=, BACKGROUNDCOLOR=, 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 1073.

Style attributes can be referenced with style references. See “style-reference” on page 1136 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” on page 1074.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>PROC REPORT STATEMENT REPORT Area</th>
<th>PROC REPORT Areas: CALLDEF, COLUMN, HEADER, LINES, SUMMARY</th>
<th>PROC TABULATE STATEMENT TABLE</th>
<th>PROC TABULATE STATEMENTS VAR, CLASS, BOX, CLASSLEVEL, KEYWORD</th>
<th>PROC PRINT 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>
<td>X</td>
</tr>
<tr>
<td>BACKGROUNDCOLOR=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>BACKGROUNDIMAGE=</td>
<td>X</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>
<td></td>
</tr>
<tr>
<td>BORDERBOTTOMSTYLE=</td>
<td>X</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: all locations other than TABLE</td>
<td></td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------</td>
<td>----------------------------------------------------------</td>
<td>--------------------------------</td>
<td>----------------------------------------------------------</td>
<td>------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>BORDERBOTTOMWIDTH</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERLEFTCOLOR</td>
<td>X</td>
<td>X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERLEFTSTYLE</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERLEFTWIDTH</td>
<td>X</td>
<td>X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERCOLOR</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERCOLORDARK</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERCOLORLIGHT</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERRIGHTCOLOR</td>
<td>X</td>
<td>X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERRIGHTSTYLE</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERRIGHTWIDTH</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORERTOPCOLOR</td>
<td>X</td>
<td>X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORERTOPSTYLE</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORERTOPWIDTH</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BORDERWIDTH</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CELLPADDING</td>
<td>X</td>
<td>X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CELLPACING</td>
<td>X</td>
<td>X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CELLWIDTH</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLASS</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLOR</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLYOVER</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FONT</td>
<td>X</td>
<td>X X X X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FONTFAMILY</td>
<td>X</td>
<td>X X X X</td>
<td></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>
<td>PROC PRINT: all locations other than TABLE</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------</td>
<td>------------------------------------------------------------</td>
<td>-------------------------------</td>
<td>------------------------------------------------------------</td>
<td>---------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>FONTSIZE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FONTCOLOR=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FONTCOLOR=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FONTCOLOR=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FONTCOLOR=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>FRAME=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HEIGHT=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HREFTARGET=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>HTMLSTYLE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>NOBREAKSPACE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>OUTPUTWIDTH=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>POSTHTML=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>POSTIMAGE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>POSTTEXT=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PREHTML=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PREIMAGE=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PRETEXT=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PROTECTSPECIALCHARS=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RULES=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>TAGATTR=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>TEXTALIGN=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>URL=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>VERTICALALIGN=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>WIDTH=</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

* When you use these attributes in this location, they affect only the text that is specified with the PRETEXT=, POSTTEXT=, PREHTML=, and POSTHTML= attributes. To alter the foreground color or the font for the text that appears in the table, you must set
For complete documentation about style attributes and their values, see Chapter 12, “Style Attributes,” on page 1073.

** To help prevent unexpected wrapping of long text strings when using PROC REPORT with the ODS RTF destination, set NOBREAKSPACE=OFF in a location that affects the LINE statement. The NOBREAKSPACE=OFF attribute must be set in the PROC REPORT code either on the LINE statement or on the PROC REPORT statement where style(lines) is specified.

---

### 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:

```plaintext
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. For the complete input data set, see “EXPREV” in *Base SAS
Procedures Guide*.

```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;
```
### Chapter 9 • Overview

#### Output 9.3  Enhanced PROC PRINT Output

**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></td>
<td><strong>170</strong></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></td>
<td><strong>47</strong></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>
Chapter 10

Style Templates

Viewing ODS Styles Supplied by SAS .................................................. 975
Program for Viewing Multiple Styles .................................................. 977
Table of Suggested ODS Styles ........................................................... 978
ODS Styles Gallery ............................................................................. 981
  EPUB Daisy Style .......................................................... 981
  HTML Styles ............................................................. 983
  Printer Styles ............................................................ 1037
  Styles for the ODS Destination for PowerPoint ......................... 1042
  Style for the ODS Destination for Excel .................................. 1043

Viewing ODS Styles Supplied by SAS

Over fifty ODS styles are available for use with ODS. 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;
   path sashelp.tmplmst;
   list styles;
run;
```
To view the source code of a specified style, submit the following code:

```sas
proc template;
   source styles.style-name;
run;
```
Log 10.1  Partial Source Code for the HTMLBlue Style

```
9    proc template;
10      source styles.htmlBlue;
define style Styles.htmlBlue;
  parent = styles.statistical;
    class GraphColors /
      'gndata12' = cxECE8C4
      'gndata11' = cxD8D8F8
      'gndata10' = cxC6E4BF
      'gndata9' = cxE6CEAD
      'gndata8' = cxE5C1D4
      'gndata7' = cxCCDFF0
      'gndata6' = cxDDDEB5
      'gndata5' = cxDBC7E7
      'gndata4' = cxD5C6B4
      'gndata3' = cxB7D4D3
      'gndata2' = cxB7B3B4
      'gndata' = cxBB22DC
      '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.

---

**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 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 979.

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 links;
    define libname ;
      blank_dups=on;
    end;
    define links;
      header = 'Samples';
      compute as '<a href="'||trim(style)||'.html">HTML</a> ' ||
        '<a href="'||trim(style)||'.pdf">PDF</a> ' ||
        '<a href="'||trim(style)||'.rtf">RTF</a>';
    end;
  end;
run;
```

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.vstyle(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";
%mend;

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.

ods noresults;

This DATA step creates a sample of each style.

data _null_;  
    set sashelp.vstyle(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.

ods results;
ods preferences;

---

**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>
<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>
### Destination

<table>
<thead>
<tr>
<th>Destination</th>
<th>Recommended Styles</th>
<th>Default Style</th>
</tr>
</thead>
</table>

**Note:** Starting with SAS 9.4M5, the BlockPrint ODS style is deprecated. Existing SAS programs that use the BlockPrint style will still function, but the BlockPrint style no longer appears in the list of available styles.

* The Moonflower style for ODS EPUB is designed for nighttime or low-light reading.
ODS Styles Gallery

**EPUB Daisy Style**

*Output 10.1  EPUB Book Title Page*
**HTML Styles**

You can view and modify the default HTML style by selecting **Tools ⇒ Options ⇒ 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 ⇒ 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: 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>$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>$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 10.6  Analysis Style: Graphs
Output 10.7  BarrettsBlue 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>$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>$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 10.8  BarrettBlue Style: Graphs

![BarrettBlue Style: Graphs](image)

![BarrettBlue Style: Graphs](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>$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>$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 10.10  BlockPrint Style: Graphs
The following style is recommended for generating accessible output when used with the ODS HTML5 destination.

**Output 10.11**  *Daisy 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>$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>$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>
The following style is recommended for generating accessible output when used with the ODS HTML5 destination.

**Output 10.12  Daisy Style: Graphs**
### Output 10.13  Default 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>$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>$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 10.14 Default Style: Graphs
### Output 10.15  Dove 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>$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>$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 10.16  Dove Style: Graphs

[Bar charts showing sales for different regions and items (BED, CHAIR, DESK, SOFA, TABLE)]

[Scatter plot with linear regression line and confidence limits for height and weight data]
Output 10.17  Dtree 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>$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>$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 10.18  Dtree Style: Graphs
### Output 10.19  EGDefault 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>$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>$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  EGDefault Style: Graphs

![Graphs showing sales data for different regions and items.](image)

- **Bed:**
  - **EAST:** Predicted Sales: $10,000, Actual Sales: $11,000
  - **WEST:** Predicted Sales: $12,000, Actual Sales: $13,000

- **Chair:**
  - **EAST:** Predicted Sales: $9,000, Actual Sales: $10,000
  - **WEST:** Predicted Sales: $11,000, Actual Sales: $12,000

- **Desk:**
  - **EAST:** Predicted Sales: $8,000, Actual Sales: $9,000
  - **WEST:** Predicted Sales: $10,000, Actual Sales: $11,000

- **Sofa:**
  - **EAST:** Predicted Sales: $15,000, Actual Sales: $16,000
  - **WEST:** Predicted Sales: $17,000, Actual Sales: $18,000

- **Table:**
  - **EAST:** Predicted Sales: $13,000, Actual Sales: $14,000
  - **WEST:** Predicted Sales: $15,000, Actual Sales: $16,000

![Scatter plot showing relationship between height and weight.](image)

- **95% Confidence Limits:**
- **95% Prediction Limits:**
- **Regression Line:**
Output 10.21  Festival 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>$999.00</td>
<td>$297.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$808.00</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>$856.00</td>
<td>$846.00</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>$812.00</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 10.22  Festival Style: Graphs

- **Graph 1:** Bar charts showing sales of BED, CHAIR, and DESK in the EAST and WEST regions. Each chart compares predicted sales to actual sales.

- **Graph 2:** Scatter plot showing the relationship between weight and height with 95% confidence limits, 95% prediction limits, and regression line.
Output 10.23  Gantt 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>$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>$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 10.24  Gantt Style: Graphs
Output 10.25  Harvest 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>$999.00</td>
<td>$297.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$808.00</td>
<td>$848.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$842.00</td>
<td>$533.00</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>$948.00</td>
<td>$488.00</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 10.26  Harvest Style: Graphs
The following style is suitable for generating accessible output when used with the ODS HTML5 destination.

*Output 10.27  HighContrast 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>$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>$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>
The following style is suitable for generating accessible output when used with the ODS HTML5 destination.

**Output 10.28**  HighContrast Style: Graphs
HTMLBlue is the default style for the HTML output in the SAS Windowing Environment and SAS Studio.

**Output 10.29  HTMLBlue 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>$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>$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>
In SAS 9.4M5, the HTMLEncore style uses the new Avenir Next for SAS fonts. For more information, see “TrueType Fonts Supplied by SAS” in SAS Language Reference: Concepts.

**Output 10.31  HTMLEncore: 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>$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>$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 10.32  HTMLEncore: Graphs 1
<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>$808.00</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>$812.00</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 10.34  Journal Style: Graphs

---

**First Graph: Bed, Chair, Desk**

- **X-axis:** Region (East, West, East, West, East, West)
- **Y-axis:** Sales ($0 to $20,000)
- **Legend:**
  - Squares: Predicted Sales
  - Lines: Actual Sales

---

**Second Graph: Sofa, Table**

- **X-axis:** Region (East, West, East, West)
- **Y-axis:** Sales ($0 to $20,000)
- **Legend:**
  - Squares: Predicted Sales
  - Lines: Actual Sales

---

**Regression Graph**

- **X-axis:** Height (50 to 70)
- **Y-axis:** Weight (25 to 150)
- **Legend:**
  - 95% Confidence Limits
  - 95% Prediction Limits
  - Regression Line
### Output 10.35  Meadow 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>$999.00</td>
<td>$297.00</td>
</tr>
<tr>
<td>3</td>
<td>SOFA</td>
<td>EAST</td>
<td>$808.00</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>$846.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$948.00</td>
<td>$488.00</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>$564.00</td>
</tr>
</tbody>
</table>
Output 10.36  Meadow Style: Graphs
<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>$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 10.38  Minimal Style: Graphs

![Bar charts showing sales data by region and predicted vs. actual sales.](image1)

![Scatter plot showing relationship between height and weight with confidence and prediction intervals.](image2)
<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>$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 10.40  
Netdraw and NoFontDefaul Styles: Graphs

- **Bar Graphs:**
  - **Sofa and Table:**
    - **Regions:** East, West
    - **Y-axis:** Sales
    - **X-axis:** Region (East, West)
    - **Legend:** Predicted Sales, Actual Sales

- **Scatter Plot:**
  - **X-axis:** Height
  - **Y-axis:** Weight
  - **Lines and Shaded Areas:**
    - 95% Confidence Limits
    - 95% Prediction Limits
    - Regression Line

---

Chapter 10 • Style Templates
<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>$846.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$842.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$856.00</td>
<td>$846.00</td>
</tr>
<tr>
<td>6</td>
<td>SOFA</td>
<td>EAST</td>
<td>$948.00</td>
<td>$488.00</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>$564.00</td>
</tr>
</tbody>
</table>
Output 10.42  Normal Style: Graphs

![Graphs showing sales for different regions for bed, chair, and desk categories.](Image)

![Graph showing relationship between weight and height with confidence intervals and regression line.](Image)
### Output 10.43  Ocean 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>$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>$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 10.44  Ocean Style: Graphs

[Bar charts showing sales for different regions (East and West) for different items (Bed, Chair, Desk, Sofa, Table).]

[Scatter plot showing weight vs. height with 95% confidence limits and prediction limits, along with the 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>$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>$808.00</td>
<td>$846.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$842.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$856.00</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>$812.00</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 10.46  Plateau Style: Graphs

**Graph 1:**
- **Title:** Sales by Region
- **Axes:**
  - **X-axis:** Region (EAST, WEST, SOFA, TABLE, DESK)
  - **Y-axis:** Sales ($0, $5,000, $10,000, $15,000, $20,000)
- **Legend:**
  - Predicted Sales
  - Actual Sales

**Graph 2:**
- **Title:** Weight vs. Height
- **Axes:**
  - **X-axis:** Height (50, 55, 60, 65, 70)
  - **Y-axis:** Weight (25, 50, 75, 100, 125, 150)
- **Legend:**
  - 95% Confidence Limits
  - 95% Prediction Limits
  - 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>$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>$808.00</td>
<td>$846.00</td>
</tr>
<tr>
<td>4</td>
<td>SOFA</td>
<td>EAST</td>
<td>$842.00</td>
<td>$533.00</td>
</tr>
<tr>
<td>5</td>
<td>SOFA</td>
<td>EAST</td>
<td>$856.00</td>
<td>$846.00</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>$812.00</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 10.49  SasWeb 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>$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>$846.00</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 10.50  SasWeb Style: Graphs
### Output 10.51  Seaside 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>$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>$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>$114.00</td>
<td>$564.00</td>
</tr>
</tbody>
</table>
Output 10.52 Seaside Style: Graphs
<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>$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 10.54 StatDoc Style: Graphs

Graph 1: Comparison of predicted and actual sales for different regions.

Graph 2: Scatter plot showing the relationship between height and weight with 95% confidence and prediction limits.

Legend:
- Predicted Sales
- Actual Sales
- 95% Confidence Limits
- 95% Prediction Limits
- Regression Line
Output 10.55  Statistical 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>$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>$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 10.56  Statistical Style: Graphs

[Bar graphs showing sales for different regions (East and West) for Bed, Chair, Desk, Sofa, and Table, with predicted and actual sales marked.]

[Scatter plot showing weight vs. height with 95% confidence and prediction limits, and a regression line.]
### Printer Styles

#### Output 10.57  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>A41.8T 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>$31,988</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>$35,962</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 10.58  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>A41.8T 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>$31,988</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>$35,962</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 10.59 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>$28,846</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 10.60 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>$28,846</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 10.61  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>$31,761</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>$38,846</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 10.62  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>$31,761</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>$38,846</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>
Pearl is the default style for PRINTER output.

**Output 10.64  Pearl Style**
Output 10.65  Sapphire Style

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</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>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>A41.8T convertible</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>$30,299</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 10.66  SasDocPrinter Style

SasDocPrinterStyle

<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</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>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>A41.8T convertible</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>$30,299</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 10.67  Seaside Printer Style

Styles for the ODS Destination for PowerPoint

PowerPointLight is the default style for output created by the ODS destination for PowerPoint.

Output 10.68  PowerPointLight Style
Output 10.69  PowerPointDark Style

Style for the ODS Destination for Excel

Output 10.70  Excel Style
Chapter 11
Style Elements

ODS Style Elements

Overview

The following tables lists all the style elements available for ODS style definitions. The tables provides a brief description of each style element and indicates the style elements from which it inherits its attributes.

Abstract style elements are not explicitly used in the ODS output. They are used for inheritance purposes only. Because of this, abstract styles do not appear in the output of destinations that generate a style sheet.
Style Elements Affecting Documents

Table 11.1  Style Elements Affecting Documents

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
</table>
| Document      | Controls the various document bodies. This generally includes things like the page background color and page margins.  
*Note:* This is an abstract style element. | Container |
| Body          | Controls the Body file. The Body file is created by the BODY= option in any ODS Markup Family Destination. | Document |
| Frame         | Controls the Frame file for HTML. The Frame file is created by the FRAME= option in any ODS Markup Family Destination. | Document |
| Contents      | Controls the Contents file. The Contents file is created by the CONTENTS= option in any ODS Markup Family Destination. | Document |
| Pages         | Controls the Page file. The Pages file is created by the PAGE= option in any ODS Markup Family Destination. | Document |

Style Elements Affecting Tables of Contents

Table 11.2  Style Elements Affecting Tables of Contents and Tables of Pages

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>Controls the Contents file</td>
<td>Document</td>
</tr>
</tbody>
</table>
| indexItem     | Controls list items and folders for Contents and Pages  
*Note:* This is an abstract style element. | Container |
<p>| ContentFolder | Controls the folders in the Contents file | IndexItem |</p>
<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByContentFolder</td>
<td>Controls the bygroup 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>Index</td>
<td>Controls miscellaneous Contents and Pages components</td>
<td>Container</td>
</tr>
<tr>
<td></td>
<td><em>Note:</em> This is an abstract style element.</td>
<td></td>
</tr>
<tr>
<td>IndexProcName</td>
<td>Controls the PROC name in the Contents and Pages files</td>
<td>Index</td>
</tr>
<tr>
<td></td>
<td><em>Note:</em> This is an abstract style element.</td>
<td></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>IndexAction</td>
<td>Determines what happens on mouse-over events for folders and items (HTML only)</td>
<td>IndexItem</td>
</tr>
<tr>
<td>FolderAction</td>
<td>Determines what happens on mouse-over events for folders (HTML only)</td>
<td>IndexAction</td>
</tr>
<tr>
<td>IndexTitle</td>
<td>Controls the title of Contents and Pages files</td>
<td>Index</td>
</tr>
<tr>
<td></td>
<td><em>Note:</em> This is an abstract style element.</td>
<td></td>
</tr>
<tr>
<td>ContentTitle</td>
<td>Controls the title of the Contents file.</td>
<td>IndexTitle</td>
</tr>
</tbody>
</table>

**Style Elements Affecting Pages**

The PAGE= option in applicable MARKUP family destinations creates the page file. See PAGE= option on page 529.
Table 11.3  Style Elements Affecting Pages

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pages</td>
<td>Controls the Page file. The Pages file is created by the PAGE= option.</td>
<td>Document</td>
</tr>
<tr>
<td>Date</td>
<td>Controls how date fields look</td>
<td>Container</td>
</tr>
<tr>
<td></td>
<td>Note: This is an abstract style element.</td>
<td></td>
</tr>
<tr>
<td>PagesDate</td>
<td>Controls the date in the Pages file</td>
<td>Date</td>
</tr>
<tr>
<td>PagesTitle</td>
<td>Controls the title of the Pages file</td>
<td>IndexTitle</td>
</tr>
<tr>
<td>PagesItem</td>
<td>Controls the leafnode item in the Pages file</td>
<td>IndexItem</td>
</tr>
<tr>
<td>PagesProcName</td>
<td>Controls the proc name in the Pages file</td>
<td>IndexProcName</td>
</tr>
<tr>
<td>PagesProcLabel</td>
<td>Controls the proc label in the Pages file</td>
<td>PagesProcName</td>
</tr>
</tbody>
</table>

Style Elements Affecting Frames

The FRAME= option in ODS MARKUP destinations creates the frame file. See FRAME= option on page 523.

Table 11.4  Style Elements Affecting Frame

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inheritance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame</td>
<td>Controls the Frame file for HTML</td>
<td>Document</td>
</tr>
</tbody>
</table>

Style Elements Affecting the Body File

The BODY style element inherits from the DOCUMENT style element. The BODY= option in applicable Markup Family destinations creates the body file. See BODY= option on page 514.

The following style elements affect the body of an HTML document:

- Tables on page 1049
- Text on page 1051
- Layouts on page 1054
### Style Elements Affecting Tables

#### Table 11.5  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>Data</td>
<td>Default style for data cells</td>
<td>Cell</td>
</tr>
<tr>
<td>Cell</td>
<td>Controls data, header, and footer cells</td>
<td>Container</td>
</tr>
<tr>
<td></td>
<td><em>Note:</em> This is an abstract style element.</td>
<td></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 empty 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>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>

#### Table 11.6  Style Elements Affecting Header Cells

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<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>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>Style Element</td>
<td>Description</td>
<td>Inherits from</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------</td>
<td>----------------------</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>HeaderStrong</td>
</tr>
<tr>
<td>HeaderEmpty</td>
<td>Controls empty 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>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>
</tbody>
</table>

**Table 11.7**  *Style Elements Affecting Footer Cells*

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inherits from</th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>
### Style Elements Affecting Text

#### Table 11.8   Elements affecting Text

<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 Body file</td>
<td>ContentsDate</td>
</tr>
<tr>
<td>SASDate</td>
<td>Controls the date field in the Body file when SASDATE option is in effect. This Element is valid only in the ODS destination for PowerPoint.</td>
<td>BodyDate</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>PageNo</td>
<td>Controls page numbers for paginated destinations (RTF, TAGSETS.RTF, Printer, PowerPoint)</td>
<td>TitlesAndFooters</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></td>
<td>Note: This is an abstract style element.</td>
<td></td>
</tr>
<tr>
<td>SystemTitle</td>
<td>Controls system title text</td>
<td>TitlesAndFooters</td>
</tr>
<tr>
<td>SystemTitle2-10</td>
<td>Controls SystemTitle text for the corresponding SystemTitle value. For example, SystemTitle3 controls SystemTitle3 text.</td>
<td>SystemTitle-SystemTitle9</td>
</tr>
<tr>
<td>SystemFooter</td>
<td>Controls system footer text</td>
<td>TitlesAndFooters</td>
</tr>
<tr>
<td>SystemFooter2-10</td>
<td>Controls SystemFooter text for the corresponding SystemFooter value. For example, SystemFooter3 controls SystemFooter3 text.</td>
<td>SystemFooter-SystemFooter9</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>
<tr>
<td>Parskip</td>
<td>Controls space between tables in RTF output</td>
<td>TitlesAndFooters</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>
<tr>
<td>UserText</td>
<td>Controls the ODS TEXT= style</td>
<td>Note</td>
</tr>
<tr>
<td>PrePage</td>
<td>Controls the ODS RTF/MEASURED PREPAGE= style</td>
<td>Note</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>Note</td>
<td>Controls the container for note banners and note contents</td>
<td>Container</td>
</tr>
<tr>
<td></td>
<td><em>Note:</em> This is an abstract style element.</td>
<td></td>
</tr>
<tr>
<td>NoteBanner</td>
<td>Controls the banner for Note</td>
<td>Note</td>
</tr>
<tr>
<td>NoteContent</td>
<td>Controls the contents for Note</td>
<td>Note</td>
</tr>
<tr>
<td>NoteContentFixed</td>
<td>Controls the contents for Note. Fixed font.</td>
<td>NoteContent</td>
</tr>
<tr>
<td>WarnBanner</td>
<td>Controls the banner for Warnings</td>
<td>Note</td>
</tr>
<tr>
<td>WarnContent</td>
<td>Controls the contents of Warnings</td>
<td>Note</td>
</tr>
<tr>
<td>WarnContentFixed</td>
<td>Controls the contents for Warnings. Fixed font.</td>
<td>WarnContent</td>
</tr>
<tr>
<td>ErrorBanner</td>
<td>Controls the banner for Errors</td>
<td>Note</td>
</tr>
<tr>
<td>ErrorContent</td>
<td>Controls the contents of Errors</td>
<td>Note</td>
</tr>
<tr>
<td>ErrorContentFixed</td>
<td>Controls the contents for Errors. Fixed font.</td>
<td>ErrorContent</td>
</tr>
<tr>
<td>FatalBanner</td>
<td>Controls the banner for Fatal</td>
<td>Note</td>
</tr>
<tr>
<td>FatalContent</td>
<td>Controls the contents of Fatal</td>
<td>Note</td>
</tr>
<tr>
<td>FatalContentFixed</td>
<td>Controls the contents for Fatal. Fixed font.</td>
<td>FatalContent</td>
</tr>
<tr>
<td>Paragraph</td>
<td>Controls paragraphs in text</td>
<td>Container</td>
</tr>
<tr>
<td>Heading1-Heading10</td>
<td>Controls Heading text for the corresponding Heading&lt;param&gt; value. For example, Heading2 controls the Heading2 text.</td>
<td>NormalText</td>
</tr>
<tr>
<td>List</td>
<td>Controls both unordered and ordered lists</td>
<td>Container</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>List2-List10</td>
<td>Controls nested bulleted lists at the corresponding Listn level. For example, List3 controls the third level bulleted list.</td>
<td>List-List9</td>
</tr>
<tr>
<td>ListItem</td>
<td>Controls bulleted list items</td>
<td>Container</td>
</tr>
<tr>
<td>ListItem2-ListItem10</td>
<td>Controls the nested bulleted list items at the corresponding ListItemn level. For example, ListItem3 controls the third level bulleted list item.</td>
<td>ListItem-ListItem9</td>
</tr>
<tr>
<td>Pagebreak</td>
<td>Controls page breaks in the text</td>
<td>No inheritance</td>
</tr>
<tr>
<td>LineContent</td>
<td>Controls the text generated by the LINE statement in PROC REPORT</td>
<td>No inheritance</td>
</tr>
<tr>
<td>Link</td>
<td>Controls the color of unvisited link text</td>
<td>No inheritance</td>
</tr>
<tr>
<td>ActiveLink</td>
<td>Controls the color of active (clicked) link text</td>
<td>No inheritance</td>
</tr>
<tr>
<td>FocusLink</td>
<td>Controls the color of the link-focus indicator</td>
<td>No inheritance</td>
</tr>
<tr>
<td>Accessibility note: For an example that uses the FocusLink style element, see Enhancing the Appearance of the Link-Focus Indicator.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VisitedLink</td>
<td>Controls the color of visited link text</td>
<td>No inheritance</td>
</tr>
</tbody>
</table>

**Style Elements Affecting Layout**

**Table 11.9 Elements Affecting Layout**

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Description</th>
<th>Inheritance</th>
</tr>
</thead>
<tbody>
<tr>
<td>LayoutContainer</td>
<td>Container for Layout</td>
<td>No inheritance</td>
</tr>
<tr>
<td>LayoutRegion</td>
<td>Region style for Layout cells</td>
<td>LayoutContainer</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 ODS Graphics output where a compiled ODS template of type STATGRAPH is used to produce graphical output. Supplied templates are stored in
Sashelp.Tmplmst. SAS/GRAPH device drivers and some global statements such as SYMBOL, PATTERN, AXIS, and LEGEND have no affect. Common ODS Graphics procedures that produce template-based graphics are SGPLOT, SG PANEL, 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 11.11  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>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></td>
<td></td>
<td>OutputHeight</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OutputWidth</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>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transparency</td>
</tr>
<tr>
<td>GraphAnnoText</td>
<td>Annotation text</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Font</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>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Recognized Attributes</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>GraphBackground</td>
<td>Background of the graph</td>
<td>Color, Transparency</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphBorderLines</td>
<td>Border around graph wall, legend border, borders to complete axis frame</td>
<td>ContrastColor, LineStyle, LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphDataText</td>
<td>Text font and color for point and line labels</td>
<td>Color, Font, <em>font-attributes</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphFootnoteText</td>
<td>Text font and color for footnote(s)</td>
<td>Color, Font, <em>font-attributes</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphGridLines</td>
<td>Horizontal and vertical grid lines drawn at major tick marks</td>
<td>Color, ContrastColor, DisplayOpts, LineStyle,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphHeaderBackground</td>
<td>Background color of the legend title</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphLabelText</td>
<td>Text font and color for axis labels and legend titles</td>
<td>Color, Font, <em>font-attributes</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphLegendBackground</td>
<td>Background color of the legend</td>
<td>Color, FrameBorder</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphMinorGridLines</td>
<td>Appearance of the grid lines.</td>
<td>ContrastColor, DisplayOpts, LineStyle, LineThickness</td>
</tr>
<tr>
<td></td>
<td></td>
<td></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></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GraphReference</td>
<td>Horizontal and vertical reference lines and drop lines</td>
<td>ContrastColor, LineStyle, LineThickness</td>
</tr>
</tbody>
</table>
### Table 11.12  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>GraphTitleText</td>
<td>Text font and color for title(s)</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Font</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or font-attributes*</td>
</tr>
<tr>
<td>GraphUnicodeText</td>
<td>Text font for Unicode values</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Font</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or font-attributes*</td>
</tr>
<tr>
<td>GraphValueText</td>
<td>Text font and color for axis tick values and legend values</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Font</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or font-attributes*</td>
</tr>
<tr>
<td>GraphWalls</td>
<td>Vertical wall(s) bounded by axes</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FrameBorderColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td>*</td>
<td>* Font-attributes can be one of the following: FONTFAMILY=, FONTSIZE=, FONTSTYLE=, FONTWEIGHT=,</td>
<td></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, colors for bands and lines</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Color</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>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Recognized Attributes</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>GraphConfidence2</td>
<td>Secondary confidence lines and bands, color for bands, and contrast color for lines</td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Color</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>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>
<tr>
<td>GraphCutLine</td>
<td>Cutline attributes for a dendogram</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ContrastColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EndColor</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>NeutralColor</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>StartColor</td>
</tr>
<tr>
<td>GraphError</td>
<td>Error line or error bar fill, ContrastColor for lines, Color for bar fill</td>
<td>CapStyle</td>
</tr>
<tr>
<td></td>
<td></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>Transparency</td>
</tr>
<tr>
<td>GraphFit</td>
<td>Primary fit lines such as a normal density curve</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>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Recognized Attributes</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>GraphFit2</td>
<td>Secondary fit lines such as a kernel density curve</td>
<td>Color, ContrastColor, LineStyle, LineThickness, MarkerSize, MarkerSymbol</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</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</td>
</tr>
<tr>
<td>GraphMissing</td>
<td>Properties for graph items representing missing values</td>
<td>Color, ContrastColor, FillPattern*, LineStyle, LineThickness, MarkerSymbol, MarkerSize, Transparency</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</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>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</td>
</tr>
<tr>
<td>GraphOutlier</td>
<td>Outlier data for the graph</td>
<td>ContrastColor&lt;br&gt;Color&lt;br&gt;MarkerSize&lt;br&gt;MarkerSymbol&lt;br&gt;LineStyle&lt;br&gt;LineThickness</td>
</tr>
<tr>
<td>GraphPrediction</td>
<td>Prediction lines</td>
<td>ContrastColor&lt;br&gt;Color&lt;br&gt;LineStyle&lt;br&gt;LineThickness&lt;br&gt;MarkerSize&lt;br&gt;MarkerSymbol</td>
</tr>
<tr>
<td>GraphPredictionLimits</td>
<td>Fills for prediction limits</td>
<td>ContrastColor&lt;br&gt;Color&lt;br&gt;MarkerSize&lt;br&gt;MarkerSymbol</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&lt;br&gt;Color&lt;br&gt;MarkerSymbol&lt;br&gt;MarkerSize&lt;br&gt;LineStyle&lt;br&gt;LineThickness</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>
</tbody>
</table>
### Table 11.13  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>
<tbody>
<tr>
<td>GraphData1</td>
<td>Primitives related to the first seven grouped data items. Color applies to filled areas. ContrastColor applies to markers and lines.</td>
<td>Color, ContrastColor, FillPattern*, LineStyle, MarkerSymbol</td>
</tr>
<tr>
<td>GraphData2</td>
<td>Primitives related to the 8th through 11th grouped data items.</td>
<td>Color, ContrastColor, FillPattern*</td>
</tr>
<tr>
<td>GraphData3</td>
<td>Primitives related to the 12th grouped data item.</td>
<td>Color, ContrastColor, FillPattern*</td>
</tr>
<tr>
<td>GraphData4</td>
<td>Primitives related to the 13th through 15th grouped data items.</td>
<td>FillPattern*</td>
</tr>
<tr>
<td>GraphData5</td>
<td>Primitives related to the 16th through 18th grouped data items.</td>
<td>FillPattern*</td>
</tr>
<tr>
<td>GraphData6</td>
<td>Primitives related to the 19th through 21th grouped data items.</td>
<td>FillPattern*</td>
</tr>
<tr>
<td>GraphData7</td>
<td>Primitives related to the 22nd through 24th grouped data items.</td>
<td>FillPattern*</td>
</tr>
<tr>
<td>GraphData8</td>
<td>Primitives related to the 25th through 27th grouped data items.</td>
<td>FillPattern*</td>
</tr>
<tr>
<td>GraphData9</td>
<td>Primitives related to the 28th through 30th grouped data items.</td>
<td>FillPattern*</td>
</tr>
<tr>
<td>GraphData10</td>
<td>Primitives related to the 31st through 33th grouped data items.</td>
<td>FillPattern*</td>
</tr>
<tr>
<td>GraphData11</td>
<td>Primitives related to the 34th through 36th grouped data items.</td>
<td>FillPattern*</td>
</tr>
<tr>
<td>GraphData12</td>
<td>Primitives related to the 37th through 39th grouped data items.</td>
<td>FillPattern*</td>
</tr>
<tr>
<td>GraphData13**</td>
<td>Primitives related to the 40th through 42th grouped data items.</td>
<td>FillPattern*</td>
</tr>
<tr>
<td>GraphData14</td>
<td>Primitives related to the 43rd through 45th grouped data items.</td>
<td>FillPattern*</td>
</tr>
<tr>
<td>GraphData15</td>
<td>Primitives related to the 46th through 48th grouped data items.</td>
<td>FillPattern*</td>
</tr>
</tbody>
</table>

* Prior to SAS 9.4M5, style attribute FillPattern is available only with the JOURNAL2, JOURNAL3, and MONOCHROMEPRINTER styles. Starting with SAS 9.4M5, style attribute FillPattern in GraphData1–GraphData11 is also available with the DEFAULT ODS style and all styles that are derived from it.

* Style elements GraphData13–GraphData15 are available only with the JOURNAL2, JOURNAL3, and MONOCHROMEPRINTER styles.
<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 fillpattern outline&quot; *</td>
</tr>
<tr>
<td>GraphBar</td>
<td>Display options for bar charts</td>
<td>DisplayOpts</td>
<td>&quot;Connect fill fillpattern outline&quot;</td>
</tr>
<tr>
<td>GraphBlock</td>
<td>Fill color for block plots</td>
<td>Color</td>
<td>GraphColors(&quot;gblock&quot;)</td>
</tr>
<tr>
<td>GraphBox</td>
<td>Display options for box plots</td>
<td>DisplayOpts</td>
<td>&quot;Caps connect fill fillpattern mean median notches outliers&quot; *</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CapStyle</td>
<td>&quot;Serif&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Connect</td>
<td>&quot;Mean&quot;</td>
</tr>
<tr>
<td>GraphEllipse</td>
<td>Display options for confidence ellipses</td>
<td>DisplayOpts</td>
<td>&quot;Fill fillpattern outline&quot; *</td>
</tr>
<tr>
<td>GraphHighLow</td>
<td>Display options for high-low plots</td>
<td>DisplayOpts</td>
<td>&quot;Fill fillpattern outline&quot;</td>
</tr>
<tr>
<td>GraphHistogram</td>
<td>Display options for histograms</td>
<td>DisplayOpts</td>
<td>&quot;Fill fillpattern outline&quot; *</td>
</tr>
<tr>
<td>GraphPolygon</td>
<td>Display options for polygon plots</td>
<td>DisplayOpts</td>
<td>&quot;Fill fillpattern outline&quot;</td>
</tr>
<tr>
<td>GraphSkins</td>
<td>Display skins</td>
<td>DataSkin</td>
<td>&quot;Crisp&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;Gloss&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;Matte&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;None&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;Pressed&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KpiSkin</td>
<td>&quot;Basic&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;Modern&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;None&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;Onyx&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>&quot;Satin&quot;</td>
</tr>
</tbody>
</table>

* Display option FillPattern is available starting with SAS 9.4M5.

** Style elements GraphHighLow and GraphPolygon are valid starting with SAS 9.4M5.
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 11.15  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>
</tr>
</thead>
<tbody>
<tr>
<td>DropShadowStyle</td>
<td>Used with text types</td>
<td>Color</td>
</tr>
<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>GraphAxisLines</td>
<td>X, Y, and Z-axis lines</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineStyle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LineThickness</td>
</tr>
<tr>
<td>GraphBackground</td>
<td>Background of the graph</td>
<td>Transparency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BackgroundColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gradient_Direction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StartColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EndColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BackgroundImage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Image</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VerticalAlign</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TextAlign</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>GraphBorderLines</td>
<td>Border around graph wall, legend border, borders to complete axis frame</td>
<td>Color, LineThickness, LineStyle</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>
</tr>
<tr>
<td>GraphDataText</td>
<td>Text font and color for point and line labels</td>
<td>Font or font-attributes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Color</td>
</tr>
<tr>
<td>GraphFloor</td>
<td>3-D floor</td>
<td>BackgroundColor, Transparency, Gradient_Direction, StartColor, EndColor, BackgroundImage, Image, VerticalAlign, TextAlign</td>
</tr>
<tr>
<td>GraphFootnoteText</td>
<td>Text font and color for footnotes</td>
<td>Font or font-attributes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Color</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>
</tr>
<tr>
<td>GraphGridLines</td>
<td>Horizontal and vertical grid lines drawn at major tick marks</td>
<td>Color, LineStyle, LineThickness, Transparency, displayopts</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>GraphLegendBackground</td>
<td>Background color of the legend</td>
<td>Color, FrameBorder, Transparency</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>
</tr>
<tr>
<td>GraphTitle1Text</td>
<td>Text font and color for the first title</td>
<td>Font or font-attributes*, Color</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>
</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>GraphWalls</td>
<td>Vertical walls bounded by axes</td>
<td>Transparency, BackgroundColor, Gradient_Direction, StartColor, EndColor, BackgroundImage, Image</td>
</tr>
</tbody>
</table>

*Font-attributes can be one of the following: FONTFAMILY=, FONTSIZE=, FONTSTYLE=, FONTWEIGHT=.

Table 11.16  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>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>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Default Attributes</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------------------</td>
<td>-------------------------------------</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></td>
<td></td>
<td>TextColor</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></td>
<td></td>
<td>TextColor</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>markers and 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>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TextColor</td>
</tr>
<tr>
<td>GraphUnderflow</td>
<td>Underflow data for the graph. Color applies to filled areas. ContrastColor applies to</td>
<td>Color</td>
</tr>
<tr>
<td></td>
<td>markers and 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>MarkerSize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MarkerSymbol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TextColor</td>
</tr>
<tr>
<td>ThreeColorAltRamp</td>
<td>Line contours, markers, and data labels with segmented range color response</td>
<td>StartColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NeutralColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EndColor</td>
</tr>
<tr>
<td>ThreeColorRamp</td>
<td>Gradient contours, surfaces, markers, and data labels with continuous color</td>
<td>StartColor</td>
</tr>
<tr>
<td></td>
<td>response</td>
<td>NeutralColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EndColor</td>
</tr>
<tr>
<td>TwoColorAltRamp</td>
<td>Line contours, markers, and data labels with segmented range color response</td>
<td>StartColor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>TwoColorRamp</td>
<td>Gradient contours, surfaces, markers, and data labels with continuous color response</td>
<td>StartColor EndColor</td>
</tr>
</tbody>
</table>

**Table 11.17**  Style Elements Affecting Device-Based 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>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>Style Element</td>
<td>Portion of Graph Affected</td>
<td>Default Attributes</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------</td>
<td>--------------------</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>GraphData4</td>
<td>Primitives related to 4th grouped data items</td>
<td>BackGroundImage, ContrastColor, Color, EndColor, Gradient_Direction, Image, LineStyle, LineThickness, MarkerSize, MarkerSymbol, StartColor</td>
</tr>
<tr>
<td>GraphData5</td>
<td>Primitives related to 5th 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>
| GraphData6    | Primitives related to 6th grouped data items | BackGroundImage  
ContrastColor  
Color  
EndColor  
Gradient_Direction  
Image  
LineStyle  
LineThickness  
MarkerSize  
MarkerSymbol  
StartColor |
| 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 |
<table>
<thead>
<tr>
<th>Style Element</th>
<th>Portion of Graph Affected</th>
<th>Default Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraphData9</td>
<td>Primitives related to 9th grouped data items</td>
<td>BackGroundImage, ContrastColor, Color, EndColor, Gradient_Direction, Image, LineStyle, LineThickness, MarkerSize, MarkerSymbol, StartColor</td>
</tr>
<tr>
<td>GraphData10</td>
<td>Primitives related to 10th grouped data items</td>
<td>BackGroundImage, ContrastColor, Color, EndColor, Gradient_Direction, Image, LineStyle, LineThickness, MarkerSize, MarkerSymbol, StartColor</td>
</tr>
<tr>
<td>GraphData11</td>
<td>Primitives related to 11th 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>
<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>
Chapter 12
Style Attributes

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 1136 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 1045.
Style Attributes Tables

Overview

For usage information about these style attributes, such as aliases, restrictions, and examples, see “Style Attributes Detailed Information” on page 1098.

EPUB Destination

The EPUB destination includes the following ODS statements:

- ODS EPUB Statement on page 215
- ODS EPUB2 Statement on page 217
- ODS EPUB3 Statement on page 237

Table 12.1  Tables

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify whether the border is collapsed or separated</td>
<td>“BORDERCollapse=COLLAPSE</td>
</tr>
<tr>
<td>Specify an ID for the table or table cell</td>
<td>“HTMLID=“string””</td>
</tr>
<tr>
<td>Specify the minimum number of lines of text that must appear in a paragraph before it is forced to move to another page</td>
<td>“NOBREAKSPACE=ON</td>
</tr>
<tr>
<td>Specify the number of lines of text that must appear at the top of a page if a paragraph is separated by a page break</td>
<td>“WIDTH=dimension ”</td>
</tr>
</tbody>
</table>

Table 12.2  Documents

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the minimum number of lines of text that must appear in a paragraph before it is forced to move to another page</td>
<td>“ORPHAN=integer”</td>
</tr>
<tr>
<td>Specify the number of lines of text that must appear at the top of a page if a paragraph is separated by a page break</td>
<td>“WINDOW=integer”</td>
</tr>
</tbody>
</table>

Excel Destination

The ODS destination for Excel includes the ODS EXCEL Statement on page 296.
### Table 12.3  Graphs

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the size of the font for tables, table cells, and graphs</td>
<td>“\texttt{FONTSIZE=dimension</td>
</tr>
<tr>
<td>Specify the style of the font for tables, table cells, and graphs. In many cases, italic and slant map to the same font.</td>
<td>“\texttt{FONTSTYLE=ITALIC</td>
</tr>
<tr>
<td>Specify justification in tables, table cells, and graphs</td>
<td>“\texttt{TEXTALIGN=alignment}”</td>
</tr>
<tr>
<td>Change the visual presentation of the text</td>
<td>“\texttt{TEXTDECORATION=presentation–options}”</td>
</tr>
<tr>
<td>Specify the number of spaces that the first line of output is indented</td>
<td>“\texttt{TEXTINDENT=n}”</td>
</tr>
</tbody>
</table>

### Table 12.4  Document

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify how to handle leading spaces and line breaks in table cells and HTML documents</td>
<td>“\texttt{ASIS=ON</td>
</tr>
<tr>
<td>Specify the string to use for the bullets in the contents file</td>
<td>“\texttt{LISTSTYLETYPE=bullet-type}”</td>
</tr>
<tr>
<td>Specify how to handle space characters in table cells</td>
<td>“\texttt{NOBREAKSPACE=ON</td>
</tr>
<tr>
<td>Specify how less-than signs (&lt;), greater-than signs (&gt;), and ampersands (&amp;) are interpreted in table cells</td>
<td>“\texttt{PROTECTSPECIALCHARS=ON</td>
</tr>
</tbody>
</table>

### Table 12.5  Tables

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify how to handle leading spaces and line breaks in table cells and HTML documents</td>
<td>“\texttt{ASIS=ON</td>
</tr>
<tr>
<td>Specify the color of the background of tables, table cells, or graphs</td>
<td>“\texttt{BACKGROUNDCOLOR=color}”</td>
</tr>
<tr>
<td>Specify an image to use as the background</td>
<td>“\texttt{BACKGROUNDIMAGE=string}”</td>
</tr>
<tr>
<td>Specify the color of the bottom border of the table</td>
<td>“\texttt{BORDERBOTTOMCOLOR=color}”</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Specify the line style of the bottom border of the selected cell</td>
<td>“BORDERBOTTOMSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the bottom border of a table or table cell</td>
<td>“BORDERBOTTOMWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the color of the border in a table or table cell if the border is just one color</td>
<td>“BORDERCOLOR=color”</td>
</tr>
<tr>
<td>Specify the color of the left border of a table</td>
<td>“BORDERLEFTCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the left border of the specified table cell</td>
<td>“BORDERLEFTSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the left border of the table</td>
<td>“BORDERLEFTWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the color of the right border of the table</td>
<td>“BORDERRIGHTCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the right border of the selected cell</td>
<td>“BORDERRIGHTSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the right border of the table</td>
<td>“BORDERRIGHTWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the border style of one or more borders</td>
<td>“BORDERSTYLE=line-style-1 &lt;, ..., line-style-4 &gt;”</td>
</tr>
<tr>
<td>Specify the color of the top border of a table or table cell</td>
<td>“BORDERTOPCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the top border of the specified table cell</td>
<td>“BORDERTOPSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the top border of the table</td>
<td>“BORDERTOPWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the width of the border of the table</td>
<td>“BORDERWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the text to show in a data tip for various reporting elements, including table cells</td>
<td>“FLYOVER=&quot;string&quot;”</td>
</tr>
<tr>
<td>Specify the font to use in table cells and graphs</td>
<td>“FONTFAMILY=&quot;string-1&lt;…, string-n&gt;&quot;”</td>
</tr>
<tr>
<td>Specify the size of the font for tables, table cells, and graphs</td>
<td>“FONTSIZE=dimension</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Specify the style of the font for tables, table cells, and graphs.</td>
<td>“FONTSTYLE=ITALIC</td>
</tr>
<tr>
<td>In many cases, italic and slant map to the same font.</td>
<td></td>
</tr>
<tr>
<td>Specify the font weight of tables, table cells, and graphs</td>
<td>“FONTWEIGHT=weight”</td>
</tr>
<tr>
<td>Specify text to place after the table cell or table</td>
<td>“POSTTEXT=”string”</td>
</tr>
<tr>
<td>Specify text to place before the table cell or table</td>
<td>“PRETEXT=”string”</td>
</tr>
<tr>
<td>Specify how less-than signs (&lt;), greater-than signs (&gt;), and ampersands (&amp;) are interpreted in table cells</td>
<td>“PROTECTSPECIALCHARS=ON</td>
</tr>
<tr>
<td>Specify how to handle space characters in table cells</td>
<td>“NOBREAKSPACE=ON</td>
</tr>
<tr>
<td>Specify a string which contains one or more of the following: format, formula, mergeacross, rotate, hidden, wrap</td>
<td>“TAGATTR=”string”</td>
</tr>
<tr>
<td>Specify justification in tables, table cells, and graphs</td>
<td>“TEXTALIGN=alignment”</td>
</tr>
<tr>
<td>Change the visual presentation of the text</td>
<td>“TEXTDECORATION=presentation–options”</td>
</tr>
<tr>
<td>Specify the number of spaces that the first line of output is indented</td>
<td>“TEXTINDENT=n”</td>
</tr>
<tr>
<td>Specify vertical justification</td>
<td>“VERTICALALIGN=BOTTOM</td>
</tr>
<tr>
<td>Specify a URL to link to from various reporting elements, including table cells</td>
<td>“URL=”uniform-resource-locator”</td>
</tr>
<tr>
<td>Specify the width of a table cell, table, line, or a graph</td>
<td>“WIDTH=dimension ”</td>
</tr>
</tbody>
</table>

**HTML5 Destination**

HTML5 belongs to the markup family of destinations. The following table contains all the style attributes for the markup family and the style attributes that are specific to ODS HTML5.
### Table 12.6 Output and Text

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the position of the background of the tables, table cells, or graphs</td>
<td>“BACKGROUNDPOSITION=position”</td>
</tr>
<tr>
<td>Specify an image to use as the background</td>
<td>“BACKGROUNDIMAGE=&quot;string&quot;&quot;</td>
</tr>
<tr>
<td>Specify whether an image is repeated horizontally, vertically, both, or not repeated</td>
<td>“BACKGROUNDREPEAT=option”</td>
</tr>
<tr>
<td>Specify the font weight of tables, table cells, and graphs</td>
<td>“FONTWEIGHT=weight”</td>
</tr>
<tr>
<td>Specify the height of a table cell, graph, or graphics in an HTML document</td>
<td>“HEIGHT=dimension ”</td>
</tr>
<tr>
<td>Specify justification in tables, table cells, and graphs</td>
<td>“TEXTALIGN=alignment”</td>
</tr>
</tbody>
</table>

### Table 12.7 HTML Document

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the position of the background of the tables, table cells, or graphs</td>
<td>“BACKGROUNDPOSITION=position”</td>
</tr>
<tr>
<td>Specify whether to put a scroll bar in the frame that references the body file</td>
<td>“BODYSCROLLBAR=YES</td>
</tr>
<tr>
<td>Specify the width of the frame that displays the body file in the HTML frame file</td>
<td>“BODYSIZE=dimension</td>
</tr>
<tr>
<td>Specify whether to put a scroll bar in the frames in the frame file that display the contents and the page files</td>
<td>“CONTENTSCROLLBAR=YES</td>
</tr>
<tr>
<td>Specify whether to put a border around the frame for an HTML file that uses frames</td>
<td>“FRAMEBORDER=ON</td>
</tr>
<tr>
<td>Specify the height of a table cell, graph, or graphics in an HTML document</td>
<td>“HEIGHT=dimension ”</td>
</tr>
<tr>
<td>Specify whether to make the entry in the table of contents a link to the body file</td>
<td>“LISTENTRYANCHOR=ON</td>
</tr>
<tr>
<td>Specify a path or URL to an image file to be used as the bullet for list items.</td>
<td>“LISTSTYLEIMAGE=string”</td>
</tr>
<tr>
<td>Specify the bottom margin for the HTML document</td>
<td>“MARGINBOTTOM=dimension”</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>Specify the left margin for the HTML document</td>
<td>“MARGINLEFT=dimension”</td>
</tr>
<tr>
<td>Specify the right margin for the HTML document</td>
<td>“MARGINRIGHT=dimension”</td>
</tr>
<tr>
<td>Specify the top margin for the HTML document</td>
<td>“MARGINTOP=dimension”</td>
</tr>
<tr>
<td>Specify the color of the link-focus indicator outline</td>
<td>“OUTLINECOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the link-focus indicator outline</td>
<td>“OUTLINESTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the line width of the link-focus indicator outline</td>
<td>“OUTLINEWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify HTML to place at page breaks in an HTML document</td>
<td>“PAGEBREAKHTML=&quot;string””</td>
</tr>
<tr>
<td>Specify the width of a table cell, table, line, or a graph</td>
<td>“WIDTH=dimension”</td>
</tr>
</tbody>
</table>

**Table 12.8  Tables**

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify how to handle leading spaces and line breaks in table cells and HTML documents</td>
<td>“ASIS=ON</td>
</tr>
<tr>
<td>Specify an image to use as the background</td>
<td>“BACKGROUNDIMAGE=&quot;string””</td>
</tr>
<tr>
<td>Specify the position of the background of the tables, table cells, or graphs</td>
<td>“BACKGROUNDPOSITION=position”</td>
</tr>
<tr>
<td>Specify whether an image is repeated horizontally, vertically, both, or not repeated</td>
<td>“BACKGROUNDREPEAT=option”</td>
</tr>
<tr>
<td>Specify the color of the bottom border of the table</td>
<td>“BORDERBOTTOMCOLOR=color ”</td>
</tr>
<tr>
<td>Specify the line style of the bottom border of the selected cell</td>
<td>“BORDERBOTTOMSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the bottom border of the table</td>
<td>“BORDERBOTTOMWIDTH=dimension ”</td>
</tr>
<tr>
<td>Specify whether the border is collapsed or separated</td>
<td>“BORDERCOLLAPSE=COLLAPSE</td>
</tr>
</tbody>
</table>

**Style Attributes Tables**
<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the color of the border in a table or table cell if the border is just one color</td>
<td>“BORDERCOLOR=color”</td>
</tr>
<tr>
<td>Specify the lighter color to use in a border that uses two colors to create a three-dimensional effect</td>
<td>“BORDERCOLORLIGHT=color”</td>
</tr>
<tr>
<td>Specify the color of the left border of a table</td>
<td>“BORDERLEFTCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the left border of the specified table cell</td>
<td>“BORDERLEFTSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the left border of the table</td>
<td>“BORDERLEFTWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the color of the right border of the table</td>
<td>“BORDERRIGHTCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the right border of the selected cell</td>
<td>“BORDERRIGHTSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the right border of the table</td>
<td>“BORDERRIGHTWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the thickness of the spacing between cells in a table</td>
<td>“BORDERSPACING=dimension”</td>
</tr>
<tr>
<td>Specify the border style of one or more borders</td>
<td>“BORDERSTYLE=line-style-1 &lt;, ..., line-style-4 &gt;”</td>
</tr>
<tr>
<td>Specify the color of the top border of the table</td>
<td>“BORDERTOPCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the top border of the specified table cell</td>
<td>“BORDERTOPSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the top border of the table</td>
<td>“BORDERTOPWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the width of the border of the table</td>
<td>“BORDERWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the amount of white space on each of the four sides of the content in a table cell</td>
<td>“CELLPADDING=dimension</td>
</tr>
<tr>
<td>Specify whether the is collapsed or separated</td>
<td>“BORDERCOLLAPSE=COLLAPSE</td>
</tr>
<tr>
<td>Specify the font to use in table cells and graphs</td>
<td>“FONTFAMILY=&quot;string-1&lt;…, string-n&gt;&quot;”</td>
</tr>
<tr>
<td>Specify the size of the font for tables, table cells, and graphs</td>
<td>“FONTSIZE=dimension</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Specify the style of the font for tables, table cells, and graphs</td>
<td>“FONTSTYLE=ITALIC</td>
</tr>
<tr>
<td>Specify the font weight of tables, table cells, and graphs</td>
<td>“FONTWEIGHT=weight”</td>
</tr>
<tr>
<td>Specify the text to show in a data tip for the table cell</td>
<td>“FLYOVER=&quot;string&quot;”</td>
</tr>
<tr>
<td>Specify the height of a table cell, graph, or graphics in an HTML document</td>
<td>“HEIGHT=dimension ”</td>
</tr>
<tr>
<td>Specify the window or frame in which to open the target of the link</td>
<td>“HREFTARGET=&quot;target&quot; ”</td>
</tr>
<tr>
<td>Specify individual attributes and values for a table or table cell in an HTML document</td>
<td>“HTMLSTYLE=&quot;string&quot;”</td>
</tr>
<tr>
<td>Specify the color of the left of a table</td>
<td>“BORDERLEFTCOLOR=color”</td>
</tr>
<tr>
<td>Specify the amount of white space between the content of the table cell and the border</td>
<td>“PADDING=dimension</td>
</tr>
<tr>
<td>Specify the amount of white space on the bottom of the content of the table cell</td>
<td>“PADDINGBOTTOM=dimension</td>
</tr>
<tr>
<td>Specify the amount of white space on the left side of the content of the table cell</td>
<td>“PADDINGLEFT=dimension</td>
</tr>
<tr>
<td>Specify the amount of white space on the right side of the content of the table cell</td>
<td>“PADDINGRIGHT=dimension</td>
</tr>
<tr>
<td>Specify the amount of white space on the top of the content of the table cell</td>
<td>“PADDINGTOP=dimension</td>
</tr>
<tr>
<td>Specify the HTML code to place after the table or table cell</td>
<td>“POSTHTML=&quot;string&quot;”</td>
</tr>
<tr>
<td>Specify an image to place before the table or table cell</td>
<td>“POSTIMAGE=&quot;external-file&quot;</td>
</tr>
<tr>
<td>Specify text to place after the table cell or table</td>
<td>“POSTTEXT=&quot;string&quot;”</td>
</tr>
<tr>
<td>Specify the HTML code to place before the table or table cell</td>
<td>“PREHTML=&quot;string&quot;”</td>
</tr>
<tr>
<td>Specify an image to place before the table or table cell</td>
<td>“PREIMAGE=&quot;external-file&quot;</td>
</tr>
<tr>
<td>Specify text to place before the table cell or table</td>
<td>“PRETEXT=&quot;string&quot;”</td>
</tr>
</tbody>
</table>
### HTML4 Destination

HTML4 belongs to the markup family of destinations. The following table contains all the style attributes for the markup family and the style attributes that are specific to ODS HTML4.

**Table 12.9 HTML Document**

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<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>“ACTIVELINKCOLOR=color”</td>
</tr>
<tr>
<td>Specify whether to put a scroll bar in the frame that references the body file</td>
<td>“BODYSCROLLBAR=YES</td>
</tr>
<tr>
<td>Specify the width of the frame that displays the body file in the HTML frame file</td>
<td>“BODYSIZE=dimension</td>
</tr>
<tr>
<td>Specify whether to put a scroll bar in the frames in the frame file that display the contents and the page files</td>
<td>“CONTENTSscrollBAR=YES</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>Specify the width of the frames in the frame file that display the contents and the page files</td>
<td>“CONTENTSIZE=dimension</td>
</tr>
<tr>
<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>“CONTENTTYPE=&quot;string&quot;&quot;</td>
</tr>
<tr>
<td>Specify the entire doctype declaration for the HTML document</td>
<td>“DOCTYPE=&quot;string&quot;&quot;</td>
</tr>
<tr>
<td>Specify whether to put a border around the frame for an HTML file that uses frames</td>
<td>“FRAMEBORDER=ON</td>
</tr>
<tr>
<td>Specify the color of the link-focus indicator outline</td>
<td>“OUTLINECOLOR=color”</td>
</tr>
<tr>
<td>Specify whether to make the entry in the table of contents a link to the body file</td>
<td>“LISTENTRYANCHOR=ON</td>
</tr>
<tr>
<td>Specify the color for the links in an HTML document that have not yet been visited</td>
<td>“LINKCOLOR=color”</td>
</tr>
<tr>
<td>Specify HTML to place at page breaks in an HTML document</td>
<td>“PAGEBREAKHTML=&quot;string&quot;&quot;</td>
</tr>
<tr>
<td>Specify the color for links that have been visited in an HTML document</td>
<td>“VISITEDLINKCOLOR=color”</td>
</tr>
<tr>
<td>Specify whether to make the image that is specified by BACKGROUNDIMAGE= into a &quot;watermark &quot;</td>
<td>“WATERMARK=ON</td>
</tr>
</tbody>
</table>

**Table 12.10  Tables**

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the thickness of the spacing between cells in a table</td>
<td>“BORDERSPACING=dimension”</td>
</tr>
<tr>
<td>Specify whether the border is collapsed or separated</td>
<td>“BORDERCollapse=COLLAPSE</td>
</tr>
<tr>
<td>Specify the amount of white space on each of the four sides of the content in a table cell</td>
<td>“CELLPADDING=dimension</td>
</tr>
<tr>
<td>Specify an ID for the table or table cell</td>
<td>“HTMLID=&quot;string&quot;”</td>
</tr>
<tr>
<td>Specify the amount of white space between the content of the table cell and the border</td>
<td>“PADDING=dimension</td>
</tr>
<tr>
<td>Specify text to insert in the HTML</td>
<td>“TAGATTR=&quot;string&quot;”</td>
</tr>
</tbody>
</table>
**Markup Family of Destinations**

The markup family of destinations includes the following ODS statements:

- ODS CSVALL Statement on page 200
- ODS HTML3 Statement on page 369
- ODS HTML Statement on page 396
- ODS CHTML Statement on page 174
- ODS PHTML Statement on page 695

**Table 12.11  Document**

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify whether styles used in an HTML document are used in CSS style files</td>
<td>“ABSTRACT=ON</td>
</tr>
</tbody>
</table>

**Table 12.12  Output and Text**

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the alternate colors for maps</td>
<td>“CONTRASTCOLOR=color”</td>
</tr>
<tr>
<td>Specify the entire doctype declaration for the HTML document</td>
<td>“DOCTYPE=”string””</td>
</tr>
<tr>
<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>“FILLRULEWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the text to show in a data tip for the table cell</td>
<td>“FLYOVER=”string””</td>
</tr>
<tr>
<td>Specify a font definition to use in tables, table cells, and graphs</td>
<td>“FONT=font-definition”</td>
</tr>
<tr>
<td>Specify the font to use in table cells and graphs</td>
<td>“FONTFAMILY=&quot;string-1&lt;…, string-n&gt;&quot;”</td>
</tr>
<tr>
<td>Specify the size of the font for tables, table cells, and graphs</td>
<td>“FONTSIZE=dimension</td>
</tr>
<tr>
<td>Specify the style of the font for tables, table cells, and graphs</td>
<td>“FONTSTYLE=ITALIC</td>
</tr>
<tr>
<td>Specify the font weight of tables, table cells, and graphs</td>
<td>“FONTWEIGHT=weight”</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Specify the font width of tables, table cells, and graphs compared</td>
<td>“FONTWIDTH=relative-width”</td>
</tr>
<tr>
<td>to the width of the usual design of the table, table cell, or graph</td>
<td></td>
</tr>
<tr>
<td>Specify the height of a table cell, graph, or graphics in an HTML</td>
<td>“HEIGHT=dimension ”</td>
</tr>
<tr>
<td>document 1</td>
<td></td>
</tr>
<tr>
<td>Specify the image to appear in a graph</td>
<td>“IMAGE=external-file ?DESC=alternative-text”</td>
</tr>
<tr>
<td>Specify the start fill color for a graph</td>
<td>“STARTCOLOR=color”</td>
</tr>
<tr>
<td>Specify justification in tables, table cells, and graphs</td>
<td>“TEXTALIGN=alignment”</td>
</tr>
<tr>
<td>Specify a transparency level for graphs</td>
<td>“TRANSPARENCY=dimension”</td>
</tr>
<tr>
<td>Specify vertical justification</td>
<td>“VERTICALALIGN=BOTTOM</td>
</tr>
<tr>
<td>Specify the width of a table cell, table, line, or a graph</td>
<td>“WIDTH=dimension ”</td>
</tr>
</tbody>
</table>

### Table 12.13  Tables

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify how to handle leading spaces and line breaks in an HTML</td>
<td>“ASIS=ON</td>
</tr>
<tr>
<td>document</td>
<td></td>
</tr>
<tr>
<td>Specify the color of the background of tables, table cells, or graphs</td>
<td>“BACKGROUNDCOLOR=color”</td>
</tr>
<tr>
<td>Specify an image to use as the background</td>
<td>“BACKGROUNDIMAGE=&quot;string&quot;”</td>
</tr>
<tr>
<td>Specify the position of the background of the tables, table cells,</td>
<td>“BACKGROUNDPOSITION=position”</td>
</tr>
<tr>
<td>or graphs</td>
<td></td>
</tr>
<tr>
<td>Specify whether an image is repeated horizontally, vertically, both,</td>
<td>“BACKGROUNDREPEAT=option”</td>
</tr>
<tr>
<td>or not repeated</td>
<td></td>
</tr>
<tr>
<td>Specify the color of the border in a table or table cell if the</td>
<td>“BORDERCOLOR=color”</td>
</tr>
<tr>
<td>border is just one color</td>
<td></td>
</tr>
<tr>
<td>Specify the color of the bottom of the table</td>
<td>“BORDERBOTTOMCOLOR=color ”</td>
</tr>
<tr>
<td>Specify the line style of the bottom of the selected cell</td>
<td>“BORDERBOTTOMSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the bottom of the table</td>
<td>“BORDERBOTTOMWIDTH=dimension ”</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>Specify the amount of white space on each of the four sides of the</td>
<td>“CELLPADDING=dimension</td>
</tr>
<tr>
<td>content in a table cell</td>
<td></td>
</tr>
<tr>
<td>Specify the name of the style sheet class to use in an HTML document</td>
<td>“CLASS=&quot;string&quot;”</td>
</tr>
<tr>
<td>for the table or table cell</td>
<td></td>
</tr>
<tr>
<td>Specify the darker color to use in a that uses two colors to create</td>
<td>“BORDERCOLORDARK=color”</td>
</tr>
<tr>
<td>a three-dimensional effect</td>
<td></td>
</tr>
<tr>
<td>Specify the lighter color to use in a that uses two colors to create</td>
<td>“BORDERCOLORLIGHT=color”</td>
</tr>
<tr>
<td>a three-dimensional effect</td>
<td></td>
</tr>
<tr>
<td>Specify the window or frame in which to open the target of the link</td>
<td>“HREFTARGET=&quot;target&quot;”</td>
</tr>
<tr>
<td>Specify an ID for the table or table cell</td>
<td>“HTMLID=&quot;string&quot;”</td>
</tr>
<tr>
<td>Specify the color of the left of a table</td>
<td>“BORDERLEFTCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the left of the specified table cell</td>
<td>“BORDERLEFTSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the left of the table</td>
<td>“BORDERLEFTWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the color of the right of the table</td>
<td>“BORDERRIGHTCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the right of the selected cell</td>
<td>“BORDERRIGHTSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the right of the table</td>
<td>“BORDERRIGHTWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the thickness of the spacing between cells in a table</td>
<td>“BORDERSPACING=dimension”</td>
</tr>
<tr>
<td>Specify the style of one or more sides of a table</td>
<td>“BORDERSTYLE=line-style-1 &lt;, ..., line-</td>
</tr>
<tr>
<td>style-4 &gt;”</td>
<td></td>
</tr>
<tr>
<td>Specify the color of the top of the table</td>
<td>“BORDERTOPCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the top of the specified table cell</td>
<td>“BORDERTOPSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the top of the table</td>
<td>“BORDERTOPWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the width of the table border</td>
<td>“BORDERWIDTH=dimension”</td>
</tr>
</tbody>
</table>
**PowerPoint Destination**

The ODS destination for PowerPoint includes the ODS POWERPOINT Statement on page 649.

**Table 12.14  Output and Text**

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify an image to use as the background</td>
<td>“BACKGROUNDIMAGE=string”</td>
</tr>
<tr>
<td>Specify whether an image is repeated horizontally, vertically, both, or not repeated</td>
<td>“BACKGROUNDREPEAT=option”</td>
</tr>
<tr>
<td>Specify the font to use in table cells and graphs</td>
<td>“FONTFAMILY=string-1,…, string-n”</td>
</tr>
<tr>
<td>Specify the size of the font for tables, table cells, and graphs</td>
<td>“FONTSIZE=dimension</td>
</tr>
<tr>
<td>Specify the style of the font for tables, table cells, and graphs</td>
<td>“FONTSTYLE=ITALIC</td>
</tr>
<tr>
<td>Specify the font weight of tables, table cells, and graphs</td>
<td>“FONTWEIGHT=weight”</td>
</tr>
<tr>
<td>Specify vertical justification</td>
<td>“VERTICALALIGN=BOTTOM</td>
</tr>
</tbody>
</table>

**Table 12.15  Document**

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify how to handle leading spaces and line breaks in table cells and HTML documents</td>
<td>“ASIS=ON</td>
</tr>
<tr>
<td>Specify the string to use for the bullets in the contents file</td>
<td>“LISTSTYLETYPE=bullet-type”</td>
</tr>
<tr>
<td>Specify the left margin for the PowerPoint slide</td>
<td>“MARGINLEFT=dimension”</td>
</tr>
<tr>
<td>Specify the right margin for the PowerPoint slide</td>
<td>“MARGINRIGHT=dimension”</td>
</tr>
</tbody>
</table>
Table 12.16  Tables

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the color of the background of tables, table cells, or graphs</td>
<td>&quot;BACKGROUNDCOLOR=color&quot;</td>
</tr>
<tr>
<td>Specify an image to use as the background</td>
<td>&quot;BACKGROUNDIMAGE=&quot;string&quot;&quot;</td>
</tr>
<tr>
<td>Specify whether an image is repeated horizontally, vertically, both, or not repeated</td>
<td>&quot;BACKGROUNDREPEAT=option&quot;</td>
</tr>
<tr>
<td>Specify the color of the bottom border of the table</td>
<td>&quot;BORDERBOTTOMCOLOR=color&quot;</td>
</tr>
<tr>
<td>Specify the line style of the bottom border of the selected cell</td>
<td>&quot;BORDERBOTTOMSTYLE=line-style&quot;</td>
</tr>
<tr>
<td>Specify the width of the bottom border of the table</td>
<td>&quot;BORDERBOTTOMWIDTH=dimension&quot;</td>
</tr>
<tr>
<td>Specify the color of the border in a table or table cell if the border is just one color</td>
<td>&quot;BORDERCOLOR=color&quot;</td>
</tr>
<tr>
<td>Specify the color of the left border of a table</td>
<td>&quot;BORDERLEFTCOLOR=color&quot;</td>
</tr>
<tr>
<td>Specify the line style of the left border of the specified table cell</td>
<td>&quot;BORDERLEFTSTYLE=line-style&quot;</td>
</tr>
<tr>
<td>Specify the width of the left border of the table</td>
<td>&quot;BORDERLEFTWIDTH=dimension&quot;</td>
</tr>
<tr>
<td>Specify the color of the right border of the table</td>
<td>&quot;BORDERRIGHTCOLOR=color&quot;</td>
</tr>
<tr>
<td>Specify the line style of the right border of the selected cell</td>
<td>&quot;BORDERRIGHTSTYLE=line-style&quot;</td>
</tr>
<tr>
<td>Specify the width of the right border of the table</td>
<td>&quot;BORDERRIGHTWIDTH=dimension&quot;</td>
</tr>
<tr>
<td>Specify the color of the top border of the table</td>
<td>&quot;BORDERTOPCOLOR=color&quot;</td>
</tr>
<tr>
<td>Specify the line style of the top border of the specified table cell</td>
<td>&quot;BORDERTOPSTYLE=line-style&quot;</td>
</tr>
<tr>
<td>Specify the width of the top border of the table</td>
<td>&quot;BORDERTOPWIDTH=dimension&quot;</td>
</tr>
<tr>
<td>Specify the width of the border of the table</td>
<td>&quot;BORDERWIDTH=dimension&quot;</td>
</tr>
<tr>
<td>Specify the amount of white space on each of the four sides of the content in a table cell</td>
<td>&quot;CELLPADDING=dimension</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>Specify the color of the foreground in tables, table cells, or graphs, which is primarily the color of text</td>
<td>“COLOR=color”</td>
</tr>
<tr>
<td>Specify the text to show in a data tip for the table cell</td>
<td>“FLYOVER=&quot;string&quot;”</td>
</tr>
<tr>
<td>Specify the font to use in table cells and graphs</td>
<td>“FONTFAMILY=string-n”</td>
</tr>
<tr>
<td>Specify the size of the font for tables, table cells, and graphs</td>
<td>“FONTSIZE=dimension</td>
</tr>
<tr>
<td>Specify the style of the font for tables, table cells, and graphs</td>
<td>“FONTSTYLE=ITALIC</td>
</tr>
<tr>
<td>Specify the font weight of tables, table cells, and graphs</td>
<td>“FONTWEIGHT=weight”</td>
</tr>
<tr>
<td>Specify the type of frame to use on a table</td>
<td>“FRAME=frame-type”</td>
</tr>
<tr>
<td>Specify the amount of white space on the bottom of the content of the table cell</td>
<td>“PADDINGBOTTOM=dimension</td>
</tr>
<tr>
<td>Specify the amount of white space on the left side of the content of the table cell</td>
<td>“PADDINGLEFT=dimension</td>
</tr>
<tr>
<td>Specify the amount of white space on the right side of the content of the table cell</td>
<td>“PADDINGRIGHT=dimension</td>
</tr>
<tr>
<td>Specify the amount of white space on the top of the content of the table cell</td>
<td>“PADDINTOP=dimension</td>
</tr>
<tr>
<td>Specify justification in tables, table cells, and graphs</td>
<td>“TEXTALIGN=alignment”</td>
</tr>
<tr>
<td>Change the visual presentation of the text</td>
<td>“TEXTDECORATION=presentation–options”</td>
</tr>
<tr>
<td>Specify the number of spaces that the first line of output is indented</td>
<td>“TEXTINDENT=n”</td>
</tr>
<tr>
<td>Specify a URL to link to from various reporting elements, including table cells in individual cells</td>
<td>“URL=&quot;uniform-resource-locator&quot;”</td>
</tr>
<tr>
<td>Specify vertical justification in individual cells</td>
<td>“VERTICALALIGN=BOTTOM</td>
</tr>
</tbody>
</table>
Printer Family of Destinations

The printer family of destinations includes the following ODS statements:

- ODS PRINTER Statement on page 722
- ODS PS Statement on page 749
- ODS PCL Statement on page 587
- ODS PDF Statement on page 597

Table 12.17  Output and Text

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the color of the background of tables, table cells, or graphs</td>
<td>“BACKGROUND_COLOR=color”</td>
</tr>
<tr>
<td>Specify an image to use as the background</td>
<td>“BACKGROUND_IMAGE=string”</td>
</tr>
<tr>
<td>Specify the position of the background of the tables, table cells, or graphs</td>
<td>“BACKGROUNDPOSITION=position”</td>
</tr>
<tr>
<td>Specify the color of the foreground in tables, table cells, or graphs, which is primarily the color of text</td>
<td>“COLOR=color”</td>
</tr>
<tr>
<td>Specify the alternate colors for maps</td>
<td>“CONTRAST_COLOR=color”</td>
</tr>
<tr>
<td>Specify a font definition to use in tables, table cells, and graphs</td>
<td>“FONT=font-definition”</td>
</tr>
<tr>
<td>Specify the font to use in table cells and graphs</td>
<td>“FONT_FAMILY=&quot;string-1&lt;…, string-n&gt;&quot;”</td>
</tr>
<tr>
<td>Specify the size of the font for tables, table cells, and graphs</td>
<td>“FONT_SIZE=dimension</td>
</tr>
<tr>
<td>Specify the style of the font for tables, table cells, and graphs</td>
<td>“FONT_STYLE=ITALIC</td>
</tr>
<tr>
<td>Specify the font weight of tables, table cells, and graphs</td>
<td>“FONT_WEIGHT=weight”</td>
</tr>
<tr>
<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>“FONT_WIDTH=relative-width”</td>
</tr>
<tr>
<td>Specify the height of a table cell, graph, or graphics in an HTML document</td>
<td>“HEIGHT=dimension ”</td>
</tr>
<tr>
<td>Specify the image to appear in a graph</td>
<td>“IMAGE=external-file ?DESC=alternative-text”</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Specify justification in tables, table cells, and graphs</td>
<td>“TEXTALIGN=alignment”</td>
</tr>
<tr>
<td>Specify vertical justification</td>
<td>“VERTICALALIGN=BOTTOM</td>
</tr>
</tbody>
</table>

**Table 12.18  Document**

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<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>“FILLRULEWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the color for the links in an HTML document that have not yet been visited</td>
<td>“LINKCOLOR=color”</td>
</tr>
<tr>
<td>Specify the string to use for the bullets in the contents file</td>
<td>“LISTSTYLETYPE=bullet-type”</td>
</tr>
<tr>
<td>Specify the bottom margin for the HTML document</td>
<td>“MARGINBOTTOM=dimension”</td>
</tr>
<tr>
<td>Specify the left margin for the HTML document</td>
<td>“MARGINLEFT=dimension”</td>
</tr>
<tr>
<td>Specify the right margin for the HTML document</td>
<td>“MARGINRIGHT=dimension”</td>
</tr>
<tr>
<td>Specify the top margin for the HTML document</td>
<td>“MARGINTOP=dimension”</td>
</tr>
</tbody>
</table>

**Table 12.19  Tables**

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify how to handle leading spaces and line breaks in an HTML document</td>
<td>“ASIS=ON</td>
</tr>
<tr>
<td>Specify the color of the background of tables, table cells, or graphs</td>
<td>“BACKGROUNDCOLOR=color”</td>
</tr>
<tr>
<td>Specify an image to use as the background</td>
<td>“BACKGROUNDIMAGE=&quot;string&quot;”</td>
</tr>
<tr>
<td>Specify the position of the background of the tables, table cells, or graphs</td>
<td>“BACKGROUNDPOSITION=position”</td>
</tr>
<tr>
<td>Specify the color of the bottom of the table</td>
<td>“BORDERBOTTOMCOLOR=color  ”</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>Specify the line style of the bottom of the selected cell</td>
<td>“BORDERBOTTOMSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the bottom of the table</td>
<td>“BORDERBOTTOMWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the amount of white space on each of the four sides of the</td>
<td>“CELLPADDING=dimension</td>
</tr>
<tr>
<td>content in a table cell</td>
<td></td>
</tr>
<tr>
<td>Specify the color of the foreground in tables, table cells, or</td>
<td>“COLOR=color”</td>
</tr>
<tr>
<td>graphs, which is primarily the color of text</td>
<td></td>
</tr>
<tr>
<td>Specify the color of the border in a table or table cell if the</td>
<td>“BORDERCOLOR=color”</td>
</tr>
<tr>
<td>border is just one color</td>
<td></td>
</tr>
<tr>
<td>Specify the text to show in a data tip for the table cell</td>
<td>“FLYOVER=”string”</td>
</tr>
<tr>
<td>Specify a font definition to use in tables, table cells, and graphs</td>
<td>“FONT=font-definition”</td>
</tr>
<tr>
<td>Specify the font to use in table cells and graphs</td>
<td>“FONTFAMILY=&quot;string-1&lt;…, string-n&gt;&quot;”</td>
</tr>
<tr>
<td>Specify the size of the font for tables, table cells, and graphs</td>
<td>“FONTSIZE=dimension</td>
</tr>
<tr>
<td>Specify the style of the font for tables, table cells, and graphs</td>
<td>“FONTSTYLE=ITALIC</td>
</tr>
<tr>
<td>Specify the font weight of tables, table cells, and graphs</td>
<td>“FONTWEIGHT=weight”</td>
</tr>
<tr>
<td>Specify the font width of tables, table cells, and graphs compared</td>
<td>“FONTWIDTH=relative-width”</td>
</tr>
<tr>
<td>to the width of the usual design of the table, table cell, or graph</td>
<td></td>
</tr>
<tr>
<td>Specify the type of frame to use on a table</td>
<td>“FRAME=frame-type”</td>
</tr>
<tr>
<td>Specify the height of a table cell, graph, or graphics in an HTML</td>
<td>“HEIGHT=dimension”</td>
</tr>
<tr>
<td>document ¹</td>
<td></td>
</tr>
<tr>
<td>Specify how to handle space characters in table cells</td>
<td>“NOBREAKSPACE=ON</td>
</tr>
<tr>
<td>Specify the amount of white space on the bottom of the content of</td>
<td>“PADDINGBOTTOM=dimension</td>
</tr>
<tr>
<td>the table cell</td>
<td></td>
</tr>
<tr>
<td>Specify the amount of white space on the left side of the content of</td>
<td>“PADDINGLEFT=dimension</td>
</tr>
<tr>
<td>the table cell</td>
<td></td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Specify the amount of white space on the right side of the content of the table cell</td>
<td>“PADDINGRIGHT=dimension</td>
</tr>
<tr>
<td>Specify the amount of white space on the top of the content of the table cell</td>
<td>“PADDINGTOP=dimension</td>
</tr>
<tr>
<td>Specify text to place after the table cell or table</td>
<td>“POSTTEXT=”string””</td>
</tr>
<tr>
<td>Specify an image to place before the table or table cell</td>
<td>“PREIMAGE=”external-file”</td>
</tr>
<tr>
<td>Specify text to place before the table cell or table</td>
<td>“PRETEXT=”string””</td>
</tr>
<tr>
<td>Specify how less-than signs (&lt;), greater-than signs (&gt;), and ampersands (&amp;) are interpreted in table cells</td>
<td>“PROTECTSPECIALCHARS=ON</td>
</tr>
<tr>
<td>Specify the width of the right of the table</td>
<td>“BORDERRIGHTWIDTH=dimension ”</td>
</tr>
<tr>
<td>Specify the types of rules to use in tables</td>
<td>“RULES=rule-type”</td>
</tr>
<tr>
<td>Specify the thickness of the spacing between cells in a table</td>
<td>“BORDERSPACING=dimension”</td>
</tr>
<tr>
<td>Specify justification in tables, table cells, and graphs</td>
<td>“TEXTALIGN=alignment”</td>
</tr>
<tr>
<td>Change the visual presentation of the text</td>
<td>“TEXTDECORATION=presentation–options”</td>
</tr>
<tr>
<td>Specify the number of spaces that the first line of output is indented</td>
<td>“TEXTINDENT=n”</td>
</tr>
<tr>
<td>Specify the color of the top of the table</td>
<td>“BORDERTOPCOLOR=color”</td>
</tr>
<tr>
<td>Specify the width of the top of the table</td>
<td>“BORDERTOPWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify a URL to link to from various reporting elements, including table cells</td>
<td>“URL=”uniform-resource-locator””</td>
</tr>
<tr>
<td>Specify vertical justification</td>
<td>“VERTICALALIGN=BOTTOM</td>
</tr>
<tr>
<td>Specify the width of the table border</td>
<td>“BORDERWIDTH=dimension”</td>
</tr>
</tbody>
</table>

**RTF Family of Destinations**

The RTF family of destinations includes the following statements:
<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the color of the background of tables, table cells, or graphs</td>
<td>“BACKGROUNDCOLOR=color”</td>
</tr>
<tr>
<td>Specify an image to use as the background</td>
<td>“BACKGROUNDIMAGE=“string””</td>
</tr>
<tr>
<td>Specify the position of the background of the tables, table cells, or graphs</td>
<td>“BACKGROUNDPOSITION=position”</td>
</tr>
<tr>
<td>Specify the color of the foreground in tables, table cells, or graphs, which is primarily the color of text</td>
<td>“COLOR=color”</td>
</tr>
<tr>
<td>Specify the alternate colors for maps</td>
<td>“CONTRASTCOLOR=color”</td>
</tr>
<tr>
<td>Specify a font definition to use in tables, table cells, and graphs</td>
<td>“FONT=font-definition”</td>
</tr>
<tr>
<td>Specify the font to use in table cells and graphs</td>
<td>“FONTFAMILY=“string-1&lt;…, string-n&gt;””</td>
</tr>
<tr>
<td>Specify the size of the font for tables, table cells, and graphs</td>
<td>“FONTSIZE=dimension</td>
</tr>
<tr>
<td>Specify the style of the font for tables, table cells, and graphs</td>
<td>“FONTSTYLE=ITALIC</td>
</tr>
<tr>
<td>Specify the font weight of tables, table cells, and graphs</td>
<td>“FONTWEIGHT=weight”</td>
</tr>
<tr>
<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>“FONTWIDTH=relative-width”</td>
</tr>
<tr>
<td>Specify the image to appear in a graph</td>
<td>“IMAGE=“external-file” ?DESC=alternative-text”</td>
</tr>
<tr>
<td>Specify justification in tables, table cells, and graphs</td>
<td>“TEXTALIGN=alignment”</td>
</tr>
<tr>
<td>Specify vertical justification</td>
<td>“VERTICALALIGN=BOTTOM</td>
</tr>
</tbody>
</table>
### Table 12.21  Document

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the color for the links in an HTML document that have not yet been visited</td>
<td>“LINKCOLOR=color”</td>
</tr>
<tr>
<td>Specify the bottom margin for the HTML document</td>
<td>“MARGINBOTTOM=dimension”</td>
</tr>
<tr>
<td>Specify the left margin for the HTML document</td>
<td>“MARGINLEFT=dimension”</td>
</tr>
<tr>
<td>Specify the right margin for the HTML document</td>
<td>“MARGINRIGHT=dimension”</td>
</tr>
<tr>
<td>Specify the top margin for the HTML document</td>
<td>“MARGINTOP=dimension”</td>
</tr>
</tbody>
</table>

### Table 12.22  Tables

<table>
<thead>
<tr>
<th>Task</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify how to handle leading spaces and line breaks in an HTML document</td>
<td>“ASIS=ON</td>
</tr>
<tr>
<td>Specify the color of the background of tables, table cells, or graphs</td>
<td>“BACKGROUNDCOLOR=color”</td>
</tr>
<tr>
<td>Specify an image to use as the background</td>
<td>“BACKGROUNDIMAGE=”string”</td>
</tr>
<tr>
<td>Specify the position of the background of the tables, table cells, or graphs</td>
<td>“BACKGROUNDPOSITION=position”</td>
</tr>
<tr>
<td>Specify the color of the bottom border of the table</td>
<td>“BORDERBOTTOMCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the bottom border of the selected cell</td>
<td>“BORDERBOTTOMSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the bottom border of the table</td>
<td>“BORDERBOTTOMWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify whether the border is collapsed or separated</td>
<td>“BORDERCOLLAPSE=COLLAPSE</td>
</tr>
<tr>
<td>Specify the color of the border in a table or table cell if the border is just one color</td>
<td>“BORDERCOLOR=color”</td>
</tr>
<tr>
<td>Specify the color of the left border of a table</td>
<td>“BORDERLEFTCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the left border of the specified table cell</td>
<td>“BORDERLEFTSTYLE=line-style”</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Specify the width of the left border of the table</td>
<td>“BORDERLEFTWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the color of the right border of the table</td>
<td>“BORDERRIGHTCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the right border of the selected cell</td>
<td>“BORDERRIGHTSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the right border of the table</td>
<td>“BORDERRIGHTWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the thickness of the spacing between cells in a table</td>
<td>“BORDERSPACING=dimension”</td>
</tr>
<tr>
<td>Specify the border style of one or more borders</td>
<td>“BORDERSTYLE=line-style-1 &lt;, ..., line-style-4 &gt;”</td>
</tr>
<tr>
<td>Specify the color of the top border of the table</td>
<td>“BORDERTOPCOLOR=color”</td>
</tr>
<tr>
<td>Specify the line style of the top border of the specified table cell</td>
<td>“BORDERTOPSTYLE=line-style”</td>
</tr>
<tr>
<td>Specify the width of the top border of the table</td>
<td>“BORDERTOPWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the width of the border of the table</td>
<td>“BORDERWIDTH=dimension”</td>
</tr>
<tr>
<td>Specify the amount of white space on each of the four sides of the content in a table cell</td>
<td>“CELLPADDING=dimension</td>
</tr>
<tr>
<td>Specify the color of the foreground in tables, table cells, or graphs, which is primarily the color of text</td>
<td>“COLOR=color”</td>
</tr>
<tr>
<td>Specify a font definition to use in tables, table cells, and graphs</td>
<td>“FONT=font-definition”</td>
</tr>
<tr>
<td>Specify the font to use in table cells and graphs</td>
<td>“FONTFAMILY=&quot;string-1&lt;…, string-n&gt;&quot;”</td>
</tr>
<tr>
<td>Specify the size of the font for tables, table cells, and graphs</td>
<td>“FONTSIZE=dimension</td>
</tr>
<tr>
<td>Specify the style of the font for tables, table cells, and graphs</td>
<td>“FONTSTYLE=ITALIC</td>
</tr>
<tr>
<td>Specify the font weight of tables, table cells, and graphs</td>
<td>“FONTWEIGHT=weight”</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<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>“FONTWIDTH=relative-width”</td>
</tr>
<tr>
<td>Specify the type of frame to use on a table</td>
<td>“FRAME=frame-type”</td>
</tr>
<tr>
<td>Specify the height of a table cell, graph, or graphics in an HTML document</td>
<td>“HEIGHT=dimension”</td>
</tr>
<tr>
<td>Specify how to handle space characters in table cells</td>
<td>“NOBREAKSPACE=ON</td>
</tr>
<tr>
<td>Specify the amount of white space between the content of the table cell and the border</td>
<td>“PADDING=dimension</td>
</tr>
<tr>
<td>Specify the amount of white space on the bottom of the content of the table cell</td>
<td>“PADDINGBOTTOM=dimension</td>
</tr>
<tr>
<td>Specify the amount of white space on the left side of the content of the table cell</td>
<td>“PADDINGLEFT=dimension</td>
</tr>
<tr>
<td>Specify the amount of white space on the right side of the content of the table cell</td>
<td>“PADDINGRIGHT=dimension</td>
</tr>
<tr>
<td>Specify the amount of white space on the top of the content of the table cell</td>
<td>“PADDINGTOP=dimension</td>
</tr>
<tr>
<td>Specify text to place after the table cell or table</td>
<td>“POSTTEXT=&quot;string&quot;”</td>
</tr>
<tr>
<td>Specify an image to place before the table or table cell</td>
<td>“PREIMAGE=&quot;external-file&quot;</td>
</tr>
<tr>
<td>Specify text to place before the table cell or table</td>
<td>“PRETEXT=&quot;string””</td>
</tr>
<tr>
<td>Specify how less-than signs (&lt;), greater-than signs (&gt;) , and ampersands (&amp;) are interpreted in table cells</td>
<td>“PROTECTSPECIALCHARS=ON</td>
</tr>
<tr>
<td>Specify the types of rules to use in tables</td>
<td>“RULES=rule-type”</td>
</tr>
<tr>
<td>Specify justification in tables, table cells, and graphs</td>
<td>“TEXTALIGN=alignment”</td>
</tr>
<tr>
<td>Change the visual presentation of the text</td>
<td>“TEXTDECORATION=presentation–options”</td>
</tr>
<tr>
<td>Specify the number of spaces that the first line of output is indented</td>
<td>“TEXTINDENT=n”</td>
</tr>
<tr>
<td>Task</td>
<td>Attribute</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Specify a URL to link to from various reporting elements, including</td>
<td>“URL=uniform-resource-locator”</td>
</tr>
<tr>
<td>table cells</td>
<td></td>
</tr>
<tr>
<td>Specify vertical justification</td>
<td>“VERTICALALIGN=BOTTOM</td>
</tr>
<tr>
<td>Specify the width of a table cell, table, line, or a graph</td>
<td>“WIDTH=dimension”</td>
</tr>
</tbody>
</table>

### Style Attributes Detailed Information

The following information can also be found in the Style Attributes Tables on page 1074. The Style Attributes Tables organize the style attributes by ODS destination and affected reporting elements. Many style attributes are valid in multiple destinations, so style attributes are repeated in those cases.

**ABSTRACT=ON | OFF**

- Specifies whether styles used in an HTML document are used in CSS style files.
  - **ON** specifies that styles are not used in CSS style files.
  - **OFF** specifies that styles are used in CSS style files.

  **Valid in** Markup family destinations, ODS HTML5 destination, and the ODS EPUB destination

**ACTIVELINKCOLOR=color**

- Specifies the color that a link in an HTML document changes to after you click it, but before the browser opens that file.

  **Valid in** Markup family destinations

  **See** color style attribute value on page 1132

**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.

  **Valid in** Markup family, printer family, PowerPoint, Excel, and RTF destinations

  **Default** OFF

  **Tip** Specify the ASIS= style attribute within the DATA style element to prevent the compressing of blank characters in table cells.
**BACKGROUNDCOLOR=**<i>color</i>
specifies the color of the background of the tables, table cells, or graphs.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>Markup family, printer family, Excel, and RTF destinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>BACKGROUND=</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 1104).</td>
</tr>
<tr>
<td>See</td>
<td>color style attribute value on page 1132</td>
</tr>
</tbody>
</table>

**BACKGROUNDIMAGE=**"<i>string</i>"
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.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>Markup family, Excel, printer family, TAGSETS.RTF, and PowerPoint destinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interactions</td>
<td>The BACKGROUNDIMAGE= attribute is overridden by the IBACK= and IMAGESTYLE=FIT options in the SAS/GRAPH GOPTIONS statement.</td>
</tr>
<tr>
<td>When you apply BACKGROUNDIMAGE= to a cell using the Report Writing Interface, the image adjusts to the size of the cell.</td>
<td></td>
</tr>
<tr>
<td>See</td>
<td>string attribute value on page 1137</td>
</tr>
</tbody>
</table>

**BACKGROUNDPOSITION=**<i>position</i>
specifies the position of the background of the tables, table cells, or graphs.

<i>position</i> can be one of the following:

- BOTTOM
- BOTTOM_CENTER
- BOTTOM_LEFT
- BOTTOM_RIGHT
- CENTER
- CENTER_BOTTOM
- CENTER_CENTER
Valid in  Markup family, printer family, and RTF destinations

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.

Valid in  Markup family and PowerPoint destinations

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.
Valid in Markup family destinations

Tip Typically, BODYSCROLLBAR= is set to AUTO.

**BODYSIZE=** *dimension | dimension% | * specifies the width of the frame that displays the body file in the HTML frame file.

*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.

Valid in Markup family destinations

Tip If *dimension* is a nonnegative number, then the unit of measure is pixels.

See dimension attribute value on page 1134

For information about the HTML files that ODS creates, see “HTML Links and References Produced by the HTML Destination ” on page 1201.

**BORDERBOTTOMCOLOR=** *color*
specifies the color of the bottom border of a table or table cell.

Valid in Markup family, printer family, PowerPoint, Excel, RTF, and Measured RTF destinations

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 1132

**BORDERBOTTOMSTYLE=** *line-style*
specifies the line style of the bottom border of the specified table cell.

*line-style* can be one of the following:

- DASHED
- DOTTED
- DOUBLE
- GROOVE
- HIDDEN
- INSET
- OUTSET
- RIDGE
- SOLID

Valid in Markup family, PowerPoint, Excel, RTF, and Measured RTF destinations

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.

For the ODS destination for Excel, when you specify the following dimensions, the thickness of the border appears as follows:

- Less than 3pt: Border width is thin.
- 3pt, but less than 5pt: Border width is medium.
- 5pt or more: Border width is thick.

Valid in: Markup family, Excel, PowerPoint, printer family, RTF, and Measured RTF destinations

See: dimension attribute value on page 1134

Example: “Example 5: Using BORDERBOTTOMCOLOR with Excel Output” on page 331

**BORDERCOLLAPSE=** COLLAPSE | SEPARATE

specifies whether the border is collapsed or separated.

Valid in: HTML5, HTML4, and EPUB destinations

Default: SEPARATE

**BORDERCOLOR=** *color*

specifies the border color of a table or table cell. The color is applied to all four borders.

Valid in: Markup family, Excel, PowerPoint, printer family, RTF, and Measured RTF destinations

See: color style attribute value on page 1132

**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.

Valid in: Markup family 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 1132

Example: “Defining a Table and Graph Style” in SAS Output Delivery System: Procedures Guide

**BORDERCOLORLIGHT=** *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.

Valid in: Markup family 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 1132

Example  “Defining a Table and Graph Style” in *SAS Output Delivery System: Procedures Guide*

**BORDERLEFTCOLOR=color**

specifies the color of the left border of the table.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>Markup family, Excel, PowerPoint, printer family, RTF, and Measured RTF destinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tip</td>
<td>You might also need to specify the BORDERLEFTWIDTH= attribute to override the style in the ODS destination.</td>
</tr>
<tr>
<td>See</td>
<td>color style attribute value on page 1132</td>
</tr>
</tbody>
</table>

**BORDERLEFTSTYLE=line-style**

specifies the line style of the left border of the specified table cell.

- **line-style**
  - can be one of the following:
  - DASHED
  - DOTTED
  - DOUBLE
  - GROOVE
  - HIDDEN
  - INSET
  - OUTSET
  - RIDGE
  - SOLID

<table>
<thead>
<tr>
<th>Valid in</th>
<th>Markup family, Excel, PowerPoint, RTF, and Measured RTF destinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tip</td>
<td>You might also need to specify the BORDERLEFTWIDTH= attribute to override the style in the ODS destination.</td>
</tr>
</tbody>
</table>

**BORDERLEFTWIDTH=dimension**

specifies the width of the left border of a table or table cell.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>Markup family, Excel, PowerPoint, printer family, RTF, and Measured RTF destinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>See</td>
<td>dimension attribute value on page 1134</td>
</tr>
</tbody>
</table>

**BORDERRIGHTCOLOR=color**

specifies the color of the right border of a table or table cell.

<table>
<thead>
<tr>
<th>Valid in</th>
<th>Markup family, Excel, PowerPoint, printer family, RTF, and Measured RTF destinations</th>
</tr>
</thead>
</table>
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 1132

**BORDERRIGHTSTYLE=**<i>line-style</i>

specifies the line style of the right border of the selected cell.

<i>line-style</i>

can be one of the following:

- DASHED
- DOTTED
- DOUBLE
- GROOVE
- HIDDEN
- INSET
- OUTSET
- RIDGE
- SOLID

Valid in  Markup family, Excel, PowerPoint, RTF, and Measured RTF destinations

Tip  You might also need to specify the BORDERRIGHTWIDTH= attribute to override the style in the ODS destination.

**BORDERRIGHTWIDTH=**<i>dimension</i>

specifies the width of the right border of the table.

Valid in  Markup family, Excel, PowerPoint, printer family, RTF, and Measured RTF destinations

See  dimension attribute value on page 1134

**BORDERSPACING=**<i>dimension</i>

specifies the vertical and horizontal thickness of the spacing between cells in a table.

Valid in  Markup family, RTF, and printer family destinations

Alias  CELLSPACING=

Default  0

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 1134

Examples  

“Modifying the Default Style with the CLASS Statement” in *SAS Output Delivery System: Procedures Guide*
BORDERSTYLE=\texttt{line-style-1 <,..., line-style-4 >}

specifies the border style of one or more borders.

\textit{line-style}

is one of the following:

- DASHED
- DOTTED
- DOUBLE
- GROOVE
- HIDDEN
- INSET
- NONE
- OUTSET
- RIDGE
- SOLID

The order in which you specify the style determines the style applied to each side.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
\textbf{Example} & \textbf{Order} & \textbf{Result} \\
\hline
\texttt{borderstyle= double;} & All four sides & Double \\
\hline
\texttt{borderstyle= solid double;} & Top and bottom & Solid \\
& Right and left & Double \\
\hline
\texttt{borderstyle= none double solid;} & Top & None \\
& Right and left & Double \\
& Bottom & Solid \\
\hline
\texttt{borderstyle= none double solid none;} & Top & None \\
& Right & Double \\
& Bottom & Solid \\
& Left & None \\
\hline
\end{tabular}
\caption{BORDERSTYLE= Option Precedence}
\end{table}

Valid in Markup family, Excel, RTF, and Measured RTF destinations

BORDERTOPCOLOR=\textcolor{color}

specifies the color of the top border of a table or table cell.

Valid in Markup family, printer family, RTF, PowerPoint, Excel, and Measured RTF destinations

Tip Specify the BORDERTOPWIDTH= attribute to override the style in the ODS destination.

See color style attribute value on page 1132
**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**

**Valid in**  
Markup family, Excel, PowerPoint, RTF, and Measured RTF destinations

**Restriction**  
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.

**Valid in**  
Markup family, Excel, PowerPoint, printer family, RTF, and Measured RTF destinations

**See**  
dimension attribute value on page 1134

**BORDERWIDTH=** *dimension*

specifies the width of the table borders. The value of BORDERWIDTH= is applied to all four borders.

**Valid in**  
Markup family, Excel, PowerPoint, RTF, and printer family destinations

**Tip**  
Typically, when BORDERWIDTH=0, the ODS destination sets RULES=NONE (see the discussion about “RULES=rule-type” on page 1127) and FRAME=VOID (see the discussion about “FRAME=frametype” on page 1115).

**See**  
dimension attribute value on page 1134

**Examples**

“Modifying the Default Style with the CLASS Statement” in *SAS Output Delivery System: Procedures Guide*
CAPSTYLE=\texttt{line-shape}

specifies the shape of the line at the end of a box whisker. \texttt{line-shape} can be one of the following:

- "BRACKET"
- "LINE"
- "NONE"
- "SERIF"

Requirement You must enclose \texttt{line-shape} in quotation marks.

CELLPADDING=\texttt{dimension | dimension\%}

specifies the amount of white space on each of the four sides of the content in a table cell.

\texttt{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.

Valid in Markup family other than HTML5, PowerPoint, printer family, and the RTF destinations

Restrictions CELLPADDING= is not valid in the HTML5 destination. All padding is done on the table cells.

In the ODS Destination for PowerPoint, the cellpadding dimension cannot go below 5 pt.

See dimension attribute value on page 1134

Example “Modifying the Default Style with the CLASS Statement” in \textit{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.

Valid in Markup family

Alias HTMLCLASS=

See string attribute value on page 1137

COLOR=color

specifies the color of the foreground in tables, table cells, or graphs, which is primarily the color of text.

Valid in Markup family, printer family, PowerPoint, and RTF destinations

Alias FOREGROUND=

Interaction The COLOR= attribute is overridden by the CBACK= option in the SAS/GRAPH GOPTIONS statement.

See color style attribute value on page 1132

Examples
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.

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 1201.)

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.

Valid in Markup family destinations

Tip Typically, CONTENTSCROLLBAR= is set to AUTO.

See For information about the HTML files that ODS creates, see “HTML Links and References Produced by the HTML Destination ” on page 1201.

CONTENT=html-code
specifies the HTML code for page breaks.

html-code
is a string of html code that defines page breaks.

Valid in HTML5 and EPUB destinations

Example The following example is from Styles.Daisy:
```html
content = '<div class="pagebreak" style="text-align: center; page-break-before: avoid; page-break-after: always">&nbsp;</div>'
```
**CONTENTSIZES=** *dimension | dimension % | *  
specifies the width of the frames in the frame file that display the contents and the page files.

*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.

**Valid in** Markup family destinations

**Requirement**

*dimension % must be a positive number between 0 and 100.

**Tip**

If *dimension* is a nonnegative number, then the unit of measure is pixels.

**See**

dimension attribute value on page 1134

“BODYSIZE=*dimension | dimension% | * ” on page 1101

For information about the HTML files that ODS creates, see “HTML Links and References Produced by the HTML Destination ” on page 1201.

**CONTENTTYPE=** "*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.

*string*

is the content type for the pages.

**Requirement**

*string* must be enclosed in quotation marks.

**Tip**

The value of *string* is usually "text/html".

**See**

string attribute value on page 1137

**Valid in** Markup family destinations

**Alias**

HTMLCONTENTTYPE=

<table>
<thead>
<tr>
<th><strong>CONTRASTCOLOR=</strong> <em>color</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>specifies the alternate colors for maps. The alternate colors are applied to the blocks on region areas in block maps.</td>
</tr>
</tbody>
</table>

**Valid in** Markup family, RTF, and printer family destinations

**See**

color style attribute value on page 1132

**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
- box plots
- bubble plots
- drop lines
- high-low charts
- histograms
- line charts
- needle plots
- pie charts
- polygon plots
- reference lines
- scatter plots
- series plots
- step plots
- vector 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 12.24 DATASKIN Values

<table>
<thead>
<tr>
<th>DATASKIN Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td><img src="image" alt="NONE Example" /></td>
</tr>
<tr>
<td>CRISP</td>
<td><img src="image" alt="CRISP Example" /></td>
</tr>
<tr>
<td>GLOSS</td>
<td><img src="image" alt="GLOSS Example" /></td>
</tr>
<tr>
<td>MATTE</td>
<td><img src="image" alt="MATTE Example" /></td>
</tr>
<tr>
<td>PRESSED</td>
<td><img src="image" alt="PRESSED Example" /></td>
</tr>
<tr>
<td>SHEEN</td>
<td><img src="image" alt="SHEEN Example" /></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 1137

Valid in  Markup family destinations
Alias  HTMLDOCTYPE=

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 1132

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 1055 and “Style Elements Affecting Device-Based Graphics” on page 1064.

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.

Valid 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 1134

FLYOVER="string"
specifies the text to show in a data tip for various reporting elements, including table cells.

string is the text of the data tip.

Restriction When using the ODS PDF destination, the FLYOVER= style attribute text should not exceed 1024 chars. If the length exceeds 1024, the text is ignored and a warning message is recorded in the SAS log.

Requirement string must be enclosed in quotation marks.

See string attribute value on page 1137
Valid in Markup family, Excel, PowerPoint, and PDF destinations

**FONT=**font-definition

specifies a font definition to use in tables, table cells, and graphs.

Valid in Markup family, RTF, and printer family destinations

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 specify only one font.

See [font-definition attribute value on page 1134](#)

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 1137](#)

Valid in Markup family, Excel, PowerPoint, RTF, and printer family destinations

Alias **FONT_FACE=**

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=**dimension | 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 1134

**size**

The value of *size* is relative to all other font sizes in the HTML document.

| Range | 1 to 7 |

Valid in Markup family, Excel, PowerPoint, RTF, and printer family destinations

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.

Valid in Excel, Markup family, PowerPoint, RTF, and printer family destinations

**Alias** FONT_STYLE=

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

“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.

Valid in Markup family, Excel, PowerPoint, RTF, and printer family destinations

**Alias** FONT_WEIGHT=

Restriction 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.

**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=** *relative-width*

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

**Valid in**
Markup family, RTF, and printer family destinations

**Alias**
FONT_WIDTH=

**Restriction**
Few fonts honor these values.

**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=** *frame-type*

specifies the type of frame to use on a table. This table shows the possible values for *frame-type* and their meanings:

**Table 12.25 Frame-type Values**

<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>Value for frame-type</td>
<td>Frame Type</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------</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>
<tr>
<td>VSIDES</td>
<td>Borders at the left and right sides</td>
</tr>
</tbody>
</table>

Valid in  Markup family, PowerPoint, RTF, and printer family destinations

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.

Valid in  Markup family destinations

**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 1134

Valid in  Markup family, RTF, and printer family destinations

Aliases  CELLHEIGHT=

OUTPUTHEIGHT=

Restriction  The HEIGHT= option does not apply to output generated as a result of GRSEG (graph segment) output.

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.
  
  **Restriction** 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.

**Valid in** Markup family destinations

**Default** _self

**Restriction** 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 Java Script.

**string**

is the ID text.

**Requirement** string must be enclosed in quotation marks.

**See** string attribute value on page 1137

**Valid 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 1137

Valid in  Markup family destinations

**IMAGE**="external-file" <?DESC=alternative-text>

specifies the image to appear in a graph. This image is positioned or tiled.

*external-file*

names a GIF or JPEG file. Use a simple filename, a complete path, or a URL.

Requirement  
_external-file must be enclosed in quotation marks.

<?DESC=alternative-text>

specifies alternative image text for the image. To change the alternative text for an image, append ?desc=mydesc to the image filename. This option helps to make reports accessible for people with a wide range of abilities. For accessibility, the text should convey the meaning of the image.

If an empty text string is specified, then the image is ignored by screen readers.

Valid in  PDF, EPUB, and HTML5 destinations

Alias  
DESC=

Accessibility note  
You can use the ACCESSIBLECHECK system option to write a note to the SAS log if no description is provided for an image.

Example  
image="questions.gif?desc=questionmark montage"

Valid in  Markup family, printer family, and RTF destinations

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:
Figure 12.1 Table of Line Patterns

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid</td>
<td>1</td>
</tr>
<tr>
<td>Short Dash</td>
<td>2</td>
</tr>
<tr>
<td>Medium Dash</td>
<td>4</td>
</tr>
<tr>
<td>Long Dash</td>
<td>5</td>
</tr>
<tr>
<td>Medium Dash Short Dash</td>
<td>8</td>
</tr>
<tr>
<td>Dash Dash Dot</td>
<td>14</td>
</tr>
<tr>
<td>Dash Dot Dot</td>
<td>15</td>
</tr>
<tr>
<td>Dash Dot</td>
<td>20</td>
</tr>
<tr>
<td>Long Dash Short Dash</td>
<td>26</td>
</tr>
<tr>
<td>Dot</td>
<td>34</td>
</tr>
<tr>
<td>Thin Dot</td>
<td>35</td>
</tr>
<tr>
<td>Short Dash Dot</td>
<td>41</td>
</tr>
<tr>
<td>Medium Dash Dot Dot</td>
<td>42</td>
</tr>
</tbody>
</table>

**LINETHICKNESS=** *dimension*

specifies the thickness of a line.

See dimension attribute value on page 1134

**LINKCOLOR=** *color*

specifies the color for the links in an HTML document that have not yet been visited.

Valid in Markup family, printer family, and RTF destinations

See color style attribute value on page 1132

**LISTENTRYANCHOR=** **ON | OFF**

in an HTML document, the LISTENTRYANCHOR= attribute specifies whether to make the entry in the table of contents a link to the body file.

**ON**

specifies to make this entry in the table of contents a link to the body file.

**OFF**

specifies not to make this entry in the table of contents a link to the body file.

Valid in Markup family destinations

**LISTSTYLEIMAGE=** *string*

specifies a path or URL to an image file to be used as the bullet for list items.

Valid in Printer family destinations

**LISTSTYLETYPE=** *bullet-type*

specifies the type of bullet to use for lists and the contents file. ODS uses bullets in the contents file.

*bullet-type*

The *bullet-type* value differs for PowerPoint and markup destinations.

The following values are valid in the markup destinations:

- ARMENIAN
- LOWER-GREEK
- CIRCLE
- LOWER-LATIN
- DECIMAL
- LOWER-ROMAN
The following values are valid in the PowerPoint destination:

- **ASTERISKS** - HYPHEN
- **BOX** - LOWER_ROMAN
- **CHECK** - PARENTHESISED_LOWER_LATIN
- **CIRCLE** - PARENTHESISED_DECIMAL
- **DECIMAL** - SQUARE
- **DIAMOND** - UPPER_ROMAN
- **DISC** - UPPER-ALPHA

Default: **DISC**

See [string attribute value on page 1137](#)

**MARGINBOTTOM=dimension**

specifies the bottom margin for the document.

Valid in: Markup family, printer family, and RTF destinations

Alias: **BOTTOMMARGIN=**

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 1134](#)

**MARGINLEFT=dimension**

specifies the left margin for the document.

Valid in: Markup family, PowerPoint, printer family, and RTF destinations

Alias: **LEFTMARGIN=**

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 1134](#)

**MARGINRIGHT=dimension**

specifies the right margin for the document.
Valid in Markup family, PowerPoint, printer family, and RTF destinations

**Alias** RIGHTMARGIN=

**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 1134

**MARGINTOP=** *dimension*

specifies the top margin for the document.

Valid in Markup family, printer family, and RTF destinations

**Alias** TOPMARGIN=

**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 1134

**MARKERSIZE=** *dimension*

specifies the marker size (both width and height).

See dimension attribute value on page 1134

**MARKERSYMBOL=** *marker-symbol*

specifies a marker symbol. *marker-symbol* can be one of the following:

*Figure 12.2 Table of Marker Symbols*

![Table of Marker Symbols]

**NEUTRALCOLOR=** *color*

specifies the middle color in a three-color ramp.

See color style attribute value on page 1132

**NOBREAKSPACE=** *ON | OFF*

specifies how to handle space characters in table cells.
ON does not let SAS break a line at a space character.

OFF lets SAS break a line at a space character if appropriate.

**Valid in** Markup family, Excel, printer family, and RTF destinations

**ORPHAN=integer**

specifies the minimum number of lines of text that must appear in a paragraph before it is forced to move to another page.

**Valid in** EPUB destination

**Default** 2

**OUTLINECOLOR=color**

specifies the color of the link-focus indicator outline.

**Valid in** HTML5 destination

**Note** The OUTLINECOLOR= attribute is valid starting with SAS 9.4M5.

**Tip** The OUTLINECOLOR= attribute is useful for making your HTML5 documents accessible. See Enhancing the Appearance of the Link-Focus Indicator in Creating Accessible SAS Output Using ODS and ODS Graphics.

**See** color style attribute value on page 1132

**OUTLINESTYLE=line-style**

specifies the line style of the link-focus indicator outline.

*line-style*

can be one of the following:

- DASHED
- DOTTED
- DOUBLE
- GROOVE
- HIDDEN
- INSET
- OUTSET
- RIDGE
- SOLID

**Valid in** HTML5 destination

**Note** The OUTLINESTYLE= attribute is valid starting with SAS 9.4M5.

**Tip** The OUTLINESTYLE= attribute is useful for making your HTML5 documents accessible. See Enhancing the Appearance of the Link-Focus Indicator in Creating Accessible SAS Output Using ODS and ODS Graphics.
OUTLINEWIDTH=dimension
specifies the line width of the link-focus indicator outline.

Valid in  HTML5 destination

Note  The OUTLINEWIDTH= attribute is valid starting with SAS 9.4M5.

Tip  The OUTLINEWIDTH= attribute is useful for making your HTML5 documents accessible. See Enhancing the Appearance of the Link-Focus Indicator in Creating Accessible SAS Output Using ODS and ODS Graphics.

See  dimension attribute value on page 1134

OUTPUTHEIGHT=dimension
specifies the height of a graph.

See  dimension attribute value on page 1134

OUTPUTWIDTH=dimension
specifies the width of a graph.

See  dimension attribute value on page 1134

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 1123
- PADDINGLEFT= on page 1123
- PADDINGRIGHT= on page 1124
- PADDINGTOP= on page 1124

Valid in  Markup family, RTF, and printer family destinations

See  dimension attribute value on page 1134

PADDINGBOTTOM=dimension | dimension%
specifies the amount of white space on the bottom of the content of the table cell.

Valid in  Markup family, PowerPoint, RTF, and printer family destinations

Default  0

See  dimension attribute value on page 1134

PADDINGLEFT=dimension | dimension%
specifies the amount of white space on the left side of the content of the table cell.

Valid in  Markup family, PowerPoint, RTF, and printer family destinations

Default  0

See  dimension attribute value on page 1134
PADDINGRIGHT=\textit{dimension} \mid \textit{dimension}\%  
specifies the amount of white space on the right side of the content of the table cell.

Valid in: Markup family, PowerPoint, RTF, and printer family destinations  
Default: 0  
See: dimension attribute value on page 1134

PADDINGTOP=\textit{dimension} \mid \textit{dimension}\%  
specifies the amount of white space on the top of the content of the table cell.

Valid in: Markup family, PowerPoint, RTF, and printer family destinations  
Default: 0  
See: dimension attribute value on page 1134

PAGEBREAKHTML=\textit{"string"}  
specifies HTML to place at page breaks in an HTML document.

\textit{string}  
is the HTML code used to place at page breaks.  
Requirement: \textit{string} must be enclosed in quotation marks.  
See: string attribute value on page 1137

Valid in: Markup family destinations

POSTHTML=\textit{"string"}  
specifies the HTML code to place after the table or table cell.

\textit{string}  
is the HTML code to place after a table or table cell.  
Requirement: \textit{string} must be enclosed in quotation marks.  
See: string attribute value on page 1137

Valid in: Markup family destinations

Example: “Modifying the Default Style with the CLASS Statement” in \textit{SAS Output Delivery System: Procedures Guide}

POSTIMAGE=\textit{external-file} \mid \textit{fileref} \langle ?DESC=alternative-text\rangle  
specifies an image to place after the table or table cell.

\textit{external-file}  
names a GIF or JPEG file. Use a simple filename, a complete path, or a URL.  
Requirement: \textit{external-file} must be enclosed in quotation marks.

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

<?DESC=alternative-text?>
specifies alternative image text for the image. To change the alternative text for an image, append ?desc=mydesc to the image filename. This option helps to make reports accessible for people with a wide range of abilities. For accessibility, the text should convey the meaning of the image.

If an empty text string is specified, then the image is ignored by screen readers.

Valid in PDF, EPUB, and HTML5 destinations

Alias DESC=

Accessibility note You can use the ACCESSIBLECHECK system option to write a note to the SAS log if no description is provided for an image.

Example image=questions.gif?desc=questionmark montage*

Valid in Markup family, printer family, PowerPoint, and RTF destinations

Restriction For the ODS destination for PowerPoint, the POSTIMAGE= attribute can be used to place an image below a table, but it cannot be used to put an image after a cell.

POSTTEXT="string"
specifies text to place after the table cell or table.

Requirement string must be enclosed in quotation marks.

See string attribute value on page 1137

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 1137

PREIMAGE="external-file" | fileref <?DESC=alternative-text?>
specifies an image to place before the table or table cell.

external-file
names a GIF or JPEG file. Use a simple filename, a complete path, or a URL.

Valid in Markup family, printer family, and RTF destinations

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 external-file in quotation marks.

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 DATA Step Statements: Reference.)
<p><?DESC=alternative-text?>
specifies alternative image text for the image. To change the alternative text for an image, append ?desc=mydesc to the image filename. This option helps to make reports accessible for people with a wide range of abilities. For accessibility, the text should convey the meaning of the image.

If an empty text string is specified, then the image is ignored by screen readers.

| Valid in | PDF, EPUB, and HTML5 destinations |
| Alias | DESC= |
| Accessibility note | You can use the ACCESSIBLECHECK system option to write a note to the SAS log if no description is provided for an image. |
| Example | image=*questions.gif?desc=questionmark montage* |

Valid in Markup family, printer family, and RTF destinations

Restriction For the ODS destination for PowerPoint, the PREIMAGE= attribute can be used to place an image above a table, but it cannot be used to put an image before a cell.

PRETEXT="string"
specifies text to place before the table cell or table.

| Valid in | Markup family, Excel, printer family, and RTF destinations |
| Requirement | Enclose string in quotation marks. |
| See | string attribute value on page 1137 |
| Example | “Example 3: Customizing the Table of Contents” on page 623 |

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.

OFF
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.

Valid in  Markup family, Excel, printer family, and RTF destinations

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>

Valid in  Markup family, PowerPoint, printer family, and RTF destinations

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.

Valid in  HTML4 destination

See  color style attribute value on page 1132

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 1137

---

**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;` The value of *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.

---

Valid in: Markup family, printer family, PowerPoint, Excel, and RTF destinations

**Alias** JUST=

**Restriction** 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.

Valid in Excel, HTML, RTF, PowerPoint, and Excel destinations

**LINE_THROUGH**
specifies that a line is drawn through the text.

Valid in HTML, printer family, measured RTF, and RTF destinations

**OVERLINE**
specifies that a line is drawn above the text.

Valid in HTML and printer family destinations

**UNDERLINE**
specifies that a line is drawn below the text.

Valid in HTML, printer family, measured RTF, and RTF destinations

**Tip**
TEXTDECORATION= can be used with inline formatting and the ODS PDF statement to enhance PDF files.

**Example**
“Formatting Cells with 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.

Valid in Markup family, printer family, Excel, PowerPoint, and RTF destinations

Alias INDENT=

Default The default value for XML is 2. For all other ODS destinations, the default value is 0.

**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.

Valid in  RTF and TAGSETS.RTF destinations

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).

Valid in  HTML destination

See  dimension attribute value on page 1134

URL="uniform-resource-locator"
specifies a URL to link to from various reporting elements, including table cells.

Valid in  Markup family, Excel, PowerPoint, printer family, and RTF destinations

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

Valid in  Markup family, Excel, printer family, and RTF destinations

Alias  VJUST=

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=**<i>color</i>

specifies the color for links that have been visited in an HTML document.

Valid in  Markup family destinations

See  color style attribute value on page 1132

**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.

Valid in  Markup family and RTF 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 1099

**WIDTH=**<i>dimension</i>

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 <i>dimension</i>, then the value will be in pixels. If a unit of measure other than pixels or percentage is specified with the <i>dimension</i>, then the HEIGHT=<i>dimension</i> is not applied to the graph.

<i>dimension</i>

is a nonnegative number.

Valid in  Markup family, Excel, printer family, and RTF destinations

Aliases  CELLWIDTH=

OUTPUTWIDTH=

Restriction  The WIDTH= option does not apply to output generated as a result of GRSEG (graph segment) output.

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.
WINDOW=integer

specifies the number of lines of text that must appear at the top of a page if a paragraph is separated by a page break.

Valid in EPUB destination

Default 2

---

**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, FFFFF00)

*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 12.27  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></td>
<td>Blue</td>
<td>Black&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
</tr>
<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>
<tr>
<td>Light</td>
<td>Vivid</td>
<td>Yellow</td>
<td></td>
</tr>
<tr>
<td>Very light</td>
<td></td>
<td>Green</td>
<td></td>
</tr>
</tbody>
</table>

<sup>1</sup> Black and white cannot be combined with a lightness value or a saturation value.

<sup>2</sup> 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:
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>
<thead>
<tr>
<th>Table 12.28 Units of Measure for Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
</tr>
<tr>
<td>cm</td>
</tr>
<tr>
<td>ex</td>
</tr>
<tr>
<td>in</td>
</tr>
<tr>
<td>mm</td>
</tr>
<tr>
<td>pt</td>
</tr>
</tbody>
</table>

Default For the PRINTERS 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)

A font specification in an ODS style can specify multiple fonts so that a reasonable substitution can be made when a font cannot be located on the current computer. The fonts are normally listed in a most-specific to most-generic order. For example, the TitleFont attribute shown in the following example specifies a list of fonts in the following order: <sans-serif>, <MTsans-serif>, Helvetica, and then Helv.

'TitleFont'="<sans-serif>,<MTsans-serif>,Helvetica,Helv",3,bold"

For table title text, SAS searches the computer for a font in the TitleFont font list in the order listed. The first font in the list that is found is used.
If a font is enclosed in angle brackets, that indicates that the resolved font is defined in the ODS\FONTS portion of the SAS registry. One way to view the SAS registry font definitions is with the DMS REGEDIT command.

"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.

   Note
   For SAS 9.4M5, ODS Styles supports the AvenirNextForSAS font.

   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 1134.

"keyword-list"
   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 12.29  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>
</tbody>
</table>
### Keywords for Font Weight

<table>
<thead>
<tr>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEMI_LIGHT*</td>
</tr>
<tr>
<td>EXTRA_LIGHT*</td>
</tr>
</tbody>
</table>

* Few fonts honor these values.

### Example

```
```

### 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 be specified only for data cells.
Appendix 1
  Output Object Table Names ........................................ 1141

Appendix 2
  Example Programs ....................................................... 1167

Appendix 3
  ODS and the HTML Destination ...................................... 1201

Appendix 4
  ODS HTML Statements for Running Examples in
  Different Operating Environments .................................. 1213

Appendix 5
  PowerPoint Transition Options and the Effect
  Options Supported .................................................... 1215
Appendix 1

Output Object Table Names

ODS Table Names Produced by Base SAS Procedures .......................... 1141
ODS Table Names Produced by Base SAS High-Performance Procedures ... 1155
ODS Table Names Produced by Base SAS Statistical Procedures ............. 1155
ODS Table Names Produced by SAS/STAT Procedures ........................ 1156
ODS Table Names Produced by SAS/STAT High-Performance Procedures ... 1160
ODS Table Names Produced by SAS Enterprise Miner High-Performance Procedures ................................................................. 1161
ODS Table Names Produced by SAS/ETS Procedures .......................... 1161
ODS Table Names Produced by SAS/ETS High-Performance Procedures .... 1163
ODS Table Names Produced by SAS/QC Procedures ............................ 1163
ODS Table Names Produced by SAS/OR Procedures ............................ 1164
ODS Table Names Produced by Graph Algorithms and Network Analysis Procedure ................................................................. 1165

ODS Table Names Produced by Base SAS Procedures

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 A1.5  ODS Table Names Produced by the CORR Procedure

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cov</td>
<td>Covariances</td>
<td>COV</td>
</tr>
<tr>
<td>CronbachAlpha</td>
<td>Coefficient alpha</td>
<td>ALPHA</td>
</tr>
<tr>
<td>CronbachAlphaDel</td>
<td>Coefficient alpha with deleted variable</td>
<td>ALPHA</td>
</tr>
<tr>
<td>Csscp</td>
<td>Corrected sums of squares and crossproducts</td>
<td>CSSCP</td>
</tr>
<tr>
<td>FisherPearsonCorr</td>
<td>Pearson correlation statistics using Fisher’s $z$ transformation</td>
<td>FISHER</td>
</tr>
<tr>
<td>FisherSpearmanCorr</td>
<td>Spearman correlation statistics using Fisher’s $z$ transformation</td>
<td>FISHER SPEARMAN</td>
</tr>
<tr>
<td>HoeffdingCorr</td>
<td>Hoeffding’s D statistics</td>
<td>HOEFFDING</td>
</tr>
<tr>
<td>KendallCorr</td>
<td>Kendall’s tau-b coefficients</td>
<td>KENDALL</td>
</tr>
<tr>
<td>PearsonCorr</td>
<td>Pearson correlations</td>
<td>PEARSON</td>
</tr>
<tr>
<td>PolychoricCorr</td>
<td>Polychoric correlations</td>
<td>POLYCHORIC</td>
</tr>
<tr>
<td>PolyserialCorr</td>
<td>Polyserial correlations</td>
<td>POLYSERIAL</td>
</tr>
<tr>
<td>SimpleStats</td>
<td>Simple descriptive statistics</td>
<td></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>
</tbody>
</table>
Table A1.7  ODS Table Names Produced by the CONTENTS Procedure or the DATASETS Procedure with the CONTENTS Statement

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<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=\langle libref.\rangle_ALL_ 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=\langle libref.\rangle_ALL_ 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=\langle libref.\rangle\_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</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</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</td>
</tr>
<tr>
<td>CommonOddsRatioTest</td>
<td>Common odds ratio exact test</td>
<td>EXACT</td>
<td>COMOR</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\text{-way table, } n &gt; 1))</td>
</tr>
<tr>
<td>CrossTabFreqs</td>
<td>Crosstabulation table</td>
<td>TABLES</td>
<td>((n\text{-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>JTTest</td>
<td>Jonckheere-Terpstra test</td>
<td>TABLES</td>
<td>JT</td>
</tr>
<tr>
<td>JTTestMC</td>
<td>Monte Carlo estimates for Jonckheere-Terpstra exact test</td>
<td>EXACT</td>
<td>JT / MC</td>
</tr>
<tr>
<td>Kappa</td>
<td>Simple kappa coefficient</td>
<td>TEST or EXACT</td>
<td>KAPPA</td>
</tr>
<tr>
<td></td>
<td></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 \times r) table, &gt; 2)</td>
</tr>
<tr>
<td>KappaTest</td>
<td>Simple kappa test</td>
<td>TEST or EXACT</td>
<td>KAPPA</td>
</tr>
<tr>
<td>KappaWeights</td>
<td>Kappa weights</td>
<td>TABLES</td>
<td>AGREE(PRINTKWTS)</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 \times 2 \times 2) table)</td>
</tr>
<tr>
<td>McNemarsTest</td>
<td>McNemar’s test</td>
<td>TABLES</td>
<td>AGREE (2 \times 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 \times 2 table)</td>
</tr>
<tr>
<td>OddsRatioExactCL</td>
<td>Exact confidence limits for the odds ratio</td>
<td>EXACT</td>
<td>OR (2 \times 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></td>
</tr>
<tr>
<td></td>
<td></td>
<td></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 or EXACT</td>
</tr>
<tr>
<td>SomersDRC</td>
<td>Somers’ $D(R</td>
<td>C)$</td>
<td>TEST or EXACT</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 or EXACT</td>
</tr>
<tr>
<td>SpearmanCorr</td>
<td>Spearman correlation</td>
<td>TEST or EXACT</td>
<td>SCORR / 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 or EXACT</td>
<td>SCORR / 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 or EXACT</td>
<td>KENTB / 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 or EXACT</td>
<td>KENTB / KENTB</td>
</tr>
<tr>
<td>TauC</td>
<td>Stuart’s tau-</td>
<td>TEST or EXACT</td>
<td>STUTC / 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 or EXACT</td>
<td>STUTC / 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>
</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>
<th>Statement</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>WtKappa</td>
<td>Weighted kappa coefficient</td>
<td>TEST</td>
<td>WTKAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or EXACT</td>
<td></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</td>
<td>WTKAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or EXACT</td>
<td></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>OVERLAY</td>
</tr>
<tr>
<td>Overlaid</td>
<td>Two or more plots on a single set of axes</td>
<td></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 Produced by Base SAS High-Performance Procedures

**Table A1.16**  ODS Table Names Produced by Base SAS High-Performance Procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPBIN</td>
<td>ODS Tables Produced by PROC HPBIN</td>
</tr>
<tr>
<td>HPCORR</td>
<td>ODS Tables Produced by PROC HPCORR</td>
</tr>
<tr>
<td>HPIMPUTE</td>
<td>ODS Tables Produced by PROC HPIMPUTE</td>
</tr>
<tr>
<td>HPSAMPLE</td>
<td>ODS Tables Produced by PROC HPSAMPLE</td>
</tr>
</tbody>
</table>

For detailed information, see the procedures in *Base SAS Procedures Guide: High-Performance Procedures*.

### ODS Table Names Produced by Base SAS Statistical Procedures

**Table A1.17**  ODS Table Names Produced by Base SAS Statistical Procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORR</td>
<td>ODS Tables Produced by PROC CORR</td>
</tr>
<tr>
<td>FREQ</td>
<td>ODS Tables Produced by PROC FREQ</td>
</tr>
<tr>
<td>UNIVARIATE</td>
<td>ODS Tables Produced by PROC UNIVARIATE</td>
</tr>
</tbody>
</table>

For detailed information, see the procedures in *Base SAS Procedures Guide: Statistical Procedures*. 
# ODS Table Names Produced by SAS/STAT Procedures

**Table A1.18  ODS Table Names Produced by SAS/STAT Procedures**

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>ACECLUS</td>
<td>ODS Tables Produced by PROC ACECLUS</td>
</tr>
<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>CAUSALGRAPH</td>
<td>ODS Tables Produced by PROC CAUSALGRAPH</td>
</tr>
<tr>
<td>CAUSALMED</td>
<td>ODS Tables Produced by PROC CAUSALMED</td>
</tr>
<tr>
<td>CAUSALTRT</td>
<td>ODS Tables Produced by PROC CAUSALTRT</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>GAMPL</td>
<td>ODS Tables Produced by PROC GAMPL</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>GEE</td>
<td>ODS Tables Produced by PROC GEE</td>
</tr>
<tr>
<td>GENMOD</td>
<td>ODS Table Produced by PROC GENMOD</td>
</tr>
<tr>
<td>GLIMMIX</td>
<td>ODS Tables Produced by Proc GLIMMIX</td>
</tr>
<tr>
<td>GLM</td>
<td>ODS Tables Produced by PROC GLM</td>
</tr>
<tr>
<td>GLMMOD</td>
<td>ODS Tables Produced by PROC GLMMOD</td>
</tr>
<tr>
<td>GLMPOWER</td>
<td>ODS Tables Produced by PROC GLMPOWER</td>
</tr>
<tr>
<td>GLMSELECT</td>
<td>ODS Tables Produced by PROC GLMSELECT</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>HPMIXED</td>
<td>ODS Tables Produced by PROC HPMIXED</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>
<tr>
<td>HPSPLIT</td>
<td>ODS Tables Produced by PROC HPSPLIT</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>
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>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 Test 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 Table 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>
<tr>
<td>Procedure</td>
<td>ODS Table Names Produced by SAS/STAT Procedures</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------</td>
</tr>
<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>PSMATCH</td>
<td>ODS Tables Produced by PROC PSMATCH</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>RMSTREG</td>
<td>ODS Tables Produced by PROC RMSTREG</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>
</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>TPSPLINE</td>
<td>ODS Tables Produced by PROC TPSLINE</td>
</tr>
<tr>
<td>TRANSREG</td>
<td>ODS Tables Produced by PROC TRANSREG</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>
</tbody>
</table>

**ODS Table Names Produced by SAS/STAT High-Performance Procedures**

*Table A1.19  ODS Table Names Produced by SAS/STAT High-Performance Procedures*

For detailed information, see the procedures in *SAS/STAT User’s Guide: High-Performance Procedures*

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAMPL</td>
<td>ODS Tables Produced by PROC GAMPL</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>HPRINCOMP</td>
<td>ODS Tables Produced by PROC HPRINCOMP</td>
</tr>
<tr>
<td>HPQUANTSELECT</td>
<td>ODS Tables Produced by PROC HPQUANTSELECT</td>
</tr>
</tbody>
</table>
## ODS Table Names Produced by SAS Enterprise Miner High-Performance Procedures

*Table A1.20  ODS Table Names Produced by SAS Enterprise Miner High-Performance Procedures*

For detailed information, see the procedures in *SAS Enterprise Miner: High-Performance Procedures*

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP4SCORE</td>
<td>ODS Tables Produced by PROC HP4SCORE</td>
</tr>
<tr>
<td>HPCLUS</td>
<td>ODS Tables Produced by PROC HPCLUS</td>
</tr>
<tr>
<td>HPDECIDE</td>
<td>ODS Tables Produced by PROC HPDECIDE</td>
</tr>
<tr>
<td>HPFOREST</td>
<td>ODS Tables Produced by PROC HPFOREST</td>
</tr>
<tr>
<td>HPNEURAL</td>
<td>ODS Tables Produced by PROC HPNEURAL</td>
</tr>
<tr>
<td>HPREDUCE</td>
<td>ODS Tables Produced by PROC HPREDUCE</td>
</tr>
<tr>
<td>HPSVM</td>
<td>ODS Tables Produced by PROC HPSVM</td>
</tr>
</tbody>
</table>

## ODS Table Names Produced by SAS/ETS Procedures

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>HPREG</td>
<td>ODS Tables Produced by PROC HPREG</td>
</tr>
<tr>
<td>HPSPLIT</td>
<td>ODS Tables Produced by PROC HPSPLIT</td>
</tr>
</tbody>
</table>

For detailed information, see the procedures in *SAS/STAT User’s Guide: High-Performance Procedures*
Table A1.21  ODS Table Names Produced by SAS/ETS Procedures

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Names Produced by PROC ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARIMA</td>
<td>ODS Tables Produced by PROC ARIMA</td>
</tr>
<tr>
<td>AUTOREG</td>
<td>ODS Tables Produced by PROC AUTOREG</td>
</tr>
<tr>
<td>COPULA</td>
<td>ODS Tables Produced by PROC COPULA</td>
</tr>
<tr>
<td>COUNTREG</td>
<td>ODS Tables Produced by PROC COUNTREG</td>
</tr>
<tr>
<td>ENTROPY</td>
<td>ODS Tables Produced by PROC ENTROPY</td>
</tr>
<tr>
<td>ESM</td>
<td>ODS Tables Produced by PROC ESM</td>
</tr>
<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>
<tr>
<td>LOAN</td>
<td>ODS Tables Produced by PROC LOAN</td>
</tr>
<tr>
<td>MDC</td>
<td>ODS Tables Produced by PROC MDC</td>
</tr>
<tr>
<td>PANEL</td>
<td>ODS Tables Produced by PROC PANEL</td>
</tr>
<tr>
<td>QLIM</td>
<td>ODS Tables Produced by PROC QLIM</td>
</tr>
<tr>
<td>SIMILARITY</td>
<td>ODS Tables Produced by PROC SIMILARITY</td>
</tr>
<tr>
<td>SIMLIN</td>
<td>ODS Tables Produced by PROC SIMLIN</td>
</tr>
<tr>
<td>SPATIALREG</td>
<td>ODS Tables Produced by PROC SPATIALREG</td>
</tr>
<tr>
<td>SPECTRA</td>
<td>ODS Tables Produced by PROC SPECTRA</td>
</tr>
<tr>
<td>SSM</td>
<td>ODS Tables Produced by PROC SSM</td>
</tr>
<tr>
<td>STATESPACE</td>
<td>ODS Tables Produced by PROC STATESPACE</td>
</tr>
<tr>
<td>SYSLIN</td>
<td>ODS Tables Produced by PROC SYSLIN</td>
</tr>
<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>
</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>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</td>
<td>ODS Tables Produced by PROC X11</td>
</tr>
<tr>
<td>X13</td>
<td>ODS Tables Produced by PROC X13</td>
</tr>
</tbody>
</table>

**ODS Table Names Produced by SAS/ETS High-Performance Procedures**

*Table A1.22*  ODS Table Names Produced by SAS/ETS High-Performance Procedures

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 Produced by SAS/QC Procedures**

*Table A1.23*  ODS Table Names Produced by SAS/QC Procedures

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>FACTEX</td>
<td>ODS Tables Produced by PROC FACTEX</td>
</tr>
<tr>
<td>MVPMODEL</td>
<td>ODS Tables Produced by PROC MVPMODEL</td>
</tr>
<tr>
<td>OPTEX</td>
<td>ODS Tables Produced by PROC OPTEX</td>
</tr>
</tbody>
</table>
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>RAREEVENTS</td>
<td>ODS Tables Produced by PROC RAREEVENTS</td>
</tr>
<tr>
<td>RELIABILITY</td>
<td>ODS Tables Produced by PROC RELIABILITY</td>
</tr>
</tbody>
</table>

## ODS Table Names Produced by SAS/OR Procedures

Table A1.24  ODS Table Names Produced by SAS/OR Procedures

For detailed information, see the procedures in the following documents:
- *SAS/OR User’s Guide: Project Management*
- *SAS/OR User’s Guide: Mathematical Programming*
- *SAS/OR User’s Guide: Local Search Optimization*
- *SAS/OR User’s Guide: Network Optimization Algorithms*

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTREE</td>
<td>ODS Tables Produced by PROC DTREE</td>
</tr>
<tr>
<td>OPTLP</td>
<td>ODS Tables Produced by PROC OPTLP</td>
</tr>
<tr>
<td>OPTLSO</td>
<td>ODS Tables Produced by PROC OPTLSO</td>
</tr>
<tr>
<td>OPTMILP</td>
<td>ODS Tables Produced by PROC OPTMILP</td>
</tr>
<tr>
<td>OPTMODEL</td>
<td>ODS Tables Produced by PROC OPTMODEL</td>
</tr>
<tr>
<td>OPTNET</td>
<td>ODS Tables Produced by PROC OPTNET</td>
</tr>
<tr>
<td>OPTQP</td>
<td>ODS Tables Produced by PROC OPTQP</td>
</tr>
</tbody>
</table>
# ODS Table Names Produced by Graph Algorithms and Network Analysis Procedure

<table>
<thead>
<tr>
<th>Procedure</th>
<th>ODS Table Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTGRAPH</td>
<td>ODS Tables Produced by PROC OPTGRAPH</td>
</tr>
</tbody>
</table>
Appendix 2

Example Programs

Examples from the Gallery of ODS Samples ................................. 1168
  EPUB Output ......................................................... 1168
  Excel Output ....................................................... 1168
  HTML Output ....................................................... 1168
  PDF Output ......................................................... 1170
  PostScript Output .................................................. 1172
  PowerPoint Output ............................................... 1173
  RTF Output ......................................................... 1175

Creating the $CNTRY Format ............................................... 1179
Creating the Charity Data Set ............................................. 1179
Creating the DIVFMT. and USETYPE. Formats ......................... 1182
Creating the DistrData Data Set ...................................... 1182
Creating the Univ ODS Document ...................................... 1182
Creating the Employee_Data Data Set .................................. 1183
Creating the Energy Data Set ......................................... 1185
Creating the Exprev Data Set ......................................... 1185
Creating the Gov Data Set ............................................ 1186
Creating the Grain_Production Data Set ................................ 1187
Creating the One Data Set ............................................ 1187
Creating the Iron Data Set ............................................ 1188
Creating the Model Data Set ......................................... 1188
Creating the Neuralgia Data Set ...................................... 1189
Creating the Plants Data Set ......................................... 1190
Creating the Plant_Stats Data Set ..................................... 1190
Creating the StatePop Data Set ....................................... 1190
Creating the Stats and Stats2 Data Sets ............................... 1192
Creating the Table1 Table Template .................................... 1192

Programs That Illustrate Inheritance .................................. 1193
  Overview ............................................................... 1193
  Using the FROM option ............................................. 1193
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 165   3300.00
Canada S  37000 A200 R 215   4300.00
Mexico S  32000 A200 R 195   4875.00
Brazil M  125000 A200 R 365   9225.00
Canada M  237000 A200 R 615  15375.00
France M  348000 A200 R 725  19125.00
Mexico M  432000 A200 R 795  19875.00
Canada L  837000 A200 R 815  22375.00
France L  748000 A200 R 775  19375.00
Mexico L  932000 A200 R 895  22375.00
Brazil S  25000 A300 R 157   3925.00
Canada S  37000 A300 R 208   5200.00
Mexico S  32000 A300 R 186   4650.00
Brazil M  125000 A300 R 351   8725.00
Canada M  237000 A300 R 610  15250.00
<table>
<thead>
<tr>
<th>Region</th>
<th>Size</th>
<th>Saletype</th>
<th>Quantity</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>M</td>
<td>Retail</td>
<td>348000</td>
<td>17850.00</td>
</tr>
<tr>
<td>Mexico</td>
<td>M</td>
<td>Retail</td>
<td>432000</td>
<td>19625.00</td>
</tr>
<tr>
<td>Canada</td>
<td>L</td>
<td>Retail</td>
<td>837000</td>
<td>20150.00</td>
</tr>
<tr>
<td>France</td>
<td>L</td>
<td>Retail</td>
<td>748000</td>
<td>19200.00</td>
</tr>
<tr>
<td>Mexico</td>
<td>L</td>
<td>Retail</td>
<td>932000</td>
<td>22000.00</td>
</tr>
<tr>
<td>Brazil</td>
<td>S</td>
<td>Wholesale</td>
<td>25000</td>
<td>3140.00</td>
</tr>
<tr>
<td>Canada</td>
<td>S</td>
<td>Wholesale</td>
<td>37000</td>
<td>4160.00</td>
</tr>
<tr>
<td>Mexico</td>
<td>S</td>
<td>Wholesale</td>
<td>32000</td>
<td>3720.00</td>
</tr>
<tr>
<td>Brazil</td>
<td>M</td>
<td>Wholesale</td>
<td>125000</td>
<td>7020.00</td>
</tr>
<tr>
<td>Canada</td>
<td>M</td>
<td>Wholesale</td>
<td>237000</td>
<td>12200.00</td>
</tr>
<tr>
<td>France</td>
<td>M</td>
<td>Wholesale</td>
<td>348000</td>
<td>14280.00</td>
</tr>
<tr>
<td>Mexico</td>
<td>M</td>
<td>Wholesale</td>
<td>432000</td>
<td>15700.00</td>
</tr>
<tr>
<td>Brazil</td>
<td>L</td>
<td>Wholesale</td>
<td>625000</td>
<td>15140.00</td>
</tr>
<tr>
<td>Canada</td>
<td>L</td>
<td>Wholesale</td>
<td>837000</td>
<td>16120.00</td>
</tr>
<tr>
<td>France</td>
<td>L</td>
<td>Wholesale</td>
<td>748000</td>
<td>15360.00</td>
</tr>
<tr>
<td>Mexico</td>
<td>L</td>
<td>Wholesale</td>
<td>932000</td>
<td>17600.00</td>
</tr>
</tbody>
</table>

**PDF Output**

```plaintext
ods html close;
```
<p>| | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1345</td>
<td>1</td>
<td>2</td>
<td>17800</td>
<td>2</td>
<td>2134</td>
<td>1</td>
<td>1</td>
<td>25199</td>
<td>3</td>
<td>4138</td>
<td>1</td>
<td>3</td>
<td>29388</td>
</tr>
<tr>
<td>1</td>
<td>1345</td>
<td>1</td>
<td>2</td>
<td>12994</td>
<td>2</td>
<td>2134</td>
<td>1</td>
<td>1</td>
<td>41578</td>
<td>3</td>
<td>4138</td>
<td>1</td>
<td>3</td>
<td>16592</td>
</tr>
<tr>
<td>1</td>
<td>1345</td>
<td>1</td>
<td>2</td>
<td>21440</td>
<td>2</td>
<td>2134</td>
<td>1</td>
<td>2</td>
<td>26560</td>
<td>3</td>
<td>4138</td>
<td>1</td>
<td>3</td>
<td>12112</td>
</tr>
<tr>
<td>1</td>
<td>1345</td>
<td>1</td>
<td>3</td>
<td>35300</td>
<td>2</td>
<td>2134</td>
<td>1</td>
<td>2</td>
<td>19388</td>
<td>3</td>
<td>4138</td>
<td>1</td>
<td>3</td>
<td>19984</td>
</tr>
<tr>
<td>1</td>
<td>1345</td>
<td>1</td>
<td>3</td>
<td>25769</td>
<td>2</td>
<td>2134</td>
<td>1</td>
<td>2</td>
<td>31990</td>
<td>3</td>
<td>4279</td>
<td>1</td>
<td>1</td>
<td>19984</td>
</tr>
<tr>
<td>1</td>
<td>1345</td>
<td>1</td>
<td>3</td>
<td>24399</td>
<td>2</td>
<td>2134</td>
<td>1</td>
<td>3</td>
<td>24399</td>
<td>3</td>
<td>4279</td>
<td>1</td>
<td>1</td>
<td>7288</td>
</tr>
<tr>
<td>1</td>
<td>1578</td>
<td>1</td>
<td>1</td>
<td>8000</td>
<td>2</td>
<td>2134</td>
<td>1</td>
<td>3</td>
<td>17811</td>
<td>3</td>
<td>4279</td>
<td>1</td>
<td>1</td>
<td>12025</td>
</tr>
<tr>
<td>1</td>
<td>1578</td>
<td>1</td>
<td>1</td>
<td>5840</td>
<td>2</td>
<td>2134</td>
<td>1</td>
<td>3</td>
<td>29388</td>
<td>3</td>
<td>4279</td>
<td>1</td>
<td>2</td>
<td>14209</td>
</tr>
<tr>
<td>1</td>
<td>1578</td>
<td>1</td>
<td>1</td>
<td>9636</td>
<td>2</td>
<td>2403</td>
<td>1</td>
<td>1</td>
<td>25464</td>
<td>3</td>
<td>4279</td>
<td>1</td>
<td>2</td>
<td>10372</td>
</tr>
<tr>
<td>1</td>
<td>1578</td>
<td>1</td>
<td>2</td>
<td>7900</td>
<td>2</td>
<td>2403</td>
<td>1</td>
<td>1</td>
<td>18588</td>
<td>3</td>
<td>4279</td>
<td>1</td>
<td>2</td>
<td>17113</td>
</tr>
<tr>
<td>1</td>
<td>1578</td>
<td>1</td>
<td>2</td>
<td>35767</td>
<td>2</td>
<td>2403</td>
<td>1</td>
<td>1</td>
<td>30670</td>
<td>3</td>
<td>4279</td>
<td>1</td>
<td>3</td>
<td>13500</td>
</tr>
<tr>
<td>1</td>
<td>1578</td>
<td>1</td>
<td>2</td>
<td>9515</td>
<td>2</td>
<td>2403</td>
<td>1</td>
<td>2</td>
<td>15494</td>
<td>3</td>
<td>4279</td>
<td>1</td>
<td>3</td>
<td>9855</td>
</tr>
<tr>
<td>1</td>
<td>1578</td>
<td>1</td>
<td>3</td>
<td>4500</td>
<td>2</td>
<td>2403</td>
<td>1</td>
<td>2</td>
<td>11310</td>
<td>3</td>
<td>4279</td>
<td>1</td>
<td>3</td>
<td>16260</td>
</tr>
<tr>
<td>1</td>
<td>1578</td>
<td>1</td>
<td>3</td>
<td>3285</td>
<td>2</td>
<td>2403</td>
<td>1</td>
<td>2</td>
<td>18661</td>
<td>3</td>
<td>4290</td>
<td>1</td>
<td>1</td>
<td>10948</td>
</tr>
<tr>
<td>1</td>
<td>1578</td>
<td>1</td>
<td>3</td>
<td>5420</td>
<td>2</td>
<td>2403</td>
<td>1</td>
<td>2</td>
<td>1482</td>
<td>3</td>
<td>4290</td>
<td>1</td>
<td>1</td>
<td>7992</td>
</tr>
<tr>
<td>1</td>
<td>1674</td>
<td>1</td>
<td>1</td>
<td>11950</td>
<td>2</td>
<td>2403</td>
<td>1</td>
<td>2</td>
<td>1081</td>
<td>3</td>
<td>4290</td>
<td>1</td>
<td>1</td>
<td>13186</td>
</tr>
<tr>
<td>1</td>
<td>1674</td>
<td>1</td>
<td>1</td>
<td>8723</td>
<td>2</td>
<td>2403</td>
<td>1</td>
<td>2</td>
<td>1783</td>
<td>3</td>
<td>4290</td>
<td>1</td>
<td>2</td>
<td>14539</td>
</tr>
<tr>
<td>1</td>
<td>1674</td>
<td>1</td>
<td>1</td>
<td>14392</td>
<td>2</td>
<td>2403</td>
<td>1</td>
<td>3</td>
<td>10009</td>
<td>3</td>
<td>4290</td>
<td>1</td>
<td>2</td>
<td>10613</td>
</tr>
<tr>
<td>1</td>
<td>1674</td>
<td>1</td>
<td>2</td>
<td>13534</td>
<td>2</td>
<td>2403</td>
<td>1</td>
<td>3</td>
<td>7306</td>
<td>3</td>
<td>4290</td>
<td>1</td>
<td>2</td>
<td>17511</td>
</tr>
<tr>
<td>1</td>
<td>1674</td>
<td>1</td>
<td>2</td>
<td>9879</td>
<td>2</td>
<td>2403</td>
<td>1</td>
<td>3</td>
<td>12054</td>
<td>3</td>
<td>4290</td>
<td>1</td>
<td>3</td>
<td>11459</td>
</tr>
<tr>
<td>1</td>
<td>1674</td>
<td>1</td>
<td>2</td>
<td>16300</td>
<td>3</td>
<td>4138</td>
<td>1</td>
<td>1</td>
<td>24850</td>
<td>3</td>
<td>4290</td>
<td>1</td>
<td>3</td>
<td>8365</td>
</tr>
<tr>
<td>1</td>
<td>1674</td>
<td>1</td>
<td>3</td>
<td>17994</td>
<td>3</td>
<td>4138</td>
<td>1</td>
<td>1</td>
<td>18140</td>
<td>3</td>
<td>4290</td>
<td>1</td>
<td>3</td>
<td>13802</td>
</tr>
</tbody>
</table>

```
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"
```
data grocery;
  input Sector $ Manager $ Department $ Sales @@;
datalines;
se 1 np1 50    se 1 p1 100   se 1 np2 120   se 1 p2  80
se 2 np1 40    se 2 p1 300   se 2 np2 220   se 2 p2  70
nw 3 np1 60    nw 3 p1 600   nw 3 np2 420   nw 3 p2  30
nw 4 np1 45    nw 4 p1 250   nw 4 np2 230   nw 4 p2  73
nw 9 np1 45    nw 9 p1 205   nw 9 np2 420   nw 9 p2  76
sw 5 np1 53    sw 5 p1 130   sw 5 np2 120   sw 5 p2  50
sw 6 np1 40    sw 6 p1 350   sw 6 np2 225   sw 6 p2  80
ne 7 np1 90    ne 7 p1 190   ne 7 np2 420   ne 7 p2  86
ne 8 np1 200   ne 8 p1 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=cxaead9
                   fontweight=bold fontsize=6];
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 is Produced by the ODS Destination for PowerPoint?

Graphics output

SAS procedure output

ODS procedure output

ODS TEXT= output

LAYOUT output

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 Govfmt -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 Govfmt. Robgrp Robfmt. ;
  LOOP: OUTPUT; WEIGHT=WEIGHT-1; IF WEIGHT>0 THEN GOTO LOOP;
  DROP WEIGHT;
datalines;
  0 1 6
  0 3 3
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;
Examples from the Gallery of ODS Samples

fontstyle=italic;
end;
define crosstabs Base.Freq.CrossTabFreqs;
notes "Crosstabulation table";

style=table {bgcolor=#BFCFFF};
cell_style=data {bgcolor=#FFFFF0};
row_var_style=rowheader {bgcolor=#BFCFFF color=rowfg.};
col_var_style=header {bgcolor=#BFCFFF color=colfg.};
row_total_style=data {bgcolor=#F0F0F0};
col_total_style=data {bgcolor=#F0F0F0};
grand_total_style=datastrong {bgcolor=#F0F0F0};
legend_style=header {bgcolor=#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_ ;
style=header {bgcolor=#BFCFFF color=#6078bf fontstyle=italic};
end;

define header RowsHeader;
text _ROW_LABEL_ / _ROW_LABEL_ ^= '';
text _ROW_NAME_ ;
style=header {bgcolor=#BFCFFF color=#6078bf fontstyle=italic};
space=0;
end;

define header ColsHeader;
text _COL_LABEL_ / _COL_LABEL_ ^= '';
text _COL_NAME_ ;
style=header {bgcolor=#BFCFFF color=#6078bf fontstyle=italic};
space=1;
end;

define header ControllingFor;
dynamic StratNum StrataVariableNames StrataVariableLabels;
text "Controlling for" StrataVariableNames / StratNum > 0;
style=header;
end;

define footer Missing;
dynamic FMissing;
text *Frequency Missing = " FMissing -12.99 / FMissing ^= 0;
style=header {bgcolor=#BFCFFF color=#6078bf fontstyle=italic};
end;

define footer NoObs;
dynamic SampleSize;
text *Effective Sample Size = 0" / SampleSize = 0;
end;

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;
Creating the $CNTRY Format

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

Creating the Charity Data Set

```sas
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 yrFmrt.;
    format School schFmrt.;
    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
```
<table>
<thead>
<tr>
<th>Name</th>
<th>Year</th>
<th>Score</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candace</td>
<td>1992</td>
<td>21.11</td>
<td>5</td>
</tr>
<tr>
<td>Danny</td>
<td>1992</td>
<td>6.89</td>
<td>23</td>
</tr>
<tr>
<td>Edward</td>
<td>1992</td>
<td>53.76</td>
<td>31</td>
</tr>
<tr>
<td>Fiona</td>
<td>1992</td>
<td>48.55</td>
<td>13</td>
</tr>
<tr>
<td>Gert</td>
<td>1992</td>
<td>24.00</td>
<td>16</td>
</tr>
<tr>
<td>Harold</td>
<td>1992</td>
<td>27.55</td>
<td>17</td>
</tr>
<tr>
<td>Ima</td>
<td>1992</td>
<td>5.98</td>
<td>9</td>
</tr>
<tr>
<td>Jack</td>
<td>1992</td>
<td>20.00</td>
<td>23</td>
</tr>
<tr>
<td>Katie</td>
<td>1992</td>
<td>22.11</td>
<td>2</td>
</tr>
<tr>
<td>Lisa</td>
<td>1992</td>
<td>18.34</td>
<td>17</td>
</tr>
<tr>
<td>Tonya</td>
<td>1992</td>
<td>55.16</td>
<td>40</td>
</tr>
<tr>
<td>Max</td>
<td>1992</td>
<td>26.77</td>
<td>34</td>
</tr>
<tr>
<td>Ned</td>
<td>1992</td>
<td>28.43</td>
<td>22</td>
</tr>
<tr>
<td>Opal</td>
<td>1992</td>
<td>32.66</td>
<td>14</td>
</tr>
<tr>
<td>Patsy</td>
<td>1993</td>
<td>18.33</td>
<td>18</td>
</tr>
<tr>
<td>Quentin</td>
<td>1993</td>
<td>16.89</td>
<td>15</td>
</tr>
<tr>
<td>Randall</td>
<td>1993</td>
<td>12.98</td>
<td>17</td>
</tr>
<tr>
<td>Sam</td>
<td>1993</td>
<td>15.88</td>
<td>5</td>
</tr>
<tr>
<td>Tyra</td>
<td>1993</td>
<td>21.88</td>
<td>23</td>
</tr>
<tr>
<td>Myrtle</td>
<td>1993</td>
<td>47.33</td>
<td>26</td>
</tr>
<tr>
<td>Frank</td>
<td>1993</td>
<td>41.11</td>
<td>22</td>
</tr>
<tr>
<td>Cameron</td>
<td>1993</td>
<td>65.44</td>
<td>14</td>
</tr>
<tr>
<td>Vern</td>
<td>1993</td>
<td>17.89</td>
<td>11</td>
</tr>
<tr>
<td>Wendell</td>
<td>1993</td>
<td>23.00</td>
<td>10</td>
</tr>
<tr>
<td>Bob</td>
<td>1993</td>
<td>26.88</td>
<td>6</td>
</tr>
<tr>
<td>Leah</td>
<td>1993</td>
<td>28.99</td>
<td>23</td>
</tr>
<tr>
<td>Becky</td>
<td>1994</td>
<td>30.33</td>
<td>26</td>
</tr>
<tr>
<td>Sally</td>
<td>1994</td>
<td>35.75</td>
<td>27</td>
</tr>
<tr>
<td>Edgar</td>
<td>1994</td>
<td>27.11</td>
<td>12</td>
</tr>
<tr>
<td>Dawson</td>
<td>1994</td>
<td>17.24</td>
<td>16</td>
</tr>
<tr>
<td>Lou</td>
<td>1994</td>
<td>5.12</td>
<td>16</td>
</tr>
<tr>
<td>Damien</td>
<td>1994</td>
<td>18.74</td>
<td>17</td>
</tr>
<tr>
<td>Mona</td>
<td>1994</td>
<td>27.43</td>
<td>7</td>
</tr>
<tr>
<td>Della</td>
<td>1994</td>
<td>56.78</td>
<td>15</td>
</tr>
<tr>
<td>Monique</td>
<td>1994</td>
<td>29.88</td>
<td>19</td>
</tr>
<tr>
<td>Carl</td>
<td>1994</td>
<td>31.12</td>
<td>25</td>
</tr>
<tr>
<td>Reba</td>
<td>1994</td>
<td>35.16</td>
<td>22</td>
</tr>
<tr>
<td>Dax</td>
<td>1994</td>
<td>27.65</td>
<td>23</td>
</tr>
<tr>
<td>Gary</td>
<td>1994</td>
<td>23.11</td>
<td>15</td>
</tr>
<tr>
<td>Suzie</td>
<td>1994</td>
<td>26.65</td>
<td>11</td>
</tr>
<tr>
<td>Benito</td>
<td>1994</td>
<td>47.44</td>
<td>18</td>
</tr>
<tr>
<td>Thomas</td>
<td>1994</td>
<td>21.99</td>
<td>23</td>
</tr>
<tr>
<td>Annie</td>
<td>1994</td>
<td>24.99</td>
<td>27</td>
</tr>
<tr>
<td>Paul</td>
<td>1994</td>
<td>27.98</td>
<td>22</td>
</tr>
<tr>
<td>Alex</td>
<td>1994</td>
<td>24.00</td>
<td>16</td>
</tr>
<tr>
<td>Lauren</td>
<td>1994</td>
<td>15.00</td>
<td>17</td>
</tr>
<tr>
<td>Julia</td>
<td>1994</td>
<td>12.98</td>
<td>15</td>
</tr>
<tr>
<td>Keith</td>
<td>1994</td>
<td>11.89</td>
<td>19</td>
</tr>
<tr>
<td>Jackie</td>
<td>1994</td>
<td>26.88</td>
<td>22</td>
</tr>
<tr>
<td>Pablo</td>
<td>1994</td>
<td>13.98</td>
<td>28</td>
</tr>
<tr>
<td>L.T.</td>
<td>1994</td>
<td>56.87</td>
<td>33</td>
</tr>
<tr>
<td>Willard</td>
<td>1994</td>
<td>78.65</td>
<td>24</td>
</tr>
<tr>
<td>Kathy</td>
<td>1994</td>
<td>32.88</td>
<td>11</td>
</tr>
<tr>
<td>Abby</td>
<td>1994</td>
<td>35.88</td>
<td>10</td>
</tr>
<tr>
<td>Arturo</td>
<td>1992</td>
<td>34.98</td>
<td>14</td>
</tr>
<tr>
<td>Grace</td>
<td>1992</td>
<td>27.55</td>
<td>25</td>
</tr>
<tr>
<td>Name</td>
<td>Year</td>
<td>Value</td>
<td>Age</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>Winston</td>
<td>1992</td>
<td>23.88</td>
<td>22</td>
</tr>
<tr>
<td>Vince</td>
<td>1992</td>
<td>12.88</td>
<td>21</td>
</tr>
<tr>
<td>Claude</td>
<td>1992</td>
<td>15.62</td>
<td>5</td>
</tr>
<tr>
<td>Mary</td>
<td>1992</td>
<td>28.99</td>
<td>34</td>
</tr>
<tr>
<td>Abner</td>
<td>1992</td>
<td>25.89</td>
<td>22</td>
</tr>
<tr>
<td>Jay</td>
<td>1992</td>
<td>35.89</td>
<td>35</td>
</tr>
<tr>
<td>Alicia</td>
<td>1992</td>
<td>28.77</td>
<td>26</td>
</tr>
<tr>
<td>Preddy</td>
<td>1992</td>
<td>29.00</td>
<td>27</td>
</tr>
<tr>
<td>Eloise</td>
<td>1992</td>
<td>31.67</td>
<td>25</td>
</tr>
<tr>
<td>Jenny</td>
<td>1992</td>
<td>43.89</td>
<td>22</td>
</tr>
<tr>
<td>Thelma</td>
<td>1992</td>
<td>52.63</td>
<td>21</td>
</tr>
<tr>
<td>Tina</td>
<td>1992</td>
<td>19.67</td>
<td>21</td>
</tr>
<tr>
<td>Eric</td>
<td>1992</td>
<td>24.89</td>
<td>12</td>
</tr>
<tr>
<td>Bubba</td>
<td>1993</td>
<td>37.88</td>
<td>12</td>
</tr>
<tr>
<td>G.L.</td>
<td>1993</td>
<td>25.89</td>
<td>21</td>
</tr>
<tr>
<td>Bert</td>
<td>1993</td>
<td>28.89</td>
<td>21</td>
</tr>
<tr>
<td>Clay</td>
<td>1993</td>
<td>26.44</td>
<td>21</td>
</tr>
<tr>
<td>Leann</td>
<td>1993</td>
<td>27.17</td>
<td>17</td>
</tr>
<tr>
<td>Georgia</td>
<td>1993</td>
<td>38.90</td>
<td>11</td>
</tr>
<tr>
<td>Bill</td>
<td>1993</td>
<td>42.23</td>
<td>25</td>
</tr>
<tr>
<td>Holly</td>
<td>1993</td>
<td>18.67</td>
<td>27</td>
</tr>
<tr>
<td>Benny</td>
<td>1993</td>
<td>19.09</td>
<td>25</td>
</tr>
<tr>
<td>Cammie</td>
<td>1993</td>
<td>28.77</td>
<td>28</td>
</tr>
<tr>
<td>Amy</td>
<td>1993</td>
<td>27.08</td>
<td>31</td>
</tr>
<tr>
<td>Doris</td>
<td>1993</td>
<td>22.22</td>
<td>24</td>
</tr>
<tr>
<td>Robbie</td>
<td>1993</td>
<td>19.80</td>
<td>24</td>
</tr>
<tr>
<td>Ted</td>
<td>1993</td>
<td>27.07</td>
<td>25</td>
</tr>
<tr>
<td>Sarah</td>
<td>1993</td>
<td>24.44</td>
<td>12</td>
</tr>
<tr>
<td>Megan</td>
<td>1993</td>
<td>28.89</td>
<td>11</td>
</tr>
<tr>
<td>Jeff</td>
<td>1993</td>
<td>31.11</td>
<td>12</td>
</tr>
<tr>
<td>Taz</td>
<td>1993</td>
<td>30.55</td>
<td>11</td>
</tr>
<tr>
<td>George</td>
<td>1993</td>
<td>27.56</td>
<td>11</td>
</tr>
<tr>
<td>Heather</td>
<td>1993</td>
<td>38.67</td>
<td>15</td>
</tr>
<tr>
<td>Nancy</td>
<td>1994</td>
<td>29.90</td>
<td>26</td>
</tr>
<tr>
<td>Rusty</td>
<td>1994</td>
<td>30.55</td>
<td>28</td>
</tr>
<tr>
<td>Mimi</td>
<td>1994</td>
<td>37.67</td>
<td>22</td>
</tr>
<tr>
<td>J.C.</td>
<td>1994</td>
<td>23.33</td>
<td>27</td>
</tr>
<tr>
<td>Clark</td>
<td>1994</td>
<td>27.90</td>
<td>25</td>
</tr>
<tr>
<td>Rudy</td>
<td>1994</td>
<td>27.78</td>
<td>23</td>
</tr>
<tr>
<td>Samuel</td>
<td>1994</td>
<td>34.44</td>
<td>18</td>
</tr>
<tr>
<td>Forrest</td>
<td>1994</td>
<td>28.89</td>
<td>26</td>
</tr>
<tr>
<td>Luther</td>
<td>1994</td>
<td>72.22</td>
<td>24</td>
</tr>
<tr>
<td>Trey</td>
<td>1994</td>
<td>6.78</td>
<td>18</td>
</tr>
<tr>
<td>Albert</td>
<td>1994</td>
<td>23.33</td>
<td>19</td>
</tr>
<tr>
<td>Che-Min</td>
<td>1994</td>
<td>26.66</td>
<td>33</td>
</tr>
<tr>
<td>Preston</td>
<td>1994</td>
<td>32.22</td>
<td>23</td>
</tr>
<tr>
<td>Larry</td>
<td>1994</td>
<td>40.00</td>
<td>26</td>
</tr>
<tr>
<td>Anton</td>
<td>1994</td>
<td>35.99</td>
<td>28</td>
</tr>
<tr>
<td>Sid</td>
<td>1994</td>
<td>27.45</td>
<td>25</td>
</tr>
<tr>
<td>Will</td>
<td>1994</td>
<td>28.88</td>
<td>21</td>
</tr>
<tr>
<td>Morty</td>
<td>1994</td>
<td>34.44</td>
<td>25</td>
</tr>
</tbody>
</table>

; 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;
```
Creating the Employee_Data Data Set

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
   M  TA2  34376  15SEP48  07JUN75  203/781-1255
1653  Alexander  Susan  Bridgeport  CT
    F  ME2  35108  18OCT52  12AUG78  203/675-7715
1400  Apple  Troy  New York  NY
   M  ME1  29769  08NOV55  19OCT78  212/586-0808
1350  Arthur  Barbara  New York  NY
    F  PA3  32886  03SEP53  01AUG78  718/383-1549
1401  Avery  Jerry  Paterson  NJ
   M  TA3  38822  16DEC38  20NOV73  201/732-8787
1499  Barefoot  Joseph  Princeton  NJ
    M  ME3  43025  29APR42  10JUN68  201/812-5665
1101  Baucum  Walter  New York  NY
   M  SCP  18723  09JUN50  04OCT78  212/586-8060
1333  Blair  Justin  Stamford  CT
   M  PT2  88606  02APR49  13FEB69  203/781-1777
1402  Blalock  Ralph  New York  NY
   M  TA2  32615  20JAN51  05DEC78  718/384-2849
1479  Bostic  Marie  New York  NY
    F  TA3  38785  25DEC56  08OCT77  718/384-8816
1403  Bowden  Earl  Bridgeport  CT
   M  ME1  28072  31JAN57  24DEC79  203/675-3434
1739  Boyce  Jonathan  New York  NY
   M  PT1  66517  28DEC52  30JAN79  212/587-1247
1658  Bradley  Jeremy  New York  NY
    M  SCP  17943  11APR55  03MAR80  212/587-3622
1428  Brady  Christine  Stamford  CT
    F  PT1  68767  07APR58  19NOV79  203/781-1212
1782  Brown  Jason  Stamford  CT
   M  ME2  35345  07DEC58  25FEB80  203/781-0019
1244  Bryant  Leonard  New York  NY
   M  ME2  36925  03SEP51  20JAN76  718/383-3334
1383  Burnette  Thomas  New York  NY
   M  BCK  25823  28JAN56  23OCT80  718/384-3569
1574  Cahill  Marshall  New York  NY
    M  PA2  28572  30APR48  23DEC80  718/383-2338
1789  Caraway  Davis  New York  NY
    M  SCP  18326  28JAN45  14APR66  212/587-9000
1404  Carter  Donald  New York  NY

quit;
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1437</td>
<td>Carter</td>
<td>Dorothy</td>
<td>Bridgeport</td>
<td>CT</td>
<td>91376</td>
</tr>
<tr>
<td>1639</td>
<td>Carter</td>
<td>Karen</td>
<td>Stamford</td>
<td>CT</td>
<td>33104</td>
</tr>
<tr>
<td>1269</td>
<td>Caston</td>
<td>Franklin</td>
<td>Stamford</td>
<td>CT</td>
<td>40260</td>
</tr>
<tr>
<td>1065</td>
<td>Chapman</td>
<td>Neil</td>
<td>New York</td>
<td>NY</td>
<td>35090</td>
</tr>
<tr>
<td>1876</td>
<td>Chin</td>
<td>Jack</td>
<td>New York</td>
<td>NY</td>
<td>39675</td>
</tr>
<tr>
<td>1037</td>
<td>Cooper</td>
<td>Anthony</td>
<td>New York</td>
<td>NY</td>
<td>32217</td>
</tr>
<tr>
<td>1983</td>
<td>Dean</td>
<td>Sandra</td>
<td>Bridgeport</td>
<td>CT</td>
<td>32925</td>
</tr>
<tr>
<td>1134</td>
<td>Delgado</td>
<td>Maria</td>
<td>Stamford</td>
<td>CT</td>
<td>33419</td>
</tr>
<tr>
<td>1118</td>
<td>Dennis</td>
<td>Roger</td>
<td>New York</td>
<td>NY</td>
<td>33462</td>
</tr>
<tr>
<td>1438</td>
<td>Donaldson</td>
<td>Karen</td>
<td>Stamford</td>
<td>CT</td>
<td>39223</td>
</tr>
<tr>
<td>1125</td>
<td>Dunlap</td>
<td>Donna</td>
<td>New York</td>
<td>NY</td>
<td>39771</td>
</tr>
<tr>
<td>1935</td>
<td>Fernandez</td>
<td>Katrina</td>
<td>Bridgeport</td>
<td>CT</td>
<td>51081</td>
</tr>
<tr>
<td>1124</td>
<td>Fields</td>
<td>Diana</td>
<td>White Plains</td>
<td>NY</td>
<td>23177</td>
</tr>
<tr>
<td>1422</td>
<td>Fletcher</td>
<td>Marie</td>
<td>Princeton</td>
<td>NJ</td>
<td>22268</td>
</tr>
<tr>
<td>1616</td>
<td>Flowers</td>
<td>Annette</td>
<td>New York</td>
<td>NY</td>
<td>34137</td>
</tr>
<tr>
<td>1406</td>
<td>Foster</td>
<td>Gerald</td>
<td>Bridgeport</td>
<td>CT</td>
<td>35185</td>
</tr>
<tr>
<td>1120</td>
<td>Garcia</td>
<td>Jack</td>
<td>New York</td>
<td>NY</td>
<td>28619</td>
</tr>
<tr>
<td>1094</td>
<td>Gomez</td>
<td>Alan</td>
<td>Bridgeport</td>
<td>CT</td>
<td>22268</td>
</tr>
<tr>
<td>1389</td>
<td>Gordon</td>
<td>Levi</td>
<td>New York</td>
<td>NY</td>
<td>25028</td>
</tr>
<tr>
<td>1905</td>
<td>Graham</td>
<td>Alvin</td>
<td>New York</td>
<td>NY</td>
<td>65111</td>
</tr>
<tr>
<td>1407</td>
<td>Grant</td>
<td>Daniel</td>
<td>Mt. Vernon</td>
<td>NY</td>
<td>68096</td>
</tr>
</tbody>
</table>
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 WI 1 194 4 3 WI 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 4 WA 1 1680 4 4 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 7 286.40 13.50
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 120454 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

;
### Nicaragua
- Code: 120932
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Internet
- Quantity: 16
- Price: 122.00
- Tax: 28.75

### Panama
- Code: 99999999
- Date: 1/2/05
- Billing Date: 1/6/05
- Method: Internet
- Quantity: 20
- Price: 88.20
- Tax: 38.40

### Saint Kitts/Nevis
- Code: 99999999
- Date: 1/2/05
- Billing Date: 1/6/05
- Method: Internet
- Quantity: 20
- Price: 41.40
- Tax: 18.00

### St. Helena
- Code: 120360
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Internet
- Quantity: 19
- Price: 94.70
- Tax: 47.45

### St. Pierre/Miquelon
- Code: 120842
- Date: 1/2/05
- Billing Date: 1/16/05
- Method: Internet
- Quantity: 16
- Price: 103.80
- Tax: 47.25

### Turks/Caicos Islands
- Code: 120372
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Internet
- Quantity: 10
- Price: 57.70
- Tax: 28.95

### United States
- Code: 120372
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Internet
- Quantity: 20
- Price: 88.20
- Tax: 38.40

### Anguilla
- Code: 99999999
- Date: 1/2/05
- Billing Date: 1/6/05
- Method: In Store
- Quantity: 15
- Price: 233.50
- Tax: 22.25

### Antigua/Barbuda
- Code: 120458
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: In Store
- Quantity: 31
- Price: 99.60
- Tax: 45.35

### Argentina
- Code: 99999999
- Date: 1/2/05
- Billing Date: 1/6/05
- Method: In Store
- Quantity: 42
- Price: 408.80
- Tax: 87.15

### Barbados
- Code: 99999999
- Date: 1/2/05
- Billing Date: 1/6/05
- Method: In Store
- Quantity: 26
- Price: 94.80
- Tax: 42.60

### Bolivia
- Code: 120127
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: In Store
- Quantity: 26
- Price: 66.00
- Tax: 16.60

### Brazil
- Code: 120127
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Catalog
- Quantity: 12
- Price: 73.40
- Tax: 18.45

### Chile
- Code: 120455
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: In Store
- Quantity: 25
- Price: 132.80
- Tax: 30.25

### Colombia
- Code: 121059
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Internet
- Quantity: 28
- Price: 361.40
- Tax: 90.45

### Dominica
- Code: 121043
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Internet
- Quantity: 35
- Price: 121.30
- Tax: 51.80

### Ecuador
- Code: 120445
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Internet
- Quantity: 21
- Price: 231.60
- Tax: 48.70

### Falkland Islands
- Code: 120932
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: In Store
- Quantity: 11
- Price: 100.90
- Tax: 50.55

### French Guiana
- Code: 120935
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Catalog
- Quantity: 15
- Price: 96.40
- Tax: 43.85

### Grenada
- Code: 120931
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Catalog
- Quantity: 19
- Price: 56.30
- Tax: 25.05

### Guadeloupe
- Code: 120445
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Catalog
- Quantity: 21
- Price: 231.60
- Tax: 48.70

### Guyana
- Code: 120844
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Catalog
- Quantity: 25
- Price: 132.80
- Tax: 30.25

### Martinique
- Code: 120931
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Internet
- Quantity: 16
- Price: 56.30
- Tax: 31.05

### Netherlands Antilles
- Code: 99999999
- Date: 1/2/05
- Billing Date: 1/6/05
- Method: In Store
- Quantity: 31
- Price: 41.80
- Tax: 19.45

### Paraguay
- Code: 120845
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Catalog
- Quantity: 17
- Price: 117.60
- Tax: 59.80

### Peru
- Code: 120845
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Catalog
- Quantity: 12
- Price: 93.80
- Tax: 41.75

### St. Lucia
- Code: 120845
- Date: 1/2/05
- Billing Date: 1/2/05
- Method: Internet
- Quantity: 19
- Price: 64.30
- Tax: 28.65

### Suriname
- Code: 120538
- Date: 1/3/05
- Billing Date: 1/3/05
- Method: Internet
- Quantity: 22
- Price: 110.80
- Tax: 29.35

---

### Creating the Gov Data Set

```plaintext
data gov;
  Label Citygovt='City Government Form' Robgrp='Number of Citizens Robbed';
  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
  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;
```
Creating the Grain_Production Data Set

```plaintext
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

```plaintext
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.

```sas
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

```sas
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 621448018932921207568469573610686654502881317293178014 66671 566079
  631507969 666125121046261533088511701311167369516282157914179791706485  -4568
  641811051 73194502173784145410695409591467724582292194553420655145948 -10940
  652532026 816707123363201962785640926221155676314091906268218759195733 -145568
  661845213 88903932680634222339564930721533118605504173294828832275400 132143
;run;
```
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 PP 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 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
Creating the StatePop Data Set

```
MA   5.530   5.788   .207   .229  1
RI    .886    .938   .061   .065  1
CT   2.982   3.148   .126   .140  1
NY  16.144  16.515  1.414  1.475  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
GA   3.507   4.352  1.956  2.127  2
FL   9.039  12.023   .708   .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    .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
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    .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   .333   .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;
```
Creating the Stats and Stats2 Data Sets

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

proc template;
  define table table1;
    mvar sysdate9;
    mdynamic 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;
Programs That Illustrate Inheritance

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;
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 /
backgroundcolor=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=celldatasimple;
      dataname=country;
      header='Country';
    end;
    define y;
```
*Programs That Illustrate Inheritance* 1197

```plaintext
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;
  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.

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;
run;
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_;set test;
  file print ods=(template='mytable');
  put _ods_;run;
ods html close;
Appendix 3
ODS and the HTML Destination

HTML Links and References Produced by the HTML Destination .......................... 1201
What Are Links and References? ................................................................. 1201
Implementing HTML Links and References .................................................. 1201
How ODS Constructs Links and References .................................................. 1203

Files Produced by the HTML Destination ....................................................... 1206
Overview ..................................................................................................... 1206
The Body File ............................................................................................ 1206
The Contents File ....................................................................................... 1209
The Page File ............................................................................................. 1209
The Frame File ........................................................................................... 1209

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
references for you. For a complete discussion of HTML tagging, consult one of the many reference books that are available on the subject.

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

```
This <A href="#target1"> link</A> points to the anchor tag in this file with NAME="target1".

more text...

The target is elsewhere in the file. In fact, it is located <A name="target1">right here</A> in this paragraph.

more text...
```

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 396 as well as “How ODS Constructs Links and References” on page 1203.

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 1203.

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 396. 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 libref.catalog</td>
<td>Not specified</td>
<td>The name of the file</td>
</tr>
<tr>
<td>external-file or libref.catalog</td>
<td>Specified, but not NONE</td>
<td>The value of the URL= suboption</td>
</tr>
<tr>
<td>external-file or libref.catalog</td>
<td>NONE</td>
<td>No information from PATH=</td>
</tr>
<tr>
<td>fileref</td>
<td>Specified or not specified</td>
<td>No information from PATH=</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 509.

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 396 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 396 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 1203. 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 1212.

**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 1206. For example, an anchor tag that links to the second HTML table of results in Figure A3.4 on page 1208 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 1209.

**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 1206. For example, an anchor tag that links to the second page of results in Figure A3.4 on page 1208 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 1209.

**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 1210.
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 1.301 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 1.651 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 .333 .387 4 NV .666 .104 .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
Appendix 4

ODS HTML Statements for Running Examples in Different Operating Environments

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 655 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>
<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>fall_over</td>
<td>y</td>
<td>left</td>
<td>left</td>
</tr>
<tr>
<td></td>
<td></td>
<td>right</td>
<td></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></td>
<td></td>
</tr>
<tr>
<td>flip</td>
<td>n</td>
<td>left</td>
<td>right</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>out</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>out_with_bounce</td>
</tr>
<tr>
<td>fracture</td>
<td>y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gallery</td>
<td>n</td>
<td>from_right</td>
<td>from_right</td>
</tr>
<tr>
<td></td>
<td></td>
<td>from_left</td>
<td></td>
</tr>
<tr>
<td>glitter</td>
<td>n</td>
<td>hexagons_from_left</td>
<td>hexagons_from_left</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></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></td>
<td></td>
</tr>
<tr>
<td>orbit</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>from_top</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>from_left</td>
<td></td>
<td></td>
</tr>
<tr>
<td>origami</td>
<td>y</td>
<td>right</td>
<td>right</td>
</tr>
<tr>
<td></td>
<td>left</td>
<td></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>double_right</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>single_left</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>single_right</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pan</td>
<td>n</td>
<td>from_bottom</td>
<td>from_bottom</td>
</tr>
<tr>
<td></td>
<td>from_left</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>from_right</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>from_top</td>
<td></td>
<td></td>
</tr>
<tr>
<td>peel_off</td>
<td>y</td>
<td>left</td>
<td>left</td>
</tr>
<tr>
<td></td>
<td>right</td>
<td></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>from_top</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>from_left</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>from_right</td>
<td></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>horizontal</td>
<td></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>smoothly_from_left</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></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></td>
</tr>
<tr>
<td></td>
<td></td>
<td>right</td>
<td>right</td>
</tr>
<tr>
<td>uncover</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_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_botttom_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.
Recommended Reading

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

- SAS ODS Graphics Editor: User’s Guide
- 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 DATA Step 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/GRAPH 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 '$' 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) and XML (Extensible Markup Language). SAS supplies many markup languages for you to use, ranging from CHTML to PHTML. 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 213
_ODS_ option
  PUT statement 60, 901
#BYLINE option
  ODS GRAPHICS statement
    (IMAGENAME option) 361
#BYVAL option
  ODS GRAPHICS statement
    (IMAGENAME option) 361
#BYVAR option
  ODS GRAPHICS statement
    (IMAGENAME option) 361

A
  absolute layout statement 504, 956
  ABSOLUTE LAYOUT statement 474, 925
  ABSTRACT= option
    ODS PACKAGE statement 579
  ABSTRACT= style attribute 1098
  accessibility
    absolute layouts 475, 926
    EPUB output 215
    EPUB3 output 237
    gridded layouts 492, 944
    HTML5 output 441
    images 455
    ODS EPUB statement 217, 253
    ODS EPUB2 statement 229
    ODS EPUB3 statement 253
    ODS LAYOUT ABSOLUTE statement 475, 926
    ODS LAYOUT GRIDDED statement 492, 944
    ODS REGION, absolute statement 486, 937
    ODS REGION, gridded statement 501, 953
    ODS, creating output 20
    Output Delivery System 20
    PDF output 600
  regions, absolute 486, 937
  regions, gridded 501, 953
  ACCESSIBLE option
    ODS PDF statement 600
  Acrobat Distiller 735, 760
  actions
    ODS CHTML statement 176
    ODS CSVALL statement 201
    ODS DOCUMENT statement 212
    ODS HTML3 statement 371
    ODS LISTING statement 505
    ODS MARKUP statement 509
    ODS OUTPUT statement 555
    ODS PHTML statement 697
    ODS PRINTER statement 725
    ODS RTF statement 767
    ODS Tagset statement 809
  ACTIVELINKCOLOR= style attribute 1098
  ActiveX devices
    CODEBASE file path 122, 182, 377, 405, 517, 703, 816
  ADVANCE option
    ODS LAYOUT GRIDDED statement 490, 942
  ALL option
    ODS EXCLUDE statement 334
  ANCHOR option
    ODS CHTML statement 177
    ODS EPUB statement 220
    ODS EPUB3 statement 240
    ODS EXCEL statement 299
    ODS HTML statement 117, 399
    ODS HTML3 statement 372
    ODS HTML5 statement 444
    ODS PHTML statement 698
    ODS Tagset statement 811
    ODS TAGSETS.RTF statement 841
    ODS WORD statement 906
  anchor tags
    base name for 512, 767, 841
    root name for 600, 725, 752
  ANCHOR= option
    ODS MARKUP statement 512
ODS PDF statement 600
ODS PRINTER statement 725
ODS PS statement 752
ODS RTF statement 767
ANTIALIAS option
ODS GRAPHICS statement 342
ANTIALIAS= option
ODS GRAPHICS statement 342
ANTIALIASMAX= option
ODS GRAPHICS statement 343
APPEND option
ODS PATH statement 586
appending HTML files 151, 433
applets
viewing HTML output 118, 400
ARCHIVE option
ODS CHTML statement 178
ODS HTML statement 118, 400
ODS HTML3 statement 373
ODS PHTML statement 699
ODS Tagset statement 812
ARCHIVE= option
ODS MARKUP statement 513
arguments
ODS OUTPUT statement 555
ARIMA procedure
ODS table names 1161
ASIS= style attribute 1098
ATTRIBUTES option
ODS CHTML statement 178
ODS HTML statement 119, 401
ODS HTML3 statement 373
ODS PHTML statement 700
ODS Tagset statement 813
ATTRIBUTES= option
ODS MARKUP statement 514
ATTRPRIORITY= option
ODS GRAPHICS statement 343
AUDIO function 279
AUTHOR option
ODS EXCEL statement 299
ODS POWERPOINT statement 652
ODS TAGSETS.RTF statement 841
ODS WORD statement 906
AUTHOR= option
ODS PDF statement 601
ODS PRINTER statement 726
ODS PS statement 753
ODS RTF statement 768

BACKGROUND= style attribute 1099
BACKGROUNDCOLOR= style attribute 1099
BACKGROUNDIMAGE= style attribute 1099
BACKGROUNDPOSITION= style attribute 1099
BACKGROUNDREPEAT= style attribute 1099
BASE option
ODS CHTML statement 179
ODS HTML statement 119, 402
ODS HTML3 statement 374
ODS HTML5 statement 445
ODS PHTML statement 700
ODS POWERPOINT statement 652
ODS Tagset statement 813
ODS TAGSETS.RTF statement 842
ODS WORD Statement 906
base text 601, 726, 753, 768, 842
HTML output 118, 402
BASE= option
ODS MARKUP statement 514
ODS PDF statement 601
ODS PRINTER statement 726
ODS PS statement 753
ODS RTF statement 768
batch jobs
ODS GRAPHICS statement for 360
benefits of ODS 4
body files 1206
creating 133, 190, 204, 386, 415, 456,
527, 593, 608, 712, 733, 759, 825
separate file per page of output 147,
429
BODY option
ODS CHTML statement 179
ODS CSV ALL statement 202
ODS HTML3 statement 374
ODS HTML5 statement 445
ODS PHTML statement 700
ODS Tagset statement 814
BODY= option
ODS destination statements 14
ODS HTML statement 119, 402
ODS MARKUP statement 514
ODS RTF statement 773
suboptions 514
BODYSCLROLLBAR= style attribute
1100
BODYSIZE= style attribute 1101
BODYTITLE option
ODS RTF statement 768
BODYTITLE_AUX option
ODS RTF statement 769
BOLD function 280
BOOKMARK= option
ODS PDF statement 600
ODS PRINTER statement 725
ODS PS statement 752
BOOKMARKGEN= option
ODS PDF statement 602
ODS PRINTER statement 727
ODS PS statement 754
BOOKMARKLIST= option
ODS PDF statement 601
ODS PRINTER statement 727
ODS PS statement 753
bookmarks
for PDF files 601, 602, 727, 753, 754
BORDER option
ODS GRAPHICS statement 343
BORDER= option
ODS GRAPHICS statement 343
BORDERBOTTOMCOLOR= style attribute 1101
BORDERBOTTOMSTYLE= style attribute 1101
BORDERBOTTOMWIDTH= style attribute 1102
BORDERCOLLAPSE= style attribute 1102
BORDERCOLOR= style attribute 1102
BORDERCOLORORDARK= style attribute 1102
BORDERCOLORORLIGHT= style attribute 1102
BORDERLEFTCOLOR= style attribute 1103
BORDERLEFTSTYLE= style attribute 1103
BORDERLEFTWIDTH= style attribute 1103
BORDERRIGHTCOLOR= style attribute 1103
BORDERRIGHTSTYLE= style attribute 1104
BORDERRIGHTWIDTH= style attribute 1104
BORDERSPACING= style attribute 1104
BORDERSTYLE= style attribute 1105
BORDERTOPCOLOR= style attribute 1105
BORDERTOPSTYLE= style attribute 1106
BORDERTOPWIDTH= style attribute 1106
BORDERWIDTH= style attribute 1106
BOTTOMMARGIN= style attribute 1106
BOX_SIZING= option
ODS EPUB statement 220
ODS EPUB3 statement 240
ODS EXCEL statement 300
ODS HTML statement 121, 403
ODS HTML5 statement 447
ODS PHTML statement 702
ODS RTF statement 769, 842
BOX_SIZING= option
ODS PCL statement 589
ODS PDF statement 603
ODS PRINTER statement 728
ODS PS statement 754
ODS WORD statement 907
buffers
number of columns in 62, 903
BULLET= style attribute 1119
BY lines
substituting in an image filename 361
BYGROUP argument
ODS EPUB statement 225
BYGROUP option
ODS EPUB3 statement 246
BYLINE= option
ODS GRAPHICS statement 344

C

CALENDAR procedure
  ODS table names 1141
CAPSTYLE= style attribute 1107
cascading style sheets
  applying to ODS output 548
  multiple, in one HTML document 547
CAT= option
  ODS DOCUMENT statement 213
CATALOG option
  ODS DOCUMENT statement 213
CATALOG procedure
  ODS table names 1141
catalogs
  copying GSREGs to 213
CATEGORY option
  ODS EXCEL statement 300
  ODS POWERPOINT statement 652
  ODS WORD statement 907
CELLHEIGHT= style attribute 1116
CELLPADDING= style attribute 1107
CELLSPACING= style attribute 1104
CELLWIDTH= style attribute 1131
character sets
  META declaration for HTML output 121, 181, 376, 404, 447, 516, 702, 815
CHARSET option
  ODS CHTML statement 181
  ODS HTML statement 121, 404
  ODS HTML3 statement 376
  ODS HTML5 statement 447
  ODS PHTML statement 702
  ODS Tagset statement 815
CHARSET= option
COLOR= style attribute 1107
  colors
    ODS PDF statement 603
    ODS PRINTER statement 728
    ODS PS statement 754
  column attributes
    values from data component 70, 166
  column definitions
    for multiple variables 66, 71, 162, 167
column pointer controls
  ODS 60, 902
COLUMN. GUTTER option
  ODS LAYOUT GRIDDED statement 490, 942
COLUMN. SPAN option
  ODS REGION statement, gridded 501, 953
COLUMN. WIDTHS option
  ODS LAYOUT GRIDDED statement 491, 943
columns
  assigning attributes to 82
    for data components 64, 160
  formats for 70, 166
  justification 211
  labels for 67, 71, 163, 167
  number in buffers 62, 903
  number in data components 62, 903
  ODS PDF statement 603
  ODS PRINTER statement 729
  ODS PS statement 755
  ODS RTF statement 769
  specifying 69, 165
COLUMNS option
  ODS LAYOUT GRIDDED statement 490, 942
  ODS REGION statement, gridded 500, 952
  ODS TAGSETS.RTF statement 842
COLUMNS= option
  ODS PCL statement 589
  ODS PDF statement 603
  ODS PRINTER statement 729
  ODS PS statement 755
  ODS RTF statement 769
COLUMNS= suboption
  FILE PRINT ODS statement 64, 160
  comma-delimited output 200
COMMENTS option
  ODS EXCEL statement 300
  ODS POWERPOINT statement 652
  ODS WORD statement 907
COMPARE procedure
  ODS table names 1142
COMPRESSION option
  ODS PDF statement 603
  ODS PRINTER statement 728
compression
  PDF files 604, 729
CONNECT= style attribute 1108
contents file 1209
CONTENTS option
  ODS CHTML statement 182
  ODS HTML statement 123, 405
  ODS HTML3 statement 377
  ODS HTML5 statement 448
  ODS PHTML statement 704
  ODS RTF statement 770
  ODS Tagset statement 817
CONTENTS procedure
  ODS table names 1144
  CONTENTS= option
    ODS MARKUP statement 518
    ODS PDF statement 604
    ODS PRINTER statement 729
CONTENTSCROLLBAR= style attribute 1108
CONTENTSIZEx= style attribute 1108, 1109
CONTRONTTYPE= style attribute 1109
CONTRASTCOLOR= style attribute 1109
CORR procedure
  ODS table names 1143
CSSSTYLE option
  ODS EPUB statement 220
  ODS EPUB3 statement 241
  ODS EXCEL statement 300
  ODS HTML statement 124, 407, 818
  ODS HTML3 statement 379
  ODS HTML5 statement 300, 449
  ODS MARKUP statement 519
  ODS PCL statement 590
  ODS PDF statement 604
  ODS PHTML statement 705
  ODS PRINTER statement 730
  ODS PS statement 755
  ODS RTF statement 770
CSV tagset 810
CSVALL destination 200
CSVALL tagset 810
CSVBYLINE tagset 810
customized output
  for output objects 43

DATA option
  ODS LISTING statement 506
data panels 506
data sets
  combined output data sets 559
  creating with/without MATCH_ALL option 566
  from output objects 555
  from similar output objects 562
  merging dissimilar output objects into 558
DATA step
  column definitions for multiple variables 66, 71, 162, 167
  ODS and 57
  ODS enhanced features in 59
  ODS examples 73
  ODS reports with 58
DATA step statements
  ODS 107
  DATAPANEL= option
    ODS LISTING statement 506
DATASETS procedure
  ODS table names 1144
DATASKIN= style attribute 1109
DATASKINMAX= option
  ODS GRAPHICS statement 345
DATE function 280
decimal point
  in numeric columns 211
default devices 506, 520, 771
DESC
  ODS LAYOUT ABSOLUTE statement 475, 926
  ODS LAYOUT GRIDDED statement 492, 944
  ODS REGION, absolute statement 486, 937
  ODS REGION, gridded statement 501, 953
DESCRIPTION
  ODS LAYOUT ABSOLUTE statement 475, 926
  ODS LAYOUT GRIDDED statement 492, 944
  ODS REGION, absolute statement 486, 937
  ODS REGION, gridded statement 501, 953
DESCRIPTION= option
  ODS PACKAGE statement 579
destination-independent input 37
DEVICD option
  ODS TAGSETS.RTF statement 842
DEVICE option
  ODS EPUB statement 222
Index

ODS EPUB3 statement 242
ODS HTML statement 125, 408
ODS HTML5 statement 451
DEVICE=
ODS MARKUP statement 520
ODS RTF statement 506, 771
DIR= option
ODS DOCUMENT statement 213
DISCRETEMAX= option
ODS GRAPHICS statement 345
DISPLAYOPTS= style attribute
DOC_SEQNO= option
LIBNAME statement, SASEDOC 170
DOCTYPE= style attribute
DOCUMENT destination 37, 212
closing 212
excluding output objects 212
selecting output objects 213
writing selection/exclusion lists to log 213
DOM option
ODS CHTML statement 184
ODS CSV ALL statement 203
ODS EPUB statement 222
ODS EPUB3 statement 242
ODS HTML statement 126, 301, 409, 450, 820
ODS HTML3 statement 380
ODS MARKUP statement 521
ODS PCL statement 591
ODS PDF statement 606
ODS PHTML statement 707
ODS TRACE statement 892
double trailing @
PUT _ODS_ statement 60, 901
DPI option
ODS EXCEL statement 301, 302
ODS POWERPOINT statement 653
DPI= option
ODS EPUB statement 225
ODS EPUB3 statement 246
ODS LISTING statement 508
ODS PCL statement 591
ODS PDF statement 606
ODS PRINTER statement 731
ODS PS statement 757
ODS RTF statement 774
DRILLTARGET= option
ODS GRAPHICS statement 345
DROPSHADOW= style attribute 1111
DTDs
creating, with XML files 542
dynamic attributes
default values for 66, 162
dynamic graphics output
attributes between tags 514
parameters between tags 530
DYNAMIC= attribute suboption
FILE PRINT ODS statement 70, 166
DYNAMIC= suboption
FILE PRINT ODS statement 66, 162

E
EMPHASIS function 281
ENCODING option
ODS CHTML statement 184
ODS CSVALL statement 203
ODS HTML statement 127, 409
ODS HTML3 statement 380
ODS HTML5 statement 451
ODS PHTML statement 707
ODS Tagset statement 820
ODS TAGSETS.RTF statement 843
ENCODING= option
ODS MARKUP statement 521
ODS RTF statement 772
ENDCOLOR= style attribute 1112
EPUB 215
EPUB destinations 38
EPUB2 217
EPUB3 237
escape characters
for inline formatting 273
EVENT option
ODS CHTML statement 184
ODS CSVALL statement 203
ODS EPUB statement 222
ODS HTML statement 127, 409
ODS HTML3 statement 380
ODS PHTML statement 707
ODS Tagset statement 820
EVENT_MAP tagset 833
EVENT= option
ODS MARKUP statement 521
examples
operating environments for 1213
programs for 1183
EXCEL 296, 904
BORDERBOTTOMSTYLE 331
CSS Stylesheet 327
style attribute 322, 325
SVG Mode 320, 912
Excel destination 38
EXCLUDE action
ODS CSVALL statement 201
ODS DOCUMENT statement 212
ODS LISTING statement 205
ODS MARKUP statement 512
INDEX

ODS PCL statement 588
ODS PDF statement 599
ODS PRINTER statement 725
ODS PS statement 751
ODS RTF statement 767
EXCLUDE option
  ODS CHTML statement 176
  ODS EPUB statement 219
  ODS EPUB3 statement 239
  ODS EXCEL statement 298
  ODS HTML statement 117, 399
  ODS HTML3 statement 371
  ODS HTML5 statement 443
  ODS PHTML statement 698
  ODS POWERPOINT statement 651
  ODS Tagset statement 809
  ODS HTML3 statement 381
  ODS HTML5 statement 452
  ODS PHTML statement 708
  ODS Tagset statement 821
EXCLUDED option
  ODS TRACE statement 892
exclusion lists 42, 336, 796
  destinations for output objects 41
OUTPUT destination 555
writing to log 806

F
FILE 63, 159
FILE option 223, 652
  ODS EPUB3 statement 245
  ODS EXCEL statement 301
  ODS TAGSETS.RTF statement 843
  ODS WORD statement 907
FILE PRINT ODS statement 59
  restrictions 71, 167
  syntax 63, 159
  without ODS suboptions 73, 169
FILE statement for ODS 63, 159
FILE= option
  ODS destination statements 14
  ODS HTML statement 119, 402
  ODS LISTING statement 507
  ODS MARKUP statement 514
  ODS PCL statement 591
  ODS PDF statement 606
  ODS PRINTER statement 731
  ODS PS statement 757
  ODS RTF statement 773
FILLPATTERN= style attribute 1112
FILLRULEWIDTH= style attribute 1112
FLYOVER= style attribute 1112
FONT_FACE= style attribute 1113
FONT= style attribute 1113
FONTFAMILY= style attribute 1113
FONTSIZE= style attribute 1113
FONTSTYLE= style attribute 1114
FONTWEIGHT= style attribute 1114
FONTWIDTH= style attribute 1115
footnotes
  in graphics output 524
  Printer output 732
  RTF output 773
FOREGROUNDCOLOR= style attribute 1107
FORMAT= attribute suboption
  FILE PRINT ODS statement 70, 166
formats
  for columns 70, 166
frame files 1209
FRAME option
  ODS CHTML statement 185
  ODS HTML statement 128, 410
  ODS HTML3 statement 381
  ODS HTML5 statement 452
  ODS PHTML statement 708
  ODS Tagset statement 821
FRAME= option
  ODS MARKUP statement 523
FRAME= style attribute 1115
FRAMEBORDER= style attribute 1116
FREQ procedure
  ODS table names 1145

G
GENERIC= attribute suboption
  FILE PRINT ODS statement 71, 167
GENERIC= suboption
  FILE PRINT ODS statement 66, 162
GFOOTNOTE option
  ODS CHTML statement 187
  ODS EPUB statement 224
  ODS EPUB3 statement 245
  ODS Excel statement 301
  ODS HTML statement 129, 412
  ODS HTML3 statement 383
  ODS HTML5 statement 453
  ODS MARKUP statement 524
  ODS PCL statement 592
  ODS PDF statement 607
  ODS HTML3 statement 709
  ODS POWERPOINT statement 653
  ODS PS statement 758
  ODS Tagset statement 823
  ODS HTML3 statement 844
  ODS WORD statement 907
GFOOTNOTE= option
  ODS PRINTER statement 732
  ODS RTF statement 773
global statements
  category descriptions 110
  ODS 107
<table>
<thead>
<tr>
<th>GPATH option</th>
<th>ODS CHTML statement</th>
<th>187</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODS HTML statement</td>
<td></td>
<td>130, 412</td>
</tr>
<tr>
<td>ODS HTML3 statement</td>
<td></td>
<td>383</td>
</tr>
<tr>
<td>ODS HTML5 statement</td>
<td></td>
<td>453</td>
</tr>
<tr>
<td>ODS PHTML statement</td>
<td></td>
<td>710</td>
</tr>
<tr>
<td>ODS Tagset statement</td>
<td></td>
<td>823</td>
</tr>
<tr>
<td>GPATH= option</td>
<td>ODS LISTING statement</td>
<td>507</td>
</tr>
<tr>
<td>ODS MARKUP statement</td>
<td></td>
<td>525</td>
</tr>
<tr>
<td>GRADIENT_DIRECTION= attribute</td>
<td></td>
<td>1116</td>
</tr>
<tr>
<td>graph segments (GRSEGs)</td>
<td></td>
<td>213</td>
</tr>
<tr>
<td>graphics</td>
<td>ODS RTF statement and</td>
<td>781</td>
</tr>
<tr>
<td></td>
<td>ODS TAGSET.RTF statement and</td>
<td>857</td>
</tr>
<tr>
<td></td>
<td>smoothing</td>
<td>342</td>
</tr>
<tr>
<td></td>
<td>template-based</td>
<td>340</td>
</tr>
<tr>
<td>graphics output</td>
<td>enabling for ODS</td>
<td>897, 898</td>
</tr>
<tr>
<td></td>
<td>ODS settings</td>
<td>896</td>
</tr>
<tr>
<td>footnotes in</td>
<td></td>
<td>524</td>
</tr>
<tr>
<td>location for</td>
<td></td>
<td>507, 525</td>
</tr>
<tr>
<td>titles in</td>
<td></td>
<td>131, 188, 384, 413, 454, 526, 711, 824</td>
</tr>
<tr>
<td>gridded layout statement</td>
<td></td>
<td>499, 951</td>
</tr>
<tr>
<td>GROUPMAX= option</td>
<td>ODS GRAPHICS statement</td>
<td>346</td>
</tr>
<tr>
<td>GRSEGs</td>
<td></td>
<td>213</td>
</tr>
<tr>
<td>copying to catalogs</td>
<td></td>
<td>213</td>
</tr>
<tr>
<td>GTITLE option</td>
<td>ODS CHTML statement</td>
<td>188</td>
</tr>
<tr>
<td>ODS E PUBL statement</td>
<td></td>
<td>224</td>
</tr>
<tr>
<td>ODS EPUB3 statement</td>
<td></td>
<td>246</td>
</tr>
<tr>
<td>ODS EXCEL statement</td>
<td></td>
<td>302, 908</td>
</tr>
<tr>
<td>ODS HTML statement</td>
<td></td>
<td>131, 413</td>
</tr>
<tr>
<td>ODS HTML3 statement</td>
<td></td>
<td>384</td>
</tr>
<tr>
<td>ODS HTML5 statement</td>
<td></td>
<td>454</td>
</tr>
<tr>
<td>ODS MARKUP statement</td>
<td></td>
<td>526</td>
</tr>
<tr>
<td>ODS PCL statement</td>
<td></td>
<td>592</td>
</tr>
<tr>
<td>ODS PDF statement</td>
<td></td>
<td>607</td>
</tr>
<tr>
<td>ODS PHTML statement</td>
<td></td>
<td>711</td>
</tr>
<tr>
<td>ODS POWERPOINT statement</td>
<td></td>
<td>653</td>
</tr>
<tr>
<td>ODS PS statement</td>
<td></td>
<td>758</td>
</tr>
<tr>
<td>ODS Tagset statement</td>
<td></td>
<td>824</td>
</tr>
<tr>
<td>ODS TAGSETS.RTF statement</td>
<td></td>
<td>844</td>
</tr>
<tr>
<td>GTITLE= option</td>
<td>ODS PRINTER statement</td>
<td>732</td>
</tr>
<tr>
<td>ODS RTF statement</td>
<td></td>
<td>773</td>
</tr>
</tbody>
</table>

**H**

HEAD tags 131, 189, 385, 414, 455, 526, 711, 824

HEADTEXT option

ODS CHTML statement 189
ODS HTML statement 131, 414
ODS HTML3 statement 385
ODS HTML5 statement 455
ODS PHTML statement 711
ODS Tagset statement 824

HEADTEXT= option

ODS MARKUP statement 526

HEIGHT

ODS LAYOUT ABSOLUTE statement 475, 927

HEIGHT option

ODS LAYOUT GRIDDED statement 492, 944

ODS REGION statement, absolute 486, 937
ODS REGION statement, gridded 501, 953

HEIGHT= option

ODS GRAPHICS statement 346

HREFTARGET= style attribute 1117

HTML destination 39

body files 1206
contents file 1209
files produced by 1206
frame files 1209
links produced by 1201
output for 743
page files 1209
references produced by 1201

HTML files

appending to 151, 433
location of 136, 419

HTML links 1201

definition 1201
implementing 1201
ODS construction of 1203

HTML output

3.2 369
applet for viewing 118, 400
base text 119, 402
character set for META declaration 121, 181, 376, 404, 447, 516, 702, 815
creating 14
inserting text 140, 422
record separator 137, 419
sample 8
separate body file per page of output 147, 429
simple form 695

HTML references 1201
Index 1241

definition 1201
implementing 1201
ODS construction of 1203
HTML tagset 114, 396
HTML version setting 48
HTML3 destination 369
HTML4 tagset 811
HTML5 441
SVG Mode 469, 471
HTMLCLASS= style attribute 1107
HTMLCONTENTTYPE= style attribute 1109
HTMLDOCTYPE= style attribute 1111
HTMLID= style attribute 1117
HTMLSTYLE= style attribute 1117

I
ID option
ODS CHTML statement 189
ODS CSVALL statement 204
ODS EPUB statement 224
ODS EPUB3 statement 246
ODS EXCEL statement 302
ODS HTML statement 132, 414
ODS HTML3 statement 385
ODS HTML5 statement 455
ODS PHTML statement 712
ODS Tagset statement 825
ODS TAGSETS.RTF statement 844
ID= option
ODS MARKUP statement 526
ODS PCL statement 593
ODS PDF statement 607
ODS PRINTER statement 733
ODS PS statement 758
ODS RTF statement 774
image filenames 348
substituting BY values 361
image files
resetting index counter 352
image format 351
IMAGE function 281
IMAGE_DPI option
ODS EPUB statement 225
ODS EPUB3 statement 246
ODS EXCEL statement 301, 302, 908
ODS POWERPOINT statement 653
ODS TAGSETS.RTF statement 845
ODS WORD statement 908
IMAGE_DPI= option
ODS LISTING statement 508
ODS MARKUP statement 527
ODS RTF statement 774
IMAGE= style attribute 1118
IMAGEMAP option
ODS GRAPHICS statement 347
IMAGEMAP= option
ODS GRAPHICS statement 347
IMAGENAME= option
ODS GRAPHICS statement 348
substituting BY values 361
INDEX= style attribute 1129
index counter
resetting 352
inheritance
example programs 1193
style elements and 1045
inline formatting 274, 277
escape characters for 273
nested 275
Unicode symbols 276
inline formatting functions 278
inline style attributes
nesting and 275
integrating output with applications 13
ITALIC function 282

J
Java devices
CODEBASE file path 123, 182, 377, 405, 517, 704, 817
JUST= style attribute 1128
justification
numeric columns 211

K
KEEPN option
ODS RTF statement 774
KEYWORDS option
ODS EXCEL statement 302, 911
ODS POWERPOINT statement 654
ODS WORD statement 908
KEYWORDS= option
ODS PDF statement 608
ODS PRINTER statement 733
ODS PS statement 759
KPISKIN= style attribute 1118

L
LABEL= attribute suboption
FILE PRINT ODS statement 71, 167
LABEL= option
ODS TRACE statement 555, 892
LABEL= suboption
FILE PRINT ODS statement 67, 163
LABELMAX= option
ODS GRAPHICS statement 348
LABELPLACEMENT= option
ODS GRAPHICS statement 349
labels
  for columns 67, 71, 163, 167
  for output objects 68, 164
LASTPAGE function 282
LAYOUT ABSOLUTE statement 474, 925
LAYOUT destination 485, 936
LAYOUT END 504, 956
LAYOUT END statement 504, 956
LAYOUT GRIDDED 489, 940
LAYOUT GRIDDED statement 489, 940
LAYOUT option
  ODS POWERPOINT statement 654
LEADERS function 282
LEFTMARGIN= style attribute 1120
LEGENDAREAMAX= option
  ODS GRAPHICS statement 349
LIBNAME 169
LIBNAME statement for SASEDOC 169
LIBNAME statement, SASEDOC 169
librefs
  assigning to ODS documents 171
  associating with output objects 169
line pointer controls
  ODS 62, 903
LINEPATTERNOBSMAX= option
  ODS GRAPHICS statement 349
LINESYMBOL= style attribute 1121
LINKCOLOR= style attribute 1119
links
  See HTML links
LISTENTRYANCHOR= style attribute 1119
LISTING destination 38
  closing 505
  excluding output objects 505
  managing 504
  opening 504
  selecting output objects 505
  writing selection/exclusion lists to log 505
  writing trace records to 892
LISTING option
  ODS TRACE statement 892
LISTING output
  creating 15
LISTSTYLEIMAGE= string 1119
LISTSTYLETYPE= style attribute 1119
LOESSMAXOBS= option
  ODS GRAPHICS statement 350
LOESSOBSMAX= option
  ODS GRAPHICS statement 350
log
  output object records 891
  writing selection/exclusion lists to 806
M
  MARGINBOTTOM= style attribute 1120
  MARGINLEFT= style attribute 1120
  MARGINRIGHT= style attribute 1120
  MARGINTOP= style attribute 1121
  MARKERSIZE= style attribute 1121
  MARKERSYMBOL= style attribute 1121
  MARKUP destination 39, 509
    closing 509, 511, 538
    excluding output objects 511
    opening 538
    selecting output objects 512
    markup files
      location of 531
      markup languages 509
  MATCH_ALL option
  ODS OUTPUT statement 556, 566
MathML
  ODS EPUB3 statement 253
  MATHML function 283
MAXLEGENDAREA= option
  ODS GRAPHICS statement 349
MAXOBS= option
  ODS GRAPHICS statement 351
MEANS procedure
  ODS table names 1145
  MEAS_EVENT_MAP tagsets.rtf 856
  MEAS_SHORT_MAP tagsets.rtf 856
  MEAS_TEXT_MAP tagsets.rtf 856
META declaration
  character set for 121, 181, 376, 404, 447, 516, 702, 815
  META tags 132, 189, 385, 414, 456, 527, 712, 825
metadata 775, 846
  author 601, 726, 753, 768, 841
  string of keywords 608, 733, 759
  subject 611, 739
  title 612, 739, 779
METATEXT option
  ODS CHTML statement 189
  ODS HTML statement 132, 414
  ODS HTML3 statement 385
  ODS HTML5 statement 456
  ODS PHTML statement 712
  ODS Tagset statement 825
  METATEXT= option
    ODS MARKUP statement 527
N
  N= option
<table>
<thead>
<tr>
<th>ODS Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE PRINT ODS statement</td>
<td>63, 159</td>
</tr>
<tr>
<td>NAME= option</td>
<td>ODS GRAPHICS statement 355</td>
</tr>
<tr>
<td>ODS DOCUMENT statement</td>
<td>NOSUBPIXEL option</td>
</tr>
<tr>
<td>NAMED_DEST= option</td>
<td>ODS GRAPHICS statement 357</td>
</tr>
<tr>
<td>ODS PDF statement</td>
<td>NOTEREF function 284</td>
</tr>
<tr>
<td>ODS PRINTER statement</td>
<td>NOTOC option</td>
</tr>
<tr>
<td>ODS PS statement</td>
<td>ODS PDF statement 609</td>
</tr>
<tr>
<td>ODS PACKAGE statement</td>
<td>ODS PRINTER statement 734</td>
</tr>
<tr>
<td>NBINSMAX= option</td>
<td>NOW argument</td>
</tr>
<tr>
<td>ODS GRAPHICS statement</td>
<td>ODS EPUB statement 225</td>
</tr>
<tr>
<td>NBSPACE function</td>
<td>NOW option</td>
</tr>
<tr>
<td>ODS EXCLUDE statement</td>
<td>ODS EPUB3 statement 247</td>
</tr>
<tr>
<td>nested inline formatting</td>
<td>NOWARN option</td>
</tr>
<tr>
<td>inline style attributes and</td>
<td>ODS EXCLUDE statement 334</td>
</tr>
<tr>
<td>NEUTRALCOLOR= style attribute</td>
<td>numeric columns</td>
</tr>
<tr>
<td>NEWCHAPTER option</td>
<td>justification of 211</td>
</tr>
<tr>
<td>ODS EPUB3 statement</td>
<td>NXYBINSMAX= option</td>
</tr>
<tr>
<td>NEWFILE option</td>
<td>ODS GRAPHICS statement 350</td>
</tr>
<tr>
<td>ODS CHTML statement</td>
<td></td>
</tr>
<tr>
<td>ODS CSV ALL statement</td>
<td></td>
</tr>
<tr>
<td>ODS HTML statement</td>
<td></td>
</tr>
<tr>
<td>ODS HTML3 statement</td>
<td></td>
</tr>
<tr>
<td>ODS HTML5 statement</td>
<td></td>
</tr>
<tr>
<td>ODS PHTML statement</td>
<td></td>
</tr>
<tr>
<td>ODS Tagset statement</td>
<td></td>
</tr>
<tr>
<td>NEWFILE= option</td>
<td></td>
</tr>
<tr>
<td>ODS MARKUP statement</td>
<td></td>
</tr>
<tr>
<td>ODS PCL statement</td>
<td></td>
</tr>
<tr>
<td>ODS PDF statement</td>
<td></td>
</tr>
<tr>
<td>ODS PRINTER statement</td>
<td></td>
</tr>
<tr>
<td>ODS PS statement</td>
<td></td>
</tr>
<tr>
<td>ODS RTF statement</td>
<td></td>
</tr>
<tr>
<td>ODS TAGSETS-RTF statement</td>
<td></td>
</tr>
<tr>
<td>NEWLINE function</td>
<td></td>
</tr>
<tr>
<td>NOANTIALIAS option</td>
<td></td>
</tr>
<tr>
<td>ODS GRAPHICS statement</td>
<td></td>
</tr>
<tr>
<td>NOBREAKSPACE= style attribute</td>
<td></td>
</tr>
<tr>
<td>NOBORDER option</td>
<td></td>
</tr>
<tr>
<td>ODS GRAPHICS statement</td>
<td></td>
</tr>
<tr>
<td>NOFOOTNOTE option</td>
<td></td>
</tr>
<tr>
<td>ODS Excel statement</td>
<td></td>
</tr>
<tr>
<td>NOGFOOTNOTE option</td>
<td></td>
</tr>
<tr>
<td>ODS EXCLUDE statement</td>
<td></td>
</tr>
<tr>
<td>NOGTITLE option</td>
<td></td>
</tr>
<tr>
<td>ODS EXCLUDE statement</td>
<td></td>
</tr>
<tr>
<td>NOIMAGEMAP option</td>
<td></td>
</tr>
<tr>
<td>ODS GRAPHICS statement</td>
<td></td>
</tr>
<tr>
<td>OD NOVERIFY</td>
<td></td>
</tr>
<tr>
<td>ODS VERIFY statement</td>
<td></td>
</tr>
<tr>
<td>ODS <em>ALL</em> CLOSE statement</td>
<td></td>
</tr>
<tr>
<td>ODS (Output Delivery System)</td>
<td></td>
</tr>
<tr>
<td>DATA step and</td>
<td>DATA step examples 73</td>
</tr>
<tr>
<td>DATA step examples 73</td>
<td>how it works 31</td>
</tr>
<tr>
<td>DATA step examples 73</td>
<td>multiple output formats 16</td>
</tr>
<tr>
<td>how it works 31</td>
<td>processing 31</td>
</tr>
<tr>
<td>multiple output formats 16</td>
<td>quick start 13</td>
</tr>
<tr>
<td>processing 31</td>
<td>registry and 44</td>
</tr>
<tr>
<td>quick start 13</td>
<td>reports with DATA step 58</td>
</tr>
<tr>
<td>registry and 44</td>
<td>samples 4</td>
</tr>
<tr>
<td>reports with DATA step 58</td>
<td></td>
</tr>
<tr>
<td>samples 4</td>
<td>ODS argument</td>
</tr>
<tr>
<td>FILE PRINT ODS statement</td>
<td>63, 159</td>
</tr>
<tr>
<td>ODS block 13</td>
<td></td>
</tr>
<tr>
<td>ODS CHTML statement 174</td>
<td></td>
</tr>
<tr>
<td>actions 176</td>
<td></td>
</tr>
<tr>
<td>actions 176</td>
<td>options 177</td>
</tr>
<tr>
<td>options 177</td>
<td></td>
</tr>
<tr>
<td>ODS column pointer controls</td>
<td>60, 902</td>
</tr>
<tr>
<td>ODS CHTML statement 174</td>
<td></td>
</tr>
<tr>
<td>actions 201</td>
<td></td>
</tr>
<tr>
<td>actions 201</td>
<td>options 202</td>
</tr>
<tr>
<td>options 202</td>
<td></td>
</tr>
<tr>
<td>ODS DECIMAL_ALIGN statement 211</td>
<td></td>
</tr>
<tr>
<td>ODS destinations</td>
<td></td>
</tr>
<tr>
<td>categories of 36</td>
<td>changing default settings 49, 53</td>
</tr>
<tr>
<td>changing default settings 49, 53</td>
<td>changing default value 51, 52</td>
</tr>
<tr>
<td>changing default value 51, 52</td>
<td>closing 173</td>
</tr>
<tr>
<td>closing 173</td>
<td>default devices 506, 520, 771</td>
</tr>
</tbody>
</table>
destination-independent input 37
excluding output objects 332
exclusion lists 42, 336, 796
SAS formatted destinations 37
selecting output objects for 793
selection lists 42, 336, 796
specifying multiple 538
system resources and 41
tagset keywords as 538
tagset names as 546
third-party formatted destinations 38
two-level tagset names as 539
ODS DOCUMENT statement 212
ODS documents
librefs for 171
ODS EPUB statement 215
ODS EPUB2 statement 217
ODS EPUB3 statement 237
ODS ESCAPECHAR statement 273
ODS ESCAPECHAR= statement
inline formatting functions for 278
ODS EXCEL statement 296, 904
ODS EXCLUDE statement 332
ODS GRAPHICS statement 340
for batch jobs 360
ODS HTML statement 114, 396
ODS HTML3 statement 369
actions 371
options 372
ODS HTML5 statement 441
HTML5 441
ODS LAYOUT 923
ODS LAYOUT overview 925
ODS line pointer controls 62, 903
ODS LISTING statement 504
actions 505
default devices 506
options 506
ODS MARKUP statement 509
actions 509, 511
creating XML files 539
creating XML files and DTD 542
default devices 520
details 538
examples 509
multiple markup output 208, 543
options 512
tagset names as ODS destinations 546
ODS NO_DECIMAL_ALIGN statement 554
ODS output
See also SAS Studio
assigning attributes to columns 82
DATA step enhanced features 59
formatting variables 63, 159
in SAS Studio 97
inserting text into 883
listing variables to include 63, 159
multiple formats 212
selected variables in 77
tracking in Results window 763
ODS OUTPUT statement 555
actions 555
arguments 555
creating data sets 559, 562, 566
detail examples 555
merging output objects into data set 558
ODS PACKAGE statement 576
options 579
ODS PATH statement 585
ODS PCL statement 587
multiple instances of same destination 593
options 589
ODS PDF statement 597
multiple instances of same destination 607
opening multiple instances of same
destination 617
opening/closing PDF destination 613
options 600
ODS PHTML statement 695
actions 697
options 698
ODS POWERPOINT statement 649
ODS PREFERENCES statement 694
ODS PRINTER statement 722
actions 725
details 740
host information 742
multiple instances of same destination 733
opening/closing PRINTER destination 740
options 725
output for HTML destination 743
output for PRINTER destination 743
printing output directly to printers 741
Windows and 742
without actions or options 725
ODS PROCLABEL statement 748
ODS PROCTITLE statement 748
ODS PS statement 749
multiple instances of same destination 758
options 752
ODS PUT statement 59, 900
ODS Region
absolute layout 485, 936
ODS Region statement, absolute 485, 936
ODS RESULTS statement 763
ODS RTF statement 764
actions 767
default devices 771
graphics and 781
opening/closing RTF destination 780
options 767
RTF output 780
ODS SELECT statement 793
ODS SHOW statement 806
ODS statements
category descriptions 110
DATA step statements 107
definition of 107
global statements 107
Output Control statements 108
procedure statements 108
SAS formatted statements 108
third-party formatted statements 108
ODS table names
ARIMA procedure 1161
Base SAS procedures 1141
CALENDAR procedure 1141
CATALOG procedure 1141
CHART procedure 1142
COMPARE procedure 1142
CONTENTS procedure 1144
CORR procedure 1143
DATASETS procedure 1144
FREQ procedure 1145
MEANS procedure 1145
PLOT procedure 1152
REPORT procedure 1152
SAS/ETS procedures 1161
SQL procedure 1152
SUMMARY procedure 1145
TABULATE procedure 1153
TIMEPLOT procedure 1153
UNIVARIATE procedure 1153
ODS Tagset statement
actions 809
arguments 810
options 811
ODS TAGSET statement 807
ODS TAGSET.RTF statement
graphics and 857
ODS TAGSETS.RTF 854
ODS TAGSETS.RTF statement 838
ODS TAGSETS.RTF suboptions 846
ODS TEXT= statement 883
ODS TRACE statement 891
contents of trace record 892
example 894
LABEL= option 555
specifying output objects 893
ODS USEGOPT statement 896
ODS VERIFY statement 899
ODS WORD
Overview 911
ODSDEST= system option
restoring defaults 959
ODSGRAPHICS= system option
restoring defaults 960
ODSSTYLE= system option
restoring defaults 961
OFF option
ODS TRACE statement 891
ODS VERIFY statement 900
ON option
ODS TRACE statement 892
ODS VERIFY statement 900
OPERATOR option
ODS TAGSETS.RTF statement 846
OPERATOR= option
ODS RTF statement 775
options
ODS CHTML statement 177
ODS CSVALL statement 202
ODS EPUB3 statement 247
ODS EXCEL statement 303, 909
ODS HTML3 statement 372
ODS LISTING statement 506
ODS MARKUP statement 511, 512
ODS PHTML statement 698
ODS RTF statement 767
OPTIONS option
ODS CHTML statement 190
ODS CSVALL statement 205
ODS HTML statement 133, 416
ODS HTML3 statement 387
ODS HTML5 statement 457
ODS MARKUP statement 528
ODS PHTML statement 713
ODS POWERPOINT statement 655
ODS Tagset statement 826
ODS TAGSETS.RTF statement 846
OPTIONS options
ODS EPUB statement 225
ORDER_TYPE option
ODS LAYOUT GRIDDED statement 492, 944
ORPHAN= style attribute 1122
OUTLINECOLOR = style attribute 1122
OUTLINESTYLE= style attribute 1122
OUTLINEWIDTH= style attribute 1123
output
integrating with applications 13
OUTPUT argument
ODS EPUB statement 225
Output Control statements 108
OUTPUT destination 38
closing 555
exclusion lists 555
selection lists 555
output objects
creating 63, 159
customized output for 43
data sets from 555, 562
determining destinations for 41, 555
excluding from LISTING destination 505
excluding from ODS destinations 332
hierarchy of 212
labels for 68, 164
librefs 169
merging dissimilar objects into data set 558
names for 67, 163
records in log 891
selecting for LISTING destination 505
selecting for ODS destinations 793
sequence number of 170
specifying 893
tracing 891
OUTPUT option
ODS EPUB3 statement 247
ODS TRACE statement 892
output pointer
number of lines for 63, 159
output strings
interpreting 277
OUTPUTFMT= option
ODS GRAPHICS statement 351
OUTPUTHEIGHT= style attribute 1116, 1123
OUTPUTWIDTH= style attribute 1123, 1131
overflow-control option
FILE PRINT ODS statement 63, 159

P
package objects
adding to 576
closing 576
opening 576
publishing 576
PACKAGE option
MARKUP statement 529
ODS CHTML statement 191
ODS CSVALL statement 206
ODS HTML statement 134, 416
ODS HTML3 statement 387
ODS HTML5 statement 459
ODS PHTML statement 714
ODS PRINTER statement 734
ODS RTF statement 775
ODS TAGSETS.RTF statement 851
PADDING= style attribute 1123
PADDINGBOTTOM= style attribute 1123
PADDINGLEFT= style attribute 1123
PADDINGRIGHT= style attribute 1124
PADDINGTOP= style attribute 1124
PAGE argument
ODS EPUB statement 225
page breaks 594, 610, 737, 761
RTF output 778
splitting tables at 774
page files 1209
PAGE option
ODS CHTML statement 191
ODS EPUB3 statement 247
ODS HTML statement 134, 417
ODS HTML3 statement 387
ODS HTML5 statement 459
ODS LISTING statement 506
ODS MARKUP statement 529
ODS PHTML statement 714
ODS Tagset statement 827
PAGEBREAKHTML= style attribute 1124
PAGEOF function 284
PAGEPANELS option
ODS TAGSETS.RTF statement 851
PANELCELLMAX= option
ODS GRAPHICS statement 351
PARAMETERS option
ODS CHTML statement 193
ODS HTML statement 136, 418
ODS HTML3 statement 389
ODS HTML5 statement 461
ODS MARKUP statement 530
ODS PHTML statement 716
ODS Tagset statement 828
PARAMETERS= option
ODS MARKUP statement 530
PATH option
ODS CHTML statement 193
ODS CSVALL statement 206
ODS HTML statement 136, 419
ODS HTML3 statement 389
ODS HTML5 statement 461
ODS PHTML statement 716
ODS Tagset statement 829
ODS TAGSETS.RTF statement 851
PATH= option
ODS MARKUP statement 531
ODS PACKAGE statement 580
ODS RTF statement 776
PCL 587
PCL destination
closing 596
opening 596
PCL option
ODS PRINTER statement 735
PCL output 735
PDF 597
PDF destination
  closing 613
  opening 613
  opening multiple instances 617
PDF files
  adding notes 609, 736, 760
  compressing 604, 729
  list of bookmarks 601, 602, 727, 753, 754
PDF function 284
PDF option
  ODS PRINTER statement 735
PDF output 735
  inserting text 612
  sample 9
PDFMARK option
  ODS PRINTER statement 735
  ODS PS statement 760
PDFNOTE option
  ODS PDF statement 609
  ODS PRINTER statement 736
  ODS PS statement 760
PDFTOC= option
  ODS PDF statement 609
  ODS PRINTER statement 736
PHTML destination
  PHTML output 695
  PHTML tagset 811
PLOT procedure
  ODS table names 1152
  pointers
    past end of line 62, 903
POP option
  about 367
  ODS GRAPHICS statement 352
POSTHTML= style attribute 1124
POSTIMAGE= style attribute 1124
PostScript files
  tags for Acrobat Distiller 735, 760
PostScript output 737
  sample 10
POSTSCRIPT option
  ODS PRINTER statement 737
POSTTEXT= style attribute 1125
PowerPoint 649
PowerPoint destination 39
PREFERENCES destination 694
PREHTML= style attribute 1125
PREIMAGE= style attribute 1125
PREPAGE= option
  ODS RTF statement 776
  ODS TAGSETS.RTF statement 852
PREPEND option
  ODS PATH statement 586
  PRETEXT= style attribute 1126
PRINT argument
  FILE PRINT ODS statement 63, 159
  PRINTER 722
PRINT option
  ODS PRINTER statement 735
Printer output
  footnotes 732
  titles 732
PRINTER output
  inserting text 739
PRINTER= option
  ODS PRINTER statement 736
PROC argument
  ODS EPUB statement 225
PROC option
  ODS EPUB3 statement 247
  procedure statements 108
  procedures
    creating data sets from output objects 562
    ODS table names, Base SAS 1141
    ODS table names, SAS/ETS 1161
    style templates with 971
    title in output 748
    PROTECTSPECIALCHARS= style attribute 1126
PRT= option
  ODS PRINTER statement 736
PS 749
PS destination
  closing 763
  opening 763
PS option
  ODS PRINTER statement 737
PS output
  inserting text 762
PUSH option
  about 367
  ODS GRAPHICS statement 352
PUT 59, 900
PUT statement
  ODS 59
<table>
<thead>
<tr>
<th>R</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RAW function</td>
<td>285</td>
</tr>
<tr>
<td>RECORD_SEPARATOR</td>
<td>option</td>
</tr>
<tr>
<td>ODS HTML statement</td>
<td>194</td>
</tr>
<tr>
<td>ODS CSVALL statement</td>
<td>207</td>
</tr>
<tr>
<td>ODS HTML5 statement</td>
<td>462</td>
</tr>
<tr>
<td>ODS PHTML statement</td>
<td>717</td>
</tr>
<tr>
<td>ODS Tagset statement</td>
<td>829</td>
</tr>
<tr>
<td>ODS TAGSETS.RTF statement</td>
<td>852</td>
</tr>
<tr>
<td>RECORD_SEPARATOR= option</td>
<td></td>
</tr>
<tr>
<td>ODS HTML statement</td>
<td>137, 419</td>
</tr>
<tr>
<td>ODS MARKUP statement</td>
<td>532</td>
</tr>
<tr>
<td>ODS RTF statement</td>
<td>776</td>
</tr>
<tr>
<td>RECSEP= option</td>
<td></td>
</tr>
<tr>
<td>ODS HTML statement</td>
<td>137, 419</td>
</tr>
<tr>
<td>ODS MARKUP statement</td>
<td>532</td>
</tr>
<tr>
<td>RTF files</td>
<td></td>
</tr>
<tr>
<td>location of</td>
<td>776, 851</td>
</tr>
<tr>
<td>record separator</td>
<td>776</td>
</tr>
<tr>
<td>style definitions</td>
<td>778</td>
</tr>
<tr>
<td>time and date of SAS program</td>
<td>777</td>
</tr>
<tr>
<td>RTF markup files</td>
<td></td>
</tr>
<tr>
<td>style definitions</td>
<td>854</td>
</tr>
<tr>
<td>RTF output</td>
<td>764, 780, 856</td>
</tr>
<tr>
<td>footnotes</td>
<td>773</td>
</tr>
<tr>
<td>graphics</td>
<td>781</td>
</tr>
<tr>
<td>image resolution</td>
<td>774</td>
</tr>
<tr>
<td>inserting text</td>
<td>779</td>
</tr>
<tr>
<td>page breaks</td>
<td>778</td>
</tr>
<tr>
<td>sample</td>
<td>12</td>
</tr>
<tr>
<td>splitting tables at page breaks</td>
<td>774</td>
</tr>
<tr>
<td>titles</td>
<td>773</td>
</tr>
<tr>
<td>translation tables</td>
<td>780</td>
</tr>
<tr>
<td>RULES= style attribute</td>
<td>1127</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SAS Explorer window</td>
<td></td>
</tr>
<tr>
<td>list of available styles</td>
<td>836</td>
</tr>
<tr>
<td>SAS formatted destinations</td>
<td>36, 37</td>
</tr>
<tr>
<td>SAS formatted statements</td>
<td>108</td>
</tr>
<tr>
<td>SAS Graphics Editor (SGE)</td>
<td></td>
</tr>
<tr>
<td>ODS HTML statement</td>
<td>138, 420</td>
</tr>
<tr>
<td>ODS HTML5 statement</td>
<td>463</td>
</tr>
<tr>
<td>SAS Studio</td>
<td>about</td>
</tr>
<tr>
<td>customized output</td>
<td>100</td>
</tr>
<tr>
<td>default output</td>
<td>98</td>
</tr>
<tr>
<td>output environment</td>
<td>98</td>
</tr>
<tr>
<td>SAS/ETS procedures</td>
<td></td>
</tr>
<tr>
<td>ODS table names</td>
<td>1161</td>
</tr>
<tr>
<td>SASDATE option</td>
<td></td>
</tr>
<tr>
<td>ODS EXCEL statement</td>
<td>318, 909</td>
</tr>
<tr>
<td>ODS POWERPOINT statement</td>
<td>661</td>
</tr>
<tr>
<td>ODS RTF statement</td>
<td>777</td>
</tr>
<tr>
<td>ODS WORD statement</td>
<td>910</td>
</tr>
<tr>
<td>SASEDOC argument</td>
<td></td>
</tr>
<tr>
<td>LIBNAME statement</td>
<td>170</td>
</tr>
</tbody>
</table>
SASEDOC engine
  LIBNAME statement with 169
  SCALE option
  ODS GRAPhICS statement 355
  SCALE= option
  ODS GRAPHICS statement 355
  SCALEMARKERS= option
  ODS GRAPHICS statement 355
  SELECT action
  ODS CSVALL statement 201
  ODS DOCUMENT statement 213
  ODS LISTING statement 505
  ODS MARKUP statement 512
  ODS PCL statement 588
  ODS PDF statement 599
  ODS PRINTER statement 725
  ODS RTF statement 767
  SELECT options
    ODS TAGSETS.RTF statement 841
  selection lists 42, 336, 796
  destinations for output objects 41
  multiple procedure steps with 797
  OUTPUT destination 555
  writing to log 806
  SGE
    ODS HTML statement 138, 420
    ODS HTML5 statement 463
  SGE= option
    ODS LISTING statement 508
  SHORT_MAP tagset 833
  SHOW action
    ODS CSVALL statement 201
    ODS DOCUMENT statement 213
    ODS LISTING statement 505
    ODS MARKUP statement 512
    ODS OUTPUT statement 555
    ODS PCL statement 588
    ODS PDF statement 599
    ODS PRINTER statement 725
    ODS PS statement 752
    ODS RTF statement 767
  SHOW argument
    ODS OUTPUT statement 558
  SHOW option
    ODS CHTML statement 176
    ODS EPUB statement 220
    ODS EPUB3 statement 240
    ODS EXCEL statement 299
    ODS GRAPHICS statement 356
    ODS HTML statement 117, 399
    ODS HTML3 statement 371
    ODS HTML5 statement 444
    ODS PHTML statement 698
    ODS POWERPOINT statement 652
    ODS Tagset statement 809
    ODS TAGSETS.RTF statement 841
    ODS WORD statement 906
  SIGMA function 285
  smoothing graphics 342
  SQL procedure
    ODS table names 1152
  STACKDEPTHMAX= option
    ODS GRAPHICS statement 357
  STARTCOLOR= style attribute 1127
  STARTPAGE option
    ODS TAGSETS.RTF statement 853
  STARTPAGE= option
    ODS PCL statement 594
    ODS PDF statement 610
    ODS PRINTER statement 737
    ODS PS statement 761
    ODS RTF statement 778
    ODS WORD statement 910
  statement 59, 900
    EPUB 215
    EPUB2 217
    EPUB3 237
    FILE 63, 159
    GRIDDED LAYOUT 504, 956
    LAYOUT ABSOLUTE 474, 504, 925, 956
    LAYOUT GRIDDED 489, 940
    LIBNAME 169
    PCL 587
    PDF 597
    PRINTER 722
    PS 749
  statements
    REGION 499, 951
  STATIC option
    ODS GRAPHICS statement 351
  STATUS option
    ODS EXCEL statement 318, 909
    ODS POWERPOINT statement 663
  STATUS= option
    ODS WORD statement 911
  STRONG function 285
  style attribute
    tagattr 322, 325
style attributes 40
  color 1132
  data values 1073
  definition 965
  dimension 1134
  font definition 1134
  format 1136
  reference 1136
  table of 1073
  values of 1073
style definitions
  RTF files 778, 854
style elements
  definition 965
  inheritances of 1045
  TABULATE procedure 965
STYLE function 285
  STYLE option
    ODS CHTML statement 195
    ODS EXCEL statement 318, 909, 910
    ODS HTML statement 138, 420
    ODS HTML3 statement 391
    ODS HTML5 statement 463
    ODS LAYOUT ABSOLUTE statement 476, 495, 927, 946
    ODS LAYOUT REGION, Absolute statement 486, 938
    ODS LAYOUT REGION, Gridded statement 503, 955
    ODS PHTML statement 718
    ODS Tagset statement 830
    ODS TAGSETS.RTF statement 854
  style overrides
    ODS EPUB statement 228
  style templates
    definition of 965
    ODS EPUB3 statement 251
    ODS MARKUP statement 532
    ODS PCL statement 595
    ODS PDF statement 611
    ODS POWERPOINT statement 663
    ODS PRINTER statement 738
    ODS PS statement 762
    procedures with 971
    verifying values 899
    viewing contents of 975
  STYLE_DISPLAY tagset 833
  STYLE_POPUP tagset 833
  STYLE_PREFIX option
    ODS WORD statement 910
  STYLE_VISIBILITY option
    ODS Word statement 910
  STYLE= option
    ODS MARKUP statement 532
    ODS PCL statement 595
    ODS PDF statement 611
  ODS PRINTER statement 738
  ODS PS statement 762
  ODS RTF statement 778
STYLESET option
  ODS CHTML statement 195
  ODS EPUB statement 228
  ODS EPUB3 statement 251
  ODS HTML statement 138, 421
  ODS HTML3 statement 391
  ODS HTML5 statement 464
  ODS PHTML statement 718
  ODS Tagset statement 831
  STYLESET= option
    ODS MARKUP statement 533
  SUB function 286
  SUBJECT= option
    ODS PDF statement 611
    ODS PRINTER statement 739
    suboptions
      ODS HTML5 statement 465
    SUBPIXEL option
    ODS GRAPHICS statement 357
  SUMMARY procedure
    ODS table names 1145
    SUPER function 286
T
  TABLE argument
    ODS EPUB statement 225
  table of contents
    ODS PDF statement 604, 609
    ODS PRINTER statement 729, 734
  TABLE option
    ODS EPUB3 statement 247
    table templates
      binding data components to 63, 159
      specifying 68, 164
      user-defined templates 89
      verifying values 899
  TABLEROWS option
    ODS TAGSETS.RTF statement 854
  tables
    splitting at page breaks 774
    uniformity across pages 596, 612, 740, 762
  tabular output 200
  TABULATE procedure
    ODS table names 1153
  style elements 965
  tag attributes
    for dynamic graphics 514
    TAGATTR= style attribute 1127
  TAGSET.RTF output
    graphics 857
    TAGSET= option
ODS MARKUP statement  534
tagsets 39
CHTML  810
CSV  810
CSVALL  810
CSVBYLINE  810
EVENT_MAP  833
HTML  114, 396
HTML4  811
keyword values for  534
keywords as ODS destinations  538
list of  32
NAMEDHTML  833
names as ODS destinations  546
PHTML  811
SHORT_MAP  833
statement  807
STYLE_DISPLAY  833
STYLE_POPUP  833
TEXT_MAP  833
two-level names as ODS destinations  539
user-defined  811
XML  810
tagsets.rtf
MEAS_EVENT_MAP  856
MEAS_SHORT_MAP  856
MEAS_TEXT_MAP  856
TEMPLATE procedure
list of available styles  836
locations for definitions  585
search order for definitions  585
user-defined table template template  89
TEMPLATE= suboption
FILE PRINT ODS statement  68, 164
text
inserting into ODS output  883
TEXT option
ODS CHTML statement  197
ODS EXCEL statement  318
ODS HTML statement  140, 422
ODS HTML3 statement  393
ODS HTML5 statement  465
ODS MARKUP statement  535
ODS PHTML statement  720
ODS Tagset statement  832
ODS TAGSETS.RTF statement  855
TEXT_MAP tagset  833
TEXT= option
ODS MARKUP statement  521
ODS PCL statement  595
ODS PDF statement  612
ODS PRINTER statement  739
ODS PS statement  762
ODS RTF statement  779
TEXTALIGN= style attribute  1128
TEXTDECORATION= style attribute  1129
TEXTINDENT= style attribute  1129
TEXTJUSTIFY= style attribute  1129
third-party formatted destinations  38
definition  36
formatting control and  40
third-party formatted statements  108
THISPAGE function  287
TICKDISPLAY= style attribute  1129
TIMEPLOT procedure
ODS table names  1153
TIPMAX= option
ODS GRAPHICS statement  358
TITLE option
ODS EPUB statement  229
ODS EPUB3 statement  251
ODS EXCEL statement  318
ODS POWERPOINT statement  663
TITLE= option
ODS PDF statement  612
ODS PRINTER statement  739
ODS RTF statement  779
ODS WORD statement  911
titles
in file metadata  612, 739, 779
in graphics output  131, 188, 384, 413, 454, 526, 711, 824
Printer output  732
procedure titles in output  748
RTF output  773
TOC_DATA option
ODS WORD statement  910
TOC_DATA= option
ODS RTF statement  779
TOC_LEVEL option
ODS WORD statement  910
TOC_TYPE option
ODS WORD statement  910
TOCENTRYINDEX function  287
TOCENTRYPAGE function  287
TOPMARGIN= style attribute  1121
TOTALCELLMAX= option
ODS GRAPHICS statement  359
trace records  891, 892
trailing @
PUT_ODS_statement  60, 901
translation tables
ODS MARKUP statement  535
RTF output  780
TRANSPARENCY= style attribute  1130
TRANTAB option
ODS CHTML statement  197, 207
ODS HTML statement  140, 423
ODS HTML3 statement  393
ODS HTML5 statement  465
U
UNICODE function 287
Unicode symbols 276
UNIFORM option
  ODS PCL statement 596
  ODS PDF statement 612
  ODS PRINTER statement 740
  ODS PS statement 762
  ODS TAGSETS.RTF statement 855
UNIVARIATE procedure
  ODS table names 1153
UNIX
  printing output directly to printer 741
URL= option
  ODS LISTING statement 508
URL= style attribute 1130
user-defined tagsets 811

V
VARIABLES= suboption
  FILE PRINT ODS statement 69, 165
VERTICALALIGN= style attribute 1130
VIDEO function 288
VISITEDLINKCOLOR= style attribute 1131
VJUST= style attribute 1130

W
WATERMARK= style attribute 1131
WHERE option
  ODS EXCLUDE statement 335
WIDTH option
  ODS LAYOUT ABSOLUTE statement 476, 927
  ODS LAYOUT GRIDDED statement 495, 946
  ODS REGION statement, absolute 487, 938
  ODS REGION statement, gridded 503, 955
WIDTH= option
  ODS GRAPHICS statement 359
  WIDTH= style attribute 1131
WINDOW= style attribute 1132
Windows
  ODS PRINTER statement with 742
  printing output directly to printer 741
WORK option 663
  ODS EPUB 229
  ODS EPUB3 251
  ODS EXCEL statement 318
  WORK= option
  ODS WORD statement 911

X
X option
  ODS LAYOUT ABSOLUTE statement 476, 928
  ODS LAYOUT GRIDDED statement 495, 947
  ODS REGION statement, absolute 487, 939
XML files
  creating 539
  creating, with DTD 542
XML tagset 810

Y
Y option
  ODS LAYOUT ABSOLUTE statement 477, 928
  ODS LAYOUT GRIDDED statement 496, 947
  ODS REGION statement, absolute 488, 939

Z
z/OS
  printing output directly to printer 741