Creating Accessible SAS® 9.4 Output Using ODS and ODS Graphics
## Contents

*What’s New for Creating Accessible Output* ................................................................. v

**Chapter 1 / Overview and General Recommendations** ........................................... 1
  - Overview of Creating Accessible Output .................................................................. 1
  - General Recommendations for Creating Accessible Output ................................. 3
  - Overview of Accessible PDF Documents .............................................................. 5
  - Configure SAS Studio for Accessible Output ......................................................... 5

**Chapter 2 / Accessibility System Options** ............................................................... 7
  - Overview of the Accessibility System Options ...................................................... 7
  - Accessibility System Option Syntax ..................................................................... 8

**Chapter 3 / General Information about Accessible Tables** .................................... 11
  - Overview of Table Accessibility .......................................................................... 11
  - General Recommendations for Tables ................................................................... 11

**Chapter 4 / Creating Accessible Tables with the PRINT Procedure** ..................... 13
  - About the PRINT Procedure ................................................................................ 13
  - Recommendations for the PRINT Procedure ...................................................... 14
  - Example: Simple Table with a Title ..................................................................... 15
  - Example: Table with the Observation Numbers ................................................ 16
  - Example: Table with a Descriptive Row Heading .............................................. 18
  - Example: Table with a Descriptive Summation Row .......................................... 19
  - Example: Table with Labels for Multiple Summation Rows ............................... 21
  - Example: Use the N Argument to Display the Number of Results ..................... 23

**Chapter 5 / Creating Accessible Tables with the REPORT Procedure** ..................... 25
  - About the REPORT Procedure ............................................................................. 25
  - Recommendations for the REPORT Procedure ................................................... 26
  - Example: Detail Report with a Table Description ............................................... 27
  - Example: Tables with Spanned Column Headers .............................................. 28
  - Example: Detail Report with Ordering and SPANROWS ..................................... 30
  - Example: Summarization Row with Custom Labels .......................................... 32
  - Example: Detail Report with ACCESSIBLETABLE System Option .................... 34
  - Example: Accessible Table Using By Directives .............................................. 36

**Chapter 6 / Creating Accessible Tables with the TABULATE Procedure** ............. 39
  - About the TABULATE Procedure ....................................................................... 40
  - Recommendations for the TABULATE Procedure ............................................. 40
  - Example: Simple Table with No Row Dimension .............................................. 41
  - Example: Table with a Single Row Dimension and No Label ............................ 42
  - Example: Table with a Single Row Dimension Whose Label Is in a Box .......... 44
  - Example: Table with Multiple ALL Columns ..................................................... 45
  - Example: Table with Summation Rows and Custom Labels .............................. 46
  - Example: Enable Accessible Tables ................................................................... 49
  - Example: Table with BY Directives .................................................................. 50
| Chapter 7 / Creating Accessible Tables and Reports with the Report Writing Interface | 53 |
| Create an Accessible Table | 53 |
| Create a Simple Accessible Report | 57 |
| Create an Accessible Report That Includes Text, Tables, and Graphs | 63 |

| Chapter 8 / Creating Accessible Graphs | 73 |
| Recommendations for Graphs | 74 |
| Inform Users about the SAS Graphics Accelerator | 76 |
| Example: Simple Histogram | 76 |
| Example: Multi-Line Series Plot That Does Not Rely on Color Alone | 78 |
| Example: Grouped Bubble Plot That Shows Fill Patterns | 80 |
| Example: Grouped Bar Chart That Shows Fill Patterns and Does Not Use Color | 81 |
| Example: Scatter Plot with High-Visibility Markers and Axis Elements | 83 |
| Example: Bar Chart with Data Tips and Drill-Down Links Created with GTL | 84 |
| About the SAS Graphics Accelerator | 89 |

| Chapter 9 / Using ODS Styles to Create Accessible Output | 91 |
| About ODS Styles | 91 |
| About Attribute Rotation Patterns for Grouped Plots | 92 |
| Recommendation: Use Daisy with Color-Priority Attribute Rotation Disabled | 94 |
| Other ODS Styles That You Can Use | 96 |
| Cases in Which ODS Style Alone Might Not Be Sufficient | 100 |
| Enhancing the Appearance of the Link-Focus Indicator | 102 |

| Chapter 10 / Appendix: ODS Syntax Related to Accessibility | 105 |
| ODS HTML5 Statement Options Related to Accessibility | 105 |
| ODS PDF Statement Options Related to Accessibility | 106 |
| ODS GRAPHICS Statement Options Related to Accessibility | 106 |
| ODS Methods in the Report Writing Interface Related to Accessibility | 106 |
| ODS Procedures Related to Accessibility | 107 |
| ODS Style Attributes Related to Accessibility | 107 |

| Chapter 11 / Appendix: Input Data for Examples | 109 |
| Appendix: Input Data for Report Writing Interface and PROC TABULATE Examples | 109 |
What’s New for Creating Accessible Output

Enhancements In SAS 9.4M6

- New system options facilitate the creation of accessible content on a system-wide level. Here is a summary of the options:
  - The ACCESSIBLECHECK option checks your SAS programs for common violations of accessibility standards and writes messages to the SAS log if violations are found.
  - The ACCESSIBLEGRAPH option enables the ACCESSIBLE_GRAPH option in the ODS HTML5 destination by default.
  - The ACCESSIBLEPDF option enables the ACCESSIBLE option in the ODS PDF destination by default.
  - The ACCESSIBLETABLE option makes table captions visible and changes the layout of some tables to make them accessible.
    This option applies to tables generated by the PRINT, REPORT, and TABULATE procedures as well as the Report Writing Interface.
  - The CAPTION= option defines a visible table caption, which is displayed when the ACCESSIBLETABLE system option has been specified. The CAPTION= option applies to the REPORT and TABULATE procedures, and the ODS Report Writing Interface.
- Several enhancements affect PDF output. See Overview of Accessible PDF Documents on page 5.
  - ATTRPRIORITy=NONE is now the default setting for the Daisy ODS style. If you are creating a graph that contains groups or multiple plots, you do not need to specify ATTRPRIORITy=NONE in the ODS GRAPHICS statement. By default, the Daisy style rotates attributes using colors and other elements such as markers, line patterns, and fill patterns. See About Attribute Rotation Patterns for Grouped Plots on page 92.
  - In addition, the Daisy, HighContrast, and HighContrastLarge styles include an enhanced FOCUSLINK indicator. You can use these styles to enhance the appearance of the link-focus indicator in graphs that have active links. For a usage example, see Example: Bar Chart with Data Tips and Drill-Down Links Created with GTL on page 84.

Graph and Style Enhancements In SAS 9.4M5

- These enhancements apply to graphs:
  - Image maps are supported with SVG output using HTML5. The IMAGEMAP option is specified in the ODS GRAPHICS statement and enables data tips and drill-down generation for a graph. See Example: Bar Chart with Data Tips and Drill-Down Links Created with GTL on page 84.
You can specify line fill patterns for supported plot types with all styles. Previously, fill patterns were available only when you used certain gray-scale styles. You can also change the appearance of the fill patterns. See Example: Grouped Bubble Plot That Shows Fill Patterns on page 80.

You can customize the appearance of focus indicators. A new FOCUSLINK style element is available to be used in a style template. Three new style attributes enable you to customize the focus outline pattern, width, and color. See Enhancing the Appearance of the Link-Focus Indicator on page 102 and Example: Bar Chart with Data Tips and Drill-Down Links Created with GTL on page 84.

Documentation Enhancements

Creating Accessible SAS Output Using ODS and ODS Graphics was updated to include recommendations and examples for the following procedures: PRINT, REPORT, and TABULATE. This update was made to the SAS 9.4M5 release of the document.
Overview of Creating Accessible Output

Introduction
Accessibility is important for the output objects, such as text, tables, and charts, that you create for your users. This document describes how you can generate accessible output using SAS procedures in conjunction with the Output Delivery System (ODS) and ODS Graphics software that is available with Base SAS.

Note: This document is subject to periodic updates as new accessibility features become available and more SAS products are tested and verified. Check for updates on the SAS Graphics Accelerator page on the SAS Support site. In the HTML version of the document, the date on which the document was last updated appears at the bottom of each page. In the PDF version of the document, the date on which the document was last updated appears at the bottom of the last page.

Audience for This Document
This document was created for the following audience:

- SAS analysts or programmers within an organization who need to produce accessible content for the organization’s customer or user base.
- Accessibility experts within an organization who need to ensure that the output is accessible.
The document was created primarily for programmers who create content, such as a website, for a broad audience. The recommendations are intended to produce attractive and useful output for all users, including sighted users as well as those with low or no vision. The document avoids recommendations for a very narrow audience. For example, the document does not delve into high-contrast or large-element graph styles.

Guidelines Used for Measuring Accessibility Compliance

Accessibility compliance was measured using the W3C Web Content Accessibility Guidelines (WCAG), version 2.0. These guidelines can be found at https://www.w3.org/WAI/standards-guidelines/wcag/.

Note: The Section 508 Information and Communication Technology (ICT) Standards and Guidelines were updated to reference WCAG 2.0 in January 2017 with the publication of the final rule. As a result, the SAS software described in this document relies on WCAG 2.0 for accessibility principles, guidelines, and success criteria.

Third-Party Tools Used to Test Accessibility

The following third-party tools are recommended for testing output:

- Freedom Scientific JAWS screen reader
- WAVE Chrome extension
- aXe browser extension

See Also


General Recommendations for Creating Accessible Output

Recommendations

To create or modify accessible output, follow the recommendations listed here:

Use any of the following software to create your output:

- Use the PRINT procedure to print the observations in a SAS data set using all or some of the variables.
  See Creating Accessible Tables with the PRINT Procedure on page 13.
- Use the REPORT procedure to combine features of the PRINT, MEANS, and TABULATE procedures with
  features of the DATA step in a single report-writing tool that can produce a variety of reports.
  See Creating Accessible Tables with the REPORT Procedure on page 25.
- Use the TABULATE procedure to present descriptive statistics in tabular format, using some or all of the
  variables in a data set. You can create a variety of tables ranging from simple to highly customized. The
  TABULATE procedure computes many of the same statistics that are computed by other descriptive
  statistical procedures such as MEANS, FREQ, and REPORT.
  See Creating Accessible Tables with the TABULATE Procedure on page 39.
- Use the Report Writing Interface (RVI) to create customized reports in an object-oriented language that is
  fully integrated with ODS.
- Use the ODS TEXT procedure to generate headings that organize your content as well as any lists or
  static text, if applicable.
- Use the SG PLOT procedure to create single-cell graphs with a wide range of plot types that can be read
  and interpreted by the SAS Graphics Accelerator.
  For a list of plot types currently supported by SAS Graphics Accelerator, see the SAS Graphics
  Accelerator product page on the SAS Support site.
  Starting with SAS 9.4M6, you can use the SG PIE procedure to create pie and donut charts. This
  procedure is preproduction.
  You can also use the SGPANEL procedure to create classification panels, and the SGSCATTER
  procedure to create panned scatter plots. However, these graphs are not currently supported by the SAS
  Graphics Accelerator. In addition, the accelerator does not support graphs that contain plot overlays.
  If you are creating graphs, after reviewing the recommendations here, see Recommendations for Graphs
  on page 74.
- The Graph Template Language (GTL) is the underlying language for the default templates that are
  provided by SAS for procedures that use ODS Graphics. Use the GTL either to modify these templates or
  to create your own highly customized charts and plots. Use GTL to create a template, and then use the
  SGRENDER procedure to render the graph.
- Use the accessibility system options to help create accessible content on a system-wide level.
Use this ODS destination and options:

- Use the DAISY style option in the ODS HTML5 destination statement.
  This option helps satisfy the WCAG 2.0 success criteria 1.4.3 Contrast (Minimum) and 1.4.1 Use of Color.
  Note: If you are using SAS Studio, configure the style as a default preference.
- Use TITLE option, which specifies the title for the HTML page. The title should be clear and descriptive.
  This option helps satisfy the WCAG 2.0 success criterion 2.4.2 Page Title.
- If you need to produce PDF output, see Overview of Accessible PDF Documents on page 5.

Use ODS LAYOUT statements to arrange content, if applicable. See an example of creating a report.
Specify a TITLE statement to give your output a title. The title should be clear and descriptive.
This option helps satisfy WCAG 2.0 success criterion 2.4.6 Headings and Labels.
Use the DESCRIPTION option, which is available in various statements that create tables, graphs, and ODS layouts and regions. This option describes the object that you are creating.
Starting with SAS 9.4M5, you can customize the appearance of focus indicators. A new FOCUSLINK style element is available to be used in a style template. Three new style attributes enable you to customize the focus outline pattern, width, and color. See Enhancing the Appearance of the Link-Focus Indicator on page 102 and Example: Bar Chart with Data Tips and Drill-Down Links Created with GTL on page 84. This option helps satisfy the WCAG 2.0 success criterion 2.4.7 Focus Visible.

See Also
- Recommendations for Graphs on page 74
- General Recommendations for Tables on page 11

What the Software Does Automatically for You
- Many tables generated from SAS define table headers automatically.
  This helps satisfy WCAG 2.0 success criterion 1.3.1 Info and Relationships.
- The SAS accessibility software inserts a LANG= property in the main HTML tag at the beginning of the HTML file. Here is an example:
  <html lang='en'>
  This property enables screen readers to use the correct language voice.
  The property helps satisfy the WCAG 2.0 success criterion 3.1.1 Language of Page.

What to Avoid
In general, you should avoid using software that is not described in Recommendations. Here are a few additional things to avoid.
- Do not use the NOGTITLE option in the ODS HTML5 statement. The option places the title outside of a graph, making it inaccessible to the SAS Graphics Accelerator.
- Do not use NOHEADER in the REPORT procedure when producing tables. The NOHEADER option produces inaccessible tables (they have no headers).
- Do not use the following to produce graphs. They are not supported by the SAS Graphics Accelerator:
  - multimedia files.
the SGMAP procedure, which is new in SAS 9.4M5. The procedure does not support the ACCESSIBLE_GRAPH HTML5 option.

- graphs produced with SAS/GRAPH.

### Overview of Accessible PDF Documents

HTML5 is the recommended destination for creating accessible content. However, if you need to create PDF output, use the ACCESSIBLE option in the ODS PDF statement. The option adds document structure tags to the PDF document that are required by the Web Content Accessibility Guidelines (WCAG) version 2.0 and the PDF Universal Accessibility (PDF/UA) format. Users with disabilities including visual impairments or blindness need these tags to read the contents of PDF files.

**TIP** Starting with SAS 9.4M6, you can specify the ACCESSIBLEPDF system option, which enables the ACCESSIBLE option in the ODS PDF destination by default.

The following features apply to SAS 9.4M6 and later releases:

- 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.

- You can specify the ACCESSIBLE_IDENTIFIER option in the ODS PDF statement to add an identifier to the metadata of the PDF file. This identifier confirms that the PDF produced by SAS meets the PDF Matterhorn Protocol.

- Support has been added for visual captions. The CAPTION= option, when used in the supported procedures, creates tables with captions that are visible and accessible in the PDF.

- Alternative text for images that are created with the POSTIMAGE= and PREIMAGE= style options or with the Report Writing Interface IMAGE method appear in PDF output.

- Images can be marked as artifacts in PDF output. For example, the following code creates an artifact:

  ```
  preimage="file.jpg?desc=" 
  ```

For more information about the ODS PDF statement, see “ODS PDF Statement” in *SAS Output Delivery System: User’s Guide*.

### Configure SAS Studio for Accessible Output

The SAS Studio interface enables you to specify default values for features that enhance the accessibility of your output.

1. In SAS Studio, click and select Preferences.

2. Click Results. Do the following:

   - Select a new default style from the HTML output style list box. *Daisy is recommended* for most output. This option eliminates the need to specify STYLE= in the ODS HTML5 destination statement in your program.

   - Select Enable accessible graph option. This option eliminates the need to specify ACCESSIBLE_GRAPH in the ODS HTML5 destination statement in your program.

   Accessible graph output can be read and interpreted by the *SAS Graphics Accelerator*. 
Select **Generate HTML graphs as SVG**. This option eliminates the need to specify OUTPUTFMT=SVG in the ODS GRAPHICS statement.

3. Click **Save**.
Accessibility System Options

Overview of the Accessibility System Options

About the Accessibility System Options

The accessibility system options facilitate the creation of accessible content on a system-wide level. The options apply to the following use cases:

- output created with the HTML5 and PDF ODS destinations
- output from the PRINT, REPORT, and TABULATE procedures as well as the Report Writing Interface
- graphical output

Note: The system options apply to SAS 9.4M6 and to later releases.

In addition, there is a system option that checks your SAS executed programs for common violations of accessibility standards and writes messages to the SAS log if violations are found.

When the system options are specified, accessible output becomes the default behavior for reports, tables, and graphs that follow the guidelines listed in this document. The system options remain in effect until they are disabled or the SAS session ends.

You can easily enable and disable any of the system options. For example, you might want to disable a system option while testing a specific procedure or because of some other business need. After executing the procedure, you can re-enable the system option. Like many other SAS system options, the accessibility options can be restricted by the administrator at your site.

Summary of the System Options

Here is a brief summary of the options:

- The ACCESSIBLECHECK option checks your SAS programs for common violations of accessibility standards and writes messages to the SAS log if violations are found.
- The ACCESSIBLEGRAPH option enables the ACCESSIBLE_GRAPH option in the ODS HTML5 destination by default.
- The ACCESSIBLEPDF option enables the ACCESSIBLE option in the ODS PDF destination by default.
The ACCESSIBLETABLE option makes table captions visible and changes the layout of some tables to make them accessible.

This option applies to tables generated by the PRINT, REPORT, and TABULATE procedures as well as the Report Writing Interface.

The options can be specified individually or they can be combined into a single statement, as shown in this example:

```
options accessiblecheck accessibletable accessiblegraph accessiblepdf;
```

For more information, see the syntax.

## Accessibility System Option Syntax

This section summarizes the syntax for the system options. For complete syntax, see [SAS System Options: Reference](#).

These system options apply to SAS 9.4M6 and to later releases.

**ACCESSIBLECHECK | NOACCESSIBLECHECK**

specifies whether to check your SAS programs for common violations of accessibility standards, and to write messages to the SAS log if violations are found.

**ACCESSIBLECHECK**

performs accessibility checking as SAS programs are executed.

Here are some examples of situations that would generate a log entry:

- images that are inserted without alternative text descriptions
- tables that include blank lines for visual effects

**NOACCESSIBLECHECK**

does not perform accessibility checking as SAS programs are executed.

Default: **NOACCESSIBLECHECK**

See “ACCESSIBLECHECK System Option” in [SAS System Options: Reference](#)

Example: `options accessiblecheck;`

**ACCESSIBLEGRAPH | NOACCESSIBLEGRAPH**

specifies whether to enable the ACCESSIBLE_GRAPH option in the ODS HTML5 destination by default. This setting applies to all graphs that are generated with the HTML5 destination while the system option is in effect.

**ACCESSIBLEGRAPH**

enables the ACCESSIBLE_GRAPH option in the ODS HTML5 destination by default. The option enables the graph to be read, interpreted, and sonified by the [SAS Graphics Accelerator](#).

When this system option is specified, you do not need to specify the ACCESSIBLE_GRAPH option in individual ODS HTML5 statements.

**NOACCESSIBLEGRAPH**

disables the ACCESSIBLE_GRAPH option in the ODS HTML5 destination. You can override this setting for graphs by specifying the ACCESSIBLE_GRAPH option in individual ODS HTML5 statements.

Default: **NOACCESSIBLEGRAPH**

See “ACCESSIBLEGRAPH System Option” in [SAS System Options: Reference](#)
ACCESSIBLEPDF | NOACCESSIBLEPDF
specifies whether to enable the ACCESSIBLE option in the ODS PDF destination by default. The setting applies to all documents that are generated with the PDF destination while the system option is in effect.

ACCESSIBLEPDF
enables the ACCESSIBLE option in the ODS PDF destination by default. When this system option is specified, you do not need to specify the ACCESSIBLE option in individual ODS PDF statements.

NOACCESSIBLEPDF
disables the ACCESSIBLE option in the ODS PDF destination. You can override this setting by specifying the ACCESSIBLE option in individual ODS PDF statements.

Default NOACCESSIBLEPDF

See “ACCESSIBLEPDF System Option” in SAS System Options: Reference

Example options accessiblepdf;

ACCESSIBLETABLE | NOACCESSIBLETABLE
specifies whether to make table captions visible and to change the layout of some tables to make them accessible.

ACCESSIBLETABLE
alters the layout of some generated tables in order to make the tables accessible. For example, when used with the TABULATE procedure, ACCESSIBLETABLE creates additional column headers instead of using row spans for column headers. This option also makes table captions visible on the page. Previously, table captions were generated to be visually hidden and available only to screen reader users.

This option applies to tables that are generated by the PRINT, REPORT, and TABULATE procedures as well as the Report Writing Interface.

NOACCESSIBLETABLE
does not change procedure tabular output.

Default NOACCESSIBLETABLE

See “ACCESSIBLETABLE System Option” in SAS System Options: Reference

Example options accessibletable;

See Also
Overview of the Accessibility System Options on page 7
General Information about Accessible Tables

Overview of Table Accessibility

Tables can be a challenge to understand for users who cannot see the tables. Table layouts are designed to visually represent how each row and column relate to one another. Tables also visually show whether each row or column has a header that describes the row or column. Table accessibility involves relaying those visual relationships to people who cannot see the table.

There are two main tasks for providing table accessibility:

- ensure that data cells can be associated with row and column headings
- provide descriptions of the tables

This document explains how to achieve those accessibility goals for supported procedures. The document also provides additional recommendations for table accessibility when applicable. See the following sections:

- Creating Accessible Tables with the PRINT Procedure on page 13
- Creating Accessible Tables with the REPORT Procedure on page 25
- Creating Accessible Tables with the TABULATE Procedure on page 39
- Creating Accessible Tables and Reports with the Report Writing Interface on page 53

General Recommendations for Tables

The following recommendations apply to the PRINT, REPORT, and TABULATE procedures and the Report Writing Interface, except where noted otherwise.

- Use the ACCESSIBLETABLE and ACCESSIBLECHECK system options. The system options facilitate the creation of accessible content on a system-wide level.
- Use the CONTENTS= option to provide a description of the data presented in the table. The description notifies screen reader users about the information that is presented in each table.
- Starting in SAS 9.4M6, the CONTENTS= option accepts #BY directives.
Note: The #BY directives are not supported by the Report Writing Interface. For the REPORT procedure, they are supported when ACCESSIBLETABLE has been specified.

- Use the CAPTION= option to define a visible table caption. The caption is displayed when the ACCESSIBLETABLE system option has been specified.

This option overrides the CONTENTS= option. If no CAPTION= attribute is provided, the CONTENTS= option is used.

Here is a simple example:

```sas
options accessibleetable accessiblecheck;
proc tabulate data=sashelp.class;
  class sex age;
  table sex, age / caption="Age Totals by Gender";
run;
```

CAPTION= accepts #BY directives.

Note: This feature applies to SAS 9.4M6 and later releases.

Note: The PRINT procedure does not support the CAPTION= option.

See Also

- Recommendations for the PRINT Procedure on page 14
- Recommendations for the REPORT Procedure on page 26
- Recommendations for the TABULATE Procedure on page 40
- Creating Accessible Tables and Reports with the Report Writing Interface on page 53
Creating Accessible Tables with the PRINT Procedure

About the PRINT Procedure

The PRINT procedure prints the observations in a SAS data set using all or some of the variables. You can create a variety of reports ranging from a simple LISTING to a highly customized report that groups the data and calculates totals and subtotals for numeric variables. For more information, see “PRINT Procedure” in Base SAS Procedures Guide.

See also Overview of Table Accessibility on page 11.
Recommendations for the PRINT Procedure

Recommendations

The Overview and General Recommendations contain best practices. In addition to the general recommendations, follow these specific recommendations if you are using the PRINT procedure.

Unless otherwise specified, the following recommendations help satisfy WCAG 2.0 success criterion 1.3.1 Info and Relationships.

- Provide descriptive labels for summation rows when you include them in BY tables. The default label for these row headings is not descriptive enough for screen reader users to understand how the information presented in the row relates to the rest of the table. Descriptions are provided through the GRANDTOTAL_LABEL and SUMLABEL options. You can customize the summation row for each individual table in a set of BY tables by including the #BYVAL() option. You can include the #BYVAL() suboption in the GRANDTOTAL_LABEL and SUMLABEL options.
  
  See Example: Table with Labels for Multiple Summation Rows on page 21.

- Use the CONTENTS option to provide a description of the data presented in the table. The description notifies screen reader users about the information that is presented in each table.
  
  See Example: Table with a Descriptive Summation Row on page 19.

- Define row headings by including one of the variables in an ID statement or by omitting the NOOBS option. Each of these methods creates row heading labels. If the ID statement is used, the value of the ID variable for each row becomes the row heading. When you omit the NOOBS option, the observation numbers appear as row headers.
  
  See Example: Table with the Observation Numbers on page 16 and Example: Table with a Descriptive Row Heading on page 18.

- If you are using a BY statement, include the same BY variable in the ID statement. Specifying the variable in both locations ensures that all tables in a set of BY tables are distinguishable from one another because the BY variable is included as a row heading in each table.
  
  See Example: Table with Labels for Multiple Summation Rows on page 21.

What to Avoid

- Do not use NOSUMLABEL with SUM commands in BY tables. This combination prevents the creation of a label for the row. As a result, screen reader users do not know what information is being presented in the row.

- Do not use BLANKLINE to insert blank lines. Blank lines create rows with no data in them, which makes it more difficult for screen reader users to understand the contents of the table.

- Do not use the N argument to display the number of results. The number of results is presented as an additional table row, and that can confuse screen reader users about the meaning of the row. This information should be presented outside of the table structure.
  
  See Example: Use the N Argument to Display the Number of Results on page 23.

- The CAPTION option is new for SAS 9.4M6. However, do not use the CAPTION option with PROC PRINT to generate visible captions. PROC PRINT does not support the CAPTION option.
Example: Simple Table with a Title

Program

This example uses the PRINT procedure to create an accessible table. The example shows you how to set up the ODS environment. The example also shows you how to provide titles and a description of the table.

Note: In SAS Studio, the example output appears in your user folder. In SAS Display Manager, the example appears in the Results window.

```sas
ods html5 (id=web) style=daisy file="procprint01.html"
   (title="Honda Cars") options(outline='no');
proc odstext;
   h1 'Honda Cars';
run;
proc print data=sashelp.cars label noobs
   contents="Honda Cars";
   label MPG_City="MPG City";
   var Model MSRP MPG_City;
   where make="Honda";
run;
```

1 Here are details about the ODS environment:
   ■ The ODS HTML5 destination statement sets the ODS environment.
   ■ The (ID=WEB) option is required for use only with SAS Studio.
   ■ The STYLE= option specifies the style Daisy, which is recommended for accessible output.
   
   Note: If you are using SAS Studio, you can configure the STYLE= option as a default preference.
   ■ 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. Specifying(TITLE=) is required by WCAG 2.0 success criterion 2.4.2 Page Title.
   ■ 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 show up as headings in the HTML outline that is read by screen readers.

2 The H1 statement provides a title for the output.

3 In the PROC PRINT statement, the NOOBS option suppresses the observation numbers. Observation numbers are not needed as row headers in this example. The CONTENTS= option provides a description of the data that is presented in the table. This enables screen reader users to know what information is being presented in the table.
Example: Table with the Observation Numbers

Program

This example omits the NOOBS option from the PROC PRINT statement. The result is a table with a column that contains the observation numbers. The example also shows you how to set up the ODS environment.

Note: In SAS Studio, the example output appears in your user folder. In SAS Display Manager, the example appears in the Results window.

```
ods html5 (id=web) style=daisy file="procprint01.html" (title="Honda Cars") options(outline='no'); /* 1 */
proc odshtml;
  h1 'Honda Cars';
run;

proc print data=sashelp.cars label contents="Honda Cars"; /* 2 */
```
Here are details about the ODS environment:

- The ODS HTML5 destination statement sets the ODS environment.
- The (ID=WEB) option is required for use only with SAS Studio.
- The STYLE= option specifies the style Daisy, which is recommended for accessible output.

Note: If you are using SAS Studio, you can configure the STYLE= option as a default preference.

- 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. Specifying (TITLE=) is required by WCAG 2.0 success criterion 2.4.2 Page Title.
- 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 show up as headings in the HTML outline that is read by screen readers.

The PROC PRINT statement does not include a NOOBS option. The NOOBS option suppresses the column in the output that identifies each observation by number. When you omit the NOOBS option, the observation numbers appear as row headers.
Output

Output 4.2  Table with the Observation Numbers

Honda Cars

```
| 2 | 1 |
```

Example: Table with a Descriptive Row Heading

This example uses an ID statement to create a column of descriptive row headings.

Program
```
ods html5 (id=web) style=daisy file="procprint03.html" (title="Honda Cars") options(outline='no');

proc odstext;
    h1 'Honda Cars';
run;

proc print data=sashelp.cars label contents="Honda Cars";
    label MPG_City="MPG City";
    var MSRP MPG_City;
    where make="Honda";
    id Model;  /* 1 */
```
run;
ods html5 (id=web) close;

1 The ID statement specifies one or more variables to print instead of the observation number at the beginning of each row.

Output

Output 4.3  Table with a Descriptive Row Heading

Honda Cars

<table>
<thead>
<tr>
<th>Model</th>
<th>MSRP</th>
<th>MPG City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Civic Hybrid 4dr manual (gas/electric)</td>
<td>$20,140</td>
<td>46</td>
</tr>
<tr>
<td>Insight 2dr (gas/electric)</td>
<td>$19,110</td>
<td>60</td>
</tr>
<tr>
<td>Pilot LX</td>
<td>$27,550</td>
<td>17</td>
</tr>
<tr>
<td>CR-V LX</td>
<td>$19,880</td>
<td>21</td>
</tr>
<tr>
<td>Element LX</td>
<td>$18,690</td>
<td>21</td>
</tr>
<tr>
<td>Civic DX 2dr</td>
<td>$13,270</td>
<td>32</td>
</tr>
<tr>
<td>Civic HX 2dr</td>
<td>$14,170</td>
<td>36</td>
</tr>
<tr>
<td>Civic LX 4dr</td>
<td>$15,850</td>
<td>32</td>
</tr>
<tr>
<td>Accord LX 2dr</td>
<td>$19,850</td>
<td>26</td>
</tr>
<tr>
<td>Accord EX 2dr</td>
<td>$22,250</td>
<td>26</td>
</tr>
<tr>
<td>Civic EX 4dr</td>
<td>$17,750</td>
<td>32</td>
</tr>
<tr>
<td>Civic Si 2dr hatch</td>
<td>$19,490</td>
<td>26</td>
</tr>
<tr>
<td>Accord LX V6 4dr</td>
<td>$23,760</td>
<td>21</td>
</tr>
<tr>
<td>Accord EX V6 2dr</td>
<td>$26,950</td>
<td>21</td>
</tr>
<tr>
<td>Odyssey LX</td>
<td>$24,950</td>
<td>18</td>
</tr>
<tr>
<td>Odyssey EX</td>
<td>$27,450</td>
<td>18</td>
</tr>
<tr>
<td>S2000 convertible 2dr</td>
<td>$33,250</td>
<td>20</td>
</tr>
</tbody>
</table>

Example: Table with a Descriptive Summation Row

This example uses a GRANDTOTAL_LABEL option in the PROC PRINT statement to specify a label for the grand total line. The example also uses a SUM statement to sum the results. Compare the code and output in this example to Example: Table with Labels for Multiple Summation Rows in order to see the code and output for multiple summation rows.

Note: Summation rows appear differently than data rows. Add labels for summation rows to help users of screen readers understand the contents of the summation row.
Program

ods html5 (id=web) style=daisy file="procprint04.html"
   (title="Total Home Runs by the Kansas City Royals in 1986")
options(outline='no');

proc odstext;
   h1 'Total Home Runs by the Kansas City Royals in 1986';
run;

proc print data=sashelp.baseball label grandtotal_label="Total Home Runs" /* 1 */
   contents="Total Home Runs by the Kansas City Royals in 1986";
   where Team="Kansas City";
   var Name Position nHome;
   label Name="Player" Team="Team" Position="Position" nHome="Home Runs";
   sum nHome;               /* 2 */
run;

ods html5 (id=web) close;

1 The GRANDTOTAL_LABEL argument specifies a label for the grand total line.
2 The SUM statement creates a row at the bottom of the table that contains the sum for the specified column.
Output

Output 4.4  Table with a Descriptive Summation Row

Total Home Runs by the Kansas City Royals in 1986

<table>
<thead>
<tr>
<th>Obs</th>
<th>Player</th>
<th>Position</th>
<th>Home Runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Salazar, Argenis</td>
<td>SS</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>Biancalana, Buddy</td>
<td>SS</td>
<td>2</td>
</tr>
<tr>
<td>84</td>
<td>Molley, Darryl</td>
<td>RF</td>
<td>7</td>
</tr>
<tr>
<td>108</td>
<td>White, Frank</td>
<td>2B</td>
<td>22</td>
</tr>
<tr>
<td>111</td>
<td>Brett, George</td>
<td>3B</td>
<td>16</td>
</tr>
<tr>
<td>133</td>
<td>McRae, Hal</td>
<td>DH</td>
<td>7</td>
</tr>
<tr>
<td>159</td>
<td>Orta, Jorge</td>
<td>DH</td>
<td>9</td>
</tr>
<tr>
<td>161</td>
<td>Quirk, Jamie</td>
<td>CS</td>
<td>8</td>
</tr>
<tr>
<td>171</td>
<td>Sundberg, Jim</td>
<td>C</td>
<td>12</td>
</tr>
<tr>
<td>200</td>
<td>Smith, Lonnie</td>
<td>LF</td>
<td>8</td>
</tr>
<tr>
<td>213</td>
<td>Kingery, Mike</td>
<td>OF</td>
<td>3</td>
</tr>
<tr>
<td>255</td>
<td>Law, Rudy</td>
<td>OF</td>
<td>1</td>
</tr>
<tr>
<td>273</td>
<td>Balboni, Stove</td>
<td>1B</td>
<td>29</td>
</tr>
<tr>
<td>322</td>
<td>Wilson, Willie</td>
<td>CF</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Total Home Runs</td>
<td></td>
<td>133</td>
</tr>
</tbody>
</table>

Example: Table with Labels for Multiple Summation Rows

This example creates an individual table with a summation row for each team and a final table with an additional summation row for the league. Compare the code and output in this example to Example: Table with a Descriptive Summation Row in order to see the code and output for a table with a single summation row.

Note: Summation rows should be presented differently than regular data rows to screen reader users. Add labels for summation rows to help users of screen readers understand the contents of the summation row.

Program

ods html5 (id=web) style=daisy file="procprint05.html" (title="Home Runs")
options(outline='no');

proc sort data=sashelp.baseball out=baseballSort;
    by Team League DESCENDING nHome;

run;

proc print data=baseballSort noobs label
  sumlabel='Total Home Runs for BYVAL(Team)' /* 1 */
  grandtotal_label="Total Home Runs for the League" /* 2 */
  contents="Home Runs by Team";
  by Team;
  id Name;
  var Position nHome;
  label Name="Player" Team="Team" Position="Position"
     nHome="Home Runs";
  sum nHome;
run;
ods html5 (id=web) close;

1 The SUMLABEL option specifies a label on the summary line for the BY group. The #BYVAL(Team)
expression customizes the summation row.
2 The GRANDTOTAL_LABEL option specifies a label for the grand total line.

Output

Although this example creates a separate table for each team, this output contains only the final table.

Output 4.5 Table with Labels for Multiple Summation Rows

<table>
<thead>
<tr>
<th>Player</th>
<th>Position</th>
<th>Home Runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barfield, Josso</td>
<td>RF</td>
<td>40</td>
</tr>
<tr>
<td>Bell, George</td>
<td>LF</td>
<td>31</td>
</tr>
<tr>
<td>Moseby, Lloyd</td>
<td>CF</td>
<td>21</td>
</tr>
<tr>
<td>Whitt, Ernie</td>
<td>C</td>
<td>16</td>
</tr>
<tr>
<td>Johnson, Cliff</td>
<td>DH</td>
<td>15</td>
</tr>
<tr>
<td>Mulliniks, Rance</td>
<td>3B</td>
<td>11</td>
</tr>
<tr>
<td>Fernandez, Tony</td>
<td>SS</td>
<td>10</td>
</tr>
<tr>
<td>Upshaw, Willie</td>
<td>1B</td>
<td>9</td>
</tr>
<tr>
<td>Garcia, Dameso</td>
<td>2B</td>
<td>6</td>
</tr>
<tr>
<td>Leach, Rick</td>
<td>DO</td>
<td>5</td>
</tr>
<tr>
<td>Iorg, Garth</td>
<td>32</td>
<td>3</td>
</tr>
<tr>
<td>Total Home Runs for Toronto</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>Total Home Runs for the League</td>
<td>3575</td>
<td></td>
</tr>
</tbody>
</table>
Example: Use the N Argument to Display the Number of Results

This example uses an N argument to print the number of observations outside of the table. The number appears in the form N=n.

When the N argument is displayed in the table, it is displayed in a way that is not accessible to screen reader users. This is because it is not an actual row of data but is presented alongside the data. To overcome this problem, print the N argument outside of the table as this example shows.

Program

```sas
ods html5 (id=web) style=daisy file="procprint06.html" (title="Honda Cars") options(outline='no');

/* macro source: http://support.sas.com/kb/24/671.html */
%macro obsnvars(ds);
    %global dset nvars nobs;
    %let dset=&dset
    %let dsid = %sysfunc(open(&dset));
    
    %if &dsid %then
        %do;
            %let nobs =%sysfunc(attrn(&dsid,nobs));
            %let nvars=%sysfunc(attrn(&dsid,nvars));
            %let rc = %sysfunc(close(&dsid));
        %end;
    %else %put open for data set &dset failed - %sysfunc(sysmsg());
    %mend obsnvars;

data cars_subset;
    set sashelp.cars;
    where make="Honda";
    run;

    %obsnvars(cars_subset)

proc odstext;
    h1 'Honda Cars';
run;

proc print data=cars_subset label noobs contents="Honda Cars";
    label MPG_City="MPG City";
    var Model MSRP MPG_City;
    where make="Honda";
run;

proc odstext;
    p 'N='||"&nobs"; /* 1 */
run;
```
The N option prints the number of observations below the table.

Output

Output 4.6  Table with the Number of Observations Printed outside the Table

Honda Cars

<table>
<thead>
<tr>
<th>Model</th>
<th>MSRP</th>
<th>MFG City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Civic Hybrid 4dr manual (gas/electric)</td>
<td>$20,140</td>
<td>45</td>
</tr>
<tr>
<td>Insight 2dr (gas/electric)</td>
<td>$19,110</td>
<td>60</td>
</tr>
<tr>
<td>Pilot LX</td>
<td>$27,560</td>
<td>17</td>
</tr>
<tr>
<td>CR-V LX</td>
<td>$19,860</td>
<td>21</td>
</tr>
<tr>
<td>Element LX</td>
<td>$18,690</td>
<td>21</td>
</tr>
<tr>
<td>Civic DX 2dr</td>
<td>$13,270</td>
<td>32</td>
</tr>
<tr>
<td>Civic HX 2dr</td>
<td>$14,170</td>
<td>36</td>
</tr>
<tr>
<td>Civic LX 4dr</td>
<td>$15,850</td>
<td>32</td>
</tr>
<tr>
<td>Accord LX 2dr</td>
<td>$19,860</td>
<td>26</td>
</tr>
<tr>
<td>Accord EX 2dr</td>
<td>$22,260</td>
<td>26</td>
</tr>
<tr>
<td>Civic EX 4dr</td>
<td>$17,750</td>
<td>32</td>
</tr>
<tr>
<td>Civic Si 2dr hatch</td>
<td>$19,450</td>
<td>26</td>
</tr>
<tr>
<td>Accord LX V6 4dr</td>
<td>$23,760</td>
<td>21</td>
</tr>
<tr>
<td>Accord EX V6 2dr</td>
<td>$26,960</td>
<td>21</td>
</tr>
<tr>
<td>Odyssey LX</td>
<td>$24,950</td>
<td>18</td>
</tr>
<tr>
<td>Odyssey EX</td>
<td>$27,450</td>
<td>18</td>
</tr>
<tr>
<td>S2000 convertible 2dr</td>
<td>$33,260</td>
<td>20</td>
</tr>
</tbody>
</table>

N=17
About the REPORT Procedure

The REPORT procedure combines features of the PRINT, MEANS, and TABULATE procedures with features of the DATA step in a single report-writing tool that can produce a variety of reports. For more information, see “Overview: REPORT Procedure” in Base SAS Procedures Guide.

See also Overview of Table Accessibility on page 11.
Recommendations for the REPORT Procedure

Recommendations

The General Recommendations for Creating Accessible Output contain best practices. In addition to the general recommendations, follow these specific recommendations if you are using the REPORT procedure.

Unless otherwise specified, the following recommendations help satisfy WCAG 2.0 success criterion 1.3.1 Info and Relationships.

- When ORDER or GROUP is used in the DEFINE statement, SPANROWS must be used in the PROC REPORT statement to generate an accessible table. SPANROWS specifies that when the value of a GROUP or ORDER column is the same in multiple rows, the value is displayed in a single cell that occupies that column in all the rows for which the value is the same. SPANROWS ensures that the headers are properly assigned to the data cells.

  See Example: Detail Report with Ordering and SPANROWS on page 30 and Example: Summarization Row with Custom Labels on page 32.

- For a summation row for a GROUP, use a COMPUTE block to provide a more descriptive label for the summary data.

  See Example: Summarization Row with Custom Labels on page 32.

- Use the CONTENTS option in the PROC REPORT statement to provide a descriptive label for the first node of the table. The CONTENTS specifies the text for the entries in the table of contents that is created.

- Starting with SAS 9.4M6, specify the ACCESSIBLETABLE system option changes the layout of some tables to make them accessible and adds visual captions to tables. If you specify the ACCESSIBLETABLE option, use the ACCESSIBLECHECK system option and check the logs for warnings. You might need to add other options to make the table fully accessible.

  See Example: Detail Report with ACCESSIBLETABLE System Option on page 34.

- Starting with SAS 9.4M6, specify the ACCESSIBLECHECK system option to detect accessibility problems. This option causes a message to be written to the SAS log if accessibility problems are found.

  See Example: Accessible Table Using By Directives on page 36.

- Starting with SAS 9.4M6, you can specify the CAPTION= option in the PROC REPORT statement and specify the ACCESSIBLETABLE system option to add visible captions to the tables. If you do not specify the CAPTION= option, the caption defaults to the text specified by the CONTENTS= option.

  See Example: Accessible Table Using By Directives on page 36.

- Starting with SAS 9.4M6, when BY directives are specified in the CAPTION= and CONTENTS= options, labels for the BY group tables are displayed in the table of contents in PDF output and in the contents file in HTML output. The labels are based on the values of the BY variable.

  See Example: Accessible Table Using By Directives on page 36.

What to Avoid

- Do not use the LINE statement. Using the LINE statement causes an inaccessible table to be generated. The LINE statement creates a table row that is not interpreted as either a table header or a data cell with a defined table header. The content presented in the LINE statement cannot be semantically related to the rest of the data presented in the table. The content might derive meaning because it is visually positioned in
relation to other data in the table, so that its relationship to the data can be inferred. However, this visual presentation is not available to screen reader users.

- If the SUPPRESS option and SUMMARIZE options are both specified in the BREAK statement, the summary row headers are not displayed and the table is not accessible.

---

**Example: Detail Report with a Table Description**

**Program**

This example uses the REPORT procedure to create an accessible table. This example shows you how to set up the ODS environment, provide titles for the output, and specifies a descriptive label for the table. This example shows how to create an accessible table before SAS 9.4M6.

```sas
ods html5 (id=web) style=daisy file="procreport01b.html"
   (title="Honda Cars") options(outline='no'); /* 1 */

title1 'Honda Cars Detail Report'; /* 2 */

proc report data=sashelp.cars contents="Honda Cars"; /* 3 */
   where Make="Honda";
run;

title1;
ods html5 (id=web) close;
```

1. Here are details about the ODS environment:
   - The ODS HTML5 destination statement sets the ODS environment.
   - The (ID=WEB) option is required for use only with SAS Studio.
   - The STYLE= option specifies the style Daisy, which is recommended for accessible output.
   
   **Note:** If you are using SAS Studio, you can configure the STYLE= option as a default preference.

2. The TITLE1 statement provides a title for the output.

3. The CONTENTS option specifies the text for the entry in the table of contents that is created. The option also creates a description of the table.
### Example: Tables with Spanned Column Headers

This example contains two methods for using spanned columns in table headers. The first example program creates a header row that spans all four of the table’s columns. The second example program creates a header row that spans only two columns of the table’s four columns. Both tables are accessible. For both examples, row headers are created based on the labels in the data set.

#### Program

This example creates a header row that spans all four of the table’s columns.

```plaintext
ods html5 (id=web) style=daisy file="procreport02.html" (title="Honda Cars") options(outline='no');

title1 "MSRP and City MPG by Model";

proc report data=sashelp.cars contents="Honda Cars";
  where Make="Honda";
  column ("Car Information" Make Model MSRP MPG_City); /* X */
run;

title1;

ods html5 (id=web) close;
```

The `COLUMN` statement creates a header row that spans all of the columns in the table and contains the text *Car Information*.
Output 5.2  Table with Header That Spans Four Columns

**MSRP and City MPG by Model**

<table>
<thead>
<tr>
<th>Make</th>
<th>Model</th>
<th>Car Data</th>
<th>MSRP</th>
<th>MPG (City)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Honda</td>
<td>Civic Hybrid 4dr manual (gas/electric)</td>
<td>$20,140</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Insight 2dr (gas/electric)</td>
<td>$19,110</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Pilot LX</td>
<td>$27,560</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>CR-V LX</td>
<td>$19,860</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Element LX</td>
<td>$18,690</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Civic DX 2dr</td>
<td>$13,270</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Civic EX 2dr</td>
<td>$14,170</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Civic LX 4dr</td>
<td>$15,650</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Accord LX 2dr</td>
<td>$19,860</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Accord EX 2dr</td>
<td>$22,260</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Civic EX 4dr</td>
<td>$17,750</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Civic Si 2dr hatch</td>
<td>$19,490</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Accord LX V6 4dr</td>
<td>$23,760</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Accord EX V6 2dr</td>
<td>$26,960</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Odyssey LX</td>
<td>$24,950</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>Odyssey EX</td>
<td>$27,450</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Honda</td>
<td>S2000 convertible 2dr</td>
<td>$33,260</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

Program

This example creates a header row that spans only two columns of the table’s four columns.

```sas
ods html5 (id=web) style=daisy file="procreport03.html" (title="Honda Cars") options(outline='no');

title1 "Horsepower and City MPG by Model";

proc report data=sashelp.cars contents="Honda Cars";
  where Make="Honda";
  column Make Model (*"Car Data" MPG_City Horsepower); /* 1 */
run;

title1;

ods html5 (id=web) close;
```

1 The COLUMN statement creates a header row that contains the text *Car Data*. This row spans only the columns labeled MPG (City) and Horsepower in the row below *Car Data*.  

---

Example: Tables with Spanned Column Headers 29
Example: Detail Report with Ordering and SPANROWS

Program

This example orders the rows alphabetically by the values of Make, and secondly by the values of Model. Defining a variable as GROUP or ORDER causes the values of that variable to become a row header. Because both Make and Model are defined as ORDER, Model becomes a second row header.

If there are additional columns that should be used as row headings to uniquely define the contents of the row, use the DEFINE statement to make additional columns into headings.

ods html5 (id=web) style=daisy file="procreport04.html" (title="All Cars") options(outline='no');

title1 "Detail Report by Make and Model";

proc report data=sashelp.cars contents="All Cars" /* 1 */
    spanrows; /* 2 */ /*
        define Make / order;
*/
define Model / order;                        /* 3 */
column Make Model MSRP MPG_City Horsepower;
run;
title1;
ods html5 (id=web) close;

1 The CONTENTS= option provides a descriptive label for the table which is needed for accessibility.
2 The SPANROWS option is required to make the table accessible because one or more variables are defined as GROUP or ORDER.
3 The second DEFINE statement defines an additional row header to help uniquely identify the data for car models. Without the statement, each row in this example would be uniquely described only by the heading Make and not by the heading Model. By defining both as headings, screen reader users are better able to determine the information in each row.

Output
Here are the first few rows of the output table.

Output 5.4  Detail Report with Ordering and SPANROWS

<table>
<thead>
<tr>
<th>Make</th>
<th>Model</th>
<th>MSRP</th>
<th>MPG (City)</th>
<th>Horsepower</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acura</td>
<td>3.5 RL 4dr</td>
<td>$43,755</td>
<td>18</td>
<td>225</td>
</tr>
<tr>
<td></td>
<td>3.5 RL w/Navigation 4dr</td>
<td>$46,100</td>
<td>18</td>
<td>225</td>
</tr>
<tr>
<td></td>
<td>MDX</td>
<td>$36,945</td>
<td>17</td>
<td>265</td>
</tr>
<tr>
<td></td>
<td>NSX coupe 2dr manual S</td>
<td>$89,765</td>
<td>17</td>
<td>290</td>
</tr>
<tr>
<td></td>
<td>RSX Type S 2dr</td>
<td>$23,820</td>
<td>24</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>TL 4dr</td>
<td>$33,195</td>
<td>20</td>
<td>270</td>
</tr>
<tr>
<td></td>
<td>TSX 4dr</td>
<td>$26,990</td>
<td>22</td>
<td>290</td>
</tr>
<tr>
<td>Audi</td>
<td>A4 1.8T 4dr</td>
<td>$25,940</td>
<td>22</td>
<td>170</td>
</tr>
<tr>
<td></td>
<td>A4 3.0 4dr</td>
<td>$31,840</td>
<td>20</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>A4 3.0 Quattro 4dr auto</td>
<td>$34,480</td>
<td>18</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>A4 3.0 Quattro 4dr manual</td>
<td>$33,430</td>
<td>17</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>A4 3.0 Quattro convertible 2dr</td>
<td>$44,240</td>
<td>18</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>A4 3.0 convertible 2dr</td>
<td>$42,490</td>
<td>20</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>A41.8T convertible 2dr</td>
<td>$35,940</td>
<td>22</td>
<td></td>
</tr>
</tbody>
</table>
Example: Summarization Row with Custom Labels

Program

This example contains two PROC REPORT statements that produce the same report. The examples include a summarization and custom labels.

```sas
ods html5 (id=web) style=daisy file="procreport05.html" (title="Batting Totals in 1986") options(outline='no');

title1 "Batting Totals in 1986";

data baseballNew;
  length team $ 30 ;
  set sashelp.baseball;
run;

proc report data=baseballNew contents="Home Runs and Hits By Team" spanrows;

  column Team Name nHome nHits;

  define Team / group;
  define Name / group;
  break after Team / summarize;

  compute after Team;

  Team="Total for " || Team;
  endcomp;

  rbreak after / summarize;
  compute after;

  Team="Total for All Teams";
  endcomp;
run;
```

```sas
ods html5 (id=web) close;
```

1 The CONTENTS= option provides a descriptive label for the table, which is needed for accessibility.
The SPANROWS option is required to make the table accessible because one or more variables are defined as GROUP or ORDER.

Both procedures use a COMPUTE block to provide a more descriptive label for the summary data. Both procedures produce the same output. The two procedures are included here to show the available options for creating a summarization row with custom labels.

Output

The following images show the partial output for the table.

**Output 5.5  Summarization Rows with Custom Labels**

**Batting Totals in 1986**

<table>
<thead>
<tr>
<th>Team at the End of 1986</th>
<th>Player's Name</th>
<th>Home Runs in 1986</th>
<th>Hits in 1986</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlanta</td>
<td>Harper, Terry</td>
<td>8</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>Horner, Bob</td>
<td>27</td>
<td>141</td>
</tr>
<tr>
<td></td>
<td>Hubbard, Glenn</td>
<td>4</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>Moreno, Omar</td>
<td>4</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>Murphy, Dale</td>
<td>29</td>
<td>163</td>
</tr>
<tr>
<td></td>
<td>Oberfell, Ken</td>
<td>5</td>
<td>136</td>
</tr>
<tr>
<td></td>
<td>Ramirez, Rafael</td>
<td>8</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>Sample, Billy</td>
<td>6</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>Simmons, Tad</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>Thomas, Andres</td>
<td>6</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Virgil, Ozzie</td>
<td>15</td>
<td>80</td>
</tr>
</tbody>
</table>

Total for Atlanta    116        1055

<table>
<thead>
<tr>
<th>Team at the End of 1986</th>
<th>Player's Name</th>
<th>Home Runs in 1986</th>
<th>Hits in 1986</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baltimore</td>
<td>Beniquez, Juan</td>
<td>6</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>Bonilla, Juan</td>
<td>1</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>Dempsey, Rick</td>
<td>13</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>Dwyer, Jim</td>
<td>8</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>Lacy, Lee</td>
<td>11</td>
<td>141</td>
</tr>
<tr>
<td></td>
<td>Lynn, Fred</td>
<td>23</td>
<td>114</td>
</tr>
<tr>
<td></td>
<td>Murray, Eddie</td>
<td>17</td>
<td>151</td>
</tr>
</tbody>
</table>
Example: Detail Report with ACCESSIBLETABLE System Option

Program

This example uses the 9.4m6 system options ACCESSIBLETABLE and ACCESSIBLECHECK and PROC REPORT CAPTION= and CONTENTS= options to create an accessible table. This example shows you how to set up the ODS environment, provide titles for the output, and specify a visible title for the table using the CAPTION= option.

```sas
options accessiblecheck accessibletable; /* 1 */
opts html5 (id=web) style=daisy file="procreport05.html"
   (title="Only Honda Cars") options(outline='no'); /* 2 */

         title1 'Honda Cars Detail Report'; /* 3 */
proc report data=sashelp.cars contents="Honda Cars" caption="Honda Cars"; /* 4 */
   where Make="Honda";
run;

   title1;

ods html5 (id=web) close;
```

1 Use the ACCESSIBLETABLE system option in 9.4M6 to generate accessible tables. The ACCESSIBLECHECK system option detects accessibility problems and writes messages to the SAS log if accessibility problems are found.

2 Here are details about the ODS environment:
   - The ODS HTML5 destination statement sets the ODS environment.
   - The (ID=WEB) option is required for use only with SAS Studio.
The STYLE= option specifies the style Daisy, which is recommended for accessible output.

**Note:** If you are using SAS Studio, you can **configure the STYLE= option** as a default preference.

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. Specifying (TITLE=) is required by WCAG 2.0 success criterion 2.4.2 Page Title.

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 show up as headings in the HTML outline that is read by screen readers.

The TITLE1 statement provides a title for the output.

Starting with SAS 9.4M6, specify the CAPTION= option in the PROC REPORT statement to add captions to tables. The caption is visible in the output when the ACCESSIBLETABLE system option is specified. If no caption is specified, the caption defaults to the text specified by the CONTENTS= option. The CONTENTS option also specifies the text for the entry in the table of contents.

### Output

**Output 5.6  Detail Report Using ACCESSIBLETABLE System Option**

![Honda Cars Detail Report](image)
Example: Accessible Table Using By Directives

About This Example
This example uses the ACCESSIBLETABLE option to enable accessible tables and enable visual captions. When the ACCESSIBLETABLE system option is specified, the captions become visible. This example uses BY directives to insert BY variable values into the caption text.

For information about using the By directives, see “Substituting BY Line Values in a Text String” in Base SAS Procedures Guide.

Program
This example uses the 9.4m6 system options ACCESSIBLETABLE and ACCESSIBLECHECK and PROC REPORT CAPTION= option and the CONTENTS= option to create an accessible table. This example shows you how to set up the ODS environment, provide titles for the output, specify a visible title for the table using the CAPTION= option, and shows how to use the By directives.

```sas
options accessiblecheck accessibleetable; /* 1 */
ods html5 (id=web) style=daisy path='path-filename'
   file='filename'
   (title="Class by Gender")
   contents='filename-to-contents-file'
   frame='filename-to-file-with-frames'
   options(outline='no'); /* 2 */

   title1 'Class Report by Gender'; /* 3 */

   proc sort data=sashelp.class out=class;
      by sex;
   run;

   proc report data=class contents='#BYVAR1 EQ #BYVAL1' caption='Students: #BYVAR1 EQ #BYVAL1';
      by sex*; /* 4 */
   run;

   title1;

   ods html5 (id=web) close;
```

1. Use the ACCESSIBLETABLE system option in 9.4M6 to generate accessible tables. The ACCESSIBLECHECK system option detects accessibility problems and writes messages to the SAS log if accessibility problems are found.

2. Here are details about the ODS environment:
   - The ODS HTML5 destination statement sets the ODS environment.
   - The (ID=WEB) option is required for use only with SAS Studio.
   - The STYLE= option specifies the style Daisy, which is recommended for accessible output.

Note: If you are using SAS Studio, you can configure the STYLE= option as a default preference.
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. Specifying (TITLE=) is required by WCAG 2.0 success criterion 2.4.2 Page Title.

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 show up as headings in the HTML outline that is read by screen readers.

The FILE=, CONTENTS=, FRAME= generates the html output files.

The TITLE1 statement provides a title for the output.

Starting with SAS 9.4M6, specify the CAPTION= option in the PROC REPORT statement to add captions to tables. The caption is visible in the output when the ACCESSIBLETABLE system option is specified. If no caption is specified, the caption defaults to the text specified by the CONTENTS= option. The CONTENTS= option also specifies the text for the entry in the table of contents.

In 9.4M6, By directives can be used with the CONTENTS= option. The CAPTION= option supports the By directives.
Output

Output 5.7  Detail Report Using ACCESSIBLETABLE System Option and By Directives

The Table of Contents

1. The Report Procedure
   Sex=F
     Sex EQ F
     Table 1
   Sex=M
     Sex EQ M
     Table 1

Class Report by Gender

Sex=F

Students: Sex EQ F

<table>
<thead>
<tr>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>F</td>
<td>13</td>
<td>56.5</td>
<td>84</td>
</tr>
<tr>
<td>Barbara</td>
<td>F</td>
<td>13</td>
<td>65.3</td>
<td>98</td>
</tr>
<tr>
<td>Carol</td>
<td>F</td>
<td>14</td>
<td>62.8</td>
<td>102.5</td>
</tr>
<tr>
<td>Jane</td>
<td>F</td>
<td>12</td>
<td>59.8</td>
<td>84.5</td>
</tr>
<tr>
<td>Janet</td>
<td>F</td>
<td>15</td>
<td>62.5</td>
<td>112.5</td>
</tr>
<tr>
<td>Joyce</td>
<td>F</td>
<td>11</td>
<td>51.3</td>
<td>50.5</td>
</tr>
<tr>
<td>Judy</td>
<td>F</td>
<td>14</td>
<td>64.3</td>
<td>90</td>
</tr>
<tr>
<td>Louise</td>
<td>F</td>
<td>12</td>
<td>56.3</td>
<td>77</td>
</tr>
<tr>
<td>Mary</td>
<td>F</td>
<td>15</td>
<td>66.5</td>
<td>112</td>
</tr>
</tbody>
</table>

Class Report by Gender

Sex=M

Students: Sex EQ M

<table>
<thead>
<tr>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Height</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfred</td>
<td>M</td>
<td>14</td>
<td>69</td>
<td>112.5</td>
</tr>
<tr>
<td>Henry</td>
<td>M</td>
<td>14</td>
<td>63.5</td>
<td>102.5</td>
</tr>
<tr>
<td>James</td>
<td>M</td>
<td>12</td>
<td>57.3</td>
<td>83</td>
</tr>
<tr>
<td>Jaffrey</td>
<td>M</td>
<td>13</td>
<td>62.5</td>
<td>84</td>
</tr>
<tr>
<td>John</td>
<td>M</td>
<td>12</td>
<td>59</td>
<td>99.5</td>
</tr>
<tr>
<td>Philip</td>
<td>M</td>
<td>16</td>
<td>72</td>
<td>150</td>
</tr>
<tr>
<td>Robert</td>
<td>M</td>
<td>12</td>
<td>64.8</td>
<td>128</td>
</tr>
<tr>
<td>Ronald</td>
<td>M</td>
<td>15</td>
<td>67</td>
<td>133</td>
</tr>
<tr>
<td>Thomas</td>
<td>M</td>
<td>11</td>
<td>57.5</td>
<td>85</td>
</tr>
<tr>
<td>William</td>
<td>M</td>
<td>15</td>
<td>66.5</td>
<td>112</td>
</tr>
</tbody>
</table>
Creating Accessible Tables with the TABULATE Procedure

About the TABULATE Procedure ................................................................. 40
Recommendations for the TABULATE Procedure ................................. 40
  Recommendations .................................................................................. 40
  What to Avoid ....................................................................................... 41
Example: Simple Table with No Row Dimension .................................... 41
  Program ................................................................................................. 41
  Output .................................................................................................... 42
Example: Table with a Single Row Dimension and No Label .................... 42
  About This Example ............................................................................. 42
  Program ................................................................................................. 43
  Output .................................................................................................... 43
Example: Table with a Single Row Dimension Whose Label Is in a Box .... 44
  About This Example ............................................................................. 44
  Program ................................................................................................. 44
  Output .................................................................................................... 44
Example: Table with Multiple ALL Columns .......................................... 45
  About This Example ............................................................................. 45
  Program ................................................................................................. 45
  Output .................................................................................................... 46
Example: Table with Summation Rows and Custom Labels ...................... 46
  About This Example ............................................................................. 46
  Program ................................................................................................. 46
  Output .................................................................................................... 48
Example: Enable Accessible Tables ......................................................... 49
  About This Example ............................................................................. 49
  Program ................................................................................................. 49
  Output .................................................................................................... 50
Example: Table with BY Directives ......................................................... 50
  About This Example ............................................................................. 50
  Program ................................................................................................. 50
  Output .................................................................................................... 51
About the TABULATE Procedure

Use the TABULATE procedure to present descriptive statistics in tabular format, using some or all of the variables in a data set. You can create a variety of tables ranging from simple to highly customized. The TABULATE procedure computes many of the same statistics that are computed by other descriptive statistical procedures such as MEANS, FREQ, and REPORT.

When used with the ACCESSIBLETABLE system option, PROC TABULATE provides the ability to create accessible output tables to add visual captions to tables.

See also Overview of Table Accessibility on page 11.

Recommendations for the TABULATE Procedure

Recommendations

The General Recommendations for Creating Accessible Output contain best practices. In addition to the general recommendations, follow these specific recommendations if you are using the TABULATE procedure.

Unless otherwise specified, the following recommendations help satisfy WCAG 2.0 success criterion 1.3.1 Info and Relationships.

- Use the CONTENTS= option in the TABLE statement to provide a description of the data presented in the table. The CONTENTS= option specifies the text for the entries in the table of contents that is created.

- The KEYLABEL statement and custom label text for the ALL keyword might be needed to provide a more descriptive label for some columns and rows. This label is especially helpful when there are multiple columns or rows with the same header text where the context of the header cell might not provide enough information to clearly know what information is being presented.

  See Example: Table with Summation Rows and Custom Labels on page 46.

- If, in a row header or column header, there is a single spanned cell across all of the rows or columns, you do not have to include that label in the table if the title or description of the table adequately describes what is being presented. You should evaluate your table and determine which way best presents the information.

- Starting with SAS 9.4M6, specify the ACCESSIBLECHECK system option to detect accessibility problems and write messages to the SAS log if accessibility problems are found.

- Starting with SAS 9.4M6, specify the ACCESSIBLETABLE system option to change the layout of some PROC TABULATE tables to make them accessible and add visual captions to the tables.

- Starting with SAS 9.4M6, specify the CAPTION= option in the TABLE statement to add captions to tables. When the ACCESSIBLETABLE system option is specified, captions specified by the CAPTION= option are visible. If no caption is specified, then the caption defaults to the text specified by the CONTENTS= option in the TABLE statement.

- Starting with SAS 9.4M6, when BY directives are specified in the CAPTION= and CONTENTS= options, labels for the BY group tables are displayed in the table of contents in PDF output and in the contents file in HTML output. The labels are based on the values of the BY variable.

- In SAS 9.4M5 and earlier releases, if your table contains row headers in the first column, for all variables in the row dimension, assign the label for each variable to "". This assignment removes the label that is positioned as a column header for the row headers. Removing the header label resolves a problem screen
readers have with reading tables with variables in the row dimension. Beginning with SAS 9.4M6, specifying
the ACCESSIBLETABLE system option creates accessible tables, including accessible row headers.
See Example: Table with a Single Row Dimension and No Label on page 42.
An alternative is to use the BOX="text" option to create a column header that labels the row headers. See
Example: Table with a Single Row Dimension Whose Label Is in a Box on page 44.

What to Avoid

■ Do not use concatenated variables in the row dimension. Divide the concatenated variables into separate
tables either by using a second TABLE statement or by using a second TABULATE procedure. The proper
headers for concatenated variables in the row dimension in a single table are not communicated to screen
reader users.
■ Do not use the NOCELLMERGE option in the TABLE statement. Using that option prevents the headers from
being correctly assigned to their data cells.

Example: Simple Table with No Row Dimension

Program
This example creates a simple accessible report using the TABULATE procedure. The report has a column
dimension but no row dimension. The example shows how to set up the ODS environment and provide titles and
a description of the table. The example uses the FORMAT procedure to provide more meaningful labels for the
values used in the TABULATE procedure.

```sas
options accessiblecheck;               /*1*/
ods html5 (id=web) style=daisy path='file-path'       /*2*/
   file='filename'
   [title="Students by Gender"]
   options (outline='no');

title1 "Number of Students by Gender";       /*3*/
proc format;
  value $sex
    "F"="Female"
    "M"="Male";
run;

proc tabulate data=sashelp.class;
  format sex $sex.;
  label sex="Gender";
  class Sex;
  table Sex / contents="Gender distribution for the class";  /*4*/
run;

title1;
ods html5 (id=web) close;
```

Example: Simple Table with No Row Dimension

The ACCESSIBLECHECK system option checks the program for common violations of accessibility standards, and writes messages to the SAS log if violations are found.

Here are details about the ODS environment:

- The ODS HTML5 destination statement sets the ODS environment.
- The (ID=WEB) option is required for use only with SAS Studio.
- The STYLE= option specifies the style Daisy, which is recommended for accessible output.

Note: If you are using SAS Studio, you can configure the STYLE= option as a default preference.

- You must supply a valid file path and filename. Alternatively, you can omit the PATH= and FILE= options.
- 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. Specifying (TITLE=) is required by WCAG 2.0 success criterion 2.4.2 Page Title.
- 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 show up as headings in the HTML outline that is read by screen readers.

The TITLE1 statement provides a title for the output.

The CONTENTS= option in the TABLE statement provides a description of the data presented in the table.

Output

Output 6.1  Simple Table with No Row Dimension

Number of Students by Gender

<table>
<thead>
<tr>
<th>Gender</th>
<th>N</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Example: Table with a Single Row Dimension and No Label

About This Example

If your table contains row headers in the first column, the default behavior is to provide a column header for the row headers. However, this causes the procedure to generate a row span (ROWSPAN) attribute in the HTML output, which the screen reader does not read correctly.

To prevent the ROWSPAN attribute from being generated, for all variables in the row dimension, you can assign the label for each variable to "". This assignment removes the label that is positioned as a column header for the row headers.

Note: For another way to eliminate the ROWSPAN attribute, see Example: Table with a Single Row Dimension Whose Label Is in a Box on page 44.
If you are using SAS 9.4M6 or later, then you can specify the ACCESSIBLETABLE system option to generate accessible tables, including accessible row headers.

**Program**

```sas
options accessiblecheck; /* X */
ods html5 (id=web) style=daisy
   path='filepath'
   file='filename' (title="Cholesterol by Gender")
   options(outline='no');
title1 "Cholesterol by Gender";
proc tabulate data=sashelp.heart;
   class Sex Chol_Status;
   table Chol_Status='', Sex / contents="Cholesterol by Gender"; /* 2 */
run;
title1;
ods html5 (id=web) close;
```

1. The ACCESSIBLECHECK system option checks the program for common violations of accessibility standards, and writes messages to the SAS log if violations are found.
2. By default, the Chol_Status variable provides a column header (Cholesterol Status) for the row headers, but does so by generating an unwanted row span, which the screen reader does not read correctly. To remedy this, the example assigns the label for the row dimension expression to "." This removes the column header for the row headers.

   **Accessibility note:** Beginning with SAS 9.4M6, you can specify the ACCESSIBLETABLE system option, and no modifications to the row headers are needed.

**Output**

**Output 6.2  Table with a Single Row Dimension and No Label**

**Cholesterol by Gender**

|       | Sex |   
|-------|-----|---
|       | Female | Male |
| Borderline | 959 | 902 |
| Desirable   | 805 | 600 |
| High        | 1010 | 781 |
Example: Table with a Single Row Dimension Whose Label Is in a Box

About This Example

This example uses the BOX= option in the TABLE statement to create a column header that labels the row headers. The BOX= option places the column header for the row headers in a position where screen reader users can properly understand the information.

This example presents another alternative to the row span issue that is described in Example: Table with a Single Row Dimension and No Label on page 42.

If you are using SAS 9.4M6 or later, then you can specify the ACCESSIBLETABLE system option to generate accessible tables, including accessible row headers.

Program

options accessiblecheck; /*1*/
ods html5 (id=web) style=daisy path='filepath'
    file='filename' (title="Cholesterol by Gender")
    options(outline='no');
title1 "Cholesterol by Gender";
proc tabulate data=sashelp.heart;
    class Sex Chol_Status;
    table Chol_Status="", Sex / box="Cholesterol Status" /*2*/
        contents="Cholesterol by Gender";
run;
title1;
ods html5 (id=web) close;

1 The ACCESSIBLECHECK system option checks the program for common violations of accessibility standards, and writes messages to the SAS log if violations are found.

2 By default, the Chol_Status variable provides a column header (Cholesterol Status) for the row headers, but does so by generating an unwanted ROWSPAN attribute in the HTML output, which the screen reader does not read correctly. To remedy this, the example does the following:
   - assigns the label for the row dimension expression to ""
   - uses the BOX= option to specify a text override for the empty box above the row headers

Accessibility note: Beginning with SAS 9.4M6, you can specify the ACCESSIBLETABLE system option, and no modifications to the row headers are needed.

Output
Output 6.3  Table with a Single Row Dimension Whose Label Is in a Box

### Cholesterol by Gender

<table>
<thead>
<tr>
<th>Cholesterol Status</th>
<th>Sex</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Female</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Borderline</td>
<td>959</td>
<td>902</td>
<td></td>
</tr>
<tr>
<td>Desirable</td>
<td>805</td>
<td>600</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>1010</td>
<td>781</td>
<td></td>
</tr>
</tbody>
</table>

**Example: Table with Multiple ALL Columns**

**About This Example**

The universal class variable ALL summarizes information from multiple categories. ALL summarizes all of the categories for class variables that are in the same parenthetical group. If the variable ALL is not contained in a parenthetical group, then ALL summarizes the categories for the dimension.

In this example, the ALL columns for the largest groups of columns (cholesterol and blood pressure) do not have other header cells above them to describe what they are labeling. Screen reader users cannot easily differentiate between the All Cholesterol totals and the All Blood Pressure totals. Therefore, a custom label is specified for each column to distinguish the column.

Custom labels are not required for the ALL columns for each of the smaller groups, such as cholesterol status for live patients versus cholesterol status for deceased patients. These columns can simply be labeled “All” because they have headers above them that describe what they are totaling.

**Program**

```sas
options accessiblecheck; /*1*/

ods html5 (id=web) style=daisy path='filepath'
file='filename' (title="Patient Status")
opts(outline='no');

title1 "Patients by Gender, Cholesterol Status, and Blood Pressure Status";

proc tabulate data=sashelp.heart;
  class Sex Status Chol_Status BP_Status;
  table Sex all, Status*(Chol_Status all) all="All Cholesterol" /*2*/ Status*(BP_Status all) all="All Blood Pressure" /
    contents="Patients by Gender, Cholesterol Status, and Blood Pressure Status";
  run;

title1;

ods html5 (id=web) close;
```

1 The ACCESSIBLECHECK system option checks the program for common violations of accessibility standards, and writes messages to the SAS log if violations are found.
A custom label is specified for the cholesterol status group and the blood pressure status group. The custom labels identify the respective ALL columns.

**Output**

**Output 6.4 Table with Multiple ALL Columns**

<table>
<thead>
<tr>
<th>Sex</th>
<th>Cholesterol Status</th>
<th>Status</th>
<th>Age Group</th>
<th>Sex</th>
<th>Cholesterol Status</th>
<th>Status</th>
<th>Age Group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Boundary High</td>
<td>Alive</td>
<td>All</td>
<td></td>
<td>Boundary High</td>
<td>Alive</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Female</td>
<td>682</td>
<td>645</td>
<td>580</td>
<td>1915</td>
<td>277</td>
<td>100</td>
<td>422</td>
</tr>
<tr>
<td>Male</td>
<td>954</td>
<td>383</td>
<td>363</td>
<td>1220</td>
<td>368</td>
<td>247</td>
<td>418</td>
</tr>
<tr>
<td>All</td>
<td>1136</td>
<td>988</td>
<td>951</td>
<td>3135</td>
<td>675</td>
<td>407</td>
<td>848</td>
</tr>
</tbody>
</table>

**Example: Table with Summation Rows and Custom Labels**

**About This Example**

As shown in Example: Table with a Single Row Dimension and No Label on page 42, it is recommended that you remove the label that is positioned as a column header for the row headers. The label generates a ROWSPAN attribute in the HTML output, which the screen reader does not read correctly.

This example similarly removes the label that is positioned as a column header for the League row headers. The example then uses the universal class variable ALL to provide custom labels for the Team summary information.

**Note:** If you are using SAS 9.4M6 or later, then you can specify the ACCESSIBLETABLE system option to generate accessible tables, including accessible row headers.

**Program**

```sas
options accessiblecheck; /*1*/
ods html5 (id=web) style=daisy path='filepath'
    file='filename' (title="Home Runs in 1986")
    options(outline='no');

title1 "Home Runs in 1986";

proc tabulate data=sashelp.baseball format=12.0;
class League Team;
var nHome;
label nHome="Home Runs"; /*2*/
keylabel sum=" ";
table League=""*"(Team="" All=""Total Home Runs for the League")/*3*/
    All="Total Home Runs in MLB",nHome /
```

Chapter 6 / Creating Accessible Tables with the TABULATE Procedure
The ACCESSIBLECHECK system option checks the program for common violations of accessibility standards, and writes messages to the SAS log if violations are found.

The LABEL statement provides a more descriptive and intuitive label for the number of home runs.

The TABLE statement creates a row for each value of League. Nested within each row are rows for each value of Team.

The labels for the row dimension expressions are assigned to "". This assignment removes the labels from the row dimensions, thereby preventing the ROWSPAN attributes from being generated.

The universal class variable ALL summarizes information from multiple categories. In this case, ALL is used to summarize quantities for each value of League. The second ALL variable summarizes quantities for the League variable as a whole.

The BOX option puts the column header for the row headers in a position where screen reader users can properly understand the information.
## Output

**Output 6.5** *Table with Summation Rows and Custom Labels*

### Home Runs in 1986

<table>
<thead>
<tr>
<th></th>
<th>Team</th>
<th>Home Runs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>American</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Baltimore</td>
<td>164</td>
</tr>
<tr>
<td></td>
<td>Boston</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>California</td>
<td>163</td>
</tr>
<tr>
<td></td>
<td>Chicago</td>
<td>111</td>
</tr>
<tr>
<td></td>
<td>Cleveland</td>
<td>153</td>
</tr>
<tr>
<td></td>
<td>Detroit</td>
<td>183</td>
</tr>
<tr>
<td></td>
<td>Kansas City</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>Milwaukee</td>
<td>118</td>
</tr>
<tr>
<td></td>
<td>Minneapolis</td>
<td>190</td>
</tr>
<tr>
<td></td>
<td>New York</td>
<td>183</td>
</tr>
<tr>
<td></td>
<td>Oakland</td>
<td>152</td>
</tr>
<tr>
<td></td>
<td>Seattle</td>
<td>153</td>
</tr>
<tr>
<td></td>
<td>Texas</td>
<td>177</td>
</tr>
<tr>
<td></td>
<td>Toronto</td>
<td>167</td>
</tr>
<tr>
<td></td>
<td><strong>Total Home Runs for the League</strong></td>
<td><strong>2185</strong></td>
</tr>
<tr>
<td><strong>National</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Atlanta</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td>Chicago</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>Cincinnati</td>
<td>134</td>
</tr>
<tr>
<td></td>
<td>Houston</td>
<td>114</td>
</tr>
<tr>
<td></td>
<td>Los Angeles</td>
<td>111</td>
</tr>
<tr>
<td></td>
<td>Montreal</td>
<td>102</td>
</tr>
<tr>
<td></td>
<td>New York</td>
<td>139</td>
</tr>
<tr>
<td></td>
<td>Philadelphia</td>
<td>144</td>
</tr>
<tr>
<td></td>
<td>Pittsburgh</td>
<td>106</td>
</tr>
<tr>
<td></td>
<td>San Diego</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>San Francisco</td>
<td>109</td>
</tr>
<tr>
<td></td>
<td>St Louis</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td><strong>Total Home Runs for the League</strong></td>
<td><strong>1390</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Total Home Runs in MLB</strong></td>
<td><strong>3575</strong></td>
</tr>
</tbody>
</table>
Example: Enable Accessible Tables

About This Example

This example uses the ACCESSIBLETABLE option to enable accessible tables and enable visual captions. In the first TABLE statement, the CAPTION= option is specified while accessible tables are enabled. This enables the captions to become visible. In the second TABLE statement, neither the CAPTION= option nor the CONTENTS= option is specified. Therefore, the BY group labels are displayed by default in the PDF table of contents and the HTML contents file.

For information about the CAPTION= option, see “CAPTION=text #BYLINE #BYVAL #BYVAR” in Base SAS Procedures Guide.

Program

```sas
options accessibleetable accessiblecheck;                             /*1*/

   title 'City and Highway Milage by Model, Type, and Origin';

   ods html5 (id=web) style=daisy path='filepath'
      file='filename'
      (title='Energy Expenditures in Millions of Dollars')
      contents='filename'
      frame='filename'
      options(outline='no');

   proc sort data=sashelp.cars out=cars;
      by type origin;
   run;

   proc tabulate data=cars;
      by type origin;
      class model;
      var mpg_city mpg_highway;
      table model, mpg_city * (mean) /caption="Mean City Milage";        /*2*/
      table model, mpg_highway * (mean);                                 /*3*/
   run;

   title1;
   ods html5 (id=web) close;
```

1 The ACCESSIBLETABLE system option alters the layout of some generated tables in order to make the tables accessible.

   The ACCESSIBLECHECK system option checks the program for common violations of accessibility standards, and writes messages to the SAS log if violations are found.

2 The CAPTION= option in the TABLE statement provides a description of the data presented in the table. Because the ACCESSIBLETABLE system option is specified, the caption is visible.

3 No description of the data presented in the second table is specified by the CONTENTS= option or the CAPTION= option. However, the BY group label is used as the table caption.
Output

The following output shows the HTML frame file. In the table of contents, default BY group labels are used. The first table displays the caption specified by the CAPTION= option. The second table uses the BY group label as a table caption, because no CAPTION= or CONTENTS= option was specified in the TABLE statement.

Output 6.6  HTML Report with Default BY Group Labels in the Table of Contents

Example: Table with BY Directives

About This Example

This example uses the ACCESSIBLETABLE option to enable accessible tables and enable visual captions. When the ACCESSIBLETABLE system option is specified, the captions become visible. This example uses BY directives to insert BY variable values into the caption text.

For information about the CAPTION= option, see "CAPTION=text #BYLINE #BYVAL #BYVAR" in Base SAS Procedures Guide.

Program

options accessibleetable accessiblecheck; /* */
The ACCESSIBLETABLE system option alters the layout of some generated tables in order to make the tables accessible.

The ACCESSIBLECHECK system option checks the program for common violations of accessibility standards, and writes messages to the SAS log if violations are found.

The CAPTION= option in the TABLE statement provides a description of the data presented in the table. The BY directives insert BY variable labels and values into the caption.

Output

The following output shows the HTML frame file. In the table of contents, BY group labels are used. The tables display captions specified by the CAPTION= option. The caption consists of text string and BY variable labels and values.
Output 6.7  HTML Report with Captions Created by BY Directives

The Table of Contents

1. The Tabulate Procedure
   Type=Hybrid Origin=Asia
   Cross-tabular summary report
   Summary Table 1: Type = Hybrid and Origin = Asia
   Summary Table 2: Type = Hybrid and Origin = Asia
   Type=SUV Origin=Asia
   Cross-tabular summary report
   Summary Table 1: Type = SUV and Origin = Asia
   Summary Table 2: Type = SUV and Origin = Asia
   Type=SUV Origin=Europe
   Cross-tabular summary report
   Summary Table 1: Type = SUV and Origin = Europe
   Summary Table 2: Type = SUV and Origin = Europe
   Type=SUV Origin=USA
   Cross-tabular summary report
   Summary Table 1: Type = SUV and Origin = USA
   Summary Table 2: Type = SUV and Origin = USA
Creating Accessible Tables and Reports with the Report Writing Interface

Create an Accessible Table
Program ............................................................................................................. 53
Output .................................................................................................................. 56
Key Ideas .............................................................................................................. 56
See Also .............................................................................................................. 57

Create a Simple Accessible Report
About This Example .......................................................................................... 57
Program .............................................................................................................. 57
Output .................................................................................................................. 59
Key Ideas .............................................................................................................. 62
See Also .............................................................................................................. 62

Create an Accessible Report That Includes Text, Tables, and Graphs
About This Example .......................................................................................... 63
Program .............................................................................................................. 63
Output .................................................................................................................. 70
Key Ideas .............................................................................................................. 72
See Also .............................................................................................................. 72

Create an Accessible Table

Program
The following table, in its original form, comes from the Daniel Kummer’s paper: Paper SAS330-2014 Toe to Toe: Comparing ODS LAYOUT and the ODS Report Writing Interface Daniel Kummer, SAS Institute Inc., Cary, NC. Accessibility features have been added to the example code to make this table accessible to screen reader users and to people with certain visual impairments.

To create the input data sets and formats, see Create the PacificSum and AcmePacific Data Sets on page 111 and Create the StateP Format on page 110.

```sas
options accessiblecheck accessibletable; /*1*/
ods html5 (id=web) style=daisy path='file-path'
   file='file-name'
   (title='Create an Accessible Table Using the Report Writing Interface')
   options(outlines='no'); /*2*/
proc odstext contents=''; /*3*/
   h1 'Create an Accessible Table Using the Report Writing Interface';
```
When using the Report Writing Interface, you can add descriptions to tables. In addition to color, you can use ODS style attributes to change the font type and add underlining to highlight data. This makes your report accessible to users with color deficiencies.

```
proc odstext contents=''
  h2 'Table Showing Percent Change in Units Sold and Median List Price'; /*4*/
run;

data _null_; set pacificSum (where=(propGroup='Commercial')) end=done;
  by state year;
  retain units12 price12;
  if _N_ = 1 then do;
    dcl odsout dak(); /*5*/
    dak.table_start(label:'Percent Change in Units Sold and Median List Price',
      caption:'percent change and median list price'); dak.head_start();
    dak.row_start();
      dak.format_cell(data: 'State', row_span: 2, vjust: 'center', style_attr: 'font_weight=bold');
      dak.format_cell(data: 'Units Sold', column_span: 2, style_attr: 'font_weight=bold');
      dak.format_cell(data: 'Median List Price', column_span: 2, style_attr: 'font_weight=bold');
    dak.row_end();
    dak.row_start();
      dak.format_cell(data: '#', style_attr: 'font_weight=bold', width: '.75in');
      dak.format_cell(data: '% Change *', style_attr: 'font_weight=bold', width: '.75in');
    dak.row_end();
    dak.head_end();
  end;
  if first.state and first.year then do;
    units12 = sUnits;
    price12 = mPrice;
  end;
  if last.state and last.year then do;
    changeUnits = ( ( sUnits - units12 ) / units12 );
    changePrice = ( ( mPrice - price12 ) / price12 );
  end;
  dak.row_start();
  dak.format_cell(data: state, just: 'left', style_elem:'RowHeader'); /*7*/
  dak.format_cell(data: sUnits, format: 'comma12.');
  if changeUnits GE 0 then dak.format_cell(data: changeUnits, format: 'percentn10.3',
    style_attr: 'fontstyle=italic fontweight=bold'); /*8*/
  else dak.format_cell(data: changeUnits,
```
The ACCESSIBLECHECK system option checks for common violations of accessibility standards and writes messages to the SAS log if problems are found.

The ACCESSIBLETABLE system option makes captions visible.

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. Specifying (TITLE=) is required by WCAG 2.0 success criterion 2.4.2 Page Title.

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.

Note: If you are using SAS Studio, you can configure the STYLE= option as a default preference. See Configure SAS Studio for Accessible Output on page 5.

The PROC ODSTEXT step creates a H1 heading for the entire document, along with some explanatory text. The CONTENTS= option is blank so that the procedure title does not go into the default table of contents.

The PROC ODSTEXT step creates a H2 heading for the table.

The TABLE_START method begins the table. The LABEL= option specifies the table’s label that is used if a table of contents is created.

The CAPTION= option adds a description to the table. This text can be read by screen readers. To make the caption visible, specify the ACCESSIBLETABLE system option.

The HEAD_START method and HEAD_END method create header for the table. It is recommended that you create headers for tables. This helps keyboard and screen reader users navigate the table.

The STYLE_ELEM:’RowHeader’ specifies that the style specifications for these cells come from the RowHeader style element. This ensures that the cells are styled as row headers.
The \texttt{STYLE_ATTR:} argument enables you to customize the text inside the table cell. It is recommended that you use bold, italic, underline, and so on, to emphasize specific text or data. Using styles other than or in addition to color is essential for those with color blindness. In this table, bold italic font is used to indicate positive percent change. Underlined text is used to indicate negative percent change.

\section*{Output}

The following output shows a table with accessibility features added.

\subsection*{Output 7.1 Create an Accessible Table with the Report Writing Interface}

\section*{Create an Accessible Table Using the Report Writing Interface}

When using the Report Writing Interface, you can add descriptions to tables.

In addition to color, you can use ODS style attributes to change the font type and add underlining to highlight data. This makes your report accessible to users with color deficiencies.

\subsection*{Table Showing Percent Change in Units Sold and Median List Price}

\begin{table}[h]
\centering
\begin{tabular}{|l|c|c|c|c|}
\hline
\textbf{State} & \textbf{Units Sold} & \textbf{\% Change} & \textbf{Median List Price} & \textbf{\% Change} \\
\hline
Alaska & 212,687 & 3.13\% & $312,650 & -1.31\% \\
California & 166,787 & 1.28\% & $294,515 & -4.25\% \\
Hawaii & 222,071 & 77.44\% & $294,362 & -13.96\% \\
Oregon & 225,676 & 0.49\% & $279,021 & -14.11\% \\
Washington & 244,856 & 12.68\% & $315,886 & 7.18\% \\
\hline
\end{tabular}
\caption{Table Showing Percent Change in Units Sold and Median List Price}
\end{table}

\section*{Key Ideas}

\begin{itemize}
\item Use the \texttt{ACCESSIBLECHECK} system option to check for common accessibility violations.
\item Supply a title for the output in the ODS HTML5 statement. This is required by WCAG 2.0 success criterion 2.4.2 Page Title.
\item The ODS HTML5 destination together with the Daisy style is recommended for accessible output. See \textit{Using ODS Styles to Create Accessible Output on page 91}.
\item Use the \texttt{H} statement to add headings that screen readers can read to the report. This is required by WCAG 2.0 success criteria 2.4.6 Headings and Labels.
\item When adding headings, it is recommended that reports have one H1 heading, and all the other headings should be subordinate with no skipped levels. It is okay to have multiple headings that are H2, H3, and so on.
\item Use visual methods in addition to color to bring focus to data and text. This is required by WCAG 2.0 success criteria 1.4.1 Use of Color.
\item Use the \texttt{DESCRIPTION=} or \texttt{CAPTION=} option where appropriate. Generally, always use \texttt{DESCRIPTION=} with tables and images, or \texttt{CAPTION=} with tables. However, when using regions and layouts for layout purposes only, \texttt{DESCRIPTION=} might be redundant.
\end{itemize}
Create a Simple Accessible Report

About This Example

This example creates a report with text and tables that are accessible to keyboard and screen reader users. The sections of the report are created using PROC ODSTEXT steps with headings added. The headings can be read correctly by screen readers and aid in navigation. A table of contents with links to the sections helps keyboard and screen reader users navigate the report.

The numbered annotations explain the graph’s accessibility features.

Program

To create the input data set AMRPop and the formats, see Create the AMRPop Data Set and Population Format on page 111.

```
options accessiblecheck; /*1*/
ods html5 (id=web) style=daisy path='file-path'
   file='file-name'
   (title='North and South American Countries by Population')
   options(outline='no'); /*2*/
proc odstext contents='';
   h1 'North and South American Countries by Population'; /*3*/
   run;
ods layout gridded
description='Table of Contents.'; /*4*/
ods region;
proc odstext contents='';
   h2 'Table of Contents'; /*5*/
   list;
   item '<a href="#toc1">Why Add Headings to Your Output?</a>'; /*6*/
   item '<a href="#toc2">WCAG Success Criterion</a>'; /*6*/
   item '<a href="#toc3">Add links to your output from a Table of Contents</a>'; /*6*/
end;
```
The ACCESSIBLECHECK system option checks for common violations of accessibility standards and writes messages to the SAS log if problems are found.

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. Specifying (TITLE=) is required by WCAG 2.0 success criterion 2.4.2 Page Title.
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 show up as headings in the HTML outline that is read by screen readers.

Note: If you are using SAS Studio, configure the STYLE= option as a default preference. See Configure SAS Studio for Accessible Output on page 5.

3 This PROC ODSTEXT step creates a H1 heading for the entire document. The CONTENTS= option is blank so that the procedure title does not go into the default table of contents.

4 The ODS LAYOUT GRIDDED and ODS REGION statements create a container for the table of contents. The DESCRIPTION= option adds a description that can be read by screen readers.

5 The first PROC ODSTEXT step creates a table of contents that links to the three sections in the report. The H1 statement creates a heading for entire page that can be read correctly by screen readers. The ITEM statements together with anchor tags create the links to the different sections.

6 The second PROC ODSTEXT steps create the section “Why Add Headings to Your Output”. The H2 statement creates a second-level heading that can be read correctly by screen readers. The <SPAN ID=> tag adds an ID to the section for linking. The ITEM statements create the text for this section.

7 The second ODS HTML5 statement changes the base name for the anchor tag to create the table of contents for the PROC PRINT output. The ANCHOR= option specifies a unique base name for the anchor tag that identifies each output object in the current body file.

8 The last PROC ODSTEXT step creates a list of links to the output tables. This acts as a table of contents for the PROC PRINT output.

9 The PROC PRINT step prints a data set with a BY variable. Each BY table is a link. The linking is enabled by the previous PROC ODSTEXT step. The LABEL specifies to use the variables' labels as column headings. The CONTENTS= option specifies text for the links in the HTML contents file.

Output

The following output shows a report with added accessibility features.
Chapter 7 / Creating Accessible Tables and Reports with the Report Writing Interface

Output 7.2  Output with Headers, Links, and a Table of Contents

North and South American Countries by Population

Table of Contents

- Why Add Headings to Your Output?
- WCAG Success Criterion
- Add links to your output from a Table of Contents

Why Add Headings to Your Output?

You can use the H statement to add headings to your output. Adding headings that describe the topic or purpose help with accessibility. This is required by WCAG 2.0 success criteria 2.4.6.

Headings also insure information, structure, and relationships are conveyed through presentation and can be programmatically determined or are available in text. This is required by WCAG 2.0 success criteria 1.3.1.

WCAG Success Criterion this Example Satisfies

- WCAG 2.0 success criterion 2.4.2. Page Title
- WCAG 2.0 success criteria 1.3.1 Info and Relationships
- WCAG 2.0 success criteria 2.4.6. Headings and Labels
- WCAG 2.0 success criterion 2.4.1. Bypass Blocks

Add Links to Your Output from a Table of Contents

You can create a table of contents, and then link from the TOC to the output.

- Small populations
- Medium populations
- Large populations
- Huge populations
### Population Size=Small

<table>
<thead>
<tr>
<th>Country</th>
<th>Population</th>
<th>Gross National Income per Capita (PPP Int.$ 2004)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAINT KITTS/NEVIS</td>
<td>42,696</td>
<td>11190</td>
</tr>
<tr>
<td>DOMINICA</td>
<td>78,940</td>
<td>5250</td>
</tr>
<tr>
<td>ANTIGUA/BARBUDA</td>
<td>81,485</td>
<td>10360</td>
</tr>
<tr>
<td>GRENAADA</td>
<td>102,024</td>
<td>7000</td>
</tr>
<tr>
<td>SAINT VINCENT/GRENADINES</td>
<td>119,051</td>
<td>6250</td>
</tr>
<tr>
<td>ST. LUCIA</td>
<td>160,765</td>
<td>5560</td>
</tr>
<tr>
<td>BARBADOS</td>
<td>269,556</td>
<td>15060</td>
</tr>
<tr>
<td>BELIZE</td>
<td>289,738</td>
<td>6510</td>
</tr>
<tr>
<td>BAHAMAS</td>
<td>323,063</td>
<td>16140</td>
</tr>
<tr>
<td>SURINAME</td>
<td>449,238</td>
<td></td>
</tr>
<tr>
<td>GUYANA</td>
<td>751,218</td>
<td>4110</td>
</tr>
</tbody>
</table>

### Population Size=Medium

<table>
<thead>
<tr>
<th>Country</th>
<th>Population</th>
<th>Gross National Income per Capita (PPP Int.$ 2004)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRINIDAD AND TOBAGO</td>
<td>1,305,236</td>
<td>11180</td>
</tr>
<tr>
<td>JAMAICA</td>
<td>2,650,713</td>
<td>3630</td>
</tr>
<tr>
<td>PANAMA</td>
<td>3,231,502</td>
<td>6870</td>
</tr>
<tr>
<td>URUGUAY</td>
<td>3,463,197</td>
<td>9070</td>
</tr>
<tr>
<td>COSTA RICA</td>
<td>4,327,228</td>
<td>9530</td>
</tr>
<tr>
<td>NICARAGUA</td>
<td>5,486,685</td>
<td>3300</td>
</tr>
<tr>
<td>PARAGUAY</td>
<td>6,158,259</td>
<td>4870</td>
</tr>
<tr>
<td>EL SALVADOR</td>
<td>6,880,951</td>
<td>4980</td>
</tr>
<tr>
<td>HONDURAS</td>
<td>7,204,723</td>
<td>2710</td>
</tr>
<tr>
<td>HAITI</td>
<td>8,527,777</td>
<td>1680</td>
</tr>
<tr>
<td>DOMINICAN REPUBLIC</td>
<td>8,894,907</td>
<td>6750</td>
</tr>
<tr>
<td>BOLIVIA</td>
<td>9,182,015</td>
<td>2590</td>
</tr>
</tbody>
</table>
Key Ideas

- Use the ACCESSIBLECHECK system option to check for common accessibility violations.
- The ODS HTML5 destination together with the Daisy style is recommended for accessible output. See Using ODS Styles to Create Accessible Output on page 91.
- Supply a title for the output in the ODS HTML5 statement. This is required by WCAG 2.0 success criterion 2.4.2 Page Title.
- Create a table of contents that consists of links to the sections, tables, and graphs in the report. Creating a table of contents at the top of your report with links helps keyboard and screen reader users. This satisfies the WCAG 2.0 success criterion 2.4.1.
- Use the H statement to add headings that screen readers can read to the report, and to each section, table, and graph. The headings should also be used to define an outline within the report. This is required by WCAG 2.0 success criteria 1.3.1 Info and Relationships and WCAG 2.0 success criteria 2.4.6 Headings and Labels.
- When adding headings, it is recommended that reports have one H1 heading, and all the other headings should be subordinate with no skipped levels. It is acceptable to have multiple headings that are H2, H3, and so on.
- Use the LABEL option when using PROC PRINT. This ensures that if you print two different data sets and the labels in the data sets are different, you can still force one of the tables to use the same label.
- Use the DESCRIPTION= option to add descriptive text where appropriate. Generally, always use DESCRIPTION= with tables and images. However, when using regions and layouts for layout purposes only, DESCRIPTION= might be redundant. This is required by WCAG 2.0 success criterion 1.3.1 Info and Relationships.

See Also

Create an Accessible Report That Includes Text, Tables, and Graphs

About This Example

The following example, in its original form, comes from Daniel Kummer’s paper: Paper SAS330-2014 Toe to Toe: Comparing ODS LAYOUT and the ODS Report Writing Interface Daniel Kummer, SAS Institute Inc., Cary, NC. Accessibility features have been added to the example code to make the report accessible to keyboard and screen reader users. For input data sets and formats, see Create the PacificSum and AcmePacific Data Sets on page 111 and Create the StateP Format on page 110.

Program

*Pie chart template;
proc template; /*1*/
define statgraph basechart;
dynamic width height;
begingraph / pad=0 border=false opaque=false
designwidth=width designheight=height;
layout region / border=false;
piechart category=propType response=soldUnits /
datalabelcontent=(category percent)
datalabellocation=outside
datalabelattrs=GraphValueText(size=9pt family='Helvetica')
categorydirection=clockwise;
endlayout;
endgraph;
end;
run;

/* define general options */
ods escapechar='^'; /*2*/

/* define colors used for the background */
%let backBlue=#006699; /*3*/

options accessiblecheck;
ods html5 (id=web) style=daisy path='file-path' file='file-name'
(title='Create an Accessible Report') accessible_graph
options (outline='no'); /* 4 */
ods layout gridded width=16in height=2in columns=2 style={background=&backBlue.}; /* 5 */
ods region width=8in column=2 height=2in;
proc odstext contents='';
  h1 'S={background=&backBlue. foreground=white font_size=50pt}
     ACME Sales Report Pacific Region'; /* 6 */
run;
ods layout end;
ods layout gridded width=16in columns=2
  description='Table of Contents and Introduction'; /* 7 */
ods region column=1 width=8in;
proc odstext contents='';
  h2 'Table of Contents'; /* 8 */
  list;
    item '<a href="#toc1">Introduction</a>'; /* 9 */
    item '<a href="#toc2">2013 Numbers for Commercial Properties</a>'; /* 10 */
    item '<a href="#toc3">2013 Numbers for Residential Properties</a>'; /* 11 */
  end;
run;
ods layout gridded width=16in columns=2
  description='Table of Contents and Introduction'; /* 12 */
ods region column=1 width=8in;
proc odstext contents='';
  h2 'Introduction'; /* 13 */
  p 'Commercial property sales accelerated in 2013, continuing this year's recovery after activity came
to a near-halt in 2012 during the financial crisis and recession.';
  p 'Buoyed by steadying job growth, improving real estate fundamentals and eager capital, buyers throughout
the first 11 months of 2013 ponied up more than $305 billion to acquire office buildings,
industrial properties, retail centers, hotels, apartments and development sites around the U.S.,
according to ACME Analytics, which tracks commercial real estate sales of more than $5 million.';
run;
ods graphics / outputfmt=svg width=4in;
proc odstext contents='';
  h3 'Units Sold by Property Group for Years 2010-2013'; /* 14 */
run;
proc sgplot data=acmePacific noborder
  description='Bar chart showing units sold'; /* 15 */
  styleattrs datacolors=(darkblue lightblue)
    datacontrastcolors=(darkblue lightblue); /* 16 */
vbar year / response=soldUnits
  barwidth=.8
  group=propgroup groupdisplay=cluster;
yaxis values=(800000 to 1150000 by 50000)
  label='Units Sold (Sum)';
xaxis label='Year';
keylegend / location=inside position=topright
  title='Property Group' across=1 noborder;
run;
ods graphics / reset=width;
Create an Accessible Report That Includes Text, Tables, and Graphs

```sas
ods region width=8in column=2;

data _null_;
dcl odsout dak();
dak.image(file:'your-file-path\pacificMed.jpg',
   description:'States of the Pacific Region: Alaska, Washington, Oregon, California, Hawaii',
   style_attr:'height=5in'};/*12*/
run;
ods layout end;
ods layout gridded columns=2 width=16in
description='Commercial Properties';/*13*/
/* region to display the heading for 2013 & commercial */
ods region width=16in style={background=&backBlue.} column_span=2;
proc odstext contents=''
   h2 '&lt;span id="toc2"&gt;$S=${background=&backBlue.
   foreground=white
   font_size=10pt}"2013 Numbers for Commercial Properties&lt;/span&gt;';
run;
ods region column=1 width=8in ;/* First pie chart */
proc odstext contents=''
   h3 'Units Sold for Commercial Property Types';
run;
proc sgrender data=acmepacific template=basechart
   description='Pie chart showing units sold for commercial property types';/*14*/
   where year=2013 and propGroup = 'Commercial';
   dynamic width='3in' height='2.5in';
run;
/* region to display the first custom RWI table */
ods region column=2 width=8in;
proc odstext contents=''
   h3 'Percent Change in Units Sold and Median List Price for Commercial Property';
run;
data _null_;
   set pacificSum (where=(propGroup='Commercial')) end=done;
   by state year;
   retain units12 price12;
   if _N_ = 1 then do;
     dcl odsout dak();
     dak.table_start(label:'Percent Change in Units Sold and Median List Price for Commercial Property.',
        description:'percent change in units sold and median list price for commercial property'/*15*/
     dak.head_start();/*16*/
     dak.row_start();
     dak.format_cell(data:'State',
        row_span:2,
        vjust:'center',
        style_attr:'font_weight=bold');
     dak.format_cell(data:'Units Sold',
        column_span:2,
        style_attr:'font_weight=bold');
     dak.format_cell(data:'Median List Price',
```
Chapter 7 / Creating Accessible Tables and Reports with the Report Writing Interface

column_span:2,
style_attr:'font_weight=bold'};
dak.row_end();
dak.row_start();
dak.format_cell(data:'#', style_attr:'font_weight=bold', width:'.75in');
dak.format_cell(data:'% Change *', style_attr:'font_weight=bold', width:'.75in');
dak.format_cell(data:'#', style_attr:'font_weight=bold', width:'.75in');
dak.format_cell(data:'% Change *', style_attr:'font_weight=bold', width:'.75in');
dak.row_end();
dak.head_end();
end;

if first.state and first.year then do;
  units12 = sUnits;
  price12 = mPrice;
end;

if last.state and last.year then do;
  changeUnits = ( ( sUnits - units12 ) / units12 );
  changePrice = ( ( mPrice - price12 ) / price12 );

dak.row_start();
  dak.format_cell(data:state, just:'left', style_elem:'RowHeader'); /*17*/
  dak.format_cell(data:sUnits, format:'comma12.');//
  if changeUnits GE 0 then dak.format_cell(data:changeUnits,
    format:'percentn10.3',
    style_attr:'fontstyle=italic fontweight=bold'); /*18*/
  else dak.format_cell(data:changeUnits,
    format:'percentn10.3',
    style_attr:'textdecoration=underline');
  dak.format_cell(data:mPrice, format:'dollar12.');//
  if changePrice GE 0 then dak.format_cell(data:changePrice,
    format:'percentn10.3',
    style_attr:'fontstyle=italic fontweight=bold');
  else dak.format_cell(data:changePrice,
    format:'percentn10.3',
    style_attr:'textdecoration=underline');

dak.row_end();
end;

if done then do;
  dak.foot_start();
  dak.row_start();
    dak.cell_start();
      dak.format_text(data:'* compared to last Years Results',
        just:'left');
    dak.cell_end();
  dak.row_end();
  dak.foot_end();

dak.table_end();
end;
run;
ods layout end;
ods layout gridded width=16in columns=2
### 2013 Numbers for Residential Properties

#### Percent Change in Units Sold and Median List Price

```plaintext
<table>
<thead>
<tr>
<th>State</th>
<th>Units Sold % Change</th>
<th>Median List Price % Change</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

if first.state and first.year then do;
units12 = sUnits;
```

Create an Accessible Report That Includes Text, Tables, and Graphs 67
price12 = mPrice;
end;

if last.state and last.year then do;
changeUnits = ( ( sUnits - units12 ) / units12 );
changePrice = ( ( mPrice - price12 ) / price12 );

dak.row_start();
    dak.format_cell(data:state, just:'left', style_elem:'RowHeader');
    dak.format_cell(data:sUnits, format:'comma12.');
    if changeUnits GE 0 then dak.format_cell(data:changeUnits,
        format:'percentn10.3',
        style_attr:'fontstyle=italic fontweight=bold');
    else dak.format_cell(data:changeUnits,
        format:'percentn10.3',
        style_attr:'textdecoration=underline');
    dak.format_cell(data:mPrice, format:'dollar12.');
    if changePrice GE 0 then dak.format_cell(data:changePrice,
        format:'percentn10.3',
        style_attr:'fontstyle=italic fontweight=bold');
    else dak.format_cell(data:changePrice,
        format:'percentn10.3',
        style_attr:'textdecoration=underline');

dak.row_end();
end;

if done then do;
dak.foot_start();
dak.row_start();
dak.cell_start();
dak.format_text(data:'* compared to last Years Results',
    just:'left');
dak.cell_end();
dak.row_end();
dak.foot_end();
dak.table_end();
end;
run;

/* region to display second pie chart */
ods region width=8in;
proc odstext contents='';
    h3 'Units Sold for Residential Property Types';
run;
proc sgrender data=acmepacific template=basechart
description='Pie chart showing units sold for residential property types.';
    where year=2013 and propGroup = 'Residential';
dynamic width='3.45in' height='2.5in';
run;

/*end the gridded layout*/
ods layout end;
ods html5 (id=web) close;

1 The ACCESSIBLECHECK system option checks for common violations of accessibility standards and writes messages to the SAS log if problems are found.
The PROC TEMPLATE statements create the template for the pie charts. Options DESIGNWIDTH= and DESIGNHEIGHT= in the BEGINGRAPH statement specify the pie chart size to minimize scaling. Dynamic variables are used to specify the dimensions so that the size of each pie chart can be adjusted without having to modify the template code. Because the pie slices are filled, the DATALABELPOSITION= option places the data labels outside of the pie slices. The DATALABELATTRS= option specifies the data label font and font size. Select a font and font size that optimize data-label visibility.

Increase the contrast between the text and background color. The original color was a lighter shade of blue. Adequate contrast is necessary for all users, especially users with low vision.

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. Specifying (TITLE=) is required by WCAG 2.0 success criterion 2.4.2 Page Title.

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 show up as headings in the html outline that is read by screen readers.

Note: If you are using SAS Studio, configure the STYLE= option as a default preference. See Configure SAS Studio for Accessible Output on page 5.

The ODS LAYOUT GRIDDED statement and the ODS LAYOUT REGION statement begin the first layout container and region that contain the report heading. Although the DESCRIPTION= option is available for these statements, in this case, it is not necessary to use DESCRIPTION=. The H1 statement creates the header and adds text that conveys the content of the entire report. Using the DESCRIPTION= option too often could result in redundancy.

The PROC ODSTEXT H statements create headings that can be read correctly by screen readers. The headings act as an outline for screen reader users. When adding headings, it is recommended that reports have one H1 heading, and all the other headings should be subordinate with no skipped levels. It is acceptable to have multiple headings that are H2, H3, and so on. The CONTENTS= option is blank so that the procedure title does not go into the default table of contents.

This PROC ODSTEXT step creates a table of contents that links to the sections in the report. The H2 statement creates a second-level heading that can be read correctly by screen readers. The ITEM statements together with anchor tags create the links to the different sections.

The DESCRIPTION= option adds alternative text to the first pie chart.

The STYLEATTRS statement specifies the fill and outline colors for the bars. In this case, a color is specified for each of two group values. The chosen colors enhance the dark and light contrast.

The Report Writing Interface IMAGE method adds an image to the report. The DESCRIPTION= option adds alternative text that describes the image.

The ODS LAYOUT GRIDDED statements creates the layout container that contains the pie chart and table for commercial properties. In this case, it is appropriate to use the DESCRIPTION= option because it can be used as a landmark to help screen reader users quickly find the section of the report that contains the pie chart and table for commercial properties.

The TABLE_START method begins the table. The LABEL= option specifies the table’s label that is used if a table of contents is created. The DESCRIPTION= options adds alternative text that describes the table.
The HEAD_START method and the HEAD_END method create a header for the table. It is recommended that you always create headers for tables. This helps keyboard and screen reader users navigate the table.

The STYLE_ELEM:'RowHeader' specifies that the style specifications for these cells come from the RowHeader style element. This ensures that the cells are styled as row headers. This also helps keyboard and screen reader users navigate the table.

The STYLE_ATTR: argument enables you to customize the text inside the table cell. It is recommended that you use bold, italic, underline, and so on, to emphasize specific text or data. Using styles other than or in addition to color is essential for those with color blindness. In this table, bold italic font is used to indicate positive percent change. Underlined text is used to indicate negative percent change.

The DESCRIPTION= option adds alternative text to the second pie chart.

**Output**

The following output shows a table with accessibility features added. To view the accessible output, see comprehensiveExample.html.
Output 7.3  Creating an Accessible Report with Text, Tables, and Graphs

ACME Sales Report
Pacific Region

Table of Contents

- Introduction
- 2013 Numbers for Commercial Properties
- 2013 Numbers for Residential Properties

Introduction

Commercial property sales accelerated in 2013, continuing this year’s recovery after activity came to a near-halt in 2012 during the financial crises and recession.

Based by declining job growth, improving real estate fundamentals and greater capital, buyers throughout the first 11 months of 2013 poured up more than $300 billion to acquire office buildings, industrial properties, retail centers, hotels, apartments and development sites around the U.S., according to ACME Analytics, which tracks commercial real estate sales of more than $10 million.

Units Sold by Property Group for Years 2010-2013

2013 Numbers for Commercial Properties

Units Sold for Commercial Property Types

Percent Change in Units Sold and Median List Price for Commercial Property

<table>
<thead>
<tr>
<th>State</th>
<th>Units Sold</th>
<th>Median List Price</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td></td>
<td>% Change</td>
<td>% Change</td>
</tr>
<tr>
<td>Alaska</td>
<td>212,667</td>
<td>$312,690</td>
</tr>
<tr>
<td>California</td>
<td>166,767</td>
<td>$294,515</td>
</tr>
<tr>
<td>Hawaii</td>
<td>222,671</td>
<td>$294,362</td>
</tr>
<tr>
<td>Oregon</td>
<td>235,678</td>
<td>$379,631</td>
</tr>
<tr>
<td>Washington</td>
<td>244,660</td>
<td>$315,668</td>
</tr>
</tbody>
</table>
| * compared to last years results

2013 Numbers for Residential Properties

Percent Change in Units Sold and Median List Price

<table>
<thead>
<tr>
<th>State</th>
<th>Units Sold</th>
<th>Median List Price</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td></td>
<td>% Change</td>
<td>% Change</td>
</tr>
<tr>
<td>Alaska</td>
<td>199,623</td>
<td>$293,093</td>
</tr>
<tr>
<td>California</td>
<td>239,805</td>
<td>$287,328</td>
</tr>
<tr>
<td>Hawaii</td>
<td>240,519</td>
<td>$287,363</td>
</tr>
<tr>
<td>Oregon</td>
<td>256,772</td>
<td>$339,972</td>
</tr>
<tr>
<td>Washington</td>
<td>163,290</td>
<td>$232,154</td>
</tr>
</tbody>
</table>
| * compared to last years results

Units Sold for Residential Property Types

- Single Family: 14.67%
- Multi-Family: 17.67%
- Townhouse: 15.27%
- Mobile Home: 16.75%
- Condominium: 16.30%
- Farmhouse: 15.68%

Pie chart percentages:
- Single Family: 14.67%
- Multi-Family: 17.67%
- Townhouse: 15.27%
- Mobile Home: 16.75%
- Condominium: 16.30%
- Farmhouse: 15.68%
Key Ideas

- Use the ACCESSIBLECHECK system option to check for common accessibility violations.
- The ODS HTML5 destination together with the Daisy style is recommended for accessible output. See Using ODS Styles to Create Accessible Output on page 91.
- Supply a title for the output in the ODS HTML5 statement. This is required by WCAG 2.0 success criterion 2.4.2 Page Title.
- Use the H statement to add headings that screen readers can read to the report, and to each section, table, and graph. The headings should also be used to define an outline within the report. This is required by WCAG 2.0 success criteria 1.3.1 Info and Relationships and WCAG 2.0 success criteria 2.4.6 Headings and Labels.
- When adding headings, it is recommended that reports have one H1 heading, and all the other headings should be subordinate with no skipped levels. It is acceptable to have multiple headings that are H2, H3, and so on.
- Create a table of contents that consists of links to the sections, tables, and graphs in the report. Creating a table of contents at the top of your report with links helps keyboard and screen reader users. This satisfies the WCAG 2.0 success criterion 2.4.1.
- Use visual styles such as underlining, bold, or italics, in addition to color, to bring focus to data and text. This is required by WCAG 2.0 success criteria 1.4.1 Use of Color.
- Use the DESCRIPTION option to add descriptive text where appropriate. Generally, always use DESCRIPTION with tables, graphs, and images. However, when using regions and layouts for layout purposes only, DESCRIPTION might be redundant. This is required by WCAG 2.0 success criterion 1.3.1 Info and Relationships.
  
  For tables, you can use the CAPTION instead of DESCRIPTION.
- It is a best practice that charts, graphs, images, and tables have descriptions, and titles are optional. If a chart, graph, and so on, does not have a title, then it should have a heading just before it that serves the same purpose for sighted and low vision users.

See Also

- Info and Relationships: Understanding SC 1.3.1
- Headings and Labels: Understanding SC 2.4.6
- Use of Color: Understanding SC 1.4.1
- For information about language elements that are related to accessibility, see Appendix: ODS Syntax Related to Accessibility on page 105.
- For information about the ODSTEXT procedure, see SAS Output Delivery System: Procedures Guide.
Recommendations for Graphs

The General Recommendations for Creating Accessible Output on page 3 contain general best practices. For example, you should use the Daisy style option in the ODS HTML5 destination statement. In addition to the general recommendations, observe the following recommendations for graphs:

- Use the **SGPLOT procedure** to create single-cell graphs with a wide range of plot types that can be read and interpreted by the **SAS Graphics Accelerator**.

  For a list of plot types currently supported by **SAS Graphics Accelerator**, see the **SAS Graphics Accelerator product page** on the SAS Support site.

  Starting with SAS 9.4M6, you can use the **SGPIE procedure** to create pie and donut charts. This procedure is preproduction.

  You can also use the **SGPANEL procedure** to create classification panels, and the **SGSCATTER procedure** to create paneled scatter plots. However, these graphs are not currently supported by the **SAS Graphics Accelerator**. In addition, the accelerator does not support graphs that contain plot overlays.

- The **Graph Template Language (GTL)** is the underlying language for the default templates that are provided by SAS for procedures that use ODS Graphics. Use the GTL either to modify these templates or to create your own highly customized charts and plots. Use GTL to create a template, and then use the **SGRENDER procedure** to render the graph.

- The graph output must be an SVG file in order to be read and interpreted by the **SAS Graphics Accelerator**. Use the ODS HTML5 destination to generate SVG graphic output by default when you use the SAS windowing environment. If you are using SAS Studio, **configure SVG output** as a default preference.

  SVG output helps satisfy WCAG 2.0 success criterion 1.4.4 Resize Text.

  **Note:** If you are using batch mode, specify **OUTPUTFMT=SVG** in the ODS GRAPHICS statement.

  If you need to produce PDF output, see Overview of Accessible PDF Documents on page 5.

- Do either of the following:

  - If you are using a release prior to SAS 9.4M6, specify the ACCESSIBLE_GRAPH option in the ODS HTML5 destination statement. The option enables the graph to be read, interpreted, and sonified by the **SAS Graphics Accelerator**.

    **Note:** If you are using SAS Studio, you can configure the ACCESSIBLE_GRAPH option as a default preference.

  - Starting with SAS 9.4M6, specify the ACCESSIBLEGRAPH system option.

    The ACCESSIBLEGRAPH system option enables the ACCESSIBLE_GRAPH option in the ODS HTML5 destination by default. When the system option is specified, you do not need to specify the ACCESSIBLE_GRAPH option in individual ODS HTML5 statements. The system option affects all ODS HTML5 output until the option is disabled or the SAS session ends.

    This option helps satisfy the WCAG 2.0 success criterion 1.1.1 Non-text Content.

- Ensure that your viewing audience becomes aware of the **SAS Graphics Accelerator** and its capabilities. See Inform Users about the SAS Graphics Accelerator on page 76.

- Use these ODS GRAPHICS statement options to enhance the accessibility of your graphs:

  - If you are creating a graph that contains groups or multiple plots, you might need to specify **ATTRPRIORITY=NONE** to ensure that the data does not use color as the primary distinction between group values. The need to specify **ATTRPRIORITY=NONE** depends on which ODS style you are using.

    The following cases apply to the Daisy style:
If you are using a release prior to SAS 9.4M6, the Daisy style uses a color-priority attribute rotation pattern by default for grouped or multiple plots. In this case, you should specify ATTRPRIORITY=NONE in the ODS GRAPHICS statement.

Starting with SAS 9.4M6, ATTRPRIORITY=NONE is the default setting for the Daisy style. The Daisy style rotates attributes using colors and other elements such as markers, line patterns, and fill patterns. No additional action is required on your part.

This option helps satisfy the WCAG 2.0 success criterion 1.4.1 Use of Color.

For more information about attribute priority, see About Attribute Rotation Patterns for Grouped Plots on page 92. See also Example: Multi-Line Series Plot That Does Not Rely on Color Alone on page 78.

- Use the IMAGEMAP option to enable data tips and drill-down generation for the graph. 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).

Specify this option before the SGPLOT procedure. If using GTL, specify the option before the SGRENDER procedure. In the plot statement of the procedure, specify the data tips and drill-down options as appropriate.

This option helps satisfy the WCAG 2.0 success criterion 1.4.4 Resize Text.

See Example: Bar Chart with Data Tips and Drill-Down Links Created with GTL on page 84.

- You might want to specify the size of a graph.

Changing the size of the graph can make it more readable, which helps satisfy WCAG 2.0 success criterion 1.4.4 Resize Text.

- Specify a TITLE statement before the PROC SGPLOT statement to give the graph a title. The title should be clear and descriptive.

This option helps satisfy WCAG 2.0 success criterion 2.4.6 Headings and Labels.

Graph titles must reside within the graph in order to be read by the SAS Graphics Accelerator. Follow these recommendations:

- Do not specify the NOGTITLE option in the ODS HTML5 destination statement. The option specifies that the title is part of the HTML page, and not part of the graph.

- If you are using GTL to create the graph, instead of using the TITLE statement, use the ENTRYTITLE statement in the BEGINGRAPH block. See Example: Bar Chart with Data Tips and Drill-Down Links Created with GTL on page 84.

- Specify the DESCRIPTION= option, which is available in the PROC SGPLOT statement. This option describes the graph that you are creating.

This option helps satisfy WCAG 2.0 success criteria 1.3.1 Info and Relationships and 1.1.1 Non-text Content.

Starting with SAS 9.4M5, you can specify line fill patterns for supported plot types. Previously, fill patterns were available only when you used certain gray-scale styles. You can also change the appearance of the fill patterns.

- In the SGPLOT procedure, the FILLPATTERN and FILLPATTERNATTRS= options were added to the supported plot types. See Example: Grouped Bubble Plot That Shows Fill Patterns on page 80.

- In GTL, FILLPATTERN was added to the DISPLAY= option, and FILLPATTERNATTRS= was added to the supported plot types.

This option helps satisfy WCAG 2.0 success criterion 1.4.1 Use of Color.

- Specify axis labels to ensure that all viewers can access the axis labels. The labels should be clear and descriptive.

This option helps satisfy WCAG 2.0 success criteria 1.4.5 Images of Text and 4.1.2 Name, Role, and Value.
When multiple plot statements are used within a procedure, specify a legend label for each statement. This enables the SAS Graphics Accelerator to convey a user-friendly label for each layer of the graph. The labels should be clear and descriptive.

See Example: Multi-Line Series Plot That Does Not Rely on Color Alone.

This option helps satisfy WCAG 2.0 success criteria 1.4.5 Images of Text and 4.1.2 Name, Role, and Value.

Inform Users about the SAS Graphics Accelerator

When you create a graph that can be read and interpreted by the SAS Graphics Accelerator, ensure that users who view your graph are aware of the accelerator and its capabilities. To accomplish this, it is recommended that you add a brief statement at the bottom of each HTML page that is created. The purpose of this statement is to make users aware of the accelerator and to help them install the accelerator if it is not installed. The statement should be readable by viewers and screen readers. The statement should include a link to the accelerator’s product page at http://support.sas.com/software/products/graphics-accelerator/index.html. That page contains main features and capabilities, supported graph types, supported browser, and an installation link.

Here is an example statement:

Accessibility note: The graphs on this page can be presented in alternative formats using the SAS Graphics Accelerator.

Example: Simple Histogram

About This Example

This simple example shows the height distribution for a class of students. The example incorporates most of the recommendations for graphs. The numbered annotations explain the graph’s accessibility features.

Program

```sas
options accessiblegraph accessiblecheck; /* 1 */
ods html5 (id=web) style=daisy path="file-path" /* 2 */ file="file-name" {title="Histogram"};

title "Height Distribution"; /* 3 */
proc sgplot data=sashelp.class /* 4 */
    description= "Histogram showing distribution of students by height";
    histogram height;
    xaxis label="Height"; /* 5 */
    yaxis label="Percent";
run;
```
Starting with SAS 9.4M6, you can specify the ACCESSIBLEGRAPH system option. The option enables your graph to be read by the SAS Graphics Accelerator by default. The system option affects all ODS HTML5 output until the option is disabled or the SAS session ends.

If you are using a release prior to SAS 9.4M6, specify the ACCESSIBLE_GRAPH option in the ODS HTML5 statement.

The ACCESSIBLECHECK system option checks for common violations of accessibility standards and writes messages to the log if they are found.

Here are details about the ODS environment:

- The ODS HTML5 destination statement sets the ODS environment.
- The (ID=WEB) option is required for use only with SAS Studio.

Note: If you remove (ID=WEB) from the statement, be sure to also remove it from the ODS HTML5 CLOSE statement.

- The STYLE= option specifies the Daisy style. The Daisy style is recommended for accessible output.

Note: If you are using SAS Studio, you can configure the STYLE= option as a default preference.

- You must supply a valid file path and filename. Alternatively, you can omit the PATH= and FILE= options.
- The (TITLE=) suboption provides a title that appears in the browser window title bar. Specifying (TITLE=) is required by WCAG 2.0 success criterion 2.4.2 Page Title.

The TITLE statement before the PROC SGPLOT statement ensures that all users and screen readers can access the graph title.

The DESCRIPTION= option in the PROC SGPLOT statement provides a description of the output image that can be read by a screen reader.

The axis LABEL= options provide labels that can be read by a screen reader.
Output

Output 8.1  Simple Histogram Example

---

See Also


---

Example: Multi-Line Series Plot That Does Not Rely on Color Alone

About This Example

This example contains multiple series plots to show stock close, low, and high prices for IBM stock. Depending on the style that is used, by default the system identifies each line using color priority. With color priority, the system cycles through a list of colors while holding the other attributes (line pattern in this case) constant until the list of colors is exhausted. However, it’s better not to use color alone to distinguish the lines. The example uses both color and line patterns to distinguish the lines, which helps satisfy WCAG 2.0 success criterion 1.4.1 Use of Color.

Important: This example cannot be used with the SAS Graphics Accelerator. The accelerator does not support multilayered plots.

Note: This example uses the ODS environment setup that was shown in the simple histogram example. For descriptions, see that example.

Program

```sas
ods html5 (id=web) style=daisy path="file-path"
file="file-name"
{title="Series"};

/* The next line is required only for releases prior to SAS 9.4M6. */
```

ods graphics / attrpriority=none; /* 1 */

title "Stock Trend"; /* 1 */

proc sgplot data=sashelp.stocks
(where=(date >= "01jan2000"d and stock = "IBM"));
description="Three series plots show stock close, low, and high prices";
series x=date y=close / legendlabel="Close Price"; /* 2 */
series x=date y=low / legendlabel="Low Price";
series x=date y=high / legendlabel="High Price";
xaxis label="Date";
yaxis label="Price";
run;

title;

dohtml (id=web) close;

1 If you are using a release prior to SAS 9.4M6, specify the ATTRPRIORITY=NONE option in an ODS GRAPHICS statement. The ATTRPRIORITY=NONE option specifies that the graph does not use a color priority for the rotation pattern. The result is a graph that uses both color and line patterns to distinguish the lines.

Starting with SAS 9.4M6, ATTRPRIORITY=NONE is the default setting for the Daisy style. By default, the Daisy style rotates attributes using colors and line patterns.

For more information, see About Attribute Rotation Patterns for Grouped Plots.

2 The LEGENDLABEL= option conveys a more informative legend label for each series line. If you do not specify the legend label, by default the label of the Y variable is used.

**Output**

**Output 8.2  Multi-Line Series Plot**

![Graph showing stock trend](image)

**See Also**

Example: Grouped Bubble Plot That Shows Fill Patterns

About This Example
Starting with SAS 9.4M5, the SGPLOT procedure enables you to specify line fill patterns for supported plot types. Previously, fill patterns were available only when you used certain gray-scale styles.

This feature helps satisfy WCAG 2.0 success criterion 1.4.1 (Use of Color). Graphs that use this feature do not rely on color alone to distinguish categories of data.

Note: This example uses the ODS environment setup that was shown in the simple histogram example. For descriptions, see that example.

Program

```sas
options accessiblegraph accessiblecheck;
ods html5 (id=web) style=daisy path="file-path"
    file="file-name"
    (title="Bubble Plot");

proc format;  /* 1 */
  value $sex
    "F"="Female"
    "M"="Male";
run;

title "Class Height, Width, and Age";

proc sgplot data=sashelp.class
   description= "Bubble plot showing height, width, and age"
   format sex $sex.;
   bubble x=height y=weight size=age / /* 2 */
      group=sex fillpattern
      nofill;
   keylegend / title=""; /* 3 */
run;

title;
ods html5 (id=web) close;
```

1 The FORMAT procedure creates a user-specified format for the SEX variable. The default values for this variable are M and F. The reformatted values of MALE and FEMALE are more descriptive and intuitive. This change enables the SAS Graphics Accelerator to convey a user-friendly identifier for each variable.

2 The BUBBLE statement groups the bubbles and specifies FILLPATTERN and NOFILL. The FILLPATTERN option enables the display of fill patterns for the plot. The NOFILL option removes the fill color from the bubbles. These options make it easier to distinguish the bubbles.
The TITLE option in the KEYLEGEND statement sets the title to NULL. This in effect removes the title from the legend.

Output

Output 8.3  Grouped Bubbles with Fill Patterns

See Also


Example: Grouped Bar Chart That Shows Fill Patterns and Does Not Use Color

About This Example

There might be times when you want to use a style other than Daisy in order to provide the highest level of contrast. Journal2 is a gray-scale style that is useful for this purpose.

The Journal2 style has another feature that enhances accessibility. When used with bar charts, Journal2 renders grouped bars with fill patterns. This example illustrates that feature. The feature helps satisfy WCAG 2.0 success criterion 1.4.1 (Use of Color).

Note: This example uses the ODS environment setup that was shown in the simple histogram example. For descriptions, see that example.

Program

    options accessiblegraph accessiblecheck;

    ods html5 (id=web) style=journal2 path="file-path" /* 1 */
    file="file-name"
    {title="Bar Chart");
proc format;
    value $sex
        "F"="Female"
        "M"="Male";
run;

title "Mean Class Height";

proc sgplot data=sashelp.class
description= "Bar chart showing mean class height";
    format sex $sex.;
    vbar age / group=sex response=height stat=mean;
    xaxis label="Age";
    yaxis label="Mean Height";
run;

title;
ods html5 (id=web) close;

1 The STYLE= option specifies the Journal2 style.
   Note: If you are using SAS Studio, you can configure the STYLE= option as a default preference.

Output

Output 8.4  Grouped Bars with Fill Patterns

See Also

"Using Gray-Scale Styles to Distinguish Grouped Bar Charts" in SAS ODS Graphics: Procedures Guide
Example: Scatter Plot with High-Visibility Markers and Axis Elements

About This Example

This example enhances the visibility of a scatter plot by enlarging the markers, axis labels, and axis tick values.

Note: This example uses the ODS environment setup that was shown in the simple histogram example. For descriptions, see that example.

Program

```sas
options accessibilegraph accessibilecheck;

ods html5 (id=web) style=daisy path="file-path"
   file="file-name"
   title="Scatter Plot";

title "Class Height and Weight";

proc sgplot data=sashelp.class
description= "Scatter plot showing students by height and weight";
scatter x=height y=weight /
   markerattrs=(symbol=circlefilled size=12px); /* 1 */
   xaxis label="Height" labelattrs=(size=12pt) /* 2 */
      valueattrs=(size=12pt) grid;
   yaxis label="Weight" labelattrs=(size=12pt) valueattrs=(size=12pt) grid;
run;

title;

ods html5 (id=web) close;
```

1. The MARKERATTRS= option in the SCATTER statement specifies the marker symbol and size.
2. The LABELATTRS= and VALUEATTRS= options in the two axis statements specify the size of the axis labels and tick values, respectively.
Output

Output 8.5  Scatter Plot with Enlarged Axis Labels, Tick Values, and Markers

Key Idea

The example uses three *ATTRS= options to specify graphics attributes. Many ODS Graphics procedure statements have similar options and suboptions that control the appearance of different parts of a graph. For example, with box plots you can specify the appearance of the mean markers, median lines, and more. Various *ATTRS= options enable you to specify attributes for lines, bars, markers, text, and so on.

See Also

- “Using Plot Options to Control Graph Appearance” in SAS ODS Graphics: Procedures Guide

Example: Bar Chart with Data Tips and Drill-Down Links Created with GTL

This example uses GTL to create a simple grouped bar chart that displays data tips and has a drill-down link on each bar. Starting with SAS 9.4M5, data tips and drill-down links are supported in SVG output. Also starting in SAS 9.4M5, the ODS style element FOCUSLINK enables you to enhance the appearance of the link-focus indicator in ODS output. This example uses these features to create the drill-down bar chart in the SVG format.

Program

```sas
%let outpath=output-path;                   /* 1 */
%let baseurl=base-url;

data sales;                                 /* 2 */
  length url $80;
```
set sashelp.prdsale(
    where=(year eq 1994 and quarter eq 1 and
country eq "U.S.A.");
url="&baseurl/" || strip(division) || "_" ||
strip(region) || ".html";
format actual predicted dollar12.0;
run;

proc sort data=work.sales;                      /* 3 */
    by division product;
run;

proc template;                                  /* 4 */
define statgraph piechart;
    dynamic title;
    begingraph / designwidth=420 designheight=340;
        entrytitle textattrs=(size=14pt weight=bold) title;
        layout region;
            piechart category=product response=actual /
                datalabellocation=outside
                datalabelattrs=(size=12pt weight=bold);
        endlayout;
    endgraph;
end;
run;

%macro createpiechart(division, region);        /* 5 */
    ods html5 accessible_graph style=daisy path="&outpath"
    file="&division._&region..html"
    (title="Product Sales for &division &region");
    proc sgrender data=sales template=piechart
description="Pie chart of product sales for
&division &region.";
        dynamic title="Product Sales for &division &region";
        where division eq "&division" and region eq "&region";
run;
    ods html5 close;
%mend createpiechart;

ods _all_ close;                                /* 6 */
%createpiechart(EDUCATION, EAST);
%createpiechart(EDUCATION, WEST);
%createpiechart(CONSUMER, EAST);
%createpiechart(CONSUMER, WEST);

proc template;                                  /* 7 */
define statgraph basechart;
    begingraph / attrpriority=none designwidth=420
designheight=340;
        entrytitle textattrs=(size=14pt weight=bold)
            "Total U.S. Sales in First Quarter of 1994";
        entryfootnote "Click a bar for product sales data.";
        layout overlay;
            barchart category=division response=actual /
                name=sales* display=all group=region
                groupdisplay=cluster url=url
The OUTPATH macro variable stores the file system path for the output files. The BASEURL macro variable stores the base URL for the drill-down link targets.

The DATA step creates the Work.Sales data set. It extracts the USA sales data for the first quarter of 1994 from Sashelp.Prdsale. It adds a URL variable, and for each observation, populates it by concatenating the value of the BASEURL macro variable, the value of the Division variable, and the value of the Region variable. Finally, the DATA step applies the format DOLLAR12 to the Actual variables.

Sort the data in Work.Sales by Division and Product so that the chart categories are in ascending order.

The TEMPLATE procedure defines the PIECHART template, which generates a target pie chart for a specific division and region. The dynamic variable TITLE is used to pass the graph title into the template at run time. The ENTRYTITLE statement displays the title for the graph within the graph’s border. In the PIECHART statement, the DATALABELLOCATION=OUTSIDE option places the pie-slice labels outside of the pie chart to improve readability. Attribute overrides are used to increase font sizes to further improve readability.

The %CREATEPIECHART macro creates an HTML file that displays a pie chart for a specified division and region. These files are used as targets for the drill-down links. The ODS HTML5 destination and the Daisy ODS style is used to render the pie charts. The ACCESSIBLE_GRAPH option enables the SAS Graphics Accelerator. The division and region values are used to form the name of the HTML output file to match the names that were used in the URL variable values in the DATA step. The SGRENDER procedure renders the PIECHART template. The DYNAMIC statement passes the graph title to the PIECHART template in the dynamic variable TITLE. The DESCRIPTION= option provides a description for the output image that can be read by a screen reader. The HTML output is stored in the path specified by the OUTPATH macro variable.

The %CREATEPIECHART macro calls generate a pie chart for each division and region.

The TEMPLATE procedure defines the BASECHART template, which generates the bar chart. Option ATTRPRIORITY=NONE in the BEGINGRAPH statement disables color-priority attribute rotation for this graph only. In the BARCHART statement, Division is the category variable, Actual is the response variable, and Region is the group variable. Option DISPLAY=ALL displays all of the bar features: outlines, fill color, and fill pattern. The URL= option specifies the URL variable in the plot data, which is used to create the drill-down links in the bar chart. The FILLPATTERNATTRS= option sets the fill-pattern color to black to improve visibility. The HEIGHTSCALE= option in the DISCRETELEGEND statement increases the size of the legend fill swatches to improve visibility.

The ODS GRAPHICS statement enables image maps, which is required for data tips and drill-down links in HTML output.
The ODS statement opens the HTML5 destination. ODS Graphics output for the ODS HTML5 destination is SVG by default, which is required to use the FOCUSLINK style element. The STYLE= option specifies the Daisy ODS style, and options PATH= and FILE= specify the location and name of the output file. The ACCESSIBLE_GRAPH option enables the SAS Graphics Accelerator. The SGRENDER procedure renders the graph in template BASECHART. The DESCRIPTION= option provides a description for the output image that can be read by a screen reader.

Note: Starting with SAS 9.4M6, the Daisy ODS style includes the FOCUSLINK style element. Prior to SAS 9.4M6, you must create a custom style that adds the FOCUSLINK style element to the Daisy style or to any other ODS style that you want to use. See Creating Your Own Style That Includes the FocusLink Style Element on page 102.

Output

Output 8.6  Drill-Down Bar Chart Created with GTL

The data tip appears when the mouse pointer is positioned on a bar. If the bar links to a drill-down graph, the mouse pointer changes to indicate an active link. If the bar has focus, the link-focus outline appears around the bar. When a bar is clicked or when Enter is pressed while a bar has focus, the drill-down chart for that bar opens in the same browser window. For example, clicking the CONSUMER WEST bar or pressing Enter while the CONSUMER WEST bar has focus opens the following pie chart.
Key Ideas

- GTL is an extension to the Output Delivery System (ODS) that enables you to create complex, sophisticated graphs. All of the graphs that are created by the SAS analytical procedures and by the SAS Statistical Graphics Procedures are generated using GTL. Users who need to go beyond the graphs created by these SAS procedures can use GTL directly to design their graphs using the TEMPLATE procedure. A graph template is a program that specifies the layout and details of a graph. You can then apply the template to your data and render graphs using the SGRENDER procedure.

- Starting with SAS 9.4M6, ODS styles Daisy, HighContrast, and HighContrastLarge include an enhanced FOCUSLINK indicator. You can use these styles to enhance the appearance of the link-focus indicator in graphs that have active links.

- You can add an enhanced FOCUSLINK indicator to any existing ODS style that does not currently include it. You can specify the indicator outline color, width, and line pattern that achieve the desired appearance. See Enhancing the Appearance of the Link-Focus Indicator on page 102.

- To create a drill-down chart, you must add a column to the chart data that contains the link URLs. In the plot statement for the drill-down chart, specify the name of the URL variable in the URL= option.

- To enable data tips and drill-down links in graphs, you must specify the IMAGEMAP option in an ODS GRAPHICS statement to enable image maps.

- Starting with SAS 9.4M5, the Default ODS style and all of the styles that are derived from it support fill patterns. This includes the Daisy style and any custom style that is derived from Daisy. For a grouped plot, when displaying fill patterns with styles that use color-priority attribute rotation such as Daisy, disable color-priority attribute rotation. This ensures that a different fill pattern is used for each group value. To disable color-priority attribute rotation, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement or in the BEGINGRAPH statement of your graph template.

- Graph titles and footnotes must reside within the graph’s boundaries in order to be read by the SAS Graphics Accelerator. The ENTRYTITLE and ENTRYFOOTNOTE statements display a title and footnote for the graph within the graph’s boundaries.

  If you instead use TITLE and FOOTNOTE statements before the SGRENDER procedure, then the title and footnote reside outside the graph and cannot be read by the SAS Graphics Accelerator.
About the SAS Graphics Accelerator

SAS Graphics Accelerator (accelerator) enables users with visual impairments or blindness to create, explore, and share data visualizations. It supports alternative presentations of data visualizations that include enhanced visual rendering, text descriptions, tabular data, and interactive sonification. Sonification uses non-speech, musical audio to convey important information about a graph, such as the graph’s overall shape and specific contours.

The accelerator is a browser extension that is used in conjunction with the ACCESSIBLE_GRAPH option in the ODS HTML5 destination statement. The ACCESSIBLE_GRAPH option inserts metadata into the ODS HTML5 output for every graph that is generated while the option is in effect. The metadata does not affect the visual display of the graph. Rather, the metadata enables accessibility features for the graph. The accelerator parses the metadata and provides descriptive information about the graph.

This functionality occurs in a three-step process.

1. In your program, specify the ACCESSIBLE_GRAPH option in an ODS HTML5 destination statement. This statement must precede the code that creates the graph.

   **TIP** Rather than specify the ACCESSIBLE_GRAPH option in the ODS HTML5 statement, you can specify the ACCESSIBLEGRAPH system option. The system option affects all ODS HTML5 output until the option is disabled or the SAS session ends. See Accessibility System Options on page 7.

2. End users open the graph in a supported browser.
   
   If the accelerator has been installed on the browser, the accelerator scans the HTML page when the graph is opened or refreshed in the browser. The accelerator detects the metadata, sounds a chime, and displays an Accelerate button.

3. The user selects Accelerate. The accelerator opens a new browser tab and displays an alternative presentation of the graph.

For more information about the accelerator, including main features and capabilities, supported graph types, supported browser, and an installation link, see SAS Graphics Accelerator on the SAS Support site.
About ODS Styles

ODS styles specify the visual attributes of ODS output such as fonts, colors, markers, lines, and so on. They are used to provide a consistent look for ODS documents. Each style is designed around a specific visual theme. Several ODS styles are delivered with SAS. You can use the TEMPLATE procedure to list the available ODS styles. For more information, see SAS Output Delivery System: User’s Guide.

Of the available ODS styles, the following are suitable for generating accessible output when used with the ODS HTML5 destination.

- **Daisy** (recommended)
- **vaDark**
- **HighContrast**
- **Journal2**

These styles provide a high level of contrast in graphics output and tabular output in most cases.
About Attribute Rotation Patterns for Grouped Plots

Group-Value Attribute Specifications

In ODS styles, the visual attributes of graphics elements that represent group values in grouped plots are determined by the GraphData1–GraphDataN style elements. Each GraphDataN element specifies unique visual attributes for a single group value. For example: in many of the styles that use color, each GraphDataN element specifies one or more of the following visual attributes for a group value:

- a color to use for filled areas
- a fill pattern to use for filled areas

*Note:* In SAS 9.4M4 and in earlier releases, only bar charts and histograms support fill patterns. Starting with SAS 9.4M5, band plots, box plots, bubble plots, high-low charts, ellipse plots, and polygon plots also support fill patterns.

- a contrasting color to use for the value or label color, the plot line color, the fill-pattern color, the shape outline color (bars, boxes, bubbles, and so on), or the marker symbol color
- a marker symbol to use for the plot marker
- a line pattern to use for the plot line or the shape outline (bars, boxes, bubbles, and so on)

The number of GraphDataN elements and the attributes that they specify vary by style. One or more of the attributes that are specified in the GraphDataN elements are combined to create a number of unique visual representations for group values. The following table shows the number of group value representations that are provided by the Daisy, VaDark, HighContrast, and Journal2 styles.

<table>
<thead>
<tr>
<th>Group-Value Attributes</th>
<th>ODS Style</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Daisy</td>
</tr>
<tr>
<td>colors specified in style</td>
<td>12</td>
</tr>
<tr>
<td>contrasting colors specified in style</td>
<td>12</td>
</tr>
<tr>
<td>marker symbols specified in style</td>
<td>7</td>
</tr>
<tr>
<td>line patterns specified in style</td>
<td>11</td>
</tr>
<tr>
<td>fill patterns specified in style</td>
<td>none prior to SAS 9.4M5</td>
</tr>
<tr>
<td>maximum unique marker symbol and color combinations</td>
<td>84</td>
</tr>
</tbody>
</table>
### About Attribute Rotation Patterns for Grouped Plots

<table>
<thead>
<tr>
<th>Group-Value Attributes</th>
<th>ODS Style</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Daisy</td>
</tr>
<tr>
<td>maximum unique line pattern and color combinations*</td>
<td>132</td>
</tr>
<tr>
<td>maximum unique fill colors ***</td>
<td>36</td>
</tr>
<tr>
<td>maximum unique fill color and fill pattern combinations †</td>
<td>396</td>
</tr>
<tr>
<td>attribute rotation pattern††</td>
<td>color-priority prior to SAS 9.4M6 default starting with SAS 9.4M6</td>
</tr>
</tbody>
</table>

* Plot markers in the Daisy, vaDark, and HighContrast styles are a combination of marker symbol and contrasting color. Plot markers in the Journal2 style are always black.

** Plot lines in the Daisy, vaDark, and HighContrast styles are a combination of line pattern and contrasting color. Plot lines in the Journal2 style are always black.

*** Fill colors in the Daisy, vaDark, and HighContrast style are three iterations of the specified colors: the colors as they are specified in the style, the colors in a lighter shade, and the colors in a darker shade.

† The Daisy, vaDark, and HighContrast styles do not support fill patterns in SAS 9.4M4 and in earlier releases.

†† See The Default Attribute Rotation Pattern on page 93 and The Color-Priority Attribute Rotation Pattern on page 93.

### The Default Attribute Rotation Pattern

In a grouped plot, each group value is assigned attributes from a GraphData1–GraphDataN style element sequentially (1 to N). This pattern generates a number of unique group value representations that vary by one or more visual attributes. Most of the ODS styles use this pattern by default. For more information, see “The Default Attribute Rotation Pattern” in [SAS Graph Template Language: User’s Guide](https://support.sas.com/documentation/cdl/en/gtc/65250/HTML/default/viewer.htm#a000051531.htm).

### The Color-Priority Attribute Rotation Pattern

Prior to SAS 9.4M6, the Daisy and HTMLBlue ODS styles use a color-priority attribute rotation pattern by default for grouped plots. Starting with SAS 9.4M6, the Daisy style uses the default attribute rotation pattern while the HTMLBlue style uses the color-priority attribute rotation pattern. The color-priority attribute rotation pattern overrides the normal GraphData1–GraphDataN marker symbol, line pattern, and fill pattern rotation in an ODS style. When enabled, it cycles through all of the available colors while holding the first marker symbol, line pattern, or fill pattern constant. When all of the colors are exhausted, the colors are repeated using the next available marker symbol, line pattern, or fill pattern. For more information, see “The Color-Priority Attribute Rotation Pattern” in [SAS Graph Template Language: User’s Guide](https://support.sas.com/documentation/cdl/en/gtc/65250/HTML/default/viewer.htm#a000051531.htm).

The color-priority attribute rotation pattern results in plot markers, plot lines, and fill patterns that use color as the primary distinction between group values. Using only color to distinguish group values is not sufficient for accessibility. The following ODS GRAPHICS statement disables the color-priority attribute rotation pattern for styles that use the color-priority attribute rotation pattern.

```sas
ods graphics / attrpriority=none;
```

When the color-priority attribute rotation pattern is disabled, all ODS styles honor the GraphData1–GraphDataN specifications. For more information about the ODS GRAPHICS statement ATTRPRIORITY= option, see “[ODS GRAPHICS Statement](https://support.sas.com/documentation/cdl/en/gtc/65250/HTML/default/viewer.htm#con_sh+F7C058W75H60F862HJ29E2091.htm)” in [SAS Graph Template Language: Reference](https://support.sas.com/documentation/cdl/en/gtc/65250/HTML/default/viewer.htm).
Recommendation: Use Daisy with Color-Priority Attribute Rotation Disabled

Use the Daisy ODS style to generate accessible graphics output. Starting with SAS 9.4M6, the Daisy style uses the default attribute rotation pattern by default. Prior to SAS 9.4M6, the Daisy style uses the color-priority attribute rotation pattern by default. If you are using a release prior to SAS 9.4M6, use the following ODS GRAPHICS statement to disable the color-priority attribute rotation pattern before you generate your output:

```{r}
ods graphics / attrpriority=none;
```

With the default attribute rotation pattern, the accessibility features of the Daisy style include:

- the use of bold colors against a white background to provide a high level of contrast.
- the use of line patterns and marker symbols in addition to color to distinguish group values. Starting with SAS 9.4M5, fill patterns are also used in addition to color to distinguish group values.

The Daisy style should be sufficient for most cases. The following figures show a sample bar chart, series plot, and table in the Daisy style when the default attribute rotation pattern is used and fill patterns are displayed.

*Figure 9.1  Sample Bar Chart in the Daisy Style*
Figure 9.2  Sample Series Plot in the Daisy Style

![Stock Performance in 1H2001](image)

Figure 9.3  Sample Table Output in the Daisy Style

<table>
<thead>
<tr>
<th>Month</th>
<th>IBM</th>
<th>Intel</th>
<th>Microsoft</th>
<th>Portfolio</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>$112.00</td>
<td>$37.00</td>
<td>$61.06</td>
<td>$210.06</td>
</tr>
<tr>
<td>February</td>
<td>$99.90</td>
<td>$28.56</td>
<td>$59.00</td>
<td>$187.46</td>
</tr>
<tr>
<td>March</td>
<td>$96.18</td>
<td>$26.31</td>
<td>$54.69</td>
<td>$177.18</td>
</tr>
<tr>
<td>April</td>
<td>$115.14</td>
<td>$30.91</td>
<td>$67.75</td>
<td>$213.80</td>
</tr>
<tr>
<td>May</td>
<td>$111.80</td>
<td>$27.01</td>
<td>$69.18</td>
<td>$207.99</td>
</tr>
<tr>
<td>June</td>
<td>$113.50</td>
<td>$29.25</td>
<td>$73.00</td>
<td>$215.75</td>
</tr>
</tbody>
</table>

For grouped plots, the Daisy style should provide a sufficient number of unique group-value representations. However, because the marker and line representations include color, grouped plots with a large number of group values might be problematic for users with color vision deficiencies. In the default attribute rotation pattern, the Daisy style cycles through the seven marker symbols using the first seven contrasting colors. A unique marker symbol and contrasting color are used for the first seven group values. Beyond seven group values, the marker symbols begin repeating with the next contrasting color for up to 36 group values. In that case, only color is used to distinguish some of the plot markers. A similar situation exists for lines and fill patterns, where a unique pattern and contrasting color are used for the first 11 group values. The Journal2 style provides a higher number of unique marker symbols, line patterns, and fill patterns that do not rely on color. For users with color vision deficiencies, the Journal2 style might be a better choice for grouped plots that have a large number of group values. See Journal2 on page 99.
Other ODS Styles That You Can Use

**VaDark**

Accessibility features of ODS style vaDark include:

- the use of bold colors against a black background to provide a slightly higher level of contrast than Daisy.
- the use of line pattern and marker symbols in addition to color to distinguish group values. Starting with SAS 9.4M5, fill patterns are also used in addition to color to distinguish group values.
- the use of white text against a black background for tables.

The following figures show a sample bar chart, series plot, and table in the vaDark style.

*Figure 9.4  Sample Bar Chart in the VaDark Style*
HighContrast

Accessibility features of ODS style HighContrast include:

- the use of bright colors against a black background to provides a slightly higher level of color contrast than Daisy.
- the use of line pattern and marker symbols in addition to color to distinguish group values. Starting with SAS 9.4M5, fill patterns are also used in addition to color to distinguish group values.
- the use of white text against a black background and larger fonts for tables.

The following figures show a sample bar chart, series plot, and table in the HighContrast style.
Figure 9.7  Sample Bar Chart in the HighContrast Style

Figure 9.8  Sample Series Plot in the HighContrast Style
Figure 9.9  Sample Table Output in the High Contrast Style

<table>
<thead>
<tr>
<th>Month</th>
<th>IBM</th>
<th>Intel</th>
<th>Microsoft</th>
<th>Portfolio</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>$112.00</td>
<td>$37.00</td>
<td>$61.06</td>
<td>$210.06</td>
</tr>
<tr>
<td>February</td>
<td>$99.90</td>
<td>$28.56</td>
<td>$59.00</td>
<td>$187.46</td>
</tr>
<tr>
<td>March</td>
<td>$96.18</td>
<td>$26.31</td>
<td>$54.69</td>
<td>$177.18</td>
</tr>
<tr>
<td>April</td>
<td>$115.14</td>
<td>$30.91</td>
<td>$67.75</td>
<td>$213.80</td>
</tr>
<tr>
<td>May</td>
<td>$111.80</td>
<td>$27.01</td>
<td>$69.18</td>
<td>$207.99</td>
</tr>
<tr>
<td>June</td>
<td>$113.50</td>
<td>$29.25</td>
<td>$73.00</td>
<td>$215.75</td>
</tr>
</tbody>
</table>

Journal2

Accessibility features of ODS style Journal2 include:
- the highest level of contrast compared to the other ODS styles
- the exclusion of color
- the use of fill shading, fill pattern, line pattern, and marker symbols to distinguish unique group values

The following figures show a sample bar chart, series plot, and table in the Journal2 style.

Figure 9.10  Sample Bar Chart in the Journal2 Style
Cases in Which ODS Style Alone Might Not Be Sufficient

There are some cases in which ODS styles alone might not provide sufficient contrast between some graphics elements. These cases can include the following:

- markers and lines that appear in color-filled or pattern-filled areas
- data values or labels that appear in color-filled or pattern-filled areas
- connecting lines that are used in grouped plots

Other cases that result in similar issues might also exist. The following figure demonstrates one of these cases.
This grouped box plot was generated using the Daisy style with color-priority attribute rotation disabled. Connecting lines join the mean in each box. Notice that only color is used to distinguish the connecting lines. Notice also that a marker symbol inside each box denotes the mean. However, because of the low contrast between the marker symbol color and the box fill color, the markers are difficult to see. You can disable the box fill color to resolve these issues as shown in the following figure.
When fill color is disabled, line pattern and line color are used to distinguish the boxes and their connecting lines. Starting with SAS 9.4M5, fill patterns can also be used to distinguish boxes. In this example, the mean marker within each box is more visible against the white background of the unfilled boxes. In similar situations with other plots, you might be able to use plot options to resolve the issues.

In some cases, you might want to increase the size of graphics elements such as marker symbols, data values, or labels in order to improve visibility. The marker symbols on the plot lines in Sample Series Plot in the Daisy Style on page 95, for example, are fairly small. Options are available in most plot statements that enable you to modify text and marker sizes, as well as the thickness of plot lines and shape outlines. For information about the options that are available in each plot statement, see SAS ODS Graphics: Procedures Guide.

Enhancing the Appearance of the Link-Focus Indicator

The FocusLink ODS Style Element

Starting with SAS 9.4M5, ODS supports 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 output in order to help satisfy WCAG 2.0 success criterion 2.4.7 Focus Visible. The FocusLink style element enables you to control the link-focus indicator outline color, line style, and line width. In SAS 9.5M5, the ODS styles that are delivered with SAS do not include the FocusLink style element. Starting with SAS 9.4M6, ODS styles Daisy, HighContrast, and HighContrastLarge include the FocusLink style element. If the ODS style that you want to use does not include the FocusLink style element, you can add it to that style as described in the next section.

Creating Your Own Style That Includes the FocusLink Style Element

The following SAS program uses the TEMPLATE procedure to create a new ODS style that is derived from the Journal2 style and includes the FocusLink style element.

```
proc template;                             /* 1 */
define style styles.journal2focuslink;  /* 2 */
parent = styles.journal2;            /* 3 */
class focuslink /                    /* 4 */
   outlinewidth = 5px
   outlinestyle = solid
   outlinecolor = black;
end;
run;
```

1 The TEMPLATE procedure compiles a style template and stores it in your private style template item store, Sasuser.Templat. For information about the TEMPLATE procedure and ODS style templates, see SAS Output Delivery System: User’s Guide.

2 The DEFINE STYLE STYLES.JOURNAL2FOCUSLINK statement defines ODS style Journal2FocusLink under Styles in your Sasuser.Templat item store. Choose a style name that is unique and reflects the purpose of the style.

3 The PARENT statement specifies the parent ODS style, Journal2, from which style Journal2FocusLink is derived. You can specify any one of the existing SAS styles as the parent.

4 The CLASS FOCUSLINK statement defines the FocusLink style element. The following style attributes are defined for style element FocusLink:
OUTLINEWIDTH specifies the line width of the outline as `number<unit>`, where `unit` is the unit of measurement. See "Units of Measurement" in SAS ODS Graphics: Procedures Guide.

OUTLINESTYLE specifies the outline line style as one of the following:
- DASHED
- DOTTED
- DOUBLE
- GROOVE
- HIDDEN
- INSET
- OUTSET
- RIDGE
- SOLID

OUTLINECOLOR specifies the outline color. The color specification must be a valid SAS color-naming-scheme color name or color code. See "Color-Naming Schemes" in SAS Graph Template Language: Reference.

All of the other style elements and attributes are inherited from the parent style, Journal2.

You must run the TEMPLATE procedure code to compile your new ODS style before you can use it.

**Using Your Focus Link ODS Style**

To use your focus link style, specify the name of the style in the `STYLE=` option in your ODS destination statement as shown in the following example.

```plaintext
ods html5 accessible_graph path="output-path" file="filename"
   style=styles.journal2focuslink;
```

Note: You cannot use your custom ODS style with the default destinations in SAS Studio. To use your custom style in SAS Studio, you must open your own ODS destination. The RESULTS tab in SAS Studio displays the output from the default ODS HTML5 destination, which uses the HTMLBlue ODS style by default. To see the output for your custom style, open the output from your ODS HTML5 destination.

In your ODS output, a solid, black, 5-pixel-wide outline appears around any link that has focus. The following figure shows an example of the modified focus-link indicator in table output.

In ODS Graphics output, the modified link-focus indicator appears on linked graphical elements only when the graph is generated in the SVG format and image maps are enabled. Graphs generated in an image format such as PNG do not honor the FocusLink style element. For an example, see Example: Bar Chart with Data Tips and Drill-Down Links Created with GTL on page 84.
Appendix: ODS Syntax Related to Accessibility

**ODS HTML5 Statement Options Related to Accessibility**

**ACCESSIBLE_GRAPH**
adds accessibility metadata to graphs that are created by ODS Graphics. The option enables the graph to be read, interpreted, and sonified by the SAS Graphics Accelerator.

Here is an example of the HTML5 statement with the ACCESSIBLE_GRAPH option.

```sas
ods html5 accessible_graph;
```

**Note:** If you are using SAS Studio, specify (ID=WEB) in the ODS statement. Here is an example of the HTML5 statement with the ID option and the ACCESSIBLE_GRAPH option.

```sas
ods html5 (id=web) accessible_graph;
```

Starting with SAS 9.4M6, you can specify the ACCESSIBLEGRAPH system option. The system option enables the ACCESSIBLE_GRAPH option in the ODS HTML5 destination by default. The system option affects all ODS HTML5 output until the option is disabled or the SAS session ends.

**STYLE**
specifies the style template to use in writing the output files.

Here is an example of the HTML5 statement with the STYLE option.

```sas
ods html5 style=daisy;
```

**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.

Here is an example of the HTML5 statement with the TITLE option and the ACCESSIBLE_GRAPH option.

```sas
ods html5 (id=web) style=daisy file="your-file-path/housing.html"
```
ODS PDF Statement Options Related to Accessibility

ACCESSIBLE | NOACCESSIBLE
specifies whether to add 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.

ACCESSIBLE_IDENTIFIER | NOACCESSIBLE_IDENTIFIER
specifies whether to add an identifier to the metadata of the PDF file confirming that the PDF produced by SAS meets the PDF Matterhorn Protocol.

For information about the options, see “ODS PDF Statement” in SAS Output Delivery System: User’s Guide.

ODS GRAPHICS Statement Options Related to Accessibility

ATTRPRIORITY
specifies a priority for cycling of the group attributes.

OUTPUTFMT
specifies the output format used to generate image or vector graphic files.

IMAGEMAP
controls data tips and drill down generation. 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).

Here is an example of the ODS GRAPHICS statement with the ATTRPRIORITY, OUTPUTFMT, and IMAGEMAP options.

```ods graphics / attrpriority=none outputfmt=svg imagemap=on;```

For information about the options, see “ODS GRAPHICS Statement” in SAS Graph Template Language: Reference.

ODS Methods in the Report Writing Interface Related to Accessibility

IMAGE Method
inserts an image into all open output destinations.

LAYOUT_ABSOLUTE Method
creates an absolute layout container.

Note: The LAYOUT_ABSOLUTE method is supported only for ODS PDF.
LAYOUT_GRIDDED Method
creates a gridded layout container.

REGION Method, Absolute
creates a region container for absolute layouts.

REGION Method, Gridded
creates a region container for gridded layouts.

TABLE_START Method
specifies the start of a table. See also Create an Accessible Table on page 53.

For a usage example, see Create an Accessible Report That Includes Text, Tables, and Graphs on page 63.

For more information about the methods, see “Reference and Examples” in SAS Output Delivery System: Advanced Topics.

ODS Procedures Related to Accessibility

ODSTEXT Procedure
creates text block templates.

See also this example on page 57.

TEMPLATE Procedure
enables you to customize the appearance of your SAS output.

See also this example on page 84.

For more information about the procedures, see SAS Output Delivery System: Procedures Guide.

ODS Style Attributes Related to Accessibility

BACKGROUNDIMAGE
specifies an image in a table, table cell, or graph to use as the background.

FONTSTYLE
specifies the style of the font for tables, table cells, and graphs. You can use the style as an alternative to using color.

FONTWEIGHT
specifies the weight of the font for tables, table cells, and graphs. You can use the weight as an alternative to using color.

POSTIMAGE
specifies an image to place after the table or table cell.

PREIMAGE
specifies an image to place before the table or table cell.

TEXTDECORATION
changes the visual presentation of the text. You can change the text as an alternative to using color.

OUTLINEWIDTH, OUTLINESTYLE, OUTLINECOLOR
specify the outline width, line style, and color of the link-focus indicator in ODS output. These attributes apply to the ODS style element FOCUSLINK. FOCUSLINK applies to SAS 9.4M5 and later releases.

For information about the attributes, see “Style Attributes Detailed Information” in SAS Output Delivery System: Advanced Topics.
Appendix: Input Data for Examples

Appendix: Input Data for Report Writing Interface and PROC TABULATE Examples

Create the Regfmt, Divfmt, and UseType Formats

```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;
```

Create the Energy Data Set

```sas
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
run;
```
Create the StateP Format

```sas
proc format;
value stateP
   1="Alaska" 2="Washington" 3="Oregon" 4="California" 5="Hawaii";
value propG
   1="Residential" 2="Commercial";
value resT
   1="Condominiums" 2="Modular Homes" 3="Single-Family"
   4="Townhouses" 5="Mobile Homes" 6="Farm Houses";
value comT
   1="Office" 2="Land" 3="Industrial"
   4="Retail" 5="Leisure" 6="Multi-Family";
run;
```
Create the PacificSum and AcmePacific Data Sets

data acmePacific;
length year 8 region division state propGroup propType $30. soldUnits medianListPrice 8;
format soldUnits comma12. medianListPrice dollar12.;
do year=2010 to 2013;
    regionN = 4;
    region = "West";
    divisionN = 9;
    division = "Pacific";
    stateN = 1 to 5;
    state = put(stateN,stateP.);
    do propGroupN =1 to 2;
        propGroup = put(propGroupN,propG.);
        do propTypeN = 1 to 6;
            if propGroupN = 1 then propType = put(propTypeN,resT.);
            else propType = put(propTypeN,comT.);
            soldUnits = 45000*ranuni(10000)+10000;
            medianListPrice = 200000*ranuni(199000)+199000;
            output;
        end;
    end;
end;
run;

proc sql;
create table pacificSum as
select state,
    year,
    propGroup,
    sum(soldUnits) as sUnits format=comma12.,
    avg(medianListPrice) as mPrice format=dollar12.
from acmePacific
where year in (2012, 2013)
group by state, year, propGroup;
quit;

Create the AMRPop Data Set and Population Format

data AMRPop; set sashelp.demographics (where=(region='AMR'));
    fmtpop = pop;
run;

proc sort;
    by pop;
run;

proc format;
    value population
        low - <1000000 = Small
        1000000 - <10000000 = Medium
        10000000 - <100000000 = Large
100000000 - <1000000000 = Huge ;
run;