Creating Accessible SAS® 9.4 Output Using ODS and ODS Graphics

SAS® Documentation
August 4, 2021

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2017. *Creating Accessible SAS® 9.4 Output Using ODS and ODS Graphics*. Cary, NC: SAS Institute Inc.

Creating Accessible SAS® 9.4 Output Using ODS and ODS Graphics

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

August 2021

9.4-P6:odsacoutput

Contents

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. This option applies to tables generated by the PRINT, REPORT, and TABULATE procedures as well as the Report Writing Interface. For the TABULATE procedure, the option changes the layout of some tables to make them accessible.
- 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 7.
- 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 112.

  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 "Bar Chart with Data Tips and Drill-Down Links Created with GTL" on page 96.

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 "Bar Chart with Data Tips and Drill-Down Links Created with GTL" on page 96.
  - 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 "Grouped Bubble Plot That Shows Fill Patterns" on page 90.
- 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 123 and "Bar Chart with Data Tips and Drill-Down Links Created with GTL" on page 96.

1

Overview and General Recommendations

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. 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
- axe browser extension
- WAVE Chrome extension

About the Examples in This Document

Most of the examples process sample data contained in a SAS data set that is stored in the SASHELP library. Each example specifies the data set name. In SAS Studio, you can access the SASHELP library in the Libraries section of the navigation pane.

Most of the examples in this document use a customized output environment in SAS Studio. The examples use the ODS destination statement to specify the style, output path, and other features related to accessibility. For information about customizing your output in SAS Studio, see "Customized Output Environment " in *SAS Output Delivery System: User's Guide*.

The examples are created using the Daisy ODS style unless otherwise indicated.

Because of size limitations, the graphs in this document are typically not shown in their default size of 640 pixels by 480 pixels. They are scaled down to meet the size requirements of our documentation production system. When graphs are reduced in size, the smaller graphs might have scaled down font sizes. Also, their numeric axes might display the tick values differently. Thus, the graphs that you generate from the example programs might not always look identical to the graphs that are shown in the figures. However, both graphs accurately represent the data.

Related Reading

- Summers, Ed, and Sean Mealin. "Section 508 and Maps: Breaking Down Barriers for People with Visual Impairments or Blindness." *Proceedings of the SAS Global Forum 2020 Conference.* Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2020/4542-2020.pdf.
- Kraus, Greg. 2020. "Creating Reports that Comply with Section 508 Using SAS® 9.4." *Proceedings of the SAS Global Forum 2020 Conference.* Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2020/4543-2020.pdf.
- Langston, Julianna, and Elizabeth Langston. "Look Ma, No Hands: How to Train Your Dragon for SAS®." *Proceedings of the SAS Global Forum 2020 Conference.* Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2020/4517-2020.pdf.
- Clarke, Terry. 2019. "Hey, I Know What, Let's Include Everybody: Accessibility." *Proceedings of the SAS Global Forum 2019 Conference.* Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2019/3196-2019.pdf.
- Kraus, Greg. 2019. "The Incredible, Accessible Report: Creating PDF Reports that Meet Compliance Standards in SAS® 9.4M6." *Proceedings of the SAS Global Forum 2019 Conference.* Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2019/3432-2019.pdf.

- Summers, Ed, Brice Smith, and Sean Mealin. 2019. "World Wide Wrangler: How to Create SAS® Data Sets from HTML Files in Seconds." *Proceedings of the SAS Global Forum 2019 Conference.* Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2019/3183-2019.pdf.
- Summers, Ed, Langston, Julianna, and Dan Heath. 2018. "Accessibility and ODS Graphics: Seven Simple Steps to Section 508 Compliance Using SAS® 9.4M5." Proceedings of the SAS Global Forum 2018 Conference. Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/1980-2018.pdf.
- Kraus, Greg. 2018. "ODS PDF Accessibility in SAS® 9.4M5: Going Beyond the Basics to Create Advanced Accessible Reports." Proceedings of the SAS Global Forum 2018 Conference. Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/2124-2018.pdf.
- Middleton, Woody. 2018. "PDF Accessibility: How SAS® 9.4M5 Enables Automatic Production of Accessible PDF Files." *Proceedings of the SAS Global Forum 2018 Conference*. Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/2129-2018.pdf.
- Sookne, Jesse, Barker, Kristin, Sumpter, Joe and Lavanya Mandavilli. 2017. "Accessibility and SAS® Visual Analytics Viewers: Which Report Viewer is Best for Your Users' Needs?" Proceedings of the SAS Global Forum 2017 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings17/SAS0655-2017.pdf.
- Walker, Glen R. 2017. "Tag, You're It! Creating Accessible (Tagged) PDF with SAS 9.4 Output Delivery System" Proceedings of the SAS Global Forum 2017 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings17/SAS0483-2017.pdf.
- Walker, Glen R. 2017. "Raising the Bar: The Pending Section 508 Refresh and the ODS HTML5 Destination." Proceedings of the SAS Global Forum 2017 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings17/SAS0641-2017.pdf.
- O'Connor, Daniel. 2017. "A Guru's Guide: Producing Section 508-Compliant Custom Reports with the Output Delivery System." Proceedings of the SAS Global Forum 2017 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings17/SAS0557-2017.pdf.
- Walker, Glen R. 2016. "A Guide to Section 508 Compliance Using SAS 9.4 Output Delivery System (ODS)." Proceedings of the SAS Global Forum 2016 Conference. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings16/SAS5280-2016.pdf.

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.
  - 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.
  - 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.
  - Use the Report Writing Interface (RWI) to create customized reports in an object-oriented language that is fully integrated with ODS.
  - Use the ODSTEXT procedure to generate headings that organize your content as well as any lists or static text, if applicable.
  - 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.

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

    For a list of plot types currently supported by the SAS Graphics Accelerator, see *SAS Graphics Accelerator: Supported Types of Data Visualizations*. You can also find that information on the SAS Graphics Accelerator product page on the SAS Support site.

    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.

    If you are creating graphs, after reviewing the recommendations here, see “Recommendations for Graphs” on page 83.
  - 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: In SAS Studio, you can configure the style as a preference for the default output environment.
  - 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 7.
- 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 123 and "Bar Chart with Data Tips and Drill-Down Links Created with GTL" on page 96.

  This option helps satisfy the WCAG 2.0 success criterion 2.4.7 Focus Visible.

See Also

- "Recommendations for Graphs" on page 83
- "General Recommendations for Tables" on page 18

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 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 accessible PDF documents in SAS 9.4M6 and later releases:

- The method that is 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 has been checked for accessibility errors and that it meets the PDF Matterhorn Protocol. It is recommended that you add this identifier after you have manually checked the generated output for accessibility errors. (The accessibility software options cannot automatically check that you have met all of the accessibility requirements.)
- 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, such as those 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*.

SAS Studio Accessibility Options

You can use SAS Studio or the SAS Windowing Environment to code and run your programs.

If you use SAS Studio, the Preferences window in SAS Studio enables you to specify default values and preferences for several features. For some results formats, such as HTML or PDF output, you can specify preferences that affect accessibility. These preferences apply to the SAS Studio default output environment.

Note: If you want to customize the SAS Studio output environment, you can disable the default output environment and establish your own output environment.

Most of the examples in this document use a customized output environment. For information about customizing your output in SAS Studio, see "Customized Output Environment " in *SAS Output Delivery System: User's Guide*.

The SAS Studio interface enables you to specify the following preferences that are related to accessibility. These preferences apply to the SAS Studio default output environment. They do not apply to the customized output environment that is used for many of the examples in this document:

1. In SAS Studio, select **Options** ⇨ **Preferences**.
2. In the navigation pane, select **SAS Programs** ⇨ **Results**. Do the following as appropriate:
   - Select **Specify an HTML output style** and select a new style from the list box. Daisy is recommended for most accessible output.
   - Select **Generate HTML graphs as SVG**. This option creates SVG graphs instead of PNG graphs in HTML output.

- Select **Enable accessible graph option**. Accessible graph output can be read and interpreted by the SAS Graphics Accelerator.
- Select **Produce PDF output** if you want to generate PDF documents. You can then do the following:
  - select a style
  - generate a table of contents
  - enable the accessible PDF option

3 Click **Save**.

2

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. This option applies to tables generated by the PRINT, REPORT, and TABULATE procedures as well as the Report Writing Interface. For the TABULATE procedure, the option changes the layout of some tables to make them accessible.

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*.

Note: 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*

Example `options accessiblegraph;`

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

makes the CONTENTS= value a visible table caption on the page. Without the ACCESSIBLETABLE option, table captions are 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.

When the CAPTION= option is also used, the ACCESSIBLETABLE option makes the CAPTION= value a visible table caption and the CONTENTS= value a table description. Both can be specified to provide a different caption and description. If the CAPTION= option is specified without a CONTENTS= option, the default CONTENTS value is used to describe the table.

The ACCESSIBLETABLE option also alters the layout of some tables that are generated with the TABULATE procedure in order to make the tables accessible. For example, when used with the TABULATE procedure, the ACCESSIBLETABLE option creates additional column headers instead of using row spans for column headers.

NOACCESSIBLETABLE

does not change procedure tabular output.

Default NOACCESSIBLETABLE

See "ACCESSIBLETABLE System Option" in *SAS System Options: Reference*

Example

```
options accessibletable;
```

Here are some recommendations for using the ACCESSIBLETABLE option, the CONTENTS= option, and the CAPTION= option:

- If you do not want a visible caption for the table, specify the CONTENTS= option but not the ACCESSIBLETABLE option.
- If you want a visible caption and a matching table description, specify the CONTENTS= option and the ACCESSIBLETABLE option.
- If you want a visible caption and a different table description, specify the CONTENTS= option, the CAPTION= option, and the ACCESSIBLETABLE option.

See Also

"Overview of the Accessibility System Options" on page 11

3

Creating Accessible Tables

Overview of Table Accessibility

About Accessible Tables

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.

Summary of the Table Procedures

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*.

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 "REPORT Procedure" in *Base SAS Procedures Guide*.

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. For more information, see "TABULATE Procedure" in *Base SAS Procedures Guide*.

About the Report Writing Interface

The Report Writing Interface enables you to create highly customized reports with the DATA step. To accomplish this, the Report Writing Interface uses an object-oriented language that is fully integrated with ODS. For more information, see "Introduction to the Report Writing Interface" in *SAS Output Delivery System: Advanced Topics*.

About the Table Examples

Some of the table examples and text come from the following paper by Greg Kraus, SAS Institute, Inc.: Paper 3432-2019 The Incredible, Accessible Report: Creating PDF Reports that Meet Compliance Standards in SAS 9.4M6.

References to other papers are cited where they occur in this document.

Most of the examples in this document use a customized output environment in SAS Studio. The examples use the ODS destination statement to specify the style, output path, and other features related to accessibility. For information about customizing your output in SAS Studio, see "Customized Output Environment " in *SAS Output Delivery System: User's Guide*.

For more information about all of the examples in the document, see "About the Examples in This Document" on page 3.

Recommendations for Tables

General Recommendations for Tables

The "General Recommendations for Creating Accessible Output " contain best practices for all output. In addition to the general recommendations, observe the following recommendations for tabular output. These recommendations apply to the PRINT, REPORT, and TABULATE procedures and the Report Writing Interface, except where noted otherwise.

- Starting with SAS 9.4M6, use the ACCESSIBLETABLE and ACCESSIBLECHECK system options. The system options facilitate the creation of accessible content on a system-wide level.
  - The ACCESSIBLETABLE system option adds visual captions to tables. The option also changes the layout of some tables that are generated with the TABULATE procedure to make them accessible. For the TABULATE procedure, always use the ACCESSIBLETABLE system option when the procedure contains a variable in the row dimension.
  - The ACCESSIBLECHECK system option detects accessibility issues and writes messages to the SAS log if accessibility issues are found.
- 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.

  Example for the PRINT procedure: "Table with a Descriptive Summation Row" on page 35.

  For the REPORT and TABULATE procedures, the CONTENTS= option specifies the text for the entries in the table of contents that is created.
- Starting with SAS 9.4M6, the CONTENTS= option accepts #BY directives.

  The #BY directives are not supported by the Report Writing Interface. For the REPORT procedure, they are supported when the ACCESSIBLETABLE system option has been specified.

  For the TABULATE procedure, 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
- 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.

  The PRINT procedure does not support the CAPTION= option.

  Starting with SAS 9.4M6, the CAPTION= option accepts #BY directives.

Recommendations Specific to the PRINT Procedure

Recommendations

In addition to the general recommendations for tables, 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 “Table with Labels for Multiple Summation Rows” on page 36.

- Define row headings by including one or more of the variables in an ID statement. 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.

  Alternatively, you can omit the NOOBS option. When you omit the NOOBS option, the observation numbers appear as row headers. However, best practice is to use columns other than the OBS column for row headers.

  See “Table with the Observation Numbers” on page 31 and “Table with a Descriptive Row Heading” on page 33.

- 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 “Table with Labels for Multiple Summation Rows” on page 36.

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 "Use the N Argument to Display the Number of Results" on page 38.

- Do not use the CAPTION= option with PROC PRINT to generate visible captions. PROC PRINT does not support the CAPTION= option. The CAPTION= option is new for SAS 9.4M6.

Recommendations Specific to the REPORT Procedure

Recommendations

In addition to the general recommendations for tables, 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 "Detail Report with Ordering and SPANROWS" on page 44 and "Summarization Row with Custom Labels" on page 45.

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

  See "Summarization Row with Custom Labels" on page 45.

- 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 "Accessible Table Using BY Directives" on page 49.

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.
- Do not use the NOHEADER option. The NOHEADER option produces inaccessible tables (they have no headers).

Recommendations Specific to the TABULATE Procedure

Recommendations

In addition to the general recommendations for tables, 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.

- Always use the ACCESSIBLETABLE system option when the procedure contains a variable in a row dimension. This option is available starting with SAS 9.4M6.
- 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.
- 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.

What to Avoid

- Do not display variable labels for concatenated or crossed 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.

Using the ACCESSIBLETABLE System Option to Change the Appearance of Tables

The default layouts of some table output in SAS can cause accessibility issues. Starting with SAS 9.4M6, the ACCESSIBLETABLE system option is available. This option changes the way that the table is visually and semantically structured in order to make it accessible. It is recommended that you always use the ACCESSIBLETABLE system option when you generate table output.

Here is an example that shows one way that the ACCESSIBLETABLE system option improves a table's accessibility. Suppose that you are using the TABULATE procedure with concatenated row variables, and you want to display the variable labels. Here is the code and output:

```
proc tabulate data=sashelp.shoes;
   class Region Subsidiary;
   var Stores;
   label Stores="Stores";
   keylabel sum=" ";
   table Region*(Subsidiary All="Total") All="Total Stores", Stores;
run;
```

Output 3.1 *(Partial Output) PROC TABULATE with Concatenated Row Variable Labels*

		Stores
Region	Subsidiary	
Africa	Addis Ababa	65.00
	Algiers	101.00
	Cairo	88.00
	Johannesburg	51.00
	Khartoum	71.00
	Kinshasa	56.00
	Luanda	30.00
	Nairobi	70.00
	Total	532.00
Asia	Subsidiary	
	Bangkok	5.00
	Seoul	59.00
	Tokyo	1.00
	Total	65.00

In the table that is produced, the Region and repeated Subsidiary row headers cause problems for screen readers trying to navigate this table. The cells cause screen readers to misread the table.

Using the ACCESSIBLETABLE system option changes the way that the table is constructed by removing the labels:

Example Code 3.1 *(Partial Output) PROC TABULATE with Concatenated Row Variables but No Labels*

```
options accessibletable;

proc tabulate data=sashelp.shoes;
   class Region Subsidiary;
   var Stores;
   label Stores="Stores";
   keylabel sum=" ";
   table Region*(Subsidiary All="Total") All="Total Stores", Stores;
run;
```

		Stores
Africa	Addis Ababa	65.00
	Algiers	101.00
	Cairo	88.00
	Johannesburg	51.00
	Khartoum	71.00
	Kinshasa	56.00
	Luanda	30.00
	Nairobi	70.00
	Total	532.00
Asia	Bangkok	5.00
	Seoul	59.00
	Tokyo	1.00
	Total	65.00

Defining Column and Row Headers

About Headers and When They Are Needed

When data tables have column and row headers, those headers must be defined in the tagging structure as headings. Defined column and row headers enable screen-reader users to easily understand the labels for each data cell as they navigate through the table.

- A column header is the row at the top of the table that identifies each column within the table.

  Most often, your tables need column headers to uniquely label each column.

- A row header is the column at the left of the table that identifies each row within the table.

  Row headers are needed if the cells in a column identify the contents of the row, in effect providing a label for the row. These labels are often a key or ID column for the rest of the data. The row headers enable screen reader users to identify labels for the data that they are reading in the rest of the row.

  If the OBS column is used, it should not be marked as a row header.

The different report writing software handles table header creation in different ways, as shown in the following sections.

PRINT Procedure

The PRINT procedure automatically creates the correct column headers for you based on the variables in the procedure.

The following code shows how to define row headers using the ID statement:

```
proc print data=sashelp.shoes noobs contents="Boot Sales in Africa";
   where Region="Africa" and Product="Boot";
   id Subsidiary;
   var Sales;
run;
```

The code produces a table with the variables defined as column headers and the subsidiaries defined as row headers.

Output 3.2 *PROC PRINT with Column and Row Headers Defined*

Subsidiary	Sales
Addis Ababa	$29,761
Algiers	$21,297
Cairo	$4,846
Johannesburg	$8,365
Khartoum	$19,282
Kinshasa	$13,921
Luanda	$6,081
Nairobi	$16,282

TABULATE Procedure

The TABULATE procedure automatically creates the correct column and row headers for you based on the variables used in the row and column dimensions.

REPORT Procedure

The REPORT procedure automatically creates the correct column headers for you. To define row headers, do both of the following in your code:

- Use either the ORDER or GROUP option in the DEFINE statement.
- Use the SPANROWS argument in the PROC REPORT statement. This argument enables row headers that span multiple rows to be assigned correctly.

```
proc report data=sashelp.shoes spanrows
            contents="United States Sales Summaries";
   where Region="United States";
   column Subsidiary Product Sales;
   define Subsidiary/group 'Subsidiary';
   define Product/group;
   define Sales/analysis sum 'Sales';
run;
```

Output 3.3 *(Partial Output) PROC REPORT with Column and Row Headers Defined*

Subsidiary	Product	Sales
Chicago	Boot	$82,483
	Men's Casual	$408,978
	Men's Dress	$261,607
	Sandal	$601
	Slipper	$329,235
	Sport Shoe	$17,347
	Women's Casual	$172,021
	Women's Dress	$293,313
Los Angeles	Boot	$85,932
	Men's Casual	$177,010
	Men's Dress	$147,670
	Sandal	$737
	Slipper	$98,866
	Sport Shoe	$16,307
	Women's Casual	$62,661
	Women's Dress	$148,129

Report Writing Interface

The Report Writing Interface in the DATA step provides significant flexibility for formatting tables. Since data in these tables is not laid out according to a predefined template, you must explicitly set both the column and the row headers.

You can define column headers in either of the following ways:

- In the FORMAT_CELL method, include a TYPE argument of H:

  ```
  obj.format_cell(text: 'Number of Stores', type: 'H');
  ```

- Wrap all of the rows containing column headers with the ROW_START and ROW_END methods:

  ```
  obj.head_start();
     obj.row_start();
        obj.format_cell(data: 'Subsidiary');
        obj.format_cell(data: 'Product');
        obj.format_cell(data: 'Sales');
     obj.row_end();
  ```

```
obj.head_end();
```

You can define row headers using the same FORMAT_CELL and TYPE technique as used with column headers:

```
obj.format_cell(text: Region, type: 'H');
```

Adding Table Descriptions with Captions and Summaries

About Table Captions and Summaries

Table captions provide a visual title or label for the table. They also provide a description of the table to screen-reading software. Table summaries are another labeling technique that provide descriptions of the table only to users of screen readers. Both labeling techniques can be used with tabular output from SAS. Both captions and summaries should uniquely identify what the table displays. Sometimes, the labels can be as simple as *Table 1*, *Table 2*, and so on. However, more helpful captions and summaries briefly describe the contents of the table. For example, *Sales for Q4*.

TABULATE and REPORT Procedures

To ensure that all users get the intended information, and to keep the visual captions and non-visual summaries synchronized, do both of the following in your program. This usage applies to the TABULATE and REPORT procedures:

- Enable the ACCESSIBLETABLE system option.
- Use the CONTENTS= argument and specify text that summarizes the contents of the table. The value of the CONTENTS= argument is used for the visible caption and the non-visual summary.

If you do not specify the ACCESSIBLETABLE option, but use only the CONTENTS= argument, only the non-visual table summary is created.

The following example shows usage for the TABULATE procedure:

```
options accessibletable;

proc tabulate data=sashelp.shoes;
   where Product="Boot";
   class Product Region;
   var Stores Sales;
   label Product="Product";
   keylabel sum=" ";
   table Region="" All="Worldwide", Sales="Sales"*f=dollar12.0 /
```

```
          contents="Boot Sales By Region";
run;
```

In the following example, the caption and summary are different:

```
options accessibletable;
proc tabulate data=sashelp.shoes;
   where Product="Boot";
   class Product Region;
   var Stores Sales;
   label Product="Product";
   keylabel sum=" ";
   table Region="" All="Worldwide", Sales="Sales"*f=dollar12.0 /
         contents="Boot Sales By Region" caption="Table 1";
run;
```

PRINT Procedure

The PRINT procedure does not create visual table captions. Use the CONTENTS= option to provide a non-visual table summary.

Report Writing Interface

To ensure that all users get the intended information, and to keep the visual captions and non-visual summaries synchronized, do both of the following in your Report Writing Interface program:

- Use the CAPTION argument for the visible caption. Specify text that summarizes the contents of the table.
- Use the DESCRIPTION argument for the non-visible summary. Specify the same text that you used for the CAPTION argument.

```
obj.table_start(description: "Sales for Q4", caption: "Sales for Q4");
```

Creating Tables from Other Procedures

Some SAS procedures produce table output that is not accessible. However, the output of these procedures can be repackaged using accessible table construction techniques in the PRINT and REPORT procedures, and in the Report Writing Interface.

For example, the UNIVARIATE procedure displays results in a multicolumn format. This layout is useful for displaying a lot of information in a condensed area. However, the layout makes it difficult for screen-reader users to understand which data cells belong with which row headers.

Here is the original code and output:

```
proc univariate data=sashelp.class;
   var Height Weight;
run;
```

Output 3.4 *(Partial Output) PROC UNIVARIATE Table with Multiple Columns, Not Accessible*

Moments			
N	19	Sum Weights	19
Mean	62.3368421	Sum Observations	1184.4
Std Deviation	5.12707525	Variance	26.2869006
Skewness	-0.2596696	Kurtosis	-0.1389692
Uncorrected SS	74304.92	Corrected SS	473.164211
Coeff Variation	8.22479143	Std Error Mean	1.17623173

Instead of displaying the results in the original format, you can do the following to make the output accessible:

- suppress the visual output with the NOPRINT option
- send the output to a new data set using the OUTTABLE option
- use PROC PRINT to display the results accessibly

```
proc univariate data=sashelp.class outtable=work.classStats noprint;
   var Height Weight;
run;

proc print data=work.classStats label noobs;
   id _VAR_;
run;
```

Output 3.5 *(Partial Output) PROC PRINT Version of PROC UNIVARIATE Results, Accessible*

Variable Name	Number of Nonmissing Observations	Number of Missing Observations	Sum of the Weights	Sum	Mean	Standard Deviation	Variance
Height	19	0	19	1184.4	62.337	5.1271	26.287
Weight	19	0	19	1900.5	100.026	22.7739	518.652

Examples for the PRINT Procedure

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.

```
options accessiblecheck;                                  /* 1 */

ods _all_ close;
ods html5 style=daisy
        path="file-path" file="procprint01.html"
        (title="Honda Cars") options(outline='no');       /* 2 */

proc odstext;
        h1 'Honda Cars';                                  /* 3  */
run;

proc print data=sashelp.cars label noobs
        contents="Honda Cars";                            /* 4  */
      label MPG_City="MPG City";
      var Model MSRP MPG_City;
      where make="Honda";
run;

ods html5 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

1. Starting with SAS 9.4M6, use the ACCESSIBLECHECK system option to check your SAS programs for common violations of accessibility standards, and to write messages to the SAS log if violations are found.
2. Here are details about the ODS environment:
   - The ODS _ALL_ CLOSE statement disables the default output environment. To customize the SAS Studio output environment, you first disable the default output environment and then establish your own output environment.
   - The ODS HTML5 destination statement sets the ODS environment. You must close this destination at the end of your program.

> **TIP** After you execute this program, be sure to open one or more ODS destinations before you execute your next program within this SAS session.

- The STYLE= option specifies the style Daisy, which is recommended for accessible output.
- The *file-path* must be accessible to the SAS Compute Server.
- 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.

3. The H1 statement provides a title for the output.
4. 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.

Output

Output 3.6 *Simple Table with a Title*

Honda Cars

Model	MSRP	MPG City
Civic Hybrid 4dr manual (gas/electric)	$20,140	46
Insight 2dr (gas/electric)	$19,110	60
Pilot LX	$27,560	17
CR-V LX	$19,860	21
Element LX	$18,690	21
Civic DX 2dr	$13,270	32
Civic HX 2dr	$14,170	36
Civic LX 4dr	$15,850	32
Accord LX 2dr	$19,860	26
Accord EX 2dr	$22,260	26
Civic EX 4dr	$17,750	32
Civic Si 2dr hatch	$19,490	26
Accord LX V6 4dr	$23,760	21
Accord EX V6 2dr	$26,960	21
Odyssey LX	$24,950	18
Odyssey EX	$27,450	18
S2000 convertible 2dr	$33,260	20

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.

```
options accessiblecheck;

ods _all_ close;
ods html5 style=daisy
         path="file-path" file="procprint02.html"
```

```
          (title="Honda Cars") options(outline='no');

proc odstext;
      h1 'Honda Cars';
run;

proc print data=sashelp.cars label contents="Honda Cars";           /*
1 */
      label MPG_City="MPG City";
      var Model MSRP MPG_City;
      where make="Honda";
run;

ods html5 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

1 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 3.7 *Table with the Observation Numbers*

Honda Cars

Obs	Model	MSRP	MPG City
150	Civic Hybrid 4dr manual (gas/electric)	$20,140	46
151	Insight 2dr (gas/electric)	$19,110	60
152	Pilot LX	$27,560	17
153	CR-V LX	$19,860	21
154	Element LX	$18,690	21
155	Civic DX 2dr	$13,270	32
156	Civic HX 2dr	$14,170	36
157	Civic LX 4dr	$15,850	32
158	Accord LX 2dr	$19,860	26
159	Accord EX 2dr	$22,260	26
160	Civic EX 4dr	$17,750	32
161	Civic Si 2dr hatch	$19,490	26
162	Accord LX V6 4dr	$23,760	21
163	Accord EX V6 2dr	$26,960	21
164	Odyssey LX	$24,950	18
165	Odyssey EX	$27,450	18
166	S2000 convertible 2dr	$33,260	20

Table with a Descriptive Row Heading

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

Program

```
options accessiblecheck;

ods _all_ close;
ods html5 style=daisy
      path="file-path" 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 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

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

Output

Output 3.8 *Table with a Descriptive Row Heading*

Honda Cars

Model	MSRP	MPG City
Civic Hybrid 4dr manual (gas/electric)	$20,140	46
Insight 2dr (gas/electric)	$19,110	60
Pilot LX	$27,560	17
CR-V LX	$19,860	21
Element LX	$18,690	21
Civic DX 2dr	$13,270	32
Civic HX 2dr	$14,170	36
Civic LX 4dr	$15,850	32
Accord LX 2dr	$19,860	26
Accord EX 2dr	$22,260	26
Civic EX 4dr	$17,750	32
Civic Si 2dr hatch	$19,490	26
Accord LX V6 4dr	$23,760	21
Accord EX V6 2dr	$26,960	21
Odyssey LX	$24,950	18
Odyssey EX	$27,450	18
S2000 convertible 2dr	$33,260	20

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 "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

```
options accessiblecheck;

ods _all_ close;
ods html5 style=daisy path="file-path" 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 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

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 3.9 *Table with a Descriptive Summation Row*

Total Home Runs by the Kansas City Royals in 1986

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

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 "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

```
options accessiblecheck;

ods _all_ close;
ods html5 style=daisy path="file-path" 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 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

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 3.10 *Table with Labels for Multiple Summation Rows*

Team=Toronto

Player	Position	Home Runs
Barfield, Jesse	RF	40
Bell, George	LF	31
Moseby, Lloyd	CF	21
Whitt, Ernie	C	16
Johnson, Cliff	DH	15
Mulliniks, Rance	3B	11
Fernandez, Tony	SS	10
Upshaw, Willie	1B	9
Garcia, Damaso	2B	6
Leach, Rick	DO	5
Iorg, Garth	32	3
Total Home Runs for Toronto		167
Total Home Runs for the League		3575

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

```
options accessiblecheck;

ods _all_ close;
ods html5 style=daisy path="file-path" 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;

ods html5 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

1 The N option prints the number of observations below the table.

Output

Output 3.11 *Table with the Number of Observations Printed outside the Table*

Honda Cars

Model	MSRP	MPG City
Civic Hybrid 4dr manual (gas/electric)	$20,140	46
Insight 2dr (gas/electric)	$19,110	60
Pilot LX	$27,560	17
CR-V LX	$19,860	21
Element LX	$18,690	21
Civic DX 2dr	$13,270	32
Civic HX 2dr	$14,170	36
Civic LX 4dr	$15,850	32
Accord LX 2dr	$19,860	26
Accord EX 2dr	$22,260	26
Civic EX 4dr	$17,750	32
Civic Si 2dr hatch	$19,490	26
Accord LX V6 4dr	$23,760	21
Accord EX V6 2dr	$26,960	21
Odyssey LX	$24,950	18
Odyssey EX	$27,450	18
S2000 convertible 2dr	$33,260	20

N=17

Examples for the Report Procedure

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.

```
options accessiblecheck;                                  /* 1 */

ods _all_ close;
ods html5 style=daisy
       path="file-path" file="procreport01b.html"
       (title="Honda Cars") options(outline='no');        /* 2 */

title1 'Honda Cars Detail Report';                        /* 3 */

proc report data=sashelp.cars contents="Honda Cars";      /* 4 */
            where Make="Honda";
run;

title1;

ods html5 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

1 Starting with SAS 9.4M6, use the ACCESSIBLECHECK system option to check your SAS programs for common violations of accessibility standards, and to write messages to the SAS log if violations are found.

2 Here are details about the ODS environment:

- The ODS _ALL_ CLOSE statement disables the default output environment. To customize the SAS Studio output environment, you first disable the default output environment and then establish your own output environment.
- The ODS HTML5 destination statement sets the ODS environment. You must close this destination at the end of your program.

> **TIP** After you execute this program, be sure to open one or more ODS destinations before you execute your next program within this SAS session.

- The STYLE= option specifies the style Daisy, which is recommended for accessible output.
- The *file-path* must be accessible to the SAS Compute Server.
- 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.

3 The TITLE1 statement provides a title for the output.

Note: Alternatively, can remove this statement and instead use the ACCESSIBLETABLE system option. That also provides a visual label for the table.

4 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.

Output

Output 3.12 *Detail Report with a Table Description*

Honda Cars Detail Report

Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	Engine Size (L)	Cylinders	Horsepower	MPG (City)	MPG (Highway)	Weight (LBS)	Wheelbase (IN)	Length (IN)
Honda	Civic Hybrid 4dr manual (gas/electric)	Hybrid	Asia	Front	$20,140	$18,451	1.4	4	93	46	51	2732	103	175
Honda	Insight 2dr (gas/electric)	Hybrid	Asia	Front	$19,110	$17,911	2	3	73	60	66	1850	95	155
Honda	Pilot LX	SUV	Asia	All	$27,560	$24,843	3.5	6	240	17	22	4387	106	188
Honda	CR-V LX	SUV	Asia	All	$19,860	$18,419	2.4	4	160	21	25	3258	103	179
Honda	Element LX	SUV	Asia	All	$18,690	$17,334	2.4	4	160	21	24	3468	101	167
Honda	Civic DX 2dr	Sedan	Asia	Front	$13,270	$12,175	1.7	4	115	32	38	2432	103	175
Honda	Civic HX 2dr	Sedan	Asia	Front	$14,170	$12,996	1.7	4	117	36	44	2500	103	175
Honda	Civic LX 4dr	Sedan	Asia	Front	$15,850	$14,531	1.7	4	115	32	38	2513	103	175
Honda	Accord LX 2dr	Sedan	Asia	Front	$19,860	$17,924	2.4	4	160	26	34	2994	105	188
Honda	Accord EX 2dr	Sedan	Asia	Front	$22,260	$20,080	2.4	4	160	26	34	3047	105	188
Honda	Civic EX 4dr	Sedan	Asia	Front	$17,750	$16,265	1.7	4	127	32	37	2601	103	175
Honda	Civic Si 2dr hatch	Sedan	Asia	Front	$19,490	$17,849	2	4	160	26	30	2782	101	166
Honda	Accord LX V6 4dr	Sedan	Asia	Front	$23,760	$21,428	3	6	240	21	30	3349	108	190
Honda	Accord EX V6 2dr	Sedan	Asia	Front	$26,960	$24,304	3	6	240	21	30	3294	105	188
Honda	Odyssey LX	Sedan	Asia	Front	$24,950	$22,498	3.5	6	240	18	25	4310	118	201
Honda	Odyssey EX	Sedan	Asia	Front	$27,450	$24,744	3.5	6	240	18	25	4365	118	201
Honda	S2000 convertible 2dr	Sports	Asia	Rear	$33,260	$29,965	2.2	4	240	20	25	2835	95	162

Table with Spanned Column Headers

This example creates a header row with three header labels that span the table's columns. The three headers are Car, Cost, and Fuel Efficiency. Row headers are created based on the labels in the data set.

Program

This example creates a header row that spans the table's columns.

```
options accessiblecheck accessibletable;

ods _all_ close;
ods html5 style=daisy path="file-path" file="procreport02.html"
       (title="Honda Cars") options(outline='no');

proc report data=sashelp.cars contents="Honda Cars";
      where Make="Honda";
      column ("Car" make model) ("Cost" msrp invoice)
           ("Fuel Efficiency" mpg_city mpg_highway);
run;

ods html5 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

Output

Output 3.13 *Table with Header That Spans Various Columns*

Honda Cars

Car		Cost		Fuel Efficiency	
Make	Model	MSRP	Invoice	MPG (City)	MPG (Highway)
Honda	Civic Hybrid 4dr manual (gas/electric)	$20,140	$18,451	46	51
Honda	Insight 2dr (gas/electric)	$19,110	$17,911	60	66
Honda	Pilot LX	$27,560	$24,843	17	22
Honda	CR-V LX	$19,860	$18,419	21	25
Honda	Element LX	$18,690	$17,334	21	24
Honda	Civic DX 2dr	$13,270	$12,175	32	38
Honda	Civic HX 2dr	$14,170	$12,996	36	44
Honda	Civic LX 4dr	$15,850	$14,531	32	38
Honda	Accord LX 2dr	$19,860	$17,924	26	34
Honda	Accord EX 2dr	$22,260	$20,080	26	34
Honda	Civic EX 4dr	$17,750	$16,265	32	37
Honda	Civic Si 2dr hatch	$19,490	$17,849	26	30
Honda	Accord LX V6 4dr	$23,760	$21,428	21	30
Honda	Accord EX V6 2dr	$26,960	$24,304	21	30
Honda	Odyssey LX	$24,950	$22,498	18	25
Honda	Odyssey EX	$27,450	$24,744	18	25
Honda	S2000 convertible 2dr	$33,260	$29,965	20	25

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.

```
options accessiblecheck;

ods _all_ close;
ods html5 style=daisy path="file-path" 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 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

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 3.14 *Detail Report with Ordering and SPANROWS*

Detail Report by Make and Model

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

Summarization Row with Custom Labels

Program

This example contains two PROC REPORT statements that produce the same report using two different techniques for including summarization rows and custom labels.

```
options accessiblecheck;

ods _all_ close;
ods html5 style=daisy path="file-path" 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;                    /* 1 */
      column Team Name nHome nHits;                    /* 2 */
```

```
      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;                                /* 3 */

run;
proc report data=sashelp.baseball contents="Home Runs and Hits By Team"
            spanrows;
      column Team Name nHome nHits;
      define Team / group;
      define Name / group;
      break after Team / summarize ;
      compute team;
          IF upcase(_break_)='TEAM' THEN  CALL
              DEFINE('team','style','style=[pretext="Total for "]');
     endcomp;
     rbreak after / summarize;
     compute after;
          IF upcase(_break_)='_RBREAK_' THEN  CALL
              DEFINE('team','style','style=[pretext="Total for all
teams"]');
     endcomp;
run;

title1;

ods html5 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

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 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 3.15 *Summarization Rows with Custom Labels*

Batting Totals in 1986

Team at the End of 1986	Player's Name	Home Runs in 1986	Hits in 1986
Atlanta	Harper, Terry	8	68
	Horner, Bob	27	141
	Hubbard, Glenn	4	94
	Moreno, Omar	4	84
	Murphy, Dale	29	163
	Oberkfell, Ken	5	136
	Ramirez, Rafael	8	119
	Sample, Billy	6	57
	Simmons, Ted	4	32
	Thomas, Andres	6	81
	Virgil, Ozzie	15	80
Total for Atlanta		116	1055
Baltimore	Beniquez, Juan	6	103
	Bonilla, Juan	1	69
	Dempsey, Rick	13	68
	Dwyer, Jim	8	39
	Lacy, Lee	11	141
	Lynn, Fred	23	114
	Murray, Eddie	17	151
			120
	Wilkerson, Curt	0	56
Total for Texas		177	1371
Toronto	Barfield, Jesse	40	170
	Bell, George	31	198
	Fernandez, Tony	10	213
	Garcia, Damaso	6	119
	Iorg, Garth	3	85
	Johnson, Cliff	15	84
	Leach, Rick	5	76
	Moseby, Lloyd	21	149
	Mulliniks, Rance	11	90
	Upshaw, Willie	9	144
	Whitt, Ernie	16	106
Total for Toronto		167	1434
Total for All Teams		3575	33294

Detail Report with ACCESSIBLETABLE System Option

Program

This example uses the system options ACCESSIBLETABLE and ACCESSIBLECHECK and PROC REPORT CAPTION= and CONTENTS= options to create an accessible table. This example shows you how to specify a visible title for the table using the CAPTION= option.

```
options accessiblecheck accessibletable;     /* 1 */

ods _all_ close;
ods html5 style=daisy path="file-path" file="procreport06.html"
       (title="Only Honda Cars") options(outline='no');

proc report data=sashelp.cars contents="Honda Cars"
        caption="Honda Cars Detail Report";        /* 2 */
            where Make="Honda";
run;

ods html5 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

1. Starting with SAS 9.4M6, use the ACCESSIBLETABLE system option to generate accessible tables. The ACCESSIBLECHECK system option detects accessibility problems and writes messages to the SAS log if accessibility problems are found.
2. 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 3.16 *Detail Report Using ACCESSIBLETABLE System Option*

Honda Cars Detail Report

Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	Engine Size (L)	Cylinders	Horsepower	MPG (City)	MPG (Highway)	Weight (LBS)	Wheelbase (IN)	Length (IN)
Honda	Civic Hybrid 4dr manual (gas/electric)	Hybrid	Asia	Front	$20,140	$18,451	1.4	4	93	46	51	2732	103	175
Honda	Insight 2dr (gas/electric)	Hybrid	Asia	Front	$19,110	$17,911	2	3	73	60	66	1850	95	155
Honda	Pilot LX	SUV	Asia	All	$27,560	$24,843	3.5	6	240	17	22	4387	106	188
Honda	CR-V LX	SUV	Asia	All	$19,860	$18,419	2.4	4	160	21	25	3258	103	179
Honda	Element LX	SUV	Asia	All	$18,690	$17,334	2.4	4	160	21	24	3468	101	167
Honda	Civic DX 2dr	Sedan	Asia	Front	$13,270	$12,175	1.7	4	115	32	38	2432	103	175
Honda	Civic HX 2dr	Sedan	Asia	Front	$14,170	$12,996	1.7	4	117	36	44	2500	103	175
Honda	Civic LX 4dr	Sedan	Asia	Front	$15,850	$14,531	1.7	4	115	32	38	2513	103	175
Honda	Accord LX 2dr	Sedan	Asia	Front	$19,860	$17,924	2.4	4	160	26	34	2994	105	188
Honda	Accord EX 2dr	Sedan	Asia	Front	$22,260	$20,080	2.4	4	160	26	34	3047	105	188
Honda	Civic EX 4dr	Sedan	Asia	Front	$17,750	$16,265	1.7	4	127	32	37	2601	103	175
Honda	Civic Si 2dr hatch	Sedan	Asia	Front	$19,490	$17,849	2	4	160	26	30	2782	101	166
Honda	Accord LX V6 4dr	Sedan	Asia	Front	$23,760	$21,428	3	6	240	21	30	3349	108	190
Honda	Accord EX V6 2dr	Sedan	Asia	Front	$26,960	$24,304	3	6	240	21	30	3294	105	188
Honda	Odyssey LX	Sedan	Asia	Front	$24,950	$22,498	3.5	6	240	18	25	4310	118	201
Honda	Odyssey EX	Sedan	Asia	Front	$27,450	$24,744	3.5	6	240	18	25	4365	118	201
Honda	S2000 convertible 2dr	Sports	Asia	Rear	$33,260	$29,965	2.2	4	240	20	25	2835	95	162

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 CONTENTS 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 SAS 9.4M6 system options ACCESSIBLETABLE and ACCESSIBLECHECK, the 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, and specify a visible title for the table using the CAPTION= option. The example also shows you how to use the BY directives.

```
options accessiblecheck accessibletable;  /* 1 */

ods _all_ close;
ods html5 style=daisy path='file-path'
          file='procreport07.html'
          (title="Class by Gender")
          contents='filename-to-contents-file'
          frame='filename-to-file-with-frames'
          options(outline='no');
```

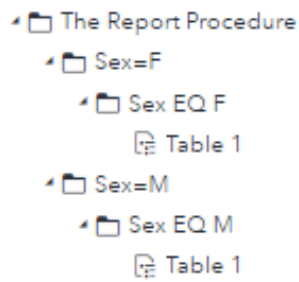
```
proc sort data=sashelp.class out=class;
   by sex;
run;

proc report data=class contents='#BYVAR1 EQ #BYVAL1'
           caption='Class Report by Gender';
       by sex; /* 2 */
run;

ods html5 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

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

2 Starting with SAS 9.4M6, specify the CAPTION= option in the PROC REPORT statement to add visible captions to tables. The caption is visible in the output when the ACCESSIBLETABLE system option is specified. BY directives can be used with the CONTENTS= option and the CAPTION= option. In this example, BY directives are used with the CONTENTS= option.

Output

Output 3.17 *Detail Report Using ACCESSIBLETABLE System Option and BY Directives*



Sex=F

Class Report by Gender

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

Sex=M

Class Report by Gender

Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Henry	M	14	63.5	102.5
James	M	12	57.3	83
Jeffrey	M	13	62.5	84
John	M	12	59	99.5
Philip	M	16	72	150
Robert	M	12	64.8	128
Ronald	M	15	67	133
Thomas	M	11	57.5	85
William	M	15	66.5	112

Examples for the TABULATE Procedure

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.

```
options  accessiblecheck;                          /* 1 */

ods _all_ close;
ods html5 style=daisy path='file-path'             /* 2 */
          file='proctabulate01.html'
         (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 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

1 Starting with SAS 9.4M6, 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 Here are details about the ODS environment:

- The ODS _ALL_ CLOSE statement disables the default output environment. To customize the SAS Studio output environment, you first disable the default output environment and then establish your own output environment.
- The ODS HTML5 destination statement sets the ODS environment. You must close this destination at the end of your program.

> **TIP** After you execute this program, be sure to open one or more ODS destinations before you execute your next program within this SAS session.

- The STYLE= option specifies the style Daisy, which is recommended for accessible output.
- The *file-path* must be accessible to the SAS Compute Server.
- 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.

3 The TITLE1 statement provides a title for the output.

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

Output

Output 3.18 *Simple Table with No Row Dimension*

Number of Students by Gender

Gender	
Female	Male
N	N
9	10

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

```
options accessiblecheck accessibletable;

ods _all_ close;
ods html5 style=daisy path='file-path'
          file='proctabulate04.html' (title="Patient Status")
          options(outline='no');

proc tabulate data=sashelp.heart;
   class Sex Status Chol_Status BP_Status;
   table Sex all, Status*(Chol_Status all) all="All
Cholesterol"      /* 1 */
      Status*(BP_Status all) all="All Blood Pressure" /
      contents="Patients by Gender, Cholesterol Status, and Blood
Pressure Status";
run;

ods html5 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

1 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 3.19 *Table with Multiple ALL Columns*

Patients by Gender, Cholesterol Status, and Blood Pressure Status

	Status								All Cholesterol	Status								All Blood Pressure
	Alive				Dead					Alive				Dead				
	Cholesterol Status			All	Cholesterol Status			All		Blood Pressure Status			All	Blood Pressure Status			All	
	Borderline	Desirable	High		Borderline	Desirable	High			High	Normal	Optimal		High	Normal	Optimal		
	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Female	682	645	588	1915	277	160	422	859	2774	619	876	420	1915	529	248	82	859	2774
Male	504	353	363	1220	398	247	418	1063	2283	454	579	187	1220	602	379	82	1063	2283
All	1186	998	951	3135	675	407	840	1922	5057	1073	1455	607	3135	1131	627	164	1922	5057

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

```
options accessibletable
accessiblecheck;                                /* 1 */

ods _all_ close;
ods html5 style=daisy path='file-path'
          file='proctabulate06.html'
          (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) /
        contents="City Mileage by Model, Type, and Origin"
        caption="Mean City
Mileage";                                  /* 2 */
   table model, mpg_highway * (mean);
run;

ods html5 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

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

2 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 3.20 *Partial Report with Default BY Group Labels in the Table of Contents*

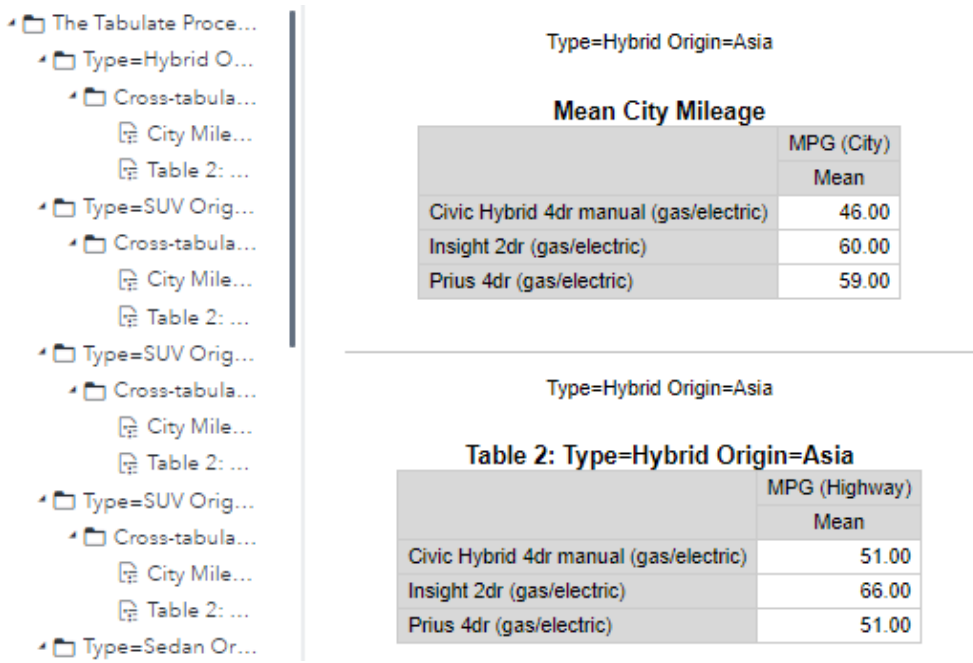


Table with BY Directives

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 accessibletable accessiblecheck;

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

ods _all_ close;
ods html5 style=daisy path='file-path'
          file='proctabulate07.html'
          (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) /
       contents="Summary Table 1"                        /* 1 */
       caption="Summary MPG City: #byvar1 = #byval1 and #byvar2 =
#byval2";
   table model, mpg_highway  * (mean) /
       contents="Summary Table 2"
       caption="Summary MPG Highway: #byvar1 = #byval1 and #byvar2 =
#byval2";
run;

title1;
ods html5 close;
/* Open one or more ODS destinations before you execute your
```

```
next program within this SAS session. */
```

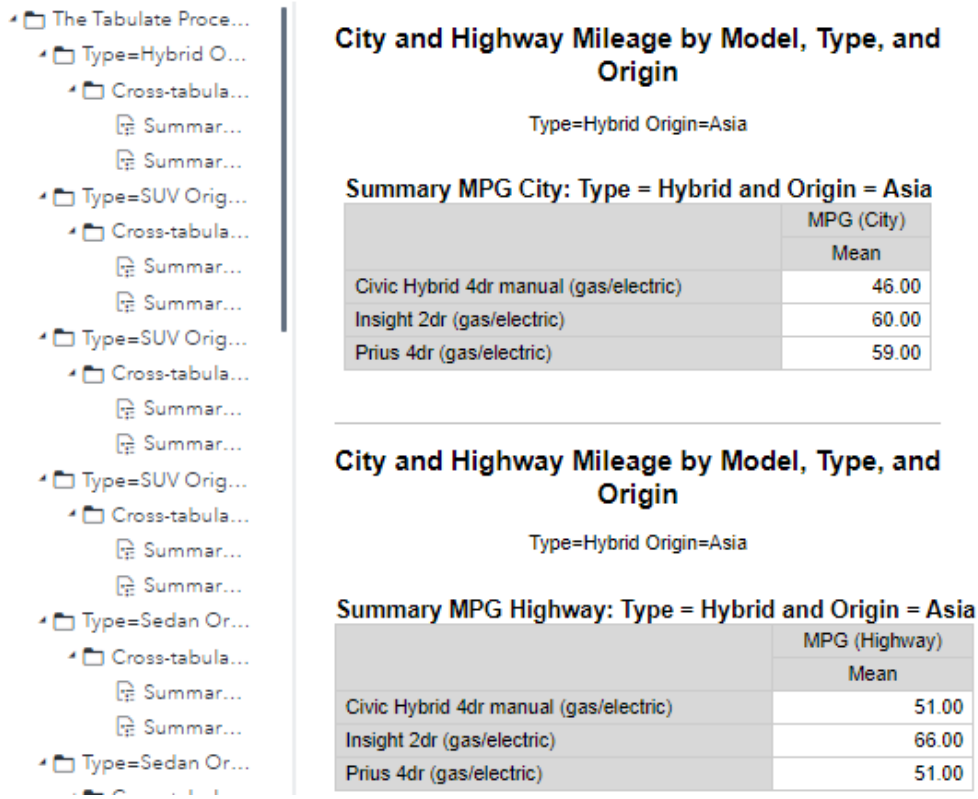
1 The CAPTION= option in the TABLE statement provides a visible table caption. The CONTENTS= option provides a table description. The ACCESSIBLETABLE system option enables this behavior.

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 strings and BY variable labels and values.

Output 3.21 *(Partial Output) HTML Report with Captions Created by BY Directives*



Examples for the Report Writing Interface

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 133 and "Create the StateP Format" on page 133.

```
options accessiblecheck
accessibletable;                          /* 1 */

ods _all_ close;
ods html5 style=daisy path='file-path'
        file='file-name'
       (title='Create an Accessible Table Using the Report Writing
Interface')

options(outline='no');                                    /* 2 */

proc odstext
contents='';                                          /* 3 */
 h1 'Create an Accessible Table Using the Report Writing Interface';
 p 'When using the Report Writing Interface, you can add descriptions
to tables.';
 p '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.' ;
run;

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();
  /* 6 */
 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.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');    /* 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,
                              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 html5 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 ACCESSIBLETABLE system option makes captions visible.

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

2 Here are details about the ODS environment:

- The ODS _ALL_ CLOSE statement disables the default output environment. To customize the SAS Studio output environment, you first disable the default output environment and then establish your own output environment.
- The ODS HTML5 destination statement sets the ODS environment. You must close this destination at the end of your program.

> **TIP** After you execute this program, be sure to open one or more ODS destinations before you execute your next program within this SAS session.

- The STYLE= option specifies the style Daisy, which is recommended for accessible output.
- The *file-path* must be accessible to the SAS Compute Server.
- 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.

3 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.

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

5 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.

6 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.

7 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.

8 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.

Output

The following output shows a table with accessibility features added.

Output 3.22 *Create an Accessible Table with the Report Writing Interface*

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.

Table Showing Percent Change in Units Sold and Median List Price

State	Units Sold		Median List Price	
	#	% Change *	#	% Change *
Alaska	212,587	***3.193%***	$312,690	-1.316%
California	166,787	***1.289%***	$294,515	-4.259%
Hawaii	222,071	***77.445%***	$294,362	-13.067%
Oregon	225,676	***2.493%***	$279,621	-14.115%
Washington	244,060	***13.689%***	$315,886	***7.184%***
* compared to last Years Results				

Key Ideas

- Use the ACCESSIBLECHECK system option to check for common accessibility violations.
- 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.
- The ODS HTML5 destination together with the Daisy style is recommended for accessible output. See Chapter 7, “Using ODS Styles to Create Accessible Output,” on page 111.
- Use the 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.
- 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.
- 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.
- Use the DESCRIPTION= or CAPTION= option where appropriate. Generally, always use DESCRIPTION= with tables and images, or CAPTION= with tables. However, when using regions and layouts for layout purposes only, DESCRIPTION= might be redundant.

See Also

- Paper SAS330-2014 Toe to Toe: Comparing ODS LAYOUT and the ODS Report Writing Interface by Daniel Kummer, SAS Institute Inc., Cary, NC
- Paper SAS5280-2016, A Guide to Section 508 Compliance Using SAS® 9.4 Output Delivery System (ODS) by Glen R Walker, SAS Institute Inc.
- 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 Chapter 8, “Appendix: ODS Syntax Related to Accessibility,” on page 127.
- For information about the ODSTEXT procedure, see *SAS Output Delivery System: Procedures Guide*.

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 134.

```
options accessiblecheck;                                        /* 1 */

ods _all_ close;
ods html5 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;                                                          /
* 6 */
   item '<a href="#toc1">Why Add Headings to Your Output?</a>';
   item '<a href="#toc2">WCAG Success Criterion</a>';
   item '<a href="#toc3">Add links to your output from a Table of
Contents</a>';
  end;
run;
ods layout end;
```

```
proc odstext
contents='';                                                /* 7 */
   h2 '<span id="toc1">Why Add Headings to Your Output?</span>';
   P '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.';
   p '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.';
run;

proc odstext contents='';
   h2 '<span id="toc2">WCAG Success Criterion this Example Satisfies</
span>';
  list;
    item  'WCAG 2.0 success criterion 2.4.2. Page Title';
    item  'WCAG 2.0 success criteria 1.3.1 Info and Relationships';
    item  'WCAG 2.0 success criteria 2.4.6. Headings and Labels';
    item  'WCAG 2.0 success criterion 2.4.1. Bypass Blocks';
  end;
run;

proc odstext contents='';
 h2 '<span id="toc3"> Add Links to Your Output from a Table of
Contents</span>';
 P 'You can create a table of contents, and then link from the TOC to
the output.';
run;

ods html5
anchor='myreport';                                          /* 8 */

proc odstext contents='';
   list;
      item  '<a href="#myreport1">Small populations</
a>' ;           /* 9 */
      item  '<a href="#myreport2">Medium populations</a>';
      item  '<a href="#myreport3">Large populations</a>' ;
      item  '<a href="#myreport4">Huge populations</a>' ;
   end;
run;
proc print data=AMRPop label
        contents='North and South American Countries by Population';
   var pop
gni;                                                        /* 10 */
   id name;
   label name='Country';
   label pop='Population';
   label fmtpop='Population Size';
   format fmtpop population.;
   by fmtpop;
run;
```

```
ods html5 close;
```

1. The ACCESSIBLECHECK system option checks for common violations of accessibility standards and writes messages to the SAS log if problems are found.
2. Here are details about the ODS environment:
   - The ODS _ALL_ CLOSE statement disables the default output environment. To customize the SAS Studio output environment, you first disable the default output environment and then establish your own output environment.
   - The ODS HTML5 destination statement sets the ODS environment. You must close this destination at the end of your program.

     TIP After you execute this program, be sure to open one or more ODS destinations before you execute your next program within this SAS session.

   - The STYLE= option specifies the style Daisy, which is recommended for accessible output.
   - The *file-path* must be accessible to the SAS Compute Server.
   - 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.
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.

Output 3.23 *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?

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

Country	Population	Gross National Income per Capita (PPP Int.$ 2004)
SAINT KITTS/NEVIS	42,696	11190
DOMINICA	78,940	5250
ANTIGUA/BARBUDA	81,485	10360
GRENADA	102,924	7000
SAINT VINCENT/GRENADINES	119,051	6250
ST. LUCIA	160,765	5560
BARBADOS	269,556	15060
BELIZE	269,736	6510
BAHAMAS	323,063	16140
SURINAME	449,238	.
GUYANA	751,218	4110

Population Size=Medium

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

Population Size=Large

Country	Population	Gross National Income per Capita (PPP Int.$ 2004)
CUBA	11,269,400	.
GUATEMALA	12,599,059	4140
ECUADOR	13,228,423	3690
CHILE	16,295,102	10500
VENEZUELA	26,749,114	5760
PERU	27,968,244	5370
CANADA	32,268,243	30660
ARGENTINA	38,747,148	12460
COLOMBIA	45,600,244	6820

Population Size=Huge

Country	Population	Gross National Income per Capita (PPP Int.$ 2004)
MEXICO	107,029,360	9590
BRAZIL	186,404,913	8020
UNITED STATES	298,212,895	39710

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 Chapter 7, “Using ODS Styles to Create Accessible Output,” on page 111.
- 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

- Paper SAS5280-2016, A Guide to Section 508 Compliance Using SAS® 9.4 Output Delivery System (ODS) by Glen R Walker, SAS Institute Inc.
- Paper SAS330-2014 Toe to Toe: Comparing ODS LAYOUT and the ODS Report Writing Interface by Daniel Kummer, SAS Institute Inc., Cary, NC
- Info and Relationships: Understanding SC 1.3.1
- Bypass Blocks: Understanding SC 2.4.1
- Page Titled: Understanding SC 2.4.2
- Headings and Labels: Understanding SC 2.4.6
- For information about language elements that are related to accessibility, see Chapter 8, “Appendix: ODS Syntax Related to Accessibility,” on page 127.

- For information about the ODSTEXT procedure, see *SAS Output Delivery System: Procedures Guide*.

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 133 and "Create the StateP Format" on page 133.

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='^';

/* define colors used for the background */
%let backBlue
=#006699;                                                     /* 2 */
```

```
options
accessiblecheck;                                                    /* 3 */

ods _all_ close;
ods html5 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                                   /* 7 */
    description='Table of Contents and Introduction';

ods region column=1
width=8in;

proc odstext contents='';
   h2 'Table of
Contents';                                                  /* 8 */
   list;
      item '<a href="#toc1">Introduction</
a>';
      item '<a href="#toc2">2013 Numbers for Commercial Properties</
a>';
      item '<a href="#toc3">2013 Numbers for Residential Properties</
a>';
   end;
run;

proc odstext contents='';
   h2 '<span id="toc1">Introduction</
span>';                              /* 9 */
   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';
run;

proc sgplot data=acmePacific noborder
       description='Bar chart showing units
sold';                      /* 10 */
  styleattrs datacolors=(darkblue lightblue)
     datacontrastcolors=(darkblue
lightblue);                           /* 11 */
   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;

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 '<span id="toc2">^S={background=&backBlue.
                  foreground=white
```

```
                    font_size=10pt}2013 Numbers for Commercial
Properties</span>';
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
 /* 15 */
(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');
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',
                          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
    description='Residential
Properties';
/* region to display the heading for 2013 & residential */
ods region width=16in column_span=2
style={background=&backBlue.};

proc odstext contents='';
   h2 '<span
id="toc3">^S={background=&backBlue.
                  foreground=white
                  font_size=10pt}2013 Numbers for Residential
Properties</span>';
run;
/* region to display the second custom RWI table */

ods region
width=8in;

proc odstext contents='';
    h3 'Percent Change in Units Sold and Median List
Price';
run;
data kumma;
data
_null_;

   set pacificSum (where=(propGroup='Residential')) 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 Residential Property',
 description:'percent change in units sold and median list price for
        residential property');
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.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');
      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.';/* 19 */
 where year=2013 and propGroup = 'Residential';
 dynamic width='3.45in' height='2.5in';
run;

/*end the gridded layout */
ods layout end;
ods html5 close;
```

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

2 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.

3 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.

4 Here are details about the ODS environment:

- The ODS _ALL_ CLOSE statement disables the default output environment. To customize the SAS Studio output environment, you first disable the default output environment and then establish your own output environment.
- The ODS HTML5 destination statement sets the ODS environment. You must close this destination at the end of your program.

> **TIP** After you execute this program, be sure to open one or more ODS destinations before you execute your next program within this SAS session.

- The STYLE= option specifies the style Daisy, which is recommended for accessible output.
- The *file-path* must be accessible to the SAS Compute Server.
- 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.

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.

6 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.

7 The ODS LAYOUT GRIDDED statements creates the layout container that contains table of contents and the introduction, which includes the bar chart. 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 table of contents and introduction.

8 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.

9 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.

10 The DESCRIPTION= option adds alternative text that describes the bar chart.

11 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.

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

13 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.

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

15 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.

16 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.

17 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.

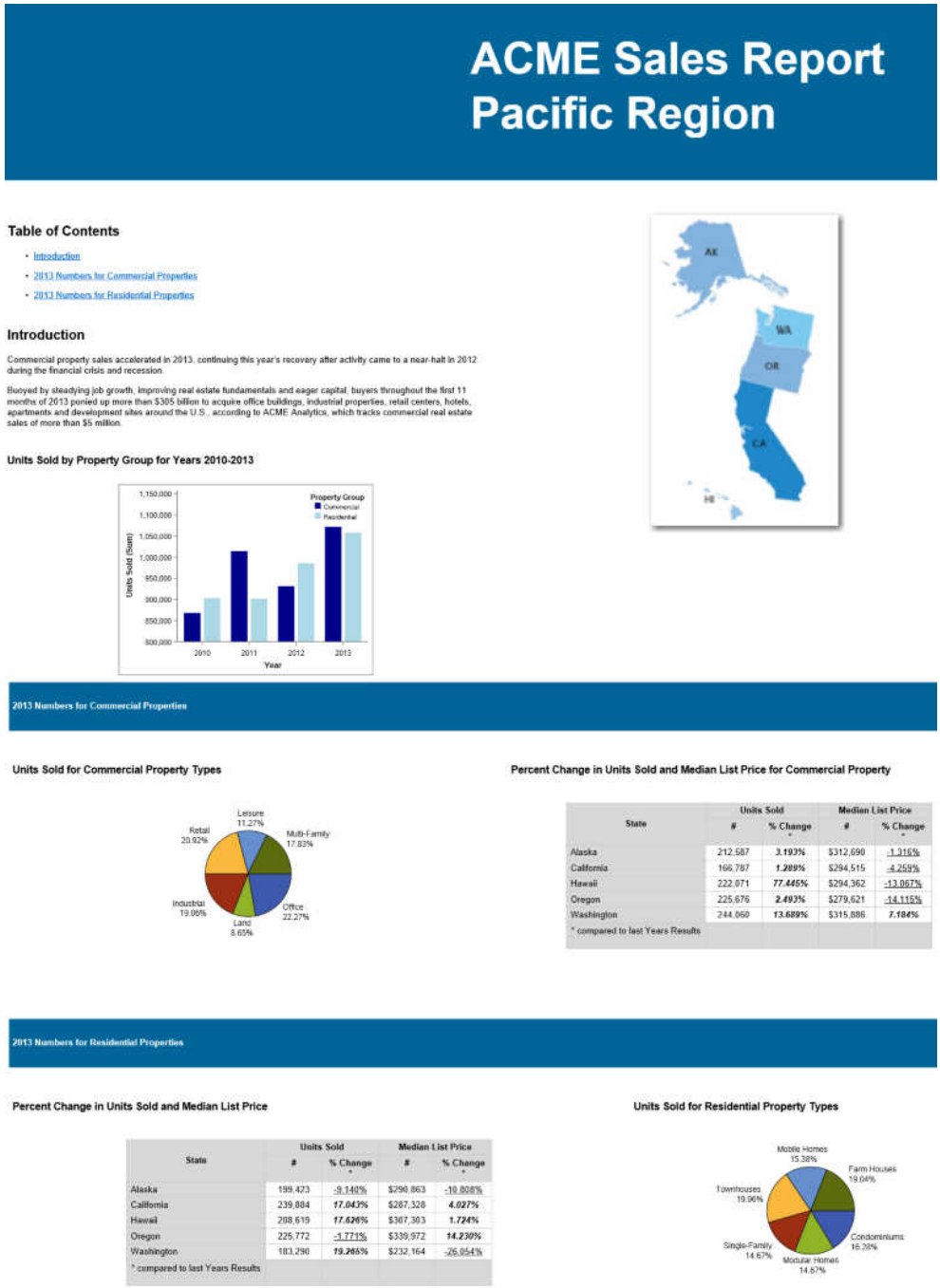
18 The STYLE_ATTR: argument enables you to customize the text inside the table cell. It is recommended that you use bold, italic, underline, and so on, to emphasize specific text or data. Using styles other than or in addition to color is essential for those with color blindness. In this table, bold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

19 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 3.24 *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 crisis and recession.

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.

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

State	Units Sold		Median List Price	
	#	% Change *	#	% Change *
Alaska	212,587	3.193%	$312,690	-1.316%
California	166,787	1.289%	$294,515	-4.259%
Hawaii	222,071	77.445%	$294,362	-13.067%
Oregon	225,676	2.493%	$279,621	-14.115%
Washington	244,060	13.689%	$315,886	7.184%

* compared to last Years Results

2013 Numbers for Residential Properties

Percent Change in Units Sold and Median List Price

State	Units Sold		Median List Price	
	#	% Change *	#	% Change *
Alaska	199,423	-9.140%	$290,863	-10.808%
California	239,884	17.043%	$287,328	4.027%
Hawaii	208,619	17.626%	$307,303	1.724%
Oregon	225,772	-1.771%	$339,972	14.230%
Washington	183,290	19.265%	$232,164	-26.054%

* compared to last Years Results

Units Sold for Residential Property Types

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 Chapter 7, “Using ODS Styles to Create Accessible Output,” on page 111.

- 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

- Paper SAS330-2014 Toe to Toe: Comparing ODS LAYOUT and the ODS Report Writing Interface by Daniel Kummer, SAS Institute Inc., Cary, NC
- Paper SAS5280-2016, A Guide to Section 508 Compliance Using SAS® 9.4 Output Delivery System (ODS) by Glen R Walker, SAS Institute Inc.
- 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 Chapter 8, “Appendix: ODS Syntax Related to Accessibility,” on page 127.
- For information about the ODSTEXT procedure, see “ODSTEXT Procedure” in *SAS Output Delivery System: Procedures Guide*.

4

Creating Accessible Graphs

Recommendations for Graphs

The "General Recommendations for Creating Accessible Output " contain best practices for all output. 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.

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

  For a list of plot types currently supported by the SAS Graphics Accelerator, see *SAS Graphics Accelerator: Supported Types of Data Visualizations*. You can also find that information on the SAS Graphics Accelerator product page on the SAS Support site.

  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.

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

  The system option affects all ODS HTML5 output until the option is disabled or the SAS session ends. If you want to enable accessible output for only a single SGPLOT procedure, you can instead specify the ACCESSIBLE_GRAPH option in the ODS HTML5 destination statement for the procedure.

  Note: In SAS Studio, you can accessible graphs as a preference for the default output environment.

  This option helps satisfy the WCAG 2.0 success criterion 1.1.1 Non-text Content.
- The graph output must be an SVG file in order to be read and interpreted by the SAS Graphics Accelerator. The graph examples in this document generate SVG output.

  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 7.
- 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 86.
- 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.

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

    TIP 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. Daisy is the recommended style.

    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 112.

- 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 "Bar Chart with Data Tips and Drill-Down Links Created with GTL" on page 96.

- 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 "Bar Chart with Data Tips and Drill-Down Links Created with GTL" on page 96.

- 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. 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 "Grouped Bubble Plot That Shows Fill Patterns" on page 90.
  - 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.

Note: The SAS Graphics Accelerator has limited support for multi-layered graphs.

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 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.

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. In the accelerator's laboratory, users can create their own data visualizations, such as graphs and maps.

Supported Operating Systems and Browser

- Operating systems:
  - Windows 10
  - macOS 10.13 or later
- Browser: Chrome 77 or later

How the Accelerator Works

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 Chapter 2, “Accessibility System Options,” on page 11.

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.

More Information

For more information about the accelerator, including main features and capabilities and an installation link, see SAS Graphics Accelerator on the SAS Support site.

See also the following documents:

- *SAS Graphics Accelerator: User’s Guide*
- *SAS Graphics Accelerator: Supported Types of Data Visualizations*

Graph Examples

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. The example also shows how to set up the ODS environment.

Program

```
options accessiblegraph accessiblecheck;            /* 1 */

ods _all_ close;
ods html5 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;

title;

ods html5 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

1 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.

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

Note: If you do not want to use the ACCESSIBLEGRAPH system option, or if you are using a release prior to SAS SAS 9.4M6, you can instead specify the ACCESSIBLE_GRAPH option in the ODS HTML5 destination statement for the procedure.

2 Here are details about the ODS environment:

- The ODS _ALL_ CLOSE statement disables the default output environment. To customize the SAS Studio output environment, you first disable the default output environment and then establish your own output environment.
- The ODS HTML5 destination statement sets the ODS environment. You must close this destination at the end of your program.

> **TIP** After you execute this program, be sure to open one or more ODS destinations before you execute your next program within this SAS session.

- The STYLE= option specifies the Daisy style. The Daisy style is recommended for accessible output.
- You must supply a valid file path and file name. The *file-path* must be accessible to the SAS Compute Server.
- 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.

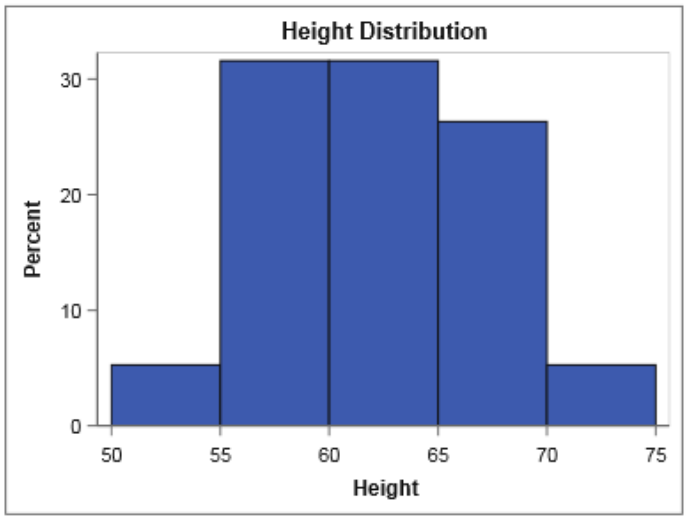
3 The TITLE statement before the PROC SGPLOT statement ensures that all users and screen readers can access the graph title.

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

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

Output

Output 4.1 *Simple Histogram Example*



See Also

"HISTOGRAM Statement" in *SAS ODS Graphics: Procedures Guide*

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.

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

```
options accessiblegraph accessiblecheck;

ods _all_ close;
ods html5 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 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

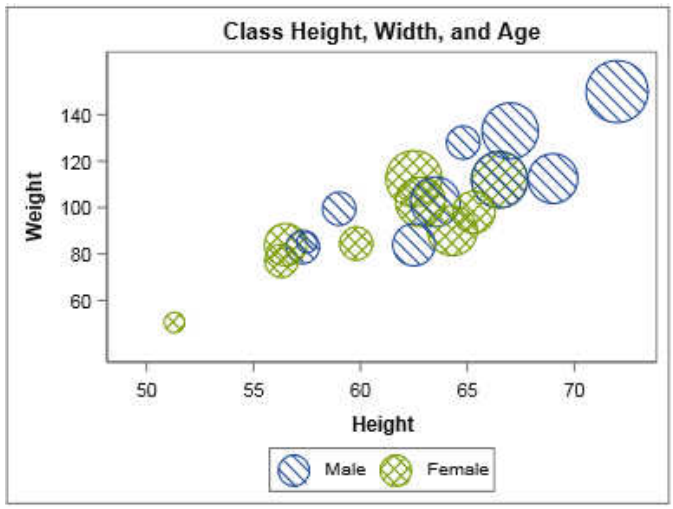
1 The FORMAT procedure creates a user-specified format for the SEX variable. The default values for this variable are M and F. The reformatted values of MALE and FEMALE are more descriptive and intuitive. This change enables the SAS Graphics Accelerator to convey a user-friendly identifier for each variable.

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.

3 The TITLE option in the KEYLEGEND statement sets the title to NULL. This in effect removes the title from the legend.

Output

Output 4.2 *Grouped Bubbles with Fill Patterns*



See Also

"BUBBLE Statement" in *SAS ODS Graphics: Procedures Guide*

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.

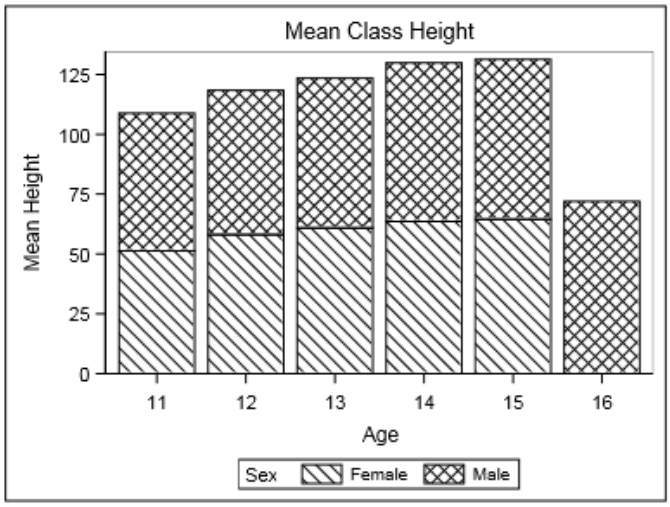
Program

```
options accessiblegraph accessiblecheck;

ods _all_ close;
ods html5 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 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

1 The STYLE= option specifies the Journal2 style.

Output

Output 4.3 *Grouped Bars with Fill Patterns*



See Also

"Using Gray-Scale Styles to Distinguish Grouped Bar Charts" in *SAS ODS Graphics: Procedures Guide*

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

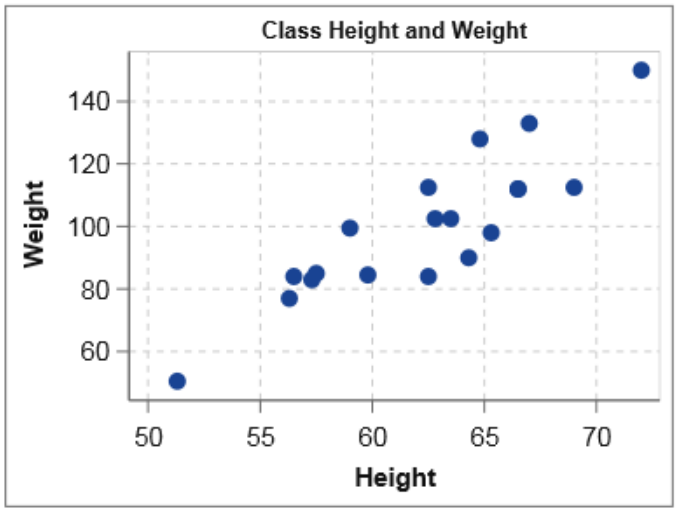
```
options accessiblegraph accessiblecheck;

ods _all_ close;
ods html5 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 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */
```

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 4.4 *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*
- "SCATTER Statement" in *SAS ODS Graphics: Procedures Guide*

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 with 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

```
%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
               fillpatternattrs=(color=black);
            discretelegend "sales" / itemsize=(heightscale=1.5);
         endlayout;
      endgraph;
   end;
run;

options accessiblegraph accessiblecheck;

ods graphics / reset imagemap=on;                 /* 8 */
ods html5                                         /* 9 */
   style=styles.daisy path="&outpath"
   file="sales.html" (title="Bar Chart of Actual Sales");
proc sgrender data=sales template=basechart
   description="Bar chart of division and actual sales.";
run;

ods html5 close;
/* Open one or more ODS destinations before you execute your
   next program within this SAS session. */

ods graphics / reset;
```

1. 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.
2. 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.
3. Sort the data in Work.Sales by Division and Product so that the chart categories are in ascending order.
4. 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.

5 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.

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

7 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.

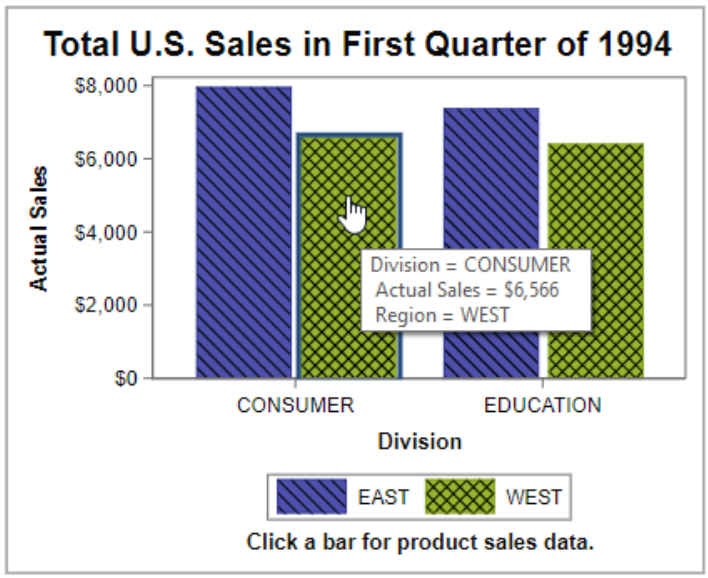
8 The ODS GRAPHICS statement enables image maps, which is required for data tips and drill-down links in HTML output.

9 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.

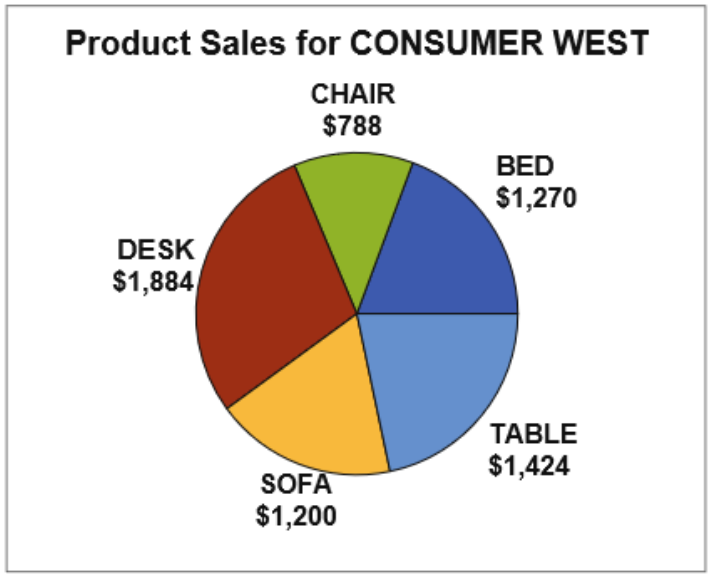
Note: Starting with SAS 9.4M6, the Daisy ODS style includes the FOCUSLINK style element. Prior to SAS 9.4M6, you must create a custom style that adds the FOCUSLINK style element to the Daisy style or to any other ODS style that you want to use. See "Creating Your Own Style That Includes the FocusLink Style Element" on page 124.

Output

Output 4.5 *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 123.
- 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.

See Also

- *SAS Graph Template Language: Reference*
- *SAS Graph Template Language: User's Guide*

5

Enhancing Your Report with Headings, Text, and Lists

Using the ODSTEXT Procedure to Enhance Your Report with Text

A report seldom consists of only tables and graphs. Typically, you provide additional text in the report to show section headings and to provide context for the data that is being presented. You can present all of this information accessibly with the ODSTEXT procedure.

The following topics explain how to use the ODSTEXT procedure:

- “Adding Accessible Headings” on page 104
- “Adding Plain Text” on page 104
- “Adding Lists” on page 104

See Also

"ODSTEXT Procedure" in *SAS Output Delivery System: Procedures Guide*

Adding Accessible Headings

When you use text to denote the beginning of a section in your report, you often use a larger and bolder font along with extra white space for the text. This formatting helps signify that the text is labeling the content that follows it.

None of this visual information is conveyed to screen reader users, however. To make sure that all users can understand the text as a heading, mark the text as an actual heading using the ODSTEXT procedure and the HEADING statement.

The following example uses the ODSTEXT procedure to specify the headings (H1 and H2):

```
proc odstext contents="";
   h1 'My Report';
   h2 'Overview';
   p 'This report contains...';
   h2 'Summary';
   p 'In conclusion...';
run;
```

Adding Plain Text

You can insert plain text into your report by using the ODSTEXT procedure with the P statement, as shown here:

```
proc odstext;
    p 'Hello world';
run;
```

Adding Lists

Use either the ODSTEXT or the ODSLIST procedure to create numbered (ordered) or bulleted (unordered) lists. These procedures structure list items as a proper accessible list. In this way, screen reader users can understand the total number of items in a list and their relationship to each other.

Example Numbered List

This example uses the ODSTEXT procedure to create the following numbered list:

1. Open SAS Studio.
2. Write your SAS program.
3. Run your SAS program.

```
proc odstext;
   list / start=1 style={liststyletype="decimal"};
      item 'Open SAS Studio.';
      item 'Write your SAS program.';
      item 'Run your SAS program.';
   end;
run;
```

Example Bulleted List

This example uses the ODSLIST procedure to create the following bulleted list. In the list, the various cities are nested within the bullet items for the continents.

- Africa
  - Addis Ababa
  - Algiers
- Asia
  - Bangkok
  - Seoul

```
proc odslist;
   item;
      p 'Africa';
      list;
        item 'Addis Ababa';
        item 'Algiers';
      end;
   end;
   item;
      p 'Asia';
      list;
        item 'Bangkok';
        item 'Seoul';
      end;
   end;
run;
```

Example of What to Avoid

When you create lists, do not rely on space characters and special characters such as asterisks (*) to format the lists. Reliance on those characters to format your list results in output that looks like a list, but it is not semantically defined as a list.

For example, do not use code similar to the following:

```
ods escapechar="^";
proc odstext;
   p '* Fruits';
   p '^_^_^_* Apple';
   p '^_^_^_* Banana';
   p '* Vegetables';
   p '^_^_^_* Carrot';
   p '^_^_^_* Turnip';
run;
```

6

Adding Non-Graph Images to Your Report

About Non-Graph Images

Many reports contain graphs, such as those that are generated with the SGPLOT procedure. However, you might want to add an image other than a graph. Such images might provide context for your report or support your brand. It is recommended that you add alternative text description to those images. For more information, see “Adding Descriptions to Images” on page 107.

You might mark images to be ignored by screen reading software if the images are decorative or if they duplicate information that appears elsewhere. For more information, see “Marking Images as Decorative” on page 108.

Adding Descriptions to Images

SAS has multiple ways to add alternative text descriptions to images.

PREIMAGE and POSTIMAGE Attributes

You can add a description to an image that is inserted with the PREIMAGE and POSTIMAGE attributes. The following techniques work wherever inline styles can be defined:

- Add a text description with the DESCRIPTION= argument:

```
ods escapechar="^";
ods text="^{style inline [preimage='high.png' description='High performance']}";
```

- Add a text description as an inline argument:

```
ods escapechar="^";
ods text="^S={preimage='high.png?desc=High performance'}";
```

Report Writing Interface

You can add a text description as an argument using the Report Writing Interface in the DATA step:

```
data _null_;
   dcl odsout obj();
   obj.image( file: "high.png", description: "High performance");
run;
```

Marking Images as Decorative

You might mark images to be ignored by screen reading software if the images are decorative or if they duplicate information that appears elsewhere. As a best practice, do not mark an image as decorative simply to avoid providing an alternative text description.

PREIMAGE and POSTIMAGE Attributes

The following techniques work wherever inline styles can be defined.

- Provide a space for the DESCRIPTION= option:

```
ods escapechar="^";
ods text="^{style inline [preimage='clouds.png' description=' ']}";
```

- Do not provide a description or a space in the inline argument:

```
ods escapechar="^";
ods text="^S={preimage='clouds.png?desc='}";
```

Report Writing Interface

You can provide an empty description as an argument using the Report Writing Interface in the DATA step:

```
data _null_;
   dcl odsout obj();
   obj.image( file: "clouds.png", description: "");
run;
```

Background Image

Another way to force screen readers to ignore an image is to use the image as a background image.

The following example defines a style for a page in a PDF document with a background image.

Note: The background image must be sized to occupy the whole page.

```
proc template;
   define style watermarkPearl;
     parent=styles.pearl;
     class body from document /
         background=_undef_
         backgroundimage="cloud-rambler-background-image.png";
     end;
run;
%let black = cx000000;
/* Code to write the first page with the background image. */
```

7

Using ODS Styles to Create Accessible Output

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 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 7.1 Group-Value Visual Attributes Specified by the Daisy, VaDark, HighContrast, and Journal2 Styles

Group-Value Attributes	ODS Style			
	Daisy	VaDark	HighContrast	Journal2
colors specified in style	12	8	8	12
contrasting colors specified in style	12	8	8	12
marker symbols specified in style	7	7	7	15
line patterns specified in style	11	11	11	11
fill patterns specified in style	none prior to SAS 9.4M5 11 starting with SAS 9.4M5	none prior to SAS 9.4M5 11 starting with SAS 9.4M5	none prior to SAS 9.4M5 11 starting with SAS 9.4M5	15
maximum unique marker symbol and color combinations[1]	84	56	56	15
maximum unique line pattern and color combinations[2]	132	88	88	11
maximum unique fill colors [3]	36	24	24	36
maximum unique fill color and fill pattern combinations [4]	396	264	264	180
attribute rotation pattern[5]	color-priority prior to SAS 9.4M6 default starting with SAS 9.4M6	default	default	default

1 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.

2 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.

3 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.

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

5 See "The Default Attribute Rotation Pattern" on page 114 and "The Color-Priority Attribute Rotation Pattern" on page 114.

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*.

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*.

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.

```
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" in *SAS Graph Template Language: Reference*.

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:

```
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 7.1 *Sample Bar Chart in the Daisy Style*

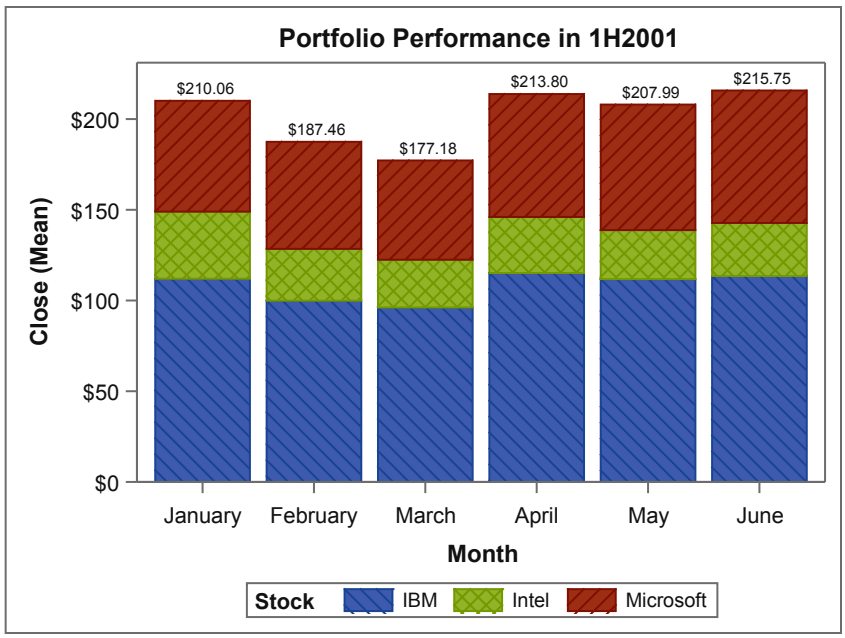


Figure 7.2 Sample Series Plot in the Daisy Style

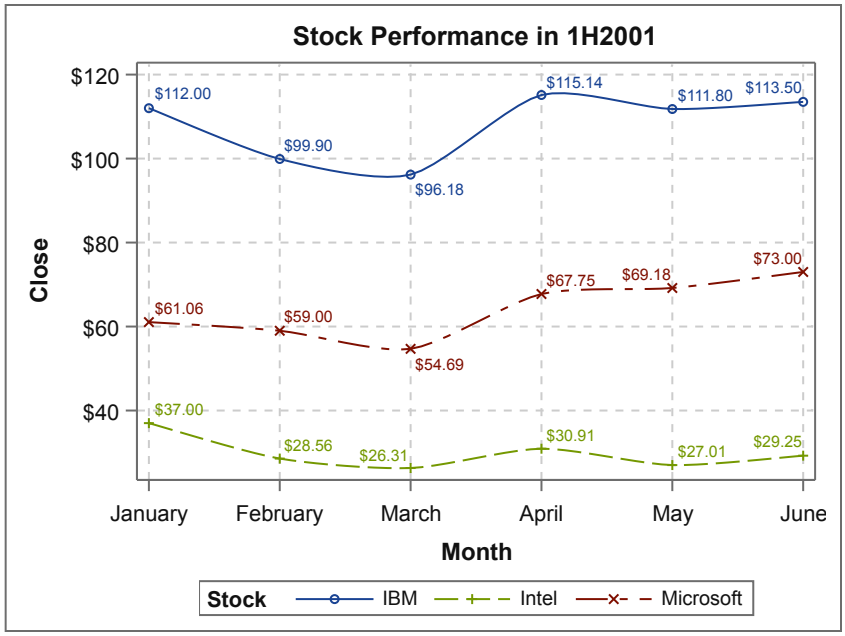


Figure 7.3 Sample Table Output in the Daisy Style

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

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 120.

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 7.4 *Sample Bar Chart in the VaDark Style*

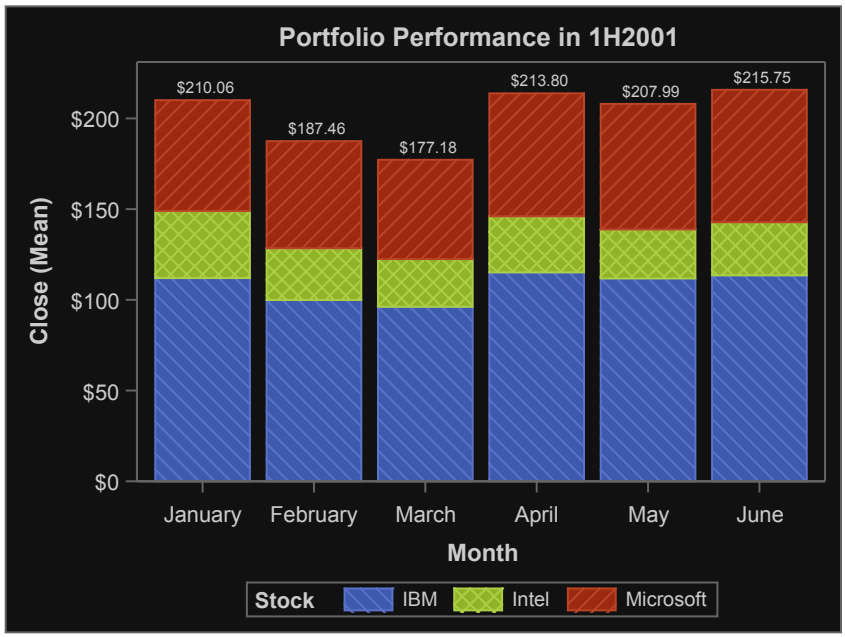


Figure 7.5 Sample Series Plot in the VaDark Style

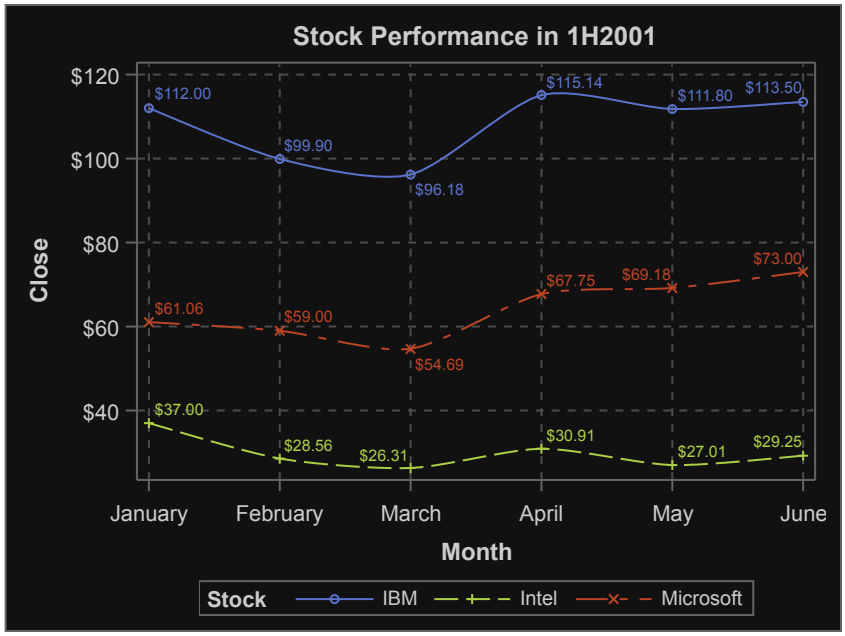


Figure 7.6 Sample Table Output in the VaDark Style

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

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 7.7 Sample Bar Chart in the HighContrast Style

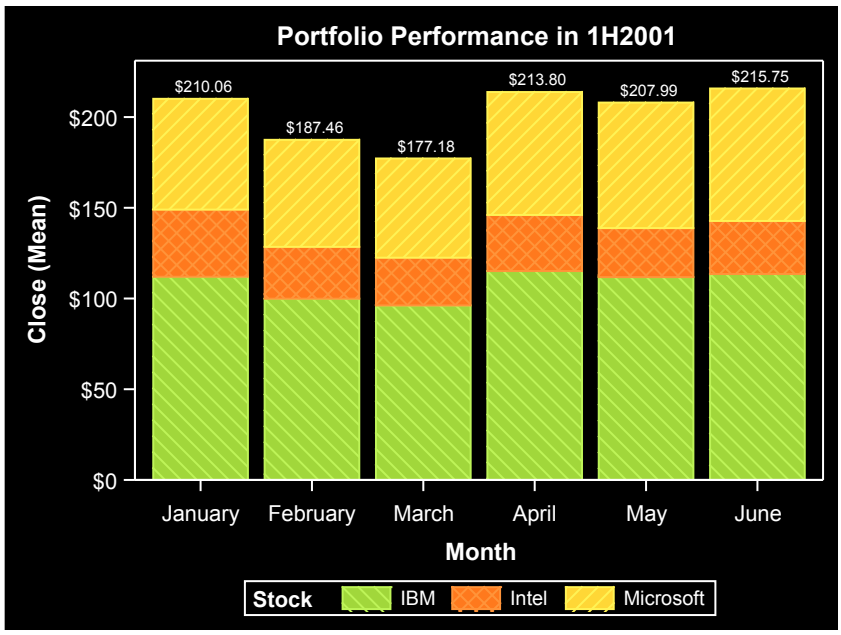


Figure 7.8 Sample Series Plot in the HighContrast Style

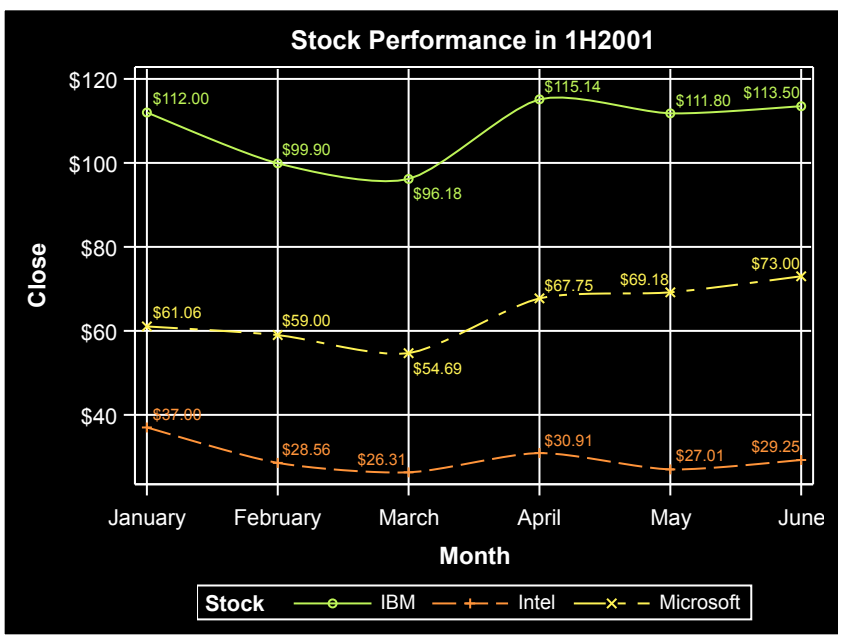


Figure 7.9 Sample Table Output in the HighContrast Style

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

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 7.10 Sample Bar Chart in the Journal2 Style

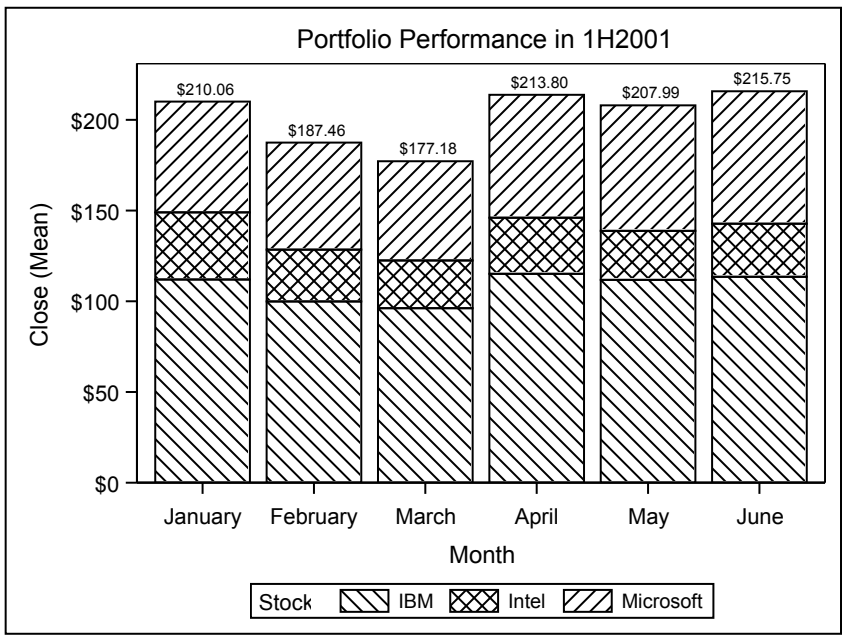


Figure 7.11 Sample Series Plot in the Journal2 Style

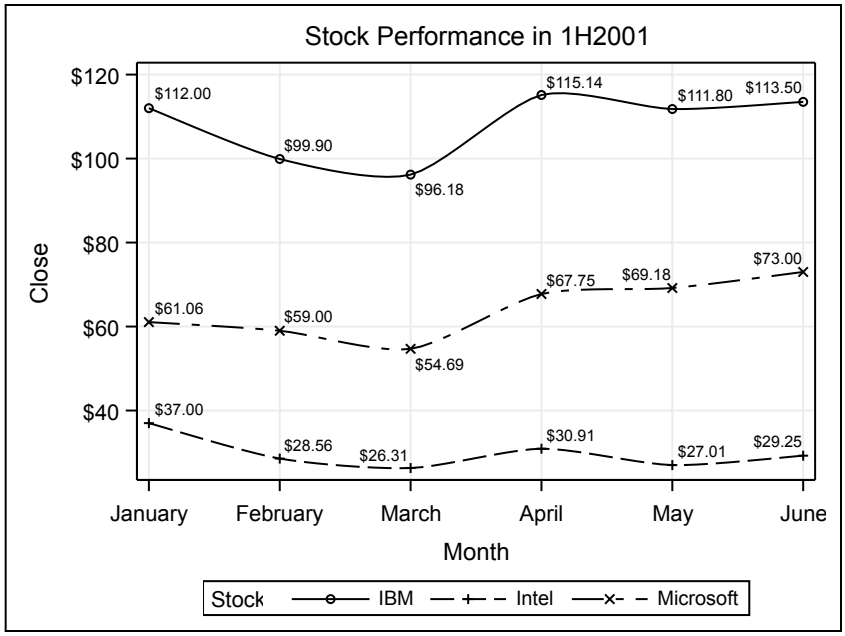


Figure 7.12 Sample Table Output in the Journal2 Style

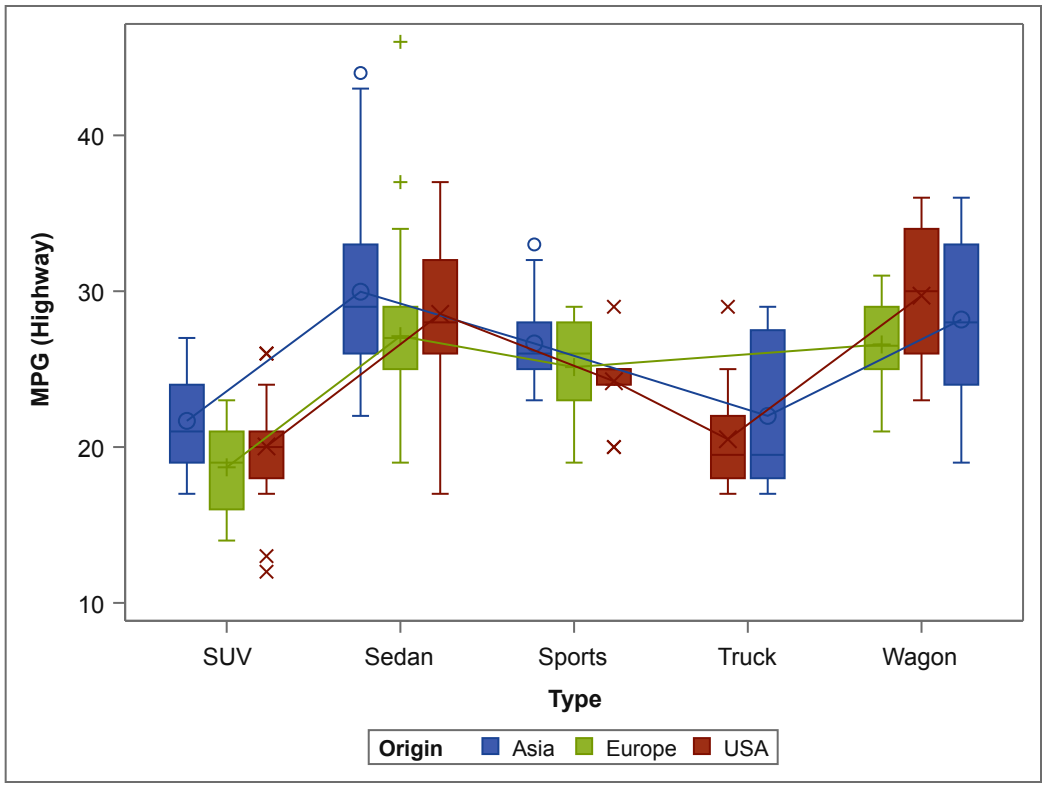
Month	IBM	Intel	Microsoft	Portfolio
January	$112.00	$37.00	$61.06	$210.06
February	$99.90	$28.56	$59.00	$187.46
March	$96.18	$26.31	$54.69	$177.18
April	$115.14	$30.91	$67.75	$213.80
May	$111.80	$27.01	$69.18	$207.99
June	$113.50	$29.25	$73.00	$215.75

Cases in Which ODS Style Alone Might Not Be Sufficient

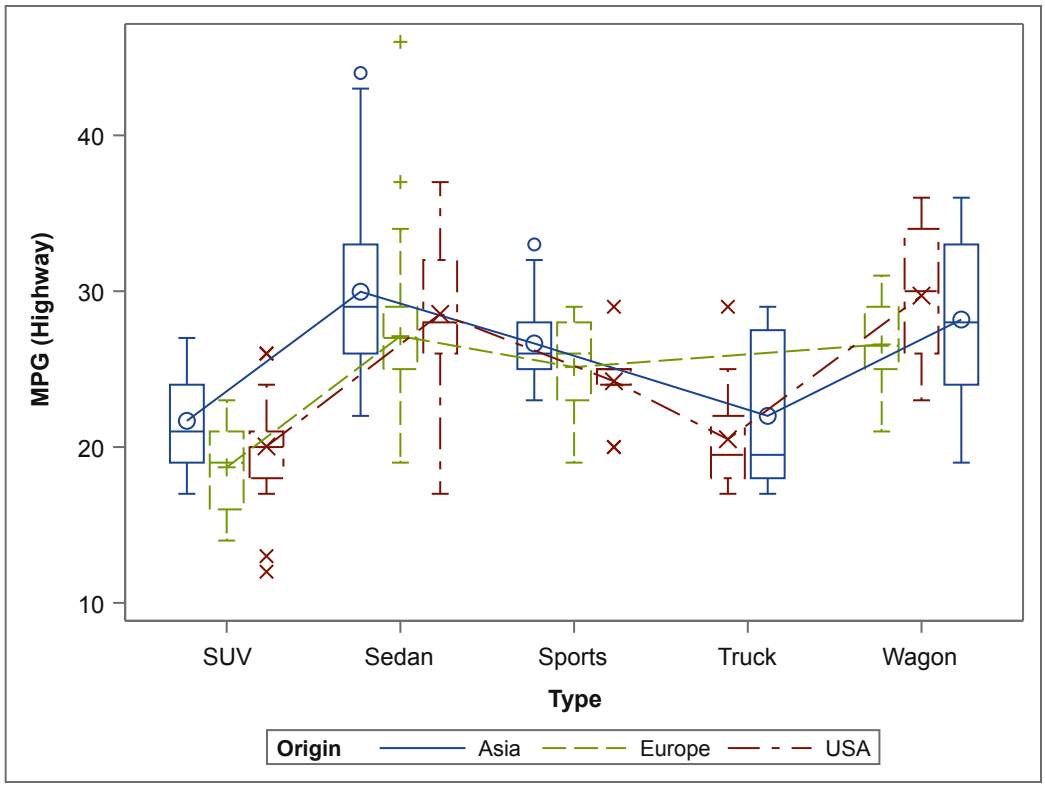
There are some cases in which ODS styles alone might not provide sufficient contrast between some graphics elements. These cases can include the following:

- 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 Figure 7.2 on page 116, 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 the style element FocusLink, which enables you to control the appearance of the link-focus indicator in ODS output. You can use this style element to make the focus indicator more prominent in your ODS 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 9.4M5, 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.

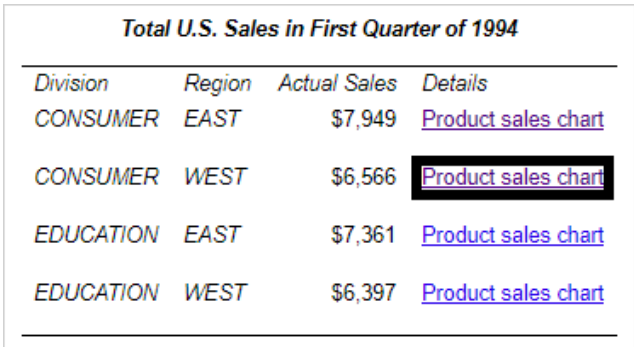
Using Your Focus Link ODS Style

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.

```
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.



Total U.S. Sales in First Quarter of 1994

Division	Region	Actual Sales	Details
CONSUMER	EAST	$7,949	Product sales chart
CONSUMER	WEST	$6,566	Product sales chart
EDUCATION	EAST	$7,361	Product sales chart
EDUCATION	WEST	$6,397	Product sales chart

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 "Bar Chart with Data Tips and Drill-Down Links Created with GTL" on page 96.

8

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:

```
ods html5 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.

```
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.

```
ods html5 style=daisy file="your-file-path/housing.html"
    (title="Create an Accessible Table Using the Report Writing Interface");
```

For information about the options, see "ODS HTML5 Statement" in *SAS Output Delivery System: User's Guide*.

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 59.

For a usage example, see "Create an Accessible Report That Includes Text, Tables, and Graphs" on page 70.

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 64.

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

See also this example on page 96.

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*.

9

Appendix: Input Data for Examples

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

Create the Regfmt, Divfmt, and UseType Formats

```
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

```
data energy;
   length State $2;
   input Region Division state $ Type Expenditures;
   datalines;
1 1 ME 1 708
1 1 ME 2 379
1 1 NH 1 597
1 1 NH 2 301
1 1 VT 1 353
1 1 VT 2 188
1 1 MA 1 3264
1 1 MA 2 2498
1 1 RI 1 531
1 1 RI 2 358
1 1 CT 1 2024
1 1 CT 2 1405
1 2 NY 1 8786
1 2 NY 2 7825
1 2 NJ 1 4115
1 2 NJ 2 3558
1 2 PA 1 6478
1 2 PA 2 3695
4 3 MT 1 322
4 3 MT 2 232
4 3 ID 1 392
4 3 ID 2 298
4 3 WY 1 194
4 3 WY 2 184
4 3 CO 1 1215
4 3 CO 2 1173
4 3 NM 1 545
4 3 NM 2 578
4 3 AZ 1 1694
4 3 AZ 2 1448
4 3 UT 1 621
4 3 UT 2 438
4 3 NV 1 493
4 3 NV 2 378
4 4 WA 1 1680
4 4 WA 2 1122
4 4 OR 1 1014
4 4 OR 2 756
4 4 CA 1 10643
4 4 CA 2 10114
4 4 AK 1 349
4 4 AK 2 329
4 4 HI 1 273
4 4 HI 2 298
```

```
;
```

Create the StateP Format

```
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";
do 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;
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;
```