

A Guide to Section 508 Compliance Using SAS® 9.4 Output Delivery System (ODS)

Glen R Walker, SAS Institute Inc.

ABSTRACT

Disability rights groups are using the court system to exact change. As a result, the enforcement of Section 508 and similar laws around the world has become a priority. When you generate SAS® output, you need to protect your organization from litigation. This paper describes concrete steps that help you use SAS® 9.4 Output Delivery System (ODS) to create SAS output that complies with accessibility standards. It also provides recommendations and code samples that are aligned with the accessibility standards defined by Section 508 and the Web Content Accessibility Guidelines (WCAG 2.0).

INTRODUCTION

Do you generate information and output for a large audience of users? Maybe you work for a government agency that is serving a local or regional population. Or you might work for an educational institution that posts reports for students. Or maybe you work for a large multi-national corporation.

If your user-base is broad, then the users' levels of abilities are probably broad too. Some users might have perfect vision. But others might have difficulty seeing small fonts or discerning between different colors. Still others might have mobility issues and can't use a mouse to navigate a report. Whatever the issue, what they all have in common is the need for equal access to your output. In the United States, almost 1 out of every 5 people have a disability (57 million people). Worldwide, it is about 1 out of 7 people. One billion people. Chances are good that someone that accesses your report will have some form of disability.

Whatever your requirements might be (whether your output contains tables, graphs, or links), SAS® 9.4 Output Delivery System (ODS) offers a tremendous number of built-in accessibility features that can help to increase the accessibility of your output. You have great information. Don't hide it from any of your users. Don't isolate any of your users by preventing access to your reports. By taking advantage of the following ODS features and by keeping in mind some basic accessibility requirements, you can generate output that is accessible to all.

A NOTE ABOUT THE EXAMPLES

The examples that follow were all created in SAS 9.4M3 and use the HTML5 destination. ODS supports over a dozen different destinations but HTML5 provides the best accessibility. PDF is also a very common destination and is talked about in the futures section near the end.

A few external third-party tools are mentioned that can assist in checking for accessibility. These tools are just an example. SAS does not endorse them nor claim they are completely accurate. You can find other tools by searching for "accessibility checker".

WHAT IS SECTION 508 AND WCAG?

Section 508 is an amendment to the Rehabilitation Act of 1973, a United States federal law, and is specific to the United States. WCAG is the Web Content Accessibility Guidelines, typically referred to as WCAG 2.0, and is an internationally accepted list of accessibility guidelines published by the Web Accessibility Initiative (WAI) of the World Wide Web Consortium (W3C). Both Section 508 and WCAG 2.0 are guidelines for creating accessible content. The United States Access Board, the federal agency that defines Section 508, will soon be refreshing Section 508 and will be retiring the current numbering system (discussed later) and will replace it with references to WCAG 2.0. For this reason, this paper uses the WCAG 2.0 numbering system but includes references to the Section 508 numbers.

www.section508.gov says: "In 1998, Congress amended the Rehabilitation Act of 1973 to require Federal agencies to make their electronic and information technology (EIT) accessible to people with disabilities. The law applies to all Federal agencies when they develop, procure, maintain, or use electronic and information technology. Under Section 508, agencies must give disabled employees and members of the public access to information that is comparable to access available to others."

WHAT DOES THIS MEAN AND HOW DOES IT AFFECT YOU?

The reports you generate must be accessible to everyone. Someone with low vision might need to zoom the text and expect the text to reflow appropriately. Or they might need to change the color contrast to make it easier to read. Someone with no vision needs to be able to navigate your report via the headers and tables. As you generate reports, there are a few key points you should keep in mind such as specifying page titles, using headings for sections and tables, and generating tables of contents, that will increase the accessibility of your reports.

COMPARING SECTION 508 WITH WCAG 2.0

Section 508 consists of several sets of standards. The ones applicable to ODS output are §1194.21, "Software applications and operating systems," and §1194.22, "Web-based intranet and internet information and applications." §1194.21 has twelve standards and §1194.22 has sixteen standards.

WCAG has 61 criteria but don't fret, not all 61 items in WCAG will be relevant to your reports. There's also a lot of overlap between Section 508 and WCAG.

WCAG 2.0 criteria are broken down into compliance levels called A, AA ("double A"), and AAA ("triple A"). Level A is a baseline minimum requirement and contains 25 items. It provides basic functionality that is required by accessibility users in order to access the report. Level AA provides functionality that not only allows the user access to the report, but allows them to use it easily and efficiently. It contains an additional 13 items to make a total of 38 items. Level AAA is icing on the cake. It has 23 additional items and takes into consideration the variety of disabilities people might have, making the total 61 items. There are several AAA items that are desirable and attainable and not too difficult to implement. Most websites try to obtain level AA compliance. Even among the 38 AA items, there will be many that don't apply to your reports.

[Appendix 1](#) has a table of the twelve §1194.21 standards and the sixteen §1194.22 standards with links from the table back into the relevant sections of this paper. [Appendix 2](#) has a table of the 38 + 1 WCAG 2.0 AA requirements with similar links. One additional WCAG AAA criterion is included, 1.4.6 Contrast (Enhanced), and is explained in the "no vision" section.

Section 508 uses letters for their standards while WCAG uses a three-part number, such as 4.2.1

CATEGORIZING THE REQUIREMENTS

For the purposes of this paper, the functional requirements will not be addressed in the order of their respective lists as shown in appendixes 1 and 2 but rather the requirements will be grouped into different categories based on how they affect users. The five groups are:

- [Mobility](#)
- [Low vision](#)
- [No vision](#)
- [Cognitive](#)
- [Hearing](#)

In the category sections below, only the requirements that are relevant to ODS will be listed. All others are included in appendixes 1 and 2. As each category is discussed, the name of the WCAG criteria is used as the section header.

ACCESSIBILITY ≠ DISABILITY

Accessibility doesn't always mean you're designing for someone with a disability. There are different degrees and phases of disabilities, some of which have nothing to do with eye site or wheelchairs.

- Permanent disability – this is either congenital or it occurs at some point in your life. For example, a person with diabetes might lose their eye site.
- Temporary disability – this is a short-lived inconvenience. For example, if you break your arm on your dominant side you might have to adjust to using a mouse with your non-dominant hand. You might lack the motor skill to effectively point and click with a mouse and need to rely on keyboard traversal.
- Situational disability – this is similar to a temporary disability but it is the environment that's causing the disability. For example, if you're viewing a mobile device in bright sunshine, you might need better color contrast ratios.
- Diminishing disability – this typically affects all of us at some point. Our abilities diminish as we age. As the eye ages, the lens starts to harden and discolor, requiring glasses or better contrast ratios. Font sizes generally have to be increased. Hearing might diminish, requiring closed captions.

The wonderful thing about accessibility is that it befits everyone, just like other accessible and universal designs. For example, curb cuts were originally designed so that wheelchair users could easily move from the sidewalk to the street. However, they also benefit bicyclists, grocery carts, and baby strollers. Closed captioning was originally intended for people with hearing loss but they also benefit people who don't speak a language as their primary language. Have you ever been in a noisy gym on a treadmill, watching the TV but reading the closed captions as you work out? Accessibility benefits everyone.

The same is true for your accessible reports. For example, one of the criterion discussed below talks about not relying solely on color to provide meaning. Even if you have perfect color vision, what happens if you print that report on a black and white printer? The bars or lines on your graph might not be distinguishable from each other.

MOBILITY

This is a large encompassing category that affects all types of users and general concerns using the keyboard. These criteria will often be used in concert with low/no vision tools. For example, a screen reader user needs to navigate using the keyboard, not a mouse.

In addition, these criteria are the basis for many other criteria in WCAG 2.0. So if something doesn't meet 2.1.1, for example, it won't meet several other WCAG criteria as well.

SECTION 508

- 1194.21(a) When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.
- 1194.22(o) A method shall be provided that permits users to skip repetitive navigation links.

WCAG 2.0

- 2.1.1 Keyboard (1194.21(a))
- 2.1.2 No Keyboard Trap
- 2.4.1 Bypass blocks (1194.22(o))
- 2.4.3 Focus order
- 3.2.1 On Focus

There is very little to do in ODS to support mobility. Since HTML is displayed in the browser, basic keyboard navigation already works in the browser, like scrolling, tabbing to links, selecting links, and so on.

Also, since ODS output contains tables and graphs from SAS procedures but doesn't contain other items on the page, such as navigation menus, side panels, banners, and so on, there is nothing to bypass so 1194.22(o) is satisfied.

However, if your output is several pages long and contains many tables and/or graphs, it would be helpful to keyboard and screen reader users if you had a table of contents at the top of your report with links to each table or graph. The following example prints a data set with a BY variable. Each BY table is a link. The ANCHOR= option is specified on the ODS statement and the anchor name is used to create a list of <H2> links at the top of the page. ODS will append a number on the ANCHOR name when it generates each table. By clicking on a link, or tabbing to it and pressing Enter, you are taken to the appropriate table.

```
data a;
    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;

ods html5 file='foo.html' anchor='myreport';

proc odstext contents='';
    p "<h1>North and South American Countries by population</h1>";
    p "<h2><a href='#myreport1'>Small populations</a></h2>";
    p "<h2><a href='#myreport2'>Medium populations</a></h2>";
    p "<h2><a href='#myreport3'>Large populations</a></h2>";
    p "<h2><a href='#myreport4'>Huge populations</a></h2>";
run;

proc print data=a label;
    var name pop gni;
    id name;
    format fmtpop population.;
    by fmtpop;
run;

ods html5 close;
```

The SAS System

North and South American Countries by population

[Small populations](#)

[Medium populations](#)

[Large populations](#)

[Huge populations](#)

The SAS System

fmlpop=Small

GLC Country Name	GLC Country Name	Population (2005)	Gross National Income per Capita (PPP Int.\$ 2004)
SAINT KITTS/NEVIS	SAINT KITTS/NEVIS	42,696	11190
DOMINICA	DOMINICA	78,940	5250
ANTIGUA/BARBUDA	ANTIGUA/BARBUDA	81,485	10360
GRENADA	GRENADA	102,924	7000
SAINT VINCENT/GRENADINES	SAINT VINCENT/GRENADINES	119,051	6250
ST. LUCIA	ST. LUCIA	160,765	5560
BARBADOS	BARBADOS	269,556	15060
BELIZE	BELIZE	269,736	6510
BAHAMAS	BAHAMAS	323,063	16140
SURINAME	SURINAME	449,238	.
GUYANA	GUYANA	751,218	4110

fmlpop=Medium

GLC Country Name	GLC Country Name	Population (2005)	Gross National Income per Capita (PPP Int.\$ 2004)
TRINIDAD AND TOBAGO	TRINIDAD AND TOBAGO	1,305,236	11180
JAMAICA	JAMAICA	2,650,713	3630
PANAMA	PANAMA	3,231,502	6870
URUGUAY	URUGUAY	3,463,197	9070
COSTA RICA	COSTA RICA	4,377,778	9570

Figure 1 - Header Links before PROC PRINT Output

LOW VISION

Low vision is a wide spread category that affects large amounts of people at different times. This can be a form of progressive decrease, such as losing near vision with age, or a temporary situation such as using a phone outside with increased glare causing the inability to easily see the screen. Color blindness, or more accurately, color deficiencies, can prevent you from distinguishing between different colors.

SECTION 508

- 1194.21(c) A well-defined on-screen indication of the current focus shall be provided that moves among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that assistive technology can track focus and focus changes.
- 1194.21(g) Applications shall not override user selected contrast and color selections and other individual display attributes.

- 1194.21(i) Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.
- 1194.21(j) When a product permits a user to adjust color and contrast settings, a variety of color selections capable of producing a range of contrast levels shall be provided.
- 1194.22(c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.

WCAG 2.0

- 1.4.1 Use of Color (1194.21(i), 1194.22(c))
- 1.4.3 Minimum Contrast (1194.21(g), 1194.21(j))
- 1.4.4 Resize Text
- 1.4.6 Contrast (Enhanced) (AAA)
- 2.4.7 Focus Visible (1194.21(c))

Use of Color

Consider the following:

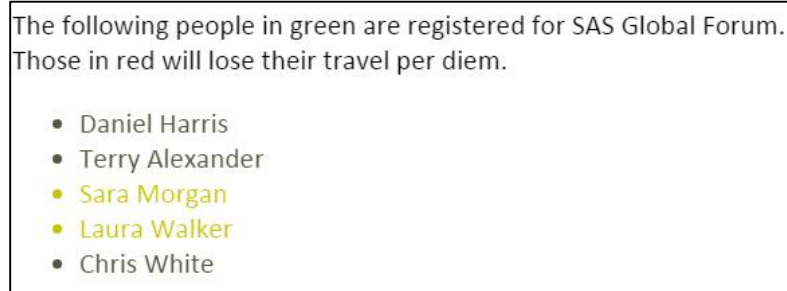


Figure 2 - Screenshot of Color Text Simulating Deuteranopia

The image above simulates deuteranopia, a form of color blindness. Since color was the only distinguishing factor, it is impossible to tell which names are in trouble. You can tell there's a difference but you don't know which color is which. Sara and Laura look greenish. Here is the same image with normal vision.

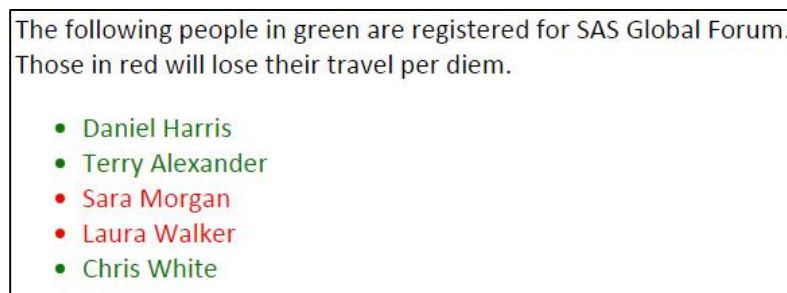


Figure 3 - Screenshot of Color Text with Normal Vision

By adding an italic font or adding additional text, you can make the list page visible for everyone.

The following people in green are registered for SAS Global Forum.
Those starred in red will lose their travel per diem.

- Daniel Harris
- Terry Alexander
- *Sara Morgan (*)*
- *Laura Walker (*)*
- Chris White

Figure 4 - Screenshot of Color Text Simulating Deuteranopia with Additional Visual Cues

Now, even though color is absent, you can tell the font is different (italic) and the extra asterisk after their names.

You don't need an eye deficiency to have trouble with color-only reports. What if you print a report on a black and white printer? You might be reading this paper in black and white and can't tell the difference in the above images. The extra text or font changes can help everyone.

The following people in green are registered for SAS Global Forum.
Those starred in red will lose their travel per diem.

- Daniel Harris
- Terry Alexander
- *Sara Morgan (*)*
- *Laura Walker (*)*
- Chris White

Figure 5 - Screenshot of Color Text in Black and White with Additional Visual Cues

If you are generating a table of information and are color coding the text based on values by using ODS styles (often called "trafficlighting"), make sure your styles include additional attributes such as bold or italic, a different font size, or extra text.

Here's a simple table with the age value colored using a format:

```
proc format;  
  value trafficlight  
    low-<14=Red;  
run;  
  
proc print data=sashelp.class;  
  var name sex  
    age / style={foreground=trafficlight.};  
run;
```

Obs	Name	Sex	Age
1	Alfred	M	14
2	Alice	F	13
3	Barbara	F	13
4	Carol	F	14
5	Henry	M	14
6	James	M	12
7	Jane	F	12

Figure 6 - Screenshot of Table Output with Red Text

If the values in red need to be distinguished from the others, you need more than just a color. This is how the output looks for those with protanopia.

Obs	Name	Sex	Age
1	Alfred	M	14
2	Alice	F	13
3	Barbara	F	13
4	Carol	F	14
5	Henry	M	14
6	James	M	12
7	Jane	F	12

Figure 7 - Screenshot of Table Output with Red Text Simulating Protanopia

You can add italics, bold, and underlining to make it more distinguishable:

```
proc format;
  value trafficlight
    low-<14=Red;
run;

proc format;
  value italics
    low-<14=italic;
  value bold
    low-<14=bold;
  value underline
    low-<14=underline;
run;

proc print data=sashelp.class;
  var name sex
    age / style={foreground=trafficlight. font_style=italics.
font_weight=bold. textdecoration=underline.};
run;
```


Obs	Name	Sex	Age
1	Alfred	M	14
2	Alice	F	<u>13</u>
3	Barbara	F	<u>13</u>
4	Carol	F	14
5	Henry	M	14
6	James	M	<u>12</u>
7	Jane	F	<u>12</u>

Figure 8 - Screenshot of Table Output with Red Text Simulating Protanopia with Additional Visual Clues

This same technique should be used when creating graphs. If you have a bar chart, scatter plot, or line plot with several series, make sure the bars, symbols, or lines of the series have more than just a color change. Use various shading styles or symbols.

Consider the following simple scatter plot:

```
goptions hsize=3 vsize=3;
proc gplot data=sashelp.class;
    plot height*weight=sex;
run;
```

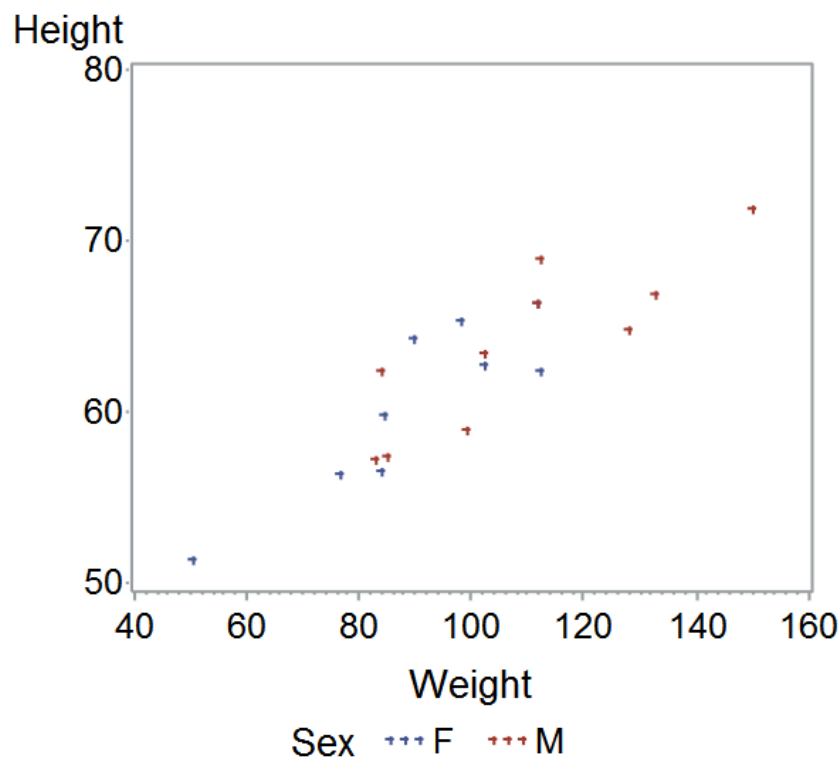


Figure 9 - Output from PROC GPLOT

Even with good vision, the default color between the two series is difficult to determine, perhaps because the symbols are small. By choosing different symbol shapes for each series, making them bigger, and using more contrasting colors, the graph becomes more readable for everyone. And when it is printed in black and white, you can still determine which symbol is which.

A few other changes made to the graph are:

- smaller horizontal and vertical size for the graph (for the purposes of fitting the graph in this paper but has the side benefit of causing the text to be larger)
- full text words in the legend (which can help with cognitive issues)

```
symbol1 value=squarefilled color=cx00ff00 /* red */ height=3;  
symbol2 value=trianglefilled color=cx00ffff /* cyan */ height=3;  
legend1 label=none value= ("Female" "Male");  
goptions hsize=3 vsize=3;  
  
proc gplot data=sashelp.class;  
    plot height*weight=sex / legend=legend1;  
run;
```

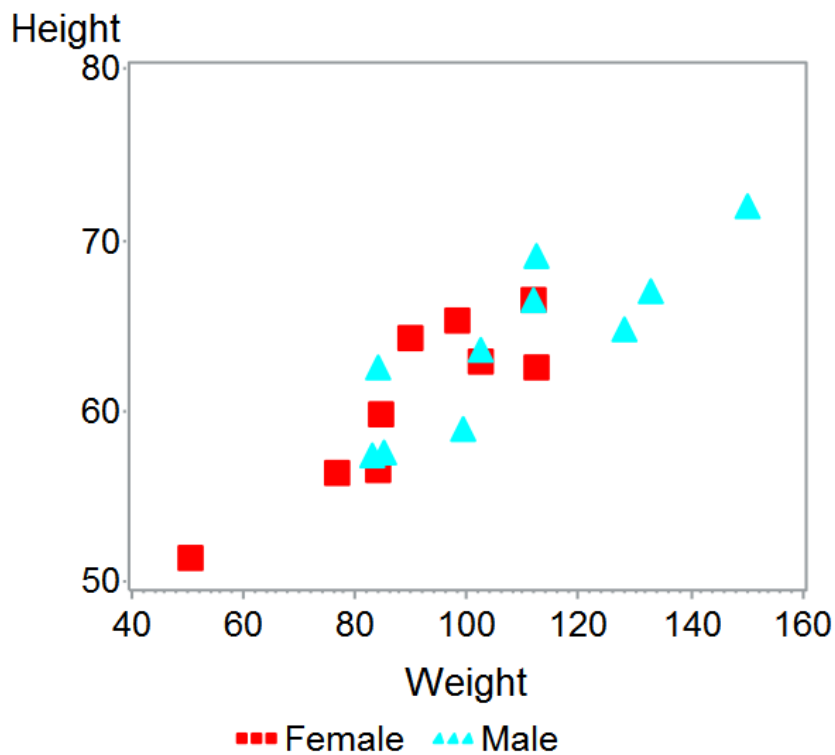


Figure 10 - Output from PROC Gplot with Larger Symbols and Contrasting Colors

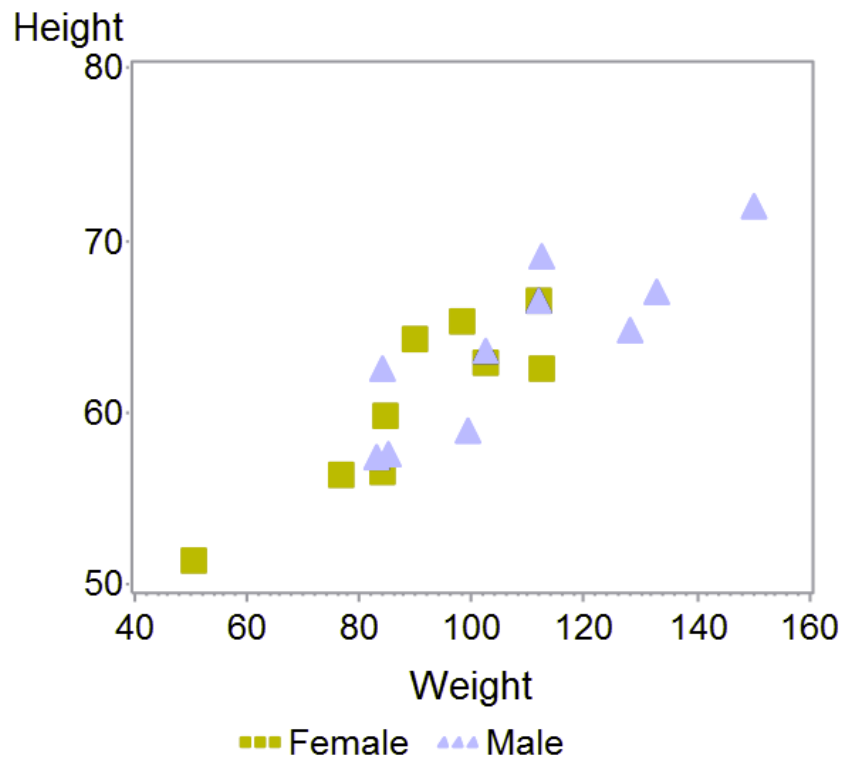


Figure 11 - Output from PROC GGPLOT with Larger Symbols and Contrasting Colors Simulating Deuteranopia

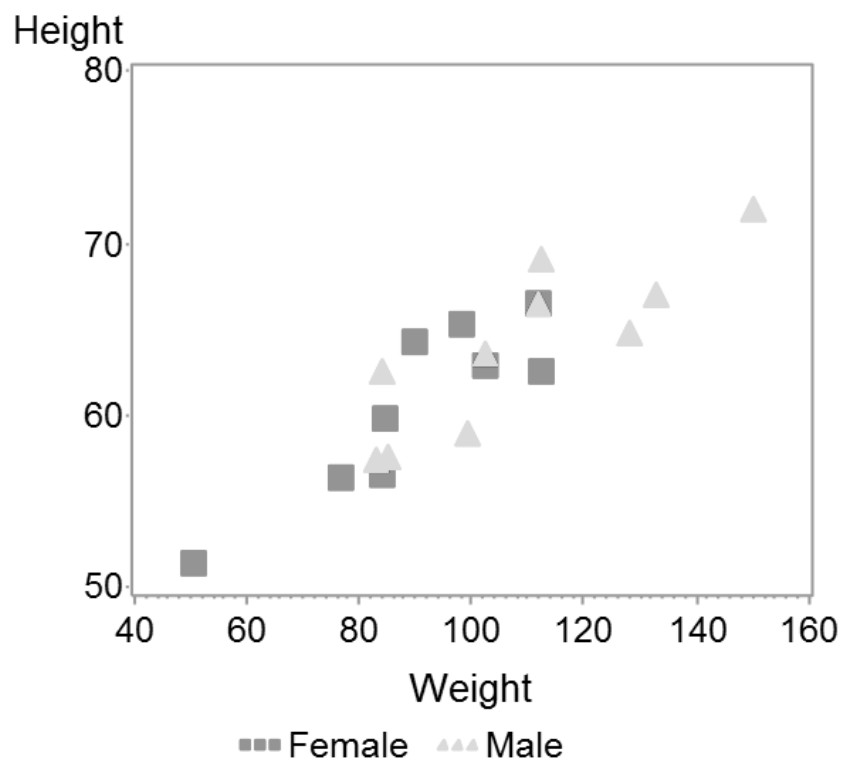


Figure 12 - Output from PROC GGPLOT with Larger Symbols and Contrasting Colors Simulating Black and White

Minimum Contrast

Contrast is the ratio between the foreground, or text color, and the background color. The ratio ranges from 1 (when the foreground and background colors are the same) to 21 (black and white). The higher the contrast, the easier it is to see. The formula to compute the ratio is rather complex and is included in [Appendix 3](#), but using a third-party tool to compute the ratio is recommended.

You might have noticed a trend over the past few years for websites to use gray text on a white background because it is "softer" on the eyes.



Figure 13 - Screenshot of support.sas.com Showing Text in Lower Right with Low Contrast.

Unfortunately, this softness is because the color contrast ratio is small. The minimum contrast ratio should be 4.5:1. In the screenshot above, the ratio is 2.85:1.

When using larger text, at least 14pt and bold, or anything larger than 18pt, the contrast ratio can be lower, 3:1. Figure 13 still doesn't satisfy that requirement.

Most of the built in ODS styles (for example, HTMLBlue [the default in SAS 9.4]), have sufficient contrast ratios. If you create your own style, or if you have a corporate style, you need to use a color contrast checker to verify that the ratio is sufficient. One example of a checker is from WebAIM, <http://webaim.org/resources/contrastchecker/>. You can find other tools by searching for "color contrast checker". Using WebAIM's tool, you can darken the text color until the ratio is greater than 4.5:1. Here's the same screenshot from support.sas.com after clicking on the "darken" button until the ratio reaches 5.41:1. It's subtly darker and a little easier to read.

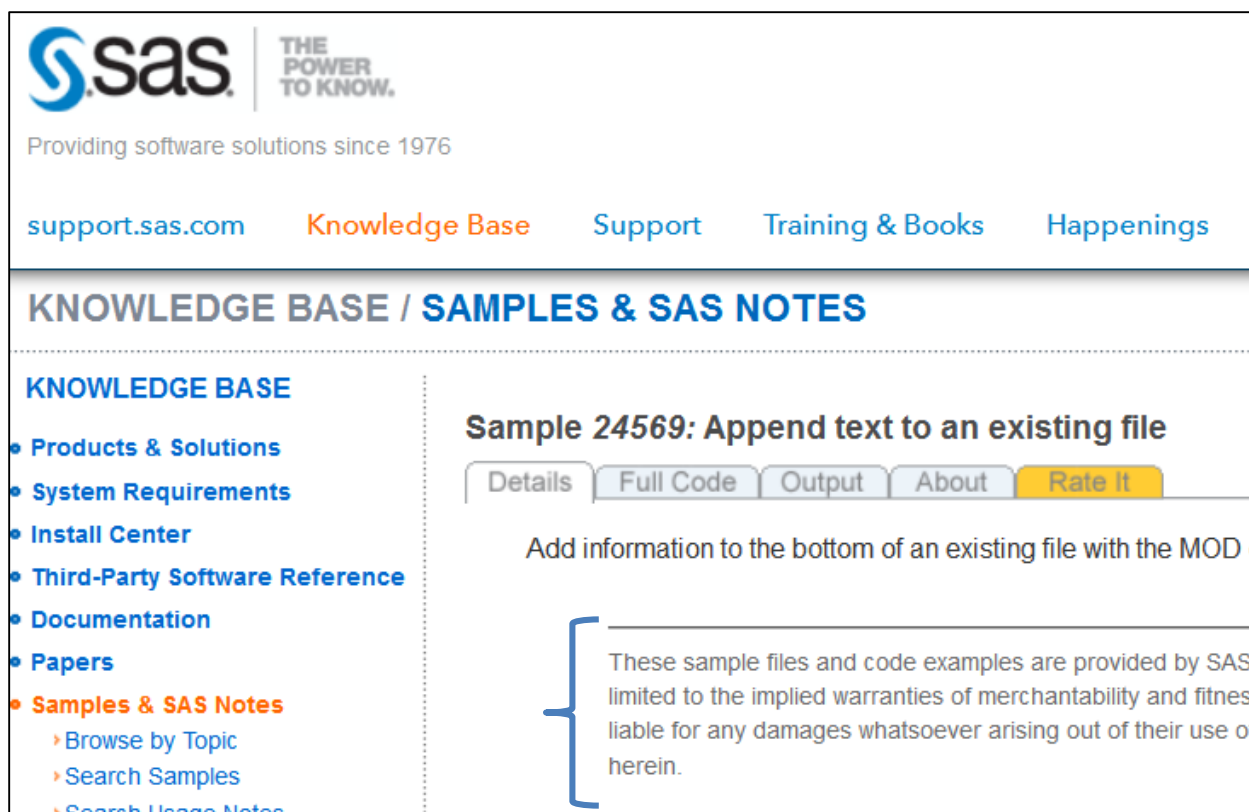


Figure 14 - Screenshot of support.sas.com Showing Text in Lower Right with Modified Higher Contrast.

The blue text in the navigation bar on the left has a ratio of 5.57, which is good. The orange text, which is the currently selected navigation item, "Samples & SAS Notes", has a contrast of 2.94, which fails the criterion. If the text were larger, it would be very close to the 3:1 ratio.

The HTMLBlue style in SAS 9.4 has a contrast ratio of 12.29:1 for the table headers (dark bold blue on a light blue background) and a ratio of 21:1 for the table cells (black text on a white background). 21 is the maximum ratio.

The SAS System					
Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5

Figure 15 - HTMLBlue style in SAS 9.4

The Default style in SAS 9.2 uses blue or black text on a gray background. The ratio for the table headers is 4.71:1 and for the table cells is 14.03:1. Both pass the minimum requirement.

<i>The SAS System</i>					
Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5

Figure 16 - Default Style in SAS 9.2

Contrast (Enhanced)

WCAG guideline 1.4.6 Contrast (Enhanced) is a AAA standard. Websites that are following the WCAG AA standard don't have to include this criterion. However, this criterion is easy to comply with using some of the standard ODS styles. To achieve this guideline, the color contrast ratio must be at least 7:1. The default HTMLBlue style in SAS 9.4 satisfies that. There are also two high contrast ODS styles, HighContrast and HighContrastLarge. They use white text on a black background, which is the maximum contrast ratio of 21:1.

```
ods html5 file='foo.html' style=highcontrast;
proc print data=sashelp.class; run;
ods html5 close;
```

<i>The SAS System</i>					
Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5

Figure 17 - HighContrast Style

Sometimes yellow or cyan text on a black background is also a good high contrast color. Here's a simple ODS style created from PROC TEMPLATE (the details of which can be found in the ODS documentation).

```
proc template;
  define style HighContrastYellow;
    parent = styles.HighContrast;
    style color_list from color_list /
      'fgA' = yellow /* title */
      'fgA1' = yellow /* data in the table */
      'fgA2' = cyan; /* row and column headers */
  end;
run;

ods html5 file='foo.html' style=highcontrastyellow;
proc print data=sashelp.class; run;
ods html5 close;
```

The SAS System					
Obs	Name	Sex	Age	Height	Weight
1	Alfred	M	14	69.0	112.5
2	Alice	F	13	56.5	84.0
3	Barbara	F	13	65.3	98.0
4	Carol	F	14	62.8	102.5

Figure 18 - Custom HighContrast Yellow and Cyan

You can see [a sample of all the styles available for HTML](#) by doing a search for "ODS Styles Gallery". Most search engines return the first result as a link to the "SAS(R) 9.4 Output Delivery System: User's Guide, Fourth Edition".

Resize Text

WCAG guideline, 1.4.4 Resize Text, means your output should be readable and usable after the text size is increased. Resizing text is handled in the browser and is typically available from the View menu or by using the shortcut keys Ctrl++, Ctrl+-, and Ctrl+0 (zero). When text is resized, it should reflow, if necessary, and objects should not overlap each other.

The following image is from a color blindness checker on <http://www.color-blindness.com/>. The first image is viewing the page at normal text size. Everything is visible.

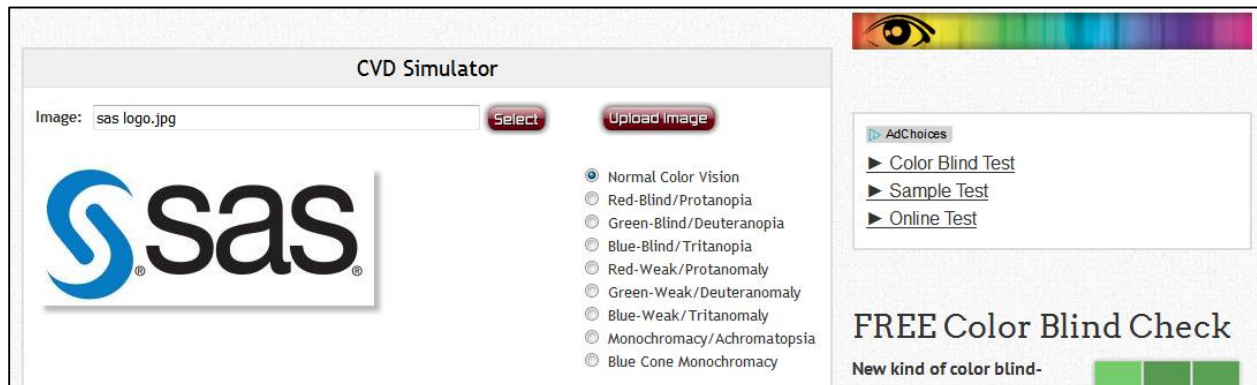


Figure 19 - Normal Text Size - <http://www.color-blindness.com/coblis-color-blindness-simulator>

After resizing the text once, the label for the second to last radio button, "Monochromacy/Achromatopsia", becomes separated from the button itself.

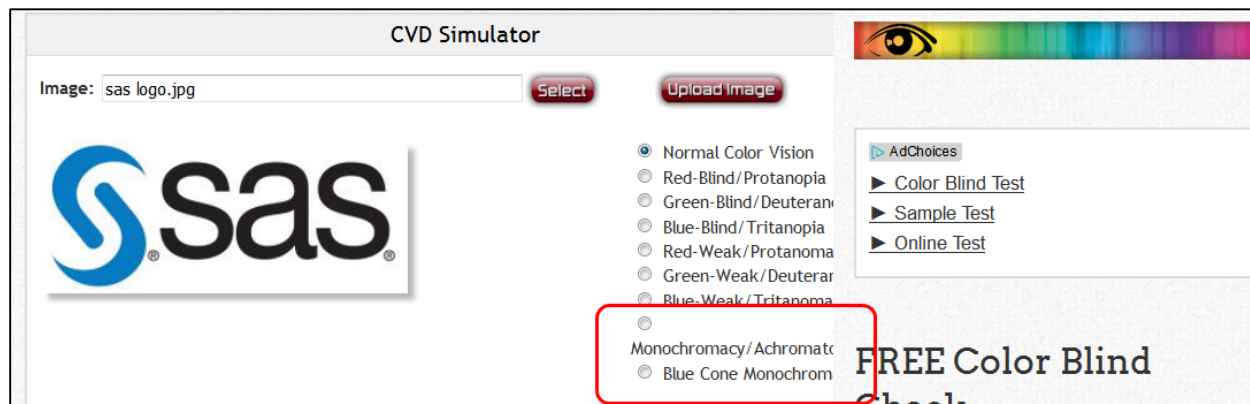


Figure 20 - One Magnification Text Size

When I first visited this page, I had resized the text before looking at the page so I didn't notice what it looked like at normal size. After resizing, then I read the page. I thought there was a list of radio buttons with the last one missing a label and then there was a header (Monochromacy/Achromatopsia) before a separate radio button (Blue Cone Monochromacy). I wasn't sure what made the Blue Cone Monochromacy radio button special that it needed its own header. It wasn't until I viewed the HTML source that I realized it was a text flow problem.

After resizing larger several more times, the text for the radio buttons becomes clipped and unreadable. The page becomes unusable to a low vision user that needs text magnified.

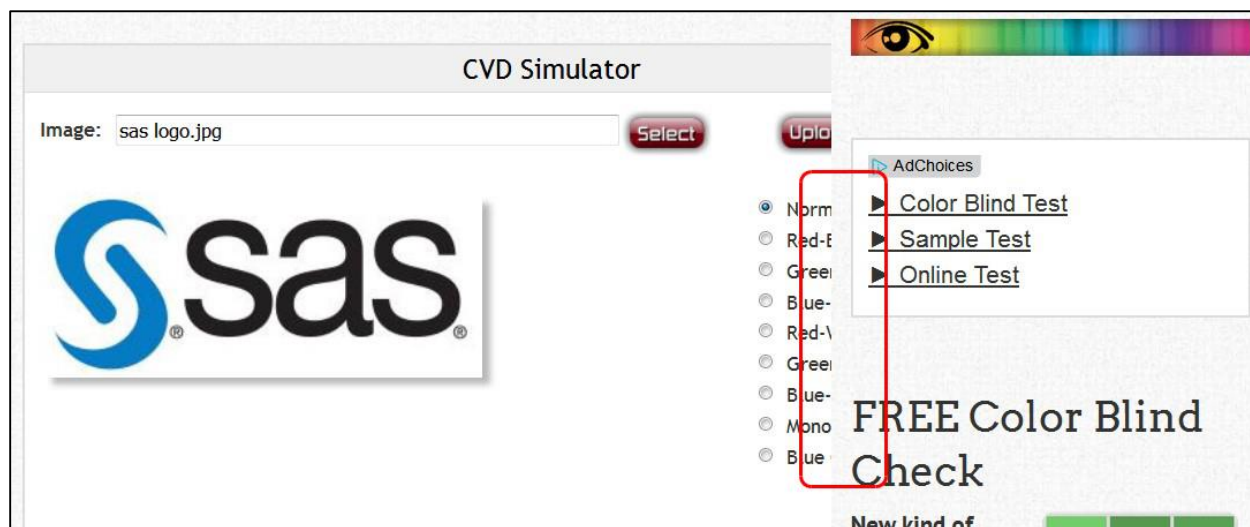


Figure 21 - Several Magnification Text Size

ODS output should not have this problem. Text and graphics should magnify correctly. But if you manually add items to your page, make sure you don't have fixed locations or sizes for HTML elements so that they will reflow properly.

Resize Graphics

When text is magnified, the corresponding graphics on the page are also magnified. Since images tend to get grainy and pixelated as they're magnified, the SVG driver for ODS graphics is recommended rather than the PNG driver. SVG is scalable so that when the page is magnified, the graph will redraw using vectors and keep its clean look. The SVG driver is the default when using the HTML5 destination.


```
ods html5 file='foo.html';
/* PNG will be grainy when magnified */
options device=png;
proc gchart data=sashelp.class;
  vbar age/
  sumvar=height type=mean discrete;
run;quit;
ods html5 close;
```

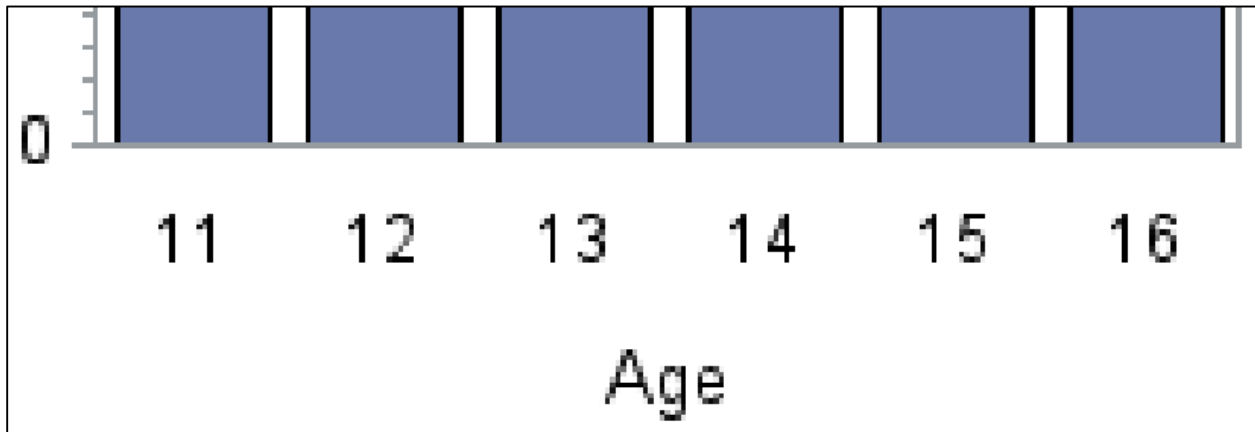


Figure 22 - Magnified PNG Output That's Pixelated

```
ods html5 file='foo.html';
/* device=svg shouldn't be necessary for the HTML5 destination because it's
the default device, but might be required if there was a previous goptions
device setting (or you can reset the device using "goptions device=;") */
options device=svg;
proc gchart data=sashelp.class;
  vbar age/
  sumvar=height type=mean discrete;
run;quit;
ods html5 close;
```

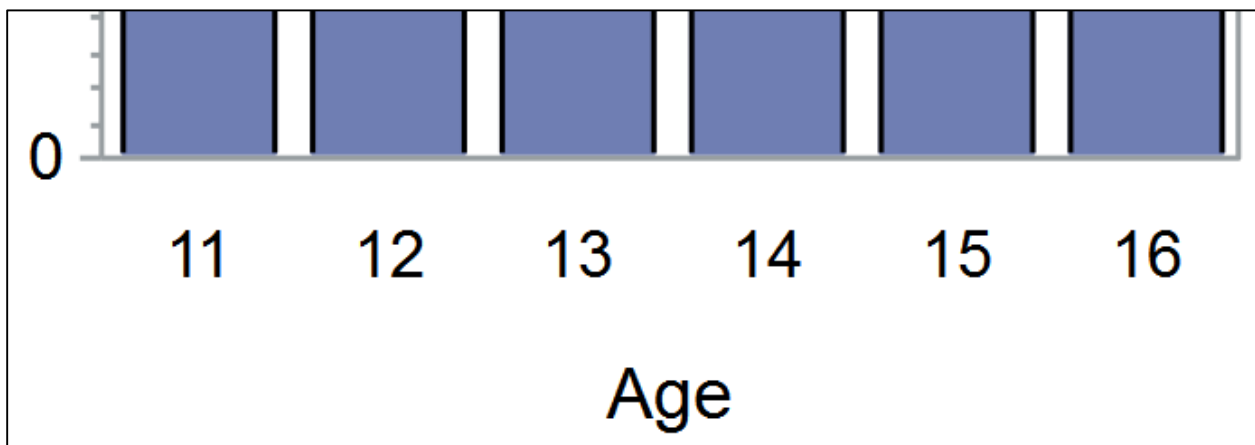


Figure 23 - Magnified SVG Output That's Clean

When using the SVG driver, additional code is generated that displays a large tooltip as your mouse floats over the graph.

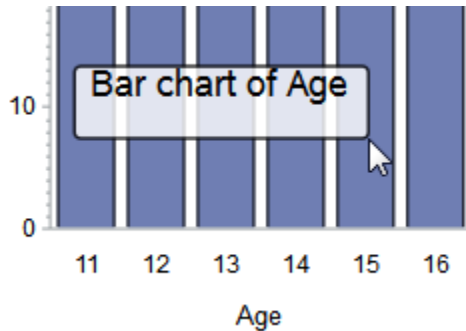


Figure 24 - Tooltip with SVG Driver

The tooltip code can be turned off using the SVGNOTIP option.

```
options dev=sasprtc printerpath=SVGNOTIP;  
ods html5 file='foo.html';  
proc gchart data=sashelp.class;  
  vbar age/  
    sumvar=height type=mean discrete;  
run;quit;  
ods html5 close;
```

To re-enable it later in your report, use

```
options dev=sasprtc printerpath=SVG;
```

Focus Visible

When navigating a website using the keyboard, the element that has focus should have a visual indicator, usually a box drawn around the element. It is the browser's responsibility to draw this focus indicator. ODS output doesn't affect the focus indicator. However, sometimes a browser's indicator is not very obvious.



Figure 25 - Default Focus Indicator Barely Noticeable Around "Medium populations"

You can include a CSS style in your output if you want a custom focus indicator. The code below sets the outline to thick (4 pixel) green outline for all elements.

```
proc odstext contents='';
```

```
p "<style>*:focus { outline: #00ff00 solid 4px; }</style>";  
run;
```



Figure 26 - Custom Green Focus Indicator

NO VISION

The items in the No Vision category assume that the user can't see the content on screen at all. These criteria aim to provide ways for the user to understand and interact with the report. Assistive technology such as a screen reader or braille device are typically required. The majority of Section 508 criteria fall into this category.

SECTION 508

- 1194.21(d) Sufficient information about a user interface element including the identity, operation and state of the element shall be available to assistive technology. When an image represents a program element, the information conveyed by the image must also be available in text.
- 1194.21(l) When electronic forms are used, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.
- 1194.22(a) A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).
- 1194.22(d) Documents shall be organized so they are readable without requiring an associated style sheet.
- 1194.22(f) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.
- 1194.22(g) Row and column headers shall be identified for data tables.
- 1194.22(h) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.
- 1194.22(i) Frames shall be titled with text that facilitates frame identification and navigation.
- 1194.22(k) A text-only page with equivalent information or functionality shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.
- 1194.22(n) When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

WCAG 2.0

- 1.1.1 Non-text Content (1194.22(a), 1194.22(k))
- 1.3.1 Info and Relationships (1194.22(d))
- 1.4.5 Images of Text
- 2.4.2 Page Title (1194.22(i))
- 2.4.6 Headings and Labels (1194.22(g), 1194.22(h))
- 3.1.1 Language of Page
- 3.1.2 Language of Parts
- 4.1.2 Name, Role, Value (1194.21(d), 1194.21(l), 1194.22(n))

Non-text Content

If you manually insert an image into your ODS output (tag) and that image has meaning for the content, you must specify alternate text for that image. The text should convey the meaning of the image. If the image is for decoration or spacing, then it should be ignored by screen readers. Alternate text for tags is specified using the alt= attribute.

```
<img src='skislopes.jpg' alt='map of ski slopes'>
```

When generating graphs, use the description= option to specify your alternative text:

```
proc gplot data=sashelp.class;  
    plot height*weight=sex / description="Scatter plot of students' heights  
graphed against weights";  
run;
```

If you use device=png, the description is used as the value of the alt= option on the generated tag. On some browsers, the alt text is displayed as a tooltip when you hover the mouse over the image.

```

```

If you use device=svg (the default for HTML5), the description is used in the <title> tag nested in the generated <svg> tag.

```
<svg xmlns="http://www.w3.org/2000/svg"...>  
<title>Scatter plot of students' heights graphed against weights  
</title>
```

Note: On some browsers, and with some screen readers, the <text> elements of the SVG are read in addition to the description.

The description is also used in the table of contents (if you use ods contents=) and in the Results window in SAS.

Alternate Reports

If you create a report that contains only graphs, the same data needs to be provided in text format, usually tables. For example, if you have a bar chart:

```
proc gchart data=sashelp.class;  
    vbar age / discrete description='Bar chart of frequency of ages';  
run;
```

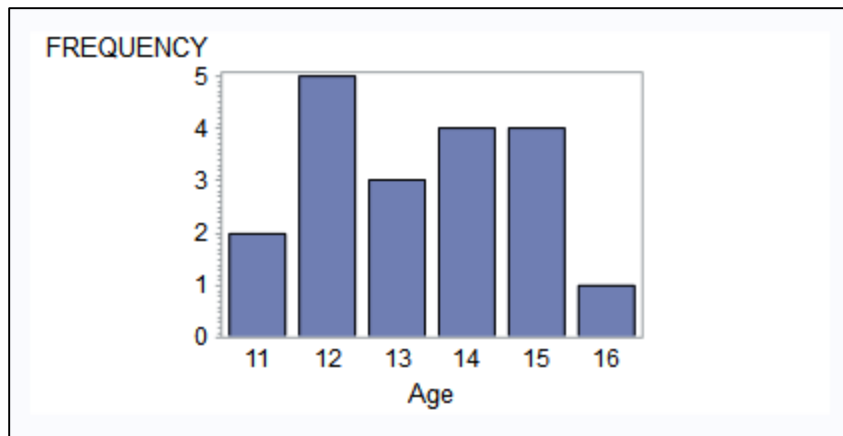


Figure 27 - Output of PROC GCHART

You should also provide a table of the same data:

```
proc freq data=sashelp.class;
  title 'Table of frequency of ages';
  tables age / nocum nopercnt;
run;
```

Table of frequency of ages

Age	Frequency
11	2
12	5
13	3
14	4
15	4
16	1

Figure 28 – Table Output of PROC FREQ

This table can exist on the same page as the graph or you can have a link to the table that opens in a separate page. If the table is on the same page, you should indicate that in the description of the graph.

```
ods html5 file='foo.html' (title='Bar chart of frequency of ages');
proc gchart data=sashelp.class;
  vbar age / discrete description='Bar chart of frequency of ages. A
table of values follows the chart.';
run;

ods noproctitle;
proc freq data=sashelp.class;
  title 'Table of frequency of ages';
  tables age / nocum nopercnt;
run;

ods html5 close;
```

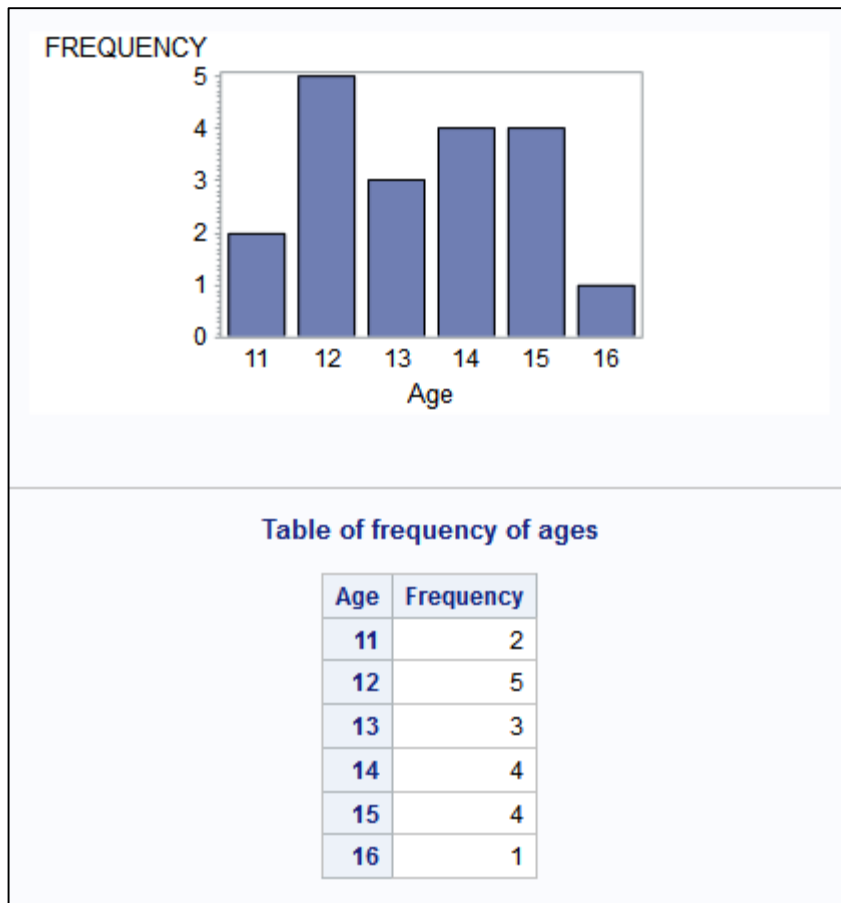


Figure 29 - Output of PROC GCHART and PROC FREQ on Same Page

To generate a table that is accessed via a link, you can use PROC ODSTEXT to add the link to your output.

```
ods html5 file='foo.html' (title='Bar chart of frequency of ages');
proc gchart data=sashelp.class;
  vbar age / discrete description='Bar chart of frequency of ages. A
link to a page that contains a table of values follows the chart.';
run;

proc odstext;
  P "<a href='external_table.html' target='_blank'>Table of values for
the Bar chart of frequency of ages</a>";
run;

ods html5 close;

ods html5 file='external_table.html' (title='Table of values for Bar chart of
frequency of ages');
ods noproctitle;
proc freq data=sashelp.class;
  title 'Table of frequency of ages';
  tables age / nocum nopercnt;
run;

ods html5 close;
```

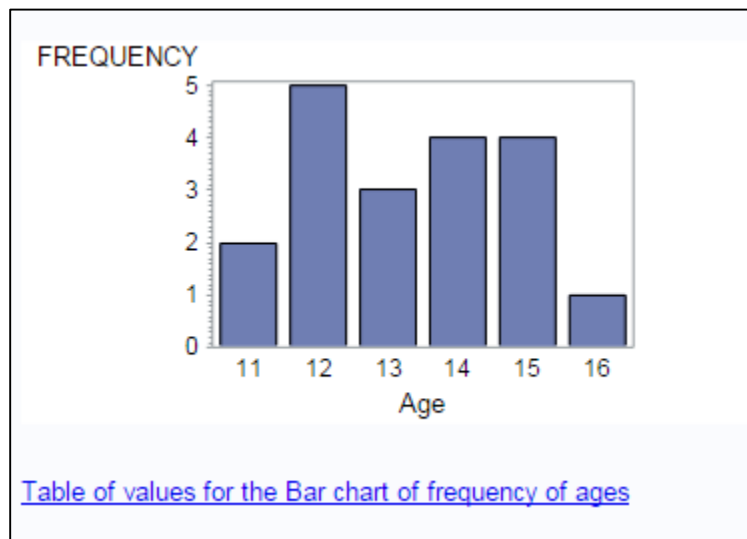


Figure 30 - Output of PROC CHART with a Link to an External Table

1194.22(f) Image Maps

1194.22(f) says that client-side image maps should be used instead of server-side image maps and the links should have alternate text to satisfy 1194.22(a). You can satisfy this criterion by creating a new variable on the data set that contains the alternate text. In the example below, in addition to ALT=, a TITLE= is specified, which displays a tooltip when you hover with the mouse, and a dummy HREF= is specified for the link. A real map would have an HREF= that points to a table or another map with additional information. The ALT= allows the screen reader to read the name of the state when you tab through the image map.

```
/* create a dataset of states with an additional variable called mylink */
data mystate;
    set maps.us2;
    mylink="alt='" || trim(statename) || "' title='" || trim(statename) || "'
href='#'";
run;

/* sort the maps dataset in reverse alphabetical order; when a map is
   created, the states will be drawn in reverse order but then the image map
   links will be drawn in alphabetical order */
proc sort data=maps.us out=myus; by descending state; run;

goptions device=png;
ods html5 file="foo.html" ;

/* chrome will honor the outline color of an image map area */
proc odtext; p '<style>area {outline-color:red}</style>';

/* create a map with each state an imagemap link */
proc gmap map=myus data=mystate;
    id state;
    choro state / html=mylink name="mystate"
                  description="Map of the United States";
run;
```

```
ods html5 close;
```

When running the example with a screen reader, the screen reader can display a list of links on the map. The ALT= text is used to populate this list and makes it very fast to jump to the right state.

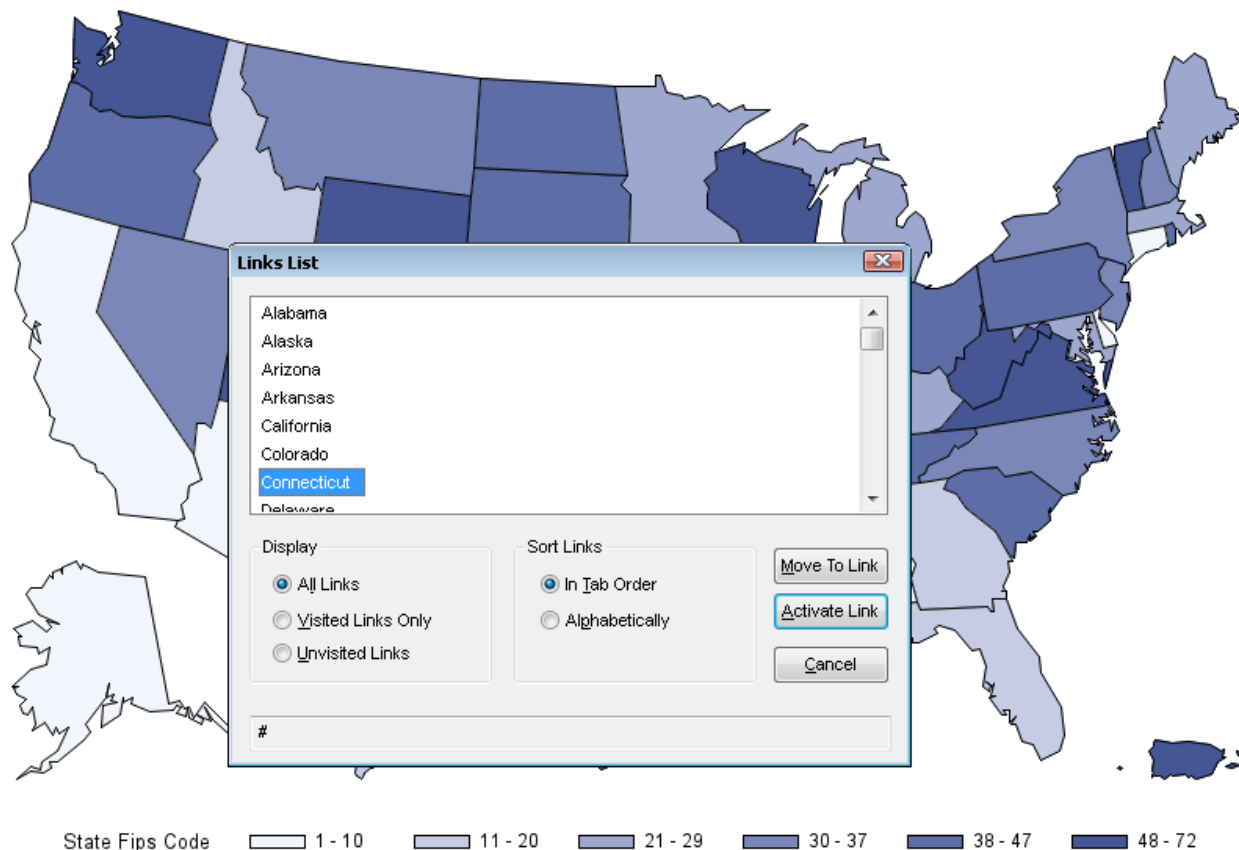


Figure 31 - Links List Dialog Box in JAWS Screen Reader Showing PROC GMAP Image Map Links

Info and Relationships

All content that has implied meaning from its physical layout or appearance must also have its meaning conveyed to assistive technology. For example, if you are using style sheets to format text to be large and bold to imply a header, that same information needs to be conveyed to the screen reader. This is typically done by using the header tags in HTML, such as <h2>.

When you generate several procedure outputs in a row, by default, each output has a "The SAS System" header in front of it. Some PROCs also print a second header with the name of the PROC. Both of these headers violate the Info and Relationships criterion because they're only styled to look like headers. They're not real headers.

```
proc freq data=sashelp.class; tables age / nocum nopercnt; run;
proc means data=sashelp.class; var age; run;
```

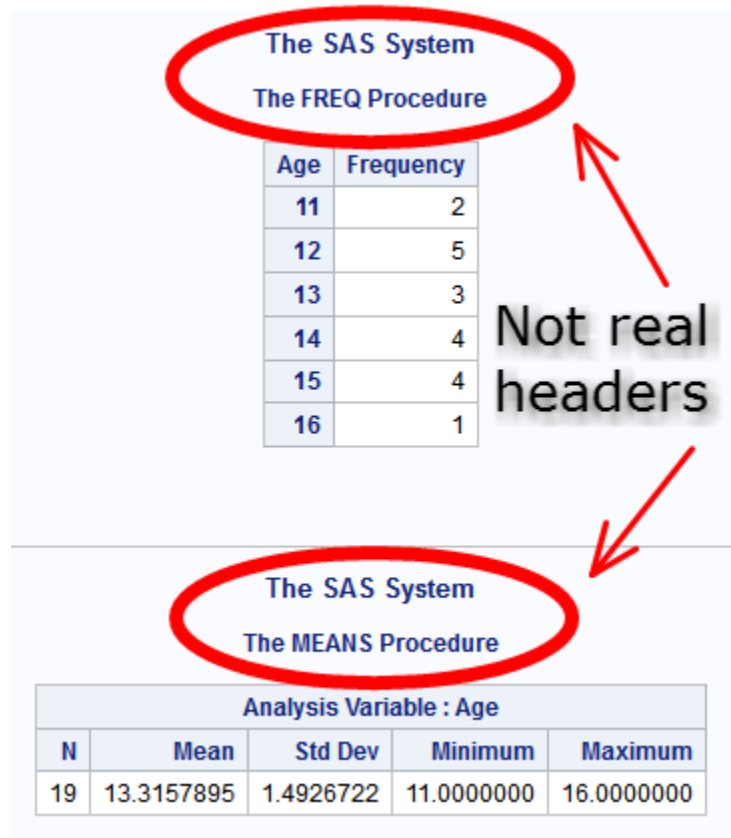



Figure 32 - Output from PROC FREQ and PROC MEANS with Fake Headers

```
<div id="IDX" class="systitleandfootercontainer" style="border-spacing: 1px">
<p><span class="c systemtitle">The SAS System</span> </p>
</div>
<div class="proc_title_group">
<p class="c proctitle">The FREQ Procedure</p>
</div>
```

To work around this problem, you can insert HTML text between the PROCs using PROC ODSTEXT. You'll also want to turn off "The SAS System" header (via the TITLE statement) and the PROC header (via ODS NOPROCTITLE statement).

Some ODS styles print a horizontal line (<hr> HTML tag) between procedure outputs. That line can be hidden by adding CSS with PROC ODSTEXT.

```
ods html5 file='foo.html';

/* turn off the title, PROC titles, and hide the horizontal line */
title;
ods noproctitle;
proc odstext contents=''; p "<style>hr {display:none}</style>"; run;

/* insert a header before PROC FREQ */
proc odstext contents='' ; p "<h1>My output from PROC FREQ</h1>"; run;
proc freq data=sashelp.class; tables age / nocum nopercnt; run;

/* insert a header before PROC MEANS*/
proc odstext contents='' ; p "<h1>My output from PROC MEANS</h1>"; run;
```

```
proc means data=sashelp.class; var age; run;

ods html5 close;
```

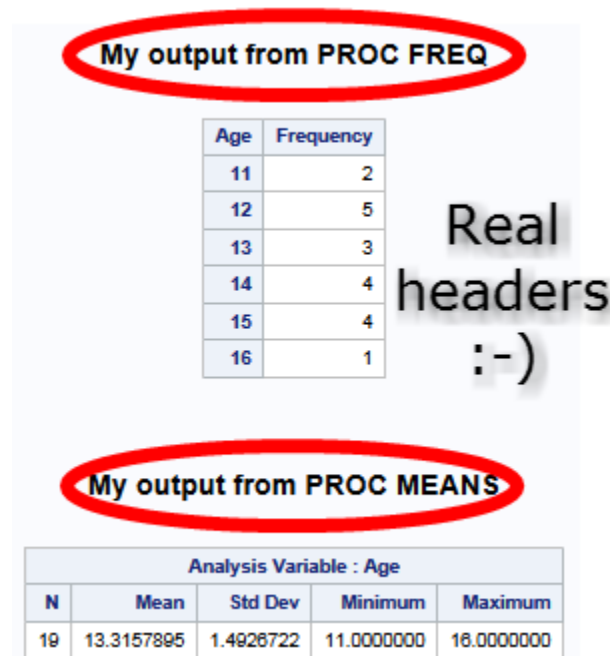


Figure 33 - Output from PROC FREQ and PROC MEANS with Proper Headers

Visually, the output doesn't look much different. The horizontal lines are gone, and the header text before each PROC is a little bigger, but it is essentially the same. However, to a screen reader, it is a huge difference. The headers are real and will be announced properly and will aid in navigation.

```
<p class="paragraph"><h1>My output from PROC FREQ</h1></p>
```

One caution about using PROC ODSTEXT. If the string does not begin and end with HTML tags, then ODS will encode the string so that it appears exactly as is in the output. Encoding is when characters that have special meaning are turned into HTML entities. For example, "<" will be changed to "<" so that a less than sign will be displayed in the output.

```
proc odstext; p "<a href='file.html'>This will generate a link</a>"; run;
proc odstext; p "This will <a href='file.html'>not generate</a> a link"; run;
```

The first example has an anchor tag (<a>) at the beginning and end of the string, so it will create output that looks like a link with all the text in blue and underlined:

[This will generate a link](#)

The second example doesn't have HTML tags at the beginning and end. The desired output is to have "not generate" look like a link:

This will [not generate](#) a link

But what you'll get is

This will not generate a link

Which is exactly the string used in PROC ODSTEXT. If you look at the generated HTML, it looks like

```
<p class="paragraph">This will &lt;a href=&apos;file.html&apos;&gt;not
generate&lt;/a&gt; a link</p>
```

You can see "[Problem Note 13433: The Protectspecialchars= option is not honored when specified within the SystemTitle and SystemFooter elements](#)" for additional details.

There are several workarounds.

1. Wrap the string in a tag that doesn't affect the display, such as

```
proc odstext; p "<span>This will <a href='file.html'>not generate</a> a
link<span/>"; run;
```

2. Use the raw function in combination with the ODS ESCAPECHAR option

```
ods escapechar="^";
proc odstext; p "^{raw This will <a href='file.html'>not generate</a> a
link}"; run;
```

3. Create a style that uses the protectspecialchars option

```
proc template;
  define style myHtmlStrings;
    parent = styles.HTMLBlue;
    style doNotEncode /
      protectspecialchars=off;
  end;
run;
ods html5 file='foo.html' style=myHtmlStrings;
proc odstext; p "This will <a href='file.html'>not generate</a> a link"
/ style=doNotEncode;
run;
```

4. Use a data step to write to the file instead of PROC ODSTEXT. See "[Usage Note 23660: How can I append to an ODS HTML output file?](#)" It involves using the NO_TOP_MATTER and NO_BOTTOM_MATTER ODS options and opening the file for update mode (FILENAME MOD)

For tables, there's a relationship between table cells and their respective headers. This can be conveyed by using the <th> table header tag and specifying whether the header is for a row (scope='row') or column (scope='col').

```
<th scope="col">Name</th>
<th scope="row">Alice</th>
```

ODS automatically generates the proper tags when tables are generated.

```
ods html5 file='foo.html';
proc print data=sashelp.class; id name; run;
ods html5 close;
```

The SAS System				
Name	Sex	Age	Height	Weight
Alfred	M	14	69.0	112.5
Alice	F	13	56.5	84.0
Barbara	F	13	65.3	98.0

Figure 34 – Table Output from PROC PRINT

```

<thead>
  <tr>
    <th class="l header" scope="col">Name</th>
    <th class="l header" scope="col">Sex</th>
    ...
  </tr>
</thead>
<tbody>
  <tr>
    <th class="l rowheader" scope="row">Alfred</th>
    <td class="l data">M</td>
    ...
  </tr>
  ...

```

Notice that ODS is styling the headers by using a CLASS= attribute. That style is making the text bold. But it is also using the <th> tag and the SCOPE= attribute to convey to the screen reader that this table cell is a column header or row header.

Images of Text

If text is required to be in an image, use the guidelines explained in the previous section, "Non-text Content".

Page Title

Having a page title helps users determine the contents of a page without having to read the whole page. The title is displayed on the tab in the browser and is also read by screen readers when the page is loaded. The default page title is "SAS Output". To change the title, use the TITLE= option on the ODS statement.

```
ods html5 file='foo.html' (title='My really important report');
```

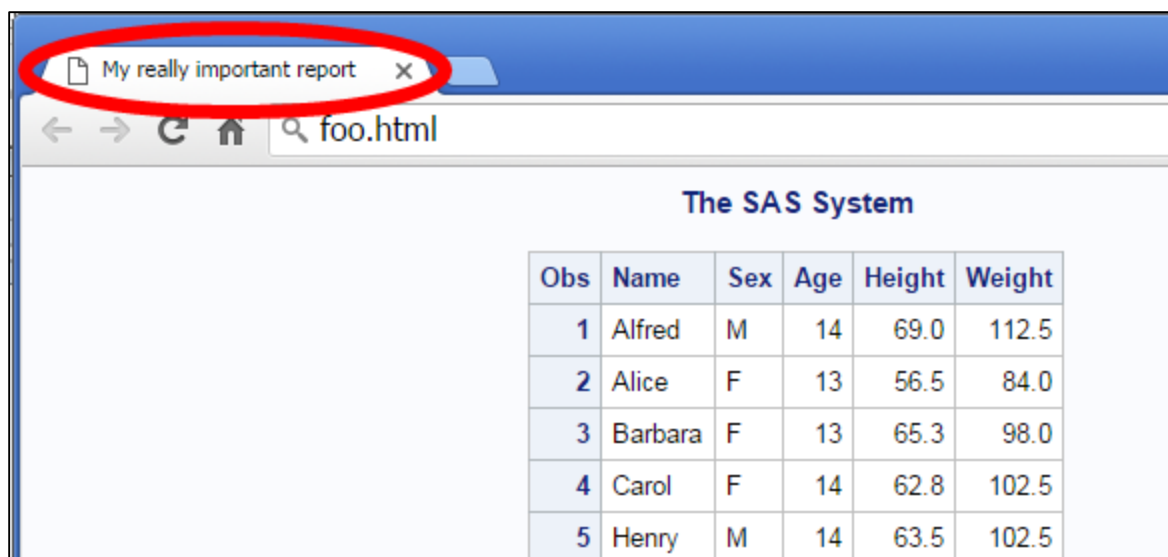


Figure 35 - Title of Report in Browser Tab on Google Chrome

1194.22(i) specifically talks about frames having titles. The <frame> tag is not supported in HTML5. You can still specify the frame= and contents= options on the ODS statement, and the generated HTML will have a <frameset> and <frame> tags, but it is not valid HTML5.

Headings and Labels

If headings and labels are specified, they should be clear and descriptive. Headings were covered in the previous "[Info and Relationships](#)" section. Labels are referring to form elements, which are not generated by ODS by default. If you inject HTML into your report using PROC ODSTEXT and specify form elements, make sure labels are specified for the form elements.

Language of Page/Parts

Having different language content labeled for assistive technology enables assistive technology users to better understand the content being presented, especially if there are multiple languages used on the same page. Many screen readers will switch voices and read the content with the correct pronunciation and accent for the language.

The language can be set on the entire report or page in the initial <html> tag and can also be set on any piece of content that has a different language.

```
<html lang='en'>
<div>Hello</div>
<div lang='es'>Hola</div>
<div lang='fr'>Bonjour</div>
```

Unfortunately, ODS does not set the language on the <html> element and there's no easy(*) workaround. But this issue has already been addressed in a future release.

If you insert your own HTML via PROC ODSTEXT and have languages other than your default, make sure you specify the lang= option on your tag.

(*) You CAN workaround the problem if you generate your own top matter for your ODS output and then append all of your report output to it using the NO_TOP_MATTER option in ODS. Your own top matter will contain the <html lang=> tag as well as other initial HTML tags. You can view a sample HTML file generated from ODS to see what should be contained in the top matter. It's before the <body> tag, which includes the definition of the ODS style. So the workaround is possible, but a bit involved. More details can be provided by contacting the author.

Name, Role, Value

When writing your own HTML, the "name, role, value" criterion is probably the most critical one to get right with respect to assistive technology. When generating HTML from ODS, most of these details are taken care of for you.

When writing HTML, in general you want to use the built in HTML tags. When you want a heading, use <h1>, <h2>, and so on. When you want a button, use <button> or <input type='button'>. Screen readers understand these built in tags. If there's a reason you can't use the built in tags but need to construct the element yourself using and <div> tags, then you must let the screen reader know what type of object you are constructing, whether that object has a value (like a slider), and what the name of the object is. This is done using a variety of HTML attributes, the details of which are worthy of a paper in and of itself.

Here's a simple example of using built in HTML tags:

```
<h2>Breaking news story</h2>
<button>Read more</button>
```

And here's the same example using <div> tags and specifying attributes that assist screen readers:

```
<div role='heading' aria-level='2'>Breaking news story</div>
<div role='button' onclick='handleclick()' onkeydown='handlekey()'
tabindex='0'>Read more</div>
```

If you are injecting your own HTML into the report using PROC ODSTEXT and you are not using the built in HTML tags, then you will need to specify name, role, and value information.

COGNITIVE

Cognitive issues typically affect thought processes. Some examples are:

- Memory
- Problem-solving
- Attention
- Reading, linguistic, and verbal comprehension
- Math comprehension
- Visual comprehension

As such, these criteria aim to make it easier for these users to understand the content as it is presented.

SECTION 508

- 1194.21(e) When bitmap images are used to identify controls, status indicators, or other programmatic elements, the meaning assigned to those images shall be consistent throughout an application's performance.

WCAG 2.0

- 2.4.4 Link Purpose in Context
- 3.2.4 Consistent Identification (1194.21(e))
- 3.3.2 Labels and Instructions

Link Purpose in Context

When a user sees a link, they need to be able to quickly tell whether they want to click on it. Whether it is going to provide the information they're expecting. The link text itself should have sufficient information to provide that. Avoid "click here" links.

In a previous section, "Alternate Reports", a link was created to allow the user to see a table of values rather than viewing a graph. The link itself was the entire sentence and was blue underlined like this:

[Table of values for Bar chart of frequency of ages](#)

Quite often, you find links that have part of the sentence as the link, and the text that's the link is rather generic, such as "click here":

[Click here](#) for a Table of values for Bar chart of frequency of ages

Links such as "click here" are difficult for screen reader users. Look back at Figure 31 - Links List Dialog Box in JAWS Screen Reader. The dialog box displays all the links on the page using the text for the link. If your report has several "click here" links, the dialog box would just show a series of "click here" items and it would be impossible to know where those links took you. In addition, the screen reader would just say "click here" when you navigated to the link. Now granted, if the screen reader user explored the text before and after the link, they could probably figure it out, but that takes time and it is a better practice to have the link text contain the relevant information.

Consistent Identification

When an object has a label, use the same term throughout your report. For example, when printing a class list, you might have a gender column. Later in the report, another table might call the column sex. Even though you might be printing 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.

```
proc print data=sashelp.class label;  
    label sex='Gender';  
run;
```

Labels and Instructions

This is typically not an issue in ODS. It concerns forms on web pages. If the user has to fill in information, make sure all the input fields are labeled and the instructions are clear, especially if formats are involved, such as phone numbers, date fields, and so on.

HEARING

These requirements are specific to audio and video content. If you are inserting audio or video clips into your report, closed captions, audio transcriptions, and audio descriptions are important for accessibility. These topics are typically outside the scope of ODS reports and will not be discussed here.

TESTING FOR ACCESSIBILITY

There are many tools for testing whether your output is compliant with the WCAG 2.0 guidelines. Earlier, a color contrast checker was mentioned. Two other tools are the [WAVE Chrome Extension](#) from WebAIM.org and [aXe](#) (The Accessibility Engine) from deque.com. aXe runs in both Firefox and Chrome but Firefox has the additional ability to highlight the areas on the page that have problems so that it's a little easier to use there.

SAS does not endorse these tools nor claim they are completely accurate, but they are handy. Both tools have good support on their respective websites but you can also contact the author if you have questions.

FUTURE

SAS is continuing to research improvements for the HTML5 destination to help make accessible reports easier to generate. Feel free to contact the author with suggestions.

TAGGED PDF

PDF is a very popular output destination. In order for PDF to be accessible, it has to follow the PDF/Universal Accessibility specification (PDF/UA) as defined in ISO 14289. Accessible PDF is called "Tagged PDF" because the file contains HTML-like tags to enable assistive technology to read PDF.

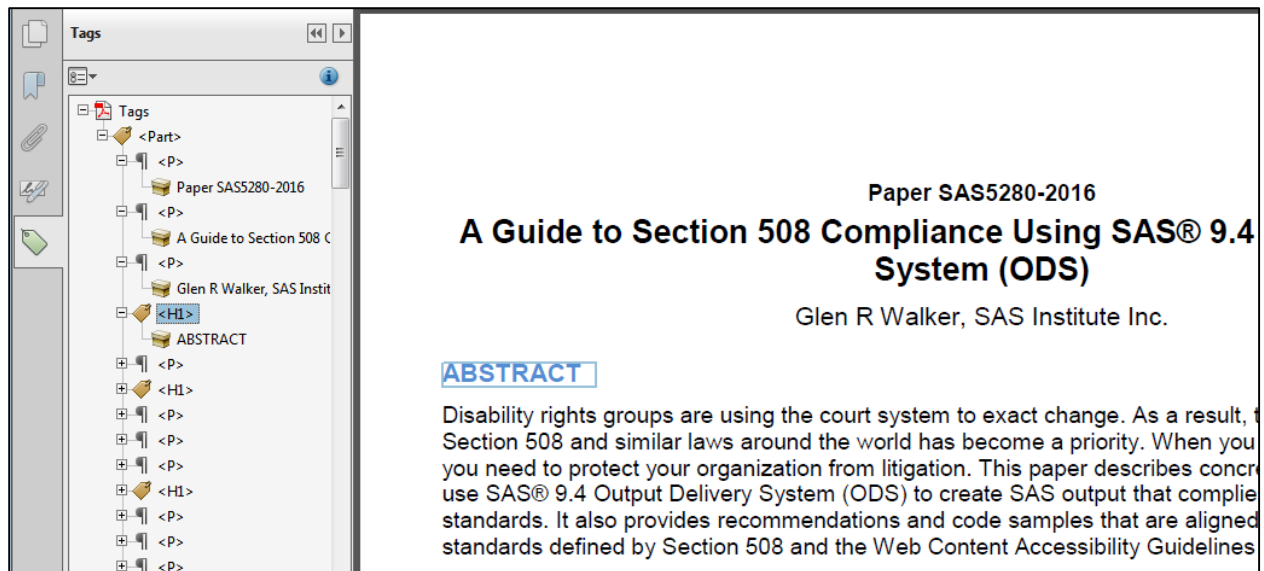


Figure 36 - Sample Tagged PDF

Notice in the tree panel on the left, there are tags that look like HTML, such as `<p>` and `<h1>`. Those tags serve the same purpose as HTML tags in that they provide information to assistive technologies. The tags do not affect the display of the document.

Promising work is ongoing at SAS with generating tagged PDF from ODS and we're excited to provide this feature in an upcoming release.

CONCLUSION

Accessibility affects everyone. Great user designs can create an experience that is delightful for a universal audience. Section 508 was a good start in giving guidance to report developers and web page designers but with the addition of the Web Content Accessibility Guidelines (WCAG 2.0), you can take a giant leap forward in accessibility. By conforming to the WCAG criteria, and considering the types of needs that your users require, reports can be generated that will allow all audiences to benefit from your work.

REFERENCES

- "Nearly 1 in 5 People Have a Disability in the U.S." 2012. United States Census Bureau News Release. CB12-134. Available <https://www.census.gov/newsroom/releases/archives/miscellaneous/cb12-134.html>. Accessed on March 7, 2016.
- "Overview." The World Bank. Available <http://www.worldbank.org/en/topic/disability/overview>. Accessed on March 7, 2016.
- General Services Administration (GSA) Government-wide Section 508 Accessibility Program. Available <http://www.section508.gov/>. Accessed on March 7, 2016.
- United States Access Board. Available <https://www.access-board.gov/>. Accessed on March 7, 2016.
- "Understanding WCAG 2.0." W3C. Available <https://www.w3.org/TR/UNDERSTANDING-WCAG20/>. Accessed on March 7, 2016.
- "Color Contrast Checker." WebAIM. Available <http://webaim.org/resources/contrastchecker/>. Accessed on March 7, 2016.

"WAVE Chrome Extension." WAVE. Available <http://wave.webaim.org/extension/>. Accessed on March 7, 2016.

"aXe: the Accessibility Engine." Deque. Available <http://www.deque.com/products/axe/>. Accessed on March 7, 2016.

"relative luminance." W3C. Available <https://www.w3.org/TR/2008/REC-WCAG20-20081211/#relativeluminancedef>. Accessed on March 7, 2016.

RECOMMENDED READING

"Section 508 Checklist for Web-based Internet Information and Applications." US Department of Veteran Affairs. Available http://www.section508.va.gov/docs/1194_22_VASection508_web_checklist.doc. Accessed on March 7, 2016.

ACKNOWLEDGMENTS

Special thanks to several members of the accessibility team at SAS:

- Julianna Langston, for guidance with the SAS examples
- Jesse Sookne, for writing ideas and commiserating
- Ed Summers, for editing and providing feedback
- Tyler Williamson, for WCAG definitions and groupings

Also, thanks to the ODS team for being patient with my questions: Donna Antle, David Kelley, Bari Lawhorn, and Dan O'Connor.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Glen R Walker
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Glen.Walker@sas.com
www.sas.com/accessibility

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

APPENDIX 1

The standards for Section 508 are defined in §1194.21, "Software applications and operating systems," and §1194.22, "Web-based Intranet and Internet Information and Applications." There are twelve standards in §1194.21, lettered A through L and sixteen in §1194.22, lettered A through P.

§1194.21 SOFTWARE APPLICATIONS AND OPERATING SYSTEMS

Standard	What it means	Section in this paper (these are links)
1194.21(a) When software is designed to run on a system that has a keyboard, product functions shall be executable from a keyboard where the function itself or the result of performing a function can be discerned textually.	Keyboard navigation	NO VISION
1194.21(b) Applications shall not disrupt or disable activated features of other products that are identified as accessibility features, where those features are developed and documented according to industry standards. Applications also shall not disrupt or disable activated features of any operating system that are identified as accessibility features where the application programming interface for those accessibility features has been documented by the manufacturer of the operating system and is available to the product developer.	not applicable	
1194.21(c) A well-defined on-screen indication of the current focus shall be provided that moves among interactive interface elements as the input focus changes. The focus shall be programmatically exposed so that assistive technology can track focus and focus changes.	Keyboard focus	LOW VISION
1194.21(d) Sufficient information about a user interface element including the identity, operation and state of the element shall be available to assistive technology. When an image represents a program element, the information conveyed by the image must also be available in text.	See WCAG 4.1.2 Name, Role, Value	NO VISION
1194.21(e) When bitmap images are used to identify controls, status indicators, or other programmatic elements, the meaning assigned to those images shall be consistent throughout an application's performance.	See WCAG 3.2.4 Consistent Identification	COGNITIVE
1194.21(f) Textual information shall be provided through operating system functions for displaying text. The minimum information that shall be made available is text content, text input caret location, and text attributes.	not applicable	
1194.21(g) Applications shall not override user selected contrast and color selections and other individual display attributes.	The browser is responsible for honoring system theme changes.	LOW VISION
1194.21(h) When animation is displayed, the information shall be displayable in at least one non-animated presentation mode at the option of the user.	not applicable	
1194.21(i) Color coding shall not be used as the only means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.	Don't rely solely on color to convey meaning.	LOW VISION

Standard	What it means	Section in this paper (these are links)
1194.21(j) When a product permits a user to adjust color and contrast settings, a variety of color selections capable of producing a range of contrast levels shall be provided.	not applicable (but could provide a report in several styles)	LOW VISION
1194.21(k) Software shall not use flashing or blinking text, objects, or other elements having a flash or blink frequency greater than 2 Hz and lower than 55 Hz.	In general, avoid any animated images or rapid screen color changes.	
1194.21(l) When electronic forms are used, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.	Label form elements	NO VISION

§1194.22 WEB-BASED INTRANET AND INTERNET INFORMATION AND APPLICATIONS

Standard	What it means	Section in this paper (these are links)
1194.22(a) A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content).	Images should have alternate text.	LOW VISION
1194.22(b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.	Closed captions should follow the video.	Hearing (but not relevant to ODS)
1194.22(c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.	Don't rely solely on color to convey meaning.	LOW VISION
1194.22(d) Documents shall be organized so they are readable without requiring an associated style sheet.	Don't use style sheets in place of structural elements (for example, <h1>) and don't use style sheets to force location on the page.	NO VISION
1194.22(e) Redundant text links shall be provided for each active region of a server-side image map.	Provide a list or table of links as an alternative to server-side image maps.	No Vision (but not relevant to ODS)
1194.22(f) Client-side image maps shall be provided instead of server-side image maps except where the	Don't use server-side image maps. The links in client-side image maps should	NO VISION

Standard	What it means	Section in this paper (these are links)
regions cannot be defined with an available geometric shape.	have alternative text to satisfy requirement 1194.22(a).	
1194.22(g) Row and column headers shall be identified for data tables.	Use <th scope='col'> and <th scope='row'> for table headers	NO VISION
1194.22(h) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.	Multiple rows of headers need proper markup. (Proper implementation of 1194.22(g) should take care of this)	NO VISION
1194.22(i) Frames shall be titled with text that facilitates frame identification and navigation.	Use the <title> element for the main page and the title attribute for any <frame> elements. <frame> is not supported in HTML5	NO VISION
1194.22(j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.	In general, avoid any animated images or rapid screen color changes.	
1194.22(k) A text-only page with equivalent information or functionality shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.	If a graphics only report is generated, then the same data needs to be provided in text format (usually tables).	NO VISION
1194.22(l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.	ODS output doesn't rely on JavaScript so that this shouldn't be an issue but in general, don't generate critical information from JavaScript.	
1194.22(m) When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l).	ODS output doesn't rely on plug-ins so that this shouldn't be an issue.	
1194.22(n) When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and	All form elements should have a label. ODS doesn't use forms.	No Vision (but not relevant to ODS)

Standard	What it means	Section in this paper (these are links)
submission of the form, including all directions and cues.		
1194.22(o) A method shall be provided that permits users to skip repetitive navigation links.	With proper headers (<h2>) in ODS, navigation skips are less of a problem. Most ODS output doesn't have extra navigation objects to skip.	MOBILITY
1194.22(p) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.	Since ODS doesn't have forms, there is no timed response. If you automatically log users off, they should have an opportunity to lengthen the login time.	No Vision (but not relevant to ODS)

APPENDIX 2

WCAG 2.0

WCAG criteria that are covered in this paper have a link back into the paper.

Level A	Level AA	Level AAA
1.1.1 Non-text Content	1.2.4 Captions (Live)	1.2.6 Sign Language (Prerecorded)
1.2.1 Audio-only and Video-only (Prerecorded)	1.2.5 Audio Description (Prerecorded)	1.2.7 Extended Audio Description (Prerecorded)
1.2.2 Captions (Prerecorded)	1.4.3 Contrast (Minimum)	1.2.8 Media Alternative (Prerecorded)
1.2.3 Audio Description or Media Alternative (Prerecorded)	1.4.4 Resize Text	1.2.9 Audio-only (Live)
1.3.1 Info and Relationships	1.4.5 Images of Text	1.4.6 Contrast (Enhanced)
1.3.2 Meaningful Sequence	2.4.5 Multiple Ways	1.4.7 Low or No Background Audio
1.3.3 Sensory Characteristics	2.4.6 Headings and Labels	1.4.8 Visual Presentation
1.4.1 Use of Color	2.4.7 Focus Visible	1.4.9 Images of Text (No Exception)
1.4.2 Audio Control	3.1.2 Language of Parts	2.1.3 Keyboard (No Exception)
2.1.1 Keyboard	3.2.3 Consistent Navigation	2.2.3 No Timing
2.1.2 No Keyboard Trap	3.2.4 Consistent Identification	2.2.4 Interruptions
2.2.1 Timing Adjustable	3.3.3 Error Suggestion	2.2.5 Re-authenticating

Level A	Level AA	Level AAA
2.2.2 Pause, Stop, Hide	3.3.4 Error Prevention (Legal, Financial, Data)	2.3.2 Three Flashes
2.3.1 Three Flashes or Below Threshold		2.4.8 Location
2.4.1 Bypass Blocks		2.4.9 Link Purpose (Link Only)
2.4.2 Page Title		2.4.10 Section Headings
2.4.3 Focus Order		3.1.3 Unusual Words
2.4.4 Link Purpose (In Context)		3.1.4 Abbreviations
3.1.1 Language of Page		3.1.5 Reading Level
3.2.1 On Focus		3.1.6 Pronunciation
3.2.2 On Input		3.2.5 Change on Request
3.3.1 Error Identification		3.3.5 Help
3.3.2 Labels or Instructions		3.3.6 Error Prevention (All)
4.1.1 Parsing		
4.1.2 Name, Role, Value		

APPENDIX 3

Formula to compute color contrast ratio. <https://www.w3.org/TR/2008/REC-WCAG20-20081211/#relativeluminancedef>

Contrast Ratio = $(L1 + 0.05) / (L2 + 0.05)$

L1 = luminance of the lighter color (larger number)

L2 = luminance of the darker color (smaller number)

$L = 0.2126 * R + 0.7152 * G + 0.0722 * B$

R, G, B = the normalized values of the red, blue, and green components of a color

(for example, #8A2BE2 would have red=0x8A, green=0x2B, blue=0xE2)

The normalized value of a color component is a little tricky. You first divide it by 255 (the maximum value for a color component) and then based on that value, you do one of two things. It's easier to visualize an if/then statement.

```

value = value / 255
if ( value <= 0.03928 ) then
    value = value / 12.92
else
    value = ((value + 0.055) / 1.055)2.4

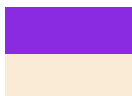
```

EXAMPLE:

Is there sufficient contrast if you use a text color of purple on a light beige background?

Text color = #8A2BE2 (dark purple)

Background color = #FAEBD7 (light beige)



L1 = lighter color (larger number) = #FAEBD7
R1 = 0xFA = 250
G1 = 0xEB = 235
B1 = 0xD7 = 215

L2 = dark color (smaller number) = #8A2BE2
R1 = 0x8A = 138
G1 = 0x2B = 43
B1 = 0xE2 = 226

The data step below computes the contrast ratio. The result is 5.08, which is larger than the minimum, 4.5, so using dark purple on light beige is ok.

```
data _null_;
  L1 = 0FAEBD7x; /* light beige */
  L2 = 08A2BE2x; /* dark purple */

  /* split up the color into red, green, and blue components */

  r1 = brshift( band(L1,0FF0000x), 16 );
  g1 = brshift( band(L1,000FF00x), 8 );
  b1 =          band(L1,00000FFx);

  r2 = brshift( band(L2,0FF0000x), 16 );
  g2 = brshift( band(L2,000FF00x), 8 );
  b2 =          band(L2,00000FFx);

  /* calculate normalized value for each color component */

  r1 = r1 / 255;
  if ( r1 <= 0.03928 ) then
    r1 = r1 / 12.92;
  else
    r1 = ((r1 + 0.055) / 1.055) ** 2.4;

  g1 = g1 / 255;
  if ( g1 <= 0.03928 ) then
    g1 = g1 / 12.92;
  else
    g1 = ((g1 + 0.055) / 1.055) ** 2.4;

  b1 = b1 / 255;
  if ( b1 <= 0.03928 ) then
    b1 = b1 / 12.92;
  else
    b1 = ((b1 + 0.055) / 1.055) ** 2.4;

  r2 = r2 / 255;
  if ( r2 <= 0.03928 ) then
    r2 = r2 / 12.92;
  else
    r2 = ((r2 + 0.055) / 1.055) ** 2.4;

  g2 = g2 / 255;
  if ( g2 <= 0.03928 ) then
    g2 = g2 / 12.92;
  else
```

```

        g2 = ((g2 + 0.055) / 1.055) ** 2.4;

b2 = b2 / 255;
if ( b2 <= 0.03928 ) then
    b2 = b2 / 12.92;
else
    b2 = ((b2 + 0.055) / 1.055) ** 2.4;

put r1= g1= b1=;
put r2= g2= b2=;

/* calculate luminance for both colors */

L1 = 0.2126 * R1 + 0.7152 * G1 + 0.0722 * B1;
L2 = 0.2126 * R2 + 0.7152 * G2 + 0.0722 * B2;

/* calculate color ratio */

L = (L1 + 0.05) / (L2 + 0.05);
put L=;
run;

```

```

r1=0.9559733532 g1=0.8307698768 b1=0.6795424696
r2=0.2541520943 g2=0.0241576324 b2=0.7605245047
L=5.0872136448
NOTE: DATA statement used (Total process time):
      real time           0.03 seconds
      cpu time            0.03 seconds

```

Output 1 - Output from Data Step