

Tag, You're It! Creating Accessible (Tagged) PDF with SAS® 9.4 Output Delivery System

Glen R Walker, SAS Institute Inc.

ABSTRACT

In the game of tag, being "it" is bad, but where accessibility compliance is concerned, being tagged is good! Tagging is required for PDF files to comply with accessibility standards such as Section 508 and the Web Content Accessibility Guidelines (WCAG). In the fourth maintenance release for SAS 9.4, the preproduction option in the ODS PDF statement, ACCESSIBLE, creates a tagged PDF file. We look at how this option changes the file that is created and focus on the SAS® programming techniques that work best with the new option. You'll then have the opportunity to try it yourself in your own code and provide feedback to SAS.

INTRODUCTION

Portable Document Format (PDF) files have been around for almost 25 years and the PDF output destination is a very popular Output Delivery System (ODS) destination. PDF files look beautiful and retain their crisp clean look when you zoom the text larger. This is a great feature when you have to zoom the text for presentations on a projector and is also beneficial for individuals with lower vision. And for those whose eyes are aging (like mine), zooming the text avoids having to put on glasses.

However, while PDF files look great, they are not so beautiful to screen readers.

Screen readers are assistive technology that present information audibly, typically with a synthesized voice, rather than visually. They allow you to hear what's on the screen and interact with the computer.

Even though PDF files created by SAS appear to be full of text, that text is not available to screen readers. That means anyone with low vision or no vision will not be able to listen to the file. Also, since a PDF file is just a collection of graphic primitives, a line might be part of a bar chart or it could be the border of a table. Additional information, or metadata, is needed to convey the purpose of the line.

In order for a PDF file to be accessible to screen readers, the contents of the file must follow the PDF/Universal Accessibility specification (PDF/UA) as defined in [ISO 14289](http://www.iso.org/iso/14289). An accessible PDF is called "Tagged PDF" because the file contains HTML-like tags to enable assistive technology to read the PDF.

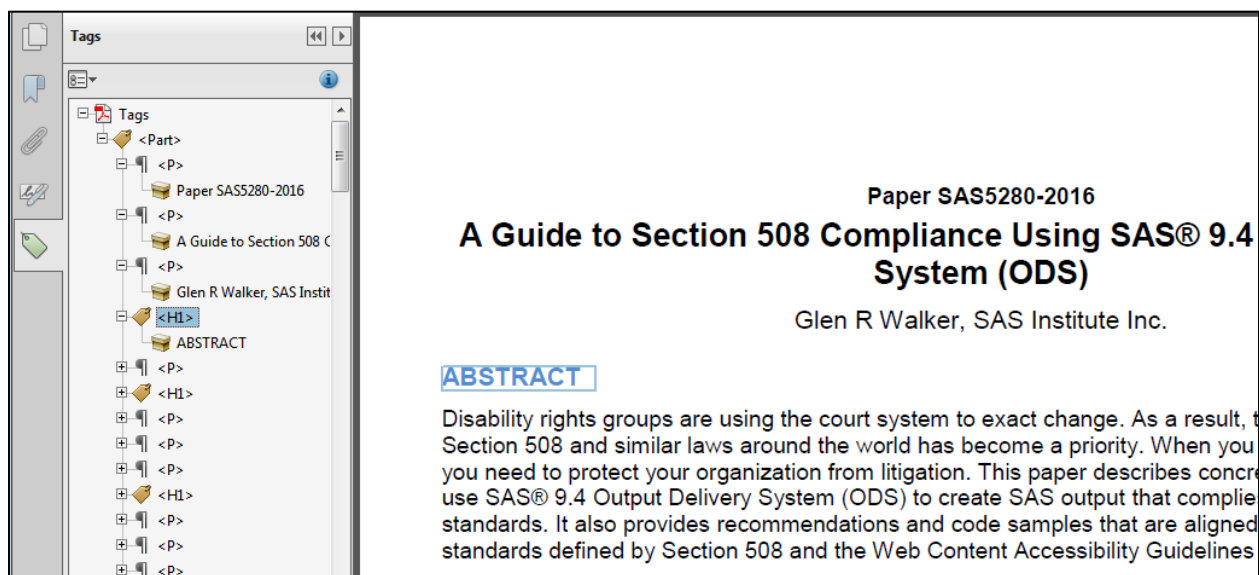


Figure 1 – Adobe Acrobat Showing a Sample Tagged PDF with the Tags Panel Open on the Left

Figure 1 is a screen shot of Adobe Acrobat XI with the tags panel turned on (View > Show/Hide > Navigation Panes > Tags). The tags panel on the left shows a tree with tags that look like HTML, such as <p>, <h1>, and <table>. Those tags serve the same purpose as HTML tags do to a browser in that they provide semantic information to assistive technologies. They convey meaning about the object. The tags themselves do not affect the display of the document. They provide a “layer” above the text on the page that allows assistive technology to navigate through the document. The order of the tags in the tree are the order that the text is to be read in the document.

Since there is extra metadata in the file, an accessible PDF file is bigger than a PDF file that is not tagged. Depending on how many tables or graphs are in the file, the size could increase by 20%. But memory is cheap and making all your PDF files accessible benefits many of people.

ACCESSIBLE PDF AND WCAG

The [Web Content Accessibility Guidelines](#) (WCAG) is an internationally accepted list of accessibility guidelines published by the Web Accessibility Initiative (WAI) of the World Wide Web Consortium (W3C). It contains guidance on how to generate output that is accessible. There are several parts of WCAG that apply to PDF files. For a detailed discussion on how WCAG applies to ODS in general, see my paper from last year, “[A Guide to Section 508 Compliance Using SAS® 9.4 Output Delivery System \(ODS\)](#)” in the “Proceedings of the SAS Global Forum 2016 Conference.”

Adobe has a nice reference of how WCAG applies to tagged PDF and has links to various W3C documents on how to implement the guidelines, [Create and Verify PDF Accessibility \(Acrobat Pro\)](https://helpx.adobe.com/acrobat/11/using/create-verify-pdf-accessibility.html#WCAGmappingtoPDFUA) (<https://helpx.adobe.com/acrobat/11/using/create-verify-pdf-accessibility.html#WCAGmappingtoPDFUA>)

Another discussion regarding tagged PDF and WCAG can be found on CommonLook.com, [WCAG 2.0 and PDF/UA – Your Questions Answered](#) (

<http://commonlook.com/WCAG-20-and-PDF-UA-Your-Questions-Answered/>).

GENERATING ACCESSIBLE TAGGED PDF FILES WITH ODS

In SAS 9.4M4 a new option, ACCESSIBLE, was added to the ODS PDF statement:

```
ods pdf file='report.pdf' accessible;
```

It can be added to any existing SAS job.

The ACCESSIBLE option for ODS PDF can also be enabled in the SAS registry, found in the menu Solutions > Accessories > Registry Editor (or by submitting the REGEDIT command).

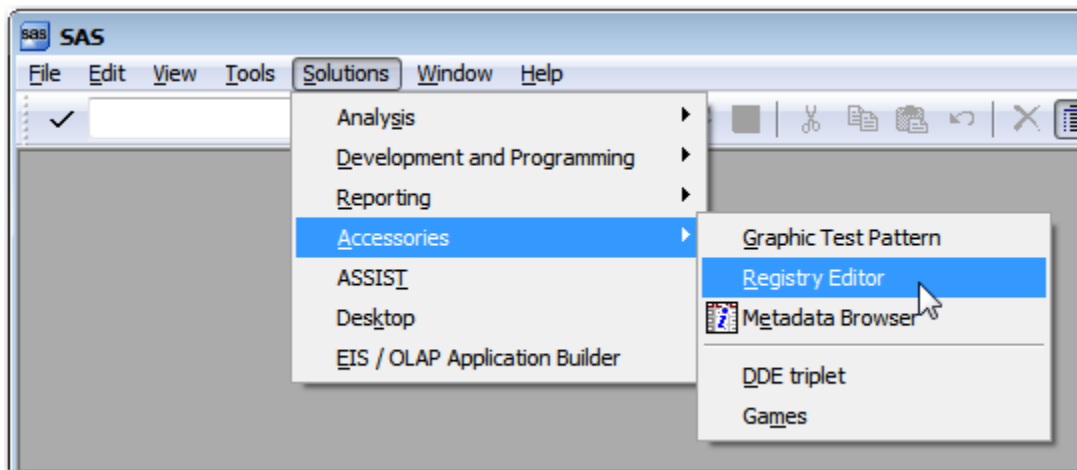


Figure 2 – Menu Path to SAS Registry Editor

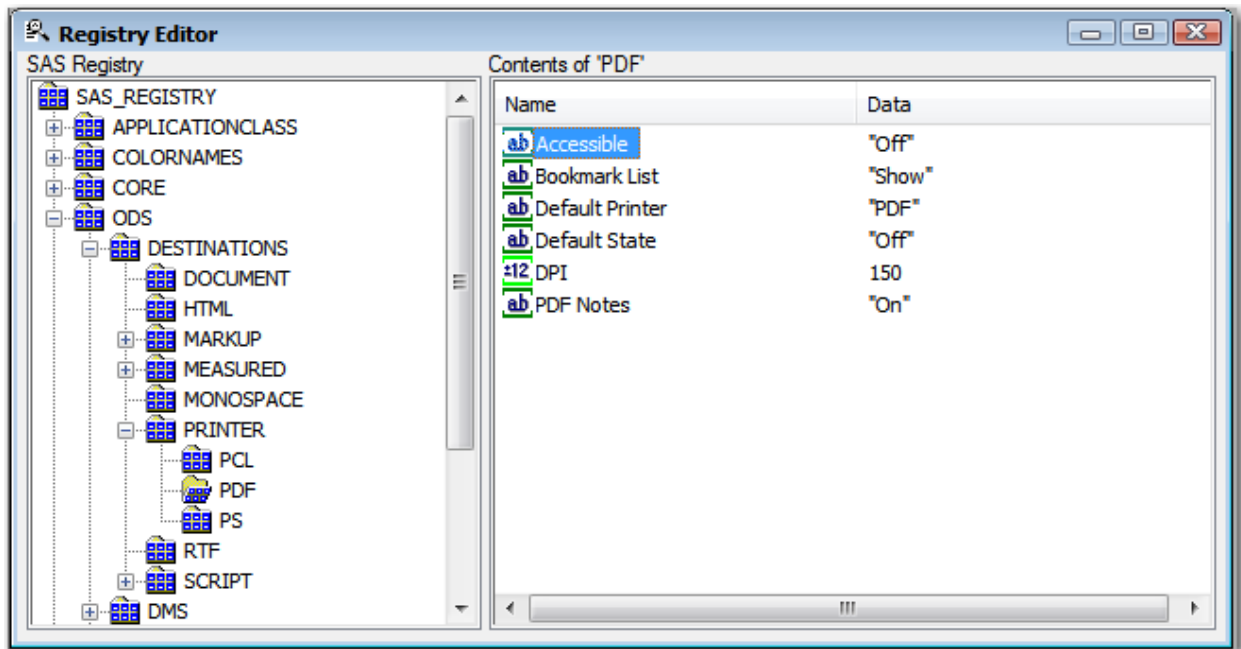


Figure 3 - SAS Registry Editor

The path to the ACCESSIBLE option in the registry is: SAS_REGISTRY > ODS > DESTINATIONS > PRINTER > PDF. Double click or press ENTER on the Accessible key and change the value to "On".

For batch processing, you can change the registry using PROC REGISTRY:

```
proc registry import='mystuff.txt'; run;
```

where 'mystuff.txt' contains:

```
[ODS\DESTINATIONS\PRINTER\PDF]
"Accessible"="On"
```

HOW TO GENERATE TAGGED TABLES WITH PROC PRINT

Here's a simple example of generating a tagged PDF file using PROC PRINT:

```
ods pdf file='students.pdf' accessible;
proc print data=sashelp.class;
  id name;
run;
ods pdf close;
```

The ID statement specifies that the NAME variable should be used as the row headers in the table. The sample output in Figure 4 shows the column headers and row headers (student names) are bolder than the rest of the data in the table.

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
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5
James	M	12	57.3	83.0

Figure 4 - PDF Output for SASHELP.CLASS

The tags can be seen in Figure 5:

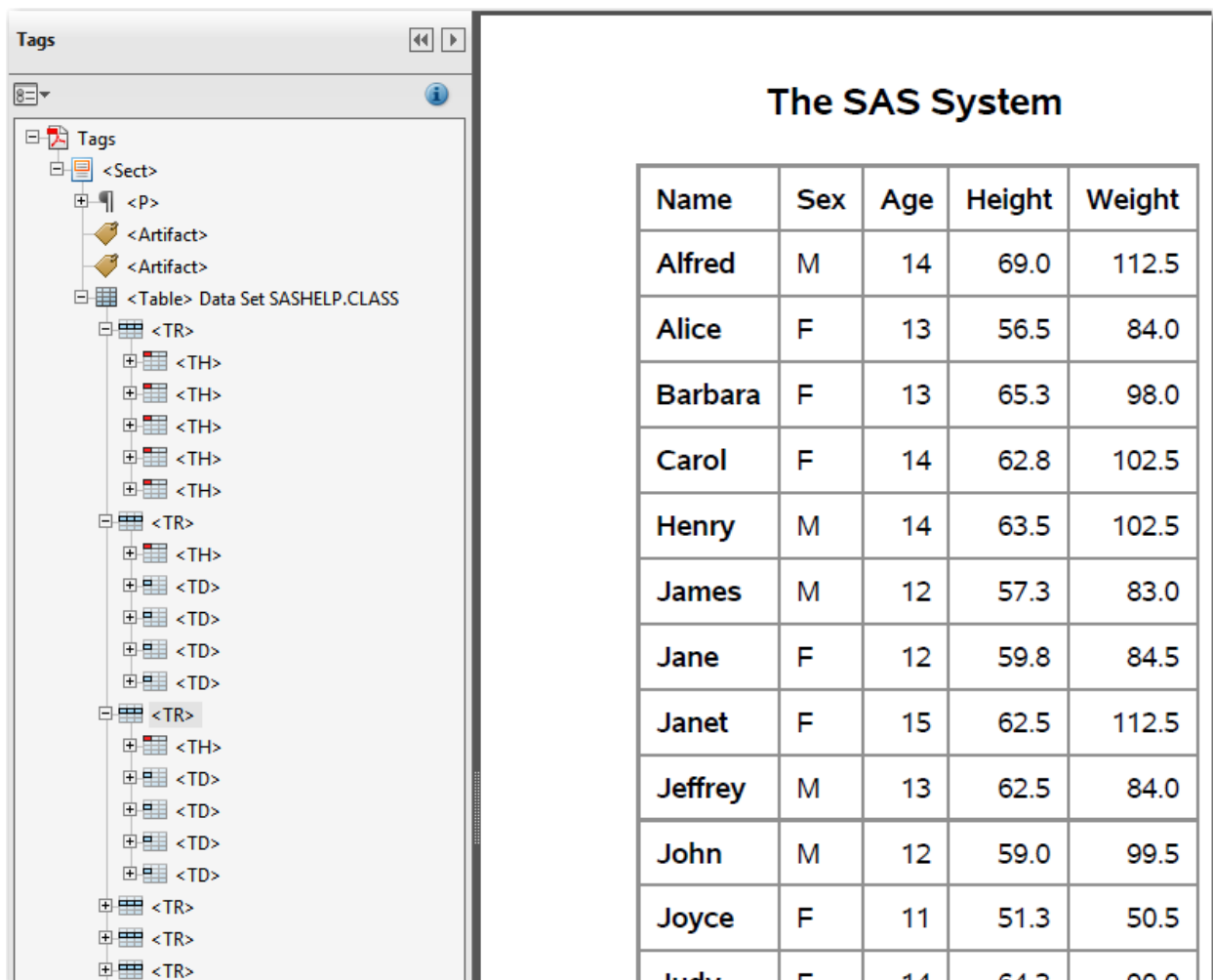


Figure 5 – Adobe Acrobat Showing the Tags Panel for the Table SASHELP.CLASS generated by PROC PRINT

The table starts with a <table> tag and contains a <tr> for each row in the table, which corresponds to each observation in the data set. Each row contains either a <th> or a <td> for the values of the variables. A <th> is a header and <td> is data. The first row in the table contains all headers (<th>) and are the variable names.

Notice that in the second and subsequent rows (<tr>), the first tag under <tr> is a <th> and the rest are <td>s. The <th> is the header for the row which is the name of the student. This is the same way HTML tables are created.

By having properly tagged headers, a screen reader user can easily navigate through a table. As they move the screen reader “focus” across a row, the column header is read (such as “age”) and then the cell value is read (“14”). As they move down a column, the row header (student name) is read and then the cell value is read. Without the tags, no information would be available to the screen reader.

This simple example generates a file that is 101KB. Without the ACCESSIBLE option on the ODS PDF statement, the file size is 82KB, so it increased by 23%.

HOW TO GENERATE TAGGED TABLES WITH PROC REPORT

PROC REPORT has partial support for accessible PDF. Column headers and spanned column headers are tagged properly but row headers are not. The following example creates a simple spanned column header using the same SASHELP.CLASS data set.

```
ods pdf file='students.pdf' accessible;  
proc report spanrows data=sashelp.class;  
  col name sex ("Health Stats" age height weight);  
  define name / style(column)=rowheader;  
run;  
ods pdf close;
```

		Health Stats		
Name	Sex	Age	Height	Weight
Alfred	M	14	69	112.5
Alice	F	13	56.5	84
Barbara	F	13	65.3	98
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5

Figure 6 - Spanned Column Header in PROC REPORT

The tags in this file are still fairly simple but there's additional metadata because of the spanned columns.

The screenshot shows a 'Tags' window on the left and a rendered table on the right. The 'Tags' window displays a tree view of the HTML structure for a table. The root is '<Table> Table 1', which contains a '<TR>' with two '<TH>' elements. The first '<TH>' is followed by another '<TH>', and the second '<TH>' is followed by three '<TD>' elements. The rendered table on the right is titled 'The SAS System' and has five columns: Name, Sex, Age, Height, and Weight. The first row of data is Alfred, M, 14, 69, 112.5. The second row is Alice, F, 13, 56.5, 84. The third row is Barbara, F, 13, 65.3, 98. The fourth row is Carol, F, 14, 62.8, 102.5. The fifth row is Henry, M, 14, 63.5, 102.5. The sixth row is James, M, 12, 57.3, 83. The seventh row is Jane, F, 12, 59.8, 84.5. The eighth row is Janet, F, 15, 62.5, 112.5. The ninth row is Jeffrey, M, 13, 62.5, 84.

Figure 7 - Tags for SASHELP.CLASS with Spanned Column Headers

The first <tr> in Figure 7 contains only two <th> tags even though there are five columns in the table. This is because the first <th> spans two columns (and has blank text) and the second <th> spans three columns ("Health Stats"). The spanning information is contained in the additional metadata. Figure 8 shows the properties of the "Health Stats" column. The "ColSpan" property is set to 3, meaning it spans three columns. When a screen reader user gets to the "Health Stats" cell, it will say "Health Stats, Spans 3 columns".

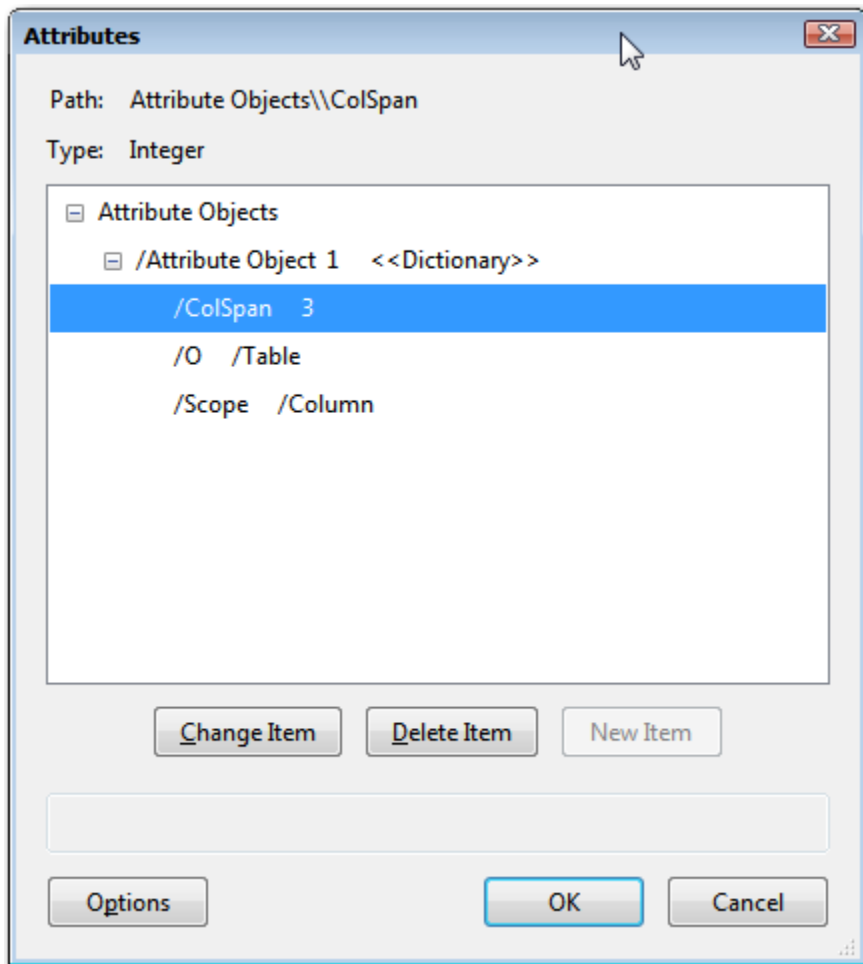


Figure 8 - Metadata for <th> tag

The second <tr> in Figure 7 contains five column headers, for the five variables in the data set.

The third <tr> contains five data cells, <td>. The first <td> really should be a <th>, row header, so that as the screen reader user moves down a column, they hear the student name before they hear the value of the cell, but PROC REPORT does not support that yet. Visually, the student name is bold so it appears to be a row header, but it's not tagged properly so the screen reader will not interpret it as a row header. This needs to be researched in a future release of SAS. As a side note, if you use the HTML5 destination in the above example, the row headers **are** marked properly and the screen reader works correctly.

As a screen reader user navigates across a row, because both the "Health Stats" column and the "Age", "Height", and "Weight" columns are all marked as column headers, both column headers are read before the value in the table cell. For example, if you're on the Barbara row in the Age column, if you move to the right, the screen reader says "Health Stats, Height, 65.3". Moving to the right again says, "Health Stats, Weight, 98". Moving left to the sex column, the screen reader just says "Sex, F". There's only one column header for the sex column.

REPORT WRITING INTERFACE

The Report Writing Interface (RWI) has been improved to support tagged PDF. More details can be found in "A Guru's Guide: Producing section 508 compliant custom reports with ODS," *Proceedings of the SAS Global Forum 2017 Conference*, by Daniel O'Connor.

To generate the same results as PROC REPORT (Figure 6), RWI can be used.


```

ods pdf file='students.pdf' accessible;
data _null_;
  set sashelp.class end=last;

  if _n_ eq 1 then do;
    declare odsout t();

    t.table_start();
    t.head_start(); /* 1 */
    t.row_start(); /* 1 */
    t.format_cell(data: '', colspan:2);
    t.format_cell(data: 'Health Stats', colspan:3);
    t.row_end();
    t.row_start(); /* 1 */
    t.format_cell(data: 'Name');
    t.format_cell(data: 'Sex');
    t.format_cell(data: 'Age');
    t.format_cell(data: 'Height');
    t.format_cell(data: 'Weight');
    t.row_end();
    t.head_end(); /* 1 */
  end;

  t.row_start();
  t.format_cell(data: name, style_elem: 'RowHeader'); /* 2 */
  t.format_cell(data: sex);
  t.format_cell(data: age);
  t.format_cell(data: height);
  t.format_cell(data: weight);
  t.row_end();

  if last then
    t.table_end();
run;
ods pdf close;

```

The comment for /* 1 */ is on the head_start() method call. Instead of using head_start(), you could add a type:'H' to the row_start() following it and then remove the head_start() and head_end().

```
t.row_start(type:'H');
```

Both head_start() and row_start(type:'H') generate the same results. They both cause column headers to be tagged.

The comment for /* 2 */ is to show how row headers should be created. The output will look like Figure 6 and the tags will look like Figure 7. Unfortunately, just as PROC REPORT does not support row headers, neither does RWI. But the code above still tries to generate row headers, which is a good practice.

And just as with PROC REPORT, if you use the HTML5 destination in the above example, the row headers **are** marked properly and the screen reader works correctly.

IMAGES

When including images in your output, make sure you specify alternate text for the image so the screen reader can read information about the image. The alternate text is specified using the DESCRIPTION option.

```
data _null_;
```

```

dcl odsout obj();
obj.image(file: "https://www.sas.com/etc/designs/saswww/apple-touch-
icon-120x120.png", description: "The 'S' SAS logo");
run;

```

You can view the alternative text in the properties of the Figure tag as seen in Figure 9.

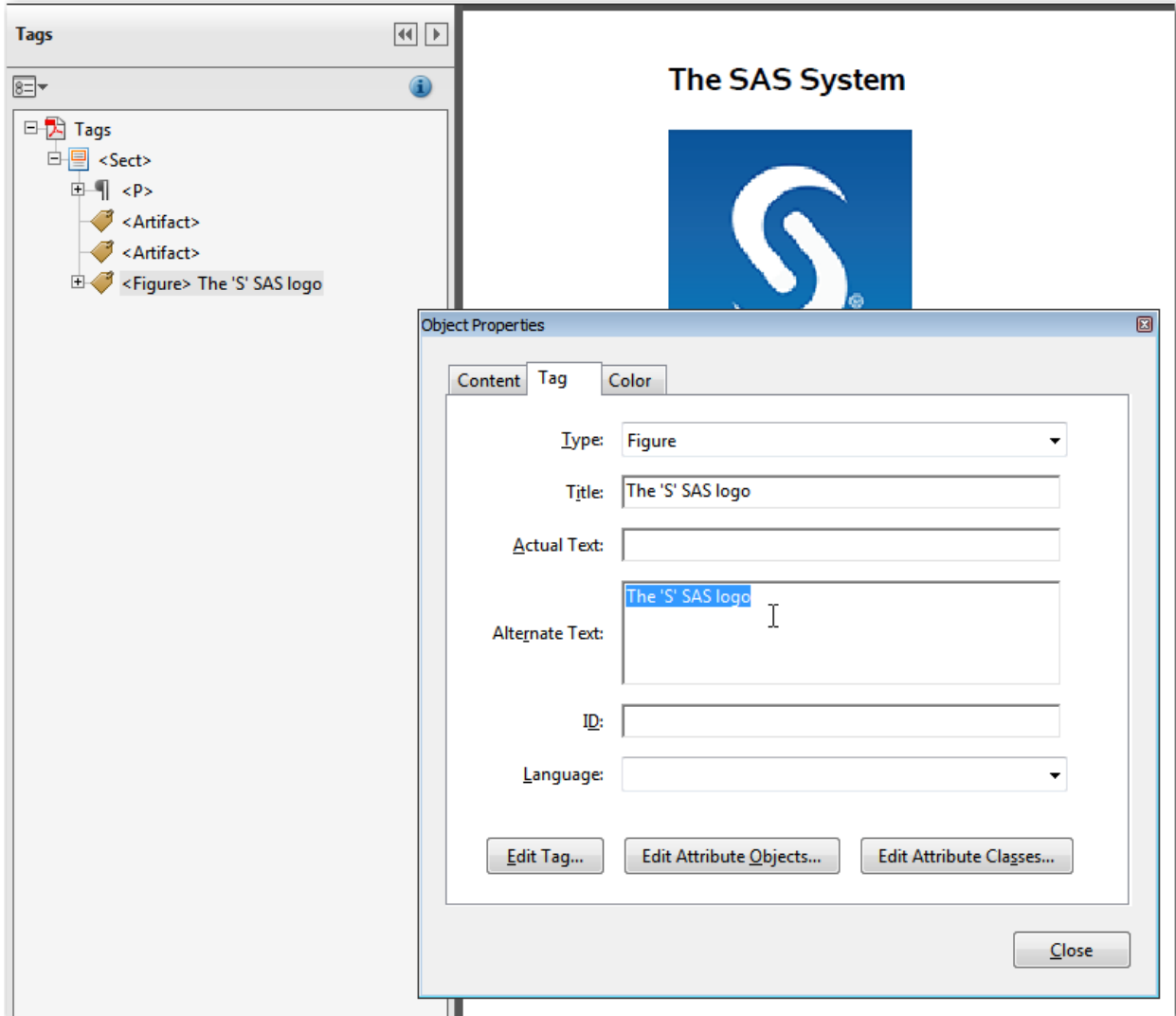


Figure 9 - Alternate Text Properties of the Figure Tag for an RWI Image

GRAPHS

The DESCRIPTION option should also be used when generating graphs and charts. While a description of the graph is a poor substitute from being able to examine all the contents of the graph, it's better than not having a description.

```

proc sgplot data=sashelp.class
  description='bar chart of the frequency of ages in SASHELP.CLASS';
  vbar age ;
run;

```

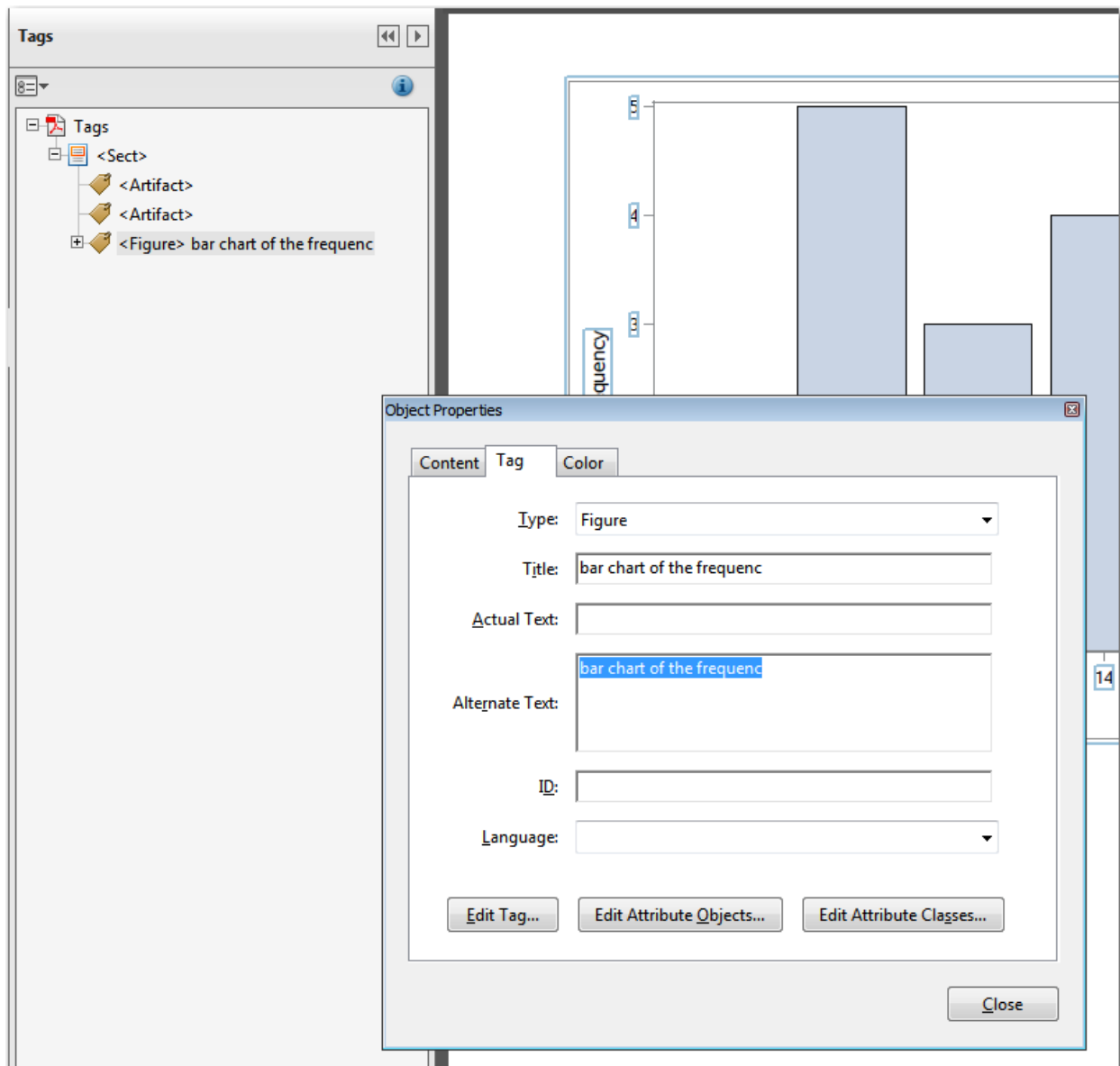


Figure 10 - Alternate Text Properties of the Figure Tag for PROC SGPLOT

Note that there is currently an issue with the alternate text truncating when tagging the SGPLOT output as seen in Figure 10. The alternate text should show “bar chart of the frequency of ages in SASHELP.CLASS” but it’s being truncated to 25 characters to “bar chart of the frequenc”.

HOW TO TEST IF YOU’RE GENERATING ACCESSIBLE PDF

Now that you’ve generated a tagged PDF file, how do you verify that it’s valid?

The ideal solution is to use a screen reader, such as JAWS or NVDA, and navigate through the document to see if you can hear everything. Use the screen reader’s quick navigation keys, such as ‘T’ to move to a table or ‘H’ to move to a heading. See if alternate text is read for images.

The PDF Association ([pdfa.org](https://www.pdfa.org)) came up with a list of possible tagged PDF failures. They call it The Matterhorn Protocol. There are 31 checkpoints in the protocol, with each checkpoint having several tests. The Matterhorn Protocol can be found at <https://www.pdfa.org/publication/the-matterhorn-protocol-1-02/>.

Walking through that list and checking your file can be very time consuming. Fortunately, there are some tools that implement the Matterhorn Protocol.

- Adobe Acrobat has an Accessibility Checker tool
<https://helpx.adobe.com/acrobat/11/using/create-verify-pdf-accessibility.html>
- CommonLook has a PDF Validator
<http://commonlook.com/commonlook-pdf-validator-user-guide/>
- Access For All has a PDF Accessibility Checker PAC-2
<http://www.access-for-all.ch/en/pdf-lab/pdf-accessibility-checker-pac.html>

All of these tools are manual. None of them can be run in batch mode, although Adobe allows you to create a process with the ActionWizard to make it a little easier to run the Accessibility Checker on a group of files. For an extra fee, Adobe sells the AutoBatch Plugin for Acrobat (<https://www.evermap.com/AutoBatch.asp>). Adobe says that AutoBatch “adds a command-line batch file functionality to Adobe Acrobat Professional.” The author of this paper has not used AutoBatch.

The Accessibility Checker tool is on the Tools menu in Adobe Acrobat as seen in Figure 11.

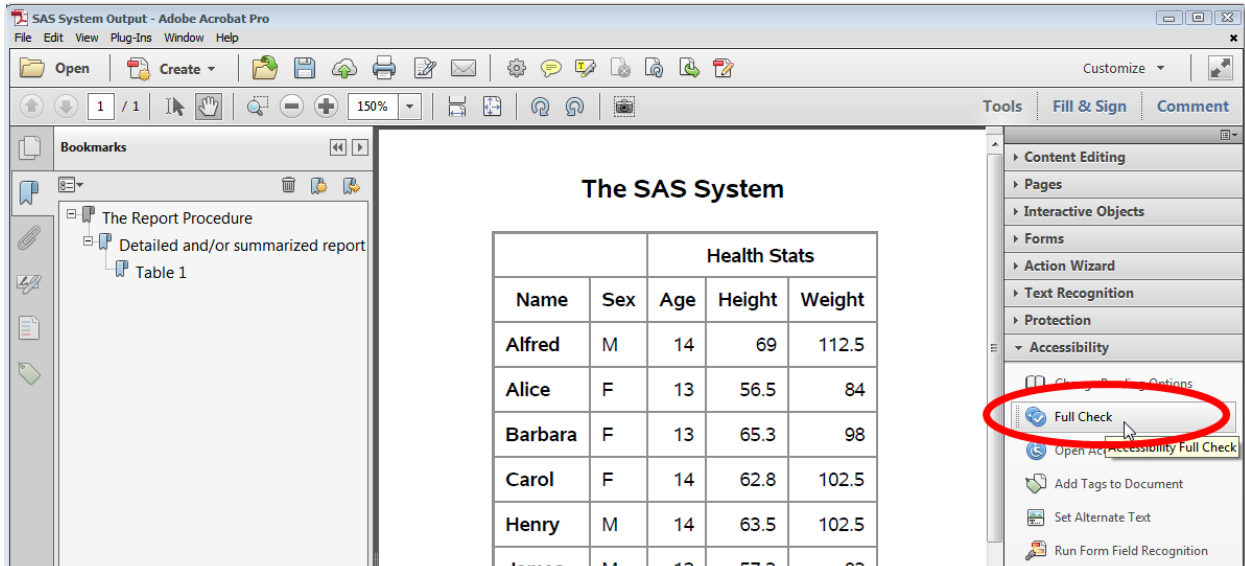


Figure 11 - Adobe Acrobat Accessibility Checker Tool

There are various options you can choose when running the tool as seen in Figure 13, one of which is whether to generate an html version of the report output (which is checked by default).

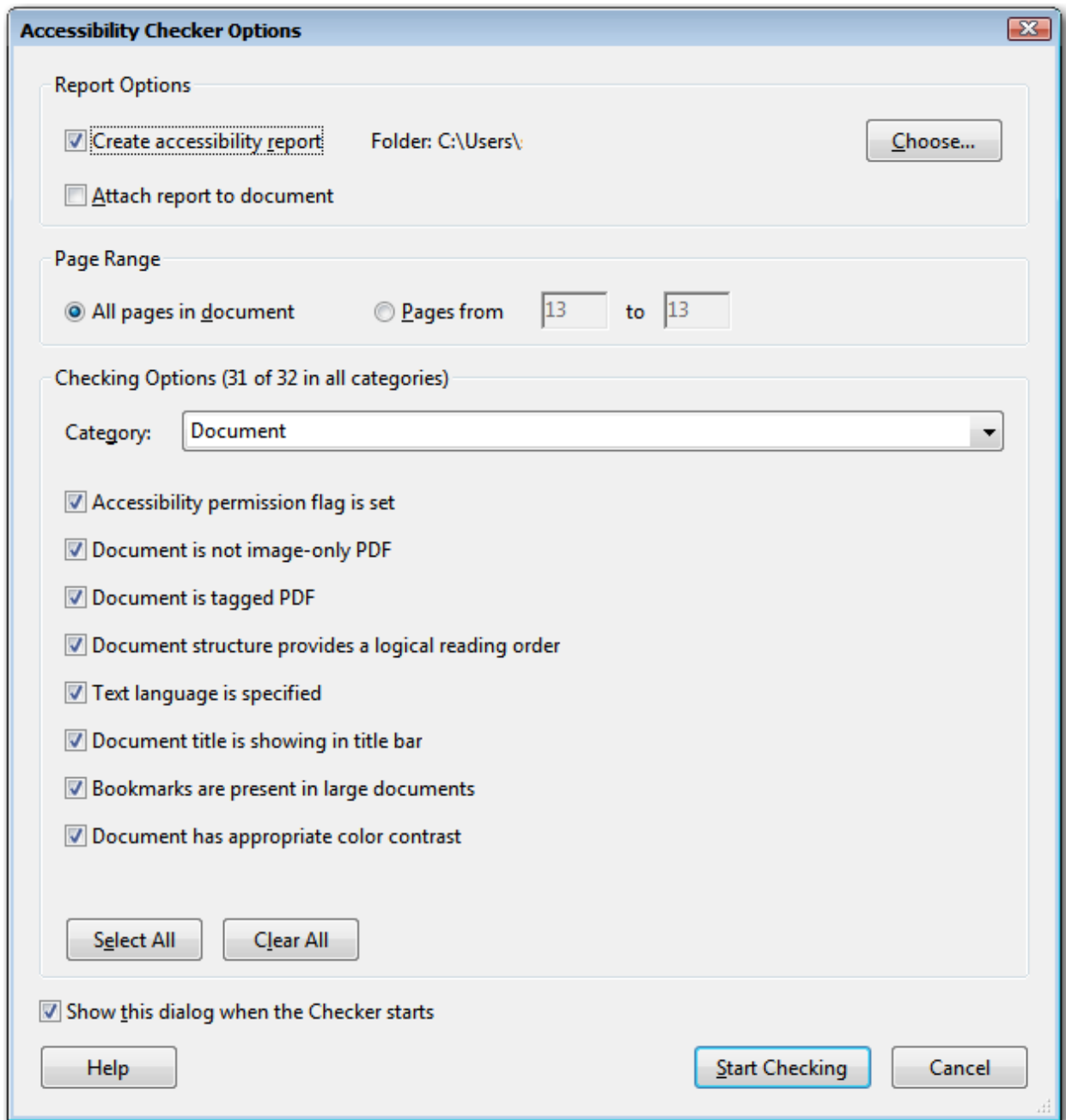


Figure 12 - Adobe Acrobat Accessibility Checker Tool Options

The output from the tool will list possible errors. In this example, there are no errors but the tool has suggestions for items to check manually, such as color contrast issues as seen in Figure 13.

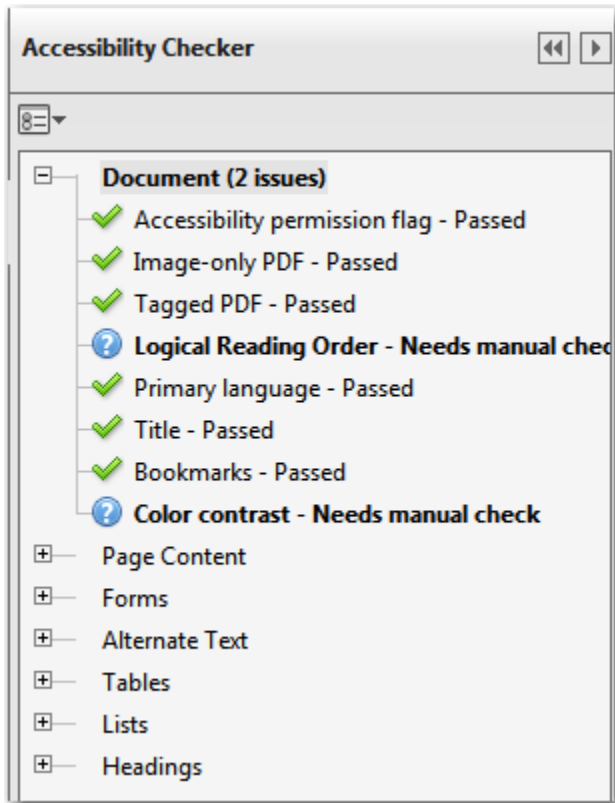


Figure 13 - Results of Adobe Acrobat Accessibility Checker Tool

It's also recommended that you view the tags panel in Adobe Acrobat. Figure 1, Figure 5, and Figure 7 all show the tags panel. As you navigate through the tree in the left panel, there's a visual outline around the associated text in the document. You can quickly navigate through the tree using the up and down arrows (and left and right arrows to expand and collapse the nodes) and watch what's highlighted in the document. You can scan the tree very quickly by holding down the down arrow key to move swiftly through the tree and watching the document on the right scroll by with the blue highlight. This is only practical for a handful of documents.

There are no tools that the author has found that can guarantee your PDF is tagged properly. The tools might find errors, but they don't check if the right tag is used. For example, in the PROC REPORT example in Figure 7 where the NAME variable was used as the row header by using the RowHeader style, the resulting tags were not <th>. They came out as <td> instead. The above tools will not catch that. Only a manual inspection will.

However, if you can verify that some of your documents are generated correctly, and the remaining 10's, 100's, or 1000's of documents you generate use the same process, then that's a good indication that the remaining files will be good too.

CONCLUSION

PDF files are wonderful for printing and often look great when viewing on a computer, but not all PDF files are accessible to everyone. Without extra metadata to tag the content, a PDF is not readable by a screen reader. The new **preproduction ACCESSIBLE** option in SAS 9.4M4 for the ODS PDF statement can help solve that problem.

The ODS development team would like to make the ACCESSIBLE option production in the next maintenance release of SAS. To do so, they would like your input regarding the necessary features required by your site to make accessible PDF content.

You can do this by trying out your production code using 9.4M4. You can request 9.4M4 from the following site: <http://support.sas.com/software/maintenance/index.html>.

Once you have it installed, add the ACCESSIBLE option to your ODS PDF statement, or change your SAS registry settings, then examine the PDF files using the tools mentioned above. Work with your accessibility coordinators to verify the files. If there are PROCs that you use often that don't seem to be generating correct output, send us feedback. SAS has hundreds of procedures and we want to prioritize the ones we work on to make them accessible. Having your feedback regarding the PROCs that are important to you will help.

Feedback can be sent to accessibility@sas.com or you can contact technical support (support.sas.com)

REFERENCES

"ISO 14289-1:2014." ISO. Available

http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=64599. Accessed on February 21, 2017.

"Web Content Accessibility Guidelines (WCAG) 2.0". W3C. Available <https://www.w3.org/TR/WCAG20/>. Accessed on February 21, 2017.

Walker, Glen. 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 <http://support.sas.com/resources/papers/proceedings16/SAS5280-2016.pdf>.

"WCAG mapping to PDF/UA". Adobe Support. Available

<https://helpx.adobe.com/acrobat/11/using/create-verify-pdf-accessibility.html#WCAGmappingtoPDFUA>. Accessed on February 21, 2017.

"WCAG 2.0 and PDF/UA – Your Questions Answered". CommonLook. Available

<http://commonlook.com/WCAG-20-and-PDF-UA-Your-Questions-Answered/>. Accessed on February 21, 2017.

ACKNOWLEDGMENTS

Special thanks to the ODS team for being patient with my questions and working to improve the accessibility of ODS: Woody Middleton, David Kelley, Dan O'Connor, and Bari Lawhorn.

RECOMMENDED READING

- "The Matterhorn Protocol 1.02." PDF Association. Available <https://www.pdfa.org/publication/the-matterhorn-protocol-1-02>. Accessed on February 21, 2017.
- Kelley, David W. 2017. "Escape the Desktop with ODS EPUB." *Proceedings of the SAS Global Forum 2017 Conference*, Cary, NC: SAS Institute Inc. Available <http://support.sas.com/resources/papers/proceedings17/SAS377-2017.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 <http://support.sas.com/resources/papers/proceedings17/SAS557-2017.pdf>.
- Walker, Glen. 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 <http://support.sas.com/resources/papers/proceedings16/SAS5280-2016.pdf>.
- "Create and Verify PDF Accessibility (Acrobat Pro)". Adobe Support. Available <https://helpx.adobe.com/acrobat/11/using/create-verify-pdf-accessibility.html>. Accessed on February 21, 2017.
- "CommonLook PDF Validator – User Guide". CommonLook. Available

<http://commonlook.com/commonlook-pdf-validator-user-guide/>. Accessed on February, 21, 2017.

- “Access For All PDF Accessibility Checker PAC-2”. Zugang Für Alle (Access For All). Available <http://www.access-for-all.ch/en/pdf-lab/pdf-accessibility-checker-pac.html>. Accessed on February 22, 2017.

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.