Paper SAS2129-2018

PDF Accessibility: How SAS® 9.4M5 Enables Automatic Production of Accessible PDF Files

Woody Middleton, SAS Institute Inc., Cary, NC

ABSTRACT

No longer do you need to begin the accessibility challenge with a raw, inaccessible PDF file. SAS® 9.4M5 now provides a framework for creating accessible PDF files. This paper will not only explain the 'automatic' accessible features provided by SAS, but show programming changes to improve the accessibility of the PDF files.

SAS creates PDF files that could best be described as uninformative to those with low or no vision. Now with SAS 9.4m5, SAS provides a framework for creating accessible PDF files. This paper will not only explain the 'automatic' accessible features provided by SAS but show "good practices" to improve the accessibility of the PDF files you create.

INTRODUCTION

The material in this paper is general in nature because each of you likely have specific coding standards to follow. Because customers have focused implementation guidelines, our standard accessible output might not meet your particular needs. Please use and/or alter these suggestions to fit your coding applications. The SAS 9.4M5 code now supports Web Content Accessibility Guidelines (WCAG) standards. *(For questions about SAS 9.4M5 accessibility conformance with WCAG 2.0 Level AA, please contact accessibility@sas.com).*

STARTING THE PROCESS

Begin the process of creating accessible PDF files by using an option from the ODS PDF destination statement, as is show here:

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

The ODS PDF statement with the ACCESSIBLE option specified creates a PDF file that is tagged. The tags in a PDF file provide information that a screen reader uses to read the contents of the PDF file to a blind or low vision consumer. The tags included in a PDF file will affect the size of the PDF, but will not affect the visible results of your SAS output. For sites requiring all PDF files generated by SAS to be accessible, your system administrator can change the default behavior of all PDF files via the SAS registry. This process requires changing the Output Delivery System (ODS) printer registry keys. The steps to do this are as follows:

1. Create a file to import into the printer registry (ex. accReg.txt). The file should contain these lines:

```
[ods\destinations\printer\pdf]
"accessible"="on"
```

2. Run the following SAS code to import the file accReg.txt:

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

The SAS registry will then be updated to use the ACCESSIBLE option when PDF files are created by ODS. To update the SASHELP registry, the above process requires the USESASHELP option on the PROC REGISTRY code and must be run by someone with Update access to the SASHELP library.

BEST PRACTICES FOR ACCESSIBLE PDF DOCUMENT CREATION

USE PROC ODSTEXT TO ORGANIZE AND LABEL THE DOCUMENT

Use the ODSTEXT procedure to add headers and create a customized table of contents. H-H6 headers allow easier navigation of the document by providing section and sub-section labels for the screen reader.

A TABLE WITH COLUMN HEADERS DEFINED, AND A LABEL PROVIDED FOR THE TABLE

In this example, PROC ODSTEXT provides the header information describing the document. The header styles H – H6 and readable tags such as paragraph (P) are supported, as well as many others.

```
ods _all_ close; ods pdf file="print01.pdf" accessible;
title1;

proc odstext;
   h1 'Honda Cars';
run;

proc print data=sashelp.cars label noobs contents="Honda Cars";
   label MPG_City="MPG City";
   var Model MSRP MPG_City;
   where make="Honda";
run;
ods _all_ close;
```

USE APPROPRIATE DESCRIPTIONS FOR YOUR TABLES AND IMAGES

Whenever possible, use clear and concise procedure descriptions to provide as much information as possible for the document reader. Most SAS procedures can provide a description of the procedure output, which is then rendered as the alternate text for the procedure. By default the “proclabel” offered by all procedure is “The <insert proc name here> Procedure”. A consumer listening to a PDF via a screen reader has no interest nor context when they hear a table described as “The PRINT Procedure”. The following are some examples of how to improve the caption offered to the screen reader.

ODS GRAPHICS PROCEDURES

The ODS Graphics procedures (SGPLOT, SGSCATTER, SGPANEL, SGMAP) all provide the DESCRIPTION=”” option to denote the alternate text. Some, but not all, procedures accept the abbreviated form “DESC=””. We recommend using the full word for procedure descriptions.

Here is a simple ODS Graphics example using the DESCRIPTION= option to provide a more detailed description of the graphic data.

PROC SGPLOT with a DESCRIPTION= Statement

```
  ods _all_ close;

  ods pdf file=pdf nogtitle startpage=no accessible;

  title "Hello world!";
  title2 "Output:  Using a description to explain the contents of a SGPLOT
example";

  proc sgplot data=sashelp.cars description="SGPLOT output describing the
sum of cylinders by car type. ";
    vbar type / response=cylinders;
  run;
  ods _all_ close;
```

With PROC SGRENDER, you can also use the alias OBJECTLABEL=. For example:

```
OBJECTLABEL= ’A Better Description than the default, The SGRender Procedure’;
```

PROC SGRENDER with a DESCRIPTION= Statement

```
options nodate nonumber;

ods graphics on / width=3.25in height=3.25in
    imagefmt=static
    tooltipmax=5;

data data;
   infile datalines;
   input sex $ 1-6 cod $ 10-29 deaths 34-38 age_group $ 40-45;
   datalines;
Female   Cerebrovascular          1592 50-60
Female   Heart Disease            3500 50-60
Female   Cancer                  10492 50-60
Male     Cerebrovascular          9667 50-60
Male     Heart Disease           12089 50-60
Male     Cancer                   8449 50-60
Male     Respiratory Disease      7400 50-60
Female   Cerebrovascular          1592 60-65
Female   Heart Disease            3500 60-65
Female   Cancer                  10492 60-65
Male     Cerebrovascular          9667 60-65
Male     Heart Disease           12089 60-65
Male     Cancer                   8449 60-65
Male     Respiratory Disease      7400 60-65
;
run;

proc sort data=data out=data1;
  by sex;
run;

proc template;
  define statgraph test;
  BeginGraph;
  entrytitle "Test for Description=";
  layout datalattice columnvar=sex rowvar=age_group /
           cellheightmin=20 cellwidthmin=20
           rowaxisopts=(label=' ')
           columnaxisopts=(label='# of Deaths')
           headerlabeldisplay=value
           shrinkfonts=true
           columns=2
           columndatarange=unionall;
      layout prototype;
         barchartparm y=deaths x=cod /
              orient = horizontal;
      endlayout;
   endlayout;
endGraph;
end;
```

```
ods _all_ close;

ods pdf style=printer accessible startpage=no;

proc sgrender data=data1 description=”A Better Description than the default, The SGRender
Procedure”
     template=test;
run;
ods pdf close;
```

TABLE DESCRIPTION

Use the CONTENTS= argument to provide a description of the data presented in the tables created by the following:

- PROC PRINT
- PROC REPORT
- PROC TABULATE

Use the OBJECTLABEL= option for Report Writing Interface tables.

PROC PRINT: BEST PRACTICES

PROC PRINT is the workhorse of the SAS tool box. It provides the ability to look at the detail records of your SAS data sets, along with summarizations. The following ‘best practice’ list describes how you can maximize the readability of your table by a screen reader.

As previously mentioned, use the CONTENTS= option to provide a description of the data presented in the tables.

A table with row (descriptive header) and column headers defined, a label provided for the table, and a descriptive summation row.

```
ods _all_ close;

ods pdf file="print03.pdf" accessible;
title1;

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

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

ods _all_ close;
```

HINTS FOR TABLE FORMATTING

Do not use PROC PRINT’s NOSUMLABEL option, because this option disables the labels.   Without the label screen readers cannot convey the context of the summation row. Summation rows should only be used when a BY variable is used so that a proper label can be generated.

Do not use the BLANKLINE option to insert blank lines. Blank lines make it more difficult for screen reader users to understand the contents of the table.   Excessive spaces or blanks makes the contents of the table data more difficult to understand.

Avoid using the N option for displaying the number of results. This option adds an additional table row that holds only a single cell, which can confuse screen reader users as to its meaning. This N information should be presented outside of the table structure (example in Greg's paper – add it to yours too?).

If your PROC PRINT code includes a BY statement, use a corresponding ID statement for the same variable so that each table is clearly labeled.

BY VARIABLE'S TABLE WITH SUMMATION ROW INCLUDING DESCRIPTIVE SUMMATION LABELS

The summation row has a descriptive label from the #BYVAL() which allows users to identify the information being presented in the row.

```
ods _all_ close;

ods pdf file="print05.pdf" accessible;
title1;

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

proc odstext;
   h1 'Total Home Runs by American League Teams in 1986';
run;

proc print data=baseballSort label sumlabel="Total Home Runs for
#BYVAL(Team)" contents="Home Runs by American League Team in 1986" noobs;
   where League="American";
   by Team;
   id Name;
   var Position nHome;
   label Name="Player" Team="Team" Position="Position" nHome="Home Runs";
   sum nHome;
run;

ods _all_ close;
```

If you want to have defined row headers, either omit the NOOBS option, or include one of the variables in an ID statement.

PROC TABULATE: BEST PRACTICES

PROC TABULATE is the workhorse of any SAS Programmer who regularly summarizes data. PROC TABULATE's ability to 'transpose' data by defining a row dimension, and its 'out of the box' ability to calculate unique percentages, makes it a popular procedure with no equal peer. The flexibility over header formatting and the aforementioned ability to transpose the look of your data also provides ways to complicate a screen reader's ability to share the information created by a TABULATE table. With the best practices mentioned below, your tables may undergo visible changes to the sighted consumers of your tables. However, the tables should prove more readable to that audience and will result in tables that a screen reader can logically and reliably convey the information your tables contain.

As previously mentioned, use the CONTENTS= option to provide a description of the data presented in the tables.

HINTS FOR ACCESSIBLE TABLE FORMATTING

Do not use concatenated variables in the row dimension, such as: table a*b, c. Instead, divide the concatenated variables into separate tables by either using a second TABLE statement or using a second PROC TABULATE step. Without these changes, the proper headers for concatenated variables in the row dimension in a single table will not be communicated to screen reader users.

For each variable in the row dimension, assign the label to “”. This will eliminate the row header that is stored as a row span in the first row of each variable instance. Otherwise, screen readers have difficulty reading tables with variables in the row dimension. One alternative is to use BOX=”…” to create a column header that labels the row headers in a way screen readers can more accurately start the table values.

```
ods _all_ close;

ods pdf file="tabulate01.pdf" accessible;

proc tabulate data=sashelp.heart;
   class Sex Status Chol_Status;
   table Chol_Status="", Sex /box="Cholesterol Status";
run;

ods _all_ close ;
```

CUSTOM LABELS FOR ALL COLUMNS

Here, the "All" columns for the largest groupings in the columns (Cholesterol and Blood Pressure) do not have other headers to describe them.Therefore, a custom label is placed on each to allow the screen reader to differentiate between both "All" columns. Also in this example, the "All" for each of the smaller groupings can simply be "All" since there are other headers above the row and column dimension variables that further describe what they are totaling.

```
ods _all_ close;

ods pdf file="tabulate02.pdf" accessible;

proc tabulate data=sashelp.heart;
   class Sex Status Chol_Status BP_Status;
   table Sex all,Status*(Chol_Status all) all="All Cholesterol"
Status*(BP_Status all) all="All Blood Pressure";
run;

ods _all_ close;
```

Avoid the table option NOCELLMERGE. The table generated by this option will result in headers that will not be correctly assigned to their data cells. Without labeling, a screen reader will have no context for the data cells.

The KEYLABEL statement and custom label text for the ALL keyword are recommended to provide a more descriptive label for some columns and rows. This is especially important if there are multiple columns or rows with the same text where the context might not provide enough information to clearly know what information is being presented.

If in a row header or column header there is a single spanned cell going across all of the rows or columns, the programmer does not have to include that label in the table if the title/description of the table adequately describes what is being presented.

```
ods _all_ close;

ods pdf file="tabulate02.pdf" accessible;

proc tabulate data=sashelp.heart;
```

```
    class bp_Status;
    var height weight;
    table (height weight)*mean, bp_status /box="Mean of Height and Weight";
    keylabel mean="This is a keylabel";
run;

ods _all_ close;
```

PROC REPORT: BEST PRACTICES

PROC REPORT is arguably the most flexible tool in a SAS programmer's tool box. PROC REPORT provides detail level information or summarization level information, while also including the ability to use conditional logic and formatting like the SAS DATA step. This flexibility helps PROC REPORT shine where PDF files are concerned. However, as with PROC TABULATE, the formatting ability provided by PROC REPORT's syntax also allows a SAS programmer to create a table that is completely incomprehensible to a screen reader. The following best practices will help you write or recode your current PROC REPORT syntax to create tables that are easily consumed by a screen reader.

HINTS FOR TABLE FORMATTING

As previously mentioned, use the CONTENTS= option to provide a description of the data presented in the tables.

When any variable is assigned the usage ORDER or GROUP, SPANROWS must be specified on the PROC REPORT statement. Using this option in conjunction with GROUP or ORDER usage provides labels so that the screen reader can better describe the content of the grouped or ordered data.

If a value is needed to help uniquely describe the row of data, use DEFINE [col] / GROUP. If you have a DISPLAY variable it is best practice to change it to GROUP.

PROC REPORT Examples:

Simple Detail Report with Table Description

```
ods _all_ close;

ods pdf file="report.pdf" accessible;

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

ods _all_ close;
```

Spanned Column Headers

```
ods _all_ close;

ods pdf file="report2.pdf" accessible;

title1;

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

ods _all_ close;
```

In the following example, an additional row header is defined to help uniquely identify the data on the row (by adding define Model / order;)

```
ods _all_ close;

ods pdf file="report3.pdf" accessible;

title1;

proc report data=sashelp.cars contents="All Cars" spanrows;
   define Make / order;
   define Model / order;
   column Make Model MSRP MPG_City Horsepower;
run;

ods _all_ close;
```

Use the Report Writing Interface (RWI) DESCRIPTION: Argument to Describe the Contents of an Image.

The RWI supports the same accessible tags as the SAS procedures, including tables and images.

Here is an example using the RWI to provide a header for a table:

```
ods _all_ close;

options nodate nonumber;
title;

ods pdf file="report.pdf" accessible=on title="&sysvlong";


ods pdf nobookmarkgen startpage=no;

data _null_;
dcl odsout obj();
    obj.image(file: "<image filename>.jpg",description: "This is an
accessible description of the JPG Image." );
run;

ods pdf bookmarkgen startpage=no;

proc print data=sashelp.cars(obs=1) contents="Sensible table description"
noobs;
var make model type DriveTrain MSRP Invoice;
run;

ods _all_ close;
```

Other Suggestions for Writing Code to Generate PDF Accessible Code.

- When constructing a table, ensure that all the columns of a table are on one line and that the row does NOT continue to the next line (sometimes called table paneling). Continuing a row is difficult for a person with limited sight to follow and could produce a confused voicing of the table data.
- Simplify the use of layout regions. Overly complex layout regions might not be read properly.

CONCLUSION

SAS 9.4M5 uses procedure options to provide a baseline for creating accessible PDF files that can be read by most screen readers. Use the built-in features to create a tagged PDF file for the basic structure of accessible PDF. Work with your compliance team and provide feedback to our team to help us enhance and add to the features in the next SAS release.

ACKNOWLEDGMENTS

Greg Kraus provided the summary information and code examples for PROC PRINT, PROC TABULATE, and PROC REPORT.

Bari Lawhorn provided significant editorial suggestions to improve the flow and content of this paper.

REFERENCES

SAS Institute Inc. 2017. SAS Institute white paper. “Tag, You’re It! Creating Accessible (Tagged) PDF with the Fourth Maintenance Release for SAS® 9.4 Output Delivery System.” Available http://support.sas.com/resources/papers/proceedings17/SAS0483-2017.pdf

SAS Institute Inc. 2018. SAS Institute white paper. “ODS PDF Accessibility in SAS® 9.4M5: Going Beyond the Basics to Create Advanced Accessible Reports.” Available http://support.sas.com/resources/papers/proceedings18/SAS2124-2018.pdf

SAS Institute Inc. 2017. SAS Institute white paper. “A Guru’s Guide: Producing Section 508-Compliant Custom Reports with the Output Delivery System.” Available http://support.sas.com/resources/papers/proceedings17/SAS0557-2017.pdf

CONTACT INFORMATION

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

Woody Middleton
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Woody.Middleton@sas.com
http://www.sas.com