

A Guru's Guide: Producing Section 508-Compliant Custom Reports with the Output Delivery System

Daniel O'Connor, SAS Institute Inc.

ABSTRACT

Creating sophisticated, visually stunning reports is imperative in today's business environment, but is your fancy report really accessible to all? Let's explore some simple enhancements that were made in the fourth maintenance release of SAS® 9.4 to Output Delivery System (ODS) layout and the Report Writing Interface that will truly empower you to accommodate people who use assistive technology. ODS now provides the tools for you to meet Section 508 compliance and to create an engaging experience for all who consume your reports.

INTRODUCTION

Extensive effort and time is spent on authoring beautiful documents that are intuitive and engaging aesthetically. Your bold design, vivid colors, and fancy diagrams engage your audience in an inspirational manner. Often that is where the design strategy ends, leaving someone with a disability neglected. While this is not only socially unacceptable, it might be against the law if you are affiliated with any type of government agency. The Americans with Disabilities Act (ADA) has embraced the Web Content Accessibility Guidelines (WCAG) 2.0 standards to ensure that individuals with any type of impairment have equal access to a document's contents. The term "accessibility" when broken down simply means the ability to access. Designing an accessible document should be an integral part of your initial design strategy. Beginning in the fourth maintenance release of SAS 9.4, ODS has added enhancements to allow the document designer to embrace the guidelines so that your report is not only beautiful, but also accessible to all. The fundamental approach to making a document accessible means the designer needs to add alternative information that will be interpreted by assistive technology such as a screen reader. Often this alternative information is in the form of invisible text describing the visual content. All the new ODS features follow a similar pattern of allowing the document designer to provide alternative information along with the visual content as the document is being created. This paper focuses on the accessibility features of the ODS LAYOUT statement, which allows you to arrange text, tables, and graphs on a page, and the ODS Report Writing Interface (RWI), which is our custom report-writing interface when you need more flexibility than a procedure provides.

WHEN REAL LIFE GETS IN THE WAY

Let's consider a real life scenario. Say you have a visual impairment that requires the use of a screen reader to navigate electronic documents. You need to access your county school system website to enroll your child and want to investigate all the available options. You might ask yourself any number of questions, such as: Is the school year-round? Should I consider a charter school? Do the schools publish ratings? What types of athletics do they provide? What types of transportation are available? Figure 1 is a screen shot of a fictitious Jester County home page that provides general information and a launch point for more detailed information. One of the first rules that should be considered when designing an accessible document is that you need to add an accessible title for the document. ODS destinations have always had a TITLE argument, which provides a non-visible description for the document. At this time, accessible features are supported in the HTML5 destination and are considered pre-production for the PDF destination. The HTML5 destination automatically generates accessible content. However, the PDF destination requires that you turn the feature on because it significantly increases the size of your document. Let's look at the syntax of the PDF ACCESSIBLE argument and the TITLE argument for these destinations:

```
ods html5 path="<some path>" file="JesterCounty.html"  
  (title="Jester County Home Page");
```

ods pdf file="JesterCounty.pdf" notoc **accessible**
title="Jester County Home Page";

ADDING ACCESSIBLE DESCRIPTIONS TO IMAGES

In today's world, web design is as much of an art as a science. Gone are the days where text dominates public web pages. Fancy graphics, videos, and interactive controls are expected in any modern-day web page. However, this presents an enormous barrier for an individual with an impairment. That is to say, sufficient attention has not been dedicated to making the content accessible. In Figure 1, the Jester County home page is dominated by images and contains an informational "Announcements" table. To make it even more difficult for someone using a screen reader, much of the text is embedded in the images. By associating some alternative text with an image, you can provide valuable information that can be consumed by the screen reader.

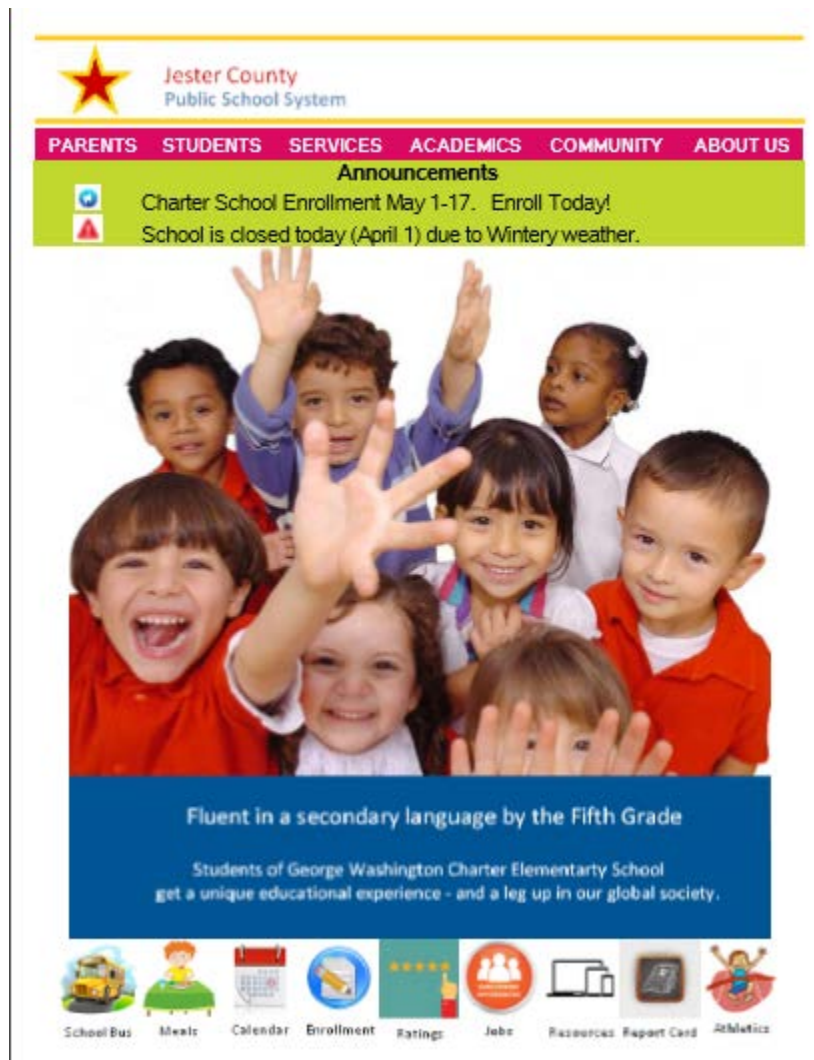


Figure 1. Jester County Public School System Home Page

Let's break this example down a bit and see how the document designer can provide alternate text when displaying an image. In Figure 2 an image is being used as a title/header image, and contains the county's logo and some text.



Figure 2. How to Add Alternative Text to an Image

We are going to use the RWI to insert the title/header image into the document, and associate some alternative text in a single statement. RWI allows you to create highly customized reports from within the DATA step. It lets you combine the powerful programming features of the DATA step with ODS features, giving you flexibility and control for every piece of output in your document. In the following code snippet, we are using the RWI IMAGE method to providing alternative text in the optional description argument. This description will be associated with the image in the document, but will not be visible. The description will be tagged in a manner that assistive technology such as a screen reader will understand, and will read when this image becomes the active item.

```
data _null_;  
  dcl odsout obj();  
  obj.image(file: "../JesterCounty.png",  
           description: "Welcome to the Jester County Public School System  
Home Page");  
run;
```

ADDING ACCESSIBLE DESCRIPTIONS TO TABLES

Next let's look at how a document designer might enhance a table to be more accessible, and explore some idiosyncrasies of screen readers. Tables are typically categorized into DATA tables and LAYOUT tables. The intent of a DATA table is to present tabular information where the rows and columns will likely have headings associated with each row and column, so the reader can understand the context of the data in each cell. On the other hand, LAYOUT tables can be much more unstructured and informational, and might or might not have headings to assist the reader. LAYOUT tables can also be used purely for presentation purposes and provide no useful information to a user using a screen reader. Clearly, using DATA tables is more desirable, but sometimes it just doesn't work out. It is very important to note that screen readers treat DATA tables very differently than LAYOUT tables. Screen readers do a really good job of conveying the information in DATA tables by reading row and column heading information in respect to each cell's data. For LAYOUT tables, your experience depends more on the actual reader that you are using. This makes it even more important that you invest additional effort into enhancing the table's accessibility if you did not intend for it to be purely for presentation purposes. In general, a best practice is to always provide a description of the contents contained in the table before the reader begins navigating the table. ODS has added a new optional description argument when using the RWI TABLE_START method to associate a non-visible description with the beginning of a table. RWI is not the only way to provide descriptions for creating accessible output. Many of our most common reporting procedures also have options to provide descriptions for accessibility, but I will just list the options in the following table for each PROC and a sample of some of the default descriptions. In most cases, the default descriptions do not provide adequate information for the reader, so it is incumbent on the document designer to ensure that adequate descriptions are provided.

PROC	OPTION	Default Description
DATA_NULL_(RWI)	DESCRIPTION	Data NULL TABLE
FREQ	CONTENTS	One-Way Frequencies
REPORT	CONTENTS	Detailed and/or summarized report
PRINT	CONTENTS	Data Set <data set name>
TABULATE	CONTENTS	Crosstabular summary report
SAS/GRAPH	DESCRIPTION	Bar chart of <variable>
SG PROCS	DESCRIPTION	The SG**** Procedure

Now that we have a better understanding of the types of accessible tables, let's look at our "Announcements" informational table in Figure 3. What type of table did we use? This is a LAYOUT type table despite the fact that the data is coming from a data source. The data itself is more unstructured and informational and does not contain specific row and column headings. However, this table is not intended to be purely for presentation purposes. Therefore, we should provide some alternate text describing the contents of the table.



Announcements	
	Charter School Enrollment May 1-17. Enroll Today!
	School is closed today (April 1) due to Wintery weather.

Figure 3. How to Make a Table More Accessible

The following RWI code produces the "Announcements" table in Figure 3, and provides an alternative text on the RWI TABLE_START method.

```

data _null_;
length image_path $ 256;
set announcements end=eof;

if ( _N_ eq 1 ) then do;
  dcl odsout obj();
  obj.table_start(
    description: "This is a list of the most recent public service
announcements.",
    style_attr: "frame=void
                rules=none
                cellspacing=0
                cellpadding=0
                width=8in");

  obj.head_start();
  obj.row_start();
  obj.format_cell(data: "Announcements",

```

```

                column_span: 2,
                style_attr: "font_size=16pt background=#c1d82f");
        obj.row_end();
        obj.head_end();
end;

if ( strip(ann_type) eq "General" ) then
    image_path = "./bullhorn8.png";
else if ( strip(ann_type) eq "Alert" ) then
    image_path= "./alert8.png";

obj.row_start();
obj.format_cell(
    data: "",
    style_attr: "preimage=' ' || strip(image_path) || ' ' background=#c1d82f");
obj.format_cell(data: ann_desc,
                style_attr: "font_size=16pt just=L background=#c1d82f");
obj.row_end();

if ( eof eq 1 ) then do;
    obj.table_end();
end;
run;

```

The announcements are coming from a data source that contains two variables, ANN_TYPE and ANN_DESC. The ANN_TYPE is used to determine what type of image is displayed in the first column of the table, and the ANN_DESC is the actual text of the announcement to be displayed in the second column. Let's first focus on the table heading "Announcements" that spans the entire table. Most screen readers treat a table without any table headings as only a presentational LAYOUT table even if a description has been provided. My initial thought from a design perspective was to not provide a heading at all, but that would mean that this table would be completely ignored by the screen readers. Because my intentions were for the screen reader read this information, I was required to provide a heading of some sort.

Next let's look at the first cell, which contains only an image. We are familiar with the best practice of always providing a description with an image from our earlier example through the RWI IMAGE method. However, in this case we are using the RWI TABLE methods to produce a stand-alone table, and I am inserting an image into a cell through the use of the style_attr override argument. So, I know what your next question is: How does the screen reader interpret an image with no alternate description? We simply script the image name as the alternate text and hope that the name of the image can convey some meaningful context.

Let's now focus on the next item in the report, the image of the children and text. This is accomplished simply by using the RWI Image method in the same manner as in the first example. In this particular case, the visible text is actually part of the image, so a description is required if you want to convey similar information to a reader using a screen reader.

```

        data _null_;
        dcl odsout obj();
        obj.image(file: './kidscombo90.png',
                description: "Fluent in a secondary language by the Fifth Grade.
Students of George Washington Charter Elementary School get a unique
educational experience - and a leg up in our global society.");
run;

```

ODS LAYOUT FOR PRESENTATION PURPOSES

Finally, let's look at the images that appear to look like buttons at the bottom of the page and are arranged in a grid-like fashion. ODS LAYOUT provides a document designer a straightforward way to arrange content in a variety of ways. ODS LAYOUT has two distinct forms, absolute and gridded. ODS LAYOUT is typically intended to be only for presentation purposes and will be ignored by the screen reader. However, we might still want the contents of the layout to be acknowledged by screen readers.



Figure 4. ODS LAYOUT for Presentation Purposes

The following code snippet uses ODS LAYOUT to produce Figure 4.

```
ods layout gridded columns=10 rows=1 row_gutter=0 column_gutter=0;
ods region;
  data _null_;
  dcl odsout obj();
  obj.image(file: "../TransportationSmall.png",
           description: "School Transportation");
  run;
.
.
.

ods region;
  data _null_;
  dcl odsout obj();
  obj.image(file: './AthleticsSmall.png',
           description: "Athletics");
  run;
ods layout end;
```

As you can see, the ODS LAYOUT and REGION statements define the gridded nature of the presentation, and the RWI Image method provides the contents for each region with some alternate text for the screen reader to consume.

SECTIONS AND ACCESSABLE ODS LAYOUTS

Despite the fact that ODS LAYOUT is primarily designed for presentation purposes does not mean that it cannot provide a value when designing accessible output. Let's look at another example that uses ODS LAYOUT to group accessible content into navigable sections that contain additional alternate text to describe the theme of each section. Here is a screen shot of the second example.

GEORGE WASHINGTON CHARTER SCHOOL

1 Presidential Ave
Cary NC 27511

(919) 123-4567
<http://www.Jester.co.org>

Grade Range: Pk-8
Charter School
Traditional Calendar

OVERALL SCHOOL PERFORMANCE GRADE

Achievement indicators	Score
Reading 40% proficiency	51
Math 40% proficiency	51
Science 40% proficiency	50
Biology 40% proficiency	
Math 1 40% proficiency	

Growth Status
Exceeded

	Score	Grade
Achievement	50	
Growth	85.0	
School Performance	59	C
40% Reading	55	C
40% Math	57	C

Formula for determining the School Performance Grade:
80 percent of the School Performance Grade is based on the school achievement score. The school achievement score is calculated using a composite method based on the points earned by a school on all of the tests measured for that school.
20 percent of the School Performance Grade is based on academic growth.

A-NG Schools: A-NG schools earned a score of 85-100 and do not have significant achievement or graduation gaps.

GRADING SCALE	
Letter Grade	Grade Range
A	80-100
B	70-80
C	60-69
D	50-59
F	Below 50

SCHOOL PROFILE

School Size: The total number of students in this school and the average number of students in schools with similar grade ranges at the district and state levels.

Our School	328
District	323
State	482

School Attendance: The average percentage of students who attend school daily.

Our School	83.6%
District	84.1%
State	82.1%

Average Class Size: The average number of students enrolled in the "typical" K-8 classroom.

	Kindergarten	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Grade 6	Grade 7	Grade 8
Our School	17	18	18	17	22	27			
Our District	22	20	18	22	21	21			
State	18	18	18	19	20	20			

* Legislation mandates that class sizes for grades 4-12 are not restricted.

Figure 5. Using ODS LAYOUT to Establish Sections in Your Document

Document designers often break the document contents down into sections, and nested subsections to clearly organize related material. This is especially important if you are designing an accessible document because the WCAG standard provides for keyboard navigational actions to allow the user to traverse the page much faster. For example, a user is looking for the information in the "School Profile"

section in Figure 5. Without any sections, users have to traverse the entire document word for word to find the desired information. Clearly establishing sections not only makes the document more visually appealing, but it also significantly improves the accessibility. Establishing formal sections and sub-sections is a new concept for ODS. Hence, we have introduced the HEADER statement in PROC ODSTEXT to allow you to do so. The HEADER statement will allow you to create up to six levels of header sections or nested sub-sections. Let's look at how we used it in this example.

```
proc odstext;
  h1 "GEORGE WASHINGTON CHARTER SCHOOL" / style={foreground=#CC0001
                                                font_size=16pt};
run;
```

Not only does this create a nice visual heading, but internally it is scripted in a manner in which the screen reader automatically understands the hierarchy of the document and allows the user to quickly navigate the document. Always start with the h1 statement, and use the "h2...h6" format to create increasingly nested sub-sections.

```
proc odstext;
  h2 "OVERALL SCHOOL PERFORMANCE GRADE" / style={foreground=#CC0001
                                                font_size=16pt};
run;
```

Now that we have established a hierarchy of section headers, we need to organize the supporting section information, and that is where ODS LAYOUT can assist us in making the document even more accessible. Similar to the RWI TABLE_START method, ODS LAYOUT has also added an optional description argument to both the LAYOUT and REGION statements to provide embedded accessible content that the screen reader will acknowledge. When ODS LAYOUT is used without the description option, the layout is intended only for presentation purposes. In Figure 5 each section's content is contained in an ODS LAYOUT, and a description was provided to describe the entire section's content as a whole. In this particular case, the REGION statements are only for presentation purposes, so the description option is omitted. Let's look at the second section and the accessible code enhancements.

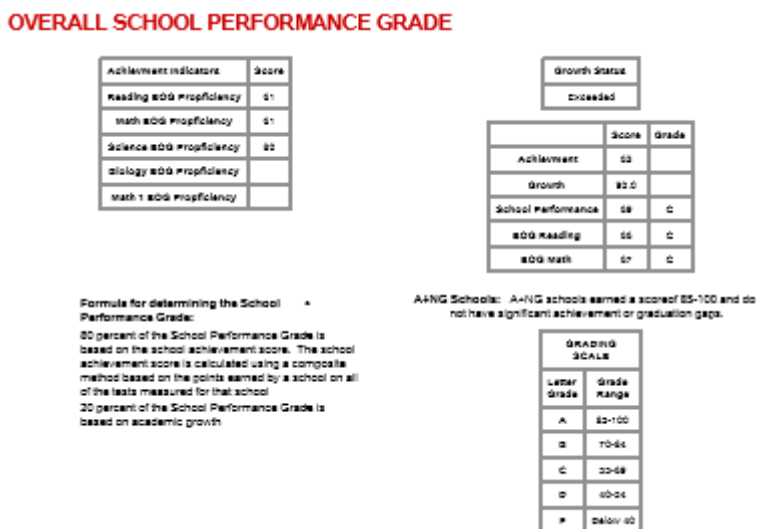


Figure 6. Providing Accessible Information to an ODS Layout Section

The following code show how to provide an accessible description on an ODS Layout statement.

```
ods layout gridded columns=2 column_widths=(50% 50%)  
  description="School Performance Grade Section which Established End of  
Grade (EOG) Proficiency Scores, and the overall grade for the school in all  
class 1 subject matter.";
```

In addition to the summary description in the LAYOUT statement, additional alternate text was provided on each of the RWI TABLE_START methods similar to approach taken in Figure 3. The “School Profile” section used the identical approach as the “Overall School Performance Grade” section by providing a detailed description in the LAYOUT statement to give an overall description of the entire section.

CONCLUSION

ODS LAYOUT and the Report Writing Interface (RWI) have added the description option in a variety of places to allow the document designer to provide alternate descriptions for use with screen readers. In addition to ODS LAYOUT and RWI, there are already existing features on two destinations and some on procedures that allow you to provide alternative descriptions in a similar manner. Organizing your document into sections not only makes your document more visually appealing, but it also make it much more accessible. However, the most important understanding is that accessibility should always be included as part of your initial design methodology.

REFERENCES

SAS® 9.4 SAS Output Delivery System: User's Guide.

Creating Accessible SAS® 9.4 Output Using ODS and ODS Graphics

CODE

<http://support.sas.com/resources/papers/proceedings17/557-2017>

ACKNOWLEDGMENTS

The author would like to thank the following SAS employees for carefully reviewing the paper:

Glen Walker
Allison Crutchfield
David Kelley
Woody Middleton
Bari Lawhorn
Daniel Kummer

RECOMMENDED READING

- Kelley, David W. “Escape the Desktop with ODS EPUB.” *Proceedings of the SAS Global Forum 2017 Conference*. Cary, NC: SAS Institute Inc.
- Walker, Glen. “Raising the Bar: The Pending Section 508 Refresh and the ODS HTML5 Destination.” *Proceedings of the SAS Global Forum 2017 Conference*. Cary, NC: SAS Institute Inc.
- Walker, Glen. “Tag, You’re It! Creating Accessible (Tagged) PDF Files with SAS® 9.4 Output Delivery System (ODS).” *Proceedings of the SAS Global Forum 2017 Conference*. Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

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

Daniel O'Connor
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Dan.OConnor@[sas.com](mailto:Dan.OConnor@sas.com)
<http://www.sas.com>

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.