

Differences in ODS formatting for HTML with Proc Print and Proc Report

Laura L. M. Thornton, USDA-ARS, Animal Improvement Programs Laboratory,
Beltsville, MD

ABSTRACT

While Proc Print is a terrific tool for data checking and data summary, Proc Report is similar to, but more powerful because it can do some basic calculations or statistics and data can be broken more effectively into manageable fields. The ODS procedure can produce HTML files from either procedure and formatting for output is slightly different for each. This presentation is meant to explore a few of the differences in the formatting applied by ODS to each procedure and methods to produce similar HTML output. It is not all inclusive, but this brief demonstration will deal specifically with id, class, and by statements within each for output to HTML by ODS.

INTRODUCTION

When reporting results from SAS output to various audiences, often a more sophisticated report is required than the basic Proc Print output. While Proc Print can be forced to produce desired output, Proc Report is a better reporting tool, allowing greater flexibility in handling outputs. However, when using ODS to publish outputs, whether to HTML or another file, differences in the handling of certain coding statements occurs. The purpose of this paper is to illustrate how each procedure works with ODS and illuminate the differences in code handling in order to allow for better selection of proper reporting tools.

OUTPUT DELIVERY SYSTEM

Various methods to produce sophisticated output exist, but ODS makes it easy to output to HTML, Excel, or other files. The HTML application produces ready to publish HTML code and provides source information for easy additions or alterations to web pages. Simple ODS code requires opening and closing arguments and can be used with most SAS procedures. A simple example of how ODS works is:

```
ods path work.template(update) sashelp.tmplmst(read) library.templat(read);
ods html file = "pathname";
    style = stylesheetname;
title "New data sheet";
Proc Print data=one_set;
    id var1 var2 var3;
run;
Proc Report data = two_set;
    column var4 var5 var6;
    define var4 / "(no.)";
run;
ods html close;
```

The code contained within the ODS output clause is relatively simple to put together as it uses code already known. The only changes are the inclusion of paths for files and templates and a style sheet guideline, if one exists. The ODS environment must be opened and closed. Additionally, with ODS HTML, HTML3 can be used to revert to the HTML 3.2 formatting (which may be desirable when using older style sheets).

PROC PRINT

Proc Print is often used to do simple reporting of results, but can be used to create HTML tables when used in conjunction with ODS. Use of id and by statements helps to produce some desired outputs. The use of the id statement sets up the columns that will be displayed and the style sheet in use defines how those columns will be displayed. The first column is designated as a column, rather than as row delineation. The class statement is used to delineate between different groups (or classes) of variables. The by statement can also be used as an additional separation scheme similar to how it is used in Proc Means or other SAS procedures; however, it is of note that data sets must be sorted by the variables in the by statement for by to be used properly.

Using the previous example and expanding on the level of sophistication in the output:

```
ods path work.template(update) sashelp.tmplmst(read) library.templat(read);
ods html file = "pathname";
    style = stylesheetname;
title "New data sheet";
Proc Print data=one_set;
```

```

id var1 var2 var3;
by vara varb;
class varx vary;
run;
ods html close;

```

This would produce a nicely styled table with little additional formatting in simple type face set to those defined in the style sheet. No individual control of columnar or row formatting is available within the Proc Print command. The output would be divided into parts of vara and varb (the by variables) and each division would have further partitions of varx and vary (the class variables) with columns of var1, var 2, and var 3 (table 1). While not particularly spectacular, it is functional for simple reporting.

Table 1. Simple Proc Print output using previously defined style sheet.

| Vara | | | |
|-------------|-------------|------------------|------------------|
| | | | |
| Varx | Var1 | Var2 | Var3 |
| | Ob1 | Ob1 a | Ob1 b |
| | Ob2 | Ob2 a | Ob2 b |
| | Ob3 | Ob3 a | Ob3 b |
| | Ob4 | Ob4 a | Ob4 b |
| Vary | Ob1 | Ob1 a | Ob1 b |
| | Ob2 | Ob2 a | Ob2 b |
| | Ob3 | Ob3 a | Ob3 b |
| | Ob4 | Ob4 a | Ob4 b |
| Varb | | | |
| Varx | Var1 | Var2 | Var3 |
| | Ob1 | Ob1 a | Ob1 b |
| | Ob2 | Ob2 a | Ob2 b |
| | Ob3 | Ob3 a | Ob3 b |
| | Ob4 | Ob4 a | Ob4 b |

| Vara | | | |
|-------------|-------------|------------------|------------------|
| Varx | Var1 | Var2 | Var3 |
| Vary | Ob1 | Ob1 a | Ob1 b |
| | Ob2 | Ob2 a | Ob2 b |
| | Ob3 | Ob3 a | Ob3 b |
| | Ob4 | Ob4 a | Ob4 b |

PROC REPORT

Proc Report allows for greater flexibility in reporting results than Proc Print. Additionally, the syntax is similar to that used for Proc Print, but does require additional definitions to set line style, font, type face, and color scheme. Generally, a style sheet will set each of those parameters, but use of the define statement can override individual parameters. Columns are defined with the use of the column statement rather than the id statement, but data sets may be still divided by class and by statements.

Again, using the simple example provided previously to expand the level of sophistication:

```
ods path work.template(update) sashelp.tmplmst(read) library.templat(read);
ods html file = "pathname";
    style = stylesheetname;
title "New data sheet";
Proc Report data = two_set;
    column var4 var5 var6;
    by varc vard;
    class varx1 vary1;
    define var4/"category" style(column)=[font weight=bold
        foreground=cx01541c background=cxedf2ed fontsize=3];
define var5 / "(no.)";
run;
ods html close;
```

The inclusion of the style clause for var1 forces the individual handling for that column to be different than all other columns defined by the column statement. However, it produces a nicely formatted table where the first column is a different font and color than the remaining columns, which is desirable when the first column is not data but a reporting category such as state or year. It appears identical to the output produced by the Proc Print command. (table 2)

Table 2. Simple Proc Report output using predefined style sheet and columnar modifications.

| Varc | | | |
|--------------|-------------|------------------|------------------|
| Varx1 | Var4 | Var5 | Var6 |
| | Ob1 | Ob1 a | Ob1 b |
| | Ob2 | Ob2 a | Ob2 b |
| | Ob3 | Ob3 a | Ob3 b |

| Varc | | | |
|--------------|-------------|------------------------|------------------------|
| Varx1 | Var4 | Var5 | Var6 |
| | Ob4 | Ob4 a | Ob4 b |
| Vary1 | Ob1 | Ob1 a | Ob1 b |
| | Ob2 | Ob2 a | Ob2 b |
| | Ob3 | Ob3 a | Ob3 b |
| | Ob4 | Ob4 a | Ob4 b |
| Vard | | | |
| Varx1 | Var4 | Var5 | Var6 |
| | Ob1 | Ob1 a | Ob1 b |
| | Ob2 | Ob2 a | Ob2 b |
| | Ob3 | Ob3 a | Ob3 b |
| | Ob4 | Ob4 a | Ob4 b |
| Vary1 | Ob1 | Ob1 a | Ob1 b |
| | Ob2 | Ob2 a | Ob2 b |
| | Ob3 | Ob3 a | Ob3 b |
| | Ob4 | Ob4 a | Ob4 b |

DIFFERENCE

With Proc Print, the style sheet defined is used in whole and column 1 is defined as a "title" column rather than a data column with the use of the id statement. The id statement is not available for use in Proc Report; instead the column statement defines individual columns.

Because Proc Report allows for special handling and override of the style sheet parameters, individual columns are handled as pieces of the whole and style sheet parameters are only applied where parameters are not redefined. In

particular, column 1 is no longer defined as a "title" column, instead it becomes strictly data reporting and is handled as such. The formatting of the column becomes data oriented and must be altered on a column to column basis within Proc Report, as illustrated in the code above.

CONCLUSION

In general, Proc Report is more manageable for producing clean, concise output reports for specific audiences. However, in using ODS to publish the reports, special care must be taken to ensure correct coding for columnar formatting. Use of by and class statements with Proc Report assists in making the formatting easier, but it is not fail proof. When outputting a simple report with minimal formatting requirements, Proc Print with an id statement is an easier and more concise programming tool, but it has many limitations in what can be done to improve formatting.

ACKNOWLEDGMENTS

I would like to thank George Wiggins for assisting with my programming skills and answering my endless programming questions as well as Daniel Null, Melvin Tooker, and Suzanne Hubbard for technical editing of this paper.

RECOMMENDED READING

SAS-L archives are an excellent source of information about SAS programming and tips and tricks. An additional resource is the SAS online documentation about ODS, Proc Print, and Proc Report. The Little SAS Book : A Primer, Second Edition by Lora D. Delwiche and Susan J. Slaughter is also a terrific reference for those just learning SAS.

CONTACT INFORMATION

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

Laura L. M. Thornton
Animal Scientist
10300 Baltimore Ave
BARC-WEST, Bldg 005, Rm. 306
Beltsville, MD, 20705-2350
Phone: 301 504 8334
Fax: 301 504 8092
E-mail: Laura.Thornton@ars.usda.gov
Web: <http://aipl.arsusda.gov/>

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.