Data _Null_ Reporting to PDF with ODS

Ajit Gemunu de Silva, Kaiser Permanente Division of Research, Oakland CA

Abstract

SAS programmers have many methods available to them for report writing including the Print and Report Procedures as well as Data _Null_ with the PUT statement. If programming inside the Windows environment, printing output from these procedures is usually seamless and practical. If one changes hardware or platform, however, printing the same report may not be so easy. Page breaks, fonts, as well as line and page settings may need to be changed. Repagination of an extant report is impossible without another utility program. Simple text reports become problematic. Printing to a PDF via ODS creates robust platform independent reports. Using Data _Null_, the PUT statement with the ODS option, and the less cited *Table* Definition from the Template Procedure (in contrast to more often discussed *Style* Definition from the Template Procedure) can make simple production reports that can be printed anywhere Adobe Acrobat or Ghostview is available.

Introduction

Printing in SAS has never been easier; that is if you restrict yourself to the Windows platform. If, however, you plan to use UNIX or virtually any other platform (save Mac OS), you may run into printing problems. Specifically, page breaks you see in the Display Manager will not necessarily be there when you go to print the file later via other software. Printing can be problematic.

Traditional Printing Settings

SAS reports, especially Data _Null_ reporting, use specific settings that must agree with the font and paper size. To get page breaks and wrapping correct, SAS system options may be set with the OPTIONS statement and these include line size (LS=) and page size (PS=). These may be revisited via additional line size and page size settings on the FILE statement. These are not redundant but rather may be used to print to less than the whole page as defined by the system options. The line size and page size settings default to the system settings if omitted from the FILE statement. If these settings are correctly applied, titles and footnotes appear in the right places indicating appropriately set page breaks. If you take this file, however, and drag it into a word processor, a text editor, or a new printing utility, then Caveat Emptor, page breaks and fonts may have been changed by the new software.

The Template Procedure

If you read ahead (or just the paper title) it becomes obvious that ODS is used to print to a PDF, however, one must digress briefly on the Template Procedure. The Template Procedure works with ODS to control output. Specifically the Template Procedure is used to modify or create Style and Table Definitions that ODS uses to render the final output. Most discussion heretofore has been about how nicely SAS can manipulate colors and fonts in output. Highlighting significant results on an HTML page is indeed quite impressive. These are the purview of the Style Definition and the Template Procedure can be used to make these changes and enhancement in ODS output. Column order, headers, and formats are the purview of the Table Definition and the Template Procedure "binds" this to the data component (from laborious DATA steps no doubt) and produces ODS output at the ODS destination. In this case the ODS destination will be a PDF that is easily opened up on most any platform that has the proper viewer. While the Template Procedure introduces a new lexicon associated with object oriented programming, the Table Definitions created can be quite simple and begin to look something like the syntax from the Report Procedure.

Creating a Table Definition

Here is a sample Table Definition:

```
proc template ;
  define table my_defn ;
    double_space = on ;
    classlevels  = on ;
    underline    = off ;
    overline     = off ;
    mvar sysdate ;

    header header1 ;
    footer footer1 ;

    define header1 ;
      text "My Report" ;
    end ;

    define footer1 ;
      text sysdate ;
    end ;

    column patient_id test date results ;

    define patient_id ;
      blank_dups = on ;
      width      = 12 ;
      format     = z8. ;
      header     = "Subject" ;
    end ;
```

```
    define test ;
      label      = "Test" ;
      format     =  test. ;
    end ;

    define date ;
      blank_dups = on ;
      width      = 10 ;
      format     = mmddyy10. ;
      label      = "Date" ;
    end ;

    define results ;
      label      = "Results" ;
      format     =  results. ;
    end ;
  end ;
run ;
```

This Table Definition double-spaces, controls underling and overlining, specifies a header and footer, but also uses CLASSLEVELS and MVAR. CLASSLEVELS is similar to the ORDER option on the DEFINE statement in the Report Procedure. It blanks out repeat values in a sorted column making the report more readable. The COLUMN statement looks right out of the Report Procedure and behaves similarly here. The MVAR statement controls macro variables. Without completely digressing into the Macro Facility, Table Definitions are produced (compiled) before data is available yet must accept macro variables when used (executed) by the following steps. MVAR tells SAS a macro variable is expected later. This prevents SAS from prematurely resolving the macro variable while creating the Table Definition. The footer from this Table Definition uses &SYSDATE. The Column Definitions may look more like familiar SAS code and can be used to specify width, formats, and labels. The BLANK_DUPS option works with CLASSLEVELS to blank out repeat values. Also note that these Column Definitions are named after the variables that will use them. This means nothing further than the name of the Table Definition need be specified in the Data _Null_.

Using a Table Definition

Here is the Data _Null_ step that uses this Table Definition:

```
data _null_ ;
 set final_data ;
 file print ods = ( template = "my_defn" ) ;
 put _ods_ ;
run ;
```

The Table Definition is called "my_defn" and quotes are required.

A Few Extras

```
options nodate
  orientation=landscape
  leftmargin=0 rightmargin=0
  topmargin=0 bottommargin=0 ;

ods pdf file="my_report.pdf" ;
```

Orientation and margins are controlled by the OPTIONS statement. ODS is used to open and name the PDF output file. These statements should precede the Data _Null_. Be sure to close the ODS PDF destination after the Data _Null_.

Conclusion

PDF is a rugged medium that frees the SAS programmer from platform related difficulties. Writing to a PDF with ODS is fairly simple using the Template Procedure. The Table Definition of the Template Procedure is easy to code and is reminiscent of the Report Procedure. In addition, SAS allows inheritance in the Template Procedure (see examples of this in the SAS documentation for the Style Definition of the Template Procedure) suggesting compact Table Definitions are possible for the advanced programmer.

Acknowledgements

Thank you Steve Wilson, Roderick Pedersen, Diane Carpenter, and Celeste Monnette. Special thanks to Ned Lewis, Paula Ray, and the Vaccine Studies Center for their support.

References

Heffner, William F., *ODS: The Data Step Knows*, SAS Institute Inc.

O'Conner, Daniel, *Next Generation Data _NULL_ Report Writing Using ODS OO Features*, SAS Institute Inc.

The Complete Guide to the SAS Output Delivery System, Version 8, Cary, NC, USA. SAS Institute Inc.

New Features in Version 8 of the SAS System, Course Notes, Cary, NC, USA. SAS Institute Inc.

Contact Information

Ajit Gemunu de Silva
Kaiser Permanente Division of Research
2000 Broadway
Oakland, CA 94612
Tel: 510.891.3440
Fax: 510.891.3508
ajit.g.desilva@kp.org