

PROC PRINT - the Granddaddy of all Procedures, Enhanced and Still Going Strong!

David Franklin, TheProgrammersCabin.com, Litchfield, NH

ABSTRACT

The PRINT procedure, or PROC PRINT, has been around since SAS first began and is considered one of the granddaddy procedures. Although this procedure has been replaced in part by the REPORT procedure, there is still a lot you can do with it.

This paper looks at first a simple dump of data, then dresses it up with the use of statements like the BY and ID statements to publication ready output. Next, output is cranked up a notch to demonstrate how PROC PRINT enhancements can be used to produce HTML (with graphics and links), RTF and PDF (with bookmarks). Along the way the paper will also touch on techniques for post processing to make your output more alive.

INTRODUCTION

PROC PRINT can trace its lineage back to the first version of SAS to be commercially released. In recent years, with the advent of PROC REPORT and the ODS system, the PRINT procedure has been pushed to the back of the reporting procedures to where it is considered as no more than being able to do a basic data dump. However, PROC PRINT has been enhanced and refined to where it is still a procedure worth knowing and having an understanding of.

FIRST, SOME DATA

For the purposes of this paper, the following Sheep Numbers data is used to produce the output:

```
data SheepNumbers;
  length Country $25 Year Number 8;
  infile cards;
  input Country $ & Year Number;
cards;
Australia 2007 85711187
Australia 2006 91028408
Australia 2005 101124891
Canada 2007 879100
Canada 2006 893800
Canada 2005 977600
China 2007 146018203
China 2006 151337202
China 2005 152035223
India 2007 64269000
India 2006 63558000
India 2005 62854000
New Zealand 2007 38460477
New Zealand 2006 40098191
New Zealand 2005 39879660
South Africa 2007 25082100
South Africa 2006 24982996
South Africa 2005 25334000
United Kingdom 2007 33946000
United Kingdom 2006 34722000
United Kingdom 2005 35253048
United States of America 2007 6165000
United States of America 2006 6230000
United States of America 2005 6135000
;
```

```
run;
```

For all the programs, the following code is used:

```
** Sort our data for the PRINT procedure. Note use of DESCENDING option
to sort in decreasing year value.;
proc sort data=SheepNumbers;
  by Country DESCENDING Year Number;
run;

** Set up standard titles and footnotes;
title1 "Sheep Numbers in Select Countries, By Year";
footnote1 "Note: Numbers for India are FAO Estimate";
footnote2 "Source: UNData, 03Nov2010";
```

Source of the data is UNData extracted November 3, 2010, which can be found at <http://data.un.org/>.

A SIMPLE DATA DUMP

First, a simple basic PRINT procedure call that “dumps” the data:

```
proc print data=SheepNumbers;
  var Country Year Number;
  title2 "(Basic Output)";
run;
```

The output is given below:

```
      Sheep Numbers in Select Countries, By Year
      (Basic Output)

Obs    Country                Year    Number
-----
  1    Australia                2007    85711187
  2    Australia                2006    91028408
  3    Australia                2005   101124891
  4    Canada                   2007    879100
  5    Canada                   2006    893800
  6    Canada                   2005    977600
  7    China                    2007   146018203
  8    China                    2006   151337202
  9    China                    2005   152035223
 10    India                    2007   64269000
 11    India                    2006   63558000
 12    India                    2005   62854000
 13    New Zealand              2007   38460477
 14    New Zealand              2006   40098191
 15    New Zealand              2005   39879660
 16    South Africa             2007   25082100
 17    South Africa             2006   24982996
 18    South Africa             2005   25334000
 19    United Kingdom           2007   33946000
 20    United Kingdom           2006   34722000
 21    United Kingdom           2005   35253048
 22    United States of America  2007    6165000
 23    United States of America  2006    6230000
 24    United States of America  2005    6135000
```

Note: Numbers for India are FAO Estimate
Source: UNData, 03Nov2010

There are no formats, except what is already defined in the dataset and no labels for the variables – this is a basic as it gets.

NOOBS AND LABEL OPTIONS

The NOOBS and LABEL options make the output a little more presentable by removing the OBS variable and allowing for column labels that are not just variable names.

```
proc print data=SheepNumbers noobs label;
  var Country Year Number;
  label Country='Country'
        Year='Year'
        Number='Reported Number';
  title2 "(Something a Little Better)";
run;
```

The output is given below:

Sheep Numbers in Select Countries, By Year
(Something a Little Better)

Country	Year	Reported Number
Australia	2007	85711187
Australia	2006	91028408
Australia	2005	101124891
Canada	2007	879100
Canada	2006	893800
Canada	2005	977600
China	2007	146018203
China	2006	151337202
China	2005	152035223
India	2007	64269000
India	2006	63558000
India	2005	62854000
New Zealand	2007	38460477
New Zealand	2006	40098191
New Zealand	2005	39879660
South Africa	2007	25082100
South Africa	2006	24982996
South Africa	2005	25334000
United Kingdom	2007	33946000
United Kingdom	2006	34722000
United Kingdom	2005	35253048
United States of America	2007	6165000
United States of America	2006	6230000
United States of America	2005	6135000

Note: Numbers for India are FAO Estimate
Source: UNData, 03Nov2010

Just by adding these two simple options the output is a little better, but we can still do more.

BY AND ID STATEMENTS

The BY and ID statements allow for a “grouping” of the output. The BY statement itself produces a separate section of the report for each BY group, while the ID statement identifies observations by using the formatted values of the variables that you list instead of by using observation numbers.

```
proc print data=SheepNumbers label;  
  by Country;  
  id Country;  
  var Year Number;  
  format Number comma12.;  
  label Country='Country'  
         Year='Year'  
         Number='Reported Number';  
  title2 "(Something even better, with BY and ID)";  
run;
```

The output from this appears below:

Sheep Numbers in Select Countries, By Year
(Something even better, with BY and ID)

Country	Year	Reported Number
Australia	2007	85,711,187
	2006	91,028,408
	2005	101,124,891
Canada	2007	879,100
	2006	893,800
	2005	977,600
China	2007	146,018,203
	2006	151,337,202
	2005	152,035,223
India	2007	64,269,000
	2006	63,558,000
	2005	62,854,000
New Zealand	2007	38,460,477
	2006	40,098,191
	2005	39,879,660
South Africa	2007	25,082,100
	2006	24,982,996
	2005	25,334,000
United Kingdom	2007	33,946,000
	2006	34,722,000
	2005	35,253,048
United States of America	2007	6,165,000
	2006	6,230,000
	2005	6,135,000

Note: Numbers for India are FAO Estimate
Source: UNData, 03Nov2010

In the PROC PRINT call, a NOOBS is implied so that option is not required. The variable in the ID statement is used as a "grouping" variable and must also be included in the BY statement -- this is what makes the COUNTRY variable appear as a grouping variable and have a spacing (blank line) between each group. The format COMMA12. is used to make the number appear in comma format so that it is easier to read. Like before, the LABEL statement and option is used to make user friendly labels appear rather than the variable names. It is important to note that the BYLINE option must be on for this to work although the usual BY line processing does not appear in the listing the usual way.

Unlike the REPORT procedure, a line underneath the column headings does not appear in this mode, but does appear when using ODS destinations as we shall see later. Given all the advancements with PROC PRINT it would be nice to have that option outside ODS.

In order for a line to appear underneath the column headings, some post-processing needs to be done in the following order:

- Make a line underneath the column headings – does not need to be all the way across except for the first and last variables (see below of an example). I find it useful to use a character not used elsewhere to set the underline character at this stage.
- When the file is read in after the PROC PRINT call, if the line carries the characters for used for the line, find the first and last positions.
- Fill the text between the first and last characters of the line with the character that is going to be the line character.

Using the previous example the use of this method will be shown. It must be noted that the output must be saved in a file, usually using the PRINTTO procedure.

```
proc printto print='E:\NESUG\2011\ODS\POSTPRC1.txt' new;
run;
proc print data=SheepNumbers label split='!';
  by Country;
  id Country;
  var Year Number;
  format Number comma12.;
  label Country='Country!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!'
```

```

Year='Year!£'
Number='Reported!Number!££££££££££££';
title2 "(Something even better, with what looks like the HEADLINE option in PROC REPORT)";
run;
proc printto;
run;

```

Note that the use of the SPLIT option that will force a line break in the column header. I have also used the '£' character for the underline – you will see shortly why.

Next is the post-processing step.

```

data _null_;
length _txt $255;
infile 'E:\NESUG\2011\ODS\POSTPRC1.txt' sharebuffers length=len;
file 'E:\NESUG\2011\ODS\POSTPRC1.txt';
input _txt $varying255. len;
if index(_txt,'£') then do;
_startpos= index(_txt,'£');
do _k=_startpos to len;
if substr(_txt,_k)='£' then _endpos=_k;
end;
substr(_txt,_startpos,_endpos)=repeat('-',_endpos-_startpos);
end;
put _txt $varying255. len;
run;

```

Because the INFILE statement uses the SHAREBUFFERS option, the INFILE and FILE statements must use the same file name. Also used is the LEN option in the INFILE statement to tell how long the incoming sting is and use this in the INPUT statement.

Only if a '£' is found in the incoming line, will the next step proceed, where we will first find the first and last position of the '£' characters in the string, and then populate that string filling in the gaps with the replacement character, which in this case is the '-' character.

At the end of the datastep, all the lines, whether replaced or not with new text, is output to the file. Below is the new output file (just the titles, column header and a couple of rows of data, as previously):

Sheep Numbers in Select Countries, By Year
(Something even better, with what looks like the HEADLINE option in PROC REPORT)

Country	Year	Reported Number
Australia	2007	85,711,187
	2006	91,028,408
	2005	101,124,891
Canada	2007	879,100
	2006	893,800
	2005	977,600
China	2007	146,018,203
	2006	151,337,202
	2005	152,035,223
India	2007	64,269,000
	2006	63,558,000
	2005	62,854,000
United Kingdom	2007	33,946,000
	2006	34,722,000
	2005	35,253,048
United States of America	2007	6,165,000
	2006	6,230,000
	2005	6,135,000

Note: Numbers for India are FAO Estimate
Source: UNData, 03Nov2010

This is actually a very tricky post-processing task to do so care is advised.

SUM STATEMENT AND N OPTION

The SUM statement allows for a summary of numbers to be produced in the listing and is useful when subgroup and total summaries are requested, while the N option prints the number of observations.

```
proc print data=SheepNumbers label n='Country Count = ' 'Total Count = ';
  title2 "(Output using the SUM Statement)";
  by Country;
  id Country;
  var Year Number;
  format Number comma18.;
  label Country='Country'
        Year='Year'
        Number='Reported Number';
  sum number;
run;
```

The SUM statement tells the procedure what to use to sum the variables while the N statement, as used here, will put out the number of observations found in each BY variable and give the label 'Country Count' and 'Total Count' for the total number of observations. Below is the new output:

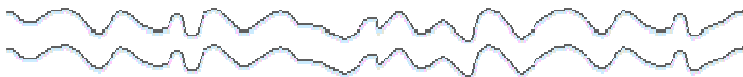
Sheep Numbers in Select Countries, By Year
(Output using the SUM Statement)

Country	Year	Reported Number
Australia	2007	85,711,187
	2006	91,028,408
	2005	101,124,891
-----		-----
Australia		277,864,486

Country Count = 3

Canada	2007	879,100
	2006	893,800
	2005	977,600
-----		-----
Canada		2,750,500

Country Count = 3



United Kingdom	2007	33,946,000
	2006	34,722,000
	2005	35,253,048
-----		-----
United Kingdom		103,921,048

Country Count = 3

United States of America	2007	6,165,000
	2006	6,230,000
	2005	6,135,000
-----		-----
United States of America		18,530,000
		=====
		1,236,975,086

Country Count = 3
Total Count = 24

Note: Numbers for India are FAO Estimate
Source: UNData, 03Nov2010

The Country Count and Total Count are not useful on the output but is presented for illustrative purposes.

Other options are available in the PROC PRINT procedure that I have rarely seen used, including the UNIFORM, DOUBLE and ROUND options, and I encourage you to read these up in the documentation and try them.

HTML, RTF AND PDF OUTPUT USING ODS

ODS has allowed the output from many procedures to be collected and reported in formats that go well beyond the text format that SAS originally had. In this paper I will just have a quick look at the three output formats that SAS provides, e.g. HTML, RTF and PDF. In each case a custom style called CUSTOM defined using a PROC TEMPLATE call is used (the use and description of PROC TEMPLATE is beyond the scope of this paper, but there are a large number of references that can be accessed that discuss the PROC TEMPLATE procedure).

First, lets look at HTML.

```
ods listing close;
ods html file='E:\NESUG\2011\ods\htmlexample.html' style=custom;
proc print data=SheepNumbers label split='!';
  by Country;
  id Country;
  var Year /style=[cellwidth=0.5in just=c];
  var Number /style=[cellwidth=1.0in];;
  format Number comma12.;
  label Country='Country'
         Year='Year'
         Number='Reported!Number';
  title2 "(HTML Output, Using Custom Template)";
run;
ods html close;
ods listing;
run
```

The first point of note is the ODS HTML statement – here the output file destination and style reference is made. The second major difference is the use of the two VAR statements. Inside ODS using the PRINT procedure, the relevant STYLE= options must follow the variables they apply to, however the next variable or set of variable must follow in later VAR statement calls. In the example above, the YEAR variable has a cell width of 0.5 inches and the column centered, and the YEAR variable has the column width set to 1 inch width. The rest of the code is as seen before except the output (see below) is in Times Roman format as defined in the TEMPLATE procedure definition that defined the style.

Below is the output produced using the ODS HTML output destination:

Sheep Numbers in Select Countries, By Year
(HTML Output, Using Custom Template)

Country	Year	Reported Number
Australia	2005	101,124,891
	2006	91,028,408
	2007	85,711,187
Canada	2005	977,600
	2006	893,800
	2007	879,100
China	2005	152,035,223
	2006	151,337,202
	2007	146,018,203
India	2005	62,854,000
	2006	63,558,000
	2007	64,269,000
New Zealand	2005	39,879,660
	2006	40,098,191
	2007	38,460,477
South Africa	2005	25,334,000
	2006	24,982,996
	2007	25,082,100
United Kingdom	2005	35,253,048
	2006	34,722,000
	2007	33,946,000
United States of America	2005	6,135,000
	2006	6,230,000
	2007	6,165,000

Note: Numbers for India are FAO Estimate
Source: UNData, 03Nov2010

It must be noted here that the use of the STYLE= options have been around since ODS started. The STYLE locations for the PRINT procedure are:

- BYLABEL -- the label for the BY variable on the line containing the SUM totals
- DATA -- the cells of all columns
- GRANDTOTAL -- the SUM line containing the grand totals for the whole report
- HEADER -- all column headings
- N -- N= table and contents
- OBS -- the data in the OBS column
- OBSHEADER -- the header of the OBS column
- TABLE -- the structural part of the report - that is, the underlying table used to set things like the width of the border and the space between cells
- TOTAL -- the SUM line containing totals for each BY group

Style attributes that are available include:

- BORDERWIDTH=
- CELLHEIGHT=
- CELLPADDING=
- CELLSPACING=
- CELLWIDTH=
- FONT=
- FONT_FACE=
- FONT_SIZE=
- FONT_STYLE=
- FONT_WEIGHT=
- JUST=
- POSTIMAGE=
- PRETEXT=
- RULES=
- VJUST=

There are others – refer to the PROC PRINT documentation.

The RTF code is the same except for the ODS RTF statement as given below with their relevant output:

```
ods rtf file='E:\NESUG\2011\ods\rtfexample.rtf' style=custom;
```

Sheep Numbers in Select Countries, By Year
(RTF Output, Using Custom Template)

Country	Year	Reported Number
Australia	2005	101,124,891
	2006	91,028,408
	2007	85,711,187
Canada	2005	977,600
	2006	893,800
	2007	879,100
China	2005	152,035,223
	2006	151,337,202
	2007	146,018,203
India	2005	62,854,000
	2006	63,558,000
	2007	64,269,000
New Zealand	2005	39,879,660
	2006	40,098,191
	2007	38,460,477
South Africa	2005	25,334,000
	2006	24,982,996
	2007	25,082,100
United Kingdom	2005	35,253,048
	2006	34,722,000
	2007	33,946,000
United States of America	2005	6,135,000
	2006	6,230,000
	2007	6,165,000

Note: Numbers for India are FAO Estimate
Source: UNData, 03Nov2010

The PDF code is also the same except for the ODS PDF statement as given below with their relevant output:

```
ods pdf file='E:\NESUG\2011\ods\pdfexample.pdf' style=custom;
```

Sheep Numbers in Select Countries, By Year
(PDF Output, Using Custom Template)

Country	Year	Reported Number
Australia	2005	101,124,891
	2006	91,028,408
	2007	85,711,187
Canada	2005	977,600
	2006	893,800
	2007	879,100
China	2005	152,035,223
	2006	151,337,202
	2007	146,018,203
India	2005	62,854,000
	2006	63,558,000
	2007	64,269,000
New Zealand	2005	39,879,660
	2006	40,098,191
	2007	38,460,477
South Africa	2005	25,334,000
	2006	24,982,996
	2007	25,082,100
United Kingdom	2005	35,253,048
	2006	34,722,000
	2007	33,946,000
United States of America	2005	6,135,000
	2006	6,230,000
	2007	6,165,000

In the ODS examples there is no blank line between the countries. In SAS version 9.2 and beyond, there is an option called BLANKLINE that specifies the number of observations to be read until a blank line appears.

It is very interesting to note that a line underneath the column headers appears in the ODS output destinations, something that had to be done via post processing when it is put to the output file without ODS destinations.

ADDING LINKS AND GRAPHICS IN HTML

Adding a link and a graphic to an HTML file is done using the PREIMAGE and PREHTML STYLE options in the PROC PRINT statement, as shown below:

```
ods listing close;
ods html file='E:\NESUG\2011\ods\htmlexample.jpg.html' style=custom;
proc print data=SheepNumbers label split='!'
      STYLE=[PREIMAGE='E:\NESUG\2011\ods\border.jpg'
      PREHTML='<a href="http://en.wikipedia.org/wiki/Sheep">Link to more about Sheep</a>'];
  by Country;
  id Country; ** Acts like a "GROUPING" variable;
  var Year /style=[cellwidth=0.5in just=c];
  var Number /style=[cellwidth=1.0in];;
  format Number comma12.;
  label Country='Country'
        Year='Year'
        Number='Reported!Number';
  title2 "(HTML Output with Graphic and Link, Using Custom Template)";
run;
ods html close;
ods listing;
run;
```

In the example the link was made to an external URL but the link could quite easily be made to other file. Refer to the output below for an example of the new output:

Sheep Numbers in Select Countries, By Year
(HTML Output with Graphic and Link, Using Custom Template)

[Link to more about Sheep](#)



Country	Year	Reported Number
Australia	2005	101,124,891
	2006	91,028,408
	2007	85,711,187
Canada	2005	977,600
	2006	893,800
	2007	879,100
China	2005	152,035,223
	2006	151,337,202
	2007	146,018,203
India	2005	62,854,000
	2006	63,558,000
	2007	64,269,000
New Zealand	2005	39,879,660
	2006	40,098,191
	2007	38,460,477
South Africa	2005	25,334,000
	2006	24,982,996
	2007	25,082,100
United Kingdom	2005	35,253,048
	2006	34,722,000
	2007	33,946,000
United States of America	2005	6,135,000
	2006	6,230,000
	2007	6,165,000

Note: Numbers for India are FAO Estimate
Source: UNData, 03Nov2010

ADDING A GRAPHIC IN RTF AND PDF

Importing a graphic into RTF uses the same style PREIMAGE option in the PROC PRINT statement, as shown below:

```
ods listing close;
ods rtf file='E:\NESUG\2011\ods\rtfexamplejpg.rtf' style=custom;
proc print data=SheepNumbers label split='!'
    STYLE=[PREIMAGE='E:\NESUG\2011\ods\border.jpg'];
    by Country;
    id Country; ** Acts like a "GROUPING" variable;
    var Year /style=[cellwidth=0.5in just=c];
    var Number /style=[cellwidth=1.0in];;
    format Number comma12.;
    label Country='Country'
           Year='Year'
           Number='Reported!Number';
    title2 "(RTF Output with Graphic, Using Custom Template)";
run;
ods rtf close;
ods listing;
run;
```

The output is given below:

Sheep Numbers in Select Countries, By Year
(RTF Output with Graphic, Using Custom Template)



Country	Year	Reported Number
Australia	2005	101,124,891
	2006	91,028,408
	2007	85,711,187
Canada	2005	977,600
	2006	893,800
	2007	879,100
China	2005	152,035,223
	2006	151,337,202
	2007	146,018,203
India	2005	62,854,000
	2006	63,558,000
	2007	64,269,000
New Zealand	2005	39,879,660
	2006	40,098,191
	2007	38,460,477
South Africa	2005	25,334,000
	2006	24,982,996
	2007	25,082,100
United Kingdom	2005	35,253,048
	2006	34,722,000
	2007	33,946,000
United States of America	2005	6,135,000
	2006	6,230,000
	2007	6,165,000

Just by changing the RTF destination to a PDF destination will also bring in the same graphic into a PDF created output file.

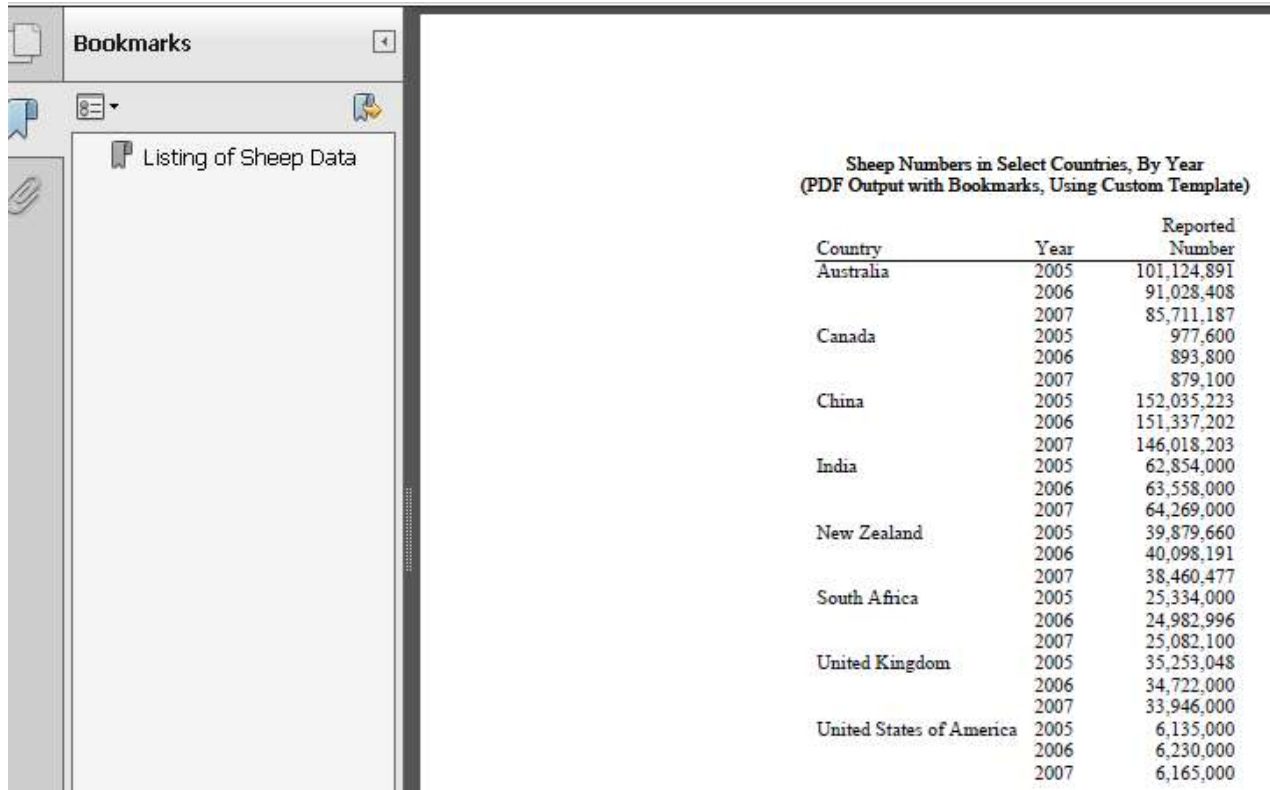
BOOKMARKS AND PDF

Bookmarks are created automatically when creating a PDF file using ODS PDF, however the bookmarks use default value which make the text look awful. There is a trick to modifying the bookmarks as the following example shows:

```
ods listing close;
ods pdf file='E:\NESUG\2011\ods\pdfexamplelink.pdf' style=custom;
ods escapechar="!";
options byline;
proc sort data=SheepNumbers;
  by Country Year;
run;
ods proclabel="Listing of Sheep Data";
proc print data=SheepNumbers label split='!' contents='';
  by Country;
  id Country;
  var Year /style=[cellwidth=0.5in just=c];
  var Number /style=[cellwidth=1.0in];;
  format Number comma12.;
  label Country='Country'
        Year='Year'
        Number='Reported!Number';
  title2 "(PDF Output with Bookmarks, Using Custom Template)";
run;
ods pdf close;
ods listing;
run;
```

The ODS PROCLABEL statement replaces the default text associated with the bookmark at the first level – this should be done before the PRINT procedure is run.

The second item is the CONTENTS option in the PROC PRINT statement – in this case the text is set to missing so that the label for the second level is missing. See below for the output:



The screenshot shows a PDF viewer interface with a 'Bookmarks' sidebar on the left. The active bookmark is 'Listing of Sheep Data'. The main content area displays a table titled 'Sheep Numbers in Select Countries, By Year (PDF Output with Bookmarks, Using Custom Template)'. The table has three columns: Country, Year, and Reported Number. The data is grouped by country, with years listed for each country.

Country	Year	Reported Number
Australia	2005	101,124,891
	2006	91,028,408
	2007	85,711,187
Canada	2005	977,600
	2006	893,800
	2007	879,100
China	2005	152,035,223
	2006	151,337,202
	2007	146,018,203
India	2005	62,854,000
	2006	63,558,000
	2007	64,269,000
New Zealand	2005	39,879,660
	2006	40,098,191
	2007	38,460,477
South Africa	2005	25,334,000
	2006	24,982,996
	2007	25,082,100
United Kingdom	2005	35,253,048
	2006	34,722,000
	2007	33,946,000
United States of America	2005	6,135,000
	2006	6,230,000
	2007	6,165,000

When creating separate output for a single PDF file using ODS PDF it is useful to run the ODS PROCLABEL statement before each report is produced so that the bookmarks are clearly set up at the ODS PDF CLOSE statement, i.e. the PDF file is generated.

CONCLUSION

The PRINT procedure is not just something used to “dump” data from a dataset to a text listing file. During the tour we have seen how simple things like NOOBS and LABEL options, and BY and ID statements make the output more presentable. An example of how we can post process a file was also shown. Next on our journey was a brief introduction into how ODS HTML/RTF/PDF has enhanced the PRINT procedure. The PRINT procedure may be one of the oldest procedures in the SAS stable but it still has life today as a serious option when considering what procedure to use to produce your output.

REFERENCES

UNData. extracted from <http://data.un.org/> on November 3, 2010

SAS Institute Inc. 2006. Base SAS® 9.1.3 Procedures Guide, Second Edition, Volumes 1, 2, 3, and 4. Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

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 registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION

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

Author Name: David Franklin
Company: TheProgrammersCabin.com
Address: 16 Roberts Rd
City, State ZIP: Litchfield, NH 03052
Work Phone: 603-275-6809
Email: dfranklin@TheProgrammersCabin.com
Web: TheProgrammersCabin.com