

Linking SAS® Reports to Spreadsheet Data

Ben Cochran, The Bedford Group, Raleigh, NC

Abstract

This paper takes a step by step approach to building a summary report with the REPORT procedure, and then building 'hot links' to the detail data that is found in Excel spreadsheets. This paper starts with the data, either a SAS data set or spreadsheet data, shows the audience how to manipulate the data to get it ready for the report, it then shows how to use PROC REPORT to get the desired result, and then it covers the code needed to link to specific spreadsheets.

Introduction

This paper covers several examples of building reports with PROC REPORT and then linking these reports to data found in Excel spreadsheets. The PROC REPORT code creates summary reports with 'hot links' in the SAS output and when they are selected, detail data is displayed in spreadsheets.

Example 1:

The first example uses the SASHELP.CLASS dataset to build a summary report where there is a row per Age_Group, and a column per Gender. The Age_Groups (Young, Middle, and Mature) are the 'hot links'. See Report 1.

	Gender					
	F		M			
age_group	Height	%	Height	%	Total Height	%
1. Young	167	31%	239	37%	406	34%
2. Middle	249	46%	195	31%	443.9	37%
3. Mature	129	24%	206	32%	334.5	28%
Total	545	100%	639	100%	1184.4	100%

Report 1.

The first illustration shows the results of clicking on the 'Young' row.

	A	B	C	D	E	F	G
1	Name	Sex	Age	Height	Weight	age_group	
2	Jane	F	12	59.8	84.5	1. Young	
3	Joyce	F	11	51.3	50.5	1. Young	
4	Louise	F	12	56.3	77	1. Young	
5	James	M	12	57.3	83	1. Young	
6	John	M	12	59	99.5	1. Young	
7	Robert	M	12	64.8	128	1. Young	
8	Thomas	M	11	57.5	85	1. Young	

Spreadsheet 1.

The first thing to do is to create a separate spreadsheet for each age group. The program below does this.

```

data class;
  set sashelp.class;
  if age lt 13 then age_group='1. Young';
  else if age lt 15 then age_group = '2. Middle';
  else age_group = '3. Mature';
run;

PROC EXPORT DATA= class(where=(age < 13))
  OUTFILE= "C:\ben\young.xls"
  DBMS=EXCEL REPLACE;
  SHEET="young";
RUN;

PROC EXPORT DATA= class(where=(age between 13 and 14))
  OUTFILE= "C:\ben\mid.xls"
  DBMS=EXCEL REPLACE;
  SHEET="middle";
RUN;

PROC EXPORT DATA= class(where=(age ge 15))
  OUTFILE= "C:\ben\old.xls"
  DBMS=EXCEL REPLACE;
  SHEET="old";
RUN;

```

Program 1.

The DATA step creates the variable AGE_Group. The next three PROC EXPORT steps create three spreadsheets named 'young.xls', 'mid.xls', and 'old.xls'. The WHERE = options control which observations go to which spreadsheet.

Now that the spreadsheets have been created with the appropriate data, its time to write the program that creates the report that will be used to link to the spreadsheets. The PROC REPORT program below does accomplishes this task. Pay close attention to the compute block. It contains the specific code that creates the 'hot links' to the spreadsheets.

```

compute age_group ;
  if _break_ eq ' ' then do;
    if age_group=: '1.' then urlstring='c:\ben\young.xls' ;
    else if age_group=: '2.' then urlstring='c:\ben\mid.xls';
    else if age_group=: '3.' then urlstring='c:\ben\old.xls';
    call define(_col_, 'URL', urlstring);
  end;
  if age_group= ' ' then age_group='Total' ;
endcompute;

```

Program Segment 1.

The URLSTRING identifies the spreadsheet that is the link target.. The CALL DEFINE statement is necessary to make the linking work.

The comparison operator '=' is a 'starts with' operator. Simply put, it means if the variable AGE_GROUP starts with a '1.' then point to the young spreadsheet. If the variable AGE_GROUP starts with a '2.' Then point to the mid spreadsheet.

The entire program is shown below:

```

proc report data=class nowd style(summary)={font_size=13pt font=('Arial') foreground=blue};
  columns age_group sex, (height height=ht_pc) height=ht_tot height=ht_totPctsum ;
  define sex / across 'Gender';
  define age_group / group ;
  define height / analysis sum format=comma12. 'Height' ;
  define ht_pc / analysis pctsum format=percent6. '%';
  define ht_totPctSum / analysis pctsum format=percent6. '%';
  define ht_tot / sum 'Total Height';
  compute age_group ;
    if _break_ eq ' ' then do;
      if age_group=: '1.' then urlstring='c:\ben\young.xls' ;
      else if age_group=: '2.' then urlstring='c:\ben\mid.xls';
      else if age_group=: '3.' then urlstring='c:\ben\old.xls';
      call define(_col_, 'URL', urlstring);
    end;
    if age_group= ' ' then age_group='Total' ;
  endcompute;
  rbreak after / summarize;
  compute after;
  sex='Total';
  endcompute;
run;

```

Program 2.

Program 2 creates Report 1.

The above example uses three different spreadsheets. The presentation will include many more examples of SAS output linking to excel spreadsheets.

Example 2:

This example also uses PROC REPORT but the focus is not so much on the report, but is on creating different tabs (or pages) within the same spreadsheet (or workbook). The EXCEL engine, introduced in SAS9, is used in this example.

The general syntax of the LIBNAME statement using the EXCEL engine is...

```
LIBNAME libref engine-name physical file name < libname-options > ;
```

The specific LIBNAME statement used in example 2 is:

```
libname ss excel 'c:\test_class1.xls';
```

Program Segment 2.

This LIBNAME statement creates and opens the **test_class** spreadsheet on the user's C drive. The number of sheets are limited only by available memory. Also, while this LIBNAME statement is in effect, and the library is active, it is under the control of the SAS session. In order to 'release' the spreadsheet so that it can be opened outside the SAS session, issue the following statement:

```
libname SS clear;
```

The above libname statement creates the SS library as shown below:

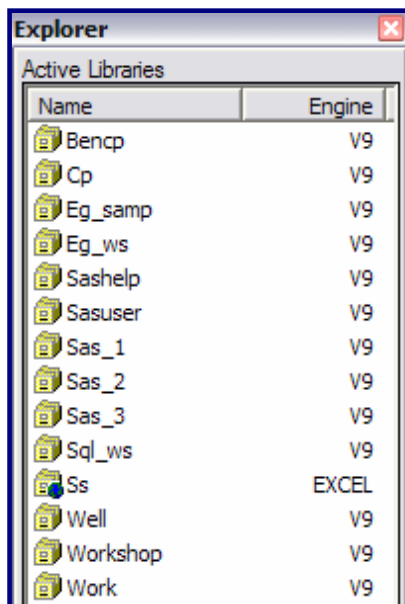


Figure 1: The Library Window.

Each 'page' in the test_class spreadsheet is treated as a separate SAS dataset in the library. And conversely, every dataset in the SS library is a physical page in the test_class spreadsheet. So, if a SAS program outputs different datasets to the SS library, it is actually writing data to different pages in the spreadsheet.

```
libname ss excel 'c:\test_class1.xls';
```

```
proc report nowd data=sashelp.class(where=(sex='F')) out=ss.females;  
  columns name sex age height weight ratio;  
  define height / analysis mean f=6.1;  
  define weight / analysis mean f=6.1;  
  define ratio / computed format = 6.2;  
  compute ratio;  
    ratio = height.mean / weight.mean;  
    if ratio > .67 then call define(_row_, 'style', 'style=[background=cyan]');  
  endcompute;
```

```
run;
```

```
proc report nowd data=sashelp.class(where=(sex='M')) out=ss.males;  
  columns name sex age height weight ratio;  
  define height / analysis mean f=6.1;  
  define weight / analysis mean f=6.1;  
  define ratio / computed format = 6.2;  
  compute ratio;  
    ratio = height.mean / weight.mean;  
    if ratio > .67 then call define(_row_, 'style', 'style=[background=cyan]');  
  endcompute;
```

```
run;
```

```
libname ss clear;
```

Notice the two LIBNAME statements. The first one creates the SS library, while the second one 'clears' the library. Also notice the OUT= options on the PROC REPORT statement and how the WHERE= option controls which observation is written to which 'page' in the spreadsheet. Notice the 'tabs' in the spreadsheet below:

Microsoft Excel - test_class1

File Edit View Insert Format Tools Data SAS Window Help Adobe PDF

Active Data: Active Selection Analyze Data Reports

H14 =

	A	B	C	D	E	F	G
1	Name	Sex	Age	Height	Weight	ratio	_BREAK_
2	Joyce	F	11	51.3	50.5	1.015842	
3	Jane	F	12	59.8	84.5	0.707692	
4	Louise	F	12	56.3	77	0.731169	
5	Alice	F	13	56.5	84	0.672619	
6	Barbara	F	13	65.3	98	0.666327	
7	Carol	F	14	62.8	102.5	0.612683	
8	Judy	F	14	64.3	90	0.714444	
9	Janet	F	15	62.5	112.5	0.555556	
10	Mary	F	15	66.5	112	0.59375	
11							
12							
13							
14							

Ready

females / males

Spreadsheet 1. Notice the 'tabs'.

Acknowledgments

I would like to acknowledge and greatly thank the Technical Support Department at SAS Institute for their helpful knowledge and expertise that they so freely gave.

Contact Information

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

Ben Cochran
 The Bedford Group
 Raleigh, NC 27607
 Phone: 919.741.0370
 Email: bedfordgroup@nc.rr.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.