With Version 6 SAS software available to all platforms, the REPORT procedure will be the first choice in report writing needs for all SAS programmers. This tutorial emphasizes the functionality of PROC REPORT, by comparing and contrasting its report writing capabilities with those available in the PRINT, QPRINT, and TABULATE procedures and the SAS DATA step.

**ABSTRACT**

Now, note that PROC REPORT can be used to create basically the same report with just as little effort.

```
libname in 'sasid.your.library';
proc report data=inv.boats;
run;
```

**INTRODUCTION**

PROC REPORT is a new procedure in Version 6 of the SAS System that does virtually everything PROC PRINT, PROC QPRINT, and the DATA step FILE and PUT statements have been used to do in the past. It also incorporates some of the nicer functionalities of PROC TABULATE. This paper demonstrates how easy report writing is with PROC REPORT by comparing some old jobs using other methods with new ones using PROC REPORT. First, it covers how PROC PRINT functionality has been incorporated into PROC REPORT. Second, it demonstrates how quickly and easily complicated reports, which used to take hours of DATA step programming, can be generated using PROC REPORT. It shows how simple tabular output can be generated with some features similar to those of PROC TABULATE. An important example showing how PROC REPORT will replace PROC QPRINT as the tool for creating quick and easily tailored reports is also given. Last, it shows how output generated by PROC REPORT can be saved into a SAS data library and easily recalled with only one step. All these things make PROC REPORT the report-writing tool of the future.

**SIMPLE PROC PRINT**

PROC PRINT has always been the best way to get a nice report with a minimum amount of time and effort. As the following example shows, it doesn't get much simpler than this.

```
libname in 'sasid.your.library';
proc print data=inv.boats;
run;
```

```
 Screen 1 Simple PROC PRINT Report
```

As seen above, PROC REPORT can create a basic report as easily as PROC PRINT. However, PROC REPORT is capable of creating much much more complicated reports with not much more effort than in the example above. It can also do things such as break your data into predefined groups and highlight breaks, clearly marking ID columns. It even provides the ability to sum specified variables, yielding subtotals and grand totals. In the past, these types of reports have been produced with PROC PRINT, but with the advent of a totally interactive PROC REPORT, ease of use will be redefined. This interactivity makes it possible to generate complicated reports without having to learn a new language.

What follows is a more complicated PROC PRINT step that produces a special report. In this report, BY groups are clearly set apart from each other; the ID columns are nicely differentiated from the detail columns; and the summary lines are clearly marked.

```
libname in 'sasid.your.library';
data one;
set in.boats;
proc sort data=one;
by item outlet customer invdate;```

```
 Screen 2 Simple PROC REPORT Report
```

```
proc report data=inv.boats;
libname in 'sasid.your.library';
```

```
 Screen 1 Simple PROC PRINT Report
```

```
 libname in 'sasid.your.library';
 proc report data=inv.boats;
 run;
```
format price dollar13.2;
END;

1. Invoke PROC REPORT with the FS option. Note that the variables in this initial screen are ordered the same as they were in the first report.

2. Move variables to the following order: ITEM, OUTLET, CUSTOMER, INVDATE, INVOICE, PRICE. (This is done by placing the cursor on the variable to be moved, hitting the MOVE key, placing the cursor on the variable around which the first variable is to be moved, and hitting either ADD_LEFT or ADD_RIGHT. The original variable selected is either added to the left or to the right of the second variable selected).

3. Individually, select OUTLET and ITEM and hit the DEFINE key. Once in the DEFINITION window, select a USAGE of ORDER, then select the OK button and back out. Define OUTLET and ITEM, respectively, to be ORDER variables.

4. In the same manner as just described, define CUSTOMER to have a WIDTH=17 and a FORMAT=$17. Looking ahead to step 5, PRICE has an insufficient WIDTH value and must be increased to 13. Therefore, for all variables to fit on the screen the WIDTH of CUSTOMER must be decreased to 17.

5. Define PRICE to have a USAGE VALUE of STATISTIC=SUM, WIDTH=13, and FORMAT=$DOLLAR13.2.

6. Select OUTLET and hit the BREAK key. In the LOCATION window, select AFTER DETAIL and the OK button. In the BREAK window, select OVERLINE, SKIP, and SUMMARIZE.

7. BREAK on ITEM, AFTER DETAIL, SKIP, SUMMARIZE.

8. Select EDIT from the action bar, RBREAK from the pull-down window, and DOUBLE OVERLINE and SUMMARIZE from the BREAK window.

As each step is executed, the screen changes to reflect the change made. When step 8 has been executed, the final report looks like this:

---

Screen 6 Complex PROC REPORT Report - First Page
---

This is a nice report generated with minimal effort. To create the same report using PROC REPORT requires just as little effort and no prior knowledge of a language. The following steps outline one of the ways to produce the same report with PROC REPORT:
The DATA step has always been the final word in report writing with
SAS software in that it has always been possible to precisely tailor
be better set apart from the detail data. One way to do this would
be to print out more descriptive text on the summary lines. In previ­
ous releases of the SAS System, creating the same report with only
this small change would require coding an extensive DATA step.

Here is where the power of PROC REPORT really begins to show
up. To add text to the summary lines requires slightly altering only
the last two steps in the previous example. Without PROC
REPORT, however, consider how long it might take to code and
debug the following DATA step to do the job:

data two;
  set one end=end;
  by item outlet;
  file print print header=header;
  if first.item then do;
    put a9 'The total price for all Items is ' a9 ' $ ' a9 ' is ' a9 ' ' a9 'grand sum';
    count=0;
    end;
  if last.outlet then do;
    count+1;
    put a9 'The total price for item ' a9 ' is ' a9 ' $ ' a9 ' grand sum';
    count=0;
    end;
  else do;
    put a9 'The total price for item ' a9 ' is ' a9 ' $ ' a9 ' grand sum';
    count+1;
    end;
  put a8 'INVOICE' a60 'PRICE' a35 '-------------' a63 price dollar13.2;
  if not end then put ' .';
  itemsum+price; itemsnm+price;
  outsum+price; outsnm+price;
  qrandsnm dollar13.2;
  if count>0 then do;
    itemsum+price; itemsnm+price;
    outsum+price; outsnm+price;
    qrandsnm dollar13.2;
    if not end then put ' .';
    put a9 'The total price for item ' a9 ' is ' a9 ' $ ' a9 ' grand sum';
    count=0;
    end;
  if count>1 then do;
    itemsum+price; itemsnm+price;
    outsum+price; outsnm+price;
    qrandsnm dollar13.2;
    if not end then put ' .';
    put a9 'The total price for item ' a9 ' is ' a9 ' $ ' a9 ' grand sum';
    count=0;
    end;
end;

In the output generated below, note that each time ITEM changes,
a line of text is printed out describing the dollar value printed to its
right. Also note the text printed on the summary line for grand totals
at the end of the report.
It quickly becomes obvious that even small changes to the original PROC PRINT report are a major undertaking. In this light, compare only the last two steps slightly modified.

1. Invoke PROC REPORT with the FS option. Note that the variables in this initial screen are ordered the same as they were in the first report.

2. Move variables to the following order: ITEM, OUTLET, CUSTOMER, INVDATE, INVOICE, PRICE. (This is done by placing the cursor on the variable to be moved, hitting the MOVE key, placing the cursor on the variable around which the first variable is to be moved, and hitting either ADD_LEFT or ADD_RIGHT. The original variable selected is added to the left or to the right of the second variable selected).

3. Individually, select OUTLET and ITEM and hit the DEFINE key. Once in the DEFINITION window, select a USAGE of ORDER variables.

4. In the same manner as just described, define CUSTOMER to have a WIDTH=17 and a FORMAT=Dollar13.2. Looking ahead to step 5, PRICE has an insufficient WIDTH value and must be increased to 13. Therefore, for all variables to fit on the screen the WIDTH of CUSTOMER must be decreased to 17.

5. Define PRICE to have a USAGE VALUE of STATISTIC=SUM, WIDTH=13, and FORMAT=Dollar13.2.

6. Select OUTLET and hit the BREAK key. In the LOCATION window, select AFTER DETAIL and the OK button. In the BREAK window, select OVERLINE, SKIP, and SUMMARIZE.

7. BREAK on ITEM, AFTER DETAIL, and this time select only the EDIT THE PROGRAM button from the BREAK window. Now type the following in the COMPUTE window:

   Line #25 'Total for ITEM=$9.4 is PRICE.SUM :dollar13.2;
   Line 

8. Select EDIT from the action bar, R DBREAK from the pull-down window, EDIT THE PROGRAM and type the following in the COMPUTE window:

   Line #25 'Total across all Boat Types is ' PRICE.SUM :dollar13.2;

In the last two steps, note EDIT THE PROGRAM is chosen in order to get to the COMPUTE window. From the COMPUTE window, LINE statements are used to write out the text and variable values at selected breaks in the detail report. Note also the convention for referencing the SUM of PRICE (varname, statistic). What follows are several pages of the report generated by the steps above.

Screen 10 Modified PROC PRINT Report

Screen 11 Modified PROC PRINT Report

Screen 12 Modified PROC REPORT Report - First Page

Screen 13 Modified PROC REPORT Report
PROC REPORT can generate the exact report displayed above, but it involves some repetition to produce all the divider lines. Therefore, what follows are the steps necessary to get output in the same format but with fewer divider lines.

1. Invoke PROC REPORT with the FS option.
2. Select SALARY and hit the DELETE key.
3. Select SEX and hit the DEFINE key. Once in the DEFINITION window, select a USAGE of ACROSS, WIDTH=2 and FORMAT=82.
4. DEFINE DEPT to have a USAGE of GROUP.
5. ADD_Below SEX the data set variable SALARY.
6. DEFINE SALARY to have a USAGE of VALUE, STATISTIC=SUM.
7. Select SALARY and hit the ADD_Below key. From the ADDING window select STATISTIC and the OK button. From the STATS window, select the N and SUM statistics, then END out of that window.
8. ADD_RIGHT of DEPT computed variable V, character data, length=1. Edit the Program, and type V=' ';
9. ADD_Below the statistic N a TITLE and type consecutive dashes in the space added.
10. ADD_Below the statistic SUM a TITLE and type consecutive dashes in the space added.
11. ADD_ABOVE the data set variable SALARY a TITLE and type consecutive dashes in the space added.
12. ADD_Below the data set variable SEX a TITLE and type consecutive dashes in the space added.

After the last change has been made in step 12, above, the final report looks like this:
Another important fact about PROC REPORT is that it incorporates the functionality of an extremely popular SUGI supplemental procedure named PROC QPRINT. PROC QPRINT was popular for its ability to generate output similar to that produced by PROC PRINT, and it allowed more user control over the page layout. It did not, however, offer the ability to calculate statistics of any kind.

A few of the nicer capabilities that PROC QPRINT had that have been incorporated into PROC REPORT are the ability to wrap an observation until it has exhausted all the variables in the data set, the ability to span selected columns with a title, allowing underlining of selected titles, and the ability to print a multiple column report (phone book style). The following PROC QPRINT example, run in Version 5 of the SAS System, demonstrates the latter three functionalities.

```
libname in 'sasid.your.library';
proc qprint data=in.personl multiple=1 zone=1 underline;
head 'Confidential Information' 1=1;
head 'Employee' x 1=1;
var employee salary;
tail;
format employee $17. salary dollar 1 g. 2;
run;
```

The output above demonstrates a multiple column listing, where as many observations as will fit on the page are printed in one column before printing out the next. Also demonstrated are the abilities to both underline a range of variables and span a range with a heading. Specifically, EMPLOYEE and SALARY are underlined and the title “Confidential Information,” followed by a line of dashes, spans these same variables.

The example that follows further demonstrates the extreme range of interactive flexibility that PROC REPORT possesses. With just a few commands, PROC REPORT can even duplicate this type of special report.

1. ADD BELOW EMPLOYEE a TITLE. Type a row of consecutive dashes in the space added below EMPLOYEE.
2. ADD BELOW SALARY a TITLE. Type a row of consecutive dashes in the space added below SALARY.
3. ADD RIGHT of EMPLOYEE a COMPUTED VARIABLE.
   a. VARIABLE NAME=d
   b. CHARACTER DATA=yes
   c. LENGTH=1
4. In the COMPUTE window, type d=';'
   a. Select FILE from the action bar.
   b. Select END from the pull-down window.
   c. Select OK to end out of the ADDING window.
5. Type a dash (-) over the column heading of the computed variable D.
6. With EMPLOYEE highlighted, select EDIT from the action bar.
7. Select SPAN from the pull-down menu.
8. Highlight SALARY and hit the ADD ABOVE key.
9. Add a TITLE. Type a row of consecutive dashes in the space added above the two columns.
10. ADD ABOVE this title another TITLE. Type "CONFIDENTIAL INFORMATION" in the space added above.
11. Select LOCALS from the action bar.
   a. Select ROPTIONS from the pull-down window.
   b. Within the ROPTIONS window, set PANELS=2.
PROC REPORT is a new, totally interactive procedure in Version 6 of base SAS software. It is as simple to use as PROC PRINT and can generate the most complicated reports, which used to require laborious DATA step coding. It is capable of generating tabular reports with many statistics similar to PROC TABULATE output. Also provided is the ability to generate special reports previously available only through the SUGI supplemental procedure PROC QPRINT. Combined with these features is the ability to store and retrieve report definitions generated with PROC REPORT in a SAS data library. All these things make PROC REPORT the report-writing tool of the future in Version 6 of base SAS software.

SAS is a registered trademark of SAS Institute Inc., Cary, NC, USA.