

An Introduction to PROC REPORT

Kim L. Kolbe Ritzow of Systems Seminar Consultants, Kalamazoo, MI

Due to the length of this paper output to these examples will not appear in the Proceedings. Copies of the paper in full can be obtained from the author.

Abstract

PROC REPORT is a powerful new Base SAS[®] Procedure in Version 6.07 of the SAS System[®]. It combines the functionality of other Base SAS PROCs such as PRINT, MEANS, FREQ, and TABULATE into a single PROC step.

This paper is designed for users with little or no knowledge of PROC REPORT and who would like to learn its basic syntax. This presentation will discuss ordering of data, grouping data, computing new variables, producing report breaks, and incorporating user-defined formats.

In addition to combining functionalities of other Base SAS PROCs, PROC REPORT also provides unique reporting capabilities not found in other Base SAS PROCs (for example, customized breaks and multi-column reports).

Background

PROC REPORT can be run in batch, in a non-windowing environment using the Display Manager (meaning you submit the PROC REPORT statements just like you would submit other SAS statements), or by using a windowing environment in the Display Manager (PROC REPORT has its own windows associated with it that you can use to guide you through the creation and modification of the reports).

While the major selling point of PROC REPORT is its windowing environment which is interactive and allows you to see the results immediately (WSYWIG-like), it is not used often in practice for three reasons. One, most people still do not have mice connected to their mainframe terminals. Secondly, it is only effective when developing against a small amount of data. A third, the more you become familiar with PROC REPORT's syntax, the more it slows you down.

This paper will concentrate on the syntax of PROC REPORT. It will not illustrate the

development of reports using the windowing environment because of the number of slides it would require to show each click of the mouse. The syntax shown in this paper can either be submitted in the Display Manager, if you are working on the PC or interactively on the mainframe, or submitted in batch if on the mainframe.

A Simple PROC REPORT

A PROC REPORT without any statements produces a default report for all of the variables and observations on the data set. PROC REPORT will determine the layout of the report and will control the appearance of the report, which may produce some undesirable results:

```
PROC REPORT DATA=SASDATA.STOREDAT NOWINDOWS;  
RUN;
```

(see Output #1 for results)

The columns of information can be limited on the reports with a COLUMNS statement, or with a DROP= or KEEP= option behind the data set name:

```
PROC REPORT DATA=SASDATA.STOREDAT NOWINDOWS;  
  COLUMNS REGION STORE DEPT SALEAMT;  
RUN;
```

(no output for this syntax)

(OR...)

```
PROC REPORT DATA=SASDATA.STOREDAT  
  (DROP=SALEDATE) NOWINDOWS;  
RUN;
```

(no output for this syntax)

The COLUMNS and DEFINE Statements

The COLUMNS statement names and specifies the variables as they will appear on the report in order from left to right. The DROP= and KEEP= options will produce the same report as the COLUMNS statement, but the order of the columns will be determined by the order they appear on the input data set. The order of the variables cannot be controlled with DROP= and KEEP= options.

The DEFINE statement lists the attributes of each of the variables appearing on the report:

```
PROC REPORT DATA=SASDATA.STOREDAT NOWINDOWS;
  COLUMNS REGION STORE DEPT SALEAMT;
  DEFINEREGION/DISPLAYWIDTH=6ORDER=FORMATTED
    'Region';
  DEFINE STORE/DISPLAY ORDER=FORMATTED 'Store';
  DEFINE DEPT/DISPLAY ORDER=FORMATTED 'Dept';
  DEFINE SALEAMT/ANALYSIS SUM FORMAT=DOLLAR10.2
    WIDTH=10 'Amount/of Sale';
RUN;
```

(see Output #2 for results)

The DEFINE statements and the options specified can come in any order.

If no DEFINE statement is used, or a DEFINE statement is used without a usage option, character variables default to DISPLAY and numerics to ANALYSIS and calculate the SUM statistic. Be careful of the numeric default of ANALYSIS for SAS date variables, while internally they are stored as a number, they need to be treated as a DISPLAY variable. Rather than letting SAS choose the default usage, you should always explicitly define the usage option on the DEFINE statement. Valid usage options on the DEFINE statement are:

DISPLAY	ACROSS	ORDER
GROUP	ANALYSIS	COMPUTED

We will see examples of most of these, except for the ACROSS usage option.

If a variable is defined with the ANALYSIS usage option, the statistic used is assumed to be SUM unless another statistic is specified. Valid statistics that can be used on the ANALYSIS usage option are:

CSS	NMISS	SUMWGT
CV	PRT	T
MAX	RANGE	USS
MEAN	STD	VAR
MIN	STDERR	
N	SUM	

Note: The statistic specified applies to when grouping is in effect on the report. If there is no grouping on the report, the value of the ANALYSIS variable will be the value of itself.

The WIDTH= option on the DEFINE statement is used to control the width of the column. By

default PROC REPORT uses the internal variable lengths to determine the width of the columns, which may not be sufficient to accommodate the variable names and/or labels.

The FORMAT= option is used to control how the value will be formatted on the report. Make sure the WIDTH= option contains a value that is adequately large enough to handle the formatted value. If you increase the format value, you probably will have to increase the WIDTH= value. However, the opposite is not always true, just because you increase the WIDTH= value it DOES NOT mean you have to increase the FORMAT= value.

SAS-defined or user-defined formats can be specified on the FORMAT= option. Examples of user-defined formats will be shown later in the paper.

The ORDER= option specifies the order of the observations on the report. The default order with PROC REPORT is FORMATTED (based on the formatted value rather than the value itself). THIS IS DIFFERENT FROM OTHER BASE SAS PROCEDURES. All other SAS Procedures use the default ORDER=INTERNAL. PROC REPORT's default may change in future releases to be consistent with other SAS Procedures. Specify ORDER=FORMATTED even though it is the current default. This will assure that PROC REPORT will generate the same report even if the default should change.

Other values that can be specified on the ORDER= option are ORDER=DATA (uses the order of the data coming in), ORDER=INTERNAL (unformatted value), ORDER=FREQ (order of the most popular occurrences), or ORDER=FORMATTED (order of the formatted value, which is the current default). All of these options assume ascending order except ORDER=DATA (which takes the internal order of the data). To reverse the order, use the DESCENDING option in addition to the ORDER= option (see the next example as to how this might be used).

Labels can be associated on the DEFINE statement by placing up to forty characters in single or double quotes. The default split operator in PROC REPORT is the '/' character. When specified in the label, the label will split (or

stack) the column heading at the specified location. The default split character can be changed from '/' to any other character by using the SPLIT= option on the PROC REPORT statement (just like you do in PROC PRINT).

The ORDER Usage Option

The ORDER usage option will order or "sort" the data. The order in which is uses depends on the value of the ORDER= value being used. Variables with the ORDER usage option must be the left-most variables on the report, otherwise it will appear as if it is not ordering the data. The COLUMN statement controls how the variables will appear on the report from left to right:

```
PROC REPORT DATA=SASDATA.STOREDAT NOWINDOWS;
COLUMNS REGION STORE DEPT SALEAMT;
DEFINE REGION/ORDER WIDTH=6 ORDER=FORMATTED
'Region';
DEFINE STORE/ORDER ORDER=FORMATTED 'Store';
DEFINDEPT/ORDERDESCENDINGORDER=FORMATTED
'Dept';
DEFINE SALEAMT/ANALYSIS SUM FORMAT=DOLLAR10.2
WIDTH=10 'Amount/of Sale';
RUN;
```

(see Output #3 for results)

Notice that when a variable has a usage of ORDER it automatically suppresses repetitive values (much like the ID statement does in PROC PRINT). THERE IS NO WAY TO PREVENT THIS FROM HAPPENING WHEN A VARIABLE HAS A USAGE DEFINED AS ORDER.

The GROUP Usage Option

The GROUP Usage Option will "group" the data (acts like a CLASS statement on PROC SUMMARY or PROC MEANS). It is used to aggregate data together that contain like values:

```
PROC REPORT DATA=SASDATA.STOREDAT NOWINDOWS;
COLUMNS REGION STORE DEPT SALEAMT;
DEFINE REGION/GROUP WIDTH=6 ORDER=FORMATTED
'Region';
DEFINE STORE/GROUP ORDER=FORMATTED 'Store';
DEFINDEPT/GROUPDESCENDINGORDER=FORMATTED
'Dept';
```

```
DEFINE SALEAMT/ANALYSIS SUM FORMAT=DOLLAR10.2
WIDTH=10 'Amount/of Sale';
RUN;
```

(see Output #4 for results)

When GROUPing many of the same things hold true as when ORDERing in that repetitive values are automatically suppressed, variables being

grouped must be left-most, the variables being grouped are always returned in order based on the ORDER= value, and no other variable can be defined with an ORDER usage option when GROUPing is in effect.

Computed Variables

New variables can be dynamically created in PROC REPORT by using a COMPUTE block:

```
PROC REPORT DATA=SASDATA.STOREDAT NOWINDOWS
HEADLINE HEADSKIP;
COLUMNS REGION STORE DEPT SALEAMT TAX NET;
DEFINE REGION/GROUP WIDTH=6 ORDER=FORMATTED
'Region';
DEFINE STORE/GROUP ORDER=FORMATTED 'Store';
DEFINDEPT/GROUPDESCENDINGORDER=FORMATTED
'Dept';
DEFINE SALEAMT/ANALYSIS SUM FORMAT=DOLLAR10.2
WIDTH=10 'Amount/of Sale';
DEFINE TAX/COMPUTED FORMAT=DOLLAR10.2
WIDTH=10 'Sales/Tax';
DEFINE NET/COMPUTED FORMAT=DOLLAR10.2
WIDTH=10 'Net/Sale';
```

```
COMPUTE TAX;
TAX=SALEAMT.SUM * .05;
ENDCOMP;
```

```
COMPUTE NET;
NET=SALEAMT.SUM - TAX;
ENDCOMP;
RUN;
```

(see Output #5 for results)

When creating new variables when grouping is in effect, it is important that the usage of the variable is specified in the COMPUTE block, such as we have done with SALEAMT.SUM. If you don't, the values will display as missing on the report and a message will appear in the log stating that the grouping did not take place.

If your report is not grouping properly check to make sure of the following:

- * Computed variables are defined with a statistic usage (SALEAMT.SUM)
- * There are no variables defined with an ORDER usage option
- * That you are GROUPing by enough variables
- * GROUP variables are the left-most columns on the report.

Subtotaling and Grand Totaling Data

Subtotals and grand totals can be created on the report with the use of the BREAK and RBREAK statements:

```
PROC REPORT DATA=SASDATA.STOREDAT NOWINDOWS
  HEADLINE HEADSKIP;
```

```
  COLUMNS REGION STORE DEPT SALEAMT TAX NET;
```

```
  DEFINE REGION/GROUP WIDTH=6 ORDER=FORMATTED
    'Region';
```

```
  DEFINE STORE/GROUP ORDER=FORMATTED 'Store';
```

```
  DEFINEDept/GROUPDESCENDINGORDER=FORMATTED
    'Dept';
```

```
  DEFINE SALEAMT/ANALYSIS SUM FORMAT=DOLLAR10.2
    WIDTH=10 'Amount/of Sale';
```

```
  DEFINE TAX/COMPUTED FORMAT=DOLLAR10.2
    WIDTH=10 'Sales/Tax';
```

```
  DEFINE NET/COMPUTED FORMAT=DOLLAR10.2
    WIDTH=10 'Net/Sale';
```

```
  COMPUTE TAX;
    TAX=SALEAMT.SUM * .05;
  ENDCOMP;
```

```
  COMPUTE NET;
    NET=SALEAMT.SUM - TAX;
  ENDCOMP;
```

```
  BREAK AFTER REGION/OL SKIP SUMMARIZE SUPPRESS;
```

```
  RBREAK AFTER/DOL DUL SUMMARIZE;
```

```
  RUN;
```

(see Output #6 for results)

The options available on the BREAK and RBREAK statements are basically the same, except BREAK supports the SUPPRESS option and the RBREAK does not. OL request a single overline, whereas DOL requests a double overline. If you specify both options on the same statement, the OL option takes precedence. UL request a single underline, DUL requests a double underline. Similarly, if you specify both options on the same statement, the UL option takes precedence. SKIP will skip one line before the next group begins printing. SUMMARIZE will summarize the data at the group level based on the specified statistic DEFINED (since SALEAMT was defined as an ANALYSIS SUM statistic, SUMMARIZE will SUM the data at the group level). SUPPRESS will suppress the group value from printing on the left-hand side of the report on the break line.

Rather than sub and grand totaling after a group, we could also total before a group. To do so,

specify BREAK BEFORE or RBREAK BEFORE. This would print the total before the detail.

Another option available on the BREAK and RBREAK statement is the PAGE option, which will go to a new page after each break.

Make sure as you add BREAK and RBREAK statements that use the SUMMARIZE option that you increase your variable's format widths accordingly to accommodate the size of the totaled value.

User-Defined Formats

User-defined formats can be associated with variables on PROC REPORT. Assuming our data had values for the variable REGION as 'N', 'S', 'E', and 'W' rather than the actual value itself ('North', 'South', 'East', and 'West'). We could create a user-defined format using PROC FORMAT and associate the format on the DEFINE statement:

```
PROC FORMAT;
  VALUE $REGIONF
    'N'='NORTH'
    'S'='SOUTH'
    'E'='EAST'
    'W'='WEST'
;
RUN;
```

```
PROC REPORT DATA=SASDATA.STOREDAT NOWINDOWS
  HEADLINE HEADSKIP;
```

```
  COLUMNS REGION STORE DEPT SALEAMT TAX NET;
```

```
  DEFINE REGION/GROUP WIDTH=6 FORMAT=$REGIONF.
    ORDER=FORMATTED 'Region';
```

```
  DEFINE STORE/GROUP ORDER=FORMATTED 'Store';
```

```
  DEFINEDept/GROUPDESCENDINGORDER=FORMATTED
    'Dept';
```

```
  DEFINE SALEAMT/ANALYSIS SUM FORMAT=DOLLAR10.2
    WIDTH=10 'Amount/of Sale';
```

```
  DEFINE TAX/COMPUTED FORMAT=DOLLAR10.2
```

```
    WIDTH=10 'Sales/Tax';
```

```
  DEFINE NET/COMPUTED FORMAT=DOLLAR10.2
```

```
    WIDTH=10 'Net/Sale';
```

```
  COMPUTE TAX;
    TAX=SALEAMT.SUM * .05;
  ENDCOMP;
```

```
  COMPUTE NET;
    NET=SALEAMT.SUM - TAX;
  ENDCOMP;
```

```
  BREAK AFTER REGION/OL SKIP SUMMARIZE SUPPRESS;
  RBREAK AFTER/DOL DUL SUMMARIZE;
```

```
  RUN;
```

(see Output #7 for result)

Be careful with user-defined formats when creating custom report breaks- odd results may occur.

Improving the Appearance of the Report

A number of SAS statements and options can be used to improve the appearance of the report, like titles, footnotes, system options, and a WHERE clause:

```
PROC REPORT DATA=SASDATA.STOREDAT NOWINDOWS
  HEADLINE HEADSKIP;
```

```
TITLE1 'AREA SALES FOR SELECT DEPARTMENTS';
TITLE2 'FOR THE WEEK OF 5/02/94 - 5/08/94';
FOOTNOTE 'PREPARED BY MARKET RESEARCH';
```

```
OPTIONS LS=132 PS=80;
```

```
WHERE DEPT IN ('PRODUCE' 'MEAT');
```

```
COLUMNS REGION STORE DEPT SALEAMT TAX NET;
```

```
DEFINE REGION/GROUP WIDTH=6 ORDER=FORMATTED
  'Region';
```

```
DEFINE STORE/GROUP ORDER=FORMATTED 'Store';
```

```
DEFINEDept/GROUPDESCENDINGORDER=FORMATTED
  'Dept';
```

```
DEFINE SALEAMT/ANALYSIS SUM FORMAT=DOLLAR10.2
  WIDTH=10 'Amount/of Sale';
```

```
DEFINE TAX/COMPUTED FORMAT=DOLLAR10.2
  WIDTH=10 'Sales/Tax';
```

```
DEFINE NET/COMPUTED FORMAT=DOLLAR10.2
  WIDTH=10 'Net/Sale';
```

```
COMPUTE TAX;
  TAX=SALEAMT.SUM * .05;
ENDCOMP;
```

```
COMPUTE NET;
  NET=SALEAMT.SUM - TAX;
ENDCOMP;
BREAK AFTER REGION/OL SKIP SUMMARIZE SUPPRESS;
RBREAK AFTER/DOL DUL SUMMARIZE;RUN;
  (see Output #8 for results)
```

The WHERE clause process the SAS data set named on the DATA= option. Therefore, computed variables cannot be checked on the WHERE clause, nor can the grouped value of a variable be checked (that is, if you said if SALEAMT >= 2500 it would refer to the detail value in the data set, not the grouped value).

In Summary

PROC REPORT is a powerful new Base SAS Procedure which provides us with more control over our report's appearance than any other existing Base SAS Proc.

We have seen in the course of this paper how PROC REPORT can order and group data, how it can compute new variables, produce report breaks, and incorporate user-defined formats.

While not discussed in this paper, PROC REPORT can do things that no other SAS Procedure is capable of: creating custom report breaks, share columns, and create multi-column reports.

Trademark Notice

SAS is a registered trademark of the SAS Institute Inc., Cary, NC, USA and other countries.

Useful Publications

SAS Institute Inc. (1990), SAS® Guide to the Report Procedure, Usage and Reference, Version 6, First Edition, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1993), SAS® Technical Report P-258, Using the REPORT Procedure in a Nonwindowing Environment, Release 6.07, Cary, NC: SAS Institute Inc.

Any questions or comments regarding the paper may be directed to the author:

Kim L. Kolbe Ritzow
Systems Seminar Consultants
Kalamazoo Office
927 Lakeway Drive
Kalamazoo, MI 49001
Phone: (616) 345-6636
Fax: (616) 345-5793