

Collating Printed Output from Multiple PROC Steps

Howard Schreier, U.S. Department of Commerce

Printed output from a SAS® job is typically organized like this:

```

First PROC step
  Output for first BY group
  Output for second BY group
  ...
Second PROC step
  Output for first BY group
  Output for second BY group
  ...
  ...

```

Often, what you really need is this:

```

First BY group
  Output from first PROC
  Output from second PROC
  ...
Second BY group
  Output from first PROC
  Output from second PROC
  ...
  ...

```

One solution is to divert the print output to disk and add a post-processing operation to collate the page images before they are printed. I'll illustrate the technique, but first I want to generate some test data:

```

data sample;
do i = 9 to 10;
  do j = 4 to 5;
    do k = 1 to i*j;
      x = ranuni(0);
      output;
    end;
  end;
end;
run;

```

By the way, all of the code I'm presenting was run using SAS 6.08 under batch MVS.

I and J are my BY variables. For each BY group, I want to print, on consecutive pages, a bar chart of the distribution of X and a table with counts and means for X.

First, I establish a scratch file to capture print lines and employ PROC PRINTTO to divert output to that file:

```

filename printemp
  'tempo.print.file';

proc printto
  print=printemp new;
run;

```

Next, I run the first procedure (CHART). In the TITLE1 statement you will see the first critical part of the technique. The title text begins with the values of the two BY variables, separated from each other and from the "real" title ("Bar Charts" in this example) by a delimiter character (the

vertical bar) which I know cannot appear elsewhere in the line. If you are unfamiliar with the mechanism for embedding BY variables in titles, it's probably because it's a Release 6.07 enhancement to the TITLE statement, documented only in a Technical Report.

```
proc chart data=sample;
by i j; vbar x;
title1 '#byval1|#byval2|'
'Bar Charts';
run; title1;
```

Next, after switching print output back to its default destination, I read the captured page images into a SAS data set. Each observation in this data set will contain one print line (PLINE) plus the corresponding BY variables (I, J) and carriage-control character (COL1):

```
proc printto; run;

data p_chart;
infile printemp;
input coll $ 1
      pline $char132.;
if coll='1' then do;
  i = input(scan(left
    (pline),1,'|'),32.);
  retain i;
  j = input(scan
    (pline, 2,'|'),32.);
  retain j;
  pline = scan(pline, 3,'|');
end;
run;
```

The coding here sheds some light on what was done earlier, in the PROC CHART step. The BY variables were embedded in the TITLE1 line because that is the first line on each page and therefore has a distinctive signature (in

this case, a value of "1" in the carriage-control column). A one-character delimiter (the vertical bar) was used to support the operation here of the SCAN function to extract the BY values.

Next, I carry out same four steps for PROC TABULATE, to generate my statistics. Since I've already processed the bar charts in my scratch file, I can use it again.

```
proc printto
  print=printemp new; run;

proc tabulate data=sample;
by i j; var x;
table n mean, x;
title1 '#byval1|#byval2|'
'TABULATE Stats';
run; title1;

proc printto; run;

data p_tabul; infile printemp;
input coll $ 1
      pline $char132.;
if coll='1' then do;
  i = input(scan(left
    (pline),1,'|'),32.);
  retain i;
  j = input(scan
    (pline, 2,'|'),32.);
  retain j;
  pline = scan(pline, 3,'|');
end;
run;
```

Now I'm ready to pull everything together the way I want it and route it to the printer. I use a SET statement accompanied by a BY statement to interleave the CHART and TABULATE pages:

```

data _null_;
options nodate nonumber;
set p_chart p_tabul; by i j;
file print noprint ls=133;
put @1 col1 $1.
    @2 pline $char132.;
run;

```

The NODATE and NONUMBER system options and the NOPRINT option on the FILE statement are appropriate because dates, page numbers, and carriage-control characters were generated by the PROCs and are passed through by the PUT statement.

I've left some rough edges. Extracting the BY values from the TITLE1 lines will un-center real title strings and un-justify date/time stamps and page numbers. Also, page numbers generated in the PROC steps will obviously end up out of sequence; in addressing this, you have the opportunity to number pages in each BY group independently.

I think this technique will work on most if not all platforms. The critical thing is to identify a distinct "signature" of the first line of a printed page. Techniques will vary by platform and will be sensitive to platform-specific nuances. For example, under Windows, each TITLE1 line begins with a hexadecimal "0C", but there is no reserved carriage-control column (instead, the rest of the title is displaced by one byte); this, among other things, necessitates changes the specific implementation, although the overall approach shown here does in fact work.

References

All of the following are publications of SAS Institute, Cary NC:

- SAS Technical Report P-222: Changes and Enhancements to Base SAS Software, Release 6.07*
- SAS Language Reference, Version 6, First Edition*
- SAS Procedures Guide, Version 6, Third Edition*
- SAS Companion for the MVS Environment, Version 6, First Edition*
- SAS Companion for the Microsoft Windows Environment, Version 6, First Edition*

SAS is a registered trademark of SAS Institute Inc.

Author Contact Information

Howard Schreier
U.S. Dept. of Commerce
Mail Stop H-2814-B
Washington DC 20230

(202) 482-4180 (voice)
(202) 482-4614 (fax)
his@cu.nih.gov