

Programmatic handling of varying report requirements.

Jim Snider, AmSouth Bank, Birmingham AL

INTRODUCTION

Below is a simple example of using PROC SQL and the SQLOBS macro variable to handle situations where there is some uncertainty or variation in the source data. This example should allow you to meet such challenges without resorting to a solution that requires re-coding based on the contents of the input data.

PRODUCE VARYING NUMBER OF REPORTS

Objective: At the end of the month, produce a report for each state that had revenue. Any state that did not have revenue will not be represented in the input data. There can be as many as 51 entities, 50 states plus Puerto Rico. Our test data will be simplified considerably, with only 3 columns (state, date, total_revenue) and 6 rows.

First, we will read the input data:

```
data st_data ;
  input state $2. +1 eom_dt mmdyy10. +1 total_rev COMMA10.2 ;
  format eom_dt mmdyy10. total_rev comma10.2 ;
  if _N_ = 1 then
    call symput("RPTMONTH", put(eom_dt, mmdyy10.));
cards;
NC 08/31/2004 1,243.25
SC 08/31/2004 721.34
GA 08/31/2004 108.00
TN 08/31/2004 230.00
FL 08/31/2004 469.50
CA 08/31/2004 75.00
run;
```

Next, we will use the SQL procedure to capture, in macro variables, the states that had revenue in August. Proc SQL will create macro variables ST1-ST6 that will hold the 2-character state abbreviations.

```
proc sql noprint ;
  select state
  into :st1-:st51
  from st_data ;
quit;
```

Proc SQL creates a macro variable called sqlobs that is set to the total number of rows in the st_data file. We will create our own macro variable called MAXST to capture that value.

```
%let maxst = &sqlobs ;
```

Now, we will produce our reports. The macro DO_RPTS will iterate 6 times.

```
%macro dorpts ;

  %do i=1 %to &sqlobs ;

    ods listing close;
    ods html file="c:\sas\&&st&i...htm" ;

    title "Monthly report for &&st&i" ;
    title2 "as of &rptmonth" ;

    proc report data=st_data missing nowd ;
```

```

        where state = "&&st&i" ;
        col state total_rev ;
        DEFINE state / GROUP FORMAT=$5. WIDTH=5 SPACING=2 LEFT "area" ;
        DEFINE total_rev / SUM FORMAT=COMMA10.2 WIDTH=15 SPACING=2 RIGHT "Total Revenue" ;
run;
ods html close ;
%end ;

%mend;

%dorpts ;

```

The macro will produce 6 HTML files with separate reports for each state. The state abbreviation will be displayed in the heading.

CONCLUSION

This a simple, straightforward example of SAS code that, in my opinion, can be useful in many programming situations. Even if your knowledge of macro programming is limited or you have no interest in using PROC SQL, I believe that you may still find it beneficial to make use of this code in certain programming situations.

CONTACT INFORMATION

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

Jim Snider
AmSouth Bank

james.snider@amsouth.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.