

Using SAS® to Create and Run SAS®

Joseph M. Filip, First Card

Utilizing the infamous Data _NULL_, writing SAS® code to create and execute SAS® code is not difficult.

The primary instance encountered for the use of this technique is when the processing is of a variable or volatile nature. It can save on maintenance for programs that are run regularly. It can also save on the need for typing several different values for a variable in an ad hoc report.

This all has to start with a controlling file or dataset. It contains information that the static processing needs. It may create a WHERE clause for a particular extract or even a PROC FORMAT to use in logic or reporting. This can work exceptionally well when using larger data files.

The first step is to identify what the controlling file is going to be. Determine how you need to use the data and decide what program logic needs to be done. Manipulate it, then write it to a temporary flat file. Include it and your program's done.

Following is example code with a macro, global variables and a file of SAS® code written by a portion of this SAS® program. This code was written for the PC but the same techniques can be used on the mainframe.

```
*****
* SET UP GLOBAL VARIABLES TO NAME THE FILES
* AND PATHS. THESE CAN EASILY BE MODIFIED FOR
* TESTING.
*****

%let ice = "g:\ssd\pmpk\ice&dtc..TXT";
%let cms = "g:\ssd\imu\&center&dtc..cms";
%let cmsm = "g:\ssd\imu\&center&dtc.m.cms";
%let dvx = "g:\ssd\pmpk\&center&dtc..dvx";
%let team = "g:\ssd\imu";
%let dvxtil = "g:\ssd\reports\ibc\collect\latestag\dvx&dtl1..txt";

%macro merge(center);

data ICE;

*****
* This data step reads in a flat file with a record length of 79
*****

infile &ice lsr=79 missover;

input @1 id $4.
      @5 workof yymmdd6.
      @11 loc $6.
      @17 PMPK_DLR $11.
      @28 age_eff 11.
      @39 age_effid 11.
      @50 name $25.;

output;

run;

libname team &team ;
*****
```

```
* THIS SAS FILE CONTAINS THE NAME OF THE TEAM,
* WHAT REPORT THE DATA IS TO APPEAR ON, THE
* COMMON ID, AND THE STANDARD NUMBER OF
* HOURS FOR NON-FULL TIME EMPLOYEES. It starts
* out as an EXCEL spreadsheet, gets converted to an SD2 SAS
* file and used here.
*****
```

```
data teamname ;

set team.teamname (rename = (cms_id = cms davox_id = davox
ic_id = id));

output;

run;
*****
* FOR DIRECT ACCESS CONVERSION OF CMS
* IDENTIFIER TO THE COMMON ICE IDENTIFIER. The
* Identifier from the CMS file needs to be used to FETCH the
* name of the team that show up on the report
*****

proc sort data = teamname (where = (cms ne '' and teamname ne '' ))
nodupkeys out = cmrice ;
by cms;
run;

%include 'c:\team.txt'; /* bring the code that was just created
                        by the */
/* SAS program back into the program
                        so it can */
/* used by the next data step */
```

```
data cms (drop = stf str_dt end_dt );

*****
* read in the next file and convert the 'cmsid' to the 'common
* identifier'
*****

infile &cms lrecl=128 ;

input @1 prt_line $80.
      @15 cmsid $4.
      @39 stf Schar7.
      @7 str_dt Schar8.
      @16 end_dt Schar8. @;

if cmsid in ( ' ' ; ___ ; '—' ) then do;
input;
delete;
return;
end;

if stf in ( ' ' ; Hours ; ' ' Talk ; ' ' Staffed ) then do;
input;
delete;
return;
end;

input @1 cms_name $14.
      @19 acd 7.
      @27 call_out 5.
      @33 avg_out 5.2
      @39 stf_hrs 7.2
      @46 avg_talk 5.2
      @51 avg_xfer 6.2
      @57 avg_acw 6.2
      @63 Avg_hndl 6.2
```

```

@69 t1_aux 8.2
@77 t1_wait 8.2
@85 Pct_acd $6.
;
id = put(upcase(cmsid), Scmsid.); /* look up the cross
                                reference identifier */

workgrp = upcase(put(cmsid, Scteam.)) /* look up the team name */

output cms;

return;

.
.
.
.
Other SAS® Code
.
.
.
.
*****
* Merge all the data together by the common identifier *
*****
data merged;

    merge davox (in = d)
          cms   (in = c)
          ICE   (in = P);
    by id;

if name eq '' then
do;
    if dvx_name ne '' then
        name = upcase(dvx_name);
    else
        name = upcase(cms_name);
end;

rpt = put(id, Srpt.);

if rpt = id then rpt = ' ';

if substr(id, 1, 1) eq 'T' then
    indy = 'Y';
else
    indy = 'N';

workgrp = upcase(put(id, Steam.));

if substr(id, 1, 1) eq 'O' then
    workgrp = 'OVERLIMITS';

output MERGED;

run;

%amend;

%merge(e) /* executes the Macro and sets the value of center equal
           to E */

```

Another example using a Where clause

```

data _null_;

file 'c:\where.txt';

set cmice end = eof;
*****
* Do any set up for the chosen method first *
*****

if _n_ eq 1 then
    put @1 '(Where = (teamname in ('; /* set up the where
                                      statement */
                                      /* this will select only those
                                      data records */
                                      /* that meet the criteria */

put @10 '"" teamname $9. ""'; /* set values to be checked
                              for */

if eof then
    put @10 '));'; /* close the where*/

run;

data temp;
*****
* This step will now only select those records that match what was *
* in the where clause. *
*****
set readin;

%include 'c:\where.txt'

output;

```

Address

Questions can be addressed to:

Joe Filip
 First Card
 2500 Westfield Drive F-2
 Elgin, IL 60123
 (847) 888-6243
 (847) 888-6698 (Fax)
 Joseph_Filip@em.fcmbd.com

The full program is available on request, preferably through E-mail.