```
options nocenter;
 /*************************************************************/
 /* Create date variables with CALL symput.  &mreport_dt is set in */
 /* driver program.                                           */
 /*************************************************************/
DATA _NULL_;
  file_dt = intnx('month', "&sysdate9"d, -1, 'e');
  fystart = intnx('year.4', file_dt, 0);

 /*************************************************************/
 /* Calculate start and end dates for each quarter of fiscal year. */
 /*************************************************************/
  q1start_dt = intnx('month', fystart, 0);
  q1end_dt   = intnx('month', fystart, 2, 'e');
  q2start_dt = intnx('month', fystart, 3);
  q2end_dt   = intnx('month', fystart, 5, 'e');
  q3start_dt = intnx('month', fystart, 6);
  q3end_dt   = intnx('month', fystart, 8, 'e');
  q4start_dt = intnx('month', fystart, 9);
  q4end_dt   = intnx('month', fystart, 11, 'e');

  FORMAT file_dt    fystart
         q1start_dt q1end_dt
         q2start_dt q2end_dt
         q3start_dt q3end_dt
         q4start_dt q4end_dt DATE9.;

  CALL symputx('mfile_dt', put(file_dt, DATE9.));
  CALL symputx('mfystart', put(fystart, DATE7.));
  CALL symputx('mqstart1', put(q1start_dt, DATE9.));
  CALL symputx('mqend1', put(q1end_dt, DATE9.));
  CALL symputx('mqstart2', put(q2start_dt, DATE9.));
  CALL symputx('mqend2', put(q2end_dt, DATE9.));
  CALL symputx('mqstart3', put(q3start_dt, DATE9.));
  CALL symputx('mqend3', put(q3end_dt, DATE9.));
  CALL symputx('mqstart4', put(q4start_dt, DATE9.));
  CALL symputx('mqend4', put(q4end_dt, DATE9.));
RUN;

 /*************************************************************/
 /* Create SAS data set with data required for the month_ytd  */
 /* format.  The values are based on the fiscal year & file date.  */
 /* There should be 12 month ranges (1 for each month of the  */
 /* fiscal year), 4 quarter ranges, and 1 YTD range.  Set format   */
 /* name, indicate NOTSORTED & MULTILABLE with HLO = 'SM'.  Mprint */
 /* option is on so look in the log for resolution values.    */
 /*************************************************************/
%MACRO loopdates;
 DATA fiscal_year_dates;
   length fmt_value $10;
 /*************************************************************/
 /* Write out observations for each month of the fiscal year. */
 /*   start_dt  = SAS date for the 1st of the month           */
 /*   end_dt    = SAS date for the last of the month          */
 /*   fmt_value = # of month in Fiscal Year & month as MONYYYY     */
 /*************************************************************/
     %DO i = 1 %TO 12;
         start_dt = intnx('month', "&mfystart"d, &i-1);
         end_dt = intnx('month', "&mfystart"d, &i-1, 'e');
         fmt_value = put(&i, z2.) !! ' ' !! put(start_dt, MONYY7.);
       OUTPUT;
     %END;
 /*************************************************************/
```

```
 /* Write out observations for each quarter of the fiscal year.    */
 /*   start_dt = SAS date for the 1st day of the quarter           */
 /*   end_dt   = SAS date for the end day of the quarter           */
 /*   fmt_value = ## & Q# & Fiscal Year as yy                      */
 /*               (## is 81-84 where 2nd digit = qtr #)            */
 /*****************************************************************/
     %DO j = 1 %TO 4;
       start_dt = "&&mqstart&j"D;
       end_dt = "&&mqend&j"D;
       fmt_value = put(80+&j,z2.)!!" Q&j "!! put(year("&mfystart"D),z4.);
       OUTPUT;
     %END;
 /*****************************************************************/
 /* Write out observation for year to date of the fiscal year.     */
 /*   start_dt = SAS date for the 1st day of the fiscal year       */
 /*   end_dt   = SAS date for the end day of the report month      */
 /*   fmt_value = 99 YTD                                           */
 /*****************************************************************/
     start_dt = "&mfystart"D;
     end_dt = "&mfile_dt"D;
     fmt_value = '99 YTD';
   OUTPUT;
   FORMAT START_DT END_DT DATE9.;
 RUN;
%MEND loopdates;
 /*****************************************************************/
 /* End macro.  Call macro.                                       */
 /*****************************************************************/
%loopdates;

Data Sales;
  input @01 trannum $2.
        @04 emp_id  $3.
        @08 tran_dt date9.
        @18 pmt_amt comma10.;
  format tran_dt date9. pmt_amt dollar10.2;
datalines;
01 123 15apr2007 $542.16
02 123 27may2007 $153.84
03 123 20jun2007 $153.12
04 123 18jul2007 $234.51
05 123 20aug2007 $215.15
06 123 19sep2007 $324.67
07 123 21oct2007 $312.16
08 456 12apr2007 $154.15
09 456 21may2007 $271.96
10 456 20jul2007 $524.12
11 456 30aug2007 $341.72
12 456 18sep2007 $218.15
13 456 25oct2007 $52.10
14 123 12apr2007 $234.00
15 123 29may2007 $426.52
16 123 16jun2007 $124.51
17 123 13jul2007 $364.56
18 123 25aug2007 $1285.17
19 123 11sep2007 $623.18
20 123 08oct2007 $346.42
21 456 12apr2007 $435.94
22 456 29may2007 $546.23
23 456 16jul2007 $432.78
24 456 25aug2007 $128.17
25 456 11sep2007 $326.79
26 456 08oct2007 $752.84
```

```
;
run;

Proc print data=fiscal_year_dates noobs;
  title1 'Date Ranges and Formatted Values';
  title2 'to be used in PROC FORMAT';
run;
 /*************************************************************/
 /* Start demo                                                */
 /*************************************************************/

 /*************************************************************/
 /* Create data set in form required by proc format           */
 /*   hlo = 'SM' indicates non-sorted data and multilabel format   */
 /*************************************************************/
DATA formatdata;
  set fiscal_year_dates;
  retain fmtname 'fmtfydates'
         hlo     'SM';
  rename start_dt  = start
         end_dt    = end
         fmt_value = label;
run;

proc print data=formatdata;
  title 'Data to be used by PROC FORMAT to create multilable format';
run;

 /*************************************************************/
 /* Create user-defined format from SAS data set              */
 /*************************************************************/
PROC FORMAT CNTLIN = formatdata;
RUN;

proc summary data=sales completetypes nway;
  format tran_dt fmtfydates.;
  class tran_dt / mlf preloadfmt;
  output out=sales_sum
    mean(pmt_amt) = avg_sales
    sum(pmt_amt)  = total_sales;
run;

proc print data=sales_sum;
  title 'sales summary';
  var tran_dt avg_sales total_sales;
run;
```