Abstract

If you have ever used macros in SAS®, you may have wondered which options to turn on and which to turn off in order to understand what was really going on. You may have actually asked yourself: "How do I find the value of the macro variable during execution?" (SYMBOLGEN is the answer) or "What is the macro code doing?" (try MLOGIC) or "Where am I in the %DO loop?" (MLOGIC, again) or "Which macro generated what code?" (MPRINT will tell you!)

There are many questions which can be answered easily by using the right option! This paper will summarize some of the SAS System Options for use with MACRO and will provide a guide for using them!

Introduction

The Macro Language shouldn't scare new users away. Too often it does- and usually because the novice has so much difficulty debugging the programs. There are three simple options for displaying macro text: MLOGIC, MPRINT and SYMBOLGEN. The SAS system options which relate to the general operation of the macro facility are IMPLMAC, MACRO, MERROR AND SERROR. The autocall facility is managed by the following options: MAUTOSOURCE, MRECALL and SASAUTOS=.

Used correctly, these ten options allow a SAS programmer to easily work with the macro facility.

Options:

IMPLMAC | NOMIMPLMAC controls whether you can use statement-style macro invocations where the macro call looks like a SAS statement. This is explained in detail in the SAS® Guide to Macro Processing, Version 6.

MACRO | NOMACRO controls whether the macro facility is available and must be specified at SAS invocation or in a configuration file.

MERROR | NOMERROR controls whether unrecognized macro names and variable references cause error messages.

SERROR | NOSERROR controls whether the macro processor issues the warning message 'Warning: Apparent symbolic reference ... not resolved.' when a macro variable reference does not match a macro variable.

MLOGIC | NOMLOGIC MLOGIC completely traces the execution of the macro. (MLOGIC in V6.06 and above provides the same information as MTRACE in V6.03)

MPRINT | NOMPRINT specifies whether the macro processor displays SAS Statements generated by macro execution and causes the macro variable references and macro expressions to appear resolved.

SYMBOLGEN | NOSYMBOLGEN specifies whether the macro processor displays the result of resolving macro variable references.

MACROGEN | NOMACROGEN is no longer necessary because MPRINT provides the same information.

MAUTOSOURCE | NOMAUTOSOURCE controls whether the autocall facility is available.

MRECALL | NOMRECALL controls whether the macro processor searches the specified autocall libraries for a member that was not found in a previous search.

SASAUTOS=: Specifies one or more autocall libraries.

The Source Code:

The following code is the source code used in the example. At SAS invocation the MACRO and MAUTOSOURCE options are specified. The MACRO option makes the macro facility available and MAUTOSOURCE make the autocall facility available. Under MVS, the user defined autocall library is specified by using the filename SASAUTOS as follows:

```sas
//SASAUTOS DD
//    DD DSN=dsn,DISP=SHR
```
where *den* is the user defined PDS (partitioned dataset) whose members are SAS macros. This feature is not used in this example but is explained in detail in the *SAS® Guide to Macro Processing, Version 6, Second Edition*.

The macro PARMSIN reads data parameters from the jobcards in ddname PARMSIN to create global macro variables. DATAIN prints a SAS dataset. The dataset has a variable for each month through and including the current month. To print the data accurately, we use the macro facility to generate the VAR statement of the PROC PRINT. The current month is input into a global macro variable using CALL SYMPUT in PARMSIN.

```
%MACRO PARMSIN;
  DATA PARMS;
  INFILE PARMSIN OBS=3;
  INPUT @4 YEAR $4. /;
     @4 MONTH $3.;
  %GLOBAL YEAR; CALL SYMPUT ('YEAR',YEAR);
  %GLOBAL MONTH; CALL SYMPUT ('MONTH',MONTH);
  RUN;
%MEND PARMSIN;

%MACRO DATAIN;
  TITLE1 'NESUG MACRO PAPER';
  %IF &MONTH = JAN %THEN %DO; TITLE3 'JANUARY'; %LET COUNT = 1 ;%END;
    %ELSE %IF &MONTH = FEB %THEN %DO; TITLE3 'FEBRUARY'; %-LET COUNT 2; %END;
    %ELSE %IF &MONTH = MAR %THEN %DO; TITLE3 'MARCH'; %-LET COUNT = 3 ;%END;
    /; /* ... KEEP GOING THROUGH DECEMBER... YOU GET THE IDEA! */
  TITLE4 "YEAR = &YEAR";
  PROC PRINT DATA=DATA&YEAR..SALEDATA;
    /* GENERATE THE VAR STATEMENT ==> &COUNT OF THEM.
       THEY'RE CALLED SALES1 THROUGH SALES(COUNT)==>ONE FOR EACH MONTH */
  VAR COMPANY
     %DO I = 1 %TO &COUNT ; SALES&I %END; ;
%MEND DATAIN;

%PARMSIN;
%DATAIN;
  RUN;
/*
*/
```
The SAS Log (no options):

We will run the job anticipating that the VAR statement generated in the PROC PRINT should have COMPANY as the first variable, COUNT1 as the second variable and if the value of MONTH read into the macro PARMSIN is JAN, then the statement will end here. If the month is FEB, there will be a COUNT2, if it's MAR there will be a COUNT3, and so on. The third and fourth lines of the title will reflect the values of the global macro variables.

---

The SAS Log (no options):

We will run the job anticipating that the VAR statement generated in the PROC PRINT should have COMPANY as the first variable, COUNT1 as the second variable and if the value of MONTH read into the macro PARMSIN is JAN, then the statement will end here. If the month is FEB, there will be a COUNT2, if it's MAR there will be a COUNT3, and so on. The third and fourth lines of the title will reflect the values of the global macro variables.

---

The SAS Log (no options):

We will run the job anticipating that the VAR statement generated in the PROC PRINT should have COMPANY as the first variable, COUNT1 as the second variable and if the value of MONTH read into the macro PARMSIN is JAN, then the statement will end here. If the month is FEB, there will be a COUNT2, if it's MAR there will be a COUNT3, and so on. The third and fourth lines of the title will reflect the values of the global macro variables.

---

The SAS Log (no options):

We will run the job anticipating that the VAR statement generated in the PROC PRINT should have COMPANY as the first variable, COUNT1 as the second variable and if the value of MONTH read into the macro PARMSIN is JAN, then the statement will end here. If the month is FEB, there will be a COUNT2, if it's MAR there will be a COUNT3, and so on. The third and fourth lines of the title will reflect the values of the global macro variables.

---

The SAS Log (no options):

We will run the job anticipating that the VAR statement generated in the PROC PRINT should have COMPANY as the first variable, COUNT1 as the second variable and if the value of MONTH read into the macro PARMSIN is JAN, then the statement will end here. If the month is FEB, there will be a COUNT2, if it's MAR there will be a COUNT3, and so on. The third and fourth lines of the title will reflect the values of the global macro variables.
The SAS Output:

<table>
<thead>
<tr>
<th>OBS</th>
<th>COMPANY</th>
<th>SALES1</th>
<th>SALES2</th>
<th>SALES3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>345</td>
<td>672</td>
<td>889</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>543</td>
<td>345</td>
<td>546</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>345</td>
<td>672</td>
<td>889</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>354</td>
<td>548</td>
<td>849</td>
</tr>
<tr>
<td>(...)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>23</td>
<td>687</td>
<td>910</td>
<td>546</td>
</tr>
<tr>
<td>14</td>
<td>24</td>
<td>675</td>
<td>549</td>
<td>543</td>
</tr>
<tr>
<td>15</td>
<td>25</td>
<td>775</td>
<td>872</td>
<td>978</td>
</tr>
</tbody>
</table>

The output looks like we wanted it: a variable for each month through March. But how did it work? It almost looks like there may be a semicolon missing in the TITLE3 SAS statement, but how could that be? Everything worked!

Looking at the SAS log tells us nothing. Let's turn on some SAS OPTIONS to display macro text: MLOGIC, MPRINT, and SYMBOLGEN and reexamine the log.
The SAS Log with options specified:

THE SAS SYSTEM

NOTE: COPYRIGHT (C) 1989 BY SAS INSTITUTE...
NOTE: SAS (R) PROPRIETARY SOFTWARE RELEASE 6.08...
NOTE: SAS SYSTEM OPTIONS SPECIFIED ARE:
SORT=4 PS=MAX, LS=80, MACRO, MAUTOSOURCE, NODATE

NOTE: THE INITIALIZATION PHASE USED 0.35 CPU SECONDS AND 2652K.

1
2 %MACRO PARMSIN;
(... same as first log...)

32
33
34 %MEND DATAIN;
35
36 OPTIONS MLOGIC MPRINT SYMBOLGEN ;
37
38 %PARMSIN;
MLOGIC(PARMSIN): BEGINNING EXECUTION.
MPRINT(PARMSIN): DATA PARMS;
MPRINT(PARMSIN): INFILE PARMSIN OBS=3;
MPRINT(PARMSIN): INPUT @4 YEAR $4. / @4 MONTH $3. ;
MLOGIC(PARMSIN): %GLOBAL YEAR
MPRINT(PARMSIN): CALL SYMPUT ('YEAR',YEAR) ;
MLOGIC(PARMSIN): %GLOBAL MONTH
MPRINT(PARMSIN): CALL SYMPUT ('MONTH',MONTH) ;
MPRINT(PARMSIN): RUN;

NOTE: THE INFILE PARMSIN IS:
DSNAME=JES2.JOB00722.SI000101,
UNIT=SYSOUT, VOLUME, DISP=NEW, BLKSIZE=80,
LRECL=80, RECFM=FB

NOTE: 2 RECORDS WERE READ FROM THE INFILE PARMSIN.
NOTE: THE DATA SET WORK.PARMS HAS 1 OBSERVATIONS AND 2 VARIABLES.
NOTE: THE DATA STATEMENT USED 0.13 CPU SECONDS AND 3225K.

MLOGIC(PARMSIN): ENDING EXECUTION.
MLOGIC(DATAIN): BEGINNING EXECUTION.
39 %DATAIN;
MPRINT(DATAIN): TITLE1 'NESUG MACRO PAPER';
SYMBOLGEN: MACRO VARIABLE MONTH RESOLVES TO MAR
MLOGIC(DATAIN): %IF CONDITION &MONTH = JAN IS FALSE
SYMBOLGEN: MACRO VARIABLE MONTH RESOLVES TO MAR
MLOGIC(DATAIN): %IF CONDITION &MONTH = FEB IS FALSE
SYMBOLGEN: MACRO VARIABLE MONTH RESOLVES TO MAR
MLOGIC(DATAIN): %IF CONDITION &MONTH = MAR IS TRUE
MPRINT(DATAIN): TITLE3 'MARCH';
MLOGIC(DATAIN): $LET (VARIABLE NAME IS COUNT)
SYMBOLGEN: MACRO VARIABLE YEAR RESOLVES TO 1994
MPRINT(DATAIN): TITLE4 "YEAR = 1994";
SYMBOLGEN: MACRO VARIABLE YEAR RESOLVES TO 1994
MPRINT(DATAIN): PROC PRINT DATA= DATA1994.SALEDATA;
SYMBOLGEN: MACRO VARIABLE COUNT RESOLVES TO 3
MLOGIC(DATAIN): %DO LOOP BEGINNING; INDEX VARIABLE I; START VALUE IS 1; STOP VALUE IS 3; BY VALUE IS 1.
SYMBOLGEN: MACRO VARIABLE I RESOLVES TO 1
MLOGIC(DATAIN): %DO LOOP INDEX VARIABLE I IS NOW 2; LOOP WILL ITERATE AGAIN.
SYMBOLGEN: MACRO VARIABLE I RESOLVES TO 2
MLOGIC(DATAIN): %DO LOOP INDEX VARIABLE I IS NOW 3; LOOP WILL ITERATE AGAIN.
SYMBOLGEN: MACRO VARIABLE I RESOLVES TO 3
MLOGIC(DATAIN): %DO LOOP INDEX VARIABLE I IS NOW 4; LOOP WILL NOT ITERATE AGAIN.
MPRINT(DATAIN): VAR COMPANY SALES1 SALES2 SALES3 ;
MLOGIC(DATAIN): ENDING EXECUTION.
40 RUN;

NOTE: THE PROCEDURE PRINT PRINTED PAGE 1.
NOTE: THE PROCEDURE PRINT USED 0.13 CFU SECONDS AND 4600K.

NOTE: THE SAS SESSION USED 0.76 CFU SECONDS AND 4600K.
NOTE: SAS INSTITUTE INC., SAS CAMPUS DRIVE, CARY, NC USA 27513-2414
NESUG MACRO PAPER 1

MARCH
YEAR = 1994

OBS  COMPANY  SALES1  SALES2  SALES3
1   11     345    672    889
2   12     543    345    546
3   13     345    672    889
(...)
13   23     687    910    546
14   24     675    549    543
15   25     775    872    978

MLOGIC indicates the beginning and ending of execution of the macros PARMSIN and DATAIN. MPRINT displays the SAS code generated by the execution of the macro DATAIN: the TITLE3 statement with the correct value for month. The notes in the SAS log are the same as without the SAS system macro options, but now with SAS code related, the notes have more meaning.

SYMBOLGEN displays the resolution of the macro variable MONTH to the value MAR. MLOGIC traces the execution of the %IF by indicating the resolution of TRUE or FALSE for each test. MLOGIC traces the execution of the %DO loop indicating where we are in the loop by displaying the values of macro variables and indicating whether the loop will reiterate. Finally, we see the VAR statement.

Conclusion

At first glance, the SAS log looks confusing, but those extra lines are necessary to thoroughly trace the execution of the macro code. If you are confident the logic is correct, you may want to use NOMPRINT NOMLOGIC SYMBOLGEN to limit display to the results of the resolutions of the macro variables.