

Debugging SAS® code in a macro

Bruce Gilson, Federal Reserve Board, Washington, DC

ABSTRACT

Debugging SAS code contained in a macro can be frustrating because the SAS error messages refer only to the line in the SAS log where the macro was invoked. This can make it difficult to pinpoint the problem when the macro contains a large amount of SAS code.

Using a macro that contains one small DATA step, this paper shows how to use the MPRINT and MFILE options along with the fileref MPRINT to write just the SAS code generated by a macro to a file. The "de-macroified" SAS code can be easily executed and debugged.

DATA SET USED IN THIS PAPER

Data set ONE used in this paper has the following values.

Obs	X1	X2	X3
1	1	2	3
2	4	5	6

DATA STEP WITH AN ERROR

The following DATA step contains an error. Array XALL has 3 elements, but the second DO loop iterates from 1 to 4, causing the array subscript to be out of range when I = 4.

```
data two;
  set one;
  array xall x1-x3;
  do i=1 to 3;
    xall(i) = xall(i) +
  100; end;
  do i=1 to 4;
    xall(i) = xall(i) + 1000;
  end;
run;
```

The SAS log helpfully shows that an array reference at line 16 is the source of the error. We can review that line of code and fix the program.

Note that in this and subsequent examples, lines 1-8 in the SAS log contain the DATA step that creates data set ONE.

```
9          data two;
10         set one;
11         array xall x1-x3;
```

12

do i=1 to 3;

1

```

13             xall(i) = xall(i) + 100;
14         end;
15         do i=1 to 4;
16             xall(i) = xall(i) + 1000;
17         end;
18     run;

```

ERROR: Array subscript out of range at line 16 column 17.

x1=1101 x2=1102 x3=1103 _I_=. i=4 _ERROR_=1 _N_=1

NOTE: The SAS System stopped processing this step because of errors.

DATA STEP IN A MACRO WITH AN ERROR

Now, suppose the same DATA step was contained in a macro.

```

%macro mac1;
    data two;
        set one;
        array xall x1-x3;
        do i=1 to 3;
            xall(i) = xall(i) +
100; end;
        do i=1 to 4;
            xall(i) = xall(i) + 1000;
        end;
    run;
%mend mac1;
%mac1

```

Now, the SAS log shows that the error occurred at line 21, where the macro was invoked, rather than at the line of code that actually caused the error.

```

9         %macro mac1;
10             data two;
11                 set one;
12                 array xall x1-x3;
13                 do i=1 to 3;
14                     xall(i) = xall(i) + 100;
15                 end;
16                 do i=1 to 4;
17                     xall(i) = xall(i) + 1000;
18                 end;
19             run;

```



```

20          %mend mac1;
21          %mac1

```

ERROR: Array subscript out of range at line 21 column

```
137. x1=1101 x2=1102 x3=1103 _I=. i=4 _ERROR_=1 _N_=1
```

NOTE: The SAS System stopped processing this step because of errors

The macro contained only one small DATA step and two statements with array references, so debugging the program might not be too difficult. In real life, a macro could contain a very large DATA step or multiple steps or could be building the SAS code, making it very difficult to identify where the problem occurred.

DATA STEP IN A MACRO WITH AN ERROR: WRITING THE SAS CODE TO A FILE

The system option MPRINT writes the text (SAS statements) generated by a macro to the SAS log. If you specify the system options MPRINT and MFILE along with the fileref MPRINT, the SAS statements displayed by the MPRINT system option are also written to a file for easy use.

To use the MPRINT and MFILE options, enter the following statements before running a macro. The fileref MPRINT is required.

```

filename mprint 'external-file-for-SAS-statements';
options mprint mfile;
run;

```

The MPRINT option can generate a lot of text in the SAS log, so after writing the SAS code to a file you can turn off MPRINT and MFILE as follows.

```

filename mprint clear;
options nomprint nomfile;
run;

```

Returning to our example, here is the same macro as in the previous section, preceded by options MPRINT and MFILE.

```

filename mprint 'c:\mysasfile1.sas';
options mprint mfile;
run;

%macro mac1;
  data two;
    set one;
    array xall x1-x3;
    do i=1 to 3;
      xall(i) = xall(i) +
100; end;
    do i=1 to 4;
      xall(i) = xall(i) + 1000;
    end;

```



```

run;
%mend mac1;
%mac1

```

The file 'c:\mysasfile1.sas' contains the following SAS code, which is identical to the DATA step in macro MAC1 (except for the indentation).

```

data two;
set one;
array xall x1-x3;
do i=1 to 3;
xall(i) = xall(i) + 100;
end;
do i=1 to 4;
xall(i) = xall(i) +
1000; end;
run;

```

If we add the DATA step that generates data set ONE to the code in 'c:\mysasfile1.sas' and submit this code, the SAS log now helpfully shows that an array reference at line 16 is the source of the error. We can review that line of code and easily fix the program.

```

9      data two;
10     set one;
11     array xall x1-x3;
12     do i=1 to 3;
13     xall(i) = xall(i) + 100;
14     end;
15     do i=1 to 4;
16     xall(i) = xall(i) + 1000;
17     end;
18     run;

```

ERROR: Array subscript out of range at line 16 column 11.

x1=1101 x2=1102 x3=1103 _I_=. i=4 _ERROR_=1 _N_=1

NOTE: The SAS System stopped processing this step because of errors.

CONCLUSION

Specifying both the MPRINT and MFILE system options along with the fileref MPRINT allows you to write the SAS code generated by a macro to an external file. This makes the SAS code easy to execute for debugging purposes.

REFERENCES

SAS Institute Inc. (2012), "*Base SAS 9.3 Procedures Guide, Second Edition*," Cary, NC: SAS Institute Inc.
SAS Institute Inc. (2011), "*SAS 9.3 Language Reference by Name, Product, and Category*," Cary, NC: SAS Institute Inc.

SAS Institute Inc. (2011), "*SAS 9.3 Macro Language: Reference*," Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

The following people contributed extensively to the development of this paper: Heidi Markovitz and Donna Hill at the Federal Reserve Board. Their support is greatly appreciated.

CONTACT INFORMATION

For more information, contact the author, Bruce Gilson

By mail: Federal Reserve Board, Mail Stop N-122, Washington, DC 20551

By e-mail: bruce.gilson@frb.gov

By phone: 202-452-2494.

TRADEMARK INFORMATION

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 registered trademarks or trademarks of their respective companies.

