Conditional ENDSAS Statement
Erik W. Tilanus, KLM Royal Dutch Airlines, Amstelveen, the Netherlands

ABSTRACT
ENDSAS; is not a statement that can be executed conditionally. This paper discusses two methods to circumvent this limitation.

INTRODUCTION
In batch software or in semi-interactive software (foreground processing, outside the Display Manager) it happens regularly that there is no sense in continuing a program without the need of an ABEND. This happens for instance when you detect that an input data set is empty, or that a user specified some selection which did not lead to matching observations. Any subsequent analysis steps or data set updates are just a waste of time.

In such case the use of an ENDSAS; statement could be helpful in order to prevent the rest of the program from being executed. Unfortunately, ENDSAS; is not a DATA step statement and so it is not possible to include it in an IF statement to execute it conditionally.

In order to achieve a conditional execution of ENDSAS;, one can appeal to the macro language. Following are two versions of a macro that let you gracefully stop the execution of your SAS® program.

Don't use one of these techniques in an interactive SAS session, as it ends the session completely, not just the application you use in the session.

%STOPIT Version 1
The first version makes use of a global macro variable SWITCH. This one is selectively filled in the DATA-step. When the variable has received its correct value, the macro will generate the ENDSAS; statement:

%MACRO STOPIT;
  %GLOBAL SWITCH;
  %IF &SWITCH = STOP %THEN
      ENDSAS %STR();
  %END;
  ...
  IF ERROR THEN CALL SYMPUT('SWITCH','STOP');
  ...
RUN;
%STOPIT

%STOPIT Version 2
The second alternative makes use of the CALL EXECUTE routine. When the CALL EXECUTE is carried out, the DATA step will pause. The SAS System will first execute the macro which is mentioned as argument and then continues with the DATA step. Statements that have been generated in the macro are inserted into the program after the DATA step. They are not part of the DATA-step in which the CALL EXECUTE is executed, since that one is already compiled before the execution:

%MACRO STOPIT;
ENDSAS;
%MEND;

......
IF ERROR THEN CALL EXECUTE('%STOPIT');

Note: in the argument of CALL EXECUTE one works with single quotation marks to prevent the macro from being executed during the compilation of the step.

CONCLUSION
To stop the execution of a SAS program prematurely without indication of an error you can use the ENDSAS; statement. Since it is no DATA step statement, it can only be used conditionally by means of the macro language.

The paper did not discuss the ABORT statement as this is primarily to indicate an abnormal end of a program.

REFERENCES
For information about the ENDSAS statement refer to:

For an introduction to the SAS macro language refer to:

For details of the SAS macro facility refer to:

SAS is a registered trademark of SAS Institute Inc. in the USA and other countries.

Other brand and product names are registered trademarks of their respective companies.

The author can be reached at:
KLM Royal Dutch Airlines
dep. AMS/RO
P.O.box 7700
1117ZL Schiphol
the Netherlands
tel: +31 20 6499384
or: +31 2972 63936
fax: +31 20 6488086
or: +31 2972 61712