

Coping with SAS Error Messages

Stella Wang
Texas Department of Public Safety, Austin

Introduction

This tutorial presents several aspects of SAS* error messages that frequently confront SAS programmers. Most of the discussion will focus on the DATA step where the majority of the errors occur.

What Kind of Errors

Errors are classified as syntax errors, logic errors, data errors and input errors. Syntax errors can be missing or invalid punctuation, misspelled keywords or symbols that SAS does not recognize. Logic errors can be array subscript out of range, data set not in order by a particular variable or IF-THEN condition not correct. The logic errors are usually the hardest to debug. Input errors can be caused by a character in a numeric field or an input data line shorter than the expected length. Data errors can be the inconsistencies in variables (i.e. name, type, length, format informat and label).

SAS interfaces to VSAM allows us to manipulate VSAM data set (i.e. read, add, delete, modify and load new VSAM data set) through SAS program. SAS automatic variables, `_RBA_`, `_IORC_`, `_FDBK_` and `_RRN_` are set each time I/O takes place between the SAS and a VSAM data set. `_RBA_` contains the relative byte address. `_IORC_` contains the VSAM I/O return code. `_FDBK_` contains the VSAM feedback code and `_RRN_` contains the relative record number. There are two types of VSAM errors: physical and logical errors. A physical error is also called an I/O error. It occurs when VSAM can not access a file due to a hardware error. Logical errors can be SAS logical errors or VSAM logical errors. If SAS detects an error before invoking VSAM, it prints

error message and stops the DATA step. If VSAM detects an error, the interface will pass the values for SAS automatic variables (i.e. `_IORC_` and `_FDBK_`). The table defines VSAM feedback codes and error descriptions is available in SAS Guide to VSAM Processing. The common VSAM feedback codes are:

4	end of file
8	duplicate primary key
12	not in key sequence
16	record not found
20	record lockout
32	no record by RBA
36	key not in the range
88	lost pointer
96	key change not allowed
108	record too big
192	RRN not found

How SAS Detects Errors

In Version 5 the SAS Supervisor was responsible for scanning SAS statements, reading data, translating statements into machine codes, creating all SAS data sets, calling SAS procedures as requested, printing error messages and ending the job. Under Version 6, the SAS Supervisor determines to invoke either a DATA step or a procedure. The Data Step Processor is responsible for the actual DATA step processing after the invocation. Valid DATA step statements start with keywords except assignment and sum statements. When DATA step processor encounters unrecognizable words or symbols it flags error, sets the automatic variable `_error_` to 1 and prints error number and explanation. The DATA step processor is also responsible for creating the executable image and executing the machine codes. During the compilation, the syntax errors are

detected and the SAS automatic variable, `_error_` is stored in the logical Program Data Vector (PDV). The data values are set to missing before the execution of DATA step. The execution phase begins after the DATA step has been successfully compiled. The logic errors, input errors and data errors are detected during the execution phase.

Where to Find SAS Error Messages

Error messages, warning messages and notes appear in a temporary file called the SAS log. The amount of information SAS prints in the SAS log is controlled by the SAS system option, `MSGLEVEL`. By default, it prints information such as the JCL, CPU time, disks and other devices allocated to the job, notes, warning messages, error messages, the number of observations and variables in the data sets. Additional messages are printed if the `MSGLEVEL` is set to I. The SAS log can be customized by using SAS statements such as `PUT`, `LIST` and `ERRORS`. SAS errors underlined with explanations printed in the SAS log. An error number is usually assigned with each message. The real error most likely is the one being underlined. It should be noted that sometimes the real error actually can be located one or two lines preceding the underlined error or even in the previous DATA step. For example, the 'variable not defined' error message may be resulted from the omission of the variable from the `KEEP` statement in the previous DATA step.

How To Avoid the Common Errors

Familiarity with the DATA step statements is the best defense against errors. There are four types of data step statements: file handling, action, control and information statements. The common errors are: syntax errors involve all four types statements, logic errors involve

action statements, data errors and input errors involve file handling statements. The version 6 SAS is more tolerant with the misspelled keyword by issuing only a warning messages. `MERGE` and `UPDATE` can generate undesirable results due to inconsistent data attributes in the common variable from different data sets.

Some useful tips for reducing SAS errors are:

- Be familiar with your input data: `PROC FREQ` outputs frequencies which provide an overall picture of your input data. `PROC MEANS` prints useful statistics about numeric variables.
- Use `PROC CONTENTS` to find out if the variables in the data set have the correct attributes.
- Understand the impact of missing data.
- Check zero in mathematics operation.
- Find out the average record length if you are dealing with variable length records. It can help you to estimate the storage it needs.
- Record the number of observations from each step and pay special attentions to the drastic changes in numbers of observation.
- Allocate enough sort space.
- Use options like `STOPOVER`, `LINELEN` with `INFILE` statement.
- Print the contents of program data vector by using `PUT _ALL_`. It maybe necessary in certain case to print the intermediate data sets.
- Re-sort the character data on the host system if the original data was generated from a platform has a different collating sequence.
- Share your experience in handling errors with your peers.

***SAS is a registered trademark of SAS
Institute, Inc., Cary, NC**

REFERENCES

**DeFoor, J. (1991) Proceedings of the
1991 South-Central SAS Regional
Conference.**

**Howard, N. (1989) 'Effective Methods
of Testing Using the SAS Software'
Proceedings of the Fourteenth Annual
SAS Users Group International
Conference, Cary, NC: SAS Institute
Inc.**

**Rabb, M.G., Henderson, D.J. & Polzin,
J.A. (1992) 'The SAS Supervisor - A
Version 6 Update' Proceedings of the
South-Central Regional SAS Conference
'92.**

**SAS Guide to VSAM Processing:
Version 6, First Edition, Cary, NC: SAS
Institute Inc. (1992).**

**SAS Language: Reference, Version 6,
First Edition, Cary, NC: SAS Institute
Inc. (1990).**

**SAS Language and Procedures: Usage,
Version 6, First Edition, Cary, NC: SAS
Institute Inc. (1989).**