

Table 24: Managing Errors, Warnings and Critical Notes



Good Programming requires Good Debugging Skills

- ✓ Know the rules (syntax) and how to apply (meaningful names)
- ✓ Know your data type (numeric, character) and values (acceptable)
- ✓ Know what results to expect (no surprises)

Selected Errors, Warnings and Notes along with tips to resolve

Syntax and Semantic Error: Compile Time Errors	Possible Resolution
Error: Missing semicolon	Add semicolon (;).
Error: Misspelled/missing keywords or incomplete SAS statement	Correct the keyword spelling or add expected SAS statement.
Error: Unbalanced quotes	Locate and add a missing quote (').
Error: Variable Type Conflict	Ensure common BY variable attributes are the same when merging data sets.
Error: WHERE clause operator requires compatible variables	Ensure both operator and variable types match, ie. both are numeric or character.
Logic Error/Data Issues	Possible Resolution
No Message: Character Field Truncated	Specify an ATTRIB or LENGTH statement early in the DATA step to capture all character data values.
Note: Invalid Data Messages	Confirm data assumptions or adjust INPUT statement to read data correctly.
Note: Missing values or incorrect results	Apply conditional logic or use functions such as SUM() or MEAN() to exclude missing values.
Note: MERGE statement has more than one data set with repeats of BY values	Ensure BY variables uniquely separate records in each data set and is not a many-to-many merge. Maybe use Proc SQL for cartesian join.
Note: Numeric to Character or Character to Numeric conversions	Use PUT() to convert numeric to character and INPUT() to convert character to numeric type.
Non-Syntax/Run-Time Error: Execution Time Errors	Possible Resolution
Error: By Group Processing	Presort data sets before specifying the BY statement in DATA step and SAS procedures.
Note: Variable uninitialized and variable not found	Make sure variable exists, spelling is correct and is initialized if it is new. Make sure dot (.) exists if a format or informat. and array index exists if an array name.

Be careful of logic errors which may produce wrong results without generating any errors!



Table 25: DATA Step Debugging Techniques

- ✓ Purpose is to identify unexpected data.
- ✓ Capture condition, display data issue and other temporary variables in SAS Log or Abort program.

Statement	Brief Description
<pre>if (age < 0 or age > 50) then put 'Age is not correct: ' age 3.;</pre>	Displays user-specified message along with missing or data values that are out of range.
<pre>if (age < 0 or age > 50) then put age = ;</pre>	Displays 'AGE =' and value of age variable.
<pre>if (age < 0 or age > 50) then put _infile_;</pre>	Displays the last record accessed by an INPUT statement and all variables currently in the Program Data Vector.
<pre>if (age < 0 or age > 50) then put _all_;</pre>	Displays values in named format of all permanent and temporary variables in the Program Data Vector.
<pre>if agegrp='OLD' and age < 19 then _error_ = 1; else _error_ = 0;</pre>	Sets the <code>_ERROR_</code> flag to be 1 to force dump listing of the input buffer of data issue, else set to 0 to prevent display.
<pre>if (age < 0 or age > 50) then list;</pre>	Displays the last record accessed by an INPUT statement with a column ruler. This helps identify incorrect INPUT statement column specifications.
<pre>if agegrp='OLD' and age < 19 then error 'AGEGRP and AGE do not match ' AGEGRP= AGE=;</pre>	ERROR statement can be used not only to write messages and variable values to the log, but to cause the <code>_ERROR_</code> flag to be set to 1, thereby forcing a dump of the permanent Program Data Vector variables.
<pre>merge C (in=inc) D (in=ind); by patno; if AGE < 19 then put 'Age < 19' INC= IND=;</pre>	Use the IN= data set option to identify which data set(s) a particular observation came from during a MERGE, SET or UPDATE operation. Displays user specified message when condition is met.
<pre>options details errors=N notes; options syntaxcheck; * SAS Version 9.1;</pre>	System options to help debug programs. DETAILS – Displays additional information such as data set labels, and SAS data libraries. ERRORS= Maximum number of observations for which complete error messages are displayed. NOTES – Writes SAS Notes to Log.
Enhanced Editor	Enhanced editor can be used to change the color of keywords in SAS programs. Color changes to red by default if a keyword is not recognized.