FIND SAS® SYSTEM ERRORS AND MAKE CORRECTIONS QUICKER:  
AN ENHANCEMENT OF THE VMS™ EVE EDITOR  

Lewis Fountain, Sanofi Winthrop Inc., Malvern, PA

ABSTRACT

This paper discusses two EVE editor (VMS operating system) commands you can use to easily find SAS errors in LOG files, switch to the line in the SAS file which has the error, and make the correction. The EVE commands can be added to the base EVE editor in less than 1 hour.

One command searches LOG files for a variety of strings associated with SAS errors. Upon identifying the line of code that produced the error, you may position the cursor before the text that caused the error and then execute the second command, switching to the corresponding line in the SAS file that produced the error so that it may be corrected. You may then switch back to the LOG file at the point where you exited it and continue looking for more errors, and switch back to the SAS file to make corrections. The two EVE editor commands greatly speed up the time it takes to search for and correct errors.

INTRODUCTION

Locating SAS errors can involve time consuming paging or searching through LOG files. Using searches can limit you to just one type of error or involves a lot of typing. When you find errors, fixing the problems may involve time consuming shuffling back and forth between LOG and SAS files. The two procedures (ERF and SW), which can easily be added to a VMS EVE editor, can simplify the process.

DISCUSSION

The procedure ERF searches a file in a forward direction for the first instance of the following strings that are associated with a SAS error message: ‘ERR’, ‘IS UNINIT’, ‘NOTE: MERGE STATE’, ‘HANGING DO’, ‘A READ OR WRITE ERR’, ‘APPARENT SYMB’, ‘WARNING’, ‘FORMAT WAS TOO SMALL’, ‘WARNING’, ‘ERROR’, ‘OMITTED DUE TO MISSING ID V’. This list will find most of the more common errors, other strings can be easily added. This ERF procedure may be repeated until you see all errors in a file.

Execution of the SW procedure while in a LOG file will capture the text starting at the present cursor position to the end of the line (leading and trailing blanks are trimmed), then switch you to the corresponding SAS file and search for the captured text. Execution of the SW procedure while in a SAS file returns you to the LOG file at the same cursor position you were at last, or to the top of the file if it is your first entry into the LOG file. This procedure can be used to capture the text that may have caused an error, and switch you to the corresponding line in the SAS file so you can fix the problem. You may repeat the SW procedure any number of times. The ERF and SW commands can be associated with function keys (our site uses F8 and F9 respectively), further simplifying the process. Alternatively, the ERF and SW commands can be executed by entering ERF or SW into the command line and pressing return.

SAMPLE SESSION

A sample session might proceed as follows:

Type ER DEMOG1.LOG and press return, puts you in the EVE editor and brings the DEMOG1.LOG file into the editing screen.

Press the F8 key (ER) which takes you to the first error message. Review the code that precedes and find the error, such as misspelling GPLOT as GGPLLOT. Put your cursor somewhere on the line before the misspelling and press F9 (SW). Make sure the cursor is ahead of any line numbers or output caused by an MPRINT option, otherwise you pick up characters that you can not match in the SAS file. The current version of DEMOG1.SAS will be called into the editing screen with the cursor on the same line that the misspelling GGPLLOT is on. Correct the error. Press F9 (SW) to switch you back to the editing screen that will now hold the LOG file for DEMOG1 at the same point you switched out of it. Hit F8 (ER) until you come across another error. Find the line the error is on, put the cursor at an appropriate spot on the line and press F9 (SW) to switch back to the SAS file for DEMOG1, which you have already corrected once. Correct the error and switch back to the LOG file again if you want.

You may add the ERF and SW editing procedures to the file that contains your current EVE editor, or create a separate file with a different name that is a copy of the EVE editor with the two new commands added to it. You then access the new EVE editor with a user created logical we called ER. Our company chose the second approach, described further in the INSTALLATION section.

USAGE NOTES

If a search using ERF takes you to a line with an error message on it, the cursor positions you 5 lines below the error message so as to skip over other error messages that often occur on consecutive lines. This greatly speeds your progress through a LOG file.

If you use the F8 or F9 keys while the command line is present, a message 'No command given' will briefly appear. This message may be ignored.

The SW command will only work on SAS and LOG files, a warning message appears otherwise.

If the string you capture in the LOG file is not in the SAS file, you are switched to the top of the SAS file.

If you try to switch to a file without a matching LOG or SAS file a warning message will appear.

If you delete all versions of a SAS or LOG file after you have pulled a version into the editor, execution of the SW command would give you the message that the file does not exist. The version of the file you pulled into the editor would still be accessible using the command BUFFER filenname.filetype.

INSTALLATION

To create the new EVE editor you need the following 5 files:

1) The file that contains the ERF command (our name ERR1.TPU, approx. 100 lines including documentation).

2) The file that contains the SW command (our name SW14.TPU,
3) The GETSAME_MASTER.FILE which contains the names and locations of the 2 new procedures (two lines long). The file name is required.

4) The GETSAME_VERSION.DAT file which contains the version of EVE you are using (1 line long). The file name is required. The version of EVE you are using can be determined by looking at the EVE$VERSION.DAT file (on directory SYS$COMMON:[SYSHELP.EXAMPLES] at our site).

5) The EVEINIT.EVE file customizes the modified EVE editor. Our site sets up the keypad to act as if it was an EDT keypad, sets the tabs, sets the cursor, and assigns the F8 and F9 keys to the ERF and SW commands, respectively. This file is one you might want to examine and modify according to your site needs.

To create the new EVE editor you need to execute two logics:

1) Execute a logical I have called BUILD. (BUILD GETSAME <return>). This logical creates the new EVE editor using the files 1-4 listed above. Files 1-4 should exist on the same directory, and the BUILD logical should be run from this same directory.

   The BUILD logical looks like:

   Build :=edittpu/nodisplay -
   /section=eve$section -
   /command=sys$common:[syshelp.example]$eve$build/noinitialization -
   /output=user_disk:10:[roos55]$eve$eletpu/section

   Italized sections are site specific. At our site the files necessary to build the new EVE editor EVELEF.TPUSSECTION, are located on SYS$COMMON:[SYSHELP.EXAMPLES] (this directory may be a VMS convention). I create the new EVE editor on my root directory USER_DISK:10:[ROOS55]; check to see the process was OK, then move the new EVE editor to its final destination, the directory BIODISK1:[BIOSTAT.PRGRM.COMMANDS]. (BIODISK1 is another logical, pointing to a user disk.)

   Running the BUILD logical will create other files with the prefix of EVE, that are usually of no value.

2) Execute the following logical, we have called ERILEW (italized sections are site specific). This logical should be run from the directory that the new EVE editor EVELEF.TPUSSECTION and the EVEINIT.EVE file are located on, which for our site is the directory BIODISK1:[BIOSTAT.PRGRM.COMMANDS]. Example, ERILEW filename.filetype <return>.

   Eriew :=edittpu/-
   Initialization=bioldisk1:[biostat.prgrm.commands]eveinit -
   /section=bioldisk1:[biostat.prgrm.commands]evelefl

   Execution of this logical runs an initialization file (EVEINIT.EVE located on BIODISK1:[BIOSTAT.PRGRM.COMMANDS], file 5 above, that sets the EVE keypad to act like an EDT keypad and adds any site specific customized keys, such as assigning ERF and SW to F8 and F9, respectively. After entering a file using the ERILEW logical, it is important to save the changes you have made to the EVE editor by running the initialization file. To do this type: SAVE EXTENDED EVE EVELEF.TPUSSECTION in the EVE command area and press return, where 'EVELEF' is the name of your new EVE editor. Quit out of this file. You need to only run the ERILEW logical once, since the changes made by running the initialization file were saved to the new EVE editor.

3) Test the new EVE editor by entering some test files using the following ERFOU logical (italized sections are site specific).