

Paper 038-30

PROC IMPORT with a Twist

Gary McQuown
Data and Analytic Solutions, Inc

Abstract

PROC IMPORT is a very useful tool for converting “text” files to SAS®¹ data sets. However, PROC IMPORT may not properly assign attributes such as the variables length and format. Writing Data Step code with infile and input statements provides greater control, but it increases the effort required as well as the probability of human error. Fortunately there are three simple tips or tricks that solves these dilemmas. This presentation should be useful for anyone who uses PROC IMPORT.

Introduction

Importing CSV, ASCII and Comma Delimited files are a common and tedious task. The task is even more tedious if there are a large number of variables. Using PROC IMPORT either through the Import Wizard or by submitting the code makes it much easier. However SAS makes a number of assumptions about how it should treat your data. This is because SAS normally will only read the first twenty lines of the input file when deciding how to format the variables. The result of SAS's best guess is often truncated variables or values that are not formatted correctly. Even if the data has been read in correctly, the given length of a variable may still be inappropriate for your intended use.

This paper suggests three possible solutions to this dilemma. The first is a combination of PROC IMPORT with a minimal amount of manual manipulation to insure that you get what you want. The second is to change the default setting that tells SAS how many observations to review in the data. The third is to use the **External File Interface (EFI)** tool.

Regardless of how PROC IMPORT is invoked (Import Wizard or written), a version of the following code is executed. When invoked, PROC IMPORT “writes code” to read the file with a data step and infile statement. Although the code written is not shown in the editor window, a log of the submitted code will be in the log window.

Example of PROC IMPORT Code

```
PROC IMPORT OUT= WORK.cc_1 DATAFILE= "C:\DAS\SUGI05_CC_1.csv"  
  DBMS=CSV REPLACE; GETNAMES=YES; DATAROW=2;  
RUN;
```

Solution 1: PROC IMPORT with Manipulation

This solution is more difficult, but certainly easier than typing hundreds of variable name and formats in by hand and safer than having SAS assume control over you data input. This method involves the following:

1. Running PROC IMPORT on the file or a small subset of the file.
2. If possible, select the option to use the first line as variable names
3. Copy the SAS log to the Program Editor
4. Delete the non-SAS code.. line number, titles, etc.
5. Modify the informats as required
6. Modify the formats as required
7. Modify the lengths as required
8. Run the new code

¹ 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.

When you run the PROC IMPORT, SAS will create a beautiful input program in the log. While the informat, length and formats may be incorrect, they should be close to what you need and the variable names will be correctly spelled and every variable will be included. By modifying this text and submitting it as your program, you can save a considerable amount of time and effort.

Example of SAS Code in the SAS Log

In the original data, the length of all the text fields is incorrect because SAS makes an educated guess based upon the first observations. Therefore the informat and format of q_1 needs to be changed from \$5 to \$50. This will tell SAS to read the values in and to store the value as a character with a length of 50. Although not “wrong” in this example, the format for the numeric values are not ideal.

```

457 data WORK.CC_2 ;
458 %let _EFIERR_ = 0; /* set the ERROR detection macro variable */
459 infile 'C:\Documents and Settings\DASC\Desktop\SUGI05_CC_1.csv' delimiter = ','
MISSOVER
459! DSD lrecl=32767 firstobs=2 ;
460 informat q_1 $5. ;
461 informat q2_6 best32. ;
462 informat q7_9 $5. ;
463 informat q8_1 best32. ;
464 informat q8_3 $5. ;
.. more code ..
475 format q_1 $5. ;
476 format q2_6 best12. ;
477 format q7_9 $5. ;
478 format q8_1 best12. ;
479 format q8_3 $5. ;
489 format check_77 $5. ;
.. more code ..
490 input
491 q_1 $
492 q2_6
493 q7_9 $
494 q8_1
495 q8_3 $
.. more code ..
506 ;
507 if _ERROR_ then call symput('_EFIERR_',1); /* set ERROR detection macro variable */
508 run;

```

A quick review of the following code shows that the informat and formats have been modified to more appropriate values and the obs numbers were removed. You can also change the name of the variables at this time. While the ‘copy / modify / past’ process takes time, it eliminates the need to type variable names, options and various format and informat statements. This can be a significant time saving device if a large number of variables are used and it virtually eliminates the possibility of misspelling variable names.

Example of Modified Code

```

data WORK.CC_1 ;
%let _EFIERR_ = 0; /* set the ERROR detection macro variable */
infile 'C:\DAS\SUGI05_CC_1.csv' delimiter = ',' MISSOVER DSD lrecl=32767 firstobs=2 ;
informat q_1 $50. ;
informat q2_6 8.2 ;
informat q7_9 $25. ;
informat q8_1 8.2. ;
informat q8_3 $30. ;
..... more variables .....
format q_1 $50 ;
format q2_6 8.2 ;
format q7_9 $25 ;
format q8_1 8.2 ;

```

```

format q8_3 $30 ;
..... more variables .....

input
  q_1 $
  q2_6
  q7_9 $
  q8_1
  q8_3 $
;
if _ERROR_ then call symput('_EFIERR_',1); /* set ERROR detection macro variable */
run;

```

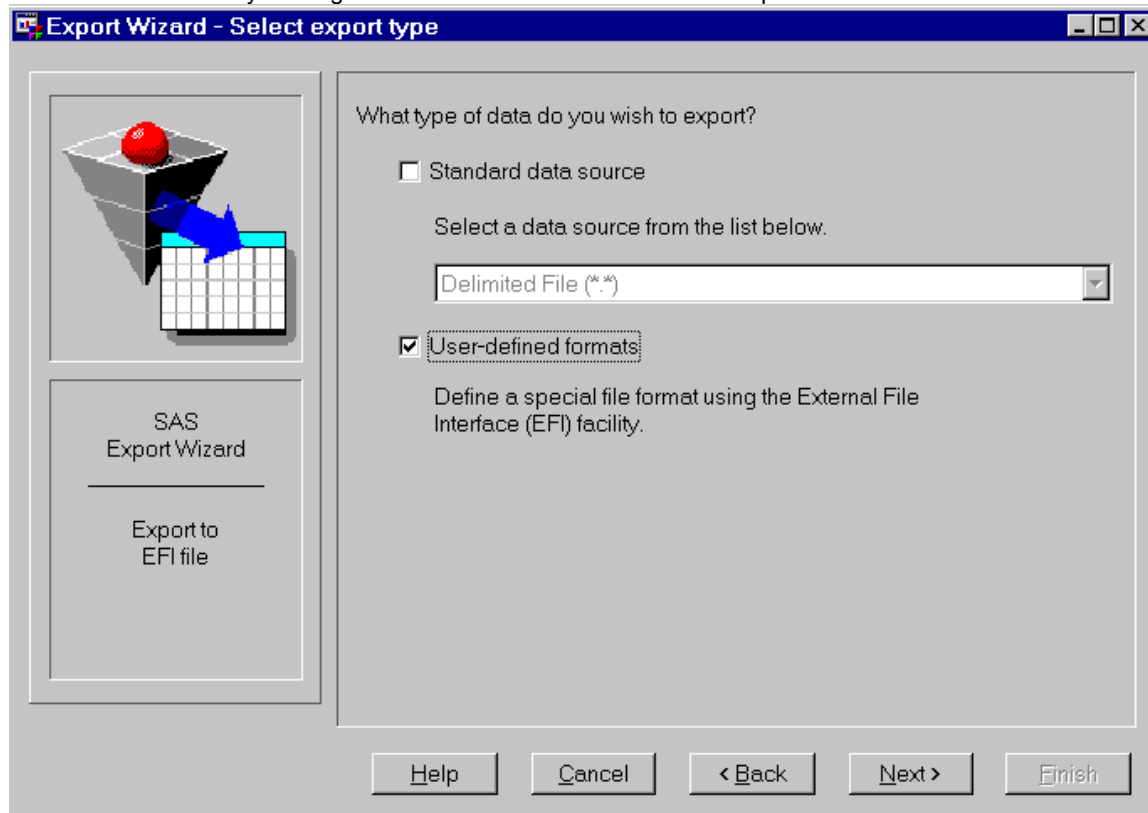
Solution 2: Change Default Setting

Changing the default setting can only be done for EFI in Version 7, Release 8 and Release 8.1. Beginning in Release 8.2 changing the default setting is applicable to the Import Wizard, PROC IMPORT and EFI. Follow the steps below to change the default behavior (<http://support.sas.com/faq/018/FAQ01819.html>).

- 1) Type regedit on the command line (white box with a check mark)
- 2) When the Registry Editor window opens, double click on the PRODUCTS icon
- 3) Double click on the BASE icon
- 4) Double click on the EFI icon
- 5) In the window on the right the Contents of EFI will be populated with EFI options
- 6) Double click on GuessingRows
- 7) When the new window opens with the old value of 20, delete it, enter the new value, and click on OK
- 8) Close the Registry Editor window
- 9) Invoke the Import Wizard, PROC IMPORT or EFI to use the new GuessingRows value

Solution 3: External File Interface (EFI)

The final solution is to use the External File Interface (EFI) tool. This tool may be invoked by typing EFI in the command line or by clicking "User Defined Formats" in the initial Import Wizard screen.



Although the EFI tool may be the easiest for non SAS programmers or novices to use, it does not record the code for later use or documentation. I also find it takes longer to use this tool than the cut and paste method in solution 1. A detailed explanation of how to use the EFI tool can be found online at <http://ftp.sas.com/techsup/download/sample/datastep/efi1.html>.

References:

SAS Institute Inc., "FAQ #1819", <http://support.sas.com/faq/018/FAQ01819.html>

Contact Information:

Your comments and questions are valued and encouraged. Contact the author at:

Author Name	Gary McQuown
Company	Data and Analytic Solutions, Inc.
Address	3127 Flintlock Road
City state ZIP	Fairfax, VA 22030
Work Phone:	703.628.5681
Email:	mcquown@DASconsultants.com
Web:	www.DASconsultants.com

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.