

# Tips for Reading and Writing Delimited Files

Michael G. Sadof, MGS Associates, Inc.

## ABSTRACT

While SAS® software gives us several options to seamlessly convert Excel spreadsheets directly into SAS data files our users sometimes request an intermediate comma delimited file for loading into a database. This occurs in part to conserve disk space while transferring between platforms. This paper will explore several methods to read or import comma delimited files and even have SAS generate the code for you. It will also remind you of the \$QUOTE format which can assist in writing out delimited files.

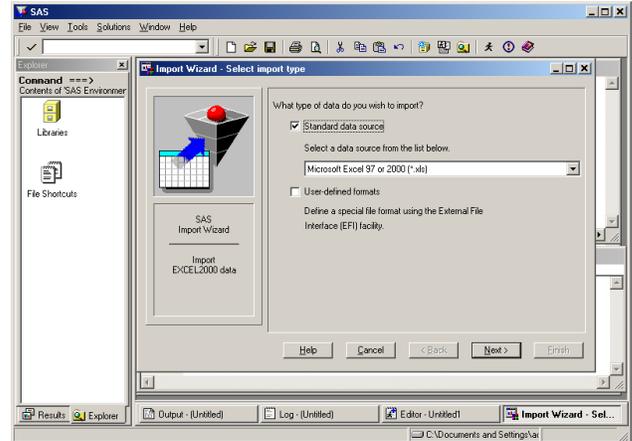
## INTRODUCTION

Version 8 of SAS software has an enhanced import/export function which allows you to read and write external files from other formats. The Import/Export wizard makes it very easy to bring in other standard type files such as: Microsoft Excel and Microsoft Access, Lotus, dBase, and various types of delimited files. However if more control is required you can use PROC IMPORT/PROC EXPORT to create and submit code directly or write detailed data step PUT statements for even more control. Variable record length delimited files are commonly used and can save space when used as an intermediate file for transferring to another system or platform.

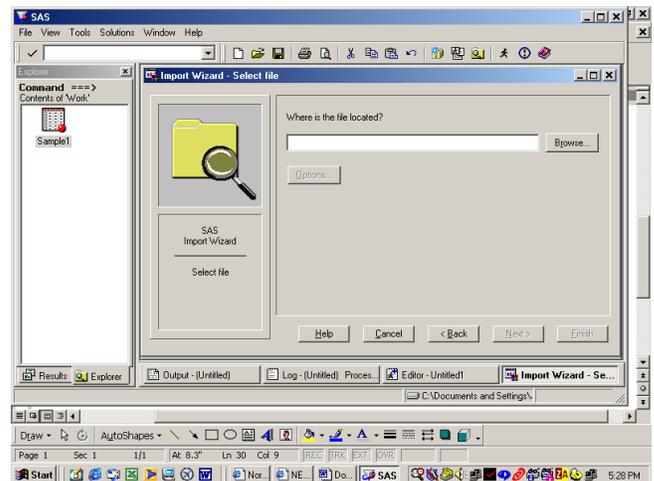
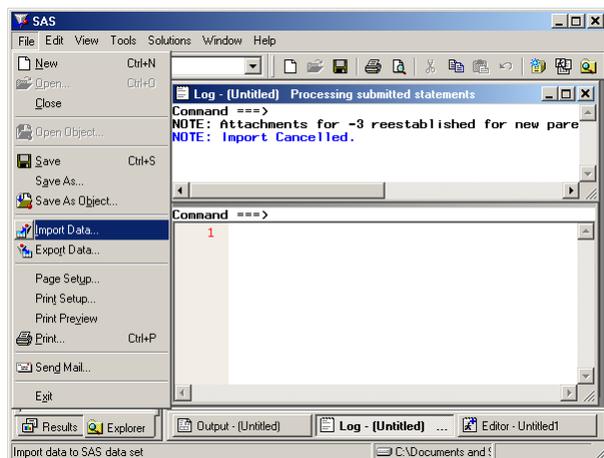
## IMPORT WIZARD

The import wizard is quite straight forward and follows all the standard conventions. You may access the wizard through the file menu in a SAS display manager session. It is shown as 'Import Data' on that menu.

Just select 'Import Data' and the next screen displayed will ask what type of data file you are trying to import.



Choose the appropriate file type from the list box. In the first example we will choose "Microsoft Excel 97 or 2000 (\*.xls)" to indicate a standard spreadsheet file. Your spreadsheet should contain column headings which will become the variable names. After selecting the file type and clicking next you will be asked for the file location.



Either type in the file name or navigate to the appropriate directory and highlight the appropriate file.

At this point you may specify options which locate a particular ranges of cells or use the entire worksheet. Choosing next brings you the next screen which ask for the name and location of the

particular SAS library in which you want to store your new data set. The library name will have to have been previously assigned or you may use the 'WORK' library. Type in the appropriate member name. Selecting next will then ask you if and where you want to store the code. This is not necessary if you do not want to store or view the code that the wizard will create. Selecting finish will then import the data to a SAS file that will be available to your display manager session. You will note that SAS will use the first non-blank row as the variable name. If we choose to store the code you may then load this code back into the SAS editor and view the syntax:

```
PROC IMPORT OUT= WORK.SAMPLE3
  DATAFILE= "...\\SAMPLE1.xls"
  DBMS=EXCEL2000 REPLACE;
  GETNAMES=YES;
RUN;
```

## EXPORT WIZARD

The Export procedure works similarly in reverse. You must first choose the SAS file you want to export and then identify what type of file you want to create. In this case we will create an Excel file. The wizard then asks for the name of the output file and you can type it in or navigate to the directory where it is to be placed.

## COMMA DELIMITED FILES

The IMPORT/EXPORT wizard will permit you to create standard comma delimited files as well. It is common practice to use these files when loading an external data base such as Oracle or Sybase. Comma delimited files take up less space than do columnar fixed length record files. To create comma delimited files you may also use the data step and identify each of the variables to be output. The following code sample shows how to create a comma delimited file with the data step.

```
data _null_;
set sample3;
file "...\\SAMPLE3.txt" dlm=',';
put name : $quote12.
    ssn : ssn11.
    age : 5.
    acct_num : $quote12. ;
run;
```

In version 8 you may use the modified list output format (the colon character follows the variable name) to create a delimited file without any spaces but with quote surrounding the text values. In this manner you can save disk space when transferring files. Please note the use of the \$QUOTE format as well as the DLM option on the file statement.

The \$QUOTE format will surround the output with double quotes (") so that embedded blanks or other characters will be clearly part of the text string. If you expect commas within the text string such as "Lastname, Firstname" you should choose another delimiter such as the pipe symbol (DLM="|") which is not likely to be found in the data stream. Notice that formatted numbers such as SSN11. do not contain quotes. Sample output follows:

```
"SMITH, Mike" | 123-45-6789 | 18 | "AB0403"
"JONES, Eve" | 123-45-6791 | 32 | "ZV0506"
"BOND, James" | 123-45-6793 | 47 | "GF9594"
```

## CONCLUSION

SAS version 8 makes it very easy to move between standard external files such as spreadsheets or text files and SAS data files. The use of the IMPORT/EXPORT wizard enables even the novice user to translate files without coding tedious input statements. The use of the DLM option on the file statement and the \$QUOTE format can assist tremendously when creating standard delimited files for others to use.

SAS is a registered trademark or a trademark of the SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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

Please feel free to contact me with question or suggestions.

**Michael G. Sadof**  
**MGS Associates, Inc.**  
Bedford, NH  
[mgs@mgsnet.net](mailto:mgs@mgsnet.net)

