A Program for Fixing SAS Transport Files on Personal Computers

WUSS, October 1994

Glenn H. Itano
Pomoxis Systems
Richmond, California

Introduction

Moving SAS datasets between different operating systems or between different versions of SAS is usually a fairly easy procedure, well documented in SAS Technical Report P-195, *Transporting SAS Files between Host Systems*. However, sometimes there are problems reading the transported datasets. In my experience, this is most often because the data have somehow inherited some undesirable characteristics. The SAS program presented here can be used in some of these cases to read the transport file and write a new file in the correct format.

SAS Transport Files

SAS datasets are written in transport file format before being copied from one operating system to another. Chapter 2 of SAS Technical Report P-195 gives very thorough instructions for creating transport files.

SAS transport files to be used on personal computers (PCs) are written as 80 byte records and should have no explicit line termination characters. On some operating systems that support the SAS System, records in standard text files are terminated with a specific character or characters. Some of these are shown in the table below. Mnemonic CR stands for carriage return and LF stands for linefeed.

Operating System	Line Termination Character(s) (HEX)	Mnemonic
DOS/Win 3.1	0D0A	CRLF
OS/2	0D0A	CRLF
VMS	0A	LF
Unix	0D	CR

When using PROC COPY or PROC CIMPORT to read the transport file, the SAS System will tell you that it is having problems reading interpreting the data. The message might not be all that informative. For example, you might see a message like this in your SAS log after running PROC COPY.

```
ERROR: Member or Library unavailable for use on file
SASCAT.&&&&&&&&.DATA.
```

When I encounter an error message like this, I run the SAS program described in this paper to create a copy of the file that should be more appropriate for importing.

The SAS Program

The SAS DATA step is one of the most versatile tools for reading and writing files in almost any format. The SAS program that I use to reformat the transport files is a SAS macro that executes one DATA step to try to determine what the offending line termination characters are. A second DATA step is executed to write the corrected file.

The macro parameters are:

_OLDFILE	the name of the original, apparently flawed transport file
_NEWFILE	the name of the corrected transport file. _OLDFILE and _NEWFILE should not point to the same file.

The first part of the macro is shown below. It reads enough data from the original file to construct two complete 80 byte records and up to two line termination characters per record. If various combinations of line termination characters are encountered, the program makes some assumptions about the record format and tries to write the new file.

```
%macro portfix(_oldfile,_newfile);
 %let _lrecl=0;
 data _null_;
    infile "&_OLDFILE" recfm=f lrecl=164;
    input @81 crorlf1 $char1.   @81 crlf1 $char2.
          @162 crorlf2 $char1.  @163 crlf2 $char2.;
    if( crlf1=crlf2='0d0a'x )then do;    /* PC */
       put "Note: file has CRLF record terminator.";
       call symput('_lrecl','82');
       stop;
       end;
    if( crorlf1=crorlf2='0a'x )then do;  /* Unix */
       put "Note: file has LF record terminator.";
       call symput('_lrecl','81');
       stop;
       end;
    if( crorlf1=crorlf2='0d'x )then do;  /* VMS */
       put "Note: file has CR record terminator.";
       call symput('_lrecl','81');
       stop;
       end;
    put "Note: I can't find anything wrong with "
        "&_OLDFILE.!" / "      Quitting now.";
    stop;
 run;
```

The second DATA step is executed if the first DATA step has encountered a known line termination character pattern. It simply reads data as either 81 or 82 byte records and writes them to the new file as 80 byte records, stripping off the line termination characters.

```
%if &_LRECL ne 0 %then %do;
 data _null_;
    infile "&_OLDFILE" recfm=f
                       lrecl=&_LRECL
                       missover;
    input text $char&_LRECL..;
    file   "&_NEWFILE" recfm=f
                       lrecl=80;
    put    text $char80.;
 run;
%end;
%mend portfix;
```

Usage

Compile the macro by either explicitly %INCLUDEing the code or by adding it to your macro autocall library.

Running the macro is simple. Just pass the name of the file that has to be converted and the name of the file that you want to create. For example, if the macro is in file "PORTFIX.MAC" and you want to read file "ALLDATA.XPT" and create file "NEWDATA.XPT", submit the following code:

```
%include "portfix.mac";
%portfix(alldata.xpt,newdata.xpt)
```

After running this macro and creating the new transport file, use PROC CIMPORT or PROC COPY to import the data. Be sure to use the LIBNAME or FILENAME statement to point to the new transport file instead of the original file. Consult SAS Technical Report P-195 for details on how to import your data.

Limitations

Be sure you have adequate disk space available to store two copies of the transport file and the imported SAS datasets.

This program is intended for use on a PC running a version of SAS from 6.04 or later. There are many versions of SAS on many platforms where this program has never been tested. Thus, there is no guarantee that this program will work in your situation. Use it with some caution. It will probably work on other SAS platforms, in particular the various flavors of Unix. It will probably not work on VMS, where probems encountered while reading SAS transport files are often caused by the file *not* having the correct embedded carriage return carriage control or because the file was defined with records with the wrong record length.

It is very possible that this program might make a mistake and interpret real data in the original file as undesirable line termination characters. This possibility is decreased somewhat by the program checking to see if the first *two* records have the same line termination characters. However, there is no guarantee that this strategy will always work correctly. For this reason, it is important that the program not delete or write over the original file. Keep this in mind if you want to modify the macro.

Distribution

If you would like to receive a copy of this program by electronic mail, send a message

To: itano@netcom.com
Subj: WUSS 94 PAPER REQUEST

If you have any questions, comments, or suggestions, the author can be contacted at

Mail: Glenn H. Itano
Pomoxis Systems
878 37th Street
Richmond, CA 94805

Email: itano@netcom.com

References

SAS Institute Inc., SAS Technical Report P-195, *Transporting SAS Files between Host Systems*, Cary, NC: SAS Institute Inc., 1989. 181 pp.