

Converting SAS Data Sets with Label Information to ASCII files

Glenn Millard

Stephen F. Austin State University
Nacogdoches, Texas

Abstract

Occasionally the need arises to convert a SAS data set to an ASCII file. In my case I needed to convert SAS files for use with SPSS. It might even be a last resort for moving SAS files between platforms if nothing else works. This particular procedure is generic in nature and is designed to extract all variables with their variable and value formats from the catalog files. In most cases, you need not know anything about the variables in the files. This procedure will not be what everyone might need, but it can be easily adjusted to handle most any data set.

The idea of trying to come up with a SAS data set to ASCII file conversion program that is generic enough to work with most data sets proved more difficult than it might seem on the surface. This is because of questions about precision and the way SAS stores numeric data. Normally SAS will store a variable as an 8 byte real number. This allows for extremely large numbers as well as a great deal of precision when you are using numbers with decimals. There is one real way to tell just how big or small a given value may be in a numeric variable. Given no further information, SAS will print a number like this using the BEST12 format. This will cover most cases, it is a variation of this I decided to use. The reason for the variation was due to SPSS. SPSS wants to know just how many decimals of precision a variable needs, and if you do not tell it, SPSS truncates the number. The BEST12 format did not work well with this so I use a 12.3 format in these cases.

To get information on a data set the PROC CONTENTS procedure is run to create a data set of information about the SAS data set to be extracted. This information is then sorted to put the variable names in the same order as they appear in the original data set. If a format or an informat exists for any variable, that information is used to determine how the variable is written to the ASCII file. Character variables are written out at whatever their maximum length is. Otherwise the previously mentioned 12.3 format is used to output the variable.

Two files are created with the above information. The first is a SAS command file that will build the exported ASCII file. The second is an SPSS command file that will create an SPSS system file from the ASCII file. If you are not moving the data to SPSS you will not need this file. I create it in either case because the information in the SPSS command file is very informative as to the columns the variables are in after export.

Since my original reason for creating this program was for moving a SAS data set to SPSS I decided to go two steps further. This data set had variable label and value label information with it and I wanted to move this information as well. I therefore extract the variable label information from the PROC CONTENTS that was run earlier. If a Catalog exists that contains value label information I run a PROC FORMAT to extract this to a data set. All this information is then added to the SPSS command file being built earlier. The SPSS command file may need to be edited before it is run. This is because the name associated with a format in the Catalog file may not have the same name as the variable it is being used with.

A macro that starts a SAS window runs at the start of the program and prompts the user for information like the libname and filename of the data set to be extracted and name of the command files that will be built to export the data set. If there is not a SAS Catalog present with value label information then leave that field blank. The information on the SPSS files you are asked to enter should be filled in, even if you are not exporting the file into SPSS. This is just to make sure the program runs properly so that the SAS command file will be created.

The only time that you should have problems with this program is when you extracting a data set that has a variable with value greater than 99,999,999 or if the variable uses more than 3 decimals of precision. To cure this you could edit the created SAS command file to export your data to expand the output widths. This could be the hard way because it moves the column locations of all the other variables. The best way to cure this would be to run a PROC DATASETS and insert a format for the variable into the data set. This program will then use this format for exporting the variable. If you need to export comma delimited files from your data set this program should could also be easily modified to create these types of files.

This program was written on a VMS system running SAS 6.07 but it should run on most machines using any version of SAS. It should prove useful to anyone that needs to convert a data set to a flat file.

```

/*****
/*
/*      SAS Data Set to ACSII conversion program
/*
/*      NAME: SASEXTRACT
/*      Summary: This program creates a SAS command that will extract
/*                a SAS data set to a flat file. It also creates a SPSS command
/*                file that will create an SPSS system file from the flat file.
/*                Variable labels are exported to the SPSS command file if they
/*                exist. Value labels that are in a SAS Catalog are also exported
/*                if they exist. The program uses a data entry window to input
/*                the filenames the program needs.
/*
*****/

```

```
/* Macro for the data entry window */
```

```
%macro showit;
%global a b c d e f g;
%window libn
#1 @15 'SAS DATA SET to ASCII conversion program'
#3 @5 'This program creates the SAS command file necessary to extract'
#4 @5 'a data set to an ASCII file. An SPSS command file to read the'
#5 @5 'ASCII file is also created so an SPSS system can be created'
#6 @5 'from the ASCII file. Any labels that may exist will also be'
#7 @5 'put in this SPSS command file. Please enter the following file'
#8 @5 'information, after entering the information will clear.'
#9 @5 'At this point hit RETURN through all the fields to close the'
#10 @5 'screen and start the program.'
#12 @10 'Enter Libname where the data set is: '
#12 @47 lib 20 a=rev_video
#13 @10 'Enter Data Set name: '
#13 @31 set 20 a=rev_video
#14 @10 'Enter filename for the SAS command file:'
#14 @51 sasc 20 a=rev_video
#15 @10 'Enter filename for exported ASCII data:'
#15 @50 sasex 20 a=rev_video
#16 @10 'If you have a SAS catalog with format information.'
#17 @10 'Enter CATALOG library name (leave blank if not):'
#17 @59 clib 20 a=rev_video
#18 @10 'Enter filename for SPSS command file:'
#18 @48 spsc 20 a=rev_video
#19 @10 'Enter name of SPSS system file you wish to'
#20 @10 'create (Leave blank if you do not need one.):'
#20 @56 spsd 20 a=rev_video;

%display libn;
%mend;
```

```
/* Macro data step. This data step is run if a SAS Catalog exists.
```

```
%macro exformat;
proc format library=&lib cntlout=exform;
run;
proc sort data=exform;
by fmtname;
options nodate nonumber pagesize=32000 linesize=78;
data _null_;
length id1 $50 id2 $50 label1 $50;
retain st 1. len;
file "&spsc" noprint mod;
if st = 1 then put 'VALUE LABELS';
set exform;
by fmtname;
st=st+1;
if first.fmtname then do;
put ' ' fmtname;
```

```

len=1;
end;
label1=trim(label);
id1 = start || " " || label1 ;
id2 = trim(id1);
idl=length(id2);
len=len+idl+3;
if len > 80 then do;
put;
len=1;
end;
put ' ' id2 "" @;
if last.fmtname then put '  /';
run;
%mend;

/* This macro tests to see if a library name was given to show a SAS */
/* Catalog exists. If it exists, the previous macro is processed and */
/* run as a data step. */

%macro analyze(h);
%if &&clib ne '' %then %do;
  &&h;
%end;
%mend;

/* Display the macro window */

%showit;

/* Set system options that will allow output files to format properly */

options nodate nonumber pagesize=32000;

/* Run PROC CONTENTS to create a data set that describes the SAS */
/* data set to be exported. The cases are then sorted to put the */
/* variables back in the same order they are in the original file. */

proc contents data=&lib.&set memtype=all out=&lib..describe noprint;
run;
proc sort data=&lib..describe;
by varnum;
run;

/* Create the SAS command file from the Proc Contents information */

data _null_;
retain st 1. en 0. ;
file "out.sas" noprint;
if st = 1 then do;
put "options missing ' '";
put 'data _null_';
put "set &lib.&set.,";

```

```

put 'put';
end;
set &lib..describe end=last;
if memname = upcase("&set");
put '  'name'' @;

/* Process numeric variables with unknown formats */

if type = 1 and formatl = 0 and informat='' then do;
  en=12+st;
  put '12.3';
  st=en+1;
end;

/* Process numeric variables with format information */

if type = 1 and formatl > 0 then do;
  en=formatl-1+st;
  st=en+1;
  if formatd = 0 then do;
    put formatl +(-1) ' ';
  end;
else do;
  form=formatl+(formatd/10);
  put form;
end;
end;
/* Process numeric variables with informat information */

if type = 1 and formatl = 0 and informat ne '' then do;
  en=12+st;
  st=en+1;
  if informl = 0 then do;
    put '12.';
  end;
else do;
  form=12+(informl/10);
  put form;
end;
end;

/* Process character variables */

if type = 2 then do;
  put '$' @;
  en=(length-1)+st;
  put st +(-1) ' ' en;
  st=en+1;
end;

if last=1 then do;
  put ';;';
  put 'run;';

```

```

/* Output total output record length to temporary file */

file 'length.out' noprint;
put en;
end;
run;

/* Read temporary file created above and add FILE statement with */
/* LRECL option to the SAS command file being created. */

data _null_;
infile 'length.out';
input en;
infile 'outs.sas' end=last1;
file "&sasc" noprint;
do i=1 to 3;
input;
put _infile_;
end;
put "file ""&sasex"" noprint lrecl=" en ",";
do until (last1=1);
input;
put _infile_;
end;

/* Create the SPSS command file */

options nodate nonumber pagesize=32000;
data _null_;
retain st 1. en 0.;
file "&spsc" noprint old;
if st = 1 then do;
put "DATA LIST FILE=&sasex /";
end;
set &lib..describe end=last;
if memname = upcase("&set");
put ' ' name ' ' @;

/* Process numeric variables with unknown formats. */

if type = 1 and formatl = 0 and informat='' then do;
en=12+st;
put st +(-1) '-' en ' (3)';
st=en+1;
end;

/* Process numeric variables with format information. */

if type = 1 and formatl > 0 then do;
en=formatl-1+st;
if formatd = 0 then do;
put st +(-1) '-' en;
end;
else do;

```

```

    put st +(-1) '-' en '(' formatd +(-1) ');
end;
st=en+1;
end;

/* Process numeric variables with informat information. */

if type = 1 and formatl = 0 and informat ne '' then do;
    en=12+st;
    if informl = 0 then do;
        put st +(-1) '-' en;
    end;
else do;
    put st +(-1) '-' en '(' informl +(-1) ');
end;
st=en+1;
end;

/* Process character variables. */

if type = 2 then do;
    put '$' @;
    en=(length-1)+st;
    put st +(-1) '-' en;
    st=en+1;
end;
run;

/* Process macro to see if a SAS catalog exists. */
/* If it exists a data set is run to export value label information. */

%analyze(%exformat);
run;

/* Export variable label information to SPSS command file. */

options nodate nonumber pagesize=32000;
data _null_;
retain cnt 0;
length sfile $20;
file "&spsc" noprint mod;
set &lib..describe end=last2;
cnt=cnt+1;
if cnt=1 then put "VARIABLE LABELS";
if last2=1 then do;
    sfile="&spsd";
    if sfile ne '' then
        put "SAVE OUTFILE=&spsd";
    put 'FINISH';
end;
if memname = upcase("&set") and label ne '';
put " " name " " label "";

run;

```

Sample program output:
SAS command file to export a data set:

```
options missing ' ';
data _null_;
set gsas.extract;
file "extracted.dat" noprint lrecl=86 ;
put
  NAME $ 1-20
  ADDR $ 21-35
  STATE $ 36-37
  ZIP $ 38-42
  VALUE1 $ 43-43
  VALUE2 1.
  VALUE3 3.
  VALUE4 12.3
  VALUE5 6.1
  VALUE6 10.
  VALUE7 5.3
  VALUE8 5.3
;
run;
```

SPSS command file to create an SPSS system file from the exported file:

```
DATA LIST FILE=extracted.dat /
  NAME $ 1-20
  ADDR $ 21-35
  STATE $ 36-37
  ZIP $ 38-42
  VALUE1 $ 43-43
  VALUE2 44-44
  VALUE3 45-47
  VALUE4 48-60 (3)
  VALUE5 61-66 (1)
  VALUE6 67-76
  VALUE7 77-81 (3)
  VALUE8 82-86 (3)
VALUE LABELS
RESPOND
1          'yes' 2          'no' /
```

```

TTYPE
1 'pre ' 2 'post ' 3 'unknown '
4 'other ' 5 'other ' /
VARIABLE LABELS
  ADDR 'ADDRESS '
  VALUE1 'TYPE '
  VALUE2 'GROUP '
  VALUE3 'COUNT '
  VALUE4 'TOTAL '
SAVE OUTFILE=new.spss
FINISH

```

Important References:

SAS Language Reference, Version 6, First Edition.
 SAS Procedures Guide, Version 6, Third Edition.
 SAS Guide to Macro Processing, Version 6, Second Edition.