A program for exporting SAS datasets to Argus

Johan Heldal
Statistics Norway

7. October 2009

1. Introduction

The program described in this note has been written funded by the ESSnet for confidentiality. It is a response to a demand for finding a practical and automated way to export SAS datasets for input to Argus without having to interfere with much manual work in establishing a metafile with information that in principle is already contained in the SAS dataset.

The program has been formulated as a SAS macro called **TOARGUS**. This macro does three things

1. It reads a SAS micro data set and puts a set of variables listed in a **%LET KEEP = …;** statement out to a .csv file (semicolon separated). All formats that may be associated with the variables in SAS are stripped leaving pure codes.
2. It creates an appropriate a metadata file (.rda) with the same first name as the .csv file. This .rda file can be used for input of the .csv micro data file to either Mu or Tau Argus. The metadata file contains the basic information that is needed for Argus to read the .csv file but can if desired be edited. Variable attributes that are specific to confidentiality problems and do not have special a special status in a SAS dataset, such as sample weights, holding indicators, household identifiers or indicator for request protection, are treated as ordinary variables and their status must be defined from Argus.
3. It creates a SAS program file (.sas) containing the **INPUT** statement to read a safe file (.saf) from Mu-argus back to SAS. The **INPUT** statement assumes that the .saf file has the same semicolon separated .csv format as the original .csv file and will often have to be edited. The file is mainly meant as a staring point for editing. Double quotes enclosing character variables in .saf files must be removed before using this **INPUT**.

The input to **TOARGUS** is specified by SAS macro variables. This is demonstrated in section 3.

SAS has '.' as default missing value for numerical variables and ' ' (blank) for character variables. With the MISSING statement single letters (a-z) can be defined as special missing values. They will then apply to all numerical variables. As I have not tested whether such special numeric values are accepted as missing values for numerical variables in ARGUS, all missing values for numerical variables are output as '.'. A single blank is defined as missing for all character variables. If other values should be defined as missing for some variables in Argus they must be defined as such in the Argus Metafile menu.

TOARGUS only considers SAS micro data files as input. If tabular data from SAS should be input to Tau-Argus it is necessary to write a SAS macro that creates the table in the format and with the information that tau-Argus requires. However, it should not be difficult to write such a macro based on **PROC FREQ** in SAS. Writing such a procedure based on **PROC TABULATE** may be more difficult.

It should also be possible to generate code list files (.cdl) automatically based on the SAS formats. This has so far not been done. It should be noted that when producing the .csv file, the program strips all formats previously assigned to the variables.

The macro and how to run it is described in the next two sections.

2. The macro TOARGUS

The macro **TOARGUS** has the following code

```
%MACRO TOARGUS;
  LIBNAME &LIBNAME &INPATH; RUN;
  %NUMBER(&KEEP,NVAR);
/*
  Creates datafile (.csv, semicolon separated);
*/
  DATA _null_; SET &LIBNAME..&DATASET (KEEP=&KEEP);
    FILE "&OUTPATH\&DATASET..csv";
      FORMAT &KEEP;     * Removes all formats;
    %DO I=1 %TO &NVAR;
        %LET VAR = %SCAN(&KEEP,&I);
        IF VTYPE(&VAR.)='C' THEN DO; * Checks if variable character;
          %IF &I<&NVAR %THEN %DO; PUT &VAR $ +(-1) ';' @; %END;
          %IF &I=&NVAR %THEN %DO; PUT &VAR $;             %END;
        END;
        IF VTYPE(&VAR.)='N' THEN DO; * Checks if variable numeric;
          IF MISSING(&VAR.) THEN &VAR = '.';
          %IF &I<&NVAR %THEN %DO; PUT &VAR +(-1) ';' @; %END;
          %IF &I=&NVAR %THEN %DO; PUT &VAR;             %END;
        END;
      %END;
  RUN;
/*
  Creates metadatafile (.rda)
*/
  PROC DATASETS LIBRARY=WORK NODETAILS NOLIST;
    CONTENTS DATA=&LIBNAME..&DATASET(KEEP=&KEEP)
    OUT=WORK.meta1 (KEEP=name type length varnum Format) varnum nods
noprint;
  RUN;
  PROC SORT DATA=WORK.meta1; BY varnum; RUN;
  DATA _null_; SET WORK.meta1;
    FILE "&OUTPATH\&DATASET..rda";
    IF _n_=1 THEN PUT '<SEPARATOR>' ' ' '";"';
    IF type=1 THEN PUT name length '.' / '<NUMERIC>';
    IF type=2 THEN PUT name length ' ' / '<RECODABLE>';
  RUN;
/*
Creates the INPUT statement for the safe file to SAS. (When using Mu-
Argus.) The code assumes that the same variables as output to the .rda file
are to be input from a .saf file with the same .csv format as the original
file. The code can be edited and must then be included in a SAS data step.
*/
  DATA _null_; SET WORK.meta1;
    FILE "&OUTPATH\&DATASET..sas";
      IF _n_=1 THEN
```

```
      PUT 'INFILE &OUTPATH\&DATASET..saf ' "DELIMITER=';'" ' DSD;' /
'INPUT' ;
      IF type=1 THEN PUT name ': '      length +(-1) '.';
      IF type=2 THEN PUT name ': $CHAR' length +(-1) '.';
      IF _n_=&NVAR THEN PUT ';';
  RUN;
%MEND TOARGUS;

%MACRO NUMBER(A,B);
/*
   This macro reads a list of names (&B) and counts the number of names in
the list.

   Example: A list of names can be gives as
   %LET NAMES = age sex income education county status;
   %NUMBER(&NAMES,NNAMES); Produces the macro variable NNAMES with the
content &NNAMES = 6.
*/
   %GLOBAL &B;
   %LET A=&A DUMMY;
   %DO I=1 %TO 1000;
      %LET F=%SCAN(&A,&I);
      %IF &F=DUMMY %THEN %GOTO C1;
      %END;
%C1: %LET &B=%EVAL(&I-1);
%MEND NUMBER;
```

3. Running the macro

An example from a Norwegian application shows how to run the macro. Macro variables being used in the **TOARGUS** must be specified. In the example the SAS dataset is read from a UNIX server but run on a PC with PC-SAS and the .csv file and the .rda files are both written to PC-disk.

```
SIGNON ovibos tbufsize=16384;
      * Calls for logon to UNIX. For this example only.;
%LET LIBNAME = NIH;
      * The libname for the SAS dataset to be read;
%LET INPATH = REMOTE '$METODER/1253/NIH_kartlegging/wk12' SERVER=ovibos;
      * Path to the SAS dataset;
%LET DATASET = utvalg;
      * The name of the SAS dataset;
%LET OUTPATH = H:\ESSnet\Confidentiality;
      * Directory for .csv and .rda files;
%LET KEEP = t_senter kjonn barn sivstand samtykke retur res utland land
fylke alder;
      * The variables to be transferred from the SAS dataset;
RUN;
/*
Run the macro.
*/
%INCLUDE "H:\ESSnet\Confidentiality\SAS-Argus\ToArgus.sas";
%TOARGUS;
```

The .rda file output from this example looks as follows

```
<SEPARATOR> ";"
t_senter 2
```

```
<RECODABLE>
kjonn 1
<RECODABLE>
barn 8 .
<NUMERIC>
sivstand 1
<RECODABLE>
samtykke 1
<RECODABLE>
retur 1
<RECODABLE>
res 1
<RECODABLE>
utland 4
<RECODABLE>
land 4
<RECODABLE>
fylke 4
<RECODABLE>
alder 8 .
<NUMERIC>
```

The .sas file produced in the last step looks as

```
INFILE &OUTPATH\&DATASET..saf DELIMITER=';' DSD;
INPUT
t_senter : $CHAR2.
kjonn : $CHAR1.
barn : 8.
sivstand : $CHAR1.
samtykke : $CHAR1.
retur : $CHAR1.
res : $CHAR1.
utland : $CHAR4.
land : $CHAR4.
fylke : $CHAR4.
alder : 8.
;
```

Now it's your turn. Suggestions for improvements and developments of code to create .cdl files and automatic generation of SAS input files for the Argus output are welcome.