

Guide to the SPSS_to_STATA conversion utility

The *SPSS_to_STATA_00.sbs* script converts data files between SPSS and STATA with no loss of precision¹ or variable formatting (see section 4 for exceptions). The version of STATA file generated will be that of the version of STATA on the machine or network drive.

It converts *all* dataset information including variable types, variable and value labels and missing value definitions. It also generates a conversion log of all information that is unavoidably lost in conversion between SPSS and STATA (this loss is inevitable given the differential data handling limits of the two packages),² and a comprehensive data dictionary, so that any absent information can be looked up as necessary.

Disclaimer: The software has not been developed or tested to a commercial standard and therefore is made available strictly on the basis that you accept it as is, and that you are solely responsible for any use made of it. The University of Essex does not give any warranties, including without limitation, as to the accuracy of the software and disclaims any liability to you or any third party anywhere in the world for any injury, damage, direct or indirect loss, consequential or economic loss or any other loss suffered as a result of the use of or reliance upon the software to the maximum extent permitted by law.

If you discover any faults or bugs in this script, or choose to adapt or improve it, please notify Alasdair Crockett, crockett@essex.ac.uk

Contents

1. Requirements
2. Running the script
3. What the script does
4. The script compared with StatTransfer version 8

This script was written by

Alasdair Crockett
Data Standards Manager
UK Data Archive - service provider of the Economic and Social Data Service (ESDS)
University of Essex
Wivenhoe Park
Colchester
Essex CO4 3SQ
England

¹ Doubles are used where appropriate to ensure no loss of precision. The STATA *compress* command is used to ensure there is no unnecessary inflation of file size given that social surveys contain many integer variables of byte format.

² The limits mentioned are relevant to STATA 9.x *intercooled* and/or *special edition* compared with SPSS 13. For a summary of data handling limits of SPSS, STATA and SAS see <http://www.ats.ucla.edu/stat/stata/faq/statalimits.htm>

1. Requirements

This script is executed within SPSS for Windows and requires that the machine it is run on has access to both SPSS for Windows and STATA for Windows (either via local or networked drives).³ STATA needs to be version 8 and upwards to define 'extended missing' values (.a to .z). The script has been tested on SPSS versions 11.0, 11.5, 12 and 13.

The script must be run against SPSS system (.sav) files with a ".sav" extension. It will not run against SPSS portable (.por) files.

The files you wish to convert must be located in a directory that you have write as well as read access to.

The script is designed to be run on a machine to which you have administrative rights. It should run on machines to which you do not but this has not been extensively tested.

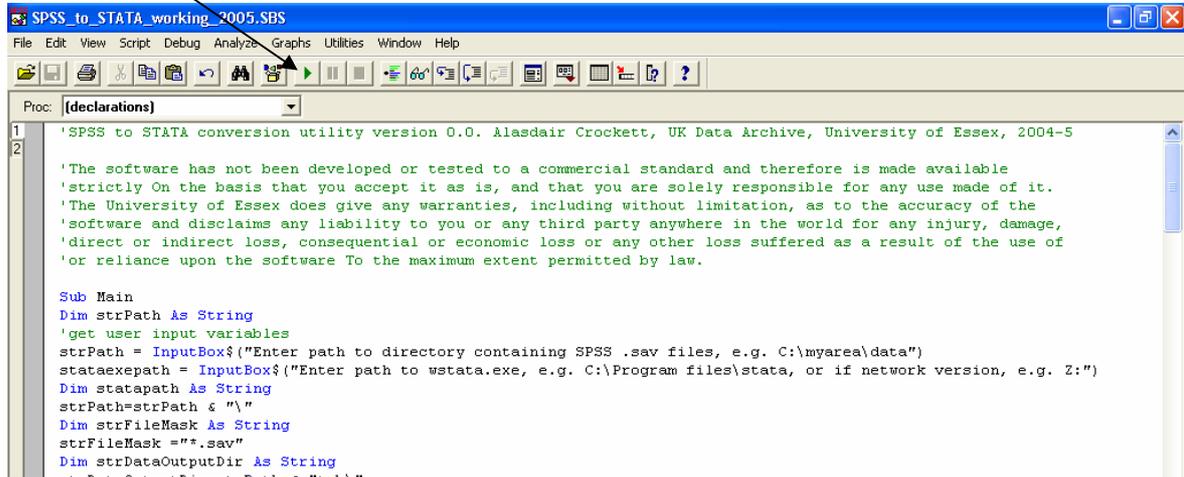
³ You can also run it on a machine with no access to STATA if you place a single inverted comma at the beginning of this line (near the end of the script):

```
' Shell stataexepath & "\wstata.exe /b /" & size & " do " & statapath & FileName & ".do"
```

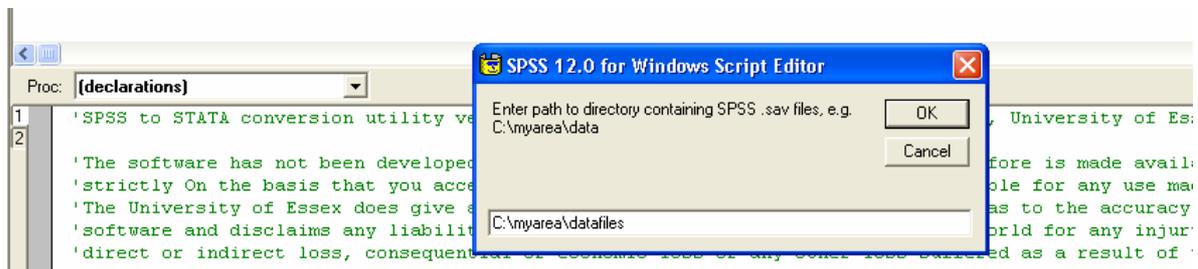
This means it will not execute the .do files, which can be executed at a later date, once there is access to STATA.

2. Running the script

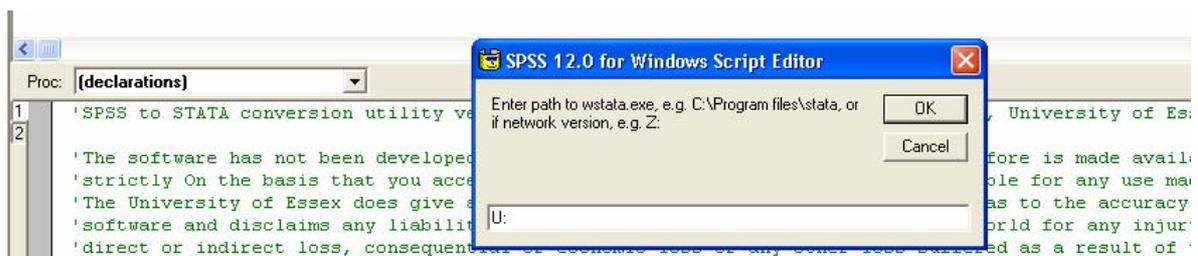
The script executes from within SPSS. Either double click on the script file, or open as a script within SPSS. Then select the run icon, which is a green triangle, as shown below:



The script will then ask you for two paths, the first is the directory with the SPSS .sav files that you wish to convert in. The directory may contain other files as well, the script will ignore them. Type in the full path to this directory, as illustrated below:



The script then asks you for the path to the STATA for Windows executable file (wstata.exe). If STATA is locally installed this path will normally be C:\Program Files\stata. Sometimes STATA is run over a network, in which case give the appropriate path. If you are unsure what this is, right click on the STATA icon or shortcut and select "properties". The "target location" is the relevant path. In the example below, STATA is run from a networked location mapped to the "U" drive.



Note that for very large data files, STATA may continue to create the final .do file after the script has finished executing. If this is the case you will notice a 'wstata' job in your program bar and/or task manager. If so, leave this to complete.

When the script finishes executing it will produce three beeps (if run on a machine on which you have administrative rights), and a pop-up box will appear announcing that it has finished (which needs to be checked before you do anything else in SPSS).

Note that SPSS does not appear to release all the memory used by the script when it finishes, so if converting large files, it is best to close SPSS (and relaunch it if desired) to free up resources.

3. What the script does

The script generates a number of outputs, most importantly the data files in STATA (.dta) format. For illustration, let us assume the script has been run against a number of SPSS .sav files in C:\myarea\data files. The script will generate the following directories and outputs:

i) [C:\myarea\data files\stata](#)

This directory is created to store the files named <data file name>.dta and <data file name>.do, where <data file name> is the name of the originating SPSS data file (minus the .sav extension).

The “*.dta” files are the STATA data files and the “*.do” files are the command files that created them.

ii) [C:\myarea\data files\tab](#)

This directory is created to store a series of files named <data file name>.tab. These are tab-delimited text exports of the SPSS data files, and are used to read the data into STATA using the “insheet” command.

iii) [C:\myarea\data files\data_documentation](#)

This directory will contain a single file called [SPSS_to_STATA_conversion.rtf](#) and a series of files (one for each data file) called <data file name>_Data_Dictionary.rtf.

The SPSS_to_STATA_conversion file contains a log of all lost or truncated information. As summarised at the top of this output the five unavoidable sources of loss are:

Truncation and potential truncation of data and/or labels

1. String variables defined with a width of > 80 characters (the intercooled STATA limit) or 244 characters (the special edition limit) in the SPSS file
2. Variable labels that are > 80 characters (the STATA limit) in the SPSS file
3. Value labels that are > 32 characters (the STATA limit) in the SPSS file

Outright loss of value labels

4. String variables that have value labels in the SPSS file
5. Non-integer values that have value labels in the SPSS file

Start of conversion warning log:

Warnings for mydatafile

i) Truncation and potential truncation of data and/or labels

Variable = **NAME2** String variable of width up to 254 chars in SPSS. STATA max. is 80 chars (244 in SE).

ii) Outright loss of value labels

Variable = **D_R** There are value labels for string variables in SPSS. These will not exist in the STATA file.

The <data file name>_Data_Dictionary files contain a comprehensive and easily readable dictionary of the data in the SPSS environment, so that lost information can be easily looked up (an example of data dictionary output is shown in the text box in section 4).

4. The script compared with StatTransfer version 8

This script is primarily designed for researchers without access to StatTransfer. StatTransfer is generally an excellent product and is used by the UK Data Archive to perform SPSS to STATA data format conversions. StatTransfer has been far more rigorously tested than this script, both by its developers and its user community.⁴

Since it is not freeware, however, not all academic staff will have access to StatTransfer. In addition, this script offers certain advantages over StatTransfer in some areas, but is more limited in others. The known pros and cons are listed below.

Pros

Creation of additional documentation (as detailed at section 3)

This gives you a reference guide to what information is (inevitably) lost on transfer between SPSS and STATA, and gives you a comprehensive data dictionary of the data in the SPSS environment to look the missing information up in.

Clearer mapping of 'user missing' values

The data dictionary also gives the STATA missing value that corresponds to the SPSS value. If there is a value label in SPSS, the SPSS missing value is appended to the beginning of the STATA value label. StatTransfer does not allow you to equate user missing values so easily and has no consistent system for conversion (.a to .z are used sequentially for SPSS user missing codes without value labels, but the first letter of the value label is used (if unique) where a label exists). With this script, both the STATA data file and the data dictionary show the mapping of SPSS to STATA user missing values as illustrated below:

Data in SPSS (-9,-8,and -7 all defined as user missing values):

value	label
-9	missing
-8	not applicable
-7	skipped

Resulting data in STATA using the SPSS_to_STATA script:

value	label
.a	(-9) missing
.b	(-8) not applicable
.c	(-7) skipped

Extract from a data dictionary file created by the script, which also shows the mapping:

Pos. = 1 **Variable = q1ib** **Variable label = Number of children in the Household**

This variable is *numeric*, the SPSS measurement level is *scale*.

SPSS user missing values = -9 (.a in STATA) and -8 (.b in STATA) and -7 (.c in STATA)

Value label information for q1ib

Value = -9	Label = missing
Value = -8	Label = not applicable
Value = -7	Label = skipped
Value = 1	Label = yes
Value = 2	Label = no

⁴ Note that if you are using StatTransfer to convert data from SPSS to STATA, to ensure optimal results always convert from SPSS system (.sav) rather than portable (.por) format, select 'optimisation' and 'allow doubles' options if using the GUI version (or the command switch /od if using the command processor).

SPSS Missing value ranges

The script will not assign user missing value ranges in SPSS as extended missing values in STATA. StatTransfer will map all SPSS values within a missing range to a single extended missing STATA code, making the original values unobtainable for manual comparison of the data matrix in SPSS and STATA.

Cons

Time format variables

The script is designed to convert all varieties of numeric, string and date format variables correctly, but certain date and time and time format variables in SPSS will appear as string variables in the STATA file (e.g. 12:56:01, if formatted as hh:mm:ss in SPSS), and will need to be manually redefined if required.

Date format variables

Whatever the date format in SPSS (American, European, Julian or sorted), the date will be displayed as ddmmyyyy in STATA, e.g. 14oct1999. StatTransfer will generally preserve a directly analogous date formatting.

Additional points

String variables that consist entirely of values that are valid numbers

These will be converted from string format in SPSS to numeric in STATA. This alteration has not been changed back as it is seen as a potentially desirable attribute and it is easy to convert the variable back within STATA if desired.