Title

stata.com

import sasxport5 — Import and export data in SAS XPORT Version 5 format

Description

`import sasxport5` and `export sasxport5` convert data from and to SAS XPORT Version 5 Transport format. The U.S. Food and Drug Administration uses this SAS XPORT Transport format as the format for datasets submitted with new drug and new device applications (NDAs).

`export sasxport5` saves the data in memory as a SAS XPORT Transport (`.xpt`) file. If needed, this command also creates `formats.xpf`—an additional XPORT file—containing the value-label definitions. These files can be easily read into SAS.

`import sasxport5` reads into memory data from a SAS XPORT Transport (`.xpt`) file. When available, this command also reads the value-label definitions stored in `formats.xpf` or `FORMATS.xpf`.

`import sasxport5, describe` describes the contents of a SAS XPORT Version 5 Transport file.

Quick start

Describe the contents of SAS XPORT Version 5 Transport file `mydata.xpt`

```
import sasxport5 mydata, describe
```

Load the contents of `mydata.xpt` into memory

```
import sasxport5 mydata
```

As above, and ignore the accompanying SAS formats file `formats.xpf`

```
import sasxport5 mydata, novallabels
```

Save data in memory to `mydata.xpt`

```
export sasxport5 mydata
```

As above, but rename variables to meet SAS XPORT restrictions

```
export sasxport5 mydata, rename
```

As above, and do not save value labels

```
export sasxport5 mydata, rename replace vallabfile(none)
```

Save `v1`, `v2`, and `v3` to `mydata.xpt`, where time variable `tvar` is equal to 2010

```
export sasxport5 v1 v2 v3 using mydata if tvar==2010
```

Menu

import sasxport5

File > Import > SAS XPORT Version 5 (*.xpt)

export sasxport5

File > Export > SAS XPORT Version 5 (*.xpt)

Syntax

Import SAS XPORT Version 5 Transport file into Stata

```
import sasxport5 filename [ , import_options ]
```

Describe contents of SAS XPORT Version 5 Transport file

```
import sasxport5 filename, describe [ member(mbrname) ]
```

Export data in memory to a SAS XPORT Version 5 Transport file

```
export sasxport5 filename [ if ] [ in ] [ , export_options ]
export sasxport5 varlist using filename [ if ] [ in ] [ , export_options ]
```

If *filename* is specified without an extension, `.xpt` is assumed. If *filename* contains embedded spaces, enclose it in double quotes.

import_options	Description
`clear`	replace data in memory
`novallabels`	ignore accompanying `formats.xpf` file if it exists
`member(`*mbrname*`)`	member to use; seldom used

export_options	Description
Main	
`rename`	rename variables and value labels to meet SAS XPORT restrictions
`replace`	overwrite files if they already exist
`vallabfile(xpf)`	save value labels in `formats.xpf`
`vallabfile(sascode)`	save value labels in SAS command file
`vallabfile(both)`	save value labels in `formats.xpf` and in a SAS command file
`vallabfile(none)`	do not save value labels

Options for import sasxport5

`describe` describes the contents of the SAS XPORT Version 5 Transport file. This option can be combined only with `member()`.

`clear` specifies that it is okay to replace the data in memory, even though the current data have not been saved to disk.

`novallabels` specifies that value-label definitions stored in `formats.xpf` or `FORMATS.xpf` not be looked for or loaded. By default, if variables are labeled in *filename*`.xpt`, then `import sasxport5` looks for `formats.xpf` to obtain and load the value-label definitions. If the file is not found, Stata looks for `FORMATS.xpf`. If that file is not found, a warning message is issued.

`import sasxport5` can use only a `formats.xpf` or `FORMATS.xpf` file to obtain value-label definitions. `import sasxport5` cannot understand value-label definitions from a SAS command file.

`member(`*mbrname*`)` specifies a member of the `.xpt` file. Although no longer often used, the original XPORT definition allowed multiple datasets to be placed in one file. The `member()` option allows you to read these old files, selecting only specific datasets (*members*) to be used by `import sasxport5`. You can obtain a list of member names by using `import sasxport5, describe`. By default, only the first member is used, unless `describe` is specified, in which case all members are described. Because it is rare for an XPORT file to have more than one member, this option is seldom used.

Options for export sasxport5

Main

`rename` specifies that `export sasxport5` may rename variables and value labels to attempt to meet the SAS XPORT restrictions, which are that names be no more than eight bytes long and that there be no distinction between uppercase and lowercase letters. Note that `rename` does not remove characters beyond the normal ASCII range, such as most Unicode characters and all extended ASCII characters. SAS may or may not support such characters in variable labels and value labels.

We recommend specifying the `rename` option. If this option is specified, any name violating the restrictions is changed to a different but related name in the file. The name changes are listed. The new names are used only in the file; the names of the variables and value labels in memory remain unchanged.

If `rename` is not specified and one or more names violate the XPORT restrictions, an error message will be issued and no file will be saved. The alternative to the `rename` option is that you can rename variables yourself with the `rename` command:

```
. rename mylongvariablename myname
```

See [D] **rename**. Renaming value labels yourself is more difficult. The easiest way to rename value labels is to use `label save`, edit the resulting file to change the name, execute the file by using `do`, and reassign the new value label to the appropriate variables by using `label values`:

```
. label save mylongvaluelabel using myfile.do
. doedit myfile.do    (change mylongvaluelabel to, say, mlvlab)
. do myfile.do
. label values myvar mlvlab
```

See [D] **label** and [R] **do** for more information about renaming value labels.

`replace` permits `export sasxport5` to overwrite existing *filename*`.xpt`, `formats.xpf`, and *filename*`.sas` files.

`vallabfile(xpf | sascode | both | none)` specifies whether and how value labels are to be stored. SAS XPORT Transport files do not really have value labels. Value-label definitions can be preserved in one of two ways:

1. In an additional SAS XPORT Version 5 Transport file whose data contain the value-label definitions
2. In a SAS command file that will create the value labels

`export sasxport5` can create either or both of these files.

`vallabfile(xpf)`, the default, specifies that value labels be written into a separate SAS XPORT Transport file named `formats.xpf`. Thus, `export sasxport5` creates two files: *filename*`.xpt`, containing the data, and `formats.xpf`, containing the value labels. No `formats.xpf` file is created if there are no value labels.

SAS users can easily use the resulting `.xpt` and `.xpf` XPORT files.
See https://www.sas.com/govedu/fda/macro.html, and click on the *FDA Submission Standards* tab. Then, click on the *Processing Data Sets Code* tab that appears below the "FDA and SAS Technology" text for SAS-provided macros for reading the XPORT files. The SAS macro `fromexp()` reads the XPORT files into SAS. The SAS macro `toexp()` creates XPORT files. When obtaining the macros, remember to save the macros at SAS's webpage as a plain-text file and to remove the examples at the bottom.

If the SAS macro file is saved as `C:\project\macros.mac` and the files `mydat.xpt` and `formats.xpf` created by `export sasxport5` are in `C:\project\`, the following SAS commands would create the corresponding SAS dataset and format library and list the data:

SAS commands

```
%include "C:\project\macros.mac" ;
%fromexp(C:\project, C:\project) ;
libname library 'C:\project' ;
data _null_ ; set library.mydat ; put _all_ ; run ;
proc print data = library.mydat ;
quit ;
```

`vallabfile(sascode)` specifies that the value labels be written into a SAS command file, *filename*`.sas`, containing SAS `proc format` and related commands. Thus, `export sasxport5` creates two files: *filename*`.xpt`, containing the data, and *filename*`.sas`, containing the value labels. SAS users may wish to edit the resulting *filename*`.sas` file to change the "libname datapath" and "libname xptfile xport" lines at the top to correspond to the location that they desire. `export sasxport5` sets the location to the current working directory at the time `export sasxport5` was issued. No `.sas` file will be created if there are no value labels.

`vallabfile(both)` specifies that both the actions described above be taken and that three files be created: *filename*`.xpt`, containing the data; `formats.xpf`, containing the value labels in XPORT format; and *filename*`.sas`, containing the value labels in SAS command-file format.

`vallabfile(none)` specifies that value-label definitions not be saved. Only one file is created: *filename*`.xpt`, which contains the data.

Remarks and examples

stata.com

All users, of course, may use these commands to transfer data between SAS and Stata, but there are limitations in the SAS XPORT Transport format, such as the eight-character limit on the names of variables (specifying `export sasxport5`'s `rename` option works around that). For a complete listing of limitations and issues concerning the SAS XPORT Transport format and an explanation of how `export sasxport5` and `import sasxport5` work around these limitations, see *Technical appendix* below.

Remarks are presented under the following headings:

Saving XPORT files for transferring to SAS

▷ Example 1: Exporting data to XPORT files

To demonstrate, we first load `auto.dta`. To save only variables `make`, `mpg`, and `weight` in `auto_sub.xpt`, we type

```
. use https://www.stata-press.com/data/r16/auto
(1978 Automobile Data)
. export sasxport5 make mpg weight using auto_sub
file auto_sub.xpt saved
```

We can save all the variables in the data to `auto.xpt` and save the value labels in `formats.xpf`. We specify the `rename` option to rename variable names and value labels that are too long or are case sensitive.

```
. export sasxport5 auto, rename
the following variable(s) were renamed in the output file:
                    displacement -> DISPLACE
                      gear_ratio -> GEAR_RAT
file auto.xpt saved
file formats.xpf saved
```

Alternatively, we can save the data in `auto.xpt` and save the value labels to a `formats.xpf` file and in a SAS command file `auto.sas`. We include the `replace` option to allow replacement of the files we created with our previous command.

```
. export sasxport5 auto, rename replace vallabfile(both)
the following variable(s) were renamed in the output file:
                    displacement -> DISPLACE
                      gear_ratio -> GEAR_RAT
file auto.xpt saved
file auto.sas saved
file formats.xpf saved
```

If we instead wanted to save the value labels only in the SAS command file, we could have typed

```
. export sasxport5 auto, rename replace vallabfile(sas)
```

If we did not want to save the value labels at all, thus creating only `auto.xpt`, we could have typed

```
. export sasxport5 typed, rename replace vallabfile(none)
```

◁

Determining the contents of XPORT files received from SAS

▷ Example 2: Describing XPORT files

To investigate the contents of the `auto.xpt` file we created above, we can type

```
. import sasxport5 auto, describe
data from auto.xpt, member(auto)
  obs:                74             01mar19:15:24:46
 vars:                12             (date shown exactly as recorded in file)
 size:             8,140
-------------------------------------------------------------------------------
variable     variable      value
  name         type        label      variable label
-------------------------------------------------------------------------------
make         str18                    Make and Model
price        numeric                  Price
mpg          numeric                  Mileage (mpg)
rep78        numeric                  Repair Record 1978
headroom     numeric                  Headroom (in.)
trunk        numeric                  Trunk space (cu. ft.)
weight       numeric                  Weight (lbs.)
length       numeric                  Length (in.)
turn         numeric                  Turn Circle (ft.)
displace     numeric                  Displacement (cu. in.)
gear_rat     numeric                  Gear Ratio
foreign      numeric      origin      Car type
-------------------------------------------------------------------------------
```

◁

Using XPORT files received from SAS

▷ Example 3: Importing XPORT files

To read data from `auto.xpt` and obtain value labels from `formats.xpf`, we can type

```
. import sasxport5 auto, clear
```

◁

Stored results

`import sasxport5, describe` stores the following in `r()`:

Scalars

`r(N)`	number of observations
`r(k)`	number of variables
`r(size)`	size of data
`r(n_members)`	number of members

Macros

`r(members)`	names of members

Technical appendix

Technical details concerning the SAS XPORT Version 5 Transport format and how `export sasxport5` and `import sasxport5` handle issues regarding the format are presented under the following headings:

A1. Overview of SAS XPORT Transport format

A SAS XPORT Transport file may contain one or more separate datasets, known as members. It is rare for a SAS XPORT Transport file to contain more than one member. See https://support.sas.com/techsup/technote/ts140.pdf for the SAS technical document describing the layout of the SAS XPORT Transport file.

A SAS XPORT dataset (member) is subject to certain restrictions:

1. The dataset may contain only 9,999 variables.
2. The names of the variables and value labels may not be longer than eight characters and are case insensitive; for example, `myvar`, `Myvar`, `MyVar`, and `MYVAR` are all the same name.
3. Variable labels may not be longer than 40 characters.
4. The contents of a variable may be numeric or string:
    a. Numeric variables may be integer or floating but may not be smaller than 5.398e–79 or greater than 9.046e+74, absolutely. Numeric variables may contain missing, which may be `.`, `._`, `.a`, `.b`, ..., `.z`.
    b. String variables may not exceed 200 characters. String variables are recorded in a "padded" format, meaning that, when variables are read, it cannot be determined whether the variable had trailing blanks.
5. Value labels are *not* written in the XPORT dataset. Suppose that you have variable `sex` in the data with values 0 and 1 and that the values are labeled for gender (0 = male, and 1 = female). When the dataset is written in SAS XPORT Transport format, you can record that the variable label `gender` is associated with the `sex` variable, but you cannot record the association with the value labels male and female.

    Value-label definitions are typically stored in a second XPORT dataset or in a text file containing SAS commands. You can use the `vallabfile()` option of `export sasxport5` to produce these datasets or files.

Value labels and formats are recorded in the same position in an XPORT file, meaning that names corresponding to formats used in SAS cannot be used. Thus, value labels may not be named

> `best`, `binary`, `comma`, `commax`, `d`, `date`, `datetime`, `dateampm`, `day`, `ddmmyy`, `dollar`, `dollarx`, `downame`, `e`, `eurdfdd`, `eurdfde`, `eurdfdn`, `eurdfdt`, `eurdfdwn`, `eurdfmn`, `eurdfmy`, `eurdfwdx`, `eurdfwkx`, `float`, `fract`, `hex`, `hhmm`, `hour`, `ib`, `ibr`, `ieee`, `julday`, `julian`, `percent`, `minguo`, `mmddyy`, `mmss`, `mmyy`, `monname`, `month`, `monyy`, `negparen`, `nengo`, `numx`, `octal`, `pd`, `pdjulg`, `pdjuli`, `pib`, `pibr`, `pk`, `pvalue`, `qtr`, `qtrr`, `rb`, `roman`, `s370ff`, `s370fib`, `s370fibu`, `s370fpd`, `s370fpdu`, `s370fpib`, `s370frb`, `s370fzd`, `s370fzdl`, `s370fzds`, `s370fzdt`, `s370fzdu`, `ssn`, `time`, `timeampm`, `tod`, `weekdate`, `weekdatx`, `weekday`, `worddate`, `worddatx`, `wordf`, `words`, `year`, `yen`, `yymm`, `yymmdd`, `yymon`, `yyq`, `yyqr`, `z`, `zd`, or any uppercase variation of these.

We refer to this as the "Known Reserved Word List" in this documentation. Other words may also be reserved by SAS; the technical documentation for the SAS XPORT Transport format provides no guidelines. This list was created by examining the formats defined in *SAS Language Reference: Dictionary, Version 8*. If SAS adds new formats, the list will grow.

6. A flaw in the XPORT design can make it impossible, in rare instances, to determine the exact number of observations in a dataset. This problem can occur only if 1) all variables in the dataset are string and 2) the sum of the lengths of all the string variables is less than 80. Actually, the above is the restriction, assuming that the code for reading the dataset is written well. If it is not, the flaw could occur if 1) the last variable or variables in the dataset are string and 2) the sum of the lengths of all variables is less than 80.

   To prevent stumbling over this flaw, make sure that the last variable in the dataset is not a string variable. This is always sufficient to avoid the problem.

7. There is no provision for saving the Stata concepts `notes` and `characteristics`.

A2. Implications for writing XPORT datasets from Stata

Stata datasets for the most part fit well into the SAS XPORT Transport format. With the same numbering scheme as above,

1. Stata refuses to write the dataset if it contains more than 9,999 variables.
2. Stata issues an error message if any variable or label name violates the naming restrictions, or if the `rename` option is specified, Stata fixes any names that violate the restrictions.

   Whether or not `rename` is specified, names will be recorded without regard to case: you do not have to name all your variables with all lowercase or all uppercase letters. Stata verifies that ignoring case does not lead to problems, complaining or, if option `rename` is specified, fixing them.
3. Stata truncates variable labels to 40 characters to fit within the XPORT limit.
4. Stata treats variable contents as follows:
   a. If a numeric variable records a value greater than 9.046e+74 in absolute value, Stata issues an error message. If a variable records a value less than 5.398e–79 in absolute value, 0 is written.

b. If you have string variables longer than 200 characters, Stata issues an error message. Also, if any string variable has trailing blanks, Stata issues an error message. To remove trailing blanks from string variable `s`, you can type

```
. replace s = rtrim(s)
```

To remove leading and trailing blanks, type

```
. replace s = trim(s)
```

5. Value-label names are written in the XPORT dataset. The contents of the value label are not written in the same XPORT dataset. By default, `formats.xpf`, a second XPORT dataset, is created containing the value-label definitions.

   SAS recommends creating a `formats.xpf` file containing the value-label definitions (what SAS calls format definitions). They have provided SAS macros, making the reading of `.xpt` and `formats.xpf` files easy. See https://www.sas.com/govedu/fda/macro.html for details.

   Alternatively, a SAS command file containing the value-label definitions can be produced. The `vallabfile()` option of `export sasxport5` is used to indicate which, if any, of the formats to use for recording the value-label definitions.

   If a value-label name matches a name on the Known Reserved Word List, and the `rename` option is not specified, Stata issues an error message.

   If a variable has no value label, the following format information is recorded:

Stata format	SAS format
%td...	MMDDYY10.
%-td...	MMDDYY10.
%#s	$CHAR#.
%-#s	$CHAR#.
% #s	$CHAR#.
all other	BEST12.

6. If you have a dataset that could provoke the XPORT design flaw, a warning message is issued. Remember, the best way to avoid this flaw is to ensure that the last variable in the dataset is numeric. This is easily done. You could, for instance, type

```
. generate ignoreme = 0
. export sasxport ...
```

7. Because the XPORT file format does not support notes and characteristics, Stata ignores them when it creates the XPORT file. You may wish to incorporate important notes into the documentation that you provide to the user of your XPORT file.

A3. Implications for reading XPORT datasets into Stata

Reading SAS XPORT Version 5 Transport format files into Stata is easy, but sometimes there are issues to consider:

1. If there are too many variables, Stata issues an error message. If you are using Stata/MP or Stata/SE, you can increase the maximum number of variables with the `set maxvar` command; see [D] **memory**.

2. The XPORT variable-naming restrictions are more restrictive than those of Stata, so no problems should arise. However, Stata reserves the following names:

   _all, _b, byte, _coef, _cons, double, float, if, in, int, long, _n, _N, _pi, _pred, _rc, _skip, str#, strL, using, with

   If the XPORT file contains variables with any of these names, Stata issues an error message. Also, the error message

   ```
   . import sasxport5 ...
   ________ already defined
   r(110);
   ```

   indicates that the XPORT file was incorrectly prepared by some other software and that two or more variables share the same name.

3. The XPORT variable-label-length limit is more restrictive than that of Stata, so no problems can arise.

4. Variable contents may cause problems:

   a. The range of numeric variables in an XPORT dataset is a subset of that allowed by Stata, so no problems can arise. All variables are brought back as doubles; we recommend that you run compress after loading the dataset:

      ```
      . import sasxport5 ...
      . compress
      ```

      See [D] **compress**.

      Stata has no missing-value code corresponding to ._. If any value records ._, then .u is stored.

   b. String variables are brought back as recorded but with all trailing blanks stripped.

5. Value-label names are read directly from the XPORT dataset. Any value-label definitions are obtained from a separate XPORT dataset, if available. If a value-label name matches any in the Known Reserved Word List, no value-label name is recorded, and instead, the variable display format is set to %9.0g, %10.0g, or %td.

   The %td Stata format is used when the following SAS formats are encountered:

   DATE, EURDFDN, JULDAY, MONTH, QTRR, YEAR, DAY, EURDFDWN, JULIAN, MONYY, WEEKDATE, YYMM, DDMMYY, EURDFMN, MINGUO, NENGO, WEEKDATX, YYMMDD, DOWNAME, EURDFMY, MMDDYY, PDJULG, WEEKDAY, YYMON, EURDFDD, EURDFWDX, MMYY, PDJULI, WORDDATE, YYQ, EURDFDE, EURDFWKX, MONNAME, QTR, WORDDATX, YYQR

   If the XPORT file indicates that one or more variables have value labels, import sasxport5 looks for the value-label definitions in formats.xpf, another XPORT file. If it does not find this file, it looks for FORMATS.xpf. If this file is not found, import sasxport5 issues a warning message unless the novallabels option is specified.

   Stata does not allow value-label ranges or string variables with value labels. If the .xpt file or formats.xpf file contains any of these, an error message is issued. The novallabels option allows you to read the data, ignoring all value labels.

6. If a dataset is read that provokes the all-strings XPORT design flaw, the dataset with the minimum number of possible observations is returned, and a warning message is issued. This duplicates the behavior of SAS.

7. SAS XPORT format does not allow notes or characteristics, so no issues can arise.

Also see

[D] **import sas** — Import SAS files

[D] **import sasxport8** — Import and export data in SAS XPORT Version 8 format

[D] **export** — Overview of exporting data from Stata

[D] **import** — Overview of importing data into Stata