

## import dbase — Import and export dBase files

Description	Quick start	Menu	Syntax
Options for import dbase	Options for export dbase	Remarks	Stored results
Also see			

## Description

`import dbase` reads into memory a version III or version IV dBase (.dbf) file. `export dbase` exports data in memory to a version IV dBase (.dbf) file.

Stata has other commands for importing data. If you are not sure that `import dbase` will do what you are looking for, see [D] [import](#) and [U] [22 Entering and importing data](#).

## Quick start

Load the contents of the dBase file called `mydata.dbf`

```
import dbase mydata
```

Write data in memory to a version IV dBase file called `mydata.dbf`

```
export dbase mydata
```

As above, but export only variables `v1` and `v2`

```
export dbase v1 v2 using mydata
```

## Menu

### import dbase

File > Import > dBase (\*.dbf)

### export dbase

File > Export > dBase (\*.dbf)

## Syntax

*Load a dBase file*

```
import dbase [using] filename [, clear case(preserve|lower|upper)]
```

*Save data in memory to a dBase file*

```
export dbase [using] filename [if] [in] [, datafmt replace]
```

*Save subset of variables in memory to a dBase file*

```
export dbase [varlist] using filename [if] [in] [, datafmt replace]
```

If *filename* is specified without an extension, .dbf is assumed for both `import dbase` and `export dbase`. If *filename* contains embedded spaces, enclose it in double quotes.

## Options for import dbase

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

`case(preserve|lower|upper)` specifies the case of the variable names after import. The default is `case(preserve)`.

## Options for export dbase

`datafmt` specifies that all variables be exported using their display format. For example, the number 1000 with a display format of `%7.2f` would export as 1000.00, not 1000. The default is to use the raw, unformatted value when exporting.

`replace` specifies that *filename* be replaced if it already exists.

## Remarks

`import dbase` reads into memory a version III or version IV dBase (`.dbf`) file. If the dBase file is not version III or IV, `import dbase` will issue an error. dBase files are often paired with shapefiles for storing geometric location data. To import a shapefile, see [SP] [spshape2dta](#).

`export dbase` exports data in memory to a version IV dBase (`.dbf`) file. dBase version IV has several file limitations when exporting.

1. Unicode is not supported.
2. Data cannot be more than 2 GB in size.
3. Data in memory must be less than 1,000,000,000 observations.
4. Data in memory must have less than 255 variables.
5. Variable names cannot exceed 10 characters in length.
6. Maximum string variable length is 255 characters.
7. Data width must be less than 4,000.

If your data in memory exceed any of these limits, `export dbase` will issue an error when trying to export the data.

To demonstrate the use of `import dbase` and `export dbase`, we will first load `autornd.dta` and export it as a dBase file named `auto.dbf`.

```
. use https://www.stata-press.com/data/r16/autornd
(1978 Automobile Data)
. export dbase auto.dbf
file auto.dbf saved
```

To import the data back into Stata, we need only to specify the filename. `import dbase` assumes an extension of `.dbf`.

```
. import dbase auto, clear
(3 vars, 74 obs)
```

We could verify that our data loaded correctly by using [list](#) or [browse](#).

## Stored results

`import dbase` stores the following in `r()`:

Scalars

<code>r(N)</code>	number of observations imported
<code>r(k)</code>	number of variables imported

## Also see

[D] [export](#) — Overview of exporting data from Stata

[D] [import](#) — Overview of importing data into Stata

[SP] [spshape2dta](#) — Translate shapefile to Stata format