Title

> **import haver** — Import data from Haver Analytics databases

Syntax

Load Haver data

>   import <u>hav</u>er *seriesdblist* [, *import_haver_options*]

Load Haver data using a dataset that is loaded in memory

>   import <u>hav</u>er, <u>fromm</u>emory [*import_haver_options*]

Describe contents of Haver database

>   import <u>hav</u>er *seriesdblist*, <u>des</u>cribe [*import_haver_describe_options*]

Specify the directory where the Haver databases are stored

>   set haverdir "*path*" [, <u>permanently</u>]

import_haver_options	Description
<u>fin</u>([*datestring*], [*datestring*])	load data within specified date range
<u>fwithin</u>([*datestring*], [*datestring*])	same as fin() but exclude the end points of range
<u>tvar</u>(*varname*)	create time variable *varname*
case(<u>lower</u>|<u>upper</u>)	read variable names as lowercase or uppercase
<u>hmissing</u>(*misval*)	record missing values as *misval*
<u>aggmethod</u>(strict|relaxed|force)	set how temporal aggregation calculations deal with missing data
frommemory	load data using file in memory
clear	clear data in memory before loading Haver database

frommemory and clear do not appear in the dialog box.

import_haver_describe_options	Description
* <u>des</u>cribe	describe contents of *seriesdblist*
<u>det</u>ail	list full series information table for each series
<u>sav</u>ing(*filename* [, verbose replace])	save series information to *filename*.dta

*describe is required.

seriesdblist is one or more of the following:

> *dbfile*
> *series*@*dbfile*
> (*series series* . . .)@*dbfile*

where *dbfile* is the name of a Haver Analytics database and *series* contains a Haver Analytics series. Wildcards ? and * are allowed in *series*. *series* and *dbfile* are not case sensitive.

Example: `import haver gdp@usecon`
Import series GDP from the USECON database.

Example: `import haver gdp@usecon c1*@ifs`
Import series GDP from the USECON database, and import any series that starts with c1 from the IFS database.

Note: You must specify a path to the database if you did not use the `set haverdir` command.

Example: `import haver gdp@"C:\data\usecon" c1*@"C:\data\ifs"`

If you do not specify a path to the database and you did not `set haverdir`, then `import haver` will look in the current working directory for the database.

Menu

File > Import > Haver Analytics database

Description

Haver Analytics (http://www.haver.com) provides economic and financial databases to which you can purchase access. The `import haver` command allows you to use those databases with Stata. The `import haver` command is provided only with Stata for Windows.

`import haver` *seriesdblist* loads data from one or more Haver databases into Stata's memory.

`import haver` *seriesdblist*, `describe` describes the contents of one or more Haver databases.

If a database is specified without a suffix, then the suffix `.dat` is assumed.

Options for import haver

`fin(`[*datestring*]`,` [*datestring*]`)` specifies the date range of the data to be loaded. *datestring* must adhere to the Stata default for the different frequencies. See [D] **datetime display formats**. Examples are 23mar2012 (daily and weekly), 2000m1 (monthly), 2003q4 (quarterly), and 1998 (annual). `fin(1jan1999, 31dec1999)` would mean from and including 1 January 1999 through 31 December 1999. Note that weekly data must be specified as daily data because Haver-week data are conceptually different than Stata-week data.

`fin()` also determines the aggregation frequency. If you want to retrieve data in a frequency that is lower than the one in which the data are stored, specify the dates in option `fin()` accordingly. For example, to retrieve series that are stored in quarterly frequency into an annual dataset, you can type `fin(1980,2010)`.

`fwithin(`[*datestring*]`,` [*datestring*]`)` functions the same as `fin()` except that the endpoints of the range will be excluded in the loaded data.

tvar(*varname*) specifies the name of the time variable Stata will create. The default is tvar(time). The tvar() variable is the name of the variable that you would use to tsset the data after loading, although doing so is unnecessary because import haver automatically tssets the data for you.

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

hmissing(*misval*) specifies which of Stata's 27 missing values (., .a, ..., .z) to record when there are missing values in the Haver database.

Two kinds of missing values occur in Haver databases. The first occurs when nothing is recorded because the data do not span the entire range; these missing values are always stored as . by Stata. The second occurs when Haver has recorded a Haver missing value; by default, these are stored as . by Stata, but you can use hmissing() to specify that a different extended missing-value code be used.

aggmethod(strict │ relaxed │ force) specifies a method of temporal aggregation in the presence of missing observations. aggmethod(strict) is the default aggregation method.

Most Haver series of higher than annual frequency has an aggregation type that determines how data can be aggregated. The three aggregation types are average (AVG), sum (SUM), and end of period (EOP). Each aggregation method behaves differently for each aggregation type.

An aggregated span is a time period expressed in the original frequency. The goal is to aggregate the data in an aggregation span to a single observation in the (lower) target frequency. For example, 1973m1–1973m3 is an aggregated span for quarterly aggregation to 1973q1.

strict aggregation method:

  1) (Average) The aggregated value is the average value if no observation in the aggregated span is missing; otherwise, the aggregated value is missing.

  2) (Sum) The aggregated value is the sum if no observation in the aggregated span is missing; otherwise, the aggregated value is missing.

  3) (End of period) The aggregated value is the series value in the last period in the aggregated span, be it missing or not.

relaxed aggregation method:

  1) (Average) The aggregated value is the average value as long as there is one nonmissing data point in the aggregated span; otherwise, the aggregated value is missing.

  2) (Sum) The aggregated value is the sum if no observation in the aggregated span is missing; otherwise, the aggregated value is missing.

  3) (End of period) The aggregated value is the last available nonmissing data point in the aggregated span; otherwise, the aggregated value is missing.

force aggregation method:

  1) (Average) The aggregated value is the average value as long as there is one nonmissing data point in the aggregated span; otherwise, the aggregated value is missing.

  2) (Sum) The aggregated value is the sum if there is at least one nonmissing data point in the aggregated span; otherwise, the aggregated value is missing.

  3) (End of period) The aggregated value is the last available nonmissing data point in the aggregated span; otherwise, the aggregated value is missing.

The following options are available with import haver but are not shown in the dialog box:

frommemory specifies that each observation of the dataset in memory specifies the information for a Haver series to be imported. The dataset in memory must contain variables named path, file, and series. The observations in path specify paths to Haver databases, the observations in file specify Haver databases, and the observations in series specify the series to import.

clear clears the data in memory before loading the Haver database.

Options for import haver, describe

describe describes the contents of one or more Haver databases.

detail specifies that a detailed report of all the information available on the variables be presented.

saving(*filename* [, verbose replace]) saves the series meta-information to a Stata dataset. By default, the series meta-information is not displayed to the Results window, but you can use the verbose option to display it.

saving() saves a Stata dataset that can subsequently be used with the frommemory option.

Option for set haverdir

permanently specifies that in addition to making the change right now, the haverdir setting be remembered and become the default setting when you invoke Stata.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Installation

Haver Analytics (http://www.haver.com) provides more than 200 economic and financial databases in the form of .dat files to which you can purchase access. The import haver command provides easy access to those databases from Stata. import haver is provided only with Stata for Windows.

Setting the path to Haver databases

If you want to retrieve data from Haver Analytics databases, you must discover the directory in which the databases are stored. This will most likely be a network location. If you do not know the directory, contact your technical support staff or Haver Analytics (http://www.haver.com). Once you have determined the directory location—for example, H:\haver_files—you can save it by using the command

```
. set haverdir "H:\haver_files\", permanently
```

Using the `permanently` option will preserve the Haver directory information between Stata sessions. Once the Haver directory is set, you can start retrieving data. For example, if you are subscribing to the USECON database, you can type

```
. import haver gdp@usecon
```

to load the GDP series into Stata. If you did not use `set haverdir`, you would type

```
. import haver gdp@"H:\haver_files\usecon"
```

The directory path passed to `set haverdir` is saved in the `creturn` value `c(haverdir)`. You can view it by typing

```
. display "`c(haverdir)'"
```

Download example Haver databases

There are three example Haver databases you can download to your working directory. Run the `copy` commands below to download HAVERD, HAVERW, and HAVERMQA.

```
. copy http://www.stata.com/haver/HAVERD.DAT haverd.dat
. copy http://www.stata.com/haver/HAVERD.IDX haverd.idx
. copy http://www.stata.com/haver/HAVERW.DAT haverw.dat
. copy http://www.stata.com/haver/HAVERW.IDX haverw.idx
. copy http://www.stata.com/haver/HAVERMQA.DAT havermqa.dat
. copy http://www.stata.com/haver/HAVERMQA.IDX havermqa.idx
```

To use these files, you need to make sure your Haver directory is not set:

```
. set haverdir ""
```

Determining the contents of a Haver database

`import haver` *seriesdblist*, `describe` displays the contents of a Haver database. If no series is specified, then all series are described.

```
. import haver haverd, describe
Dataset: haverd
```

Variable	Description	Time span	Frequency	Source
FXTWB	Nominal Broad Trade-W..	03jan2005-02mar2012	Daily	FRB
FXTWM	Nominal Trade-Weighte..	03jan2005-02mar2012	Daily	FRB
FXTWOTP	Nominal Trade-Weighte..	03jan2005-02mar2012	Daily	FRB

Summary

```
    number of series described: 3
             series not found: 0
```

Above we describe the Haver database `haverd.dat`, which we already have on our computer and in our current directory.

By default, each line of the output corresponds to one Haver series. Specifying `detail` displays more information about each series, and specifying *seriesname*@ allows us to restrict the output to the series that interests us:

```
. import haver FXTWB@haverd, describe detail
```

FXTWB	Nominal Broad Trade-Weighted Exchange Value of the US$ (1/97=100)

```
    Frequency: Daily                    Time span: 03jan2005-02mar2012
    Number of Observations: 1870        Date Modified: 07mar2012 11:27:33
    Aggregation Type: AVG               Decimal Precision: 4
    Difference Type: 0                  Magnitude: 0
    Data Type: INDEX                    Group: R03
    Primary Geography Code: 111         Secondary Geography Code:
    Source: FRB                         Source Description: Federal Reserv..
```

Summary

```
    number of series described: 1
              series not found: 0
```

You can describe multiple Haver databases with one command:

```
. import haver haverd haverw, describe
  (output omitted)
```

To restrict the output to the series that interest us for each database, you could type

```
. import haver (FXTWB FXTWOTP)@haverd FARVSN@haverw, describe
  (output omitted)
```

Loading a Haver database

`import haver` *seriesdblist* loads Haver databases. If no series is specified, then all series are loaded.

```
. import haver haverd, clear
```

Summary

```
     Haver data retrieval: 10 Dec 2012 11:41:18
     # of series requested: 3
     # of database(s) used: 1 (HAVERD)
      All series have been successfully retrieved
```

Frequency

```
    highest Haver frequency: Daily
     lowest Haver frequency: Daily
 frequency of Stata dataset: Daily
```

The table produced by `import haver` *seriesdblist* displays a summary of the loaded data, frequency information about the loaded data, series that could not be loaded because of errors, and notes about the data.

The dataset now contains a time variable and three variables retrieved from the HAVERD database:

```
. describe
Contains data
  obs:         1,870
 vars:             4
 size:        59,840
```

variable name	storage type	display format	value label	variable label
time	double	%td		
fxtwb_haverd	double	%10.0g		Nominal Broad Trade-Weighted Exchange Value of the US$ (1/97=100)
fxtwm_haverd	double	%10.0g		Nominal Trade-Weighted Exch Value of US$ vs Major Currencies (3/73=100)
fxtwotp_haverd	double	%10.0g		Nominal Trade-Weighted Exchange Value of US$ vs OITP (1/97=100)

```
Sorted by:  time
     Note:  dataset has changed since last saved
```

Haver databases include the following meta-information about each variable:

HaverDB	database name
Series	series name
DateTimeMod	date/time the series was last modified
Frequency	frequency of series (from daily to annual) as it is stored in the Haver database
Magnitude	magnitude of the data
DecPrecision	number of decimals to which the variable is recorded
DifType	relevant within Haver software only: if =1, percentage calculations are not allowed
AggType	temporal aggregation type (one of AVG, SUM, EOP)
DataType	type of data (e.g., ratio, index, US$, %)
Group	Haver series group to which the variable belongs
Geography1	primary geography code
Geography2	secondary geography code
StartDate	data start date
EndDate	data end date
Source	Haver code associated with the source for the data
SourceDescription	description of Haver code associated with the source for the data

When a variable is loaded, this meta-information is stored in variable characteristics (see [P] **char**). Those characteristics can be viewed using char list:

```
. char list fxtwb_haverd[]
fxtwb_haverd[HaverDB]:       HAVERD
fxtwb_haverd[Series]:        FXTWB
fxtwb_haverd[DateTimeMod]:   07mar2012 11:27:33
fxtwb_haverd[Frequency]:     Daily
fxtwb_haverd[Magnitude]:     0
fxtwb_haverd[DecPrecision]: 4
fxtwb_haverd[DifType]:       0
fxtwb_haverd[AggType]:       AVG
fxtwb_haverd[DataType]:      INDEX
fxtwb_haverd[Group]:         R03
fxtwb_haverd[Geography1]:    111
fxtwb_haverd[StartDate]:     03jan2005
fxtwb_haverd[EndDate]:       02mar2012
fxtwb_haverd[Source]:        FRB
fxtwb_haverd[SourceDescription]:
                             Federal Reserve Board
```

You can load multiple Haver databases/series with one command. To load the series FXTWB and FXTWOTP from the HAVERD database and all series that start with V from the HAVERMQA database, you would type

```
. import haver (FXTWB FXTWOTP)@haverd V*@havermqa, clear
(output omitted )
```

`import haver` automatically `tsset`s the data for you.

Loading a Haver database from a describe file

You often need to search through the series information of a Haver database(s) to see which series you would like to load. You can do this by saving the output of `import haver, describe` to a Stata dataset with the `saving(filename)` option. The dataset created can be used by `import haver, frommemory` to load data from the described Haver database(s). For example, here we search through the series information of database HAVERMQA.

```
. import haver havermqa, describe saving(my_desc_file)
(output omitted )
. use my_desc_file, clear
. describe
Contains data from my_desc_file.dta
  obs:            161
  vars:             8                           10 Dec 2012 11:41
  size:        19,642
```

variable name	storage type	display format	value label	variable label
path	str1	%9s		Path to Haver File
file	str8	%9s		Haver File Name
series	str7	%9s		Series Name
description	str80	%80s		Series Description
startdate	str7	%9s		Start Date
enddate	str7	%9s		End Date
frequency	str9	%9s		Frequency
source	str3	%9s		Source

```
Sorted by:
```

The resulting dataset contains information on the 164 series in HAVERMQA. Suppose that we want to retrieve all monthly series whose description includes the word "Yield". We need to keep only the observations from our dataset where the frequency variable equals "Monthly" and where the description variable contains "Yield".

```
. keep if frequency=="Monthly" & strpos(description,"Yield")
(152 observations deleted)
```

To load the selected series into Stata, we type

```
. import haver, frommemory clear
```

Note: We must `clear` the described data in memory to load the selected series. If you do not want to lose the changes you made to the description dataset, you must save it before using `import haver, frommemory`.

Temporal aggregation

If you request series with different frequencies, the higher frequency data will be aggregated to the lowest frequency. For example, if you request a monthly and a quarterly series, the monthly series will be aggregated. In rare cases, a series cannot be aggregated to a lower frequency and so will not be retrieved. A list of these series will be stored in r(noaggtype).

The options fin() and fwithin() are useful for aggregating series by hand.

Daily data

Haver's daily frequency corresponds to Stata's daily frequency. Haver's daily data series are business series for which business calendars are useful. See [D] **datetime business calendars** for more information on business calendars.

Weekly data

Haver's weekly data are also retrieved to Stata's daily frequency. See [D] **datetime business calendars** for more information on business calendars.

Stored results

import haver stores the following in r():

Scalars
    r(k_requested)   number of series requested
    r(k_noaggtype)   number of series dropped because of invalid aggregation type
    r(k_nodisagg)   number of series dropped because their frequency is lower than that of the output
                           dataset
    r(k_notindata)   number of series dropped because data were out of the date range specified in
                           fwithin() or fin()
    r(k_notfound)   number of series not found in the database

Macros
    r(noaggtype)   list of series dropped because of invalid aggregation type
    r(nodisagg)   list of series dropped because their frequency is lower than that of the output dataset
    r(notindata)   list of series dropped because data were out of the date range specified in fwithin()
                           or fin()
    r(notfound)   list of series not found in the database

import haver, describe stores the following in r():

Scalars
    r(k_described)   number of series described
    r(k_notfound)   number of series not found in the database

Macros
    r(notfound)   list of series not found in the database

Acknowledgment

import haver was written with the help of Daniel C. Schneider of the House of Finance at Goethe University, Frankfurt, Germany.

Also see

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

[D] **import delimited** — Import delimited text data

[D] **odbc** — Load, write, or view data from ODBC sources

[TS] **tsset** — Declare data to be time-series data