

frames save — Save a set of frames on disk

[Description](#)[Remarks and examples](#)[Quick start](#)[Stored results](#)[Menu](#)[Also see](#)[Syntax](#)[Options](#)

Description

`frames save` saves a set of frames as a Stata frameset (`.dtas`) file.

Quick start

Save frames A, B, and C in file `myframeset.dtas`

```
frames save myframeset, frames(A B C)
```

Save, in file `myframeset.dtas`, frames A and B as well as all frames linked, through `fmlink`, to A and B

```
frames save myframeset, frames(A B) linked
```

Menu

Data > Frames Manager

Syntax

```
frames save filename, frames(framelist) [options]
```

If *filename* is specified without an extension, `.dtas` is assumed. If *filename* contains embedded spaces or other special characters, enclose it in double quotes.

<i>options</i>	Description
* <code>frames(<i>framelist</i>)</code>	specify frames to be saved
<code>replace</code>	overwrite existing <code>.dtas</code> file
<code>linked</code>	save frames linked to those in <i>framelist</i>
<code>relaxed</code>	ignore missing linked frames
<code>complevel(#)</code>	specify compression level; default is <code>complevel(1)</code>
<code>nolabel</code>	omit value labels from the saved frames
<code>orphans</code>	save all value labels
<code>emptyok</code>	save specified frames even if they have zero observations and zero variables
<code>all</code>	save <code>e(sample)</code> with the frames in which it exists; programmer's option

*`frames(framelist)` is required.

`all` does not appear in the dialog box.

Options

`frames(framelist)` specifies the frames to be saved. *framelist* is a list of frame names separated by a space. To save all frames in memory, specify `frames(_all)`. `frames()` is required.

`replace` permits `frames save` to overwrite *filename* if it already exists.

`linked` specifies that all frames linked to those in *framelist* are also saved. Linkages are established by the `frlink` command. Note that if frame A is linked to frame B through `frlink`, and frame B is similarly linked to frame C, then saving frame A with the `linked` option will also save frames B and C, as well as other frames linked to B and C, and so forth.

`relaxed` is allowed only with the `linked` option. `relaxed` specifies that an error message not be issued if a linked frame does not exist.

`complevel(#)` specifies the compression level to be used. # may be any integer from 0 to 9; the default is `complevel(1)`. `complevel(0)` means no compression; a larger # means more compression. The compression level can also be set with `set dtascomplevel`; type `help set dtascomplevel` to learn more. `complevel()` overrides the `dtascomplevel` setting.

`no label` specifies that value labels from the saved frames are omitted.

`orphans` specifies that all value labels be saved, including those not attached to any variable.

`emptyok` specifies that the frames be saved even if they contain zero observations and zero variables.

The following option is available with `frames save` but is not shown in the dialog box:

`all` specifies that `e(sample)` be saved with the frames in which it exists. `all` is a programmer's option.

Remarks and examples

[stata.com](https://www.stata.com)

Data frames allow you to work with multiple datasets in memory and to access variables across those datasets. `frames save` allows you to save the data from multiple frames into a single file; the resulting file is referred to as a Stata frameset and uses the `.dtas` extension. You can simply specify the list of frames you want to save or specify that the listed frames and those linked to them be saved.

► Example 1: Save multiple frames

Suppose that we have two frames in memory and we want to save data from both in a single file. To demonstrate, we first create a frame named `census` and load a dataset with population data by state:

```
. clear all
. frame create census
. frame change census
. sysuse census
(1980 Census data by state)
```

Next we create a frame named `housing` in which we load housing data by state:

```
. frame create housing
. frame change housing
. webuse hsnq
(1980 Census housing data)
```

Now we save both frames, `census` and `housing`, into a file called `myframeset.dtas`:

```
. frames save myframeset, frames(census housing) replace
file myframeset.dtas saved
```

Whenever we wish to load the data frames from `myframeset.dtas`, we can use `frames use`.



▷ Example 2: Save linked frames

One advantage of working with data frames is that you can access values from one frame in another by linking the two frames. Furthermore, when you save data from a frame, you may wish to save data from the frames it is linked to, which we demonstrate below.

Continuing with our frames from [example 1](#), we can use `frame pwf` to check which frame is the working frame:

```
. frame pwf
(current frame is housing)
```

Our current frame is `housing`. We now use `frlink` to link frame `census` to frame `housing`, matching observations on values of `state`:

```
. frlink 1:1 state, frame(census)
(1 observation in frame housing unmatched)
```

The message indicates that only the value of `state` in frame `housing` was not matched in frame `census`. We can use `frames describe` to get a summary of the data in each frame:

```
. frames describe
```

```
Frame: census
Contains data from C:\Program Files\Stata18\ado\base\c\census.dta
Observations:      50          1980 Census data by state
Variables:         13          23 Mar 2023 16:16
```

Variable name	Storage type	Display format	Value label	Variable label
<code>state</code>	<code>str14</code>	<code>%-14s</code>		State
<code>state2</code>	<code>str2</code>	<code>%-2s</code>		Two-letter state abbreviation
<code>region</code>	<code>int</code>	<code>%-8.0g</code>	<code>cenreg</code>	Census region
<code>pop</code>	<code>long</code>	<code>%12.0gc</code>		Population
<code>pop1t5</code>	<code>long</code>	<code>%12.0gc</code>		Pop, < 5 year
<code>pop5_17</code>	<code>long</code>	<code>%12.0gc</code>		Pop, 5 to 17 years
<code>pop18p</code>	<code>long</code>	<code>%12.0gc</code>		Pop, 18 and older
<code>pop65p</code>	<code>long</code>	<code>%12.0gc</code>		Pop, 65 and older
<code>popurban</code>	<code>long</code>	<code>%12.0gc</code>		Urban population
<code>medage</code>	<code>float</code>	<code>%9.2f</code>		Median age
<code>death</code>	<code>long</code>	<code>%12.0gc</code>		Number of deaths
<code>marriage</code>	<code>long</code>	<code>%12.0gc</code>		Number of marriages
<code>divorce</code>	<code>long</code>	<code>%12.0gc</code>		Number of divorces

Sorted by: state

Frame: default

Contains data

Observations: 0

Variables: 0

Sorted by:

Frame: housing

Contains data from <https://www.stata-press.com/data/r18/hsng.dta>

Observations: 50 1980 Census housing data

Variables: 13 23 Mar 2023 16:16

Variable name	Storage type	Display format	Value label	Variable label
state	str14	%14s		State
division	int	%8.0g	division	Census division
region	int	%8.0g	region	Census region
pop	long	%10.0g		Population in 1980
popgrow	float	%6.1f		Pop. growth 1970-80
popden	int	%6.1f		Pop/sq. mile
pcturban	float	%8.1f		Percent urban
faminc	long	%8.2f		Median family inc., 1979
hsng	long	%10.0g		Hsng units 1980
hsnggrow	float	%8.1f		% housing growth
hsngval	long	%9.2f		Median hsng value
rent	long	%6.2f		Median gross rent
census	byte	%10.0g		

Sorted by: state

Note: Dataset has changed since last saved.

We can see that frame `housing` has a variable named `census`; this is the variable that `frlink` created to store the information needed to link the frames. We can also see that the `default` frame is empty because we have not loaded a dataset into that frame.

We can now save frame `housing` and all frames linked to it by typing the following:

```
. frames save myframeset, frames(housing) linked replace
file myframeset.dtas saved
```

This saves frame `housing`, as well as frame `census`, because it is linked to frame `housing`. The `replace` option replaces file `myframeset.dtas` if it already exists.

We now drop frame `census` using `frame drop`:

```
. frame drop census
```

Note that if we try to save frame `housing` and the frames linked to it, we get an error message:

```
. frames save myframeset, frames(housing) linked replace
linked frame does not exist
Frame census is linked from frame housing, but frame census does not
exist. Use option relaxed if you wish to ignore this error and proceed
anyway.
r(111);
```

Stata is attempting to save frame `census` because it is linked to frame `housing`, but it does not exist. To save the frames we specified, and any existing frames linked to them, we can use the `relaxed` option to ignore any linked frame that does not exist:

```
. frames save datasets, frames(housing) linked replace relaxed
(file datasets.dtas not found)
file datasets.dtas saved
```

We no longer get an error message, but because frame `census` does not exist, only frame `housing` gets saved.

◀

Stored results

`frames save` stores the following in `r()`:

Scalars

<code>r(complevel)</code>	compression level
<code>r(compsize)</code>	size, in bytes, of compressed file
<code>r(compratio)</code>	compression ratio, defined as the ratio of compressed size to uncompressed size

Macros

<code>r(fn)</code>	pathname of saved frameset file
<code>r(frames)</code>	list of frames saved, listed in the same order as in option <code>frames()</code> ; if <code>frames(_all)</code> is used, then the working frame is listed first, followed by the remaining frames in alphabetical order
<code>r(first)</code>	first frame in <code>r(frames)</code>

Also see

[D] [frames describe](#) — Describe frames in memory or in a file

[D] [frames use](#) — Load a set of frames from disk

[D] [frames](#) — Data frames

[D] [save](#) — Save Stata dataset