

snapshot — Save and restore data snapshots[Syntax](#)[Remarks and examples](#)[Menu](#)[Stored results](#)[Description](#)[Also see](#)[Option](#)

Syntax

Save snapshot

```
snapshot save [ , label("label") ]
```

Change snapshot label

```
snapshot label snapshot# "label"
```

Restore snapshot

```
snapshot restore snapshot#
```

List snapshots

```
snapshot list [ _all | numlist ]
```

Erase snapshots

```
snapshot erase _all | numlist
```

Menu

Data > Data Editor > Data Editor (Edit)

Description

`snapshot` saves to disk and restores from disk copies of the data in memory. `snapshot`'s main purpose is to allow the Data Editor to save and restore data snapshots during an interactive editing session. A more popular alternative for programmers is `preserve`; see [\[P\] preserve](#).

Snapshots are referred to by a *snapshot#*. If no snapshots currently exist, the next snapshot saved will receive a *snapshot#* of 1. If snapshots do exist, the next snapshot saved will receive a *snapshot#* one greater than the highest existing *snapshot#*.

`snapshot save` creates a temporary file containing a copy of the data currently in memory and attaches an optional label (up to 80 characters) to the saved snapshot. Up to 1,000 snapshots may be saved.

`snapshot label` changes the label on the specified snapshot.

`snapshot restore` replaces the data in memory with the data from the specified snapshot.

`snapshot list` lists specified snapshots.

`snapshot erase` erases specified snapshots.

Option

`label(label)` is for use with `snapshot save` and allows you to label a snapshot when saving it.

Remarks and examples

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

`snapshot` was created to allow a user using the Data Editor to save and restore snapshots of their data while editing them interactively. It is similar to a checkpoint save in a video game, where after you have made a certain amount of progress, you wish to make sure you will be able to return to that point no matter what may happen in the future.

`snapshot` does not overwrite any copies of your data that you may have saved to disk. It saves a copy of the data currently in memory to a temporary file and allows you to later restore that copy to memory.

`snapshot` saves the date and time at which you create a snapshot. It is a good idea to also give a snapshot a label so that you will be better able to distinguish between multiple snapshots should you need to restore one.

□ Technical note

Although we mention above the use of the Data Editor and we demonstrate below the use of `snapshot`, we recommend that data cleaning not be done interactively. Instead, we recommend that data editing and cleaning be done in a reproducible manner through the use of do-files; see [\[U\] 16 Do-files](#).



▷ Example 1

You decide to make some changes to the `auto` dataset. You make a snapshot of the data before you begin making changes, and you make another snapshot after the changes:

```
. use http://www.stata-press.com/data/r13/auto
(1978 Automobile Data)
. snapshot save, label("before changes")
snapshot 1 (before changes) created at 19 Apr 2013 21:32
. generate gpm = 1/mpg
. label variable gpm "Gallons per mile"
. snapshot save, label("after changes")
snapshot 2 (after changes) created at 19 Apr 2013 21:34
```

You go on to do some analyses, but then, for some reason, you accidentally drop the variable you previously created:

```
. drop gpm
```

Luckily, you made some snapshots of your work:

```
. snapshot list
snapshot 1 (before changes) created at 19 Apr 2013 21:32
snapshot 2 (after changes) created at 19 Apr 2013 21:34
. snapshot restore 2
```

```
. describe gpm
```

variable name	storage type	display format	value label	variable label
gpm	float	%9.0g		Gallons per mile

4

Stored results

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

Scalars

`r(snapshot)` sequence number of snapshot saved

Also see

[D] [edit](#) — Browse or edit data with Data Editor

[P] [preserve](#) — Preserve and restore data