

*saving\_option* — Option for saving graph to disk

[Description](#)   [Quick start](#)   [Syntax](#)   [Option](#)   [Remarks and examples](#)   [Also see](#)

## Description

Option `saving()` saves the graph to disk.

## Quick start

Save the graph to `mygraph.gph` in the working directory

```
graph_command ..., ... saving(mygraph)
```

As above, and replace `mygraph.gph` if it already exists

```
graph_command ..., ... saving(mygraph, replace)
```

Save to a specific directory using a fully qualified path name

```
graph_command ..., ... saving("my directory/mygraph")
```

Allow the directory name to include " or ' using compound quotes

```
graph_command ..., ... saving("my "little" directory/mygraph")
```

## Syntax

<i>saving_option</i>	Description
<code>saving(<i>filename</i> [, <i>suboptions</i>])</code>	save graph to disk
<code>saving()</code> is <i>unique</i> ; see [G-4] <b>concept: repeated options</b> .	

<i>suboptions</i>	Description
<code>asis</code>	freeze graph and save as is
<code>replace</code>	okay to replace existing <i>filename</i>

## Option

`saving(filename [, suboptions])` specifies the name of the diskfile to be created or replaced. If *filename* is specified without an extension, `.gph` will be assumed.

## Suboptions

`asis` specifies that the graph be frozen and saved just as it is. The alternative—and the default if `asis` is not specified—is known as *live format*. In live format, the graph can continue to be edited in future sessions, and the overall look of the graph continues to be controlled by the chosen scheme (see [G-4] [schemes intro](#)).

Say that you type

```
. scatter yvar xvar, ... saving(mygraph)
```

That will create file `mygraph.gph`. Now pretend you send that file to a colleague. The way the graph appears on your colleague's computer might be different from how it appears on yours. Perhaps you display titles on the top and your colleague has set his scheme to display titles on the bottom. Or perhaps your colleague prefers the *y* axis on the right rather than the left. It will still be the same graph, but it might have a different look.

Or perhaps you just file away `mygraph.gph` for use later. If you store it in the default live format, you can come back to it later and change the way it looks by specifying a different scheme or can edit it.

If, on the other hand, you specify `asis`, the graph will look forever just as it looked the instant it was saved. You cannot edit it; you cannot change the scheme. If you send the as-is graph to colleagues, they will see it in exactly the form you see it.

Whether a graph is saved as-is or live makes no difference for printing. As-is graphs usually require fewer bytes to store, and they generally display more quickly, but that is all.

`replace` specifies that the file may be replaced if it already exists.

## Remarks and examples

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

To save a graph permanently, you add `saving()` to the end of the `graph` command (or any place among the options):

```
. graph ... , ... saving(myfile) ...  
(file myfile.gph saved)
```

You can also achieve the same result in two steps:

```
. graph ... , ...  
. graph save myfile  
(file myfile.gph saved)
```

The advantage of the two-part construction is that you can edit the graph between the time you first draw it and save it. The advantage of the one-part construction is that you will not forget to save it.

## Also see

[G-2] [graph export](#) — Export current graph

[G-2] [graph manipulation](#) — Graph manipulation commands

[G-2] [graph save](#) — Save graph to disk

[G-4] [concept: gph files](#) — Using gph files