

## Description

Option `name()` specifies the name of the graph being created.

## Syntax

<i>name_option</i>	Description
<code>name(<i>name</i>[ , <i>replace</i>])</code>	specify name
<code>name()</code> is <i>unique</i> ; see [G-4] <b>Concept: repeated options</b> .	

## Option

`name(name[ , replace])` specifies the name of the graph. If `name()` is not specified, `name(Graph, replace)` is assumed.

In fact, `name(Graph)` has the same effect as `name(Graph, replace)` because *replace* is assumed when the name is *Graph*. For all other *names*, you must specify suboption *replace* if a graph under that name already exists.

## Remarks and examples

When you type, for instance,

```
. scatter yvar xvar
```

you see a graph. The graph is also stored in memory. For instance, try the following: close the Graph window, and then type

```
. graph display
```

Your graph will reappear.

Every time you draw a graph, that previously remembered graph is discarded, and the new graph replaces it.

You can have more than one graph stored in memory. When you do not specify the name under which the graph is to be remembered, it is remembered under the default name *Graph*. For instance, if you were now to type

```
. scatter y2var xvar, name(g2)
```

You would now have two graphs stored in memory: Graph and g2. If you typed

```
. graph display
```

or

```
. graph display Graph
```

you would see your first graph. Type

```
. graph display g2
```

and you will see your second graph.

Do not confuse Stata's storing of graphs in memory with the saving of graphs to disk. Were you now to exit Stata, the graphs you have stored in memory would be gone forever. If you want to save your graphs, you want to specify the `saving()` option (see [G-3] [saving\\_option](#)) or you want to use the `graph save` command (see [G-2] [graph save](#)); either result in the same outcome.

You can find out what graphs you have in memory by using `graph dir`, drop them by using `graph drop`, rename them by using `graph rename`, and so on, and of course, you can redisplay them by using `graph display`. See [G-2] [graph manipulation](#) for the details on all of those commands.

You can drop all graphs currently stored in memory by using `graph drop _all` or `discard`; see [G-2] [graph drop](#).

## Also see

[G-2] [graph display](#) — Display graph stored in memory

[G-2] [graph drop](#) — Drop graphs from memory

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

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

[G-3] [saving\\_option](#) — Option for saving graph to disk

