

graph use — Display graph stored on disk

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

Description

`graph use` displays (draws) the graph previously saved in a `.gph` file and, if the graph was stored in live format, loads it.

If *filename* is specified without an extension, `.gph` is assumed.

Quick start

Display a graph saved on disk as `mygraph.gph`

```
graph use mygraph
```

As above, and name the graph “MyGraph”

```
graph use mygraph, name(MyGraph)
```

Display saved graph with the overall look defined by the monochromatic Stata manual scheme `s2manual`

```
graph use mygraph, scheme(s2manual)
```

Syntax

```
graph use filename [ , options ]
```

| <i>options</i> | Description |
|-----------------------------------------------------|--------------------------------------|
| <code>nodraw</code> | do not draw the graph |
| <code>name(<i>name</i> [, <i>replace</i>])</code> | specify new name for graph |
| <code><u>s</u>cheme(<i>schemename</i>)</code> | overall look |
| <code><u>p</u>lay(<i>recordingname</i>)</code> | play edits from <i>recordingname</i> |

Options

`nodraw` specifies that the graph not be displayed. If the graph was stored in live format, it is still loaded; otherwise, `graph use` does nothing. See [G-3] *nodraw_option*.

`name(name [, replace])` specifies the name under which the graph is to be stored in memory, assuming that the graph was saved in live format. *filename* is the default name, where any path component in *filename* is excluded. For example,

```
. graph use mydir\mygraph.gph
```

will draw a graph with the name `mygraph`.

If the default name already exists `graph#` is used instead, where `#` is chosen to create a unique name.

If the graph is not stored in live format, the graph can only be displayed, not loaded, and the `name()` is irrelevant.

`scheme(schemename)` specifies the scheme controlling the overall look of the graph to be used; see [G-3] [scheme_option](#). If `scheme()` is not specified, the default is the *schemename* recorded in the graph being loaded.

`play(recordingname)` applies the edits from *recordingname* to the graph, where *recordingname* is the name under which edits previously made in the Graph Editor have been recorded and stored. See [Graph Recorder](#) in [G-1] [Graph Editor](#).

Remarks and examples

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

Graphs can be saved at the time you draw them either by specifying the `saving()` option or by subsequently using the `graph save` command; see [G-3] [saving_option](#) and [G-2] [graph save](#). Modern graphs are saved in live format or as-is format; see [G-4] [Concept: gph files](#). Regardless of how the graph was saved or the format in which it was saved, `graph use` can redisplay the graph; simply type

```
. graph use filename
```

In a prior session, you drew a graph by typing

```
. twoway qfitci mpg weight, stdf ||  
      scatter mpg weight      ||  
      , by(foreign, total row(1)) saving(cigraph)
```

The result of this was to create file `cigraph.gph`. At a later date, you can see the contents of the file by typing

```
. graph use cigraph
```

You might now edit the graph (see [G-1] [Graph Editor](#)), or print a copy of the graph.

Also see

[G-2] [graph combine](#) — Combine multiple graphs

[G-2] [graph replay](#) — Replay multiple graphs

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

[G-3] [name_option](#) — Option for naming graph in memory

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

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