Title

stata.com

concept: repeated options — Interpretation of repeated options

Description Remarks and examples Also see

Description

Options allowed with `graph` are categorized as being

unique
rightmost
merged-implicit
merged-explicit

What this means is described below.

Remarks and examples

stata.com

It may surprise you to learn that most `graph` options can be repeated within the same `graph` command. For instance, you can type

```
. graph twoway scatter mpg weight, msymbol(Oh) msymbol(O)
```

and rather then getting an error, you will get back the same graph as if you omitted typing the `msymbol(Oh)` option. `msymbol()` is said to be a *rightmost* option.

`graph` allows that because so many other commands are implemented in terms of `graph`. Imagine that an ado-file that constructs the "`scatter mpg weight, msymbol(Oh)`" part, and you come along and use that ado-file, and you specify to it the option "`msymbol(O)`". The result is that the ado-file constructs

```
. graph twoway scatter mpg weight, msymbol(Oh) msymbol(O)
```

and, because `graph` is willing to ignore all but the rightmost specification of the `msymbol()` option, the command works and does what you expect.

Options in fact come in three forms, which are

1. *rightmost*: take the rightmost occurrence;
2. *merged*: merge the repeated instances together;
3. *unique*: the option may be specified only once; specifying it more than once is an error.

You will always find options categorized one of these three ways; typically that is done in the syntax diagram, but sometimes the categorization appears in the description of the option.

`msymbol()` is an example of a *rightmost* option. An example of a *unique* option is `saving()`; it may be specified only once.

Concerning *merged* options, they are broken into two subcategories:

2a. *merged-implicit*: always merge repeated instances together,

2b. *merged-explicit*: treat as *rightmost* unless an option within the option is specified, in which case it is merged.

`merged` can apply only to options that take arguments because otherwise there would be nothing to merge. Sometimes those options themselves take suboptions. For instance, the syntax of the `title()` option (the option that puts titles on the graph) is

`title("`*string*`" [ "`*string*`" [ ... ] ] [ , `*suboptions*` ])`

`title()` has suboptions that specify how the title is to look and among them is, for instance, `color()`; see [G-3] ***title_options***. `title()` also has two other suboptions, `prefix` and `suffix`, that specify how repeated instances of the `title()` option are to be merged. For instance, specify

```
...title("My title") ...title("Second line", suffix)
```

and the result will be the same as if you specified

```
...title("My title" "Second line")
```

at the outset. Specify

```
...title("My title") ...title("New line", prefix)
```

and the result will be the same as if you specified

```
...title("New line" "My title")
```

at the outset. The `prefix` and `suffix` options specify exactly how repeated instances of the option are to be merged. If you do not specify one of those options,

```
...title("My title") ...title("New title")
```

the result will be as if you never specified the first option:

```
...title("New title")
```

`title()` is an example of a *merged-explicit* option. The suboption names for handling *merged-explicit* are not always `prefix` and `suffix`, but anytime an option is designated *merged-explicit*, it will be documented under the heading *Interpretation of repeated options* exactly what and how the merge options work.

❑ Technical note

Even when an option is *merged-explicit* and its merge suboptions are not specified, its other suboptions are merged. For instance, consider

```
...title("My title", color(red)) ...title("New title")
```

`title()` is *merged-explicit*, but because we did not specify one of its merge options, it is being treated as *rightmost*. Actually, it is almost being treated as rightmost because, rather than the `title()` being exactly what we typed, it will be

```
...title("New title", color(red))
```

This makes ado-files work as you would expect. Say that you run the ado-file `xyz.ado`, which constructs some graph and the command

```
graph ..., ...title("Std. title", color(red)) ...
```

You specify an option to `xyz.ado` to change the title:

```
. xyz ..., ...title("My title")
```

The overall result will be just as you expect: your title will be substituted, but the color of the title (and its size, position, etc.) will not change. If you wanted to change those things, you would have specified the appropriate suboptions in your `title()` option. ❑

Also see

[G-2] **graph** — The graph command