Title

stata.com

mata mosave — Save function's compiled code in object file

Description Syntax Options Remarks and examples Also see

Description

`mata mosave` saves the object code for the specified function in the file *fcnname*`.mo`.

Syntax

```
: mata mosave fcnname() [ , dir(path) complete replace ]
```

This command is for use in Mata mode following Mata's colon prompt. To use this command from Stata's dot prompt, type

```
. mata: mata mosave ...
```

Options

`dir(`*path*`)` specifies the directory (folder) into which the file should be written. `dir(.)` is the default, meaning that if `dir()` is not specified, the file is written into the current (working) directory. *path* may be a directory name or may be the sysdir shorthand `STATA`, `BASE`, `SITE`, `PLUS`, `PERSONAL`, or `OLDPLACE`; see [P] **sysdir**. `dir(PERSONAL)` is recommended.

`complete` is for use when saving class definitions. It specifies that the definition be saved only if it is complete; otherwise, an error message is to be issued. See [M-2] **class**.

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

Remarks and examples

stata.com

See [M-1] **How** for an explanation of object code.

Remarks are presented under the following headings:

Example of use

`.mo` files contain the object code for one function. If you store a function's object code in a `.mo` file, then in future Mata sessions, you can use the function without recompiling the source. The function will appear to become a part of Mata just as all the other functions documented in this manual are. The function can be used because the object code will be automatically found and loaded when needed.

For example,

```
: function add(a,b) return(a+b)
: add(1,2)
  3
```

```
: mata mosave add()
(file add.mo created)
: mata clear
: add(1,2)
  3
```

In the example above, function `add()` was saved in file `add.mo` stored in the current directory. After clearing Mata, we could still use the function because Mata found the stored object code.

Where to store .mo files

Mata could find `add()` because file `add.mo` was in the current directory, and our ado-path included `.`:

```
. adopath
  [1]  (BASE)      "C:\Program Files\Stata17\ado\base\"
  [2]  (SITE)      "C:\Program Files\Stata17\ado\site\"
  [3]              "."
  [4]  (PERSONAL)  "C:\ado\personal\"
  [5]  (PLUS)      "C:\ado\plus\"
  [6]  (OLDPLACE)  "C:\ado\"
```

If later we were to change our current directory

```
. cd ..\otherdir
```

Mata would no longer be able to find the file `add.mo`. Thus the best place to store your personal `.mo` files is in your `PERSONAL` directory. Thus rather than typing

```
: mata mosave example()
```

we would have been better off typing

```
: mata mosave example(), dir(PERSONAL)
```

Use of .mo files versus .mlib files

Use of `.mo` files is heartily recommended. The alternative for saving compiled object code are `.mlib` libraries; see [M-3] **mata mlib** and [M-1] **Ado**.

Libraries are useful when you have many functions and want to tie them together into one file, especially if you want to share those functions with others, because then you have only one file to distribute. The disadvantage of libraries is that you must rebuild them whenever you wish to remove or change the code of one. If you have only a few object files, or if you have many but sharing is not an issue, `.mo` libraries are easier to manage.

Also see

[M-3] **mata mlib** — Create function library

[M-3] **Intro** — Commands for controlling Mata