

mata set — Set and display Mata system parameters[Description](#)[Option](#)[Syntax](#)[Remarks and examples](#)[Also see](#)

Description

`mata query` shows the values of Mata's system parameters.

`mata set` sets the value of the system parameters:

`mata set matabase` specifies the maximum amount of memory, in kilobytes, that may be consumed before Mata starts looking to drop autoloading functions that are not currently being used. The default value is 2000, meaning 2000 kilobytes. This parameter affects the efficiency with which Stata runs. Larger values cannot hurt, but once `matabase` is large enough, larger values will not improve performance.

`mata set mataalnum` turns program line-number tracing on or off. The default setting is off. This setting modifies how programs are compiled. Programs compiled when `mataalnum` is turned on include code so that, if an error occurs during execution of the program, the line number is also reported. Turning `mataalnum` on prevents Mata from being able to optimize programs, so they will run more slowly. Except when debugging, the recommended setting for this is off.

`mata set mataoptimize` turns compile-time code optimization on or off. The default setting is on. Programs compiled when `mataoptimize` is switched off will run more slowly and, sometimes, much more slowly. The only reason to set `mataoptimize` off is if a bug in the optimizer is suspected.

`mata set matafavor` specifies whether, when executing code, Mata should favor conserving memory (`space`) or running quickly (`speed`). The default setting is `space`. Switching to `speed` will make Mata, in a few instances, run a little quicker but consume more memory. Also see [\[M-5\] favorspeed\(\)](#).

`mata set matastrict` sets whether declarations can be omitted inside the body of a program. The default is off. If `matastrict` is switched on, compiling programs that omit the declarations will result in a compile-time error; see [\[M-2\] declarations](#). `matastrict` acts unexpectedly but pleasingly when set/reset inside ado-files; see [\[M-1\] ado](#).

`mata set matalibs` sets the names and order of the `.mlib` libraries to be searched; see [\[M-1\] how](#). `matalibs` usually is set to `"lmatbase;lmatado"`. However it is set, it is probably set correctly, because Mata automatically searches for libraries the first time it is invoked in a Stata session. If, during a session, you erase or copy new libraries along the `ado-path`, the best way to reset `matalibs` is with the `mata mlib index` command; see [\[M-3\] mata mlib](#). The only reason to set `matalibs` by hand is to modify the order in which libraries are searched.

`mata set matamofirst` states whether `.mo` files or `.mlib` libraries are searched first. The default is off, meaning libraries are searched first.

Option

permanently specifies that, in addition to making the change right now, the setting be remembered and become the default setting when you invoke Stata in the future.

Syntax

```
: mata query

: mata set matacache      # [ , permanently ]
: mata set matalnum      { off | on }
: mata set mataoptimize  { on | off }
: mata set matafavor     { space | speed } [ , permanently ]
: mata set matastrict    { off | on } [ , permanently ]
: mata set matalibs      "libname;libname;..."
: mata set matamofirst   { off | on } [ , permanently ]
```

These commands are for use in Mata mode following Mata's colon prompt. To use these commands from Stata's dot prompt, type

```
. mata: mata query
. mata: mata set ...
```

Remarks and examples

stata.com

Remarks are presented under the following headings:

*Relationship between Mata's mata set and Stata's set commands
c() values*

Relationship between Mata's mata set and Stata's set commands

The command

```
: mata set ...
```

issued from Mata's colon prompt and the command

```
. set ...
```

issued from Stata's dot prompt are the same command, so you may set Mata's (or even Stata's) system parameters either way.

The command

```
: mata query
```

issued from Mata's colon prompt and the command

```
. query mata
```

issued from Stata's dot prompt are also the same command.

c() values

The following concerns Stata more than Mata.

Stata's c-class, `c()`, contains the values of system parameters and settings along with certain other constants. `c()` values may be referred to in Stata, either via macro substitution (`'c(current_date)'`, for example) or in expressions (in which case the macro quoting characters may be omitted). Stata's `c()` is also available in Mata via Mata's `c()` function; see [M-5] [c\(\)](#).

Most everything set by `set` is available via `c()`, including Mata's set parameters:

`c(matacache)` returns a numeric scalar equal to the cache size.

`c(matalnum)` returns a string equal to "on" or "off".

`c(mataoptimize)` returns a string equal to "on" or "off".

`c(matafavor)` returns a string equal to "space" or "speed".

`c(matastrict)` returns a string equal to "on" or "off".

`c(matalibs)` returns a string of library names separated by semicolons.

`c(matamofirst)` returns a string equal to "on" or "off".

The above is in Stata speak. Rather than referring to `c(matacache)`, we would refer to `c("matacache")` if we were using Mata's function. The real use of these values, however, is in Stata.

Also see

[R] [query](#) — Display system parameters

[R] [set](#) — Overview of system parameters

[P] [creturn](#) — Return c-class values

[M-3] [intro](#) — Commands for controlling Mata