Title

stata.com

more( ) — Create –more– condition

Description Syntax Remarks and examples Conformability
Diagnostics Also see

Description

`more()` displays —`more`— and waits for a key to be pressed. That is, `more()` does that if `more` is on, and it does nothing otherwise. `more` can be turned on and off by Stata's `set more` command (see [R] **more**) or by the functions below.

`setmore()` returns whether `more` is on or off, encoded as 1 and 0.

`setmore(`*onoff*`)` sets `more` on if *onoff* $\neq 0$ and sets `more` off otherwise.

`setmoreonexit(`*onoff*`)` sets `more` on or off when the current execution ends. It has no effect on the current setting. The specified setting will take effect when control is passed back to the Mata prompt or to the calling ado-file or to Stata itself, and it will take effect regardless of whether execution ended because of a `return`, `exit()`, error, or abort. Only the first call to `setmoreonexit()` has that effect. Later calls have no effect whatsoever.

Syntax

void `more()`

real scalar `setmore()`

void `setmore(`*real scalar onoff*`)`

void `setmoreonexit(`*real scalar onoff*`)`

Remarks and examples

stata.com

`setmoreonexit()` is used to ensure that the `more` setting is restored if a program wants to temporarily reset it:

```
setmoreonexit(setmore())
setmore(0)
```

Only the first invocation of `setmoreonexit()` has any effect. This way, a subroutine that is used in various contexts might also contain

```
setmoreonexit(setmore())
setmore(0)
```

and that will not cause the wrong `more` setting to be restored if an earlier routine had already done that and yet still cause the right setting to be restored if the subroutine is the first to issue `setmoreonexit()`.

Conformability

`more()` takes no arguments and returns *void*.

`setmore()`:

	result:	1 × 1

`setmore(`*onoff*`)`, `setmoreonexit(`*onoff*`)`:

	onoff:	1 × 1
	result:	*void*

Diagnostics

None.

Also see

[P] **more** — Pause until key is pressed

[M-4] **io** — I/O functions