Title

stata.com

pause — Program debugging command

Description Syntax Remarks and examples Reference Also see

Description

If pause is on, the `pause` [*message*] command displays *message* and temporarily suspends execution of the program, returning control to the keyboard. Execution of keyboard commands continues until you type `end` or `q`, at which time execution of the program resumes. Typing `BREAK` in pause mode (as opposed to pressing the *Break* key) also resumes program execution, but the break signal is sent to the calling program.

If pause is off, `pause` does nothing.

Pause is off by default. Type `pause on` to turn pause on. Type `pause off` to turn it back off.

Syntax

`pause` { `on` | `off` | [*message*] }

Remarks and examples

stata.com

`pause` assists in debugging Stata programs. The line `pause` or `pause` *message* is placed in the program where problems are suspected (more than one `pause` may be placed in a program). For instance, you have a program that is not working properly. A piece of this program reads

```
generate 'tmp'=exp('1')/'2'
summarize 'tmp'
local mean=r(mean)
```

You think that the error may be in the creation of `'tmp'`. You change the program to read

```
generate 'tmp'=exp('1')/'2'
pause Just created tmp        /* this line is new */
summarize 'tmp'
local mean=r(mean)
```

Let's pretend that your program is named `myprog`; interactively, you now type

```
. myprog
```
(*output from your program appears*)

That is, `pause` does nothing because pause is off, so `pause`s in your program are ignored. If you turn pause on,

```
. pause on
. myprog
```
(*any output* `myprog` *creates up to the* `pause` *appears*)
```
pause:  Just created tmp
-> . describe
```
  (*output omitted*)
```
-> . list
```
  (*output omitted*)

```
-> . end
execution resumes...
(remaining output from myprog appears)
```

The "->" is called the pause-mode prompt. You can give any Stata command. You can examine variables and, if you wish, even change them. If while in pause mode, you wish to terminate execution of your program, you type BREAK (in capitals):

```
. myprog
(any output myprog creates up to the pause appears)
pause:  Just created tmp
-> . list
  (output omitted)
-> . BREAK
sending Break to calling program...
--Break--
r(1);

. _
```

The results are the same as if you pressed *Break* while your program was executing. If you press the *Break* key in pause mode (as opposed to typing BREAK), however, it means only that the execution of the command you have just given interactively is to be interrupted.

Notes:

- You may put many pauses in your programs.
- By default, pause is off, so the pauses will not do anything. Even so, you should remove the pauses after your program is debugged because each execution of a do-nothing pause will slow your program slightly.
- pause is implemented as an ado-file; this means that the definitions of local macros in your program are unavailable to you. To see the value of local macros, display them in the pause message; for instance,

  ```
  pause Just created tmp, i=`i'
  ```

  When the line is executed, you will see something like

  ```
  pause:  Just created tmp, i=1
  -> . _
  ```

- Remember, temporary variables (for example, tempvar tmp ... gen `tmp'=...) are assigned real names, such as __00424, by Stata; see [P] **macro**. Thus, in pause mode, you want to examine __00424 and not tmp. Generally, you can determine the real name of your temporary variables from describe's output, but in the example above, it would have been better if pause had been invoked with

  ```
  pause Just created tmp, called `tmp', i=`i'
  ```

  When the line was executed, you would have seen something like

  ```
  pause:  Just created tmp, called __00424, i=1
  -> . _
  ```

- When giving commands that include double quotes, you may occasionally see the error message "type mismatch", but then the command will work properly:

  ```
  pause:  Just created tmp, called __00424, i=1
  -> . list if __00424=="male"
  type mismatch
  (output from request appears as if nothing is wrong)
  -> . _
  ```

Reference

Becketti, S. 1993. ip4: Program debugging command. *Stata Technical Bulletin* 13: 13–14. Reprinted in *Stata Technical Bulletin Reprints*, vol. 3, pp. 57–58. College Station, TX: Stata Press.

Also see

[P] **program** — Define and manipulate programs

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

[P] **trace** — Debug Stata programs

[U] 18 Programming Stata