#delimit — Change delimiter

Description Syntax Remarks and examples Also see

Description

The `#delimit` command resets the character that marks the end of a command. It can be used only in do-files or ado-files.

Syntax

`#delimit { cr | ; }`

Remarks and examples

`#delimit` (pronounced *pound-delimit*) is a Stata preprocessor command. `#`*commands* do not generate a return code, nor do they generate ordinary Stata errors. The only error message associated with `#`*commands* is "unrecognized #command".

Commands given from the console are always executed when you press the *Enter*, or *Return*, key. `#delimit` cannot be used interactively, so you cannot change Stata's interactive behavior.

Commands in a do-file, however, may be delimited with a carriage return or a semicolon. When a do-file begins, the delimiter is a carriage return. The command '`#delimit ;`' changes the delimiter to a semicolon. To restore the carriage return delimiter inside a file, use `#delimit cr`.

When a do-file begins execution, the delimiter is automatically set to carriage return, even if it was called from another do-file that set the delimiter to semicolon. Also, the current do-file need not worry about restoring the delimiter to what it was because Stata will do that automatically.

▷ Example 1

```
/*
     When the do-file begins, the delimiter is carriage return:
*/
use basedata, clear

/*
     The last command loaded our data.
     Let's now change the delimiter:
*/
#delimit ;
summarize sex
               salary ;
/*
     Because the delimiter is semicolon, it does not matter that our
     command took two lines.
     We can change the delimiter back:
*/

#delimit cr
summarize sex salary

/*
     Now our lines once again end on return.  The semicolon delimiter
     is often used when loading programs:
*/

capture program drop fix
program fix
     confirm var `1'
     #delimit ;
     replace `1' = .  if salary>=. | salary==0 |
                         hours>=.  | hours==0 ;
     #delimit cr
end

fix var1
fix var2
```

◁

❑ Technical note

Just because you have long lines does not mean that you must change the delimiter to semicolon. Stata does not care that the line is long. There are also other ways to indicate that more than one physical line is one logical line. One popular choice is `///`:

```
     replace `1' = .  if salary>=. | salary==0 | ///
                         hours>=.  | hours==0
```

See [P] **comments**.

❑

Also see

[P] **comments** — Add comments to programs

[U] 16.1.3 Long lines in do-files

[U] 18.11.2 Comments and long lines in ado-files

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on citing Stata documentation.