

# C Stata manual pages for Unix

## Contents

- [conren](#) — Set the color, etc., of Stata(console)
- [stata](#) — Stata invocation command
- [pystata](#) — Call Stata from Python

**conren** — Set the color, etc., of Stata(console)

## Syntax

### *High-level commands*

```
conren  
  
conren style #  
  
conren ul #  
  
conren test  
  
conren clear
```

### *Low-level commands*

```
set conren  
  
set conren clear  
  
set conren [ sf | bf | it ] { result | { txt | text } | input | error |  
    link | hilite } [ char [ char ... ]]  
  
set conren { ulon | uloff } [ char [ char ... ]]  
  
set conren reset [ char [ char ... ]]  
  
set conren off [ char [ char ... ]]
```

where *char* is

```
{ any_character | < # > }
```

## Note

This command concerns Stata for Unix only and, in particular, the Stata you launch by typing `stata` or `stata-se`, not `xstata` or `xstata-se`, also known as Stata(console) or the non-GUI version of Stata.

## Description

`conren` and `set conren` may improve display of the output on your screen. Some terminals, for instance, can display colors, but Stata may not know that your terminal has that capability. Some terminals have multiple intensities or boldfaces. Some terminals can underline. The high-level `conren` command lets you set a display style, underlining scheme, or both from among a selection of predefined settings.

`conren style` followed by a scheme number sets color and font codes on the basis of the underlying scheme.

`conren ul` followed by an underlining scheme number sets the codes that allow underlining.

`conren` with no arguments displays a message explaining the command and telling the range of style and underlining scheme numbers available.

`conren test` displays three columns of output in `sf` (standard face) font, **bf (boldface) font**, and *it (italics) font*, showing the assignment of colors with and without underlining.

`conren clear` clears all the currently defined display style and underlining definitions.

The low-level `set conren` command lets you view and set specific display and underlining codes.

`set conren` displays a list of the currently defined display codes.

`set conren clear` clears all codes.

`set conren` followed by a font type (`bf`, `sf`, or `it`), a display context (`result`, `text`, `input`, `error`, `link`, or `hilite`) and a series of space-separated characters sets the code for the specified font type and display context. If the font type is omitted, the code is set to the same specified code for all three font types.

`set conren ulon` and `set conren uloff` set the codes for turning underlining on and off.

`set conren reset` sets the code that will turn off all display and underlining codes.

`set conren off` sets the code used by Stata when it exits and returns control back to the operating system.

When Stata launches, it is as if you have typed

```
. conren clear
```

which is equivalent to the low-level command

```
. set conren clear
```

meaning that Stata is to assume that your monitor cannot display different colors, intensities, or underlining. Stata makes this assumption because, were Stata to assume otherwise and your terminal could not provide the capability, the result could look really bad. Thus, a few minutes of playing around can be well worth the effort, and you do not have to be a computer expert to do set these codes. You cannot hurt anything permanently by typing the wrong command.

The next-to-worst thing that can happen is that Stata's output will look so bad that you cannot even read it, and then just exit Stata. Stata will be fine the next time you launch it.

The worst thing that can happen is that your window/screen/terminal will look so garbled that you will have to close it and open a new one (or, if it really is a separate terminal, turn it off and turn it back on).

Once you are happy with your settings, you can put the `set conren` commands in your `profile.do` so that they are executed at the start of every Stata session.

## Finding a color scheme

First, let's try various color schemes. What will work and look good depends on your terminal/monitor and whether you are using a white or black background. (We really prefer a black background for Stata, and if you are using a white background, we recommend that you try black someday. We prefer a black background for Stata(console) because, by default, it uses green and yellow for most output, and these colors do not show up well on a light-colored background. Switching the background color, however, is something that you will have to take up with Unix, not Stata.)

First, type the following:

```
. conren
```

Doing so first tells you the number of possible display schemes and underlining schemes available. There are a few underlining schemes and many more display schemes. Some of these schemes were designed with black backgrounds in mind, whereas others were designed for white backgrounds. We suggest that you first select a display style scheme, and then after finding the scheme you like, explore the possible underlining schemes.

You would type

```
. conren style 1
```

to try display style scheme 1. `conren style` and `conren ul` automatically run `conren test` so that you can see the result on your screen. If the result is obviously bad, move on and try another scheme. If the resulting color scheme might be reasonable, try out Stata and see what you think. Try several commands, and look at a few help files to see if the selected display style scheme is appropriate. You can always return to the default with

```
. conren clear
```

which may be hard if you cannot even see what you are typing. Remember, if things are really bad, just type `exit` and then restart Stata.

Try all the prerecorded schemes to determine which one you like best.

## Can your terminal underline?

Type `conren test` to look at the various output types. Is the word `underlined` truly underlined—with the underlining on the same line and actually touching the characters—or is it instead more crudely rendered with a string of dashes underneath, on a second line?

If the word `underlined` truly is underlined, skip this section; evidently Stata has already figured out that your terminal can underline and is doing that.

Sometimes, Stata cannot figure that out for itself. Let's see if you can underline. Type

```
. conren ul 1
```

Now look at the output from `conren test` again. Is `underlined` underlined or just a mess? If it is a mess, you can remove the underlining codes (while leaving the display style codes untouched) by typing

```
. conren ul 0
```

You can now try the other available underlining schemes to see if they work any better for you.

## If you had success

So let's say that you discovered that what works best for you is

```
. conren style 4
. conren ul 1
```

and you just had no success with boldface at all. The next time you enter Stata, if you want the prettier look, you will have to type those two commands. To avoid having to do that, create a file `profile.do` and put those two lines in that file. Actually, we suggest that you put the lines in the file as

```
quietly conren style 4
quietly conren ul 1
```

because, if you also use Stata in batch mode, using the `quietly` option will prevent odd messages from appearing when Stata starts.

## If you did not have success

Well, now you really need to be technical. It is possible to make Stata's output look prettier if you know the escape sequences to cause special effects on your terminal.

Pretend that the codes for your terminal to turn underlining on and off were `Esc-[4m` and `Esc-[24m`. You could tell Stata that by typing

```
. set conren ulon <27> [ 4 m
. set conren uloff <27> [ 2 4 m
```

Escape has the decimal code 27, and you can type decimal codes by enclosing them in less-than and greater-than signs. You can just type regular characters. Remember, however, that you must type at least one space between each pair of characters.

All the features can be set in this way. If you type

```
. set conren
```

Stata will report what is currently set.

## Also see

[P] [smcl](#) — Stata Markup and Control Language

**stata** — Stata invocation command

## Syntax

```
xstata-mp [-option [-option [...]]] [stata_command]
```

```
xstata-se [-option [-option [...]]] [stata_command]
```

```
xstata [-option [-option [...]]] [stata_command]
```

```
stata-mp [-option [-option [...]]] [stata_command]
```

```
stata-se [-option [-option [...]]] [stata_command]
```

```
stata [-option [-option [...]]] [stata_command]
```

where the *options* are

- h display usage diagram
- q suppress logo and initialization messages
- s set background (batch) mode and log in SMCL (console only)
- b set background (batch) mode and log in plain text (console only)

## Description

`xstata-mp` starts the GUI version of Stata/MP. `xstata-se` starts the GUI version of Stata/SE. `xstata` starts the GUI version of Stata/BE.

`stata-mp` starts the console version of Stata/MP. `stata-se` starts the console version of Stata/SE. `stata` starts the console version of Stata/BE.

## Remarks and examples

stata.com

Here are the explanations for the startup options:

Typing `stata -h` does not enter Stata but instead shows the syntax diagram for invoking Stata.

The `-q` option suppresses all the initialization messages, including the Stata logo, when Stata is invoked.

The `-s` and `-b` options are for users who do not wish to run Stata interactively. Typing

```
% stata -s do bigjob
```

tells Stata to execute the commands in `bigjob.do`, suppress all screen output, and route the output to `bigjob.smcl`. Typing

```
% stata -b do bigjob
```

tells Stata to execute the commands in `bigjob.do`, suppress all screen output, and route the output to `bigjob.log`.

You may also use redirection (but it is not recommended):

```
% stata < bigjob.do > bigjob.log &
```

Warning: redirection will not work if your do-file contains either the `#delimit` commands or comment delimiters (`/*` and `*/`, `//`, or `///`). It also cannot create SMCL output. Hence, we recommend using options directly: `stata -s do bigjob &` or `stata -b do bigjob &`.

If you do use the first method, bear in mind that Stata automatically begins a log. If the `stata_command` is of the form

$$\{\text{do} \mid \text{run}\} \ [path] \ filename \ [.suffix]$$

the output is routed to `filename.smc1` in the current directory if you typed `-s` and to `filename.log` in the current directory if you typed `-b`. Otherwise, the output is routed to `stata.smc1` or `stata.log`. In either case, if the log file already exists, it is silently erased before starting.

Whether you use options or redirection, whenever Stata is running without a terminal, it sets `c(mode)` to contain `batch`. If Stata is running in interactive mode, `c(mode)` is set to be empty. See [\[P\] creturn](#).

## □ Technical note

Many users log in to Unix computers from Windows or Mac computers. If you are one of these users, there are several third-party packages that allow you to display X Windows graphics on Windows or Mac computers.

□

**pystata** — Call Stata from Python

You can call Stata from Python using the `pystata` Python package. This includes a suite of API functions and IPython magic commands that can be used to interact with Stata and Mata. To learn more about the `pystata` Python package, view the online documentation at <https://www.stata.com/python/pystata>. Or see [P] [PyStata module](#) for more information.

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).