15 Saving and printing output—log files

Contents

15.1 Overview
- 15.1.1 Starting and closing logs
- 15.1.2 Appending to an existing log
- 15.1.3 Suspending and resuming logging

15.2 Placing comments in logs
15.3 Logging only what you type
15.4 The log-button alternative
15.5 Printing logs
15.6 Creating multiple log files for simultaneous use

15.1 Overview

Stata can record your session into a file called a log file but does not start a log automatically; you must tell Stata to record your session. By default, the resulting log file contains what you type and what Stata produces in response, recorded in a format called Stata Markup and Control Language (SMCL); see [P] **smcl**. The file can be printed or converted to plain text for incorporation into documents you create with your word processor.

To start a log:	`. log using` *filename*
Your session is now being recorded in file *filename*`.smcl`.	
To temporarily stop logging:	
Temporarily stop:	`. log off`
Resume:	`. log on`
To stop logging and close the file:	`. log close`
You can now print *filename*`.smcl` or type:	`. translate` *filename*`.smcl` *filename*`.log`
to create *filename*`.log` that you can load into your word processor.	
You can also create a PDF of *filename*`.smcl` on Windows or Mac:	`. translate` *filename*`.smcl` *filename*`.pdf`
Alternative ways to start logging:	
append to an existing log:	`. log using` *filename*`, append`
replace an existing log:	`. log using` *filename*`, replace`
Using the GUI:	
To start a log:	click on the **Log** button
To temporarily stop logging:	click on the **Log** button, and choose **Suspend**
To resume:	click on the **Log** button, and choose **Resume**
To stop logging and close the file:	click on the **Log** button, and choose **Close**
To print previous or current log:	select **File > View...**, choose file, right-click on the Viewer, and select **Print**

Also, `cmdlog` will produce logs containing solely what you typed—logs that, although not containing your results, are sufficient to re-create the session.

To start a command-only log:	`. cmdlog using` *filename*
To stop logging and close the file:	`. cmdlog close`
To re-create your session:	`. do` *filename*`.txt`

15.1.1 Starting and closing logs

With great foresight, you begin working in Stata and type `log using session` (or click on the **Log** button) before starting your work:

```
. log using session
-------------------------------------------------------------------------------
      name:  <unnamed>
       log:  C:\example\session.smcl
  log type:  smcl
 opened on:  17 Mar 2014, 12:35:08

. use http://www.stata-press.com/data/r14/census5
(1980 Census data by state)

. tabulate region [freq=pop]

     Census |
     region |      Freq.     Percent        Cum.
------------+-----------------------------------
         NE | 49,135,283       21.75       21.75
    N Cntrl | 58,865,670       26.06       47.81
      South | 74,734,029       33.08       80.89
       West | 43,172,490       19.11      100.00
------------+-----------------------------------
      Total | 225,907,472      100.00

. summarize median_age

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
  median_age |         50       29.54    1.693445       24.2       34.7

. log close
      name:  <unnamed>
       log:  C:\example\session.smcl
  log type:  smcl
 closed on:  17 Mar 2014, 12:35:38
-------------------------------------------------------------------------------
```

There is now a file named `session.smcl` on your disk. If you were to look at it in a text editor or word processor, you would see something like this:

```
{smcl}
{com}{sf}{ul off}{txt}{.-}
      name:  {res}<unnamed>
       {txt}log:  {res}C:\example\session.smcl
  {txt}log type:  {res}smcl
 {txt}opened on:  {res}17 Mar 2014, 12:35:08

{com}. use http://www.stata-press.com/data/r14/census5
{txt}(1980 Census data by state)

{com}. tabulate region [freq=pop]

     {txt}Census {c |}
     region {c |}      Freq.     Percent        Cum.
{hline 12}{c +}{hline 35}
         NE {c |}{res} 49,135,283       21.75       21.75
{txt}    N Cntrl {c |}{res} 58,865,670       26.06       47.81
```

(*output omitted*)

What you are seeing is SMCL, which Stata understands. Here is the result of typing the file using Stata's `type` command:

```
. type session.smcl
-------------------------------------------------------------------------------
      name:  <unnamed>
       log:  C:\example\session.smcl
  log type:  smcl
 opened on:  17 Mar 2014, 12:35:08
. use http://www.stata-press.com/data/r14/census5
(1980 Census data by state)
. tabulate region [freq=pop]
     Census |
     region |      Freq.     Percent        Cum.
------------+-----------------------------------
         NE | 49,135,283       21.75       21.75
    N Cntrl | 58,865,670       26.06       47.81
      South | 74,734,029       33.08       80.89
       West | 43,172,490       19.11      100.00
------------+-----------------------------------
      Total |225,907,472      100.00
. summarize median_age
    Variable |       Obs        Mean   Std. Dev.       Min        Max
-------------+--------------------------------------------------------
  median_age |        50       29.54   1.693445       24.2       34.7
. log close
      name:  <unnamed>
       log:  C:\example\session.smcl
  log type:  smcl
 closed on:  17 Mar 2014, 12:35:38
-------------------------------------------------------------------------------
. _
```

What you will see is a perfect copy of what you previously saw. If you use Stata to print the file, you will get a perfect printed copy, too.

SMCL files can be translated to plain text, which is a format more useful for inclusion into a word processing document. If you type `translate` *filename*`.smcl` *filename*`.log`, Stata will translate *filename*`.smcl` to text and store the result in *filename*`.log`:

```
. translate session.smcl session.log
```

The resulting file `session.log` looks like this:

```
-------------------------------------------------------------------------------
      name:  <unnamed>
       log:  C:\example\session.smcl
  log type:  smcl
 opened on:  17 Mar 2014, 12:35:08
. use http://www.stata-press.com/data/r14/census5
(1980 Census data by state)
. tabulate region [freq=pop]
     Census |
     region |      Freq.     Percent        Cum.
------------+-----------------------------------
         NE | 49,135,283       21.75       21.75
    N Cntrl | 58,865,670       26.06       47.81
      South | 74,734,029       33.08       80.89
  (output omitted)
```

When you use `translate` to create *filename*`.log` from *filename*`.smcl`, *filename*`.log` must not already exist:

```
. translate session.smcl session.log
file session.log already exists
r(602);
```

If the file does already exist and you wish to overwrite the existing copy, you can specify the `replace` option:

```
. translate session.smcl session.log, replace
```

See [R] **translate** for more information.

On Windows and Mac, you can also convert your SMCL file to a PDF to share it more easily with others:

```
. translate session.smcl session.pdf
```

See [R] **translate** for more information.

If you prefer, you can skip the SMCL and create text logs directly, either by specifying that you want the log in `text` format,

```
. log using session, text
```

or by specifying that the file to be created be a `.log` file:

```
. log using session.log
```

15.1.2 Appending to an existing log

Stata never lets you accidentally write over an existing log file. If you have an existing log file and you want to continue logging, you have three choices:

- create a new log file
- append the new log onto the existing log file by typing `log using` *logname*`, append`
- replace the existing log file by typing `log using` *logname*`, replace`

For example, if you have an existing log file named `session.smcl`, you might type

```
. log using session, append
```

to append the new log to the end of the existing log file, `session.smcl`.

15.1.3 Suspending and resuming logging

Once you have started logging your session, you can turn logging on and off. When you turn logging off, Stata temporarily stops recording your session but leaves the log file open. When you turn logging back on, Stata continues to record your session, appending the additional record to the end of the file.

Say that the first time something interesting happens, you type `log using results` (or click on **Log** and open `results.smcl`). You then retype the command that produced the interesting result (or double-click on the command in the Review window, or use the *PgUp* key to retrieve the command; see [U] **10 Keyboard use**). You now have a copy of the interesting result saved in the log file.

You are now reasonably sure that nothing interesting will occur, at least for a while. Rather than type `log close`, however, you type `log off`, or you click on **Log** and choose **Suspend**. From now on, nothing goes into the file. The next time something interesting happens, you type `log on` (or click on **Log** and choose **Resume**) and reissue the (interesting) command. After that, you type `log off`. You keep working like this—toggling the log on and off.

15.2 Placing comments in logs

Stata treats lines starting with a "*" as comments and ignores them. Thus, if you are working interactively and wish to make a comment, you can type "*" followed by your comment:

```
. * check that all the spells are completed
. _
```

Stata ignores your comment, but if you have a log going the comment now appears in the file.

❑ Technical note

`log` can be combined with `#review` (see [U] **10 Keyboard use**) to bail you out when you have not adequately planned ahead. Say that you have been working in front of your computer, and you now realize that you have done what you wanted to do. Unfortunately, you are not sure exactly what it is you have done. Did you make a mistake? Could you reproduce the result? Unfortunately, you have not been logging your output. Typing `#review` will allow you to look over what commands you have issued, and, combined with `log`, will allow you to make a record. You can also see the commands that you have issued in the Review window. You can save those commands to a file by selecting the commands to save, right-clicking on the Review window, and selecting **Save Selected...**.

Type `log using` *filename*. Type `#review 100`. Stata will list the last 100 commands you gave, or however many it has stored. Because `log` is making a record, that list will also be stored in the file. Finally, type `log close`.

❑

15.3 Logging only what you type

Log files record everything that happens during a session, both what you type and what Stata produces in response.

Stata can also produce command log files—files that contain only what you type. These files are perfect for later going back and creating a Stata do-file.

`cmdlog` creates command log files, and its basic syntax is

`cmdlog using` *filename* [`, append replace`]	creates *filename*`.txt`
`cmdlog off`	temporarily suspends command logging
`cmdlog on`	resumes command logging
`cmdlog close`	closes the command log file

See [R] **log** for all the details.

Command logs are plain text files. If you typed

```
. cmdlog using session
(cmdlog C:\example\session.txt opened)
. use http://www.stata-press.com/data/r14/census5
(Census Data)
. tabulate region [freq=pop]
  (output omitted)
. summarize median_age
  (output omitted)
. cmdlog close
(cmdlog C:\example\session.txt closed)
```

file `mycmds.txt` would contain

```
use http://www.stata-press.com/data/r14/census5
tabulate region [freq=pop]
summarize median_age
```

You can create both kinds of logs—full session logs and command logs—simultaneously, if you wish. A command log file can later be used as a do-file; see [R] **do**.

15.4 The log-button alternative

The capabilities of the `log` command (but not the `cmdlog` command) are available from Stata's GUI interface; just click on the **Log** button or select **Log** from the **File** menu.

You can use the Viewer to view logs, even logs that are in the process of being created. Just select **File > View...**. If you are currently logging, the filename to view will already be filled in with the current log file, and all you need to do is click on **OK**. Periodically, you can click on the **Refresh** button to bring the Viewer up to date.

You can also use the Viewer to view previous logs.

You can access the Viewer by selecting **File > View...**, or you can use the `view` command:

```
. view myoldlog.smcl
```

15.5 Printing logs

You print logs from the Viewer. Select **File > View...**, or type `view` *logfilename* from the command line to load the log into the Viewer, and then right-click on the Viewer and select **Print**.

You can also print logs by other means; see [R] **translate**.

15.6 Creating multiple log files for simultaneous use

Programmers or advanced users may want to create more than one log file for simultaneous use. For example, you may want a log file of your whole session but want a separate log file for part of your session.

You can create multiple logs by using `log`'s `name()` option; see [R] **log**.