



An Introduction to Using the Command Line Interface (CLI) to Work with Files and Directories

Mac OS

by bertram lyons
senior consultant
avpreserve



Using the Command Line Interface: Mac OS

Introduction

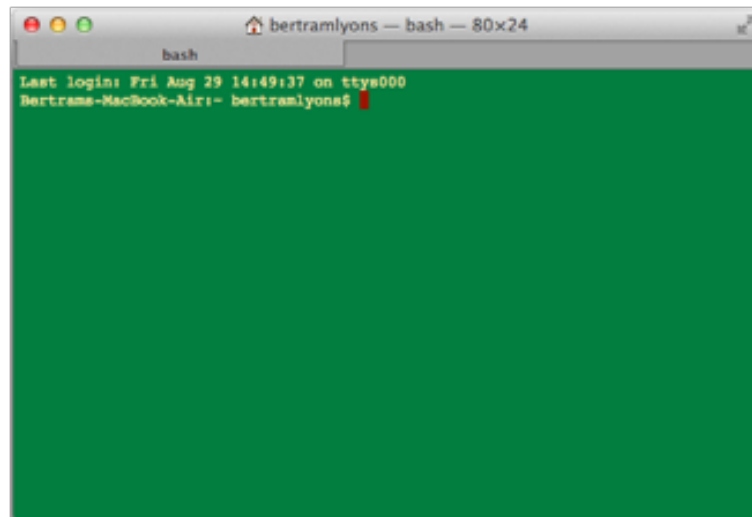
Every computer operating system has a different mechanism for communicating commands. Windows is the well known operating system for Microsoft machines; MAC OS is the Apple operating system, which itself is similar to UNIX/LINUX and other ?NIX operating systems. Each system requires a different syntax to communicate commands, although mostly all operating systems allow the same functions to be performed, even if the syntax differs.

In this demonstration, I'll be focusing on how to work with command line on your MAC OS, but it actually can be applied to most UNIX/LINUX interfaces as well. If you work on a PC, you can search online for the analogous MS DOS commands and attempt to use them in the MS DOS command prompt.

Using the MAC OS Terminal Interface

OPEN TERMINAL

The Terminal is a program included with all versions of Mac OS X. It is located in the Utilities folder within the Applications folder. When launched, it provides a command line interface (CLI) to control the underpinnings of the UNIX based operating system.



A FEW BASICS

When the Terminal is first opened, a message similar to the following appears:

```
Last login: Fri Aug 29 14:49:37 on ttys000  
Bertrams-MacBook-Air:~ bertramlyons$
```

Line 1 shows the last time the Terminal was used. If the date or last connection is unfamiliar, it could indicate someone has unwanted access to your computer.

Line 2 is the prompt, or command prompt. It shows the machine name (Bertrams-MacBook-Air), a colon, the present working directory (~) a space, the current user (bertramlyons) and a dollar sign (or a hash (#) if the shell has super user privileges).

THE CURRENT DIRECTORY

The current directory, also known as the working directory, is where you are. In the Finder, this is equivalent to having a window open and viewing the files. To determine the current directory,

Using the Command Line Interface: Mac OS

type `pwd` after the prompt, which produces an output similar to this:

```
Bertrams-MacBook-Air:~ bertramlyons$ pwd
/Users/bertramlyons
```

`pwd` stands for “print working directory”, and will likely output “/Users/(yourusername)” if you enter it upon first opening Terminal.

You may also notice that the current directory is displayed in a different form in the prompt. In this case, “~” is equivalent to “/Users/bertramlyons”, the home folder.

LISTING THE CURRENT DIRECTORY

To see what is contained in the current directory, you can use a list command (`ls`) with additional options, `-a` (all files and directories, including hidden), `-l` (output as a true structured list), `-R` (list recursively through all directories and files within the current directory). For example, the generic `ls` command yields a columnar list of directories and files:

```
Bertrams-MacBook-Air:~ bertramlyons$ ls
```

Applications	Movies	bert
Creative Cloud Files	Music	bin
Desktop	Pictures	index.html
Documents	Public	social-feed-manager
Downloads	Tresors	test1.txt
Library	bagger	websites

Adding the `-l` option to the `ls` command will give you more structured output (with more information, too):

```
Bertrams-MacBook-Air:~ bertramlyons$ ls -l
```

total 72

drwx-----	3	bertramlyons	staff	102	Apr	9	14:06	Applications
drwx-----@	3	bertramlyons	staff	102	Jun	26	08:03	Creative Cloud Files
drwxr-xr-x+	25	bertramlyons	staff	850	Sep	2	09:41	Desktop
drwx-----+	7	bertramlyons	staff	238	Aug	23	23:41	Documents
drwx-----+	55	bertramlyons	staff	1870	Sep	2	09:26	Downloads
drwx-----@	55	bertramlyons	staff	1870	Jun	6	08:11	Library
drwx-----+	5	bertramlyons	staff	170	Jul	16	10:25	Movies
drwx-----+	5	bertramlyons	staff	170	Feb	21	2014	Music
drwx-----+	7	bertramlyons	staff	238	Aug	23	23:41	Pictures
drwxr-xr-x+	6	bertramlyons	staff	204	May	6	21:17	Public
drwx-----@	5	bertramlyons	staff	170	Aug	22	15:02	Tresors
drwxr-xr-x	6	bertramlyons	staff	204	May	7	15:11	bagger
drwxr-xr-x	2	bertramlyons	staff	68	Jun	4	10:48	bert
drwxr-xr-x	17	bertramlyons	staff	578	May	11	23:02	bin
-rw-r--r--	1	bertramlyons	staff	14500	Apr	28	13:22	index.html
drwxr-xr-x	9	bertramlyons	staff	306	May	11	23:04	social-feed-manager
-rw-r--r--	1	bertramlyons	staff	18696	Apr	28	09:13	test1.txt
drwxr-xr-x	4	bertramlyons	staff	136	May	6	21:17	websites

Using the Command Line Interface: Mac OS

Here you can see the permissions settings for each file/directory in the first column on the left. “drwxrwxrwx” is the most you’ll see. The “d” identifies the object as a directory. If there is a “-” where the “d” is supposed to be, that identifies the object as a file. Following the first character, there is the potential for three sets of the characters “rwx”. On file systems there are three types of users who can be assigned permissions: Owner, Group, EveryoneElse. Each trio of “rwx” characters represents one of the three entities above, in the same respective order. Also, three types of permissions can be documented for a given file/directory: Read, Write, Execute (rwx).

So, for each file directory, the permissions documentation tells you:

- If it is a file or directory
- Whether the owner can read, write, and/or execute
- Whether the people in a certain group can read, write, and/or execute
- Whether everyone else who comes into contact with it can read, write, and/or execute

In the `ls -l` results you can also see things such as the owner (bertramlyons), the group that the file is part of (staff), the filesize, and the date last modified.

Adding the `-a` or `-R` option to the `ls` command will give you much more; `-a` reveals any hidden files, `-R` reads recursively through all directories. Try it for yourself.

```
Bertrams-MacBook-Air:~ bertramlyons$ ls -a -l -R
```

The `ls` command has many, many options, and is worth learning. For example, `ls -F` will postfix a slash (/) on every directory and an asterisk (*) on every executable file.

CHANGING DIRECTORIES

Now that you can see what is in your current directory, you may decide to navigate to a different directory. To change the current directory, use the `cd` command, as in:

```
Bertrams-MacBook-Air:~ bertramlyons$ cd Documents
Bertrams-MacBook-Air:Documents bertramlyons$ pwd
/Users/bertramlyons/Documents
```

You are now in your Documents directory, just as if you had double clicked on the Documents folder in the Finder. Note how the prompt has changed to include the path “~/Documents”. Remember, you can get a list of all the documents in this directory with the `ls` (list) command:

```
Bertrams-MacBook-Air:Documents bertramlyons$ ls -l
total 11136
-rw-r--r--  1 bertramlyons  staff   5698278 Mar 18 12:56 FileZilla_3.7.4.1_
i686-apple-darwin9.app.tar.bz2
drwxr-xr-x  6 bertramlyons  staff         204 Jun  6 12:55 Microsoft User Data
drwxr-xr-x  2 bertramlyons  staff           68 Jan 14 2014 WebEx
```

There are two directories and one file in my Documents folder.

Using the `cd` command has many options and shortcuts. The example above uses the `cd` command to move into a directory that is within the current working directory. That is why we

Using the Command Line Interface: Mac OS

could type: `cd Documents`. The system looked for a directory in the current working directory named Documents. You can also use global paths to change to any directory in the file system. For example, I may want to change directly to a directory on my desktop called desktop:

```
Bertrams-MacBook-Air:Desktop bertramlyons$ cd /Users/bertramlyons/
Desktop/desktop
Bertrams-MacBook-Air:desktop bertramlyons$ pwd
/Users/bertramlyons/Desktop/desktop
```

By typing in the global path, I was able to move directly to the desired directory.

There are shortcuts that you can use as well in combination with `cd`:

- “`cd ~`” will always take you directly to your home folder (for a given user)
- “`cd ..`” will always take you up a step in the hierarchy (or back a step) of the root file structure

Give it a try. You can always use `pwd` to see where you are.

DIRECTORY MANAGEMENT

You can use the CLI to create and delete directories just as you would use the Finder on your computer. The words ‘directory’ and ‘folder’ are often used interchangeably to identify organizational structures in a file system. You can place other directories or files within a directory. The command used in this case is called `mkdir`.

```
Bertrams-MacBook-Air:Desktop bertramlyons$ mkdir TestDirectory
```

This command creates a new directory on my Desktop called “TestDirectory.” The `mkdir` command by default creates a directory at the current working directory. Be sure you know where you are currently if you intend to create a directory without providing a global path. If I want to create a directory at a very specific location, then it is best to use the entire path in your `mkdir` command:

```
Bertrams-MacBook-Air:~ bertramlyons$ mkdir /Users/bertramlyons/Desktop/
TestDirectory2
```

These two commands both produced new directories on my Desktop with the specified names:

```
Bertrams-MacBook-Air:Desktop bertramlyons$ pwd
/Users/bertramlyons/Desktop
Bertrams-MacBook-Air:Desktop bertramlyons$ ls
TestDirectory
TestDirectory2
```

Now you may also want to delete a directory from your filesystem. Anytime you talk about deleting in your CLI, things get a little scary. Don’t be afraid, just learn to be very careful about using delete commands. To remove a directory, we use the command, `rmdir`. The following two commands do exactly the same thing (as long as my current working directory is my Desktop):

Using the Command Line Interface: Mac OS

```
Bertrams-MacBook-Air:Desktop bertramlyons$ rmdir TestDirectory
```

```
Bertrams-MacBook-Air:Desktop bertramlyons$ rmdir /Users/bertramlyons/  
Desktop/TestDirectory
```

It is always safer to be very specific when using a delete command. I always use the second example because it is less likely to cause problems if I mistype something or if my current working directory is different than what I think it is. To be even safer, I always use the `pwd` command before I do anything to files or directories. Then I always know where I am before I assert a command.

```
Bertrams-MacBook-Air:Desktop bertramlyons$ pwd  
/Users/bertramlyons/Desktop  
Bertrams-MacBook-Air:Desktop bertramlyons$ rmdir /Users/bertramlyons/  
Desktop/TestDirectory2
```

You can then use the `ls` command to verify that you deleted the directory as you intended.

```
Bertrams-MacBook-Air:Desktop bertramlyons$ pwd  
/Users/bertramlyons/Desktop  
Bertrams-MacBook-Air:Desktop bertramlyons$ ls  
TestDirectory
```

Here it shows that I removed `TestDirectory2` but not `TestDirectory`.

FILE MANAGEMENT

In similar fashion to directories, you can also create, delete, copy, and move/rename files with the CLI on your computer. There are also editing options for some file types, but this tutorial does not cover that advanced step.

CREATE FILES

The easiest file to generate from the CLI is a standard text file. There are a couple of ways to create a file using the command line. One option is to echo or print text and to have that text export to a file. This can be used very simply to get a file started. For example:

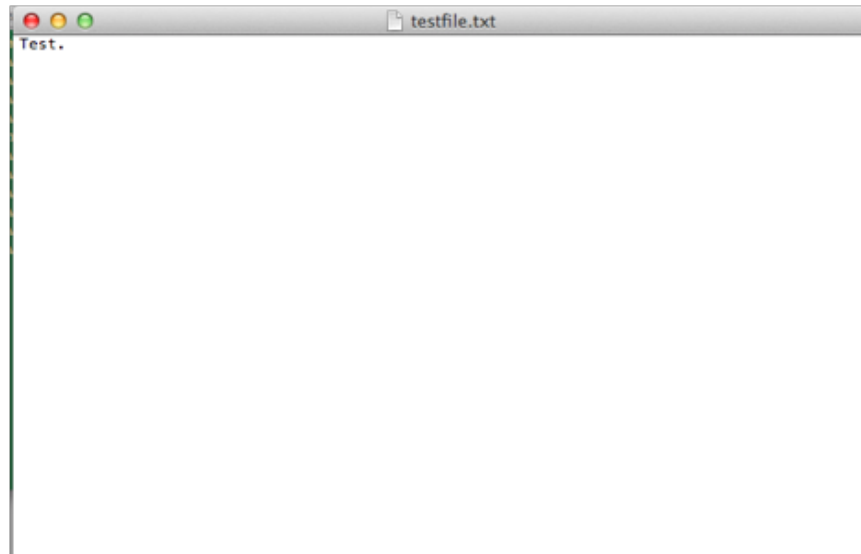
```
Bertrams-MacBook-Air:Desktop bertramlyons$ printf 'Test.' > testfile.txt
```

In this example, we use the `printf` command to print some type of text, in this case: `'Test.'`. Notice that we use single quotes to surround the text we want to print. This tells the computer that this is a text block and to ignore the content within the quotes. The `>` character is a command that tells the system to write the results of the previous command to a new file. In this case, we are asking it to write the results of our `printf` command to a file called `testfile.txt`. Since `testfile.txt` does not already exist, it will create a new file with that name. And remember, here it is defaulting to create the new file at the current working directory, which happens to be my desktop. You could be more specific and include the entire path of the file you want to create:

```
Bertrams-MacBook-Air:Desktop bertramlyons$ printf 'Test.' > /Users/  
bertramlyons/Desktop/testfile.txt
```

Using the Command Line Interface: Mac OS

Both of the above commands create a text file called testfile.txt on my desktop. When I open the file, I can see that it is a basic text file with the only contents being "Test.":



OPEN FILES

You might want to open the file from the command line in order to check that you created it correctly. You can use one of two easy options to do so. The open command opens the file just as if you had used a browser to open the file. It selects the default program to open the file type based on the extension. In this case:

```
Bertrams-MacBook-Air:Desktop bertramlyons$ open testfile.txt
```

or

```
Bertrams-MacBook-Air:Desktop bertramlyons$ open /Users/bertramlyons/  
Desktop/testfile.txt
```

Both of these commands will open the text file using the TextEdit program on your computer.

If you only want to read the contents of the file to verify it is what you think it is, you can also use a command called cat. The cat command reads the text of the file directly into the CLI.

```
Bertrams-MacBook-Air:Desktop bertramlyons$ cat testfile.txt  
Test.
```

DELETE FILES

To delete a file from your computer, you can use a similar command to the one learned above for deleting directories. The same caution should apply to deleting files, and you should always use pwd before you delete to make sure you know where you are in the file system. You should also use global paths to minimize risk.

```
Bertrams-MacBook-Air:Desktop bertramlyons$ pwd  
/Users/bertramlyons/Desktop  
Bertrams-MacBook-Air:Desktop bertramlyons$ rm /Users/bertramlyons/  
Desktop/testfile.txt
```

Using the Command Line Interface: Mac OS

This rm command removed my testfile.txt from my computer.

COPY FILES

Copying files is the process of making an exact copy of a file in a separate location on the file system. At the end of the process, there are two files, when originally there was one. The command, cp, has a syntax much like a sentence: copy what where. In the example below, I create a new testfile.txt on my Desktop and then make another copy in my Documents folder:

```
Bertrams-MacBook-Air:Desktop bertramlyons$ pwd
/Users/bertramlyons/Desktop
Bertrams-MacBook-Air:Desktop bertramlyons$ printf 'Test.' >/Users/
bertramlyons/Desktop/testfile.txt
Bertrams-MacBook-Air:Desktop bertramlyons$ ls
testfile.txt
Bertrams-MacBook-Air:Desktop bertramlyons$ cp ./testfile.txt /Users/
bertramlyons/Documents/
Bertrams-MacBook-Air:Desktop bertramlyons$ cd /Users/bertramlyons/
Documents/
Bertrams-MacBook-Air:Desktop bertramlyons$ pwd
/Users/bertramlyons/Documents
Bertrams-MacBook-Air:Documents bertramlyons$ ls
testfile.txt
```

MOVE FILES

Moving files is the process of renaming a file so that it (1) has a new name and lives in the same location or (2) has a new path and therefore lives in a new location, or a combination of both. At the end of the process, there is only one file. No copies are produced. For example, below I create a new file called testfile.txt on my Desktop and then move it to my Documents folder.

```
Bertrams-MacBook-Air:Desktop bertramlyons$ pwd
/Users/bertramlyons/Desktop
Bertrams-MacBook-Air:Desktop bertramlyons$ printf 'Test.' >/Users/
bertramlyons/Desktop/testfile.txt
Bertrams-MacBook-Air:Desktop bertramlyons$ ls
testfile.txt
Bertrams-MacBook-Air:Desktop bertramlyons$ mv ./testfile.txt /Users/
bertramlyons/Documents/testfile.txt
Bertrams-MacBook-Air:Desktop bertramlyons$ ls
Bertrams-MacBook-Air:Desktop bertramlyons$ cd /Users/bertramlyons/
Documents/
Bertrams-MacBook-Air:Desktop bertramlyons$ pwd
/Users/bertramlyons/Documents
Bertrams-MacBook-Air:Documents bertramlyons$ ls
testfile.txt
```

The file testfile.txt no longer exists on the Desktop. It has been moved to my Documents folder.

The mv command will overwrite files without warning. Before moving a file to a new destination, it is important to be aware of any files at that destination that might accidentally be overwritten

Using the Command Line Interface: Mac OS

by the command. The mv command can be used very simply to rename files in given directories. If I now want to change the name of testfile.txt to prodfile.txt, the mv command makes it easy. First change directory to the directory where the testfile.txt lives, then use the simple sentence syntax of move what where:

```
Bertrams-MacBook-Air:~ bertramlyons$ cd /Users/bertramlyons/Documents
Bertrams-MacBook-Air:Documents bertramlyons$ ls
testfile.txt
Bertrams-MacBook-Air:Documents bertramlyons$ mv testfile.txt prodfile.txt
Bertrams-MacBook-Air:Documents bertramlyons$ ls
prodfile.txt
```

Making Use of These Commands in a Library/Archive Environment

One essential use of these commands in a practical environment is having the ability to generate file lists from received hard drives or digital collections of large file counts.

By combining a few of the commands covered above (in tandem with spreadsheet exercises learned previously in this tutorial), you can easily walk away with a nice csv or excel spreadsheet that lists the basic information about every file on your drive.

For example, assume you received a drive from a donor and you want to do a quick assessment of the content without having to click around through every folder on the drive. If the drive is visible by your computer (or a local networked computer), you can use the following commands to get a listing. Imagine the drive is connected to your laptop:

First, open Terminal. Then use the pwd command to see where you are:

```
Bertrams-MacBook-Air:~ bertramlyons$ pwd
/Users/bertramlyons
```

Next, since all drives that connect to your mac laptop can be found in the path /Volumes, use the cd command to change your working directory to Volumes and then use the ls command to see what is there:

```
Bertrams-MacBook-Air:~ bertramlyons$ cd /Volumes
Bertrams-MacBook-Air:Volumes bertramlyons$ ls
Macintosh HD SIMPLETOUGH  blyons_bu  blyons_bu-1
```

Ok, good. Now we can see that there are four volumes: Macintosh HD (that's the main hard drive for my computer), SIMPLETOUGH (that's the external drive I plugged in), blyons_bu and blyons_bu-1 (those are backup volumes that I connect to virtually). In this case, I know I plugged in the SIMPLETOUGH and that is the drive I'm after, so I'll use a cd command to move to that directory, and then I'll use the ls command to see what is in it:

```
Bertrams-MacBook-Air:Volumes bertramlyons$ cd SIMPLETOUGH
Bertrams-MacBook-Air:SIMPLETOUGH bertramlyons$ ls
$RECYCLE.BIN          System Volume Information
GJ segments          afc2004004
Get_started_for_Mac.app  edit.seam
Recycled
```

Using the Command Line Interface: Mac OS

Ok, now the powerful part. The listing above just shows the top level of my drive. Some directories, some files, but it's not showing me everything on the drive. To do that I have to add a "-R" to my ls command, which makes the command function recursively — that is it lists all files and folders in all folders from the current working directory down. Since we have navigated to the root of the hard drive, if we use the recursive list command from here it will list everything on the drive. Furthermore, because there will be many files listed, I want to print the result of the command to a text file on my computer so I can use that data elsewhere. So we'll combine the ls command with the > (print to file) command:

```
Bertrams-MacBook-Air:SIMPLETOUGH bertramlyons$ ls -aLR >/Users/bertramlyons/Desktop/SIMPLETOUGH_list.txt
```

Now, if go to my Desktop, I'll see the new SIMPLETOUGH_list.txt file. If I open it, I'll have a text file that lists all contents on my drive. And even better, the files are delimited in such a way (with spaces) that I can import that .txt file into software such as Excel and use text-to-columns functions to parse the data cleanly into columns. The sky is the limit after that!

Appendix - Chart of Commands Discussed

Following is a chart of the commands explained above in this tutorial.

Command	Name	Description
pwd	Print working directory	Returns the full path of the current directory in which the CLI is reading.
ls	List	Lists the contents of the current working directory.
cd	Change directory	Changes the current working directory to a new location.
mkdir	Make directory	Creates a new directory.
rmdir	Remove directory	Deletes a specified directory.
printf	Print	Prints specified text.
>	Print to file	Prints the results of a command to a specified file.
open	Open	Opens a specified file in the OS environment.
cat	CAT	Opens a specified file in the CLI environment.
rm	Remove	Deletes a specified file or files.
cp	Copy	Copies a specified file or files to a specified destination.
mv	Move	Moves a specified file or files to a specified destination