

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

## **Windows**

by bertram lyons senior consultant avpreserve

#### Introduction

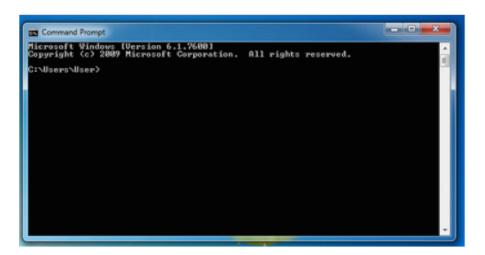
Every computer operating system has a different mechanism for communicating commands. Windows is the well known operating system for Microsoft machines; MAC OS is the Apple operating system, which itself is similar to UNIX/LINUX and other ?NIX operating systems. Each system requires a different syntax 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 Windows machine. If you work in a MAC OS environment or a UNIX/LINUX environment, there is another tutorial that you can use.

# Using the Windows Terminal Interface

#### **OPEN TERMINAL**

The Command Prompt is a program included with all versions of Windows. It is located in the Accessories folder within the Programs folder. When launched, it provides a command line interface (CLI) to control the underpinnings of the Windows operating system.



#### A Few Basics

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

```
Microsoft Windows [Version 6.1.7601]

Copyright <c> 2009 Microsoft Corporation. All rights reserved.
```

C:\Users\blyo>

Line 1 shows the Windows version you have installed. Line 2 shows a copyright statement. Line 3 is blank.

Line 4 is the prompt, or command prompt. It shows your current location in the system—the drive id, a colon, the present working directory, and a ">" symbol. A blinking prompt follows and this is where you can begin entering commands).

#### THE CURRENT DIRECTORY

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

current directory, type chdir (or cd) after the prompt, which produces an output similar to this:

C:\Users\blyo>chdir

C:\Users\blyo

C:\Users\blyo>cd

C:\Users\blyo

chdir (cd) stands for "change directory", which if used without any other information will output your current directory. It will likely output "\Users\(yourusername)\)" if you enter it upon first opening Command Prompt.

#### LISTING THE CURRENT DIRECTORY

To see what is contained in the current directory, you can use a directory list command (dir) with additional options, /a (indicates that attributes will follow), /t:w (indicates the output should display the time the files were last written), /s (list recursively through all directories and files within the current directory), or /p (indicates that the output should display one page at a time, useful when you ask to list many files), among others.

For example, the standard dir command yields a list of directories and files at the top level of your current working directory:

#### C:\Users\blyo>dir



In the dir results you can see things such as the date created, whether a row represents a directory <DIR> or a file (shows filesize instead of <DIR>), and the name of the directory or file on the filesystem.

The Is command has many, many options, and is worth learning. For example, /O will list by files in sorted order. See Appendix B at the end of this document for full listing of options to use with the dir command.

#### 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:

C:\Users\blyo>cd Downloads

C:\Users\blyo\Downloads>

You are now in your Downloads directory, just as if you had double clicked on the Downloads folder in the finder window. Note how the prompt has changed to include the path "C:\Users\blyo\Downloads". Remember, you can get a list of all the documents in this directory with the dir (list) command:

C:\Users\blyo\Downloads>dir

There are currently 4 directories and 195 files in my Downloads folder. How many are in yours?

Using the chdir (or cd) command has many options and shortcuts. The example above uses the chdir (or cd) command to move into a directory that is within the current working directory. That is why we could type: chdir Downloads. The system looked for a directory in the current working directory named Downloads. 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 in my Downloads called Test:

C:\Users\blyo\Downloads>chdir C:\Users\blyo\Downloads\Test

C:\Users\blyo\Downloads\Test>

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 chdir (or cd):

- "chdir .." will always take you up a step in the hierarchy (or back a step) of the root file structure)
- /d option allows you to switch drives, for example: chdir /d e:\pics would take you to a location on your e drive called /pics

Give it a try. You can always use chdir (or cd) without any other information to see where you are.

#### DIRECTORY MANAGEMENT

You can use the CLI to create and delete directories just as you would use the windows 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.

C:\Users\blyo>mkdir test1

This command creates a new directory in my blyo folder called "test1." 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:

```
C:\Users\blyo>mkdir C:\Users\blyo\test2
```

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 blyo):

```
C:\Users\blyo>rmdir test1
```

```
C:\Users\blyo>rmdir C:\Users\blyo\test2
```

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 chdir (or cd) command before I do anything to files or directories. Then I always know where I am before I assert a command:

```
C:\Users\blyo>chdir
C:\Users\blyo>
```

```
C:\Users\blyo>rmdir C:\Users\blyo\test2
```

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

```
C:\Users\blyo>dir
```

#### 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 prompt. One option is to to use the copy con function. There are a few steps to doing this. First make sure you are in the directory where you want to create the file. Then enter the following and press enter:

```
C:\Users\blyo>copy con myfile.txt
```

This will take you to a fresh line. You can create the file line by line by typing text. When you are finished, press enter to get to a new/fresh line. Then press Ctrl and Z together at the same time and release. This should produce a ^Z on the new line. Press enter and the file will be saved to your working directory. You can use the dir command to verify the existence of the new file.

Additionally, you could use the start notepad command to open notepad and create a new file.

#### Using the Command Line Interface: Windows

Once saved, the file will be saved into your working directory:

```
C:\Users\blyo>start notepad myfile2.txt
```

Give it a try and see what happens:

#### **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 very easy option to do so. Just type the path of the file into the command prompt and press enter. This 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:

```
C:\Users\blyo>myfile.txt
```

or

```
C:\Users\blyo>C:\Users\blyo\myfile.txt
```

Both of these commands will open the text file using the Notepad 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 type. The type command reads the text of the file directly into the CLI.

```
C:\Users\blyo>type myfile.txt
```

If the file is very large, you may want to read it one page at a time in your command prompt and you can use the "more" command in place of the "type" command.

```
C:\Users\blyo>more myfile.txt
```

#### **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 chdir (or cd) before you delete to make sure you know where you are in the file system. You should also use global paths to minimize risk. The delete command is "del."

```
C:\Users\blyo>chdir
C:\Users\blyo>
C:\Users\blyo>del C:\Users\blyo\myfile.txt
```

This del command removed my myfile.txt file 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, copy, has a syntax much like a sentence: copy what where. In the example below, I create a new testfile.txt in my blyo directory and then make another copy in my Downloads

#### folder:

You can use the dir command to check that the new file exists in both locations as expected.

#### **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 in my blyo folder and then move it to my Downloads folder.

The file testfile.txt no longer exists in my blyo folder. It has been moved to my Downloads folder.

You can use the dir command at both locations to verify this.

The move command will not overwrite files without warning. If there is a file with the same name, the prompt will ask you whether you would like to overwrite the existing file.

The move 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 move 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:

# 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 Command Prompt. Then use the chdir command to see where you are:

```
C:\Users\blyo>chdir
C:\Users\blyo>
```

Next, identify the location of the drive you would like to list. You can use the chdir command with the /d option to change to the appropriate drive:

```
C:\Users\blyo>chdir /d E:\
E:\>
```

Ok, good. Now use the dir command to see what is at the top level of the drive

```
E:\>dir
```

Ok, now the powerful part. The dir listing 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" and "/s" to my dir 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 dir command with the > (print to file) command (which we did not cover, but now you will see how it works). First, navigate back to your home directory:

```
E:\>chdir C:\Users\blyo
C:\Users\blyo>
```

We do this because the output of the next command will write the file to the current working directory. In most archival cases, we do not want to write a file to the hard drive we are evaluating. You could also specify the path to create the file more specifically if you want to be more careful. I'll show both examples below:

```
C:\Users\blyo>dir e:\ /a /s >drivelist.txt
```

C:\Users\blyo>dir e:\ /a /s >C:\Users\blyo\drivelist.txt

Now, if I go to my blyo folder, I'll see the new drivelist.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 A -Chart of Commands Discussed

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

Command	Name	Description	
chdir (or cd)	Print working directory	Returns the full path of the current directory in which the CLI is reading.	
dir	List	Lists the contents of the current working directory.	
chdir (or 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.	
copy con	Create file	Prints specified text to a new text file	
>	Print to file	Prints the results of a command to a specified file.	
[filename]	Open	Opens a specified file in the OS environment.	
type	Туре	Opens a specified file in the CLI environment.	
del	Delete	Deletes a specified file or files.	
сору	Сору	Copies a specified file or files to a specified destination.	
move	Move	Moves a specified file or files to a specified destination	

## Appendix B -Breakdown of dir Command Options

#### dir

Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/A[[:]attributes]] [/B] [/C] [/D] [/L] [/N] [/O[[:]sortorder]] [/P] [/Q] [/R] [/S] [/T[[:]timefield]] [/W] [/X] [/4]

[drive:][path][filename]	Specifies drive, directory, or files to list.		
/A	Displays files with specified attributes.		
attributes	D Directories	R Read-only files	
	H Hidden files	A Files ready for archiving	
	S System files	I Not content indexed files	
	L Reparse Points	- Prefix meaning not	

/B	Uses bare format (no heading information or summary).	
/C	Display the thousand separator in file sizes. This is the default. Use /-C to disable display of separator.	
/D	Same as wide but files are list sorted by column.	
/L	Uses lowercase.	
/N	New long list format where filenames are on the far right.	
/0	List by files in sorted order.	
sortorder	N By name (alphabetic) S By size (smallest first) E By extension (alphabetic) D By date/time (oldest first) G Group directories first - Prefix to reverse order	
/P	Pauses after each screenful of information.	
/Q	Display the owner of the file.	
/R	Display alternate data streams of the file.	
/S	Displays files in specified directory and all subdirectories.	
/T	Control what time field displayed or used for sorting	
timefield	C Creation	
A Last Access		
W Last Written		
/W	Uses wide list format.	
/X	This displays the short names generated for non-8dot3 file names.  The format is that of /N with the short name inserted before the long name. If no short name is present, blanks are displayed in its place.	
/4	Displays four-digit years	