

Command Line Basics

In the Contents table below you will find brief descriptions of, and links to, articles that have to do with basic information regarding use of a command line interface. The intent of these Command Line Basics articles is to provide some information for those interested in the [WinZip Command Line Support Add-On](#), but who have little experience with or little knowledge concerning the use of a command line interface. There are a good number of articles and a lot of information here, but users who are experienced with using a command line interface are likely not to find anything new here. For specific help with the add-on, please see the information in **WinZip Command Line Add-on Help** located in the WinZip program group.

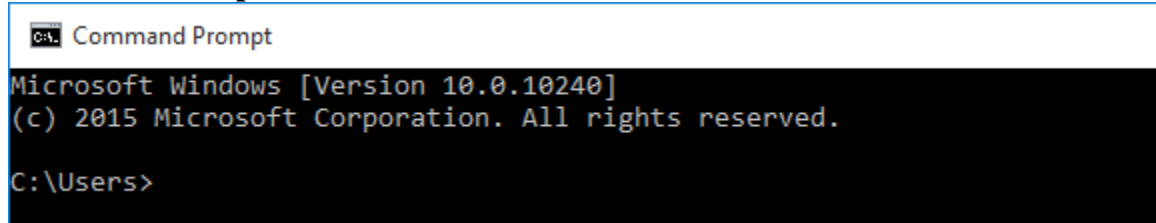
Other sources of similar or additional information can be readily found by performing an internet search using terms such as **command line**, **command prompt**, **DOS commands**, and or **batch files**. Anyone desiring more than basic material should consider reading from other sources.

Contents
<u>What is a command line interface?</u> An explanation, with examples, recommendations, and links to more information
<u>An explanation of "directory" and "path"</u> Two important and often used terms with descriptions and an example
<u>Command line syntax</u> How commands should be entered
<u>Displaying help for commands</u> Help text available while using the Command Prompt
<u>Delimiters, spaces, and special characters</u> Information about these terms, their usage, and special handling details
<u>The Path environment variable</u> What the Path variable is, how to configure it permanently, or use it temporarily
<u>Hints for using the CD command</u> Actual examples using the CD command and why it may be useful
<u>Using switches with commands</u> What command switches are and how to use them

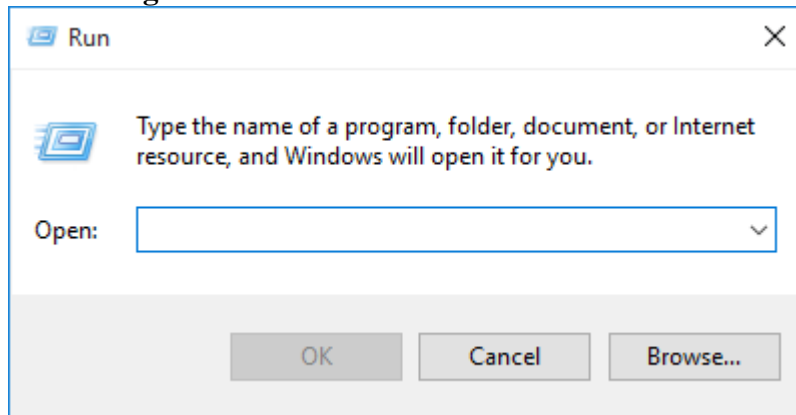
What is a command line interface?

A command line interface is a means used for interacting with software or files by typing commands to perform specific tasks. Typed commands can be acted on immediately if typed in a **Command Prompt** window or the **Run** dialog. As an alternative, if commands are saved in a script, such as a batch file (.bat), the commands will be performed when the script runs. A saved script file allows one to run a repeated task easily and can even be run by Task Scheduler.

Command Prompt window



Run dialog



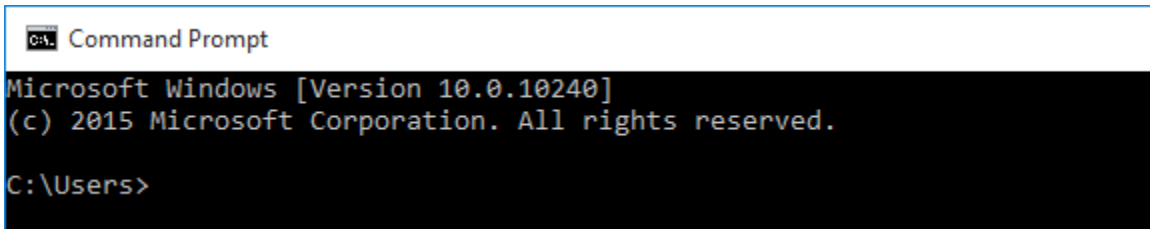
For the purpose of getting used to using a command line interface, it would be best to start off using the Command Prompt as this will give you immediate feedback if you have entered something incorrectly. In Windows versions on which you might have WinZip and the WinZip Command Line Support Add-on installed, the Command Prompt is a Windows application (cmd.exe) that interprets and runs your commands in the way DOS would have done (for the most part). A good reference document is available on [Wikipedia](#) that also includes links to other articles of interest. Of these, the link to a list of [DOS commands](#) could be very useful.

An explanation of "directory" and "path"

It is important to understand the two terms **directory** and **path** as you begin working with the Command Prompt. It may also be useful to you for other situations. Here are brief descriptions for each term:

Directory: A term that has the same basic meaning as **folder**, but predates the graphical folder concept and does not generally support virtual locations. (Alternate definition: A directory is a folder.)

Path: This term is descriptive in that it represents a type of "road map" to a specific file or directory. (Alternate definition: A path is a list, beginning with a drive letter, that tells which folders to open so that you can find a file or another folder.)



```
Command Prompt
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Users>
```

Using both of these terms and looking at the Command Prompt picture above, if you opened a Windows Explorer folder window (directory) and wanted to get to example.docx that was saved in the Users directory, you would start at **Local Disk C**, then you would open the **Users** directory and find example.docx inside. The entire path to the file would be:

C:\Users\example.docx

Command line syntax

When using a command, you must know and use the correct **syntax** of that command. The syntax of a command would be the rules regarding how the command is entered and will include options (if available) that, when included, will cause a change of some sort in the result of running the command. Setting up some commands can be a lot like using grammar to properly write a sentence, but in this case having rules governing where and how to place switches, filenames, and/or other options.

In some cases you can type a command with nothing else, press your **Enter** key on the keyboard, and see a correct result display. For example, if you want to display the current time in the Command Prompt window you can type **time**, press Enter, and the time will display. A second line will also display allowing you to **Enter the new time**. If you press Enter again without entering a new time, the Command Prompt will return to its earlier location.

Other commands only work, or may only give an expected response, if you enter additional information. For example, the command **CD**, meaning "change directory", will only display the directory you currently are in if nothing else is typed after it. Adding the correct information after CD will allow you to move into another, specific directory.

If a command is typed and then run with the wrong syntax, an error message will display. If you run a command with incorrect syntax from some type of script, such as a batch file, the script will end (i.e. it won't work) without indicating why. If you have redirected the display output to a file (see the end of the next article), that file will include the error message.

Displaying help for commands

While working with the Command Prompt, if you know what command you should use, but you aren't sure that you know the proper [syntax](#) and/or you don't know all of the options for it, you can usually display the help text for that command. To do this, type the command followed by a space, a forward slash, and a question mark. You must press **Enter** on your keyboard to have the Command Prompt run what you have typed. For example, if you want to see the help text for the **CD** command you would type:

CD /?

and then press the Enter key. What you would see at this point would be:

```
CHDIR [/D] [drive:][path]  
CHDIR [..]  
CD [/D] [drive:][path]  
CD [..]
```

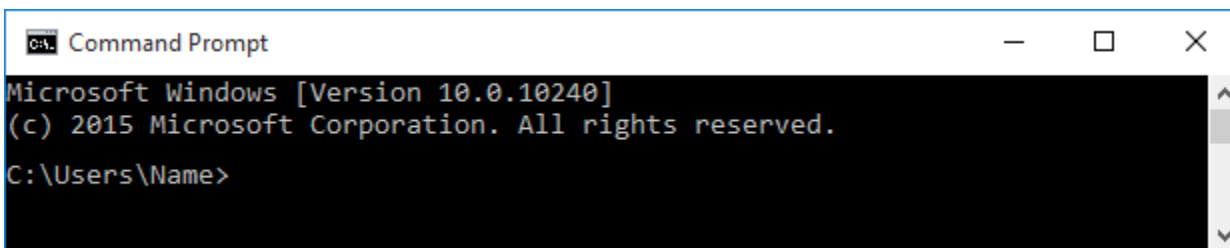
followed by some explanation of the syntax and options. Please note that **CHDIR** is treated as if it were the same command as **CD**. It is included in this help text to let you know that they are the same. If you display the help text for CHDIR you will see the same text.

If you would like to save the help text for a command as a file for later use, you can redirect the display output to a text file. To do this, add a space, the greater than symbol, and a filename with an extension that will be recognized as a text file (typically .txt or .log). If you want the file saved to a different directory, put a [Path](#) in front of the filename. Here is an example:

```
CD >C:\Users\Logon\CD.txt
```

Delimiters, spaces, and special characters

In most cases, a space in your command line is treated as a delimiter. A delimiter tells the [command line interface](#) that a boundary, or a break, has been reached. To demonstrate this, let's say that you wanted to run a file named example.zip from a command line and this file is one that you have saved in the WinZip folder. If your command prompt is presently in your user folder, as pictured below (the command prompt is in the **Name** folder in the picture), you would have to include the [Path](#) to example.zip in your command in order to run it.



However, since **Program Files** (or **Program Files (x86)** for 64-bit operating systems running a 32-bit version of WinZip) is in the path to the WinZip folder AND since there is a space in that name, if you enter the command in this way:

```
C:\Program Files\WinZip\example.zip
```

a message will display saying:

```
'C:\Program' is not recognized as an internal or external operable program or batch file.
```

This is because the space, acting as a delimiter, has told the command prompt that "C:\Program" is some sort of command, which is false. To correct this error, you would surround the entire command with double quotes, like this:

```
"C:\Program Files\WinZip\example.zip"
```

Another instance where you would want to use double quotes would be if something on your command line, such as a filename, includes a DOS special character such as the ampersand (&). One place where you might encounter this special character situation is with the WinZip Command Line Support Add-On. It is quite possible that a password that you would have to type for an encrypted Zip file would include such a character.

A notable exception where a space does not act as a delimiter is the CD command. If you view the help text for the CD command, you will see that it states the exception in this way:

CHDIR... does not treat spaces as delimiters...

The PATH environment variable

When you issue a command, the interface must know the location of the file or program to which the command refers. Like the example in the article [Delimiters, spaces, and special characters](#), which runs example.zip, the path must be included (C:\Program Files\WinZip\example.zip). If you only type example.zip on your command line and then press **Enter** the result will be much the same as asking someone to hand you an item without first telling that person that the item is in another room.

If you will be issuing more than one command to the same location during a session, you can make things easier by editing your computer's **PATH** variable. This can be done temporarily or permanently. The examples below assume that multiple commands are going to be issued to the WinZip add-on which is found in the WinZip folder.

To temporarily change the PATH variable in a Command Prompt session so that it includes the WinZip folder, type:

```
PATH=%PATH%;C:\Program Files\WinZip
```

and press Enter (double quotes are not necessary as PATH does not use spaces as a delimiter). Once you have done this, you can type a command line that begins with either WZZIP or WZUNZIP during this Command Prompt session. If you do not have the path to the WinZip folder in your PATH variable, you must always include that path in your command, as in the example below.

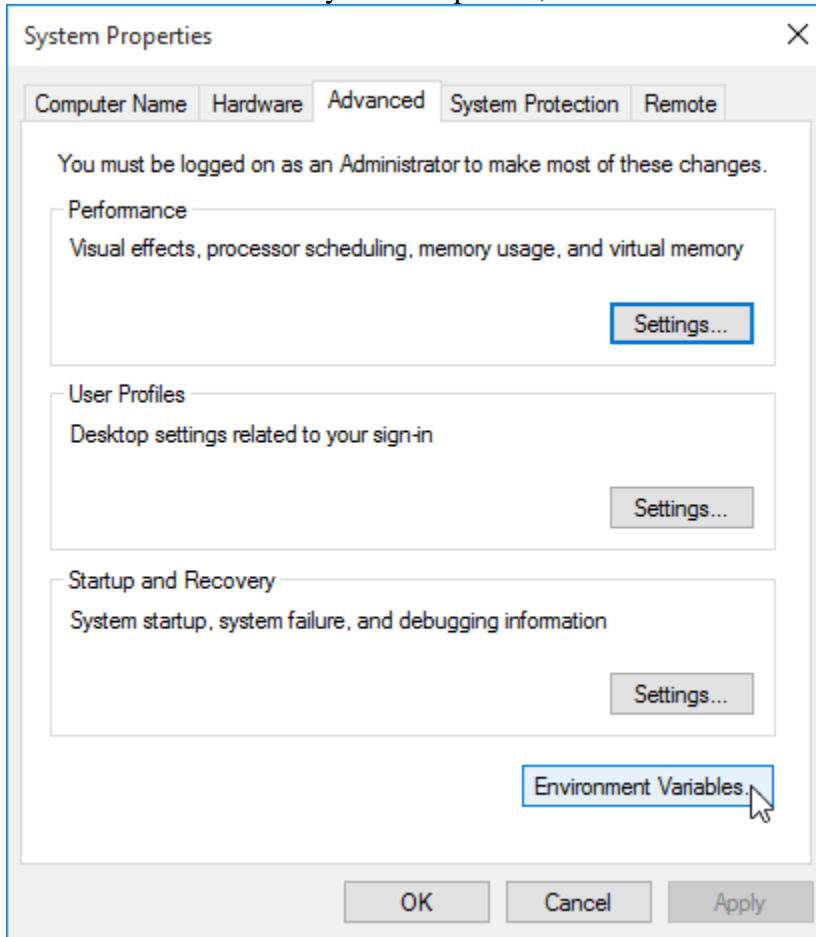
```
"C:\Program Files\WinZip\WZZIP"
```

Note: The entire command above must be surrounded by double quotes due to the space in **Program Files**.

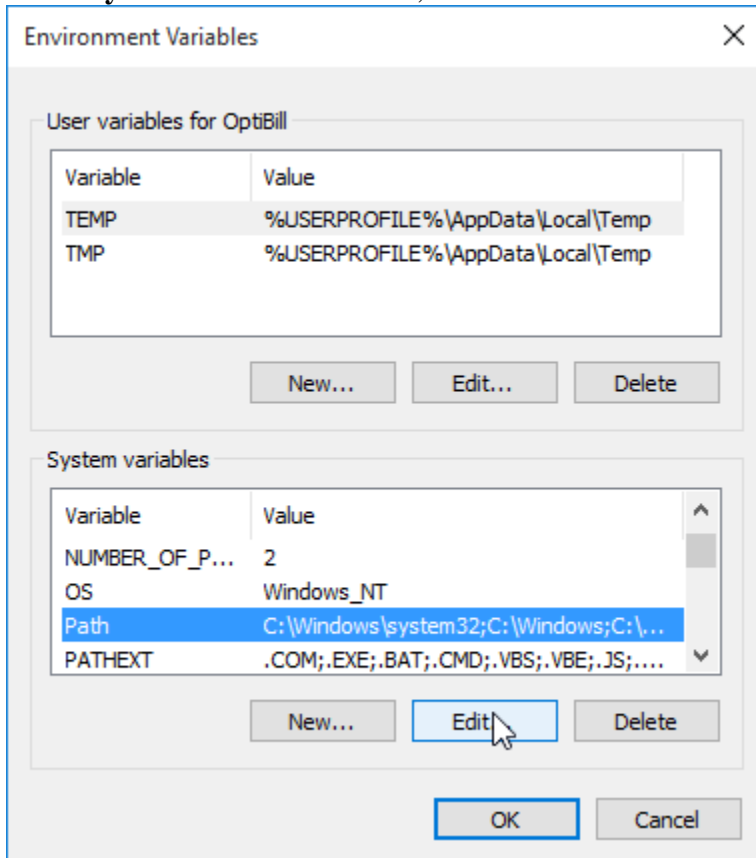
To permanently include the WinZip folder in the path, you would need to edit the **Path Environment Variable**. To do this:

1. Open **System** from the Control Panel
(change the Control Panel from the Category view to easily find this)
2. Click the link for **Advanced system settings** on the left side of System

3. In the Advanced tab of System Properties, click **Environment Variables**



4. In the **System variables** section, scroll to and select **Path** and then click **Edit**



5. In the **Variable value** text box, place your cursor at the end of the current value

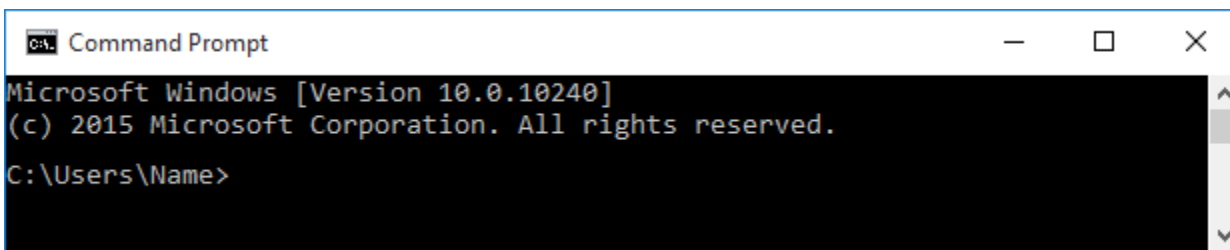
6. Type a semicolon as a delimiter
7. Type the full path to the WinZip folder (usually C:\Program Files\WinZip)
8. Click **OK** in each open dialog and close System

The next time you open the Command Prompt or run a script, you will no longer need to enter the Path to the WinZip folder for any commands using the add-on.

Hints for using the CD command

The **CD** command has already been mentioned in previous articles: [Command line syntax](#), [Displaying help for commands](#), and [Delimiters, spaces, and special characters](#). The intent of this article is to give a more complete explanation of how to use the CD command. Some of this information is not obvious when you view the [help text](#) available in the Command Prompt.

The CD command is used to [change directories](#), which means it moves the command prompt into a different folder. Please look at the picture below to see that, in this case, the command prompt is in the **Name** subdirectory, which is inside the **Users** directory, which is on the **C drive**.



```
Command Prompt
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Users\Name>
```

If you plan to run a number of commands involving files in another directory, you will either need to enter the [path](#) to this other directory whenever referring to the files in it or you will need to change directories in the command prompt window.

For example, let us say that you will be working with files saved in a directory named Letters which was created in the Documents directory. Since these two are both subdirectories below the path of the current directory (C:\User\Name) you can use the CD command to change directories in one of two ways.

Change to the Letters directory in 2 steps

- Type **CD Documents** and press Enter

```
C:\Users\Name>cd documents
C:\Users\Name\Documents>
```

- Type **CD Letters** and press Enter

```
C:\Users\Name\Documents>cd letters
C:\Users\Name\Documents\Letters>
```

Note: The Command Prompt does not care if you use capital or lower case letters.

Change to the Letters directory in 1 step

- Type **CD Documents\Letters** and press Enter

```
C:\Users\Name>cd documents\letters
C:\Users\Name\Documents\Letters>
```

The backslash character in the example above represents a folder (directory) so that this command has told the Command Prompt to move down into the Documents subdirectory and continue into the Letters subdirectory.

If you want to move to a higher subdirectory you can use **CD..** which will move the command prompt to the subdirectory that is one higher than its current location. If you started where we just ended (the Letters subdirectory) and you used **CD..** you would end up in the Documents subdirectory. If you used it twice, you would find yourself in the Name subdirectory.

If you want to move to a subdirectory of another parent directory or if you want to move up to the root of the drive, you would want to use **CD ** (please note that there is a space between CD and the backslash character). If you type **CD ** and press Enter, you will then be at the root of the current drive.

```
C:\Users\Name\Documents\Letters>cd \  
C:\>
```

Starting back in the Letters subdirectory, if you wanted to move directly from there to C:\Invoices\January you would type **CD \Invoices\January** and press Enter.

```
C:\Users\Name\Documents\Letters>cd \invoices\january  
C:\Invoices\January>
```

In the example above the first backslash instructs the Command Prompt to begin at the root of the drive, move into the Invoices directory and then continue into the January subdirectory.

Using switches with commands

A command line switch (also known as an option, a parameter, or a flag) acts as a modifier to the command you are issuing in the Command Prompt window, in a batch file, or in other scripts. Usually, a switch is a single letter preceded by a forward slash. For an example, we will look at the **DIR** command and its switches.

The DIR command is used to display a listing of files and folders in the current folder. For example, if you open a Command Prompt window, use "[CD \](#)" to move to the Documents folder, type DIR, and press Enter, you may see something similar to the picture below.

```
Directory of C:\Users\Name\Documents  
09/23/2015  02:08 PM    <DIR>      .  
09/23/2015  02:08 PM    <DIR>      ..  
05/25/2010  11:54 AM             4,265,797 Adhesive properties.docx  
12/10/2004  12:25 AM             14,336 Curried Favor.pub  
01/04/2001  02:19 PM             531,222 Forms of custard.xlsx  
09/18/2011  10:11 PM           2,739,122 Historical Manicotti.pptx  
09/23/2015  02:04 PM    <DIR>      Letters  
09/23/2015  02:04 PM    <DIR>      My WinZip Files  
          4 File(s)          7,550,477 bytes  
          4 Dir(s) 52,373,315,584 bytes free
```

The items are listed alphabetically by name (far right column), regardless of whether they are files or directories. Each line contains the item's timestamp, <DIR> in the next column (if the item is a directory), the size in bytes (if the item is a file), and then the name. Totals are displayed at the bottom and there is some introductory information at the beginning (not displayed in the picture).

If you now display the help text for this command by typing **DIR /?** you will see a brief sentence describing what DIR does followed by the [syntax](#) for the command with brief descriptions for each option. The syntax for DIR should look like this:

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

Each option for the DIR command is in brackets. As already demonstrated, you can use DIR all by itself. However, you may want to display this list in a particular format, in a particular order, and/or you may want to display only particular information. You can even display a listing from another directory.

For example, if you wanted to:

- List the contents of the Name subdirectory in the Users directory
- See the contents of subdirectories below the Logon directory
- Display only the names of the items
- Display the names sorted in reverse alphabetic order

you could type: **DIR Users\Name /s /b /o:-n** and then press Enter. The output of your command would begin with the subfolder that is last alphabetically (possibly the Video subfolder) listing the path to this folder first (C:\Users\Logon\Video) and then paths to each file. The output should then work backwards through the alphabet ending with the subfolder that is first alphabetically.

By experimenting with the various switches, you may find a result that would be particularly useful. You may want to record the result of the command you run in a log file, which you can do by [redirecting](#) the display output.

This is the last article in the Command Line Basics section. If you would like more information you may want to do an internet search for **command line**, **command prompt**, **DOS commands**, **batch files**, or some other topic of interest. You may also want to review the information in the Related Articles listed below. If you have any questions regarding this information as it relates to the [WinZip Command Line Support Add-On](#), please send an email message to [Technical Support](#).

Related Articles:

[Command Line Basics articles](#)

[Methods to try fixing invalid Zip files with the command line add-on](#)

[How to create a log file when using WinZip Command Line Support Add-on](#)

[Create individual Zip files with a command line](#)

[How to include the current date in the filename - WinZip Command Line Support Add-on](#)