



FREE eBook

LEARNING

cmd

Free unaffiliated eBook created from
Stack Overflow contributors.

#cmd

Table of Contents

About.....	1
Chapter 1: Getting started with cmd.....	2
Remarks.....	2
Examples.....	2
Opening a Command Prompt.....	2
Navigating in cmd.....	2
Commands in CMD.....	3
Features.....	8
Hello World.....	8
Comments.....	9
Chapter 2: Using xcopy command.....	11
Introduction.....	11
Parameters.....	11
Examples.....	11
Copying multiple files including tree structure.....	11
Credits.....	12

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [cmd](#)

It is an unofficial and free cmd ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official cmd.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with cmd

Remarks

This section provides an overview of what cmd is, and why a developer might want to use it.

It should also mention any large subjects within cmd, and link out to the related topics. Since the Documentation for cmd is new, you may need to create initial versions of those related topics.

Examples

Opening a Command Prompt

The command prompt comes pre-installed on all Windows NT, Windows CE, OS/2 and eComStation operating systems, and exists as `cmd.exe`, typically located in

```
C:\Windows\system32\cmd.exe
```

On Windows 7 the fastest ways to open the command prompt are:

- Press `Win`, type "cmd" and then press `Enter`.
- Press `Win+R`, type "cmd" then then press `Enter`.

It can also be opened by navigating to the executable and double-clicking on it.

In some cases you might need to run `cmd` with elevated permissions, in this case right click and select "Run as administrator". This can also be achieved by pressing `Control+Shift+Enter` instead of `Enter`.

Navigating in cmd

One of the most common things you'll need to do in the command prompt is navigate your file system. To do this, we'll utilize the `cd` and `dir` keywords. Start by opening up a command prompt using one of the methods mentioned [here](#). You most likely see something similar to what's below, where `UserName` is your user.

```
C:\Users\UserName>
```

Regardless of where in your file structure you are, if your system is like most, we can start with this command:

```
cd C:\
```

This will change your current directory to the `C:\` drive. Notice how the screen now looks like this

```
C:\>
```

Next, run a `dir` so we can see anything in the `c:\` drive

```
dir
```

This will show you a list of files and folders with some information about them, similar to this:

```
06/15/2016 12:32 PM <DIR> Users
07/19/2016 09:00 AM <DIR> Windows
                3 File(s)      1,161 bytes
                13 Dir(s)    427,349,934,080 bytes free
C:\>
```

There's lots of good info here, but for basic navigation, we just care about the right-most column. Notice how we have a `Users` folder. That means we can run this

```
cd Users
```

Now if you run `dir` again, you'll see all the files and folders in your `c:\Users` directory. Now, we didn't find what we wanted here, so let's go back to the parent folder. Rather than type the path to it, we can use `..` to go up one folder like so

```
cd ..
```

Now we are back in `c:\`. If you want to go up multiple folders at once, you can put a backslash and another set of periods like so: `cd ..\..\`, but we only needed one folder.

Now we want to look in that `Program Files` folder. To avoid confusing the system, it's a good idea to put quotes around the directories, especially when there are spaces in the name. So this time, we'll use this command

```
C:\>cd "Program Files"
```

Now you are in `C:\Program Files>` and a `dir` command now will tell you anything that's in here.

So, say we get tired of wandering around to find the folder and looked up exactly where we were needing to go. Turns out it's `C:\Windows\Logs` Rather than do a `..` to `Windows` to `Logs`, we can just put the full path like so:

```
cd "C:\Windows\Logs"
```

And that's the basics of navigating the command prompt. You can now move through all your folders so you can run your other commands in the proper places.

Commands in CMD

The available commands will be displayed, including a brief description, in tabular format. In Windows 10 the following commands are listed:

Command	Description
ASSOC	Displays or modifies file extension associations.
ATTRIB	Displays or changes file attributes.
BREAK	Sets or clears extended CTRL+C checking.
BCDEDIT	Sets properties in boot database to control boot loading.
CACLS	Displays or modifies access control lists (ACLs) of files.
CALL	Calls one batch program from another.
CD	Displays the name of or changes the current directory.
CHCP	Displays or sets the active code page number.
CHDIR	Displays the name of or changes the current directory.
CHKDSK	Checks a disk and displays a status report.
CHKNTFS	Displays or modifies the checking of disk at boot time.
CLS	Clears the screen.
CMD	Starts a new instance of the Windows command interpreter.
COLOR	Sets the default console foreground and background colors.
COMP	Compares the contents of two files or sets of files.
COMPACT	Displays or alters the compression of files on NTFS partitions.
CONVERT	Converts FAT volumes to NTFS. You cannot convert the current drive.
COPY	Copies one or more files to another location.
DATE	Displays or sets the date.
DEL	Deletes one or more files.
DIR	Displays a list of files and subdirectories in a directory.
DISKPART	Displays or configures Disk Partition properties.
DOSKEY	Edits command lines, recalls Windows commands, and creates macros.

Command	Description
DRIVERQUERY	Displays current device driver status and properties.
ECHO	Displays messages, or turns command echoing on or off.
ENDLOCAL	Ends localization of environment changes in a batch file.
ERASE	Deletes one or more files.
EXIT	Quits the CMD.EXE program (command interpreter).
FC	Compares two files or sets of files, and displays the differences between them.
FIND	Searches for a text string in a file or files.
FINDSTR	Searches for strings in files.
FOR	Runs a specified command for each file in a set of files.
FORMAT	Formats a disk for use with Windows.
FSUTIL	Displays or configures the file system properties.
FTYPE	Displays or modifies file types used in file extension associations.
GOTO	Directs the Windows command interpreter to a labeled line in a batch program.
GPRESULT	Displays Group Policy information for machine or user.
GRAFTABL	Enables Windows to display an extended character set in graphics mode.
HELP	Provides Help information for Windows commands.
ICACLS	Display, modify, backup, or restore ACLs for files and directories.
IF	Performs conditional processing in batch programs.
LABEL	Creates, changes, or deletes the volume label of a disk.
MD	Creates a directory.

Command	Description
MKDIR	Creates a directory.
MKLINK	Creates Symbolic Links and Hard Links
MODE	Configures a system device.
MORE	Displays output one screen at a time.
MOVE	Moves one or more files from one directory to another directory.
OPENFILES	Displays files opened by remote users for a file share.
PATH	Displays or sets a search path for executable files.
PAUSE	Suspends processing of a batch file and displays a message.
POPD	Restores the previous value of the current directory saved by PUSHHD.
PRINT	Prints a text file.
PROMPT	Changes the Windows command prompt.
PUSHHD	Saves the current directory then changes it.
RD	Removes a directory.
RECOVER	Recovers readable information from a bad or defective disk.
REM	Records comments (remarks) in batch files or CONFIG.SYS.
REN	Renames a file or files.
RENAME	Renames a file or files.
REPLACE	Replaces files.
RMDIR	Removes a directory.
ROBOCOPY	Advanced utility to copy files and directory trees
SET	Displays, sets, or removes Windows environment variables.
SETLOCAL	Begins localization of environment changes in a batch file.
SC	Displays or configures services (background processes).

Command	Description
SCHTASKS	Schedules commands and programs to run on a computer.
SHIFT	Shifts the position of replaceable parameters in batch files.
SHUTDOWN	Allows proper local or remote shutdown of machine.
SORT	Sorts input.
START	Starts a separate window to run a specified program or command.
SUBST	Associates a path with a drive letter.
SYSTEMINFO	Displays machine specific properties and configuration.
TASKLIST	Displays all currently running tasks including services.
TASKKILL	Kill or stop a running process or application.
TIME	Displays or sets the system time.
TITLE	Sets the window title for a CMD.EXE session.
TREE	Graphically displays the directory structure of a drive or path.
TYPE	Displays the contents of a text file.
VER	Displays the Windows version.
VERIFY	Tells Windows whether to verify that your files are written correctly to a disk.
VOL	Displays a disk volume label and serial number.
XCOPY	Copies files and directory trees.
WMIC	Displays WMI information inside interactive command shell.

To get more insight about a specific command use the `/?` option, e.g. the `tree` command gives:

```
tree /?

Graphically displays the folder structure of a drive or path.

TREE [drive:][path] [/F] [/A]

    /F    Display the names of the files in each folder.
    /A    Use ASCII instead of extended characters.
```

Features

Microsoft Command Prompt is a *command-line interpreter* (CLI) for the Windows operating systems.

A CLI is program intended primarily to read operating system instructions typed on a keyboard by the user. It is therefore addressed also as a *command-line interface*, to contrast it with graphical interfaces.

As these interfaces (whether textual or graphical) shield the user from directly accessing to the operating system kernel, they are also said *shells*.

Given the name of the Command Prompt executable file, `cmd.exe`, the Command Prompt is friendly named `cmd`. Given its OS piloting role, it is also said the *console*.

Like other shells, `cmd` can read batch of instructions from a file. In this case the `cmd` shell acts as a language interpreter and the file content can be regarded as an actual program. When executing these batch programs, there is no intermediate compilation phase. They are typically read, interpreted and executed line by line. Since there is no compilation, there is no production of a separated executable file. For this reason the programs are denoted *batch scripts* or *shell scripts*.

Note that the instructions entered interactively might have a slightly different syntax from those submitted as a script, but the general principle is that what can be entered from the command line can be also put in a file for later reuse.

Hello World

Command Prompt batch scripts have extension `.cmd` or `.bat`, the latter for compatibility reasons.

To create a hello-word-script, you first need a place where to type it. For simple scripts, also the Windows Notepad will do. If you are serious about shell scripting, you need more effective tools. There are anyway several free alternatives, such as [Notepad++](#).

In your designated editor type:

```
echo Hello World
pause
```

Save it as `hello.cmd`

If you are using "Notepad" as an editor, you should pay much attention to the saved name, as Notepad tends to add always a `.txt` extension to your files, which means that the actual name of your file might be `hello.cmd.txt`. To avoid this, in the save dialog box:

1. In the `File name` field enter the name in double quotes, e.g. `"hello.cmd"`
2. In the `Save as type` field select All Files, instead of the default Text Document option.

If the file has been saved properly, its icon should be similar to (Windows Vista):



You may also consider to disable the option "Hide extension for known file types" in File Explorer folder view options. In this case, file names are always displayed with their extensions.

To execute `hello.cmd` there are two possibilities. If you are using the Windows graphical shell, just double click on its icon.

If you want to use the Command Prompt itself, you must first identify the directory where you saved `hello.cmd`. In this regard, if you open File Explorer with **Win+E**. In the windows listing files, you normally read the name of the directory path containing them. You can therefore identify the directory of `hello.cmd`. Windows directory names tend to be quite long and typing them is error prone. It is better if you select and copy the directory path in the clipboard for later pasting.

Start the Command Prompt. You read a line similar to this.

```
Microsoft Windows [Version ...]
(c) ... Microsoft Corporation. All rights reserved.

C:\Users\...>
```

The version/year of Windows of course depends on yours. In the the final line, before `>`, you read the path of the directory which is current. You should make current the directory where your script is. For this reason enter the change directory command `cd`, using a line similar to the following:

```
cd <dirpath>
```

Instead of `<dirpath>`, paste the name of the directory you previously copied.

To paste the directory path, in Windows 10, you just need to type `Ctrl-C`, as you would in an editor. For older systems you should be able to do this by right clicking in the `cmd` window. After entering the command, note that current path, before `>`, changes accordingly.

You can now run your hello script by simply entering:

```
hello
```

Comments

The script prints an output similar to:

```
C:\Users\...>echo Hello World
Hello World

C:\Users\...>pause
Press any key to continue . . .
```

The lines hosting the symbol `>` restate the script instructions as if you had entered interactively. This can be disabled writing:

```
@echo off
```

as the first line of your script. This might reduce the clutter, but you have less hints on what is going on, with respect to those script commands that do not give visible outputs.

The last command, `pause`, prompts you to hit any key. When you do, you exit `hello`. If you run `hello` from the console, you don't really need it, because, when `hello` terminates its execution, `cmd.exe` remains open and you can read `hello` output. When double-clicking in Explorer, you start `cmd.exe` for the time necessary to execute `hello`. When `hello` terminates, `cmd.exe` does the same and you have no possibility to read `hello` output. `pause` command prevents `hello` from exiting until you hit a key, which gives also the possibility to read the output.

Finally, despite the name of the script is `hello.cmd`, it is not necessary to type the whole name, its `hello` stem is sufficient. This mechanism works for executables too, with extension `.exe`. What if there is a script `hello.cmd` and an executable `hello.exe` in the same directory? The former has priority in the Command Prompt, so `hello.cmd` will be executed.

Read [Getting started with cmd online](https://riptutorial.com/cmd/topic/2548/getting-started-with-cmd): <https://riptutorial.com/cmd/topic/2548/getting-started-with-cmd>

Chapter 2: Using xcopy command

Introduction

Xcopy, copies files and directories, including subdirectories.

Parameters

Parameter	Details
/h	Copies files with hidden and system file attributes. By default, xcopy does not copy hidden or system files.
/r	Copies read-only files.
/s	Copies directories and subdirectories, unless they are empty. If you omit /s, xcopy works within a single directory.
/y	Suppresses prompting to confirm that you want to overwrite an existing destination file.
/D	Orders CMD to only copy files that are newer than their destination opposite.

Examples

Copying multiple files including tree structure

If you want to xcopy files with specific type to a new folder keeping the current folder structure you need only to do this

```
xcopy [SourcePath]*.mp3 [DestinationPath] /sy
```

Read Using xcopy command online: <https://riptutorial.com/cmd/topic/5123/using-xcopy-command>

Credits

S. No	Chapters	Contributors
1	Getting started with cmd	Abood , antonio , Bassie , Community , David Morales , David Starkey , DavidPostill , hellyale , sarvajeetsuman , Stephen Leppik
2	Using xcopy command	pollirrata , Serhat E