

How do I run two commands in one line in Windows CMD?

Asked 8 years, 1 month ago Active 1 month ago Viewed 833k times



I want to run two commands in a Windows CMD console.

863

In Linux I would do it like this



```
touch thisfile ; ls -lstrh
```



184

How is it done on Windows?

windows

batch-file

command-line

cmd

edited May 29 '18 at 23:45



Soviut

71.8k

37

152

220

asked Nov 8 '11 at 18:31



flybywire

204k

176

372

488

17 Answers



Like this on all Microsoft OSes since 2000, and still good today:

1188

```
dir & echo foo
```



If you want the second command to execute only if the first exited successfully:



```
dir && echo foo
```

The single ampersand (&) syntax to execute multiple commands on one line goes back to Windows XP, Windows 2000, and some earlier NT versions. (4.0 at least. according to one commenter here.)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



Historical data follows, for those who may find it educational.

Prior to that, the && syntax was only a feature of the shell replacement 4DOS before that feature was added to the Microsoft command interpreter.

In Windows 95, 98 and ME, you'd use the pipe character instead:

```
dir | echo foo
```

In MS-DOS 5.0 and later, through some earlier Windows and NT versions of the command interpreter, the (undocumented) command separator was character 20 (Ctrl+T) which I'll represent with ^T here.

```
dir ^T echo foo
```

edited Oct 24 '17 at 2:02

answered Nov 8 '11 at 18:33



[djdanlib](#)

14.6k

1

15

24

26 At least down to NT 4. – [Joey](#) Nov 8 '11 at 18:37

17 Works on Win 8.0 and 8.1 as well. – [MEMark](#) Nov 25 '13 at 8:53

161 Just for the sake of clarity: you should note the single & symbol here is explicitly for a live command prompt. When commands are run by passing a string to be executed as a script (shortcuts, batch files, etc..), then double && are required. – [ZaLiTHkA](#) Apr 3 '14 at 20:28

6 @Fallenreaper Make sure that you are aware of the practical difference between the two: See Raihan's answer below. – [Moshe Katz](#) Jul 23 '14 at 19:30

6 Got ya. A && B, B only will run if A is successful, whereas A & B will run B after A, no matter what the outcome of A is. Thanks for the heads up – [Fallenreaper](#) Jul 25 '14 at 18:44



A quote from the documentation:

553

- Source: Microsoft, Windows XP Professional Product Documentation, [Command shell overview](#)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



You can run multiple commands from a single command line or script using conditional processing symbols. When you run multiple commands with conditional processing symbols, the commands to the right of the conditional processing symbol act based upon the results of the command to the left of the conditional processing symbol.

For example, you might want to run a command only if the previous command fails. Or, you might want to run a command only if the previous command is successful.

You can use the special characters listed in the following table to pass multiple commands.

- `& [...]`
`command1 & command2`
 Use to separate multiple commands on one command line. Cmd.exe runs the first command, and then the second command.
- `&& [...]`
`command1 && command2`
 Use to run the command following `&&` only if the command preceding the symbol is successful. Cmd.exe runs the first command, and then runs the second command only if the first command completed successfully.
- `|| [...]`
`command1 || command2`
 Use to run the command following `||` only if the command preceding `||` fails. Cmd.exe runs the first command, and then runs the second command only if the first command did not complete successfully (receives an error code greater than zero).
- `() [...]`
`(command1 & command2)`
 Use to group or nest multiple commands.
- `;` or `,`
`command1 parameter1;parameter2`
 Use to separate command parameters.

edited Oct 21 '18 at 10:35



I say Reinststate Monica

296 8 22

answered Nov 8 '11 at 18:36



Raihan

8,279 3 23 38

40 Try `cmd /c "echo foo & echo bar"` . – Raihan Nov 1 '13 at 17:44

2 `vagrant up && vagrant ssh` worked without quotation marks on Windows 10. – furman87 Oct 29 '15 at 0:12

Thanks! I didn't know about `;&` to make it synchronous and run both in parallel. I was stuck because I was using `;&` with `if` in synchronous and

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



- 1 documentation link broken, new location technet.microsoft.com/en-us/library/bb490954.aspx – [nkef](#) Jun 26 '17 at 12:14
- 6 @TetraDev & does not make the commands asynchronous or parallel. command1 & command2 will run command2 *after* command1 has completed. The success or failure of command1 is irrelevant. – [bornfromanegg](#) Sep 13 '17 at 9:50 ✎

& is the Bash equivalent for ; (run commands) and && is the Bash equivalent of && (run commands only when the previous has not caused an error).

71

edited Jul 15 '17 at 17:09

[Peter Mortensen](#)

24.3k 19 89 118

answered Nov 8 '11 at 18:37

[manojlds](#)

232k 53 403 379

- 1 this is also true for csh, tcsh and many more shells. I've never seen ; before in Linux – [phuciv](#) May 21 '14 at 10:05
- 9 @LưuVĩnhPhúc in sh-style shells, ; means to run the first command, wait for it to finish, then run the second command. & means to run the first command, put it to background, and run the second command. So both programs launch simultaneously. Note that these aren't combining symbols, they are trailing symbols to the first command; you can launch a single command in background with programme & without having a second command. – [M.M](#) Dec 17 '14 at 0:46 ✎

You can use & to run commands one after another. Example: `c:\dir & vim myFile.txt`

24

answered Nov 8 '11 at 18:34

[scrappedcola](#)

9,653 1 24 41

If you want to create a cmd shortcut (for example on your desktop) add /k parameter (/k means keep, /c will close window):

18

```
cmd /k echo hello && cd c:\ && cd Windows
```

answered Nov 12 '13 at 6:39

[TroodoN-Mike](#)

You can use *call* to overcome the problem of environment variables being evaluated too soon - e.g.

14

```
set A=Hello & call echo %A%
```

edited Mar 13 '16 at 11:14



Stephan

40.8k 5 36 65

answered Mar 13 '16 at 11:06



SSi

179 1 5

1 This works only if the variable is not already set, in which case the `echo` prints out the *old* value, not the new value. At least, this is what I observe on my Windows 7 box. – [Craig M. Brandenburg](#) Mar 21 '16 at 23:15

I use `set A=` first to make sure the variable is not set. – [SSi](#) May 11 '16 at 9:48

1 A better method is to use delayed expansion (`setlocal EnableDelayedExpansion` in a script or `cmd /v`). For example: `timeout 5 && cmd /v /c echo !TIME! && echo %TIME%` . – [Nick](#) Feb 1 '17 at 15:02

4 Use `^` after the first percent to get the new value: `set A=Hello & call echo %^A%` – [Nicolas](#) Mar 28 '17 at 21:00

A number of [processing symbols](#) can be used when running several commands on the same line, and may lead to processing redirection in some cases, altering output in other case, or just fail. One important case is placing on the same line commands that manipulate variables.

8

```
@echo off
setlocal enabledelayedexpansion
set count=0
set "count=1" & echo %count% !count!
```

```
0 1
```

As you see in the above example, when commands using variables are placed on the same line, you must use delayed expansion to update your variable values. If your variable is indexed, use `CALL` command with `%%` modifiers to update its value on the same line:

```
set "i=5" & set "arg%i!=MyFile%i!" & call echo path%i!=%temp%\%%arg%i!%%
```

```
path5=C:\Users\UserName\AppData\Local\Temp\MyFile5
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).





Graham

5,112

14

46

65



sambul35

960

11

17

given example didn't work for me but below one did: `cmd /V:ON /c "set i=5 & set arg!i!=MyFile!i! & echo path!i!=%temp%\%arg!i!%"` – [sactiw](#)
May 10 '17 at 17:10

7

So, I was trying to enable the specific task of running `RegAsm` (register assembly) from a context menu. The issue I had was that the result would flash up and go away before I could read it. So I tried piping to `Pause`, which does not work when the command fails (as mentioned here [Pause command not working in .bat script](#) and here [Batch file command PAUSE does not work](#)). So I tried `cmd /k` but that leaves the window open for more commands (I just want to read the result). So I added a `pause` followed by `exit` to the chain, resulting in the following:

```
cmd /k C:\Windows\Microsoft.NET\Framework\v4.0.30319\regasm.exe "%1" /codebase \"%1\" & pause & exit
```

This works like a charm -- `RegAsm` runs on the file and shows its results, then a "Press any key to continue..." prompt is shown, then the command prompt window closes when a key is pressed.

P.S. For others who might be interested, you can use the following `.reg` file entries to add a `dllfile` association to `.dll` files and then a `RegAsm` command extension to that (notice the escaped quotes and backslashes):

```
[HKEY_CLASSES_ROOT\.dll]
"Content Type"="application/x-msdownload"
@="dllfile"

[HKEY_CLASSES_ROOT\dllfile]
@="Application Extension"

[HKEY_CLASSES_ROOT\dllfile\Shell\RegAsm]
@="Register Assembly"

[HKEY_CLASSES_ROOT\dllfile\Shell\RegAsm\command]
@="cmd /k C:\\Windows\\Microsoft.NET\\Framework\\v4.0.30319\\regasm.exe \"%1\" /codebase
 \"%1\" & pause & exit"
```

Now I have a nice right-click menu to register an assembly.

edited May 23 '17 at 12:26

answered Jan 26 '15 at 20:35

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



[lifehacker.com/...](#) – Anthon Dec 23 '16 at 7:51

cmd /c ipconfig /all & Output.txt

6

This command execute command and open `Output.txt` file in a single command

edited Dec 17 '14 at 0:44



Jeremy Collette

42 6

answered Apr 16 '14 at 5:41



dpp.2325

101 1 3

I try to have two pings in the same window, and it is a serial command on the same line. After finishing the first, run the second command.

4

The solution was to combine with `start /b` on a Windows 7 command prompt.

Start as usual, without `/b`, and launch in a separate window.

The command used to launch in the same line is:

```
start /b command1 parameters & command2 parameters
```

Any way, if you wish to parse the output, I don't recommend to use this. I noticed the output is scrambled between the output of the commands.

edited Jul 15 '17 at 17:13



Peter Mortensen

24.3k 19 89 118

answered Mar 18 '17 at 11:16



sdcxp

41 1

In order to execute two commands at the same time, you must put an `&` (ampersand) symbol between the two commands. Like so:

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



Cheers!

answered Dec 13 '16 at 4:41



PryroTech

433 2 10

Well, you have two options: Piping, or just & :

4

```
DIR /S & START FILE.TXT
```

Or,

```
tasklist | find "notepad.exe"
```

Piping (|) is more for taking the output of one command, and putting it into another. And (&) is just saying run this, and that.

edited Jul 15 '17 at 17:14



Peter Mortensen

24.3k 19 89 118

answered Apr 29 '17 at 23:16



PyDever

78 1 9

Could you edit your post to include some explanation what does & and | do and how they differ, if at all? Right now people unfamiliar with these concepts are unable to decide for themselves which one should be used in their cases. – Mirek Długosz Apr 29 '17 at 23:41

As I was. My bad. I will edit the post immediately. Appreciate it. – PyDever May 2 '17 at 6:43

Should be "findstr" instead of just "find" – Anu Shubin Joseph Raj Jul 1 at 8:31

No, `cd / && tree && echo %time%`. The time echoed is at when the first command is executed.

2

The piping has some issue, but it is not critical as long as people know how it works.

edited Jul 15 '17 at 17:11

answered Jul 22 '16 at 8:56

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



▲ **It's simple:** just differentiate them with `&&` signs. Example:

1 `echo "Hello World" && echo "GoodBye World".`

▼ *"Goodbye World" will be printed after "Hello World".*

edited Nov 6 at 16:04



scientist_7
362 2 16

answered Jun 5 '17 at 15:06



Rajan Dhanowa
19 2

3 Independent of the fact, that there are already old answers that shows the same, it's still not quite correct. `&&` is a **conditional** operator, the next command is only executed when the first command succeeded (errorlevel=0) – [jeb](#) Jun 7 '17 at 8:04

of course. it's self evident if the person wanna run two commands those two will be correct and thus everything will go good – [Rajan Dhanowa](#) Jun 7 '17 at 18:19

1 Rajan: he means "the next command is only executed if the first command succeeds". – [Ivan](#) Jun 23 '17 at 15:44

2 *"Goodbye World will be printed after Hello World"* provided printing `HeLlO WorlD` did not fail. as [@jeb](#) has said, `&&` is conditional. `&` runs commands regardless if the previous was successful or not. – [Pranav A.](#) Aug 19 '17 at 23:55

▲ One more example: For example, when we use the `gulp` build system, instead of

1 `gulp - default > build`

▼ `gulp build - build build-folder`

`gulp watch - start file-watch`

`gulp dist - build dist-folder`

We can do that with one line:

```
cd c:\xampp\htdocs\project & gulp & gulp watch
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).





Peter Mortensen

24.3k 19 89 118



Mile Mijatovic

1,527 14 30

When you try to use or manipulate variables in one line beware of their content! E.g. a variable like the following

1

```
PATH=C:\Program Files (x86)\somewhere;"C:\Company\Cool  
Tool";%USERPROFILE%\AppData\Local\Microsoft\WindowsApps;
```

may lead to a lot of unhand-able trouble if you use it as %PATH%

1. The closing parentheses terminate your group statement
2. The double quotes don't allow you to use %PATH% to handle the parentheses problem
3. And what will a referenced variable like %USERPROFILE% contain?

edited Sep 23 '17 at 10:19

user6250760

answered Aug 11 '17 at 22:22



V1510N

395 3 14

2 What has this to do with running two commands in one line in Windows CMD? – Peter Mortensen Dec 19 '17 at 0:56

Yes there is. It's & .

0

&& will execute command 2 when command 1 is complete providing it didn't fail.

& will execute regardless.

edited Nov 6 at 16:56



scientist_7

362 2 16

answered Apr 19 at 3:29



هادی کشاورز Hadi

Keshavarz

35 5



answer activity.