

Aliases in Windows command prompt



I have added `notepad++.exe` to my Path in Environment variables.

293

Now in command prompt, `notepad++.exe filename.txt` opens the `filename.txt`. But I want to do just `np filename.txt` to open the file.



I tried using `DOSKEY np=notepad++`. But it is just bringing to the forefront an already opened notepad++ without opening the file. How can I make it open the file?



160

Thanks.

windows

alias

command-prompt

edited Jul 23 '18 at 15:50



brian d foy

104k

30

176

484

asked Dec 11 '13 at 22:11



Romonov

2,711

8

33

48

2 Change the executable name to anything you want. You can do this from the File Explorer! – [IgorGanapolsky](#) Mar 31 '15 at 18:22

7 Boy, howdy, isn't cmd awful compared to BASH, where doing stuff like this is a breeze. Oy vey. – [ncmathsadist](#) Mar 8 '18 at 21:13

14 Answers



To add to josh's answer,

354

you may make the alias(es) *persistent* with the following steps,



1. Create a `.bat` or `.cmd` file with your `DOSKEY` commands.

2. Run `regedit` and go to `HKEY_CURRENT_USER\Software\Microsoft\Command Processor`



3. Add String Value entry with the name `AutoRun` and the *full* path of your `.bat/.cmd` file.

For example, `%USERPROFILE%\alias.cmd`, replacing the initial segment of the path with `%USERPROFILE%` is useful for syncing among multiple machines.

This way, every time `cmd` is run, the aliases are loaded.

For completeness, here is a template to illustrate the kind of aliases one may find useful.

```
@echo off
```

```
:: Temporary system path at cmd startup
```

```
set PATH=%PATH%; "C:\Program Files\Sublime Text 2\"
```

```
:: Add to path by command
```

```
DOSKEY add_python26=set PATH=%PATH%; "C:\Python26\"
```

```
DOSKEY add_python33=set PATH=%PATH%; "C:\Python33\"
```

```
:: Commands
```

```
DOSKEY ls=dir /B
DOSKEY sublime=sublime_text $*
    ::sublime_text.exe is name of the executable. By adding a temporary entry to system
    path, we don't have to write the whole directory anymore.
DOSKEY gsp="C:\Program Files (x86)\Sketchpad5\GSP505en.exe"
DOSKEY alias=notepad %USERPROFILE%\Dropbox\alias.cmd

:: Common directories

DOSKEY dropbox=cd "%USERPROFILE%\Dropbox\%"
DOSKEY research=cd %USERPROFILE%\Dropbox\Research\
```

- Note that the `$*` syntax works after a directory string as well as an executable which takes in arguments. So in the above example, the user-defined command `dropbox research` points to the same directory as `research`.
- As Rivenfall pointed out, it is a good idea to include a command that allows for convenient editing of the `alias.cmd` file. See `alias` above. If you are in a `cmd` session, enter `cmd` to restart `cmd` and reload the `alias.cmd` file.

When I searched the internet for an answer to the question, somehow the discussions were either focused on persistence only or on some usage of DOSKEY only. I hope someone will benefit from these two aspects being together here!

Here's a `.reg` file to help you install the `alias.cmd`. It's set now as an example to a dropbox folder as suggested above.

Windows Registry Editor Version 5.00

```
[HKEY_CURRENT_USER\Software\Microsoft\Command Processor]
"AutoRun"="%USERPROFILE%\alias.cmd"
```

edited Oct 16 '18 at 5:24



TwiN

1,873 1 8 21

answered Jan 10 '14 at 9:43



Argyll

4,085 3 11 31

-
- 2 exactly what I needed; works perfectly; I recommend adding a doskey to actually edit the `env.cmd` file – [Rivenfall](#) Jan 7 '15 at 14:09
-
- 7 This is naive and inefficient. The autorun batch file will be run for every instance of `cmd.exe`, including the `system` function. It needs to exit if a certain variable (e.g. `AUTORUN`) is defined. Otherwise set up the environment (`set AUTORUN=1`) and set up doskey in a single pass using the macrofile option instead of running `doskey.exe` to define each alias. – [eryksun](#) Aug 2 '15 at 9:27
-
- 7 Put it in answer in a different post? – [Argyll](#) Aug 2 '15 at 11:17
-
- 4 @eryksun can you post or link to a less naive example? I know how to exit if autorun is defined, but confused over how to set or unset it in the first place without having already run CMD. – [matt wilkie](#) Sep 6 '17 at 18:23
-
- 3 Suddenly had flashbacks about `autoexec.bat` – [aaaantoine](#) Nov 16 '17 at 15:33
-

You need to pass the parameters, try this:

```
doskey np=notepad++.exe $*
```

Edit (responding to Romonov's comment) Q: *Is there any way I can make the command prompt remember so I don't have to run this each time I open a new command prompt?*

`doskey` is a textual command that is interpreted by the command processor (e.g. `cmd.exe`), it can't know to modify state in some other process (especially one that hasn't started yet).

People that use `doskey` to setup their initial command shell environments typically use the `/K` option (often via a shortcut) to run a batch file which does all the common setup (like- set window's title, colors, etc).

```
cmd.exe /K env.cmd
```

`env.cmd`:

```
title "Foo Bar"
doskey np=notepad++.exe $*
...
```

edited Apr 15 '17 at 1:03



Leigh

27.8k 7 46 90

answered Dec 11 '13 at 23:01



josh poley

5,966 1 17 24

-
- 7 This works for the command prompt in which I run this command. But if I close the window and open a new command prompt. It doesn't remember the `np` command. Is there any way I can make the command prompt remember so I don't have to run this each time I open a new command prompt? – [Romonov](#) Dec 13 '13 at 21:37
-
- 5 Same behavior without changing the PATH: `doskey npp="C:\Program Files (x86)\Notepad++\notepad++.exe" $*` – [Matt Bierner](#) Oct 9 '14 at 20:23
-
- 4 `doskey.exe` has nothing to do with `cmd.exe`. It sets an alias for the current or a specified executable in the console window, which is hosted by an instance of `conhost.exe`. Console aliases are matched and substituted at the beginning of a line of input before the client application (e.g. `cmd.exe` or `powershell.exe`) reads the line. They can't be used generically as commands, e.g. not in batch files or piped into. – [eryksun](#) Aug 2 '15 at 9:31
-

If you're just going for some simple commands, you could follow these steps:

84

1. Create a folder called **C:\Aliases**
2. Add **C:\Aliases** to your path (so any files in it will be found every time)
3. Create a **.bat** file in C:\Aliases for each of the aliases you want

Maybe overkill, but unlike the (otherwise excellent) answer from [@Argyll](#), this solves the problem of this loading every time.

For instance, I have a file called **dig2.bat** with the following in it:

```
@echo off
echo.
dig +noall +answer %1
```

Your **np** file would just have the following:

```
@echo off
echo.
notepad++.exe %1
```

FWIW, I have about 20 aliases (separate .bat files) in my C:\Aliases directory - I just create new ones as necessary. Maybe not the neatest, but it works fine.

UPDATE: Per an excellent suggestion from user @Mav, it's even better to use **%*** rather than **%1**, so you can pass multiple files to the command, e.g.:

```
@echo off
echo.
notepad++.exe %*
```

That way, you could do this:

```
np c:\temp\abc.txt c:\temp\def.txt c:\temp\ghi.txt
```

and it will open all 3 files.

edited Oct 4 '18 at 16:20

answered Sep 12 '16 at 21:57



[roryhewitt](#)

1,886 1 18 26

16 This was the most simple solution that covered all my needs. – [arthropod](#) Apr 5 '17 at 23:38

3 I personally do use this method for a long time. This is one such very easy to do method if someone does not want to go on the way of doskey . – [bantya](#) Mar 6 '18 at 16:52

4 One advantage of this method (cmd files as aliases) is that if you use WSL (Linux subsystem for Windows) these cmds are available in bash as well. Though often (depending what you are aliasing) you need to do some path manipulation using wslpath.sh or similar before you call the cmd file – [Alex Perry](#) Apr 14 '18 at 19:05

1 @roryhewitt Oh yes, specifying the full path to the ls.exe should definitely work. Good point. I have used the doskey alternative though. – [Qwerty](#) Sep 12 '18 at 18:36

2 Might want to add %* instead of %1 to pass in all the arguments as opposed to just the first. – [Mav](#) Oct 4 '18 at 3:07



29



Given that you added notepad++.exe to your PATH variable, it's extra simple. Create a file in your System32 folder called np.bat with the following code:

```
@echo off
call notepad++.exe %*
```

The %* passes along all arguments you give the np command to the notepad++.exe command.

EDIT: You will need admin access to save files to the System32 folder, which was a bit wonky for me. I just created the file somewhere else and moved it to System32 manually.

answered Apr 13 '15 at 17:58

[Velixo](#)



I already have a folder of little .bat utility files so I like this better than messing with the registry or a .cmd file – [Rich](#) Apr 21 '16 at 14:04

- 1 I just tried that and it doesn't work identically to calling Notepad++ directly. If you use wildcards in the filename you're opening, and call NPP directly, it works, e.g. you can do "notepad++.exe *somefiles*" and matching files will open. When I tried that with your suggested batch file, i.e. "npp *somefiles*", it did open an NPP instance but did not open the files I passed. Any thoughts? – [SSilk](#) Aug 31 '16 at 13:58

Alternatively you can use [cmdr](#) which lets you add aliases just like linux:

27

```
alias subl="C:\Program Files\Sublime Text 3\subl.exe" %*
```

answered Nov 24 '16 at 10:24



[Nishant Shreshth](#)

12.1k 3 32 33

- 7 Although this doesn't answer the question. It is important for people from a Linux background to understand that there is an alternative to Windows CMDs that can suit their immediate needs. – [JosephMCasey](#) Mar 31 '17 at 23:12

- 1 There are quite a few alternatives. Git for Windows comes with one, "git-bash". – [cowlinator](#) Dec 21 '18 at 23:54

To add alias in Cmder, see an example [here](#). – [jdhao](#) Feb 1 at 11:49

Also, you can create an alias.cmd in your path (for example C:\Windows) with the command

14

```
@echo %2 %3 %4 %5 %6 > %windir%\%1.cmd
```

Once you do that, you can do something like this:

```
alias nameOfYourAlias commands to run
```

And after that you can type in command line

```
nameOfYourAlias
```

this will execute

```
commands to run
```

BUT the best way for me is just adding the path of a program.

```
setx PATH "%PATH%;%ProgramFiles%\Sublime Text 3" /M
```

And now I run sublime as



Console Aliases in Windows 10

9

To define a console alias, use `Doskey.exe` to create a macro, or use the `AddConsoleAlias` function.

doskey

```
doskey test=cd \a_very_long_path\test
```

To also pass [parameters](#) add `$*` at the end: `doskey short=longname $*`

AddConsoleAlias

```
AddConsoleAlias( TEXT("test"),
                  TEXT("cd \\<a_very_long_path>\\test"),
                  TEXT("cmd.exe"));
```

More information here [Console Aliases](#), [Doskey](#), [Parameters](#)



You want to create an alias by simply typing:

6

```
c:\>alias kgs kubectl get svc
```

```
Created alias for kgs=kubectl get svc
```

And use the alias as follows:

```
c:\>kgs alfresco-svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
alfresco-svc	ClusterIP	10.7.249.219	<none>	80/TCP	8d

Just add the following `alias.bat` file to you path. It simply creates additional batch files in the same directory as itself.

```
@echo off
echo.
for /f "tokens=1,* delims= " %%a in ("%*") do set ALL_BUT_FIRST=%%b
echo @echo off > C:\Development\alias-script\%1.bat
echo echo. >> C:\Development\alias-script\%1.bat
```

```
echo %ALL_BUT_FIRST% %* >> C:\Development\alias-script\%1.bat
echo Created alias for %1=%ALL_BUT_FIRST%
```

An example of the batch file this created called `kgs.bat` is:

```
@echo off
echo.
kubectl get svc %*
```

answered Sep 7 '18 at 13:15



Steaton

69 1 5

6

Actually, I'll go you one better and let you in on a little technique that I've used since I used to program on an Amiga. On *any* new system you use, be it personal or professional, step one is to create two folders: `C:\BIN` and `C:\BATCH`. Then modify your path statement to put both at the start in the order `C:\BATCH;C:\BIN`;[rest of path].

Having done that, if you have little out-of-the-way utilities that you need access to simply copy them to the `C:\BIN` folder and they're in your path. To temporarily override these assignments, you can add a batch file with the same name as the executable to the `C:\BATCH` folder and the path will find it before the file in `C:\BIN`. It should cover anything you might ever need to do.

Of course, these days the canonical correct way to do this would be to create a symbolic junction to the file, but the same principle applies. There is a little extra added bonus as well. If you want to put something in the system that conflicts with something already in the path, putting it in the `C:\BIN` or `C:\Batch` folder will simply *pre-empt* the original - allowing you to override stuff either temporarily or permanently, or rename things to names you're more comfortable with - without actually altering the original.

edited Jun 8 at 13:16



SherylHohman

6,187 8 43 53

answered Nov 8 '16 at 19:16



David

71 2 1

1 This is the same answer as roryhewitt. – Jean-François Fabre ♦ Nov 8 '16 at 19:23

2 Actually, no it's not. I said 'one better'. The built-in option to override or underide an override that's already in place. The simple segregation of executables from batch files. And rory's solution does not specify *where* in the path the folder should go. Most will therefore put it at the end of the path. Being at the end instead of the beginning, his solution will not allow overrides in the first place. Rory's solution is approximately the same as the solution I myself originally arrived at - 25 years ago. I've refined the model somewhat since then. – David Nov 22 '16 at 15:54

Whatever. People using an Amiga cannot be all bad. – Jean-François Fabre ♦ Nov 23 '16 at 20:50

Fair enough :) In my case, the Aliases folder IS at the beginning of the path, but in any case, I personally don't want to override the default - my aliases always have different names. So I use 'dig2' and 'digx' as aliases to 'dig', but still have 'dig' available (without needing to specify its folder). Also +1 for Amiga :) – roryhewitt Jan 12 '17 at 18:52 ✎

Expanding on [roryhewitt](#) answer.

4

An advantage to using `.cmd` files over DOSKEY is that these "aliases" are then available in other shells such as PowerShell or WSL (Windows subsystem for Linux).

▼ The only gotcha with using these commands in bash is that it may take a bit more setup since you might need to do some path manipulation before calling your "alias".

eg I have vs.cmd which is my "alias" for editing a file in Visual Studio

```
@echo off
if [%1]==[] goto nofiles
start "" "c:\Program Files (x86)\Microsoft Visual Studio
11.0\Common7\IDE\devenv.exe" /edit %1
goto end
:nofiles
start "" "C:\Program Files (x86)\Microsoft Visual Studio
11.0\Common7\IDE\devenv.exe" "[PATH TO MY NORMAL SLN]"
:end
```

Which fires up VS (in this case VS2012 - but adjust to taste) using my "normal" project with no file given but when given a file will attempt to attach to a running VS opening that file "within that project" rather than starting a new instance of VS.

For using this from bash I then add an extra level of indirection since "vs Myfile" wouldn't always work

```
alias vs='/usr/bin/run_visual_studio.sh'
```

Which adjusts the paths before calling the vs.cmd

```
#!/bin/bash
cmd.exe /C 'c:\Windows\System32\vs.cmd' "`wslpath.sh -w -r $1`"
```

So this way I can just do

```
vs SomeFile.txt
```

In either a command prompt, Power Shell or bash and it opens in my running Visual Studio for editing (which just saves my poor brain from having to deal with VI commands or some such when I've just been editing in VS for hours).

answered Apr 14 '18 at 20:02



Alex Perry

115 7

▲ This solution is not an apt one, but serves purpose in some occasions.

2

▼ First create a folder and add it to your system path. Go to the executable of whatever program you want to create alias for. Right click and send to Desktop(Create Shortcut). Rename the shortcut to whatever alias name is comfortable. Now, take the shortcut and place in your folder.

From run prompt you can type the shortcut name directly and you can have the program opened for you. But from command prompt, you need to append **.lnk** and hit enter, the program will be opened.

answered Feb 13 '18 at 7:49



BarathVutukuri

414 3 14



Since you already have `notepad++.exe` in your path. Create a shortcut in that folder named `np` and point it to `notepad++.exe` .

answered Jan 24 '17 at 12:21



[Shravan](#)
36 6



First, you could create a file named `np.cmd` and put it in the folder which in PATH search list. Then, edit the `np.cmd` file as below:

```
@echo off
notepad++.exe
```

answered Oct 11 '17 at 9:20



[迷茫的量子](#)
1



Using doskey is the right way to do this, but it resets when the Command Prompt window is closed. You need to add that line to something like `.bashrc` equivalent. So I did the following:

1. Add "C:\Program Files (x86)\Notepad++" to system path variable
2. Make a copy of `notepad++.exe` (in the same folder, of course) and rename it to `np.exe`

Works just fine!

answered Sep 13 '16 at 18:08



[balajimc55](#)
778 2 7 13

protected by [Andy ♦](#) Oct 12 '17 at 1:22

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus](#) does not count).

Would you like to answer one of these [unanswered questions](#) instead?