

Super User is a question and answer site for computer enthusiasts and power users. Join them; it only takes a minute:

[Sign up](#)

Here's how it works:

Anybody can ask a question

Anybody can answer

The best answers are voted up
and rise to the top



Is there windows equivalent to the .bashrc file in linux?

▲ Just discovered doskey emacs=emacs -nw \$*, but without something like .bashrc, I'll have to type that every time I start a shell, right?

61

[windows](#)

[bashrc](#)



21

edited Feb 22 '16 at 15:34



[strupo](#)

7 3

asked May 23 '10 at 10:44

[herrturtur](#)

migrated from [stackoverflow.com](#) May 23 '10 at 11:10

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

1 [This question](#) isn't identical, but the answer covers your scenario. – [Marcelo Cantos](#) May 23 '10 at 10:51

7 Answers

▲ This is a very good question. I found [this](#). I suppose you could make a cmd script and have it run when starting cmd :-?

63

▼ ; **Run** a command when CMD.exe starts
[HKEY_LOCAL_MACHINE\Software\Microsoft\Command **Processor**]
"AutoRun"=-

EDIT: I just tried it. I have `AutoRun=C:\mini\bashrc.cmd` and `bashrc.cmd` is

```
@echo off
set TEST_VAR=something
```

when I start cmd and enter `echo %TEST_VAR%` it says `something` . So it works :)

answered May 23 '10 at 10:53



[nc3b](#)

889 1 6 10

9 Small snippet I use: `doskey ♦=exit` (where ♦ is `[char]4`). Enables you to close the shell with Ctrl+D, Enter. – [Joey](#) May 23 '10 at 12:13

This work great, nice job nc3b. – [Robert S Ciaccio](#) Jul 28 '10 at 3:57

2 HKEY_CURRENT_USER\Software\Microsoft\Command Processor with AutoRun=%HOMEPATH%\bashrc.bat So every user would have his own startup-sequence – [bambam2174](#) Jul 18 '13 at 8:29 ✎

5 I must be an idiot but where do I put this code? – [Marcel](#) Oct 10 '14 at 23:46

1 probably easiest to set to... `AUTORUN=%USERPROFILE%\autorun.cmd` so that it will use one for the current user... cool trick though, will try it when I get home... I mostly use bash lately, but this would help. – [Tracker1](#) Sep 18 '15 at 22:40

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

20

```
cmd.exe /k "%HOMEDRIVE%\%HOMEPATH%\cmd-startup.bat"
```

cmd-startup.bat in your home dir would then work just like .bashrc

answered May 23 '10 at 10:56



Martin

451 3 6



Modification of mc3b's answer for those who are windows noobs like myself.

13

Make a file in C:\bashrc.cmd or wherever you want your "rc" file to reside.



```
@echo off
set TEST_VAR=something
```

Run regedit by searching for it or running it in cmd.

In the folders on the left navigate to:

HKEY_LOCAL_MACHINE\Software\Microsoft\Command Processor

Right click on the Command Processor folder and click New -> String Value . Enter the key AutoRun . Double click on AutoRun and enter the bashrc location C:\bashrc.cmd .

Test: Open a new cmd window and type:

```
echo %TEST_VAR%
```

if it says something , it works.

answered May 19 '15 at 2:32



Humdinger

245 2 11

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

5

Assuming your shell is Powershell, not cmd, you can create a *profile* for your shell. Such profiles basically are scripts which are executed when you start a shell. Just like `.bashrc`. An important part of [this post](#) might be the following:

The locations (on Windows Vista) of the profiles for the powershell.exe host are as follows:

- `%windir%\system32\WindowsPowerShell\v1.0\profile.ps1`
This is for all users of the computer and for all shells.
- `%windir%\system32\WindowsPowerShell\v1.0\Microsoft.PowerShell_profile.ps1`
This is for all users of the computer, but it is only for the Microsoft.PowerShell shell.
- `%UserProfile%\Documents\WindowsPowerShell\profile.ps1`
This is for the current user only and all shells.
- `%UserProfile%\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1`
This is for the current user only and only for the Microsoft.PowerShell shell.

These profiles aren't created by default. They exist only if you create them.

It seems to be the same on Windows 7, 8 and 10. You can also find more information [here](#).

Note: You may need to run the following. Otherwise, your profile may not be executed.

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

You can also run the following as administrator to change the configuration for all users.

```
Set-ExecutionPolicy RemoteSigned
```

edited Sep 27 '18 at 18:48



Community ♦

1

answered Feb 26 '16 at 4:49



JojoatXGME

151 1 4

EDIT: Found this: [http://msdn.microsoft.com/en-us/library/bb613488\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb613488(VS.85).aspx)

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

NOTE: You will need to have PowerShell script execution enabled. If this is disabled, you'll need to run an elevated PowerShell and execute `Set-ExecutionPolicy Unrestricted`. Obviously, this is not recommended for security reasons, but there ya go.

```
powershell.exe -noexit -File "%HOMEDRIVE%\%HOMEPATH%\ps-startup.ps1"
```

Have a file named `ps-startup.ps1` in your user directory and your shortcut will execute it as a PowerShell file on startup. Just run PowerShell with this shortcut from here on out.

edited Jan 13 '15 at 18:55

answered Jan 13 '15 at 18:44



lytedev

31 2

This can be achieved in a "better" way through powershell profiles. You can find the location of your current profile by: `echo $profile`. You can add lines to that file, and that should do it for future sessions. For existing sessions, you can reload the profile with: `& $profile`. – ashic Dec 16 '15 at 10:54 ✎

Basically nc3b's answer works, but Chocolatey's [DevBox-Common](#) package will **make it easier**.

3

1. If you have not already installed Chocolatey, [install it](#).
2. Install DevBox-Common package by `choco install devbox-common -y`
 - You should type this command inside **administrative** `cmd.exe` or PowerShell.
3. Now you have `.bashrc.bat` and `.bashrc.include.aliases-common.bat` in your home path. Every file matching `.bashrc.include.*.bat` will be executed at every command shell start.

answered May 12 '16 at 13:44



hakatashi

131 3

To all people coming from a unix-like world and landing on this question (i.e. everybody, since it asks about `.bashrc`): **Radically improve your Windows console life and stop using stock `cmd.exe` and the default Windows terminal!! No one asking this question, or looking at answers to this question, should be using the current default `cmd.exe` and terminal!**

0

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

Install and use [Cmder](#) and your `.bashrc` equivalent becomes `%CMDER_ROOT%\config\user_profile.cmd`, while also *significantly* improving your shell/console experience.

Opinions, extensions, and loosely related recommendations follow...

Cmder is a metapackage that packages up [ConEmu](#), [Clink](#), and a pile of extra settings and stuff. ConEmu is the terminal emulator that provides the startup script capability, which (through the [default ConEmu settings Cmder provides](#)) ends up routing to the indicated `%CMDER_ROOT%\config\user_profile.cmd`.

You may want to just explicitly use ConEmu and Clink independently, rather than use the bundled Cmder package.

To do that, and to add in some more "gotta work on windows, what should I do" bliss, below are some good/opinionated things to do. These are admittedly VERY loosely related to the initial question, but I've recently been through a Windows setup again, have typed up most of this already, and the target audience in this question is correct, so here you go...

1. Use [scoop.sh](#) for *all* your installs

- scoop is a great package manager for windows (similar to apt/yum/pacman/emerge/whatever)... where all packages install to the user (with no admin privileges needed)
- definitely install the `extras` bucket for many non-console things (`scoop bucket add extras`)
- you can also use [chocolatey](#), but (my opinion) only if under duress and the thing you want is not in a scoop bucket
- or just make your own scoop manifest
- it is sad day when the thing I want isn't in a scoop bucket :(
- unlike scoop, chocolatey needs admin privileges for its installs

2. Use [Clink](#) as your shell instead of `cmd.exe`

- `scoop install clink`
- gives you GNU readline goodness (like ctrl-r, ctrl-s cmd history navigation) and more
- be sure to also `clink autorun install` to make it run when `cmd.exe` is run

3. Use [ConEmu](#) as your terminal emulator

- `scoop install conemu`
- the Windows terminal is *terrible* (but they are [working on it](#))
- be sure to configure it as the default terminal

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

- SO MANY other configuration possibilities
 - if overwhelmed, maybe just stick with [Cmder](#) and its reasonable defaults (`scoop install cmder`), but I personally think it is cleaner to install/run Clink and ConEmu separately.

4. Install [Gow](#) (like, *right now*)

- `scoop install gow`
- > [100 GNU tools](#) (`grep` , `vim` , `sed` , `ls` , `cat` , `curl` , `wc` , `less` , `find` , etc, etc) you know and love, directly in your `cmd.exe` - like terminal without non-native weirdness
- watch for name-collision renames like `find->gfind` and `awk->gawk`
- I use Gow pretty much exclusively for basic Windows work instead of mucking with WSL, Cygwin, MSYS, git bash, etc.
- Gow tools get as close to the GNU/*nix-on-Windows as I've experienced so far while staying native to Windows, without the headaches of Cygwin environments, WSL filesystem nastiness (which also [being worked on](#)), gitbash windows, etc

5. Install other handy tools as you see fit (with scoop)

- the [main](#) and [extras](#) scoop buckets have *many* more good tools and applications (python, openssh, ripgrep, fzf, sysinternals, sublime-text, meld, ccleaner, etc)
 - although definitely not a GNU/unix thing, [sysinternals](#) is an essential/classic collection of Windows native tools.
- use `scoop search` and `scoop info` to find the ones you want, or browse the buckets (there are [several buckets](#), not just main and extras)

answered Jun 7 at 23:00



Russ

292 3 6