

Git for Windows: .bashrc or equivalent configuration files for Git Bash shell

212 I've just installed Git for Windows and am delighted to see that it installs Bash.

I want to customise the shell in the same way I can under Linux (e.g. set up aliases like `ll for ls -l`), but I can't seem to find `.bashrc` or equivalent configuration files.

What should I be editing?



41

windows

git

git-bash

edited Nov 23 '18 at 18:12



Peter Mortensen

14.2k 19 88 114

asked Jul 30 '11 at 14:27



iftheshoefritz

2,274 2 27 33

1 Read this for more recent versions : <https://stackoverflow.com/questions/32186840/git-for-windows-doesnt-execute-my-bashrc-file> – Victor Petit Mar 16 '18 at 14:35

7 Answers

267 Create a `.bashrc` file under `~/.bashrc` and away you go. Similarly for `~/.gitconfig`.

`~` is usually your `C:\Users\<your user name>` folder. Typing `echo ~` in the Git Bash terminal will tell you what that folder is.

If you can't create the file (e.g. running Windows), run the below command:



```
copy > ~/.bashrc
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook

-
- 3 It's a file, not a folder. ~/.bashrc is in your home directory, so it should be the directory you're in as soon as you start the git bash shell. just create a file called .bashrc and .gitconfig there. – [Charles Ma](#) Nov 11 '11 at 23:47
-
- 12 To create the .bashrc file you can execute the following from DOS prompt: copy > .bashrc. It creates the file with "The syntax of the command is incorrect." but you can edit it now. – [tmorell](#) May 26 '13 at 15:49
-
- 25 Just type notepad ~/.bashrc from the bash prompt and notepad will be launched with this file opened or will ask to create it, if it doesn't exist. If you're comfortable to vim, just type vim ~/.bashrc . It worked nicely to me. – [Gerardo Lima](#) Jul 10 '13 at 9:32
-
- 9 There are a trick in windows to create files like .profile (without letters before dot): right click -> create new file -> text file -> name it .profile. (NOTE THE DOT at the end) -> hit enter – [Denis](#) Apr 17 '15 at 9:01
-
- 10 You can also simply type touch ~/.bashrc – [Chiel ten Brinke](#) Oct 7 '15 at 7:56
-



92



In newer versions of Git for Windows, Bash is started with `--login` which causes Bash to not read `.bashrc` directly. Instead it reads `.bash_profile`.

If this file does not exist, create it with the following content:

```
if [ -f ~/.bashrc ]; then . ~/.bashrc; fi
```

This will cause Bash to read the `.bashrc` file. From my understanding of [this issue](#), Git for Windows should do this automatically. However, I just installed version 2.5.1, and it did not.

edited Aug 8 '18 at 16:25



[Peter Mortensen](#)

14.2k 19 88 114

answered Sep 2 '15 at 12:24



[harsel](#)

1,225 9 10

-
- 5 Thanks. This was exactly what was happening with my setup. – [jmk2142](#) Sep 6 '15 at 21:06
-
- Indeed, that's what happened to me recently after upgrade to 2.5. Thanks for your answer. – [Vincent](#) Sep 10 '15 at 3:17
-
- 1 Thank you so much, you save my day!!! – [Steve Lam](#) Sep 12 '15 at 10:44
-

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google





I had to add a user environment variable, `HOME`, with `C:\Users\<your user name>` by going to *System, Advanced System Settings*, in the *System Properties* window, the *Advanced* tab, *Environment Variables*...

6



Then in my `C:\Users\<your user name>` I created the file `.bashrc`, e.g., `touch .bashrc` and added the desired aliases.

edited Aug 8 '18 at 16:23



Peter Mortensen

14.2k 19 88 114

answered Sep 30 '14 at 5:35



Stan S.

107 2 8

add the user environment variable `HOME` and then create a `.bashrc` file with the required contents for example: `#!/bin/bash export TERM=msys -`
[snassr](#) May 21 '15 at 14:31



I think the question here is how to find `.bashrc` file on Windows.

5



Since you are using Windows, you can simply use commands like

```
start .
```

OR

```
explorer .
```

to open the window with the root directory of your Git Bash installation where you'll find the `.bashrc` file. You may need to create one if it doesn't exist.

You can use Windows tools like Notepad++ to edit the file instead of using Vim in your Bash window.

edited Aug 8 '18 at 16:38



Peter Mortensen

14.2k 19 88 114

answered Jun 15 '16 at 0:17



A-patel-guy

107 2 8

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

Google

Facebook

4 1) Start by opening up git-bash.exe in Administrator mode. (Right click the file and select "Run as Administrator", or change settings in Properties → Compatibility → Run this program as administrator.)

2) Run `cd ~`. It will take you to `C:/Users/<Your-Username>`.

3) Run `vi .bashrc`. This will open you up into the editor. Hit INSERT and then start entering the following info:

```
alias ll="ls -la" # this changes the default ll on git bash to see hidden files.
cd "C:\directory\to\your\work\path\"
ll # this shows your your directory before you even type anything.
```

edited Sep 25 '16 at 20:07



SiHa

3,595 6 19 34

answered Feb 5 '16 at 17:22



Ruben Arevalo

111 1 2

1 Um, you don't need admin to edit your own files ... – SamB Apr 28 at 0:33

-2 Sometimes the files are actually located at `~/`. These are the steps I took to starting [Zsh](#) as the default terminal on [Visual Studio Code](#)/Windows 10.

- `cd ~/`
- `vim .bashrc`
- Paste the following...

```
if test -t 1; then
exec zsh
fi
```

- Save/close Vim.

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Facebook



14.2k

19

88

114



144

1

2

13

I'm not able to open the terminal anymore. While open the terminal it closed automatically. – [maheshwagmare](#) Jun 21 '18 at 9:24

- 1 This solution doesn't help if an user doesn't have zsh in his/her machine. @maheshwagmare If you still have an issue, remove three lines you added from .bashrc file. – [YoungChoi](#) Mar 22 at 19:17

If you want to have *projects choice list* when you open Git Bash:

-4

- Edit `ppath` in the code header to your Git projects *path*, put this code into **.bashrc** file, and copy it into your `$HOME` directory (in Windows Vista / Windows 7 it is often `C:\Users\%YOU%`)

```
#!/bin/bash
ppath="/d/-projects/-github"
cd $ppath
unset PROJECTS
PROJECTS+=(".")
i=0

echo
echo -e "projects:\n-----"

for f in *
do
    if [ -d "$f" ]
    then
        PROJECTS+=("$f")
        echo -e $((++i)) "- \e[1m$f\e[0m"
    fi
done

if [ ${#PROJECTS[@]} -gt 1 ]
then
    echo -ne "\nchoose project: "
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



```
    echo "there is no projects"
fi
unset PROJECTS
```

- You may want set this file as executable inside *Git Bash*, **chmod +x .bashrc** (but it's probably redundant, since this file is stored on an [NTFS](#) filesystem)

edited Aug 8 '18 at 16:22



Peter Mortensen

14.2k 19 88 114

answered Sep 26 '14 at 17:29



s1w_

130 6

This config is completely tangential to the question. – [Matthew Read](#) Sep 25 '16 at 19:03

Join **Stack Overflow** to learn, share knowledge, and build your career.

[Email Sign Up](#)

OR SIGN IN WITH



Google

[Facebook](#)