1. In Git, we need to configure the default text editor that will be used when Git needs you to type in a message. If not configured, Git uses your system’s default editor.

If we want to configure the default edit in Git as Notepad++, we can run the following command line in Git bash as below

1. On a 32-bit Windows system, or you have a 64-bit editor on a 64-bit system, you should run t his:

$ git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe'

-multiInst -nosession"

1. If you have a 32-bit editor on a 64-bit system, the program will be installed in C:\Program Files(x86), then run this:

$ git config --global core.editor "'C:/Program Files (x86)/Notepad++/notepad++.exe'

-multiInst -nosession"

After running the above command, we can find the argument we gave in the file under the directory “C:\Users\<UserName>\.gitconfi”. We can make changes within this file and it will take effect.

Note that within the above command, make sure the spaces are correct. For example, there is a space between .exe’ and the -multiInst. Missing this space will make this command invalid, and we won’t be able to use the Notepad++ as the default editor.

1. The following command can generate a .gitignore file in the repository
2. touch .gitignore
3. create a txt file and give the name as “.gitignore.”
4. create a new text file , open it and save as “.gitignore” by choosing the file type as ‘all’ under windows file explorer gui.
5. Solve the problem of .gitignore not working problem

This happens to me too often and I always end up googling the answer so this post is probably more of a location I know I can come to find the answer, although by writing it down hopefully it may sink in that I should stop getting too excited on a new project.

New project scenario

You’re all very excited about your new project and you think its about time you committed this to source control. Obviously you’re using [Git](http://git-scm.com/) so you initialise a new repository and commit your files. You then setup a remote repository at [Github](http://github.com/) and it asks you whether you want it create a .gitignore file – you do. So now you have a repository remotely and locally. Easiest thing to do is pull from the remote, setup your remote and push to it. The other scenario might be you’ve committed locally and then realise you need to add a .gitignore file which you do and then commit.

In both cases you will now see the files in your [standardised .gitignore file](https://github.com/github/gitignore) are not being ignored. After a few head scratches you realise its because the .gitignore file should be added to your repo first before any commits.

The solution!

Long story short you have to remove all tracked files and add them back in using the below commands

git rm -r --cached .

git add .

git commit -m ".gitignore is now working"

The first line unstages and removes the paths to your files from the git index recursively.

The second line adds all your files back in but because the .gitignore is present it will not add files that should be ignored!

The final line commits all your files back to the index.

As much as I’d like to take credit for this knowledge I’m going to have to point you in the direction of the [stackoverflow post](http://stackoverflow.com/questions/1139762/gitignore-file-not-ignoring)that has helped me in the past.