

Git Ignore

In Git, the term "ignore" is used to specify intentionally untracked files that Git should ignore. It doesn't affect the Files that already tracked by Git.

Sometimes you don't want to send the files to Git service like GitHub. We can specify files in Git to ignore.

The file system of Git is classified into three categories

Tracked

Tracked files are such files that are previously staged or committed.

Untracked:

Untracked files are such files that are not previously staged or committed.

Ignored:

Ignored files are such files that are explicitly ignored by git. We have to tell git to ignore such files.

Generally, the Ignored files are artifacts and machine-generated files. These files can be derived from your repository source or should otherwise not be committed. Some commonly ignored files are as follows:

- dependency caches
- compiled code
- build output directories, like /bin, /out, or /target
- runtime file generated, like .log, .lock, or .tmp
- Hidden system files, like Thumbs.db or.DS_Store
- Personal IDE config files, such as .idea/workspace.xml

Git Ignore Files

Git ignore files is a file that can be any file or a folder that contains all the files that we want to ignore. The developers ignore files that are not necessary to execute the project. Git itself creates many system-generated ignored files. Usually, these files are hidden files. There are several ways to specify the ignore files. The ignored files can

be tracked on a **.gitignore** file that is placed on the root folder of the repository. No explicit command is used to ignore the file.

There is no explicit git ignore command; instead, the **.gitignore** file must be edited and committed by hand when you have new files that you wish to ignore. The **.gitignore** files hold patterns that are matched against file names in your repository to determine whether or not they should be ignored.

How to Ignore Files Manually

There is no command in Git to ignore files; alternatively, there are several ways to specify the ignore files in git. One of the most common ways is the **.gitignore** file. Let's understand it with an example.

The **.gitignore** file:

Rules for ignoring file is defined in the **.gitignore** file. The **.gitignore** file is a file that contains all the formats and files of the ignored file. We can create multiple ignore files in a different directory. Let's understand how it works with an example:

Step1: Create a file named **.gitignore** if you do not have it already in your directory. To create a file, use the command **touch** or **cat**. It will use as follows:

1. \$ **touch .gitignore**

Or

1. \$ **cat .gitignore**

The above command will create a **.gitignore** file on your directory. Remember, you are working on your desired directory. Consider the below command:

The above command will create a **.gitignore** file on your directory. Remember, you are working on your desired directory. Consider the below command:

```
HiMaNshU@HiMaNshU-PC MINGW64 ~/Desktop/GitExample2 (test2)
$ touch .gitignore
```

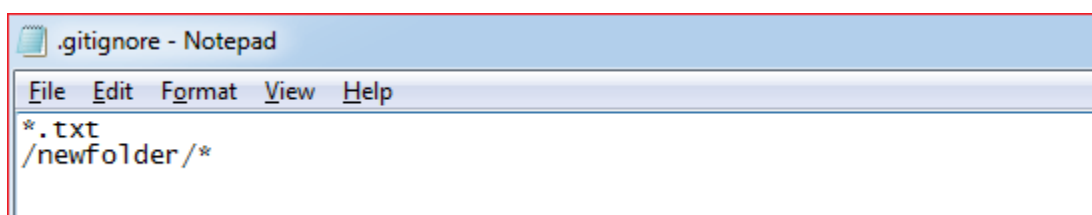
The above command will create a file named .gitignore. We can track it on the repository. Consider the below image:

Name	Date modified	Type	Size
.git	11/5/2019 11:33 AM	File folder	
New folder	11/5/2019 11:31 AM	File folder	
.gitignore	11/5/2019 11:32 AM	Text Document	1 KB
design.css	10/15/2019 2:07 PM	Cascading Style S...	1 KB
index.jsp	9/19/2019 6:10 PM	JSP File	2 KB
master.jsp	9/19/2019 6:10 PM	JSP File	1 KB
merge the branch	9/20/2019 6:05 PM	File	1 KB
newfile.txt	10/15/2019 2:20 PM	Text Document	1 KB
newfile1.txt	10/15/2019 2:27 PM	Text Document	1 KB
newfile2.txt	11/3/2019 5:22 PM	Text Document	1 KB
README.md	9/19/2019 6:10 PM	MD File	1 KB

As you can see from the above image, a .gitignore file has been created for my repository.

As you can see from the above image, a .gitignore file has been created for my repository.

Step2: Now, add the files and directories to the **.gitignore** file that you want to ignore. To add the files and directory to the .git ignore the file, open the file and type the file name, directory name, and pattern to ignore files and directories. Consider the below image:



In the above file, I have given one format and a directory to ignore. The above format ***.txt** will ignore all the text files from the repository, and **/newfolder/*** will ignore the newfolder and its sub-content. We can also give only the name of any file to ignore.

Step3: Now, to share it on Git, we have to commit it. The .gitignore file is still now in staging area; we can track it by git status command. Consider the below output:

```
HiMaNshU@HiMaNshU-PC MINGW64 ~/Desktop/GitExample2 (test2)
$ git status
On branch test2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   newfile2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
```

Now to stage it, we have to commit it. To commit it, run the below command:

1. \$ git add .gitignore
2. \$ git commit -m "ignored directory created."

The above command will share the file .gitignore on Git. Consider the below output.

```
HiMaNshU@HiMaNshU-PC MINGW64 ~/Desktop/GitExample2 (test2)
$ git commit -m " ignored directory created"
[test2 9d9470e] ignored directory created
2 files changed, 2 insertions(+)
create mode 100644 .gitignore
```

Now, we have ignored a pattern file and a directory in Git.