

Stage Files: `git add`

Git does not automatically save all changes as text editors do. Instead, you need to tell Git which files you are adding to the checkpoint that you will save. This process is called *staging*.

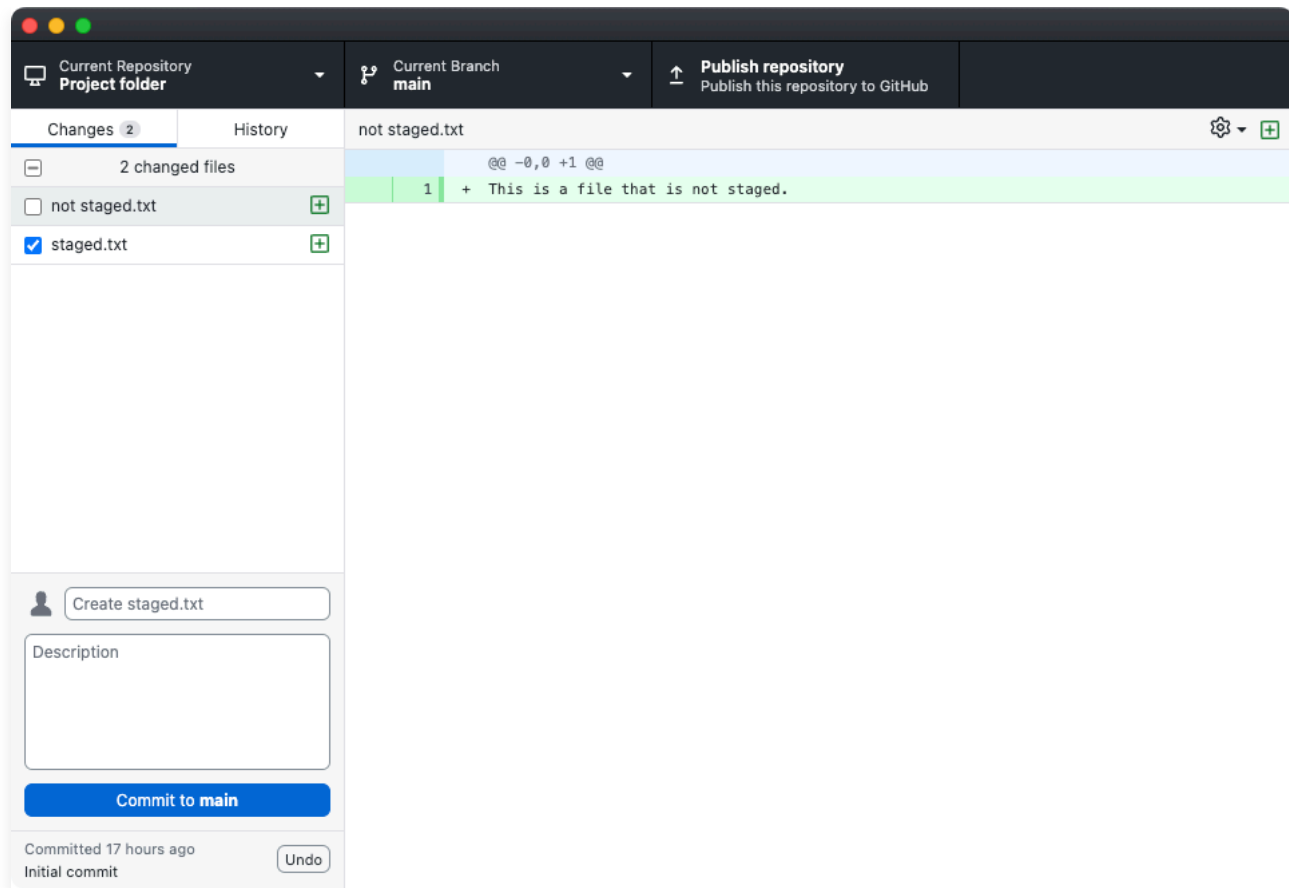
To stage a file in GitHub Desktop:

1. Make changes to any files in the **Project folder**. When you open GitHub Desktop, you will see the files with changes.

If the files are unchanged, they won't appear. This shows that

Git only saves *changes* to files.

To *stage* a file, simply use the checkmark. By default, GitHub Desktop stages all the files that have changed. So in the case of this GUI, this step is more like de-stage the files whose changes you don't want to save.



The file shown is the one **not** being staged. Notice the lack of a checkmark on the left.

In the command line:

- `git add` stages files.

Follow Along | Stage

1. In the **Project folder**, create a blank text file called **staged.txt**. And write in it "This is an example of a text file to stage."
2. Create another blank text file and call it **not staged.txt**. And write in it "This is a file that is not staged."

These two text documents will explain a simple concept that, when expanded to massive projects, may change how you look at them.

3. Now open GitHub Desktop and open the repository **Project folder**.
4. Only check the **staged.txt** file.

The **Staged** panel displays the files that are getting committed changes. The next step is creating a checkpoint.

