Git Revert

In Git, the term revert is used to revert some changes. The git revert command is used to apply revert operation. It is an undo type command. However, it is not a traditional undo alternative. It does not delete any data in this process; instead, it will create a new change with the opposite effect and thereby undo the specified commit. Generally, git revert is a commit.

It can be useful for tracking bugs in the project. If you want to remove something from history then git revert is a wrong choice.

Moreover, we can say that git revert records some new changes that are just opposite to previously made commits. To undo the changes, run the below command:

**Syntax:**

1. $ git revert

Git Revert Options:

Git revert allows some additional operations like editing, no editing, cleanup, and more. Let's understand these options briefly:

**< commit>:** The commit option is used to revert a commit. To revert a commit, we need the commit reference id. The git log command can access it.

1. $ git revert **<commit-ish>**

**<--edit>:** It is used to edit the commit message before reverting the commit. It is a default option in git revert command.

1. $ git revert -e **<commit-ish>**

**-m parent-number /--mainline parent-number:** it is used to revert the merging. Generally, we cannot revert a merge because we do not know which side of the merge should be considered as the mainline. We can specify the parent number and allows revert to reverse the change relative to the specified parent.

**-n/--no edit:** This option will not open a text editor. It will directly revert the last commit.

1. $ git revert -n **<commit-ish>**

**--cleanup=<mode>:** The cleanup option determines how to strip spaces and comments from the message.

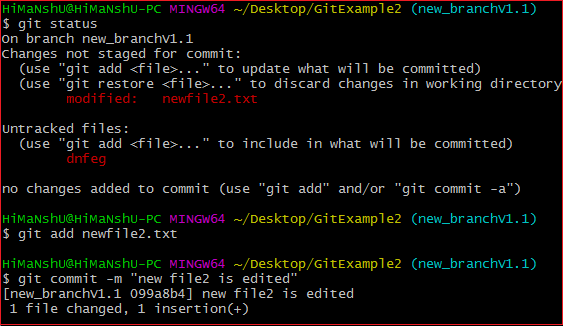
**-n/--no-commit:** Generally, the revert command commits by default. The no-commit option will not automatically commit. In addition, if this option is used, your index does not have to match the HEAD commit.

The no-commit option is beneficial for reverting more than one commits effect to your index in a row.

Let's understand how to revert the previous commits.

Git Revert to Previous Commit

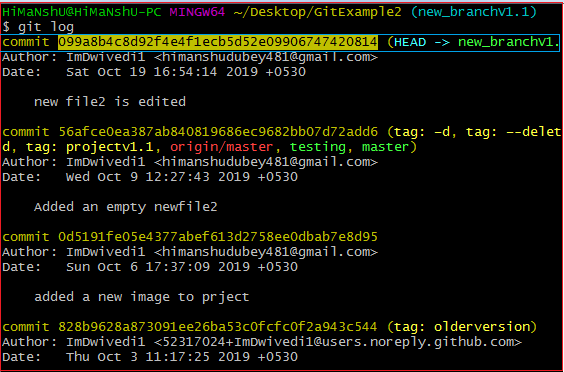
Suppose you have made a change to a file say **newfile2.txt** of your project. And later, you remind that you have made a wrong commit in the wrong file or wrong branch. Now, you want to undo the changes you can do so. Git allows you to correct your mistakes. Consider the below image:



As you can see from the above output that I have made changes in newfile2.txt. We can undo it by git revert command. To undo the changes, we will need the commit-ish. To check the commit-ish, run the below command:

1. $ git log

Consider the below output:



In the above output, I have copied the most recent commit-ish to revert. Now, I will perform the revert operation on this commit. It will operate as:

1. $ git revert 099a8b4c8d92f4e4f1ecb5d52e09906747420814

The above command will revert my last commit. Consider the below output:



As you can see from the above output, the changes made on the repository have been reverted.

Git Revert Merge

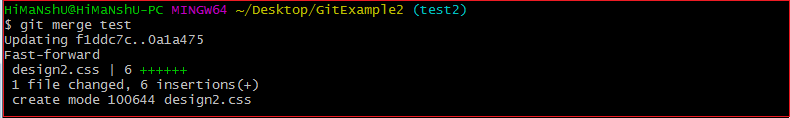
In Git, merging is also a commit that has at least two parents. It connects branches and code to create a complete project.

A merge in Git is a commit that has at least two parents. It brings together multiple lines of development. In a work-flow where features are developed in branches and then merged into a mainline, the merge commits would typically have two parents.

How to Revert a Merge

Usually, reverting a merge considered a complicated process. It can be complex if not done correctly. We are going to undo a merge operation with the help of git revert command. Although some other commands like git reset can do it. Let's understand how to revert a merge. Consider the below example.

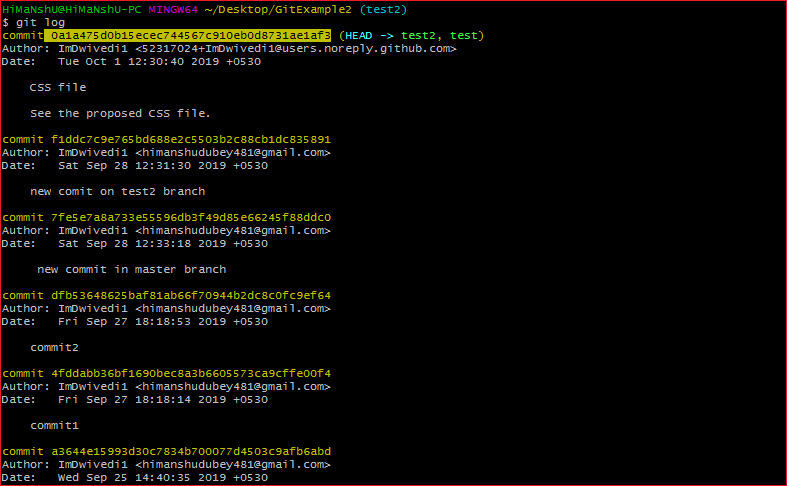
I have made some changes to my file **design2.css** on the test and merge it with **test2**. Consider the below output:



To revert a merge, we have to get its reference number. To check commit history, run the below command:

1. $ git log

The above command will display the commit history. Consider the below output:



From the above output, copy your merging commit that you to want to revert and run the below command:

1. $ git revert **<commit** reference**>** -m 1

The above command will revert the merging operation. Here, -m 1 is used for the first parent as the mainline. Merge commit has multiple parents. The revert needs additional information to decide which parent of the merge shall be considered as the mainline. In such cases, the parameter -m is used. Consider the below output:



From the above output, we can see that the previous merge has been reverted.