

GIT Branches Lab

Author: Om Luitel (10th Sept, 2023)

Using Git Reset: Soft and Hard Options for Branch Rewinding

To go back to a previous commit in Git, you can use the git reset command along with different options to specify how you want to reset the branch pointer. The options you mentioned, "soft" and "hard," control how Git handles the changes in your working directory and staging area.

Here's how you can use git reset with "soft" and "hard" options:

1. Soft Reset (Preserves Changes):

A soft reset moves the branch pointer to a previous commit while preserving the changes in your working directory and staging area. This means the changes from the specified commit are moved to the staging area for you to review and commit again.

```
git reset HEAD~<number of commits you'd like to go back> --soft
```

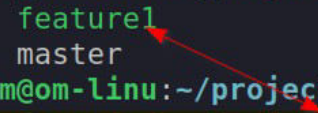
• Hard Reset (Discards Changes):

A hard reset moves the branch pointer to a previous commit and discards all changes in your working directory and staging area. Be cautious when using this option, as it permanently deletes uncommitted changes.

```
git reset HEAD~<number of commits you'd like to go back> --hard
```

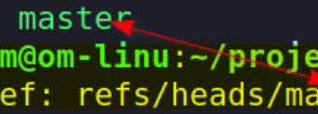
We are currently inside the 'feature1' branch, and let's locate where HEAD is residing. Since we are in the 'feature1' branch, HEAD is pointing to the 'feature1' branch.

```
om@om-linu:~/projects/git-project/branch02$ git switch feature1
Switched to branch 'feature1'
om@om-linu:~/projects/git-project/branch02$ git branch
* feature1
  master
om@om-linu:~/projects/git-project/branch02$ cat .git/HEAD
ref: refs/heads/feature1
om@om-linu:~/projects/git-project/branch02$
```

A red arrow points from the asterisk next to 'feature1' in the 'git branch' output to the 'ref: refs/heads/feature1' line in the 'cat .git/HEAD' output.

Let's analyze what will happen if we switch branches to 'master.'

```
om@om-linu:~/projects/git-project/branch02$ git switch master
Switched to branch 'master'
om@om-linu:~/projects/git-project/branch02$ git branch
  feature1
* master
om@om-linu:~/projects/git-project/branch02$ cat .git/HEAD
ref: refs/heads/master
om@om-linu:~/projects/git-project/branch02$
```

A red arrow points from the asterisk next to 'master' in the 'git branch' output to the 'ref: refs/heads/master' line in the 'cat .git/HEAD' output.

Now, We can see that HEAD is pointing to the 'master' branch.

Now lets see the git log on feeature1 branch:

```
om@om-linu:~/projects/git-project/branch02$ git log
commit a30518e958787092e1d67222d8632ba99a8c3396 (HEAD -> feature1)
Author: omluitel <omprakashluitelessm@gmail.com>
Date:   Sun Sep 10 14:37:40 2023 +1000

    This commit is for feature1 branch featurefile01.txt

commit 772303af114b273b2b2c6ad7baec0b34470586ae
Author: omluitel <omprakashluitelessm@gmail.com>
Date:   Sun Sep 10 13:59:57 2023 +1000

    Committed two.txt to the master branch

commit a1f10155c1fd52841d3971302c9a12a0bf37cf2b
Author: omluitel <omprakashluitelessm@gmail.com>
Date:   Sun Sep 10 13:47:53 2023 +1000

    Committed one.txt to the master branch
om@om-linu:~/projects/git-project/branch02$
```

We can see 3 commits logs on feature1 branch.

Now, we're utilizing the 'git switch HEAD~<number of commits you'd like to go back>' command to navigate to a previous commit in our project's history.

```
om@om-linu:~/projects/git-project/branch02$ git log
commit a30518e958787092e1d67222d8632ba99a8c3396 (HEAD -> feature1)
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 14:37:40 2023 +1000

    This commit is for feature1 branch featurefile01.txt

commit 772303af114b273b2b2c6ad7baec0b34470586ae
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:59:57 2023 +1000

    Committed two.txt to the master branch

commit a1f10155c1fd52841d3971302c9a12a0bf37cf2b
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:47:53 2023 +1000

    Committed one.txt to the master branch
om@om-linu:~/projects/git-project/branch02$ git switch HEAD~1 --detach
HEAD is now at 772303a Committed two.txt to the master branch
om@om-linu:~/projects/git-project/branch02$ git log
commit 772303af114b273b2b2c6ad7baec0b34470586ae (HEAD)
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:59:57 2023 +1000

    Committed two.txt to the master branch

commit a1f10155c1fd52841d3971302c9a12a0bf37cf2b
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:47:53 2023 +1000

    Committed one.txt to the master branch
om@om-linu:~/projects/git-project/branch02$
```

can see 3 commits

can only see 2 commits

We had 3 commits logs initially but when we ran git switch HEAD~1 --detach , and ran git log we can only see 2 commits.

Now, we wanna work on master branch:

Suppose I accidentally committed the 'feb9c14dacab4077' commit, and I want to reverse the changes. How can I achieve this?

1. Undo the change and keep the file in the working directory.
2. Undo the commit while keeping the file in the working directory and staged.
3. Undo the commit and discard the changes entirely.

Option:01 Undo the change and keep the file in the working directory | Not in staging area.

```
om@om-linu:~/projects/git-project/branch02$ git log
commit feb9c14dacab40770d34665cfb0986c770f0262f (HEAD -> master)
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 14:29:40 2023 +1000
```

Committed three.txt to the master branch

```
commit 772303af114b273b2b2c6ad7baec0b34470586ae
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:59:57 2023 +1000
```

Committed two.txt to the master branch

```
commit alf10155c1fd52841d3971302c9a12a0bf37cf2b
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:47:53 2023 +1000
```

Committed one.txt to the master branch

```
om@om-linu:~/projects/git-project/branch02$ git res
reset      restore
```

```
om@om-linu:~/projects/git-project/branch02$ git reset HEAD~1
```

```
om@om-linu:~/projects/git-project/branch02$ ls
```

```
one.txt  three.txt  two.txt
```

```
om@om-linu:~/projects/git-project/branch02$ git log
commit 772303af114b273b2b2c6ad7baec0b34470586ae (HEAD -> master)
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:59:57 2023 +1000
```

Committed two.txt to the master branch

```
commit alf10155c1fd52841d3971302c9a12a0bf37cf2b
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:47:53 2023 +1000
```

Committed one.txt to the master branch

```
om@om-linu:~/projects/git-project/branch02$
```

```
om@om-linu:~/projects/git-project/branch02$ git status
```

On branch master

Untracked files:

(use "git add <file>..." to include in what will be committed)

three.txt

nothing added to commit but untracked files present (use "git add" to track)

```
om@om-linu:~/projects/git-project/branch02$
```


Option02: Undo the commit while keeping the file in the working directory and staged | soft reset.

```
om@om-linu:~/projects/git-project/branch02$ git log
commit a3ca09b97ee6f7d90e145c2047f65592a36db94a (HEAD -> master)
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 16:17:21 2023 +1000

    Third commit

commit 772303af114b273b2b2c6ad7baec0b34470586ae
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:59:57 2023 +1000

    Committed two.txt to the master branch

commit a1f10155c1fd52841d3971302c9a12a0bf37cf2b
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:47:53 2023 +1000

    Committed one.txt to the master branch
om@om-linu:~/projects/git-project/branch02$ git reset HEAD~1 --soft
om@om-linu:~/projects/git-project/branch02$ git log
commit 772303af114b273b2b2c6ad7baec0b34470586ae (HEAD -> master)
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:59:57 2023 +1000

    Committed two.txt to the master branch

commit a1f10155c1fd52841d3971302c9a12a0bf37cf2b
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:47:53 2023 +1000

    Committed one.txt to the master branch
```

```
om@om-linu:~/projects/git-project/branch02$ tree
.
├── one.txt
├── three.txt
└── two.txt

0 directories, 3 files
om@om-linu:~/projects/git-project/branch02$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   three.txt

om@om-linu:~/projects/git-project/branch02$
```

Option03: Undo the commit and discard the changes entirely, hard reset.

```
om@om-linu:~/projects/git-project/branch02$ git log
commit d3da0a92fdbea34ab245e5f5254df430f1cd47f0 (HEAD -> master)
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 16:26:56 2023 +1000

    three.txt commit

commit 772303af114b273b2b2c6ad7baec0b34470586ae
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:59:57 2023 +1000

    Committed two.txt to the master branch

commit a1f10155c1fd52841d3971302c9a12a0bf37cf2b
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:47:53 2023 +1000

    Committed one.txt to the master branch
om@om-linu:~/projects/git-project/branch02$ git reset HEAD~1 --hard
HEAD is now at 772303a Committed two.txt to the master branch
om@om-linu:~/projects/git-project/branch02$ git log
commit 772303af114b273b2b2c6ad7baec0b34470586ae (HEAD -> master)
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:59:57 2023 +1000

    Committed two.txt to the master branch

commit a1f10155c1fd52841d3971302c9a12a0bf37cf2b
Author: omluitel <omprakashluitelessm@gmail.com>
Date: Sun Sep 10 13:47:53 2023 +1000

    Committed one.txt to the master branch
om@om-linu:~/projects/git-project/branch02$ ls
one.txt  two.txt
om@om-linu:~/projects/git-project/branch02$
```