

# GIT: From Beginner To Fearless

GIT Real Life Activity 02:

I accidentally committed to master on my local and pushed to remote. What do I do?

Brian Gorman, Author/Instructor/Trainer

©2019 - MajorGuidanceSolutions

# Introduction

---

You have been working on your local branch which is in sync with remote, and you have pushed a couple of commits. You are therefore currently one or more commits ahead of master [or some other integration branch your team uses that you should not push to]. Obviously, you don't want to affect the master branch or your team's main working-tree integration branch.

You were about to perform a merge to your local branch, so you switch to the working tree master/develop branch, fetch and pull to make sure you are up-to-date locally.

You step away for a minute, a meeting, or some other distraction, and you come back to continue where you left off working.

You forget you were in the middle of working on merging master to your local branch and you make some changes and commit and then push to remote without thinking – and without checking what branch you are on. You have now moved the team master at remote to a commit that is out of sync, both with the branch you actually wanted to push to, as well as on the remote.

Panic sets in, and you immediately notify the team that something is broken. If you are lucky, you can also stop the automated build in time to prevent the commit from propagating further in your pipeline.

How do you fix the broken master branch, both locally, and at remote?

In this activity, we will run through this scenario so that we both have the confidence and the know-how to fix the repository.

The steps to fix this are:

- 1) Stop the pipeline build if possible to prevent auto-deployment
- 2) Backup master – for our own peace of mind
- 3) Checkout a new branch from master
- 4) Restore local master to the correct commit via a reset operation
- 5) Force push the corrected branch to master at remote
- 6) Pick the changes you wanted to save into your feature branch
- 7) Merge updated master into your branch to resolve any conflicts
- 8) Merge your feature into master
- 9) Clean it all up

Let's get started.

## Step 1: Setup – working on your feature branch:

- a) Check out a new branch “rebase-pick-scenario-two”.

This branch is going to be the branch you are performing your feature updates on.

```
[git checkout -b rebase-pick-scenario-two]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingDirectory
(master)
$ git checkout -b rebase-pick-scenario-two
Switched to a new branch 'rebase-pick-scenario-two'
```

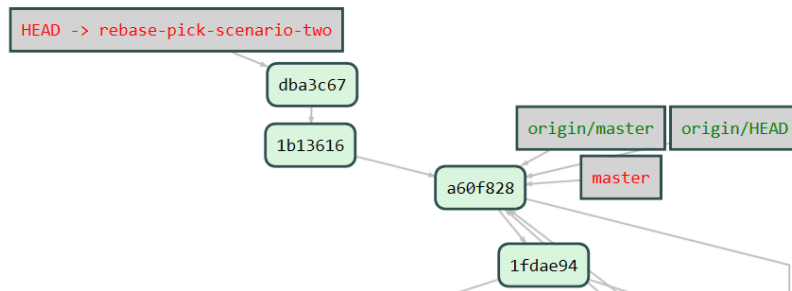
Simulate work by moving the branch ahead a couple of commits. Simply put in a couple changes. Here I've changed the about page in the default web repo:

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingDirectory
(rebase-pick-scenario-two)
$ git add .

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingDirectory
(rebase-pick-scenario-two)
$ git commit -m "Created info on the about page commit 1"
[rebase-pick-scenario-two 1b13616] Created info on the about page commit 1
1 file changed, 3 insertions(+)

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingDirectory
(rebase-pick-scenario-two)
$ git add .

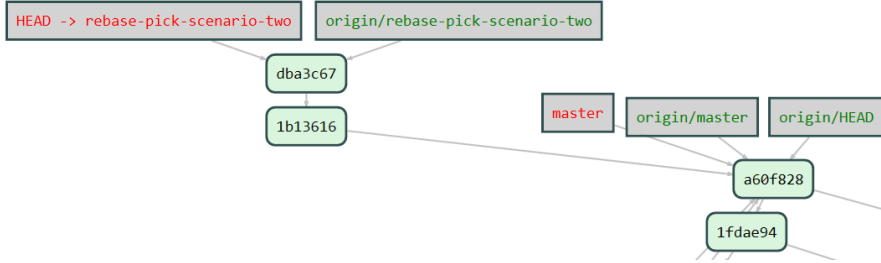
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingDirectory
(rebase-pick-scenario-two)
$ git commit -m "Created info on the about page commit 2"
[rebase-pick-scenario-two dba3c67] Created info on the about page commit 2
1 file changed, 3 insertions(+)
```



## Notes

b) Create a pull request at GitHub for the two commits.

```
git push origin -u rebase-scenario-two
```



## Rebase pick scenario two #1

[Open](#) blgorman wants to merge 2 commits into `master` from `rebase-pick-scenario-two`

Conversation 0 Commits 2 Checks 0 Files changed 1

blgorman commented now

rebase and pick scenario 2

blgorman added some commits 4 minutes ago

- Created info on the about page commit 1 1b13616
- Created info on the about page commit 2 dba3c67

Add more commits by pushing to the `rebase-pick-scenario-two` branch on blgorman/DefaultWeb\_WorkingDirectory.

This branch has no conflicts with the base branch  
Merging can be performed automatically.

[Merge pull request](#) You can also open this in [GitHub Desktop](#) or view [command line instructions](#).

DO NOT merge this at GitHub yet.



## Step 2: Simulate another developer affecting master

a) In the time you've been working, another developer's changes were merged


It is up to you if you want to create a conflict. Perhaps the first time through this scenario avoid having files that are in direct conflict with your changes.

Create a change with PR at GitHub from the “other” developer:



## Another dev moves it forward #2

 **Open** blgorman wants to merge 1 commit into `master` from `rebase-activity-two-otherdev` 


Conversation 0 Commits 1 Checks 0 Files changed 1

 blgorman commented yesterday Owner + 👤 ...

This change simulates another dev moving the repo forward with valid work

  Another dev moves it forward ... Verified 62962b8

Add more commits by pushing to the `rebase-activity-two-otherdev` branch on blgorman/DefaultWeb\_WorkingDirectory.



 This branch has no conflicts with the base branch  
Merging can be performed automatically.

**Merge pull request** You can also open this in GitHub Desktop or view command line instructions.


### b) Merge the pull request

Merge the PR and delete the branch



## Another dev moves it forward #2


 **Merged** blgorman merged 1 commit into `master` from `rebase-activity-two-otherdev` now 


Conversation 0 Commits 1 Checks 0 Files changed 1


 blgorman commented yesterday Owner + 👤 ...

This change simulates another dev moving the repo forward with valid work

  Another dev moves it forward ... Verified 62962b8

 blgorman merged commit 3fb1d23 into `master` now Revert

 Pull request successfully merged and closed  
You're all set—the `rebase-activity-two-` branch can be safely deleted. Delete branch

 Write Preview AA B i “ < > @

## Step 3: Sync your local branch with master

### a) Checkout your local master branch

```
[git checkout master]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb  
ario-two)  
$ git checkout master  
Switched to branch 'master'  
Your branch is up to date with 'origin/master'.
```

b) Fetch and Pull to get up to date with remote

```
[git fetch --prune]
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defaultw
$ git fetch --prune
From https://github.com/blgorman/DefaultWeb_WorkingDirectory
- [deleted]          (none)      -> origin/patch-1
- [deleted]          (none)      -> origin/patch-2
- [deleted]          (none)      -> origin/patch-3
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
a60f828..3fb1d23  master       -> origin/master
```

```
[git pull]
```

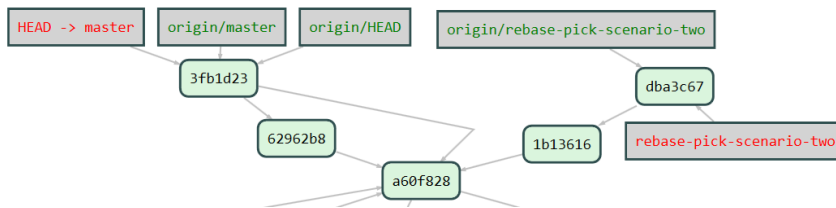
```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/de
$ git pull
Updating a60f828..3fb1d23
Fast-forward
 details.html | 1 +
 1 file changed, 1 insertion(+)
```

c) Make sure you are even with master locally

```
[git log --oneline]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defaul
$ git log --oneline
3fb1d23 (HEAD -> master, origin/master, origin/HEAD) Merge
tivity-two-otherdev
62962b8 Another dev moves it forward
```

Note that your other branch is still one or more commits ahead as well:



## Step 3: Push a commit to local master, then sync

a) On local master, make another change and commit it

Again, here it is up to you if you want to create a conflict or not. Assuming you would notice changes were missing if you were on the same file as before, I'm selecting a third file – Contact Us- and changing it the copyright date:

```

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defaultweb_workingDirectory (master)
$ git status -s
M ContactUs.html

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defaultweb_workingDirectory (master)
$ git commit -am "Updated copyright on the ContactUs.html page"
[master f37f194] Updated copyright on the ContactUs.html page
1 file changed, 1 insertion(+), 1 deletion(-)

```

Note: My changes are on master, so now I'm one commit ahead of remote master, but I forget to check that... and then I

#### b) Push to remote without thinking

Warning: Don't do this if you are working on an important repository unless you are confident you know what you are doing to "undo the mess it created."

[git push]

Or

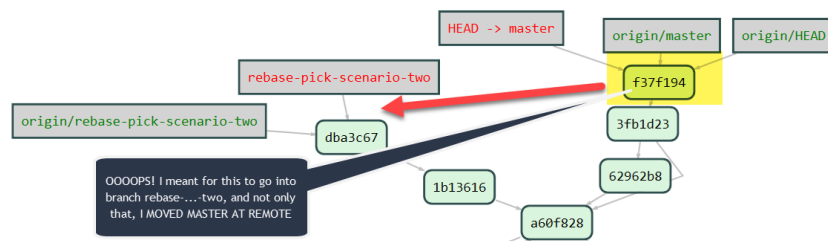
[git push origin master]

```

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defaultweb_workingDirectory (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 330 bytes | 82.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/blgorman/DefaultWeb_workingDirectory.git
3fb1d23..f37f194 master -> master

```

Watch the magic happen



#### c) Immediately notify your team and stop the build

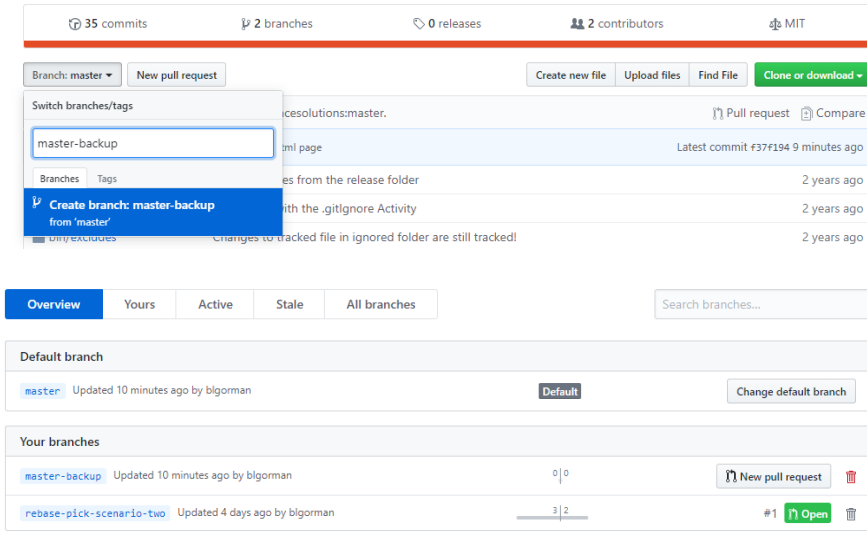
At this point, you've created a large problem in that you've pushed a commit to a public branch that your team wants to keep sacred. Also, you've likely triggered your automated build pipeline, which could create other problems. If possible, immediately stop the build. If you don't catch this in time, hopefully you can re-deploy a previous build as fast as possible.

Also, make sure your team is aware you are going to be rewriting master's history, and to please do not pull master locally or create any branches off of master locally or at GitHub until you have completed the repairs.

## Step 4: Repair Master

- a) Remember that everything you do at GIT can be undone, and branches are inexpensive.

First, for your own peace-of-mind, go to GitHub and make a backup of the master branch. You will NOT need this, but it will make you feel safer in what you are about to do:

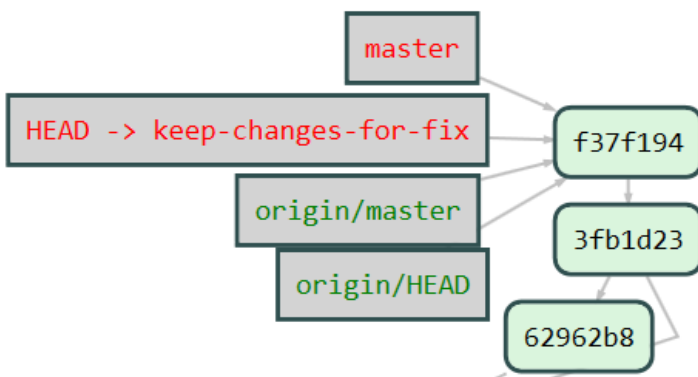


- b) On your local machine, create a new branch “keep-commit-for-fix” from master

Locally, just checkout a new branch from master that you will use later to restore your missing changes. In this scenario, I’ve only committed once, but there may be times when you could have a chain of commits. Either way, as long as you keep them around locally, you’ll be good to go.

```
[git checkout -b keep-changes-for-fix]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/
$ git checkout -b keep-changes-for-fix
Switched to a new branch 'keep-changes-for-fix'
```





- c) Determine the id of the last good commit, and make note of any commits that you will want to get back later.

I've only got one commit [f37f194] that is bad to get back later. You might have more than one. Determine your last good commit to keep on master. In this case, for me, the commit I need is [3fb1d23]. Depending on your workflow, the last good commit likely has a message such as "Merge pull request #n.... from ...."

```
[git checkout master]
[git log --oneline]
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
g
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingDirec
$ git log --oneline
f37f194 (HEAD -> master, origin/master, origin/HEAD, keep-changes-for-fix) U
e ContactUs.html page
3fb1d23 Merge pull request #2 from blgorman/rebase-activity-two-otherdev
62962b8 Another dev moves it forward
```

Make note of commits:

[last good commit]: [3fb1d23]

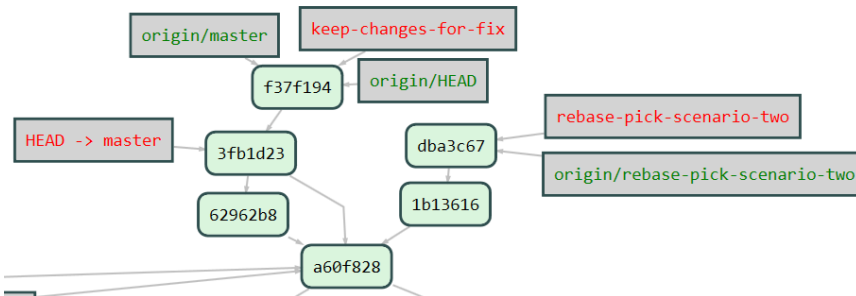
[commits to restore later]: [f37f194, ....] //you might have more than one

- d) Hard reset your local master to the last good commit:

Having identified the last good commit to keep, perform a hard reset against master [make sure you are on the master branch locally]

```
[git reset --hard <good-commit-id>]
```

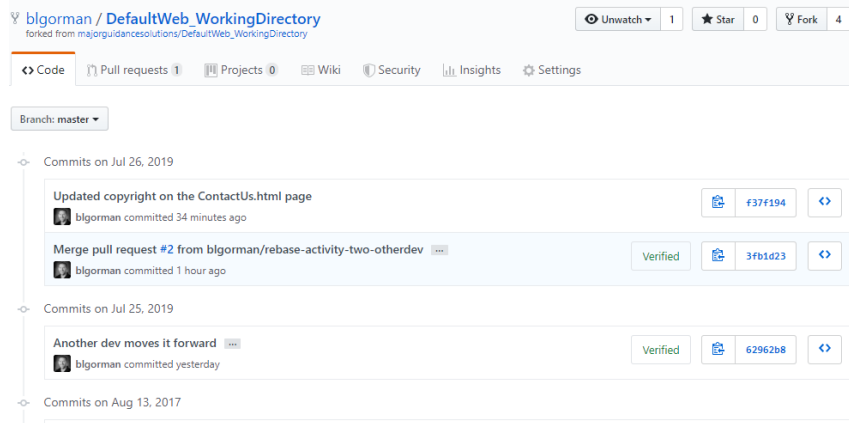
```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingDirectory (master)
$ git reset --hard 3fb1d23
HEAD is now at 3fb1d23 Merge pull request #2 from blgorman/rebase-activity-two-otherdev
```



Now local master is set to where it should be.

### e) Force push the reset to remote

This will be the scariest part of the whole operation, because now remote has to also be reset. If you want, first take a look at the commits at REMOTE:



Here we can see the same history we have locally with the “bad” commit. There are also more commits to follow – your history likely has a bunch of commits.

Again: You should NOT run a force against a public branch unless you know what you are doing and everyone is aware of it. This will rewrite public history and could cause you problems. Since we told our team what is going on, nobody has the bad commit in their master branch history yet and no branches are created off of this bad commit.

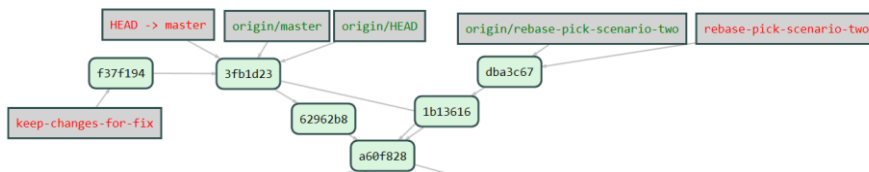
Make sure you are on master and run the command:

```
[git push -f origin master]
```

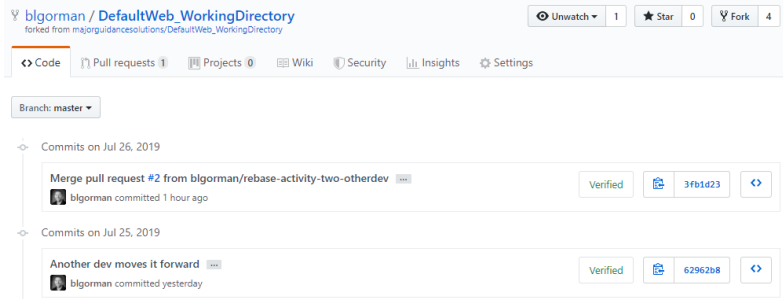
Or

```
[git push -f]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defaultweb_workingDirectory (master)
$ git push -f
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/blgorman/DefaultWeb_WorkingDirectory.git
+ f37f194...3fb1d23 master -> master (forced update)
```



At this point it is safe to let your teammates resume work against master.



As we can see, the bad commit is gone from the master branch in our public repository. This is what we intended.

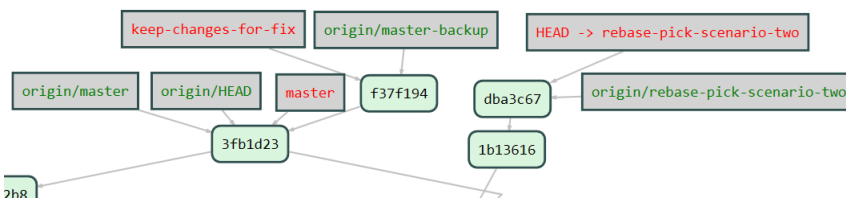
- f) Make sure to re-sync your local master so it doesn't have the bad commit [still from master branch locally]

```
[git fetch --prune]
[git pull]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defaultweb_WorkingDirectory (master)
$ git fetch --prune
From https://github.com/blgorman/DefaultWeb_WorkingDirectory
* [new branch]      master-backup -> origin/master-backup

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defaultweb_WorkingDirectory (master)
$ git pull
Already up to date.

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defaultweb_WorkingDirectory (master)
$ git log --oneline
3fb1d23 (HEAD -> master, origin/master, origin/HEAD) Merge pull request #2 from blgorman/rebase-activity-two-otherdev
62962b8 Another dev moves it forward
```



This step is likely unnecessary, but will give you peace-of-mind. Note: we were “already up to date”

## Step 5: Pick your commit and put it on the branch where you meant to commit

- a) Make sure to checkout your feature branch:

```
[git checkout rebase-pick-scenario-two]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defaultweb_WorkingDirectory (master)
$ git checkout rebase-pick-scenario-two
Switched to branch 'rebase-pick-scenario-two'
Your branch is up to date with 'origin/rebase-pick-scenario-two'.
```

```
[git log --oneline]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/Defaultweb_WorkingDirectory (rebase-pick-scenario-two)
$ git log --oneline
dba3c67 (HEAD -> rebase-pick-scenario-two, origin/rebase-pick-scenario-two) Created info on the about page commit 2
1b13616 Created info on the about page commit 1
```

b) Merge local master like you had intended to do before you messed everything up:

```
[git merge master]
```

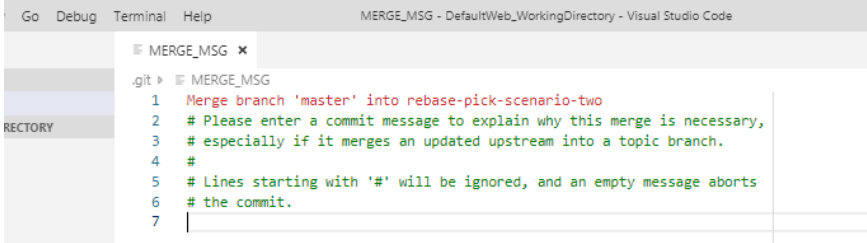
Or

```
[git merge origin/master]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb
ario-two)
```

```
$ git merge master
```

```
hint: Waiting for your editor to close the file...
```



And the merge completes:

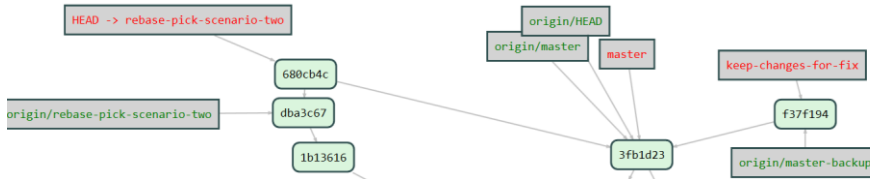
```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb
ario-two)
```

```
$ git merge master
```

```
Merge made by the 'recursive' strategy.
```

```
details.html | 1 +
```

```
1 file changed, 1 insertion(+)
```



Now we are even with master but also two commits ahead. However, we want to get the commit f37f194 into our tree as well before we merge at remote.

c) Pick the commit(s) that contain your remaining changes

Here we have the one commit [f37f194] that we wanted to put in our feature.

```
[git cherry-pick f37f194]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_workingDirectory (rebase-pick-scenario-two)
```

```
$ git cherry-pick f37f194
```

```
[rebase-pick-scenario-two abff8a5] Updated copyright on the ContactUs.html page
```

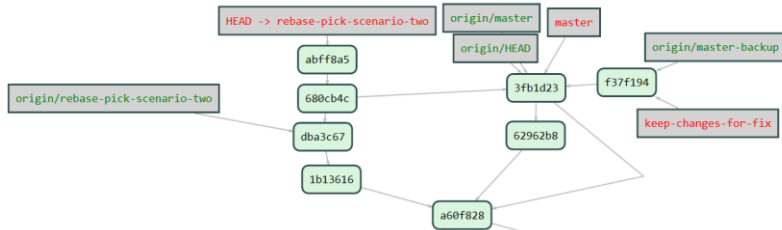
```
Date: Fri Jul 26 14:53:00 2019 -0500
```

```
1 file changed, 1 insertion(+), 1 deletion(-)
```

Because there was no conflict, this just works. If there would have been a conflict, we would have to resolve that before committing the cherry-pick.

As expected, the commit gets a new ID, but we have our changes in the correct branch now:

```
abff8a5 (HEAD -> rebase-pick-scenario-two) Updated copyright on the ContactUs.html page
680cb4c Merge branch 'master' into rebase-pick-scenario-two
3fb1d23 (origin/master, origin/HEAD, master) Merge pull request #2 from blgorman/rebase-activity
wo-otherdev
62962b8 Another dev moves it forward
db3c67 (origin/rebase-pick-scenario-two) Created info on the about page commit 2
```



Performing a diff on the commits abff8a5 and f37f194 should be clean at this point since they would contain the same bits.

#### d) Push your changes to remote to update your open PR

Now that we have everything, and we are synced up, push to remote

[git push]

```

blgor@CORSIAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory (rebase-pick-scenario-two)
$ git push
Enumerating objects: 9, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 638 bytes | 319.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/blgorman/DefaultWeb_WorkingDirectory.git
    dba3c67..abff8a5 rebase-pick-scenario-two -> rebase-pick-scenario-two
  
```

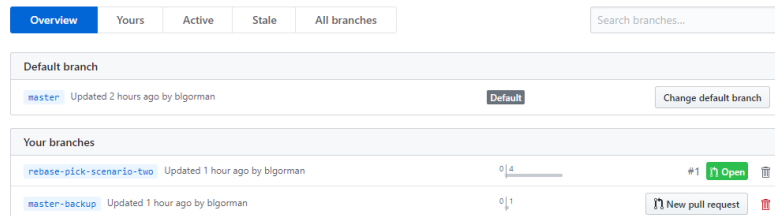
Review the commits you are adding to master:

[git log --oneline --reverse master..]



```





blgor@CORSIAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory (rebase-pick-scenario-two)
$ git log --oneline --reverse master..
1b13616 Created info on the about page commit 1
dba3c67 Created info on the about page commit 2
680cb4c Merge branch 'master' into rebase-pick-scenario-two
abff8a5 (HEAD -> rebase-pick-scenario-two, origin/rebase-pick-scenario-two) Updated copyright on the ContactUs.html page
  
```

#### e) Merge the PR and delete the branch



## Rebase pick scenario two #1

 Open blgorman wants to merge 4 commits into `master` from `rebase-pick-scenario-two` 

 Conversation 0  Commits 4  Checks 0  Files changed 2



blgorman commented 4 days ago





Owner

+ 😊 ...

rebase and pick scenario 2



blgorman added some commits 4 days ago

-  Created info on the about page commit 1 1b13616
-  Created info on the about page commit 2 dba3c67
-  Merge branch 'master' into rebase-pick-scenario-two 680cb4c
-  Updated copyright on the ContactUs.html page abff8a5

Add more commits by pushing to the `rebase-pick-scenario-two` branch on `blgorman/DefaultWeb_WorkingDirectory`.



✓ This branch has no conflicts with the base branch  
Merging can be performed automatically.

Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



blgorman merged commit `bf6968` into `master` 2 minutes ago

Revert



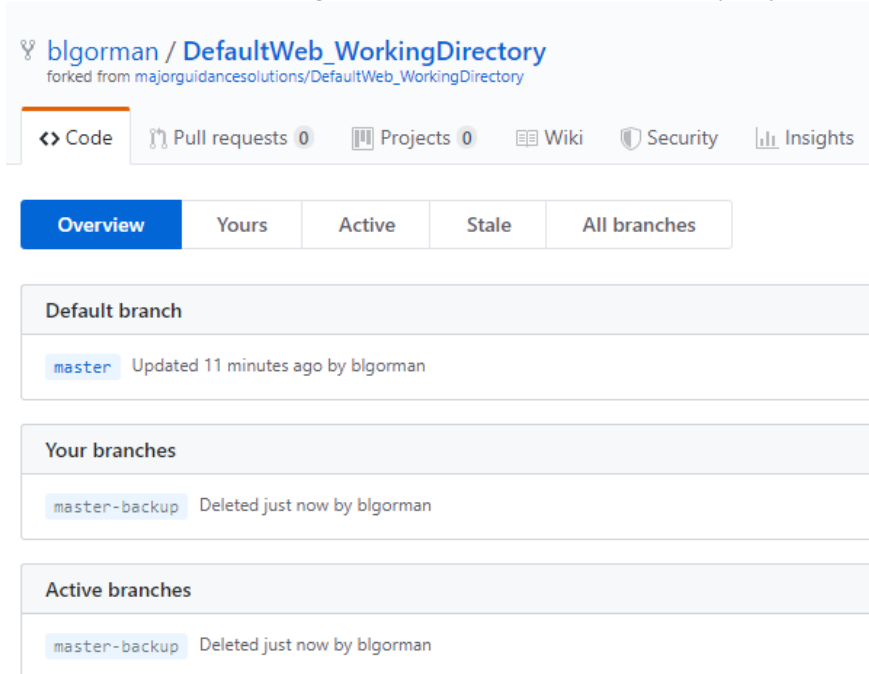
Pull request successfully merged and closed

You're all set—the `rebase-pick-scenario_` branch can be safely deleted.

Delete branch

## Step 6: Cleanup

- a) Remove the backup branch and all fully-merged branches at GitHub  
We know our commits are good and we don't need that backup anymore.



So I have nothing left but master at remote.

- b) Fetch and prune locally, update local master

```
[git checkout master]
[git fetch --prune]
[git pull]
```

```
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

blgor@CORSAlRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory (master)
$ git fetch --prune
gFrom https://github.com/blgorman/Defaultweb_WorkingDirectory
- [deleted]      (none)    -> origin/master-backup
- [deleted]      (none)    -> origin/rebase-pick-scenario-two
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), done.
t 3fb1d23..bff6968  master    -> origin/master

blgor@CORSAlRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory (master)
$ git pull
Updating 3fb1d23..bff6968
Fast-forward
 About.html | 6 +++++
 ContactUs.html | 2 +-
 2 files changed, 7 insertions(+), 1 deletion(-)

blgor@CORSAlRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory (master)
$
```

### c) Check branches, then cleanup

```
[git branch -a]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory (master)
$ git branch -a
keep-changes-for-fix
* master
rebase-pick-scenario-two
remotes/origin/HEAD -> origin/master
remotes/origin/master
remotes/upstream/master

blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory (master)
$
```

```
[git branch -vv | grep ': gone'] | awk '{print $1}' | xargs
git branch -d]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory (master)
$ git branch -vv | grep ': gone'] | awk '{print $1}' | xargs git branch -d
Deleted branch rebase-pick-scenario-two (was abff8a5).

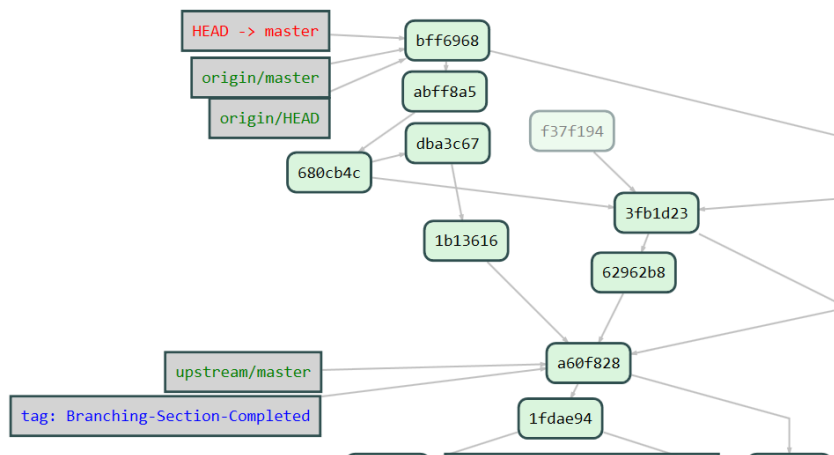
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory (master)
$
```

```
[git branch -D keep-changes-for-fix]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory (master)
$ git branch -D keep-changes-for-fix
Deleted branch keep-changes-for-fix (was f37f194).
```

```
[git log --oneline]
```

```
blgor@CORSAIRONE MINGW64 /c/MGS/DemoArena/defaultweb/DefaultWeb_WorkingDirectory (master)
$ git log --oneline
bff6968 (HEAD -> master, origin/master, origin/HEAD) Merge pull request #1 from blgorman/ck-scenario-two
abff8a5 Updated copyright on the ContactUs.html page
680cb4c Merge branch 'master' into rebase-pick-scenario-two
3fb1d23 Merge pull request #2 from blgorman/rebase-activity-two-otherdev
62962b8 rebase-pick-scenario-two
```



This concludes our git commit to local and remote master activity.



## Closing Thoughts

In this activity we worked through a scenario where we really messed up. First, we made a bunch of commits in our feature, but then another developer got their changes into master before us.

We were going to merge master locally, but walked away in the middle and forgot to move back to our feature branch after pulling the latest master.

We made another mistake by committing locally and then pushing that commit off to remote.

At that point, we had to stop the presses, and cleanup the branches. We backed up master for peace-of-mind at GitHub, and created a keep-commit-fix branch locally to have a reference to our work.

Next we worked locally to hard reset the master branch and force-push it to GitHub. Once that was good, we made sure we were up-to-date, and then moved back to our feature branch.

On our feature branch, we cherry-picked the commit with our work, and then pushed to remote and merged.

Finally, we cleaned everything up so that we are in a good state to move forward with future development.

In the end, we have our repository where it needs to be with all of our changes, and we didn't have any lost work, even though we had merged and pushed to master and then had to fix our mistake.

Notes