



RESOLVING MERGE CONFLICTS

DevOps Certification Training

support@intellipaat.com
+91-7022374614
US: 1-800-216-8930(Toll Free)

RE SOLVING MERGE CONFLICTS

Step 1: Let us first simulate a merge conflict by performing the following merge.

```
$ git merge dev2
```

```
ubuntu@ip-172-31-26-120:~/dev1/devops$ git merge dev2
Auto-merging feature.c
CONFLICT (content): Merge conflict in feature.c
Automatic merge failed; fix conflicts and then commit the result.
```

Step 2: Once we have identified, there is a merge conflict we should go ahead and use the command given below:

```
$ git mergetool
```

```
ubuntu@ip-172-31-26-120:~/dev1/devops$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
tortoisemerge emerge vimdiff
Merging:
feature.c

Normal merge conflict for 'feature.c':
  {local}: modified file
  {remote}: modified file
Hit return to start merge resolution tool (vimdiff): █
```

Step 3: The merge tool looks something like this, the top leftmost column is for Dev1 Branch changes, the center column is for the original code i.e. the master's code before any commits, and the right most column are the dev 2 branch changes. The area below these columns is where we make the changes, this is the place where we have the merged code.



Step 4: Once you have resolved the changes, save the file using “:wq”, vim command for save and exit. Do the same for all the files.



The screenshot shows a vim editor window with the merged code. The status bar at the bottom indicates `feature.c`, `3,14`, and `All`. The command line shows `:wq` entered.

Step 5: After this step, see the status of your local repository you can see the file in conflict has been modified successfully and is merged with master. This modified file can now be committed to the master

```
$ git status
```

```
ubuntu@ip-172-31-26-120:~/dev1/devops$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  modified:   feature.c

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

    feature.c.orig
    feature_BACKUP_2828.c
    feature_BASE_2828.c
    feature_LOCAL_2828.c
    feature_REMOTE_2828.c
```

Step 6: And finally, commit your changes, to the branch and then push it to the remote repository

```
$ git commit -m "merged feature"
```

```
[ubuntu@ip-172-31-26-120:~/dev1/devops$ git commit -m "merged feature"
[master f0ecdbd] merged feature
Committer: Ubuntu <ubuntu@ip-172-31-26-120.us-east-2.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 feature.c
```