**Steps to resolve competing line change merge conflicts between remote branches.**

This conflict occurs when the latest update from master is not synced with the slave causing 2 different changes to be pushed on the same line. We face this issue in common files if multiple users push the code at the same time. These conflicts can also be resolved using the web editor.

$ git status

# On branch slave\_branch

# You have unmerged paths.

# (fix conflicts and run "git commit")

#

# Unmerged paths:

# (use "git add ..." to mark resolution)

#

# both modified: file.txt

#

no changes added to commit (use "git add" and/or "git commit -a")

1. **$ git checkout slave\_branch**
2. **$ git pull origin master**
3. **Open the file with conflicts.**
4. To see the beginning of the merge conflict in your file, search the file for the conflict marker <<<<<<<. When you open the file in your text editor, you'll see the changes from the HEAD or base branch after the line <<<<<<< HEAD. Next, you'll see =======, which divides your changes from the changes in the other branch, followed by >>>>>>> BRANCH-NAME.
5. Decide if you want to keep only your branch's changes, keep only the other branch's changes, or make a brand new change, which may incorporate changes from both branches. Delete the conflict markers <<<<<<<, =======, >>>>>>> and make the changes you want in the final merge. In this example, both changes are incorporated into the final merge:
6. **$ git add file.txt**
7. **$ git commit**
8. **$ git push –u origin slave\_branch**

**Steps to resolve Removed file merge conflicts between remote branches.**

This conflict occurs when a file is deleted in master and the slave branch is not synched with this change. If the slave tries to make changes in the same file conflict will rise when merging the two remote branches. These conflicts are too complex to be resolved using a web editor.

$ git status

# On branch master

# Your branch and 'origin/master' have diverged,

# and have 1 and 2 different commits each, respectively.

# (use "git pull" to merge the remote branch into yours)

# You have unmerged paths.

# (fix conflicts and run "git commit")

#

# Unmerged paths:

# (use "git add/rm ..." as appropriate to mark resolution)

#

# deleted by us: file.txt

#

# no changes added to commit (use "git add" and/or "git commit -a")

1. $ git checkout slave\_branch
2. $ git pull origin master
3. $ git status
4. Option 1: $ git rm file.txt

Option 2: $ git add file.txt

1. $ git push –u origin slave\_branch