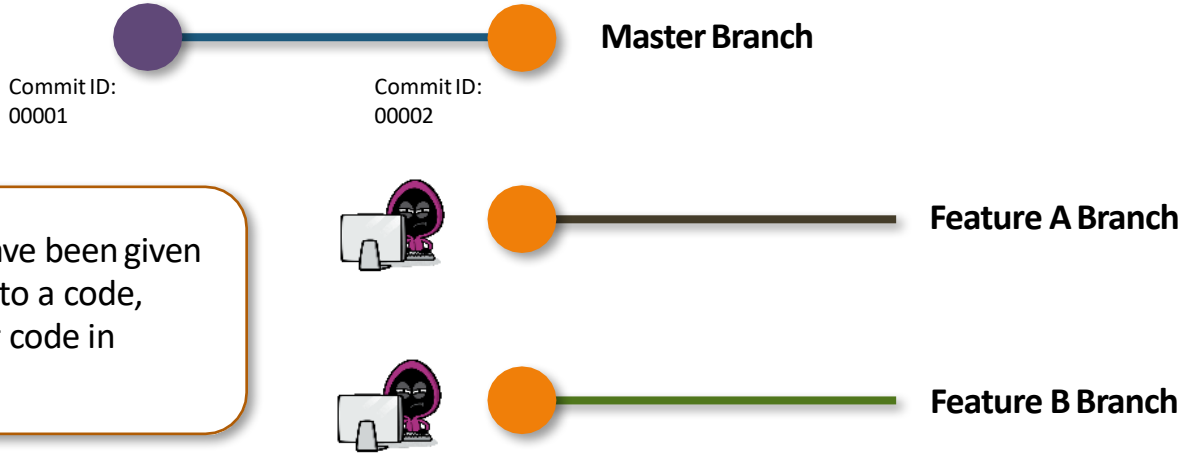
Merge Conflicts

Merge conflicts occur when we try to merge two branches, which have the same file updated by two different developers. Let's understand it using a scenario:



Imagine, two developers have been given the task of adding features to a code, they both have to add their code in ***functions.c*** file

Merge Conflicts

The functions.c file looks something like this as ofnow,

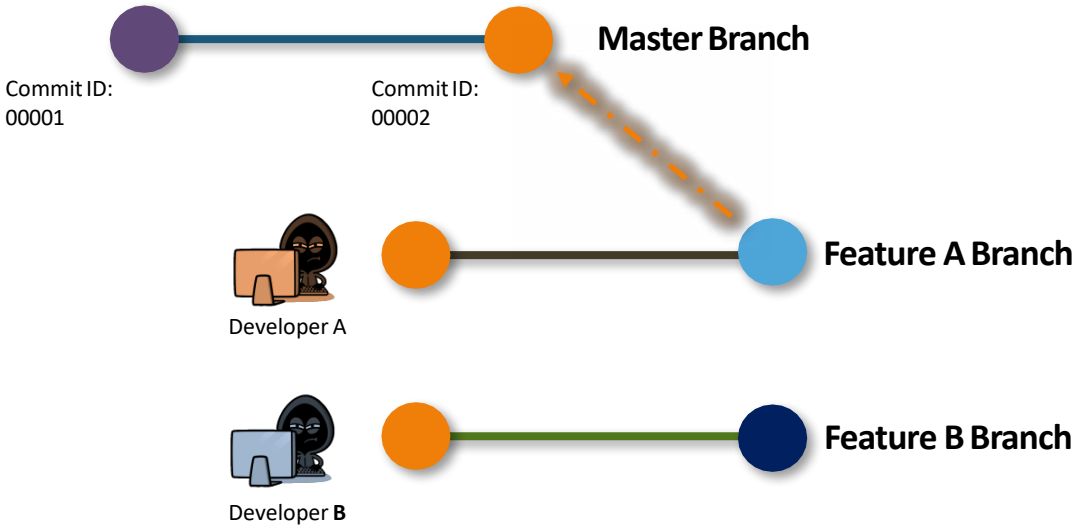
```
Main()
{
Function1()
  {
      //Initial Code
  }

}
```

function.c

Merge Conflicts

Developer A finished his code, and pushes the changes to the master branch



Merge Conflicts

```
Main()
{
Function1()
  {
      //Initial Code
  }
   Function2()
    {
        //Developer A Code
     }

}
```

function.c

After the **Developer A** changes his code and pushes it to master, the code on the **Master** branch looks something like this

```
Main()
{
Function1()
  {
      //Initial Code
  }
Function3()
    {
        //Developer B Code
    }

}
```

function.c

After the **Developer B** changes his code, the code on **Feature B** branch looks something like this

Merge Conflicts

Comparing the two code, we can see Feature A Branch is missing Developer A code. Therefore if we merge Feature A Branch with Master Branch, logically Developer A changes will disappear

Master Branch

```
Main()
{
Function1()
  {
      //Initial Code
  }
   Function2()
    {
        //Developer A Code
     }

}
```

function.c

Feature A Branch

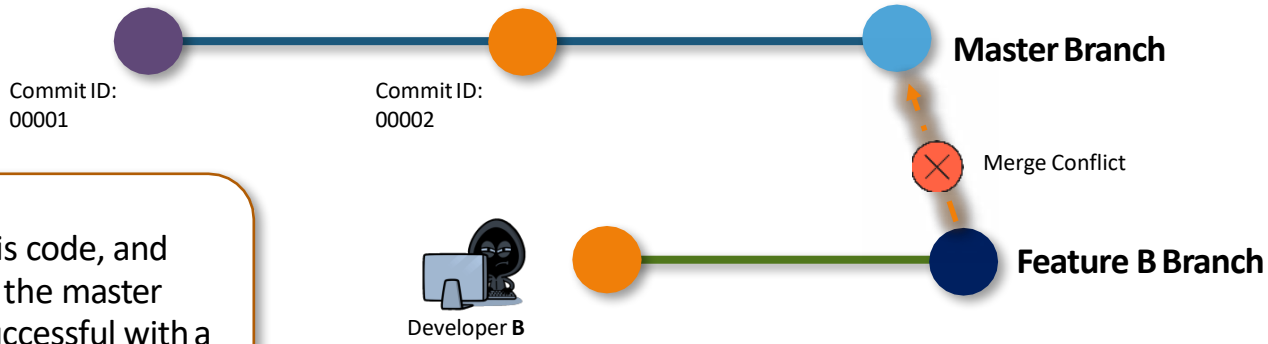
```
Main()
{
Function1()
  {
      //Initial Code
  }
Function3()
    {
        //Developer B Code
    }

}
```

function.c

Merge Conflicts

To solve this, git has a fail safe. If the Master branch has been moved forward in commits, compared to the branch which is being merged, it creates a conflict.



Developer B finished his code, and pushes the changes to the master branch. The merge is unsuccessful with a conflict message

Hands-on – Simulating a Merge Conflict

Merge Conflicts

This is the message, you will get when you merge a branch, which has a conflicting file

```
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.
```

Merge Conflict Message

How to resolve Merge Conflicts?

Once we have identified, there is a merge conflict we should go ahead and use the command

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):
```

Hit enter after this prompt, and then you should enter the mergetool

How to resolve Merge Conflicts?

The merge tool looks something like this, the top leftmost column is for Dev1 Branch changes, the centre 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.

How to resolve Merge Conflicts?

Once you have resolved the changes, save the file using "**:wq"**, vim command for save and exit. Do the same for all the files

How to resolve Merge Conflicts?

After this step, see the status of you 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

```
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
```

There will be some other files which have been created, these files are a copy of the original files which have been changed, you can delete them, if not needed.

How to resolve Merge Conflicts?

Finally commit your changes, to the branch and then push it to the remote repository

```
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
```