

Gitlab Conflict Resolution Guidelines

Introduction

Conflicts occur, usually, when two developers change the same metadata at the same time. Sometimes, these conflicts arise due to multiple user stories from the same developer.

Before choosing between your changes or the current changes in the VS Code, if the conflicts are not clear, it is a best practice to check the file in the target branch and analyse the existing metadata that is triggering the conflict. Contact the responsible developer if any doubts persist.

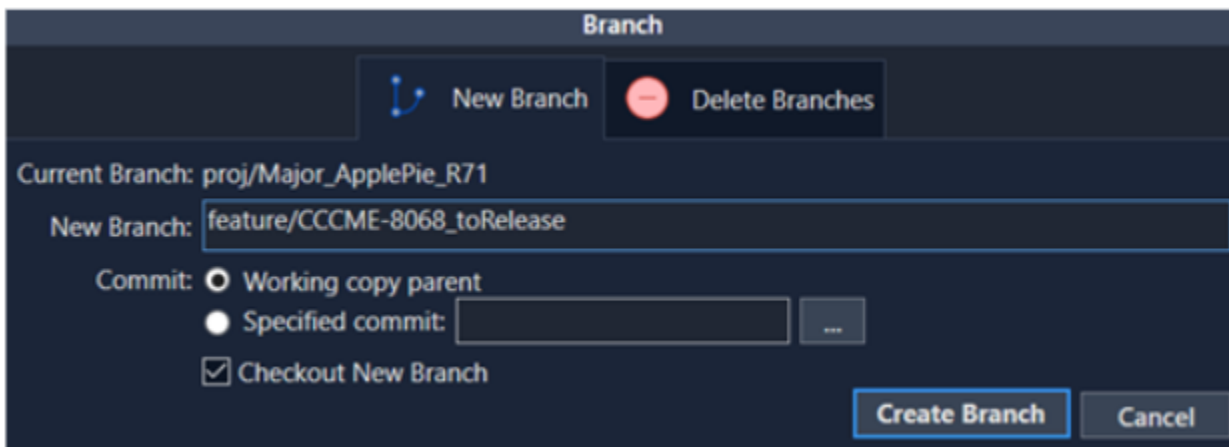
In the VS Code, choose the best option to solve the conflicts. It is possible that none of the options presented are completely accurate: in this case, you will need to choose one option and manually fix the file.

When done, stage your changes and commit them into the branch.

Steps

When merging your feature branch into a major branch, if the merge is blocked by conflicts, you should follow these steps:

1. Create a new branch "<Jira-number>_toRelease" from the current release branch

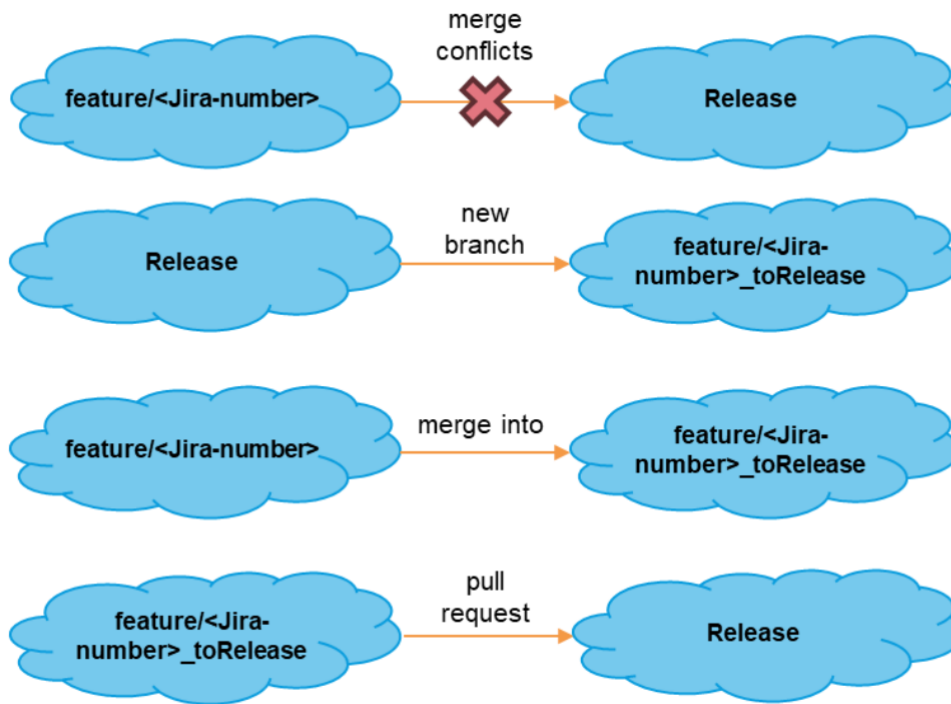


2. Merge your original feature branch "<Jira-number>" into the new branch that was created previously

3. Solve the conflicts, which includes selecting only your changes, and ignoring other modifications that were not committed by you

4. Commit all the changes to the current branch "<Jira-number>_toRelease"

5. Create a new pull request from the new branch "<Jira-number>_toRelease" into the current release branch (the old pull request, still blocked by the merge conflicts, should be closed automatically)



the same conflicts will appear here, to solve in this branch