Week :8

**Objectives**

* Explain how to resolve the conflict during merge.

In this hands-on lab, you will learn how to:

* Implement conflict resolution when multiple users are updating the trunk (or master) in such a way that it results into a conflict with the branch’s modification.

**Prerequisites**

The following are the pre-requisites to complete this hands-on lab:

* Hands-on ID: **“Git-T03-HOL\_001”**

Notes\*:

|  |
| --- |
| Please follow the below steps for creating a free account in GitHub.  Do not use cognizant credentials to login to GitHub. |

Estimated time to complete this lab: **30 minutes.**

Please follow the instructions to complete the hands-on. Each instruction expect a command for the Git Bash.

1. Verify if master is in clean state.
2. Create a branch **“GitWork”.** Add a file “hello.xml”.
3. Update the content of “hello.xml” and observe the status
4. Commit the changes to reflect in the branch
5. Switch to master.
6. Add a file **“hello.xml”** to the master and add some different content than previous.
7. Commit the changes to the master
8. Observe the log by executing **“git log –oneline –graph –decorate –all”**
9. Check the differences with Git diff tool
10. For better visualization, use P4Merge tool to list out all the differences between master and branch
11. Merge the bran to the master
12. Observe the git mark up.
13. Use 3-way merge tool to resolve the conflict
14. Commit the changes to the master, once done with conflict
15. Observe the git status and add backup file to the .gitignore file.
16. Commit the changes to the .gitignore
17. List out all the available branches
18. Delete the branch, which merge to master.
19. Observe the log by executing **“git log –oneline –graph –decorate”**

Output:

Microsoft Windows [Version 10.0.26100.4652]

(c) Microsoft Corporation. All rights reserved.

C:\Users\harsh>mkdir git-merge-conflict-lab

C:\Users\harsh>cd git-merge-conflict-lab

C:\Users\harsh\git-merge-conflict-lab>git init

Initialized empty Git repository in C:/Users/harsh/git-merge-conflict-lab/.git/

C:\Users\harsh\git-merge-conflict-lab>git status

On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

C:\Users\harsh\git-merge-conflict-lab>git checkout -b GitWork

Switched to a new branch 'GitWork'

C:\Users\harsh\git-merge-conflict-lab>echo "<msg>Hello from GitWork branch</msg>" > hello.xml

C:\Users\harsh\git-merge-conflict-lab>git add hello.xml

C:\Users\harsh\git-merge-conflict-lab>git commit -m "Added hello.xml in GitWork"

[GitWork (root-commit) d884957] Added hello.xml in GitWork

1 file changed, 1 insertion(+)

create mode 100644 hello.xml

C:\Users\harsh\git-merge-conflict-lab>echo "<msg>Updated content from GitWork branch</msg>" > hello.xml

C:\Users\harsh\git-merge-conflict-lab>git add hello.xml

C:\Users\harsh\git-merge-conflict-lab>git commit -m "Updated content in hello.xml in GitWork"

[GitWork c34dfc4] Updated content in hello.xml in GitWork

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

C:\Users\harsh\git-merge-conflict-lab>git checkout master

error: pathspec 'master' did not match any file(s) known to git

C:\Users\harsh\git-merge-conflict-lab>git checkout main

error: pathspec 'main' did not match any file(s) known to git

C:\Users\harsh\git-merge-conflict-lab>git branch -m main master

fatal: no branch named 'main'

C:\Users\harsh\git-merge-conflict-lab>echo "Initial file" > init.txt

C:\Users\harsh\git-merge-conflict-lab>git add init.txt

C:\Users\harsh\git-merge-conflict-lab>git commit -m "Initial commit"

[GitWork 17edbe0] Initial commit

1 file changed, 1 insertion(+)

create mode 100644 init.txt

C:\Users\harsh\git-merge-conflict-lab>git branch

\* GitWork

C:\Users\harsh\git-merge-conflict-lab>\* master

'\*' is not recognized as an internal or external command,

operable program or batch file.

C:\Users\harsh\git-merge-conflict-lab>git checkout -b GitWork

fatal: a branch named 'GitWork' already exists

C:\Users\harsh\git-merge-conflict-lab>echo "<msg>Different content from master branch</msg>" > hello.xml

C:\Users\harsh\git-merge-conflict-lab>git add hello.xml

C:\Users\harsh\git-merge-conflict-lab>git commit -m "Added conflicting hello.xml in master"

[GitWork 3163acb] Added conflicting hello.xml in master

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

C:\Users\harsh\git-merge-conflict-lab>git log --oneline --graph --decorate --all

\* 3163acb (HEAD -> GitWork) Added conflicting hello.xml in master

\* 17edbe0 Initial commit

\* c34dfc4 Updated content in hello.xml in GitWork

\* d884957 Added hello.xml in GitWork

C:\Users\harsh\git-merge-conflict-lab>git diff GitWork master -- hello.xml

fatal: bad revision 'master'

C:\Users\harsh\git-merge-conflict-lab>git merge GitWork

Already up to date.

C:\Users\harsh\git-merge-conflict-lab><<<<<<< HEAD

<< was unexpected at this time.

C:\Users\harsh\git-merge-conflict-lab><msg>Different content from master branch</msg>

The syntax of the command is incorrect.

C:\Users\harsh\git-merge-conflict-lab>=======

C:\Users\harsh\git-merge-conflict-lab><msg>Updated content from GitWork branch</msg>

The syntax of the command is incorrect.

C:\Users\harsh\git-merge-conflict-lab>>>>>>>> GitWork

>> was unexpected at this time.

C:\Users\harsh\git-merge-conflict-lab>git add hello.xml

C:\Users\harsh\git-merge-conflict-lab>git commit -m "Resolved merge conflict in hello.xml"

On branch GitWork

nothing to commit, working tree clean

C:\Users\harsh\git-merge-conflict-lab>echo "\*.orig" >> .gitignore

C:\Users\harsh\git-merge-conflict-lab>git add .gitignore

C:\Users\harsh\git-merge-conflict-lab>git commit -m "Ignored backup merge files"

[GitWork 71edba2] Ignored backup merge files

1 file changed, 1 insertion(+)

create mode 100644 .gitignore

C:\Users\harsh\git-merge-conflict-lab>git branch

\* GitWork

C:\Users\harsh\git-merge-conflict-lab>