

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**”

```

mkanm@Kanmani MINGW64 ~/GitDemo (GitNewBranch)
$ git config --global merge.tool p4merge
git config --global mergetool.keepBackup false

mkanm@Kanmani MINGW64 ~/GitDemo (GitNewBranch)
$ git checkout master
git status
Switched to branch 'master'
On branch master
nothing to commit, working tree clean

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git checkout -b GitWork
Switched to a new branch 'GitWork'

mkanm@Kanmani MINGW64 ~/GitDemo (GitWork)
$ echo "<message>Hello from GitWork branch</message>" > hello.xml
git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it

mkanm@Kanmani MINGW64 ~/GitDemo (GitWork)
$ git commit -m "Added hello.xml in GitWork branch"
[GitWork 895d9d2] Added hello.xml in GitWork branch
1 file changed, 1 insertion(+)
create mode 100644 hello.xml

mkanm@Kanmani MINGW64 ~/GitDemo (GitWork)
$ git checkout master
Switched to branch 'master'

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ echo "<message>Hello from master branch</message>" > hello.xml
git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git commit -m "Added hello.xml in master branch"
[master 07f8d27] Added hello.xml in master branch

```

```

MINGW64:/c/Users/mkanm/GitDemo
mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git commit -m "Added hello.xml in master branch"
[master 07f8d27] Added hello.xml in master branch
1 file changed, 1 insertion(+)
create mode 100644 hello.xml

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git log --oneline --graph --decorate --all
* 07f8d27 (HEAD -> master) Added hello.xml in master branch
| * 895d9d2 (GitWork) Added hello.xml in GitWork branch
|/
| * f3f4c80 (origin/GitNewBranch, GitNewBranch) Initial commit in GitNewBranch
|/
* eb39665 Added branchfile.txt in GitNewBranch
* 88c907a (origin/master) Added .gitignore to ignore log files and logs folder
* 2dd964b Added welcome.txt

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git diff master GitWork
diff --git a/hello.xml b/hello.xml
index 82868a7..0bf4c19 100644
--- a/hello.xml
+++ b/hello.xml
@@ -1,1 @@
- <message>Hello from master branch</message>
+ <message>Hello from GitWork branch</message>

mkanm@Kanmani MINGW64 ~/GitDemo (master)
$ git merge GitWork
Auto-merging hello.xml
CONFLICT (add/add): Merge conflict in hello.xml
Automatic merge failed; fix conflicts and then commit the result.

mkanm@Kanmani MINGW64 ~/GitDemo (master|MERGING)
$ notepad++ hello.xml

```

```
speech SS.txt x producer and consumer.txt x multithreading.txt x matrixmulti.txt x armstrong.txt x abstract class.txt x git
1 <<<<<<< HEAD
2 <message>Hello from master branch</message>
3 =====
4 <message>Hello from GitWork branch</message>
5 >>>>>>> GitWork
6
```

KANMANI MURUGHAIYAN

SUPERSET ID: 6407636