**4 . GIT - HOL**

**Git Conflict Resolution Hands-on Lab**

**Objectives**

* Explain how to resolve conflicts during a merge.

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

* Implement conflict resolution when multiple users update the trunk (or master), resulting in a conflict with branch modifications.

**Prerequisites**

* Hands-on ID: “Git-T03-HOL\_001”
* Please create a free GitHub account (do not use Cognizant credentials).
* Estimated time: 30 minutes.

**Step-by-step Instructions**

**Step 0: Setup or navigate to your Git repository**

* Open Git Bash.
* Navigate to your repo folder or create a new one:

mkdir GitDemo

cd GitDemo

git init

**Step 1: Verify if the main (or master) branch is clean**

git status

Output should show:

On branch main

nothing to commit, working tree clean

**Step 2: Create a branch GitWork and add a file hello.xml**

git branch GitWork

git checkout GitWork

echo "<message>Hello from GitWork branch</message>" > hello.xml

**Step 3: Update hello.xml content and check status**

echo "<update>First update on hello.xml</update>" >> hello.xml

git status

You will see hello.xml as untracked.

**Step 4: Stage and commit the file on GitWork branch**

git add hello.xml

git commit -m "Add hello.xml with initial content on GitWork branch"

**Step 5: Switch back to main branch**

git checkout main

**Step 6: Add a different hello.xml file on main**

echo "<message>Hello from main branch</message>" > hello.xml

git add hello.xml

git commit -m "Add hello.xml with different content on main branch"

**Step 7: View commit log for all branches**

git log --oneline --graph --decorate --all

This shows commits diverged between main and GitWork.

**Step 8: Check differences between main and GitWork**

git diff main GitWork

**Step 9: Use P4Merge tool to visualize differences**

**Set P4Merge as difftool:**

git config --global diff.tool p4merge

git config --global difftool.p4merge.cmd "p4merge \$LOCAL \$REMOTE"

**Run:**

git difftool main GitWork

**Step 10: Merge GitWork branch into main (conflict will occur)**

git merge GitWork

Git will report a merge conflict on hello.xml.

**Step 11: Resolve conflict using 3-way merge tool (P4Merge)**

**Run:**

git mergetool

* P4Merge will open showing base, local (main), and incoming (GitWork) versions.
* Manually resolve conflicts by editing the file in P4Merge.
* Save and close P4Merge.

**Step 12: Commit resolved changes to main**

git commit -m "Resolve merge conflict for hello.xml after merging GitWork"

**Step 13: Check status and add backup files to .gitignore**

git status

echo "\*.backup" >> .gitignore

git add .gitignore

git commit -m "Add backup files to .gitignore"

**Step 14: List all branches**

git branch -a

**Step 15: Delete the merged branch GitWork**

git branch -d GitWork

**Step 16: View final commit log graph**

git log --oneline --graph --decorate

**output**





