DN 4.0 JAVA FSE SOLUTIONS – WEEK 8

Skill: GIT

Hands-On 4

Objective

This document outlines the steps taken to create a merge conflict and use a 3-way merge tool to resolve it. The exercise demonstrates how to handle simultaneous changes to the same file in different branches.

Step 1: Create and Commit Changes on Two Branches

This section details the setup of the scenario that will cause the merge conflict.

- **Verify master branch status:** First, I verified that the master branch was in a clean state before starting the exercise.
    - **Command:**
    - git status

    Output:

    -
```
manas@Manasa MINGW64 ~/GitDemo (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```
- **Create and commit on a new branch:** I created a new branch named GitWork, added a file named hello.xml, and committed the changes to this branch.
    - **Commands:**
    - git checkout -b GitWork
    - echo "This is the content from the branch." > hello.xml
    - git add .
    - git commit -m "Add hello.xml to GitWork branch"

Output:

```
manas@Manasa MINGW64 ~/GitDemo (master)
$ git checkout -b GitWork
echo "This is the content from the branch." > hello.xml
git status
Switched to a new branch 'GitWork'
On branch GitWork
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hello.xml

nothing added to commit but untracked files present (use "git add" to track)
```

```
manas@Manasa MINGW64 ~/GitDemo (GitWork)
$ echo "Updated content from the branch." > hello.xml
git status
On branch GitWork
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hello.xml

nothing added to commit but untracked files present (use "git add" to track)

manas@Manasa MINGW64 ~/GitDemo (GitWork)
$ git add .
git commit -m "Update hello.xml in GitWork branch"
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next ti
me Git touches it
[GitWork 45956f1] Update hello.xml in GitWork branch
 1 file changed, 1 insertion(+)
 create mode 100644 hello.xml
```

- **Switch to master and create a conflict:** I switched back to the master branch and added a file with the same name, hello.xml, but with different content.
    - **Commands:**
    - git checkout master
    - echo "This is the content from the master." > hello.xml
    - git add .
    - git commit -m "Add hello.xml to master branch"

Output:

```
manas@Manasa MINGW64 ~/GitDemo (GitWork)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

manas@Manasa MINGW64 ~/GitDemo (master)
$ echo "This is the content from the master." > hello.xml

manas@Manasa MINGW64 ~/GitDemo (master)
$ git add .
git commit -m "Add hello.xml to master branch"
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next ti
me Git touches it
[master e64e383] Add hello.xml to master branch
 1 file changed, 1 insertion(+)
 create mode 100644 hello.xml
```

-

Step 2: Simulate and Resolve the Conflict

This section details how to merge the conflicting changes and resolve them using the P4Merge tool.

- **Attempt to merge:** I tried to merge the GitWork branch into master. As expected, this resulted in a merge conflict.
    - **Command:**
    - git merge GitWork

Output:

```
manas@Manasa MINGW64 ~/GitDemo (master)
$ git diff master GitWork
diff --git a/hello.xml b/hello.xml
index 4a16e4b..9f2c6ef 100644
--- a/hello.xml
+++ b/hello.xml
@@ -1 +1 @@
-This is the content from the master.
+Updated content from the branch.

manas@Manasa MINGW64 ~/GitDemo (master)
$ git difftool master GitWork

Viewing (1/1): 'hello.xml'
Launch 'p4merge' [Y/n]? y

manas@Manasa 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
```

- **Observe the git markup:** I opened the hello.xml file to see the conflict markers added by Git. The file now contains both versions of the content.
    - **Command:**
    - notepad++ hello.xml

Output:

```
manas@Manasa MINGW64 ~/GitDemo (master|MERGING)
$ notepad++ hello.xml
```

```
<<<<<<< HEAD
This is the content from the master.
=======
Updated content from the branch.
>>>>>>> GitWork
```

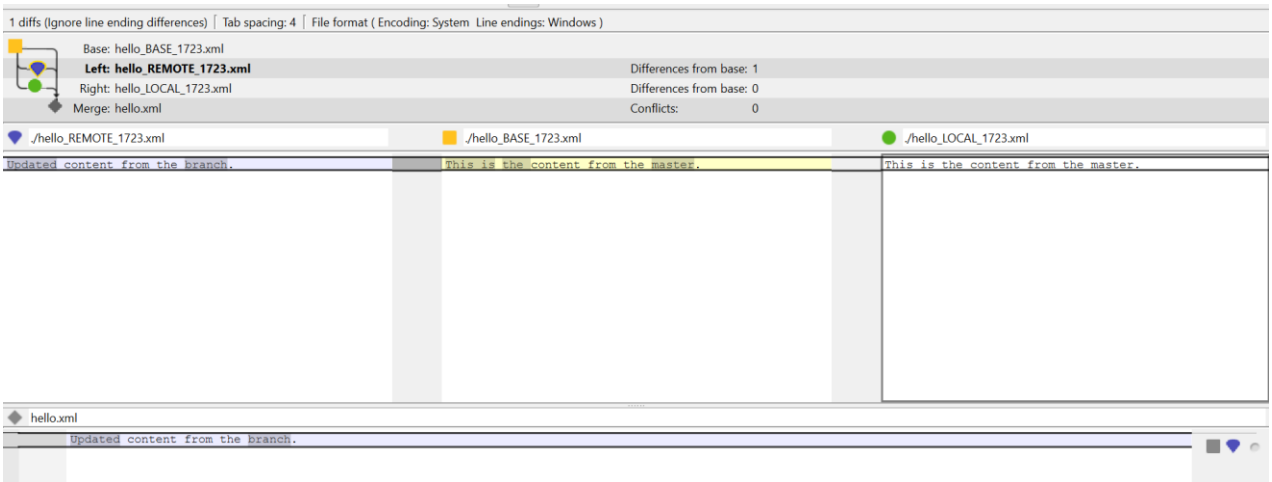
- **Resolve with mergetool:** I used git mergetool to open P4Merge, which provided a visual, 3-way view to resolve the conflict. I then chose to accept the desired changes, saved the file, and closed the tool.
    - **Command:**
    - git mergetool

Output:

```
manas@Manasa MINGW64 ~/GitDemo (master|MERGING)
$ 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:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecmerge
p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff
Merging:
hello.xml

Normal merge conflict for 'hello.xml':
  {local}: created file
  {remote}: created file
Hit return to start merge resolution tool (p4merge):
error: invalid path './hello_BASE_1723.xml'
```



- **Commit the resolved changes:** After resolving the conflict, I staged and committed the merged file.
    - **Commands:**

- git add hello.xml
- git commit -m "Merge branch 'GitWork' and resolve conflict"

Output:

```
manas@Manasa MINGW64 ~/GitDemo (master|MERGING)
$ git add .
git commit -m "Merge branch 'GitWork' into master and resolve conflict"
[master 76bc232] Merge branch 'GitWork' into master and resolve conflict
```

Step 3: Clean up and Verify

The final steps involve cleaning up the repository and verifying the result.

- **Update .gitignore:** After the merge, Git created a backup file with a .orig extension. I added a rule to the .gitignore file to ensure these files are ignored in the future.
  - **Commands:**
  - git status
  - echo "*.orig" >> .gitignore
  - git add .gitignore
  - git commit -m "Add .orig files to .gitignore"

Output:

```
manas@Manasa MINGW64 ~/GitDemo (master)
$ git status
echo "*.orig" >> .gitignore
On branch master
Your branch is ahead of 'origin/master' by 4 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

manas@Manasa MINGW64 ~/GitDemo (master)
$ git add .gitignore
git commit -m "Add .orig files to .gitignore"
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next t
ime Git touches it
[master 102c3f7] Add .orig files to .gitignore
 1 file changed, 1 insertion(+), 1 deletion(-)
```

- **Delete the branch:** I deleted the GitWork branch since its changes were successfully merged into master.

- **Command:**
- git branch -d GitWork

- **Observe the log:** I used git log to observe the commit history, which now shows a single merge commit.
  - **Command:**
  - git log --oneline --graph --decorate

Output:

```
manas@Manasa MINGW64 ~/GitDemo (master)
$ git branch -d GitWork
Deleted branch GitWork (was 45956f1).

manas@Manasa MINGW64 ~/GitDemo (master)
$ git log --oneline --graph --decorate
* 102c3f7 (HEAD -> master) Add .orig files to .gitignore
*   76bc232 Merge branch 'GitWork' into master and resolve conflict
|\
| * 45956f1 Update hello.xml in GitWork branch
* | e64e383 Add hello.xml to master branch
|/
* 41b3847 Add newfile.txt to GitNewBranch
* 580210e (origin/master) Add .gitignore to ignore log files and folder
* 309b924 First commit
```