

Resolving Merge Conflict

1. Verify if master is in clean state

```
$ git checkout main
Already on 'main'
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)
```

```
$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

2. Create branch GitWork and add hello.xml

```
$ git checkout -b Gitwork
Switched to a new branch 'Gitwork'
```

```
$ echo "<greeting>Hello from Gitwork Branch</greeting>">>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
```

3. Update content of hello.xml and observe status

```
$ echo "<greeting>Hello again from Gitwork branch - updated</greeting>">>hello.xml
```

```
$ git status
On branch Gitwork
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   hello.xml

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   hello.xml
```

4. Commit changes to branch

```
$ git add hello.xml
warning: in the working copy of 'hello.xml', LF will be replaced by CRLF the next time Git touches it
```

```
$ git commit -m "Updated hello.xml in Gitwork branch"
[Gitwork ba0287e] Updated hello.xml in Gitwork branch
1 file changed, 2 insertions(+)
create mode 100644 hello.xml
```

5. Switch to main

```
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 2 commits.
(use "git push" to publish your local commits)
```

6. Add hello.xml in main with different content

```
$ echo "<greeting>Hello from Main branch</greeting>">>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
```

7. Commit changes to main

```
$ git commit -m "Add hello.xml in Main branch"
[main 81197c2] Add hello.xml in Main branch
1 file changed, 1 insertion(+)
create mode 100644 hello.xml
```

8. Observe log

```
$ git log --oneline --graph --decorate --all
* 81197c2 (HEAD -> main) Add hello.xml in Main branch
| * ba0287e (Gitwork) Updated hello.xml in Gitwork branch
|/
* 4a1bb08 Add file1 and file2 in GitNewBranch
* 3302d54 Add .gitignore to ignore .log files and log folder
* 8ebf8cf (origin/main) This is a Multi-line commit message
* 9efd41a Initial commit
```

9. Check differences with gitdiff tool

```
$ git diff GitWork
diff --git a/hello.xml b/hello.xml
index 8cd4d7b..46cbd26 100644
--- a/hello.xml
+++ b/hello.xml
@@ -1,2 +1 @@
-<greeting>Hello from GitWork Branch</greeting>
-<greeting>Hello again from GitWork branch - updated</greeting>
+<greeting>Hello from Main branch</greeting>
```

10. Use P4Merge for better visualization

```
$ git difftool GitWork
viewing (1/1): 'hello.xml'
Launch 'p4merge' [Y/n]? Y
```



11. Merge branch into 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.
```

12. Observe the git mark-up

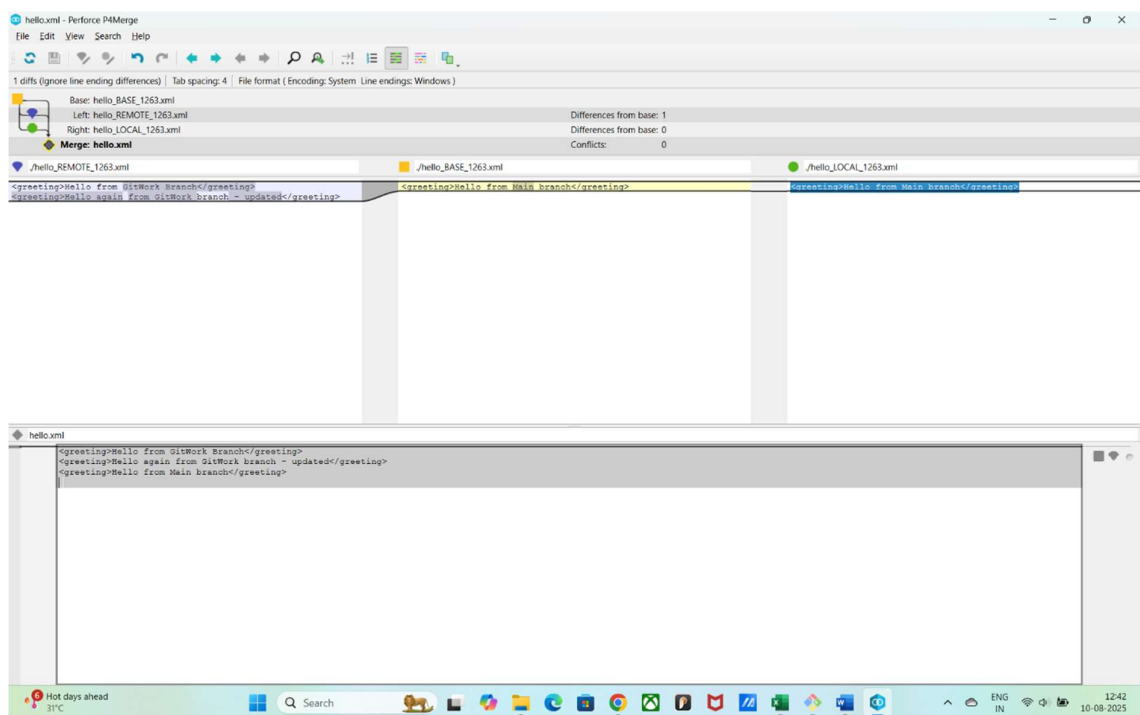
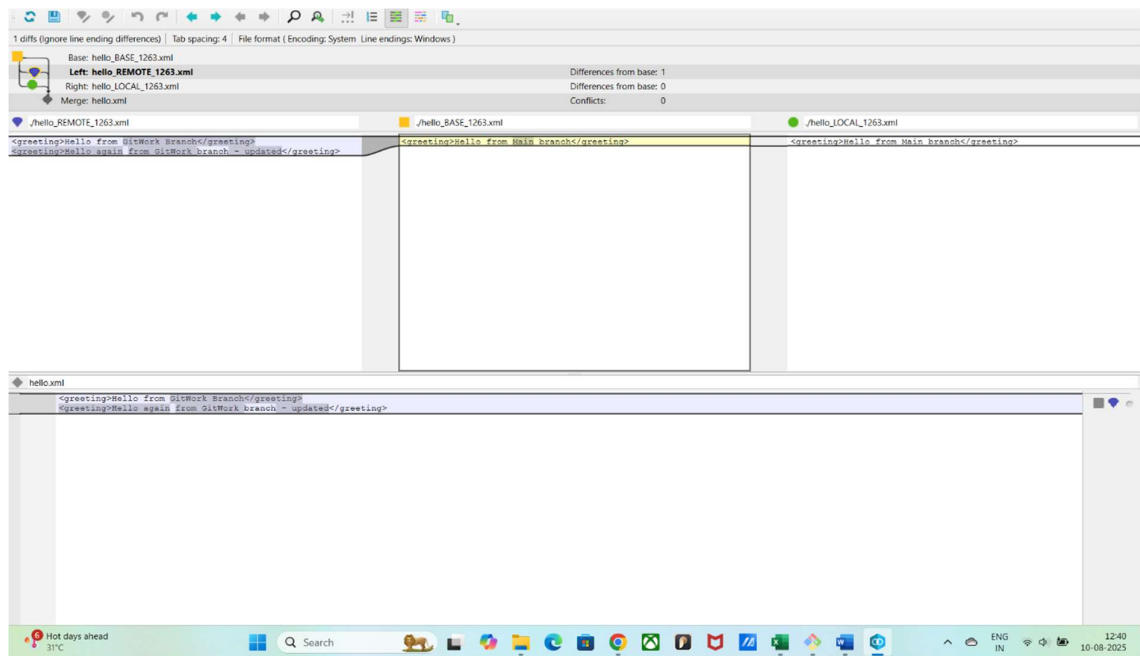
```
$ git status
On branch main
Your branch is ahead of 'origin/main' by 3 commits.
  (use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both added:      hello.xml

no changes added to commit (use "git add" and/or "git commit -a")
```

13. Use 3-way merge tool to resolve conflict



14. Commit after conflict resolution

```
$ git commit -m "Merged"  
[main 7afb431] Merged
```

15. Observe git status & add backup file to .gitignore

```
$ git status
On branch main
Your branch is ahead of 'origin/main' by 5 commits.
(use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hello.xml.orig

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

```
$ echo "*.orig" >> .gitignore
```

```
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced
by CRLF the next time Git touches it
```

```
$ git commit -m "Ignore merge backup files"
[main 05038e2] Ignore merge backup files
1 file changed, 1 insertion(+)
```

16. List all available branches

```
$ git branch
Gitwork
* main
```

17. Delete merged branch

```
$ git branch -d Gitwork
Deleted branch Gitwork (was ba0287e).
```

18. Observe final log

```
$ git log --oneline --graph --decorate
* 05038e2 (HEAD -> main) Ignore merge backup files
* 7afb431 Merged
| \
|  * ba0287e updated hello.xml in Gitwork branch
* | 81197c2 Add hello.xml in Main branch
|/
* 4a1bb08 Add file1 and file2 in GitNewBranch
* 3302d54 Add .gitignore to ignore .log files and log folder
* 8ebf8cf (origin/main) This is a Multi-line commit message
* 9efd41a Initial commit
```