

Merge Conflicts in Git

1 Scenario Setup

Suppose you have a repository with a single file:

```
repo/  
└─ names.txt
```

`names.txt` contains:

```
Alice  
Bob
```

Your branches:

- `main` → the default branch
 - `feature` → you created to add some names
-

2 Create a Conflict

Step 1: Modify in `main`

```
git checkout main  
echo "Charlie" >> names.txt  
git commit -am "Add Charlie to main"
```

`names.txt` now:

```
Alice  
Bob  
Charlie
```

Step 2: Modify in **feature** branch

```
git checkout -b feature
echo "David" >> names.txt
git commit -am "Add David to feature branch"
```

names.txt in **feature**:

```
Alice
Bob
David
```

3 Merge **feature** into **main**

```
git checkout main
git merge feature
```

Git Response:

```
Auto-merging names.txt
CONFLICT (content): Merge conflict in names.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Git cannot automatically decide what to do — **both branches changed the same part of the file.**

4 How Git Shows the Conflict

Open **names.txt**:

```
Alice
Bob
<<<<<<< HEAD
Charlie
=====
David
>>>>>> feature
```

- <<<<<< HEAD → content from **current branch (main)**
 - ===== → separates the changes
 - >>>>>> feature → content from **branch being merged (feature)**
-

5 Resolve the Conflict

You have options:

1. **Keep both changes:**

Alice
Bob
Charlie
David

2. **Choose main only:**

Alice
Bob
Charlie

3. **Choose feature only:**

Alice
Bob
David

4. **Combine creatively:**

Alice
Bob
Charlie & David

After editing, **stage the file**:

```
git add names.txt
```

Then **commit the merge**:

```
git commit -m "Resolved merge conflict in names.txt"
```

6 DevOps / Real-world Tips

Always pull latest changes before starting work:

```
git pull origin main
```

- - **Work in feature branches**, not `main`.
 - **Small, frequent commits** → conflicts easier to resolve.
 - Use **VS Code / GitKraken / GitHub Desktop** to resolve visually.
 - Merge conflicts are normal in DevOps teams — knowing how to resolve quickly is key.
-

✓ Key Takeaways:

Step	Git Command / Concept
See conflict	<code>git status</code> shows conflicted files
Resolve file manually	Edit file, remove <<<<<<, =====, >>>>>>
Stage resolved file	<code>git add <file></code>
Finish merge	<code>git commit</code>
