

# Resolving Conflicts

During team development, it is perfectly normal to get conflicts during team sharing actions (commit, update, etc). These conflicts must be resolved for the code to be consistent.

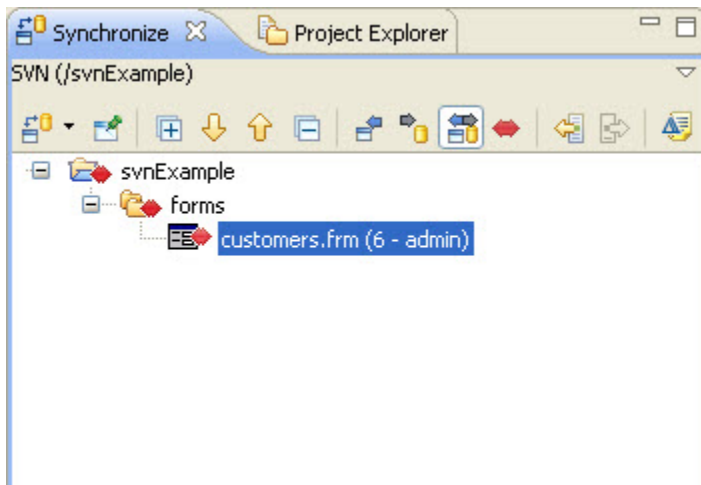
## Why Do Conflicts Occur

1. Technical Explanation - The reason a conflict occurs is during the update, commit, or synchronization process, the team provider has found a file that has been changed in both the developer and the repository versions since the developer has last performed a team action on the file.
2. Practical Reasons
  - Developers are working closely together in the code, making changes in the same workspace project files (example, working in the same form or form's script editor)
  - A developer has not ran an update before working in the code and there were changes the developer did not get before editing a file.
  - A developer has not regularly committed his code, and someone else has changed files since his last commit.

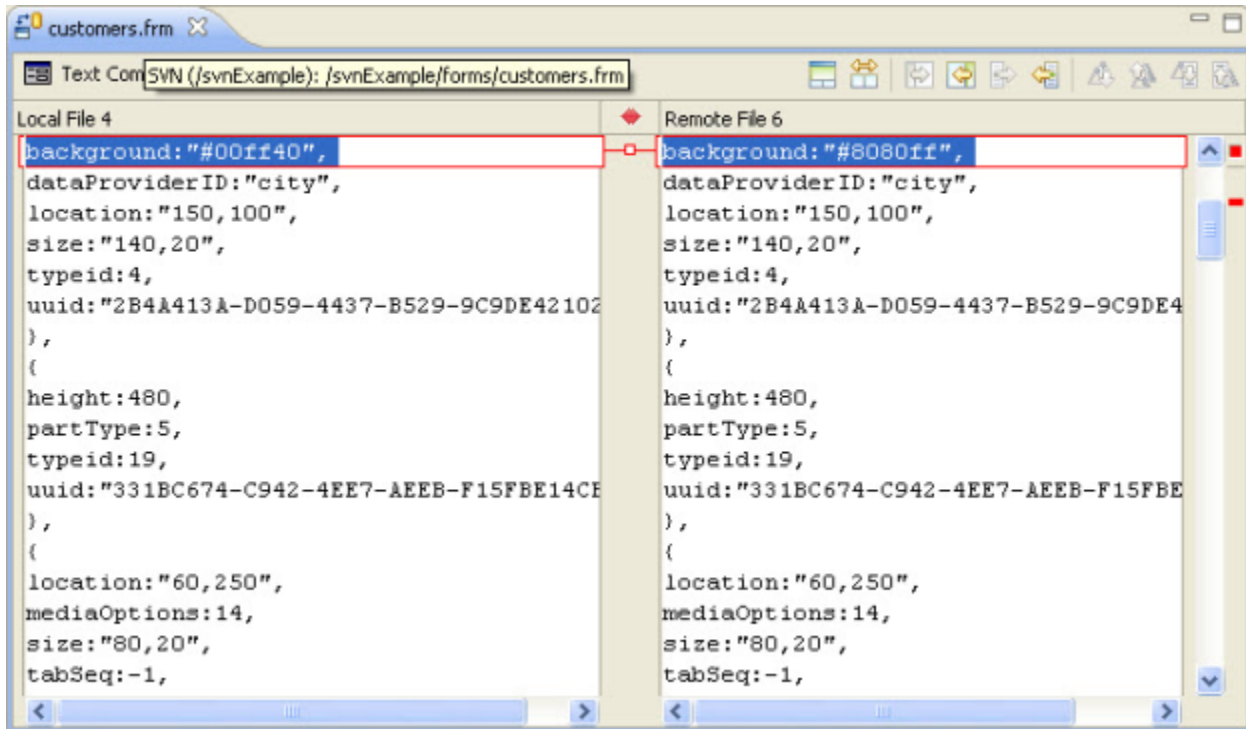
## Resolving Conflicts


Keep in mind that conflicts do not resolve themselves. The Team Synchronization perspective does give us tools that help resolve conflicts easier, but in the end, a developer must make a decision on what the correct code is. The basic premise of resolving conflicts is that the developer makes his copy of the code correct and then commits (one exception is the **Override and Update** command, which removes the conflict from the file and treats it like an update).

Conflicts will appear in the **Synchronize** view of the **Team Synchronization** perspective:



Conflicts may be fixed by hand. The easiest way to manual correct conflicts is to select which local changes should be quashed by conflicting incoming changes. Open the conflict in the **Compare Editor**.



You should see the conflicting values in the JSON file. To manually override a single conflict, select the small red handle between the conflicts. To override all local conflicts select the button  to copy all changes from right to left. You may also directly type into the local copy of the file and make the changes. After changing the file, you must right click on the local copy of the file and select **Mark as Merged**. This will mark the file to be committed in the next commit and remove the conflict.

You may also choose not to make any changes by hand, but rather decide that either your copy or the repository copy is correct. Right-Click the conflict(s) that you wish to change and from the contextual popup menu, select **Override and Update** to use the repository copy as the correct one, or select **Mark as Merged** to use the local copy as the correct one. **Override and Update** will be an update to your local workspace, and **Marked as Merged** will be a commit to the repository.

