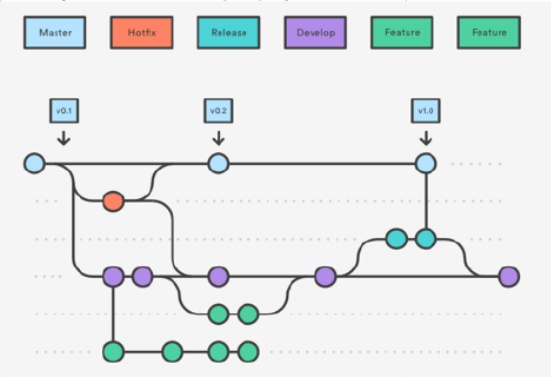


GIT BRANCH/MERGE example

1. [download kdiff3](#), [compile](#), [install](#) (setup git config --global to use kdiff3 as difftool and mergetool.
- a. http://collab.micron.com/mfg/MPGDRAMPE/PESOFT/WORMHOLE/Shared%20Documents/git_merge.pdf

| | | | |
|--|--|---|--|
| Gitflow Workflow | Same as Feature Branch Workflow except roles are assigned to feature branches. | | |
| | master | master contains abridged version with TAGS. | Tag with version number |
| | hotfix (short lived) | Only branch forked directly from master | Make fix, merge into master, and version up. |
| | release (short lived) | | |
| One time setup | develop | develop contains all history of project. | |
| git branch develop | |  | |
| git push -u origin develop | | | |
| Each Developer | | | |
| git clone ssh://user@host/path/to/repo.git | | | |
| git checkout -b develop origin/develop | | | |
| Developer works on feature | | | |
| git checkout -b <feature> develop | <feature1> | | |
| git status | <feature2> | | |
| git add/rm <files> | <featureN> | | |
| git commit -m "*****" | | | |
| Finish feature ->> | git pull origin develop | | |
| Merge <feature> to develop branch | git checkout develop | | |
| Delete <feature> branch AFTER you verify merge worked. | git merge --no-ff <feature> | | |
| | git push | Same as Centralized Workflow | |
| | Conflicts ?? --> | | |
| | git branch -d <feature> | | |
| | Ready to release?? --> | git checkout -b release-<#.#> develop | Feature frozen |
| | | Update documents, test, and prepare for release | |
| | | Release -> | git checkout master |
| Found a Bug!! Checkout from master. | | | git merge --no-ff release-<#.#> |
| git checkout -b issue-<####> master | | | git push |
| Fix the bug and merge fix to master and tag. | | | git checkout develop |
| git checkout master | | | git merge --no-ff release-<#.#> |
| git merge --no-ff issue-<####> | | | git push |
| git push | | | git branch -d release-<#.#> |
| git tag -a <#.#.####> -m "*****" master | | | git tag -a <#.#> -m "*****" master |
| git push --tags | | | git push --tags |
| Update develop branch if no release branch exists | | | Done |
| git checkout develop | | | |
| git merge --no-ff issue-<####> | | | |
| git push | | | |
| git branch -d issue-<####> | | | |
| Update release branch if release branch exists | | | |
| git checkout release-<####> | | | |
| git merge --no-ff issue-<####> | | | |
| git push | | | |
| git branch -d issue-<####> | | | |
| When release is complete, it will push to develop. | | | |

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [228]$ git config --global --list | egrep "^user|^core.excludes|^merge|^diff"
user.name=Francis Trujillo
user.email=fttrujillo@micron.com
core.excludesfile=/home/fttrujillo/.gitignore
merge.tool=kdifff3
diff.tool=kdifff3
difftool.kdifff3.cmd=kdifff3 $LOCAL $REMOTE
difftool.prompt=false
```

=====

FIRST TIME BY ADMIN

=====

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [196]$ git branch develop
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [197]$ git push -u origin develop
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [199]$ git branch -v
develop dfa9dd1 added new file to master
* master dfa9dd1 added new file to master
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [200]$ git branch -r
origin/develop
origin/master
```

=====

FIRST TIME BY DEVELOPER

=====

o Add a branch called 'some_feature' from develop

=====

(SKIP clone if you have this already)

```
git clone ssh://git@stash.micron.com:7999/pesoft/git-learning.git
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [202]$ git checkout -b some_feature develop
Switched to a new branch 'some_feature'
```

=====

VERIFY WHICH LOCAL BRANCH WE ARE EDITING

=====

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [203]$ git branch -v
  develop    dfa9dd1 added new file to master
  master     dfa9dd1 added new file to master
* some_feature dfa9dd1 added new file to master
```

=====

ADD DIRECTORY/FILE, COMMIT TO some_feature BRANCH..

***** **DO NOT, DO NOT PUSH AFTER COMMIT !!!** *****

=====

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [204]$ mkdir my_cool_feature
pesoft9:/u/fttrujillo/PESOFT/git-learning [205]$ cd my_cool_feature
pesoft9:/u/fttrujillo/PESOFT/git-learning/my_cool_feature [206]$ cat > something_cool.txt
This is
to setup a file
that we can use as a MERGE conflict later.
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [211]$ git add .
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [212]$ git status
On branch some_feature
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
```

```
    new file:   my_cool_feature/something_cool.txt
pesoft9:/u/fttrujillo/PESOFT/git-learning [213]$ git commit -m "this is the initial something cool"
[some_feature 2fc4566] this is the initial something cool
1 file changed, 3 insertions(+)
create mode 100644 my_cool_feature/something_cool.txt
```

***** **DO NOT, DO NOT PUSH AFTER COMMIT !!!** *****

=====

FEATURE FINISHED

- o pull develop to ensure up to date
- o switch to develop branch and MERGE in your feature.
- o push merged changes into develop

=====

pesoft9:/u/fttrujillo/PESOFT/git-learning [218]\$ **git pull origin develop**

From ssh://stash.micron.com:7999/pesoft/git-learning

* branch develop -> FETCH_HEAD

Already up-to-date.

pesoft9:/u/fttrujillo/PESOFT/git-learning [219]\$ **git checkout develop**

Switched to branch 'develop'

Your branch is up-to-date with 'origin/develop'.

pesoft9:/u/fttrujillo/PESOFT/git-learning [220]\$ **git merge --no-ff some_feature**

Merge made by the 'recursive' strategy.

my_cool_feature/something_cool.txt | 3 +++

1 file changed, 3 insertions(+)

create mode 100644 my_cool_feature/something_cool.txt

pesoft9:/u/fttrujillo/PESOFT/git-learning [221]\$ **git push**

Counting objects: 5, done.

Delta compression using up to 8 threads.

Compressing objects: 100% (4/4), done.

Writing objects: 100% (5/5), 578 bytes | 0 bytes/s, done.

Total 5 (delta 1), reused 0 (delta 0)

remote:

remote: Create pull request for develop:

remote: <https://stash.micron.com/stash/projects/PESOFT/repos/git-learning/compare/commits?sourceBranch=refs/heads/develop>

remote:

To ssh://git@stash.micron.com:7999/pesoft/git-learning.git

dfa9dd1..259f797 develop -> develop

pesoft9:/u/fttrujillo/PESOFT/git-learning [230]\$ **git branch -d some_feature**

Deleted branch some_feature (was 2fc4566).

pesoft9:/u/fttrujillo/PESOFT/git-learning [231]\$ **git branch -v**

* develop 259f797 Merge branch 'some_feature' into develop

master dfa9dd1 added new file to master

=====

LOGS

=====

pesoft9:/u/fttrujillo/PESOFT/git-learning [223]\$ **alias glog**
git log --graph --pretty=format:"%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset" --abbrev-commit --date=relative --all

pesoft9:/u/fttrujillo/PESOFT/git-learning [222]\$ **glog**
* 259f797 - (HEAD, origin/develop, develop) Merge branch 'some_feature' into develop (22 minutes ago) <Francis Trujillo>
|\n| * 2fc4566 - this is the initial something cool (33 minutes ago) <Francis Trujillo>
|/
* dfa9dd1 - (origin/master, master) added new file to master (46 minutes ago) <Francis Trujillo>
* 600f2bc - initial commit (47 minutes ago) <Francis Trujillo>

=====

SIMULATE USER1 on host nsglnxdev1.micron.com branching and changing a file.

=====

nsglnxdev1:/home/fttrujillo/PESOFT/git-learning [117]\$ **git checkout -b shared_feature develop**
Switched to a new branch 'shared_feature'

nsglnxdev1:/home/fttrujillo/PESOFT/git-learning [119]\$ **vi my_cool_feature/something_cool.txt**

nsglnxdev1:/home/fttrujillo/PESOFT/git-learning [120]\$ **cat my_cool_feature/something_cool.txt**
This is
to setup a file

I am adding a line to create merge conflict.
that we can use as a MERGE conflict later.
Also, adding a line after.....

DO git add and git commit.....

=====

SIMULATE USER2 on host pesoft9.micron.com also branching new feature and changing the same file as USER1

=====

pesoft9:/u/fttrujillo/PESOFT/git-learning [234]\$ **git checkout -b next_feature develop**
Switched to a new branch 'next_feature'

pesoft9:/u/fttrujillo/PESOFT/git-learning [235]\$ **vi my_cool_feature/something_cool.txt**

pesoft9:/u/fttrujillo/PESOFT/git-learning [236]\$ **cat my_cool_feature/something_cool.txt**
This is
to setup a file
that we can use as a MERGE conflict later.
[This is 2nd user line.](#)

=====

NOW, LET'S HAVE USER1 DO MERGE - This developer is first to check in changes.

=====

nsglnxdev1:/home/fttrujillo/PESOFT/git-learning [121]\$ **git pull origin develop**
From ssh://stash.micron.com:7999/pesoft/git-learning
* branch develop -> FETCH_HEAD
Already up-to-date.

nsglnxdev1:/home/fttrujillo/PESOFT/git-learning [134]\$ **git checkout develop**
Switched to branch 'develop'

```
nsglnxdev1:/home/fttrujillo/PESOFT/git-learning [135]$ git merge --no-ff shared_feature
Merge made by recursive.
 my_cool_feature/something_cool.txt | 2 ++
1 files changed, 2 insertions(+), 0 deletions(-)
nsglnxdev1:/home/fttrujillo/PESOFT/git-learning [136]$ git push
Counting objects: 8, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 536 bytes, done.
Total 5 (delta 2), reused 0 (delta 0)
remote:
remote: Create pull request for develop:
remote: https://stash.micron.com/stash/projects/PESOFT/repos/git-learning/compare/commits?sourceBranch=refs/heads/develop
remote:
To ssh://git@stash.micron.com:7999/pesoft/git-learning.git
 259f797..12edc0c develop -> develop
```

```
nsglnxdev1:/home/fttrujillo/PESOFT/git-learning [137]$ git branch -d shared_feature
Deleted branch shared_feature (was 095e493).
```

=====

NOW, LET'S HAVE USER2 MERGE.

=====

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [238]$ g
STATUS
=====
On branch next_feature
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   my_cool_feature/something_cool.txt

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

M my_cool_feature/something_cool.txt
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [239]$ git add .
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [240]$ git commit -m "adding my two cents"
```

```
[next_feature e522d7c] adding my two cents
```

```
1 file changed, 2 insertions(+)
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [241]$ git pull origin develop
```

```
From ssh://stash.micron.com:7999/pesoft/git-learning
```

```
* branch      develop    -> FETCH_HEAD
```

```
Auto-merging my_cool_feature/something_cool.txt
```

```
CONFLICT (content): Merge conflict in my_cool_feature/something_cool.txt
```

```
Automatic merge failed; fix conflicts and then commit the result.
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [243]$ git branch -v
```

```
develop      259f797 [behind 2] Merge branch 'some_feature' into develop
```

```
master       dfa9dd1 added new file to master
```

```
* next_feature e522d7c adding my two cents
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [245]$ git fetch origin develop
```

```
From ssh://stash.micron.com:7999/pesoft/git-learning
```

```
* branch      develop    -> FETCH_HEAD
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [251]$ g
```

```
STATUS
```

```
=====
```

```
On branch next_feature
```

```
You have unmerged paths.
```

```
(fix conflicts and run "git commit")
```

```
Unmerged paths:
```

```
(use "git add <file>..." to mark resolution)
```

```
both modified: my_cool_feature/something_cool.txt
```

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

```
UU my_cool_feature/something_cool.txt
```



```
pesoft9:/u/fttrujillo/PESOFIT/git-learning [252]$ git mergetool my_cool_feature/something_cool.txt
```

Merging:

my_cool_feature/something_cool.txt

Normal merge conflict for 'my_cool_feature/something_cool.txt':

{local}: modified file

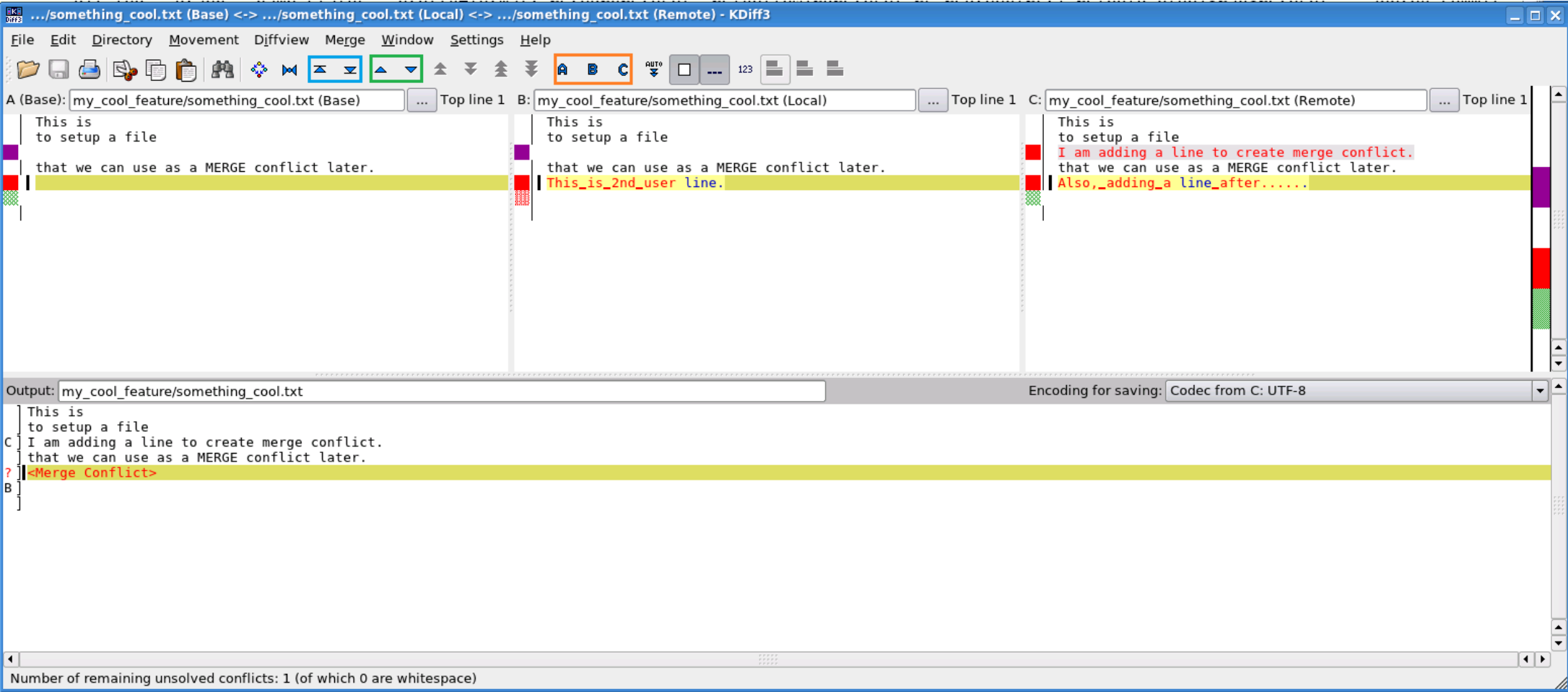
{remote}: modified file

Whew....we had to work to get a simple CONFLICT.

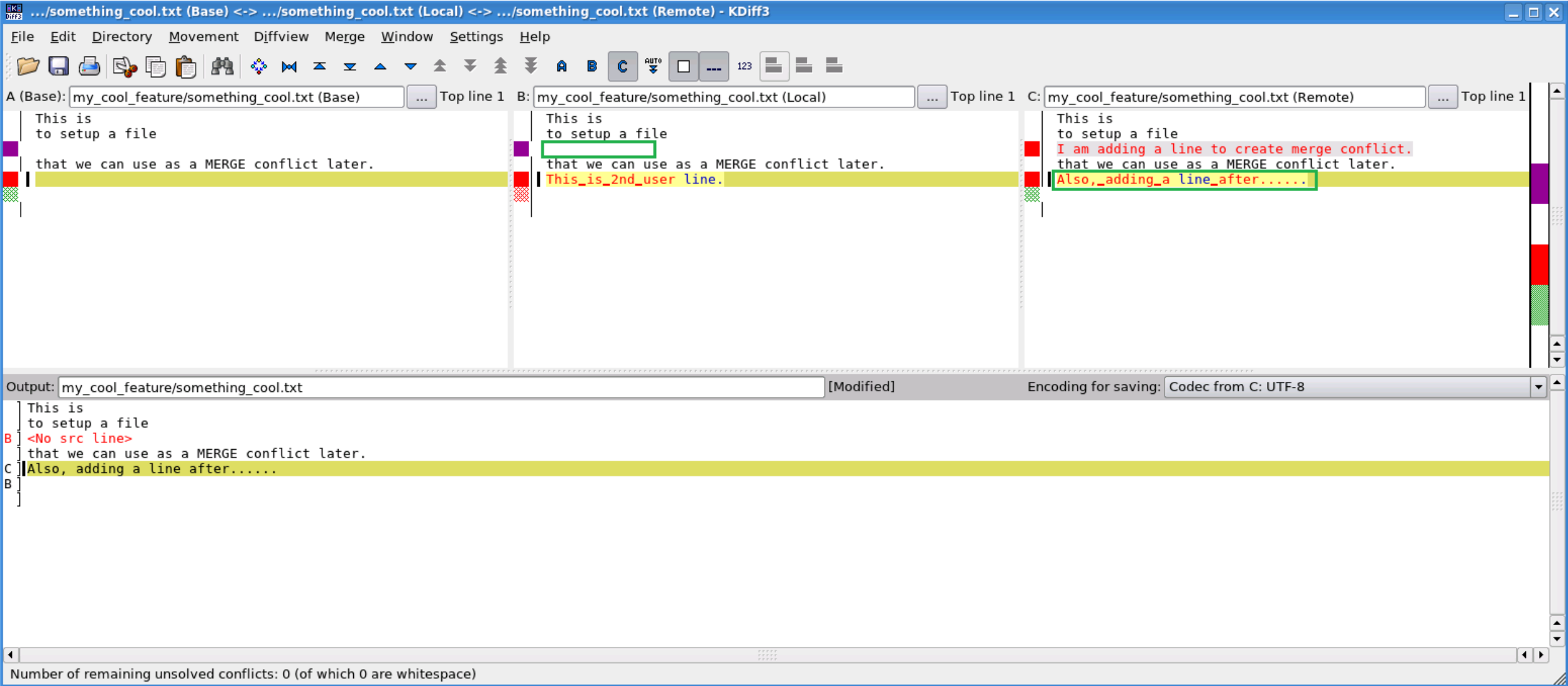
Now, if you have git configured correctly the last git mergetool command should pull up this GUI.

BASE is local and shows the file before we started mucking with it in **A**.
Local shows USER2(me) changes in **B**.
Remote shows USER1 changes in **C**.

Blue arrows GoToFirstDelta, GoToLastDelta
Green arrows GoToPrevDelta, GoToNextDelta (after you position the yellow using arrows, then choose A, B, or C)



In this example, I chose B for the first delta and C for the second delta.
Notice, the **little ?** and words **Merge Conflict** disappeared in the bottom window.
SAVE. EXIT.



```
pesoft9:/u/fttrujillo/PESOFT/git-learning [276]$ g
STATUS
```

=====
On branch next_feature
All conflicts fixed but you are still merging.
(use "git commit" to conclude merge)

Changes to be committed:

modified: my_cool_feature/something_cool.txt

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

my_cool_feature/something_cool.txt.orig ← We do not need this file. Nice to have to diff against if paranoid.

```
M my_cool_feature/something_cool.txt
?? my_cool_feature/something_cool.txt.orig
```

This is to show you the original file and your merge conflict resolution.

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [277]$ sdiff my_cool_feature/something_cool.txt.orig my_cool_feature/something_cool.txt
This is                                     This is
to setup a file                             to setup a file
I am adding a line to create merge conflict. <
that we can use as a MERGE conflict later.  <
<<<<<<< HEAD                               <
This is 2nd user line.                       <
                                              <
=====                                     <
Also, adding a line after.....              Also, adding a line after.....
>>>>>>> 12edc0ce2759ea6b388394af0685cfabf7f1b956 |
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [278]$ rm my_cool_feature/something_cool.txt.orig
/bin/rm: remove regular file `my_cool_feature/something_cool.txt.orig'? y
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [279]$ g
STATUS
```

```
=====
```

```
On branch next_feature
```

```
All conflicts fixed but you are still merging.
```

```
(use "git commit" to conclude merge)
```

```
Changes to be committed:
```

```
    modified:   my_cool_feature/something_cool.txt
```

```
M   my_cool_feature/something_cool.txt
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [280]$ git commit -m "merged conflict resolved"
[next_feature 8fe2636] merged conflict resolved
```

Now, start at the top of the recipe again.....do not skip....read the cheatsheet.

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [285]$ git pull origin develop
```

```
From ssh://stash.micron.com:7999/pesoft/git-learning
```

```
* branch      develop  -> FETCH_HEAD
```

```
First, rewinding head to replay your work on top of it...
```

```
Fast-forwarded develop to 12edc0ce2759ea6b388394af0685cfabf7f1b956.
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [286]$ git checkout develop
```

```
Already on 'develop'
```

```
Your branch is up-to-date with 'origin/develop'.
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [287]$ git merge --no-ff next_feature
```

```
Merge made by the 'recursive' strategy.
```

```
my_cool_feature/something_cool.txt | 2 +-  
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [288]$ git push
Counting objects: 9, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 923 bytes | 0 bytes/s, done.
Total 9 (delta 3), reused 0 (delta 0)
remote:
remote: Create pull request for develop:
remote:  https://stash.micron.com/stash/projects/PESOFT/repos/git-learning/compare/commits?sourceBranch=refs/heads/develop
remote:
To ssh://git@stash.micron.com:7999/pesoft/git-learning.git
    12edc0c..ed63d87  develop -> develop
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [289]$ git branch -v
* develop    ed63d87 Merge branch 'next_feature' into develop
master      dfa9dd1 added new file to master
next_feature 8fe2636 merged conflict resolved
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [290]$ git branch -d next_feature
Deleted branch next_feature (was 8fe2636).
```

```
pesoft9:/u/fttrujillo/PESOFT/git-learning [291]$ git branch -v
* develop ed63d87 Merge branch 'next_feature' into develop
master dfa9dd1 added new file to master
```

```
pesoft9:/u/ftrujillo/PES0FT/git-learning [292]$ glog2
```

```
*   ed63d87 - (HEAD, origin/develop, develop) Merge branch 'next_feature' into develop (7 minutes ago) <Francis Trujillo>
```

```
| \
| *   8fe2636 - merged conflict resolved (10 minutes ago) <Francis Trujillo>
```

```
| /
| \
| *   12edc0c - Merge branch 'shared_feature' into develop (57 minutes ago) <Francis Trujillo>
```

```
| \
| *   095e493 - my updates on cool feature (59 minutes ago) <Francis Trujillo>
```

```
| /
| |   M my_cool_feature/something_cool.txt
```

```
| *   e522d7c - adding my two cents (54 minutes ago) <Francis Trujillo>
```

```
| /
| |   M   my_cool_feature/something_cool.txt
```

```
*   259f797 - Merge branch 'some_feature' into develop (2 hours ago) <Francis Trujillo>
```

```
| \
| *   2fc4566 - this is the initial something cool (2 hours ago) <Francis Trujillo>
```

```
| /
| |   A   my_cool_feature/something_cool.txt
```

```
*   dfa9dd1 - (origin/master, master) added new file to master (2 hours ago) <Francis Trujillo>
```

```
| A     linux_repo_aliases.txt
```

```
*   600f2bc - initial commit (2 hours ago) <Francis Trujillo>
```

```
| A     README.txt
```

```
pesoft9:/u/ftrujillo/PES0FT/git-learning [293]$ █
```