GIT corrupt files (<<<<<<<<HEAD)

Asked 6 years, 9 months ago Active 3 years, 2 months ago Viewed 18k times

14

3

I have this in my files after some trouble with VS2012 git-plugin:

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
<<<<<<< HEAD
using NHibernate;
=======
>>>>>>> dd2c3d7dfe81074e7c5a73f8e4ca2584481a74f1

namespace Controll.Hosting.Tests
{
[TestClass]
public class TestBase
{
<<<<<<< HEAD
    protected ISessionFactory SessionFactory;

    [TestInitialize]
    public void InitializeTestBase()
    {
            SessionFactory = NHibernateHelper.GetSessionFactoryForMockedData();
=======
    [ClassInitialize]
    public void InitializeTest()
    {
        Console.WriteLine("Settings NHibernateHelper.IsInTesting -> True");
        NHibernateHelper.IsInTesting = true;
>>>>>>> dd2c3d7dfe81074e7c5a73f8e4ca2584481a74f1
        }
    }
}
```

How can i reset my files?

git git-merge merge-conflict-resolution

edited Mar 8 '13 at 12:15
cfi
9,076 6 49 85

asked Mar 8 '13 at 9:49
ErikTJ
1,771 2 18 37

8 Those are merge conflicts. Files are not corrupted. – vedarthk Mar 8 '13 at 9:51

I see, can i force git overwrite everything with the new changes? – ErikTJ Mar 8 '13 at 9:52

4 possible duplicate of [Git merge left marks in my files](#) – random Jul 15 '13 at 3:27

@vedarthk: When this makes scripts nonfunctional, or when it happens to data files, or logs — I venture to say, for any file type — then yes, the merge conflicts have corrupted the files. Sure, the files can be recovered, but if they are made to be a non-functional state, I call them corrupted. – jvriesem Nov 14 at 22:29

3 Answers

23

What you had wasn't **trouble** but **conflicts**. This happens when the files are modified by two different persons at the same place (you both add/remove/modify things inside the same lines).

You can simply update your files manually, by deciding to keep everything between `<<<<<<< HEAD` and `=======` , or between `=======` and `>>>>>>>` , or some mix of the two. Once you *resolve* all your *conflicts*, you just need to commit your changes.

To discard local changes on a file, you can do

```
git checkout yourfile
```

or, for all files using

```
git checkout -- .
```

You can also decide, for each file, if you want to keep your version or the repository version with

```
git checkout --ours yourfile # Your version
git checkout --theirs yourfile # Repository version
```

answered Mar 8 '13 at 9:53
alestanis
19.1k 4 39 64

Ok, thanks. I will accept this, but can i chose to keep all my local changes (im moving back to a really old repo and many files are affected)? – ErikTJ Mar 8 '13 at 9:55

1 @ErikTJ You shouldn't decide to do something on *all* files. The best thing, even if it's a bit annoying, is to look at each file and decide according to what git points out as conflicts. – alestanis Mar 8 '13 at 9:57

Your Q is answered best by alestanis already. Still for easy lookup:

There's more info about merging at this Q.

And `git help merge` is quite explicitly helpful as well:

> HOW TO RESOLVE CONFLICTS
>
> After seeing a conflict, you can do two things:
>
> · Decide not to merge. The only clean-ups you need are to reset the index file to the HEAD commit to reverse 2. and to clean up working tree changes made by 2. and 3.; `git merge --abort` can be used for this.
>
> · Resolve the conflicts. Git will mark the conflicts in the working tree. Edit the files into shape and `git add` them to the index. Use `git commit` to seal the deal.
>
> You can work through the conflict with a number of tools:
>
> · Use a mergetool. `git mergetool` to launch a graphical mergetool which will work you through the merge.
>
> · Look at the diffs. `git diff` will show a three-way diff, highlighting changes from both the HEAD and MERGE_HEAD versions.
>
> · Look at the diffs from each branch. `git log --merge -p <path>` will show diffs first for the HEAD version and then the MERGE_HEAD version.
>
> · Look at the originals. `git show :1:filename` shows the common ancestor, `git show :2:filename` shows the HEAD version, and `git show :3:filename` shows the MERGE_HEAD version.

edited May 23 '17 at 12:32
Community ♦
1 1

answered Mar 8 '13 at 10:02
cfi
9,076 6 49 85

0

Using SourceTree for git to manage my builds with Kdiff installed has helped me resolve 99% of these issues really efficiently.

I'm just left with trying to remove these from my SOU file in .Net, usually replacing the SOU file with an older version resolves this.

answered Oct 25 '16 at 14:01
Dan
11 4
