

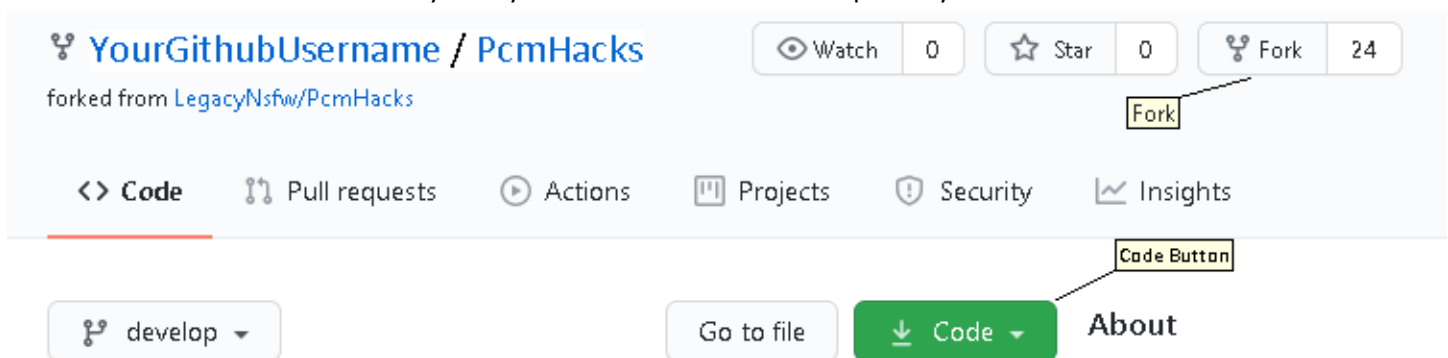
# How to Git with PcmHacks in Visual Studio

If not already done,

- Install Visual Studio. (2017 & 2019 using the built in Git were used to create this document)
- Optional, Install [Prebuilt Windows Toolchain for m68k-elf](#) (m68k-elf-gcc\*.exe)  
Not required unless you intend to build the kernel yourself, else copy kernel.bin from a release zip.  
It's a default install, you can Un-Check; 'Add binary directory to %PATH%' it is not needed.
- Create a GitHub account.

Open your favorite internet browser and login to GitHub,

- Navigate to the [LegacyNsfw PcmHacks GitHub repository](#).
- Select: Fork  
This creates a Fork of the PcmHacks repository in your GitHub Repositories.  
When done it should leave you in your fork of the PcmHacks repository.

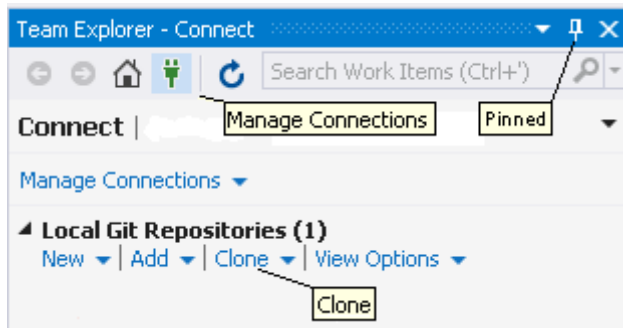


- Select: Code (the Button)  
Make note of the Clone HTTPS URL for later use.  
It should look something like: `https://github.com/YourGithubUsername/PcmHacks.git`  
DO NOT **Open with Visual Studio** from here!

Clone PcmHacks from within Visual Studio,

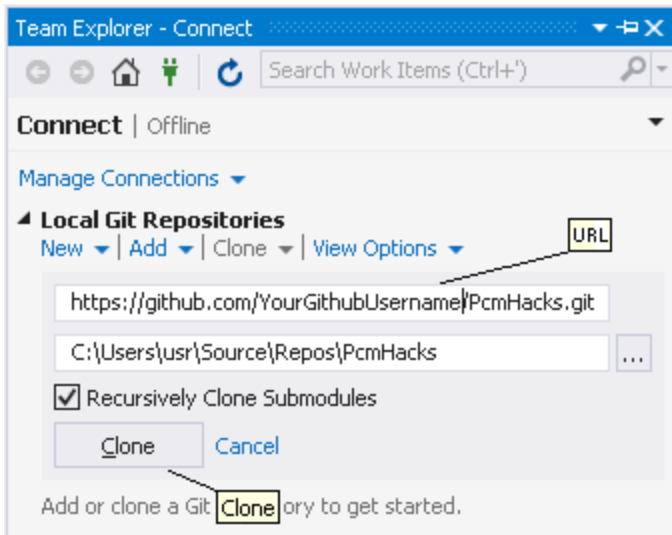
- Open: Visual Studio
- Select: View
- Select: Team Explorer

TIP: Pin Team Explorer Open for the duration of this document.



If not on 'Team Explorer - Connect' page,

- Select: Manage Connections
- Select: Clone (the Link)



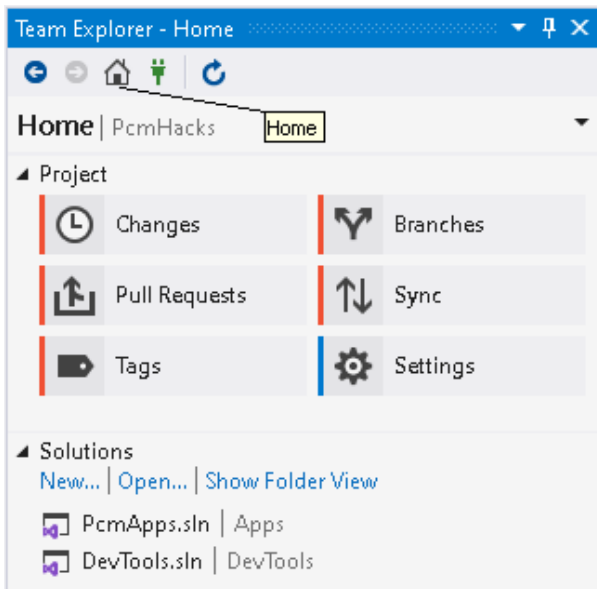
Enter the Clone "HTTPS" URL you made note of previously.

- Select: Clone (the Button)

Wait for it ... Don't rush it!

If Solution Explorer opens, close / hide it.

The PcmHacks repository has two solutions; it doesn't know which to open so it leaves it up to the user.



To open the PcmApps Solution,

- Open: Team Explorer
- Select: Home

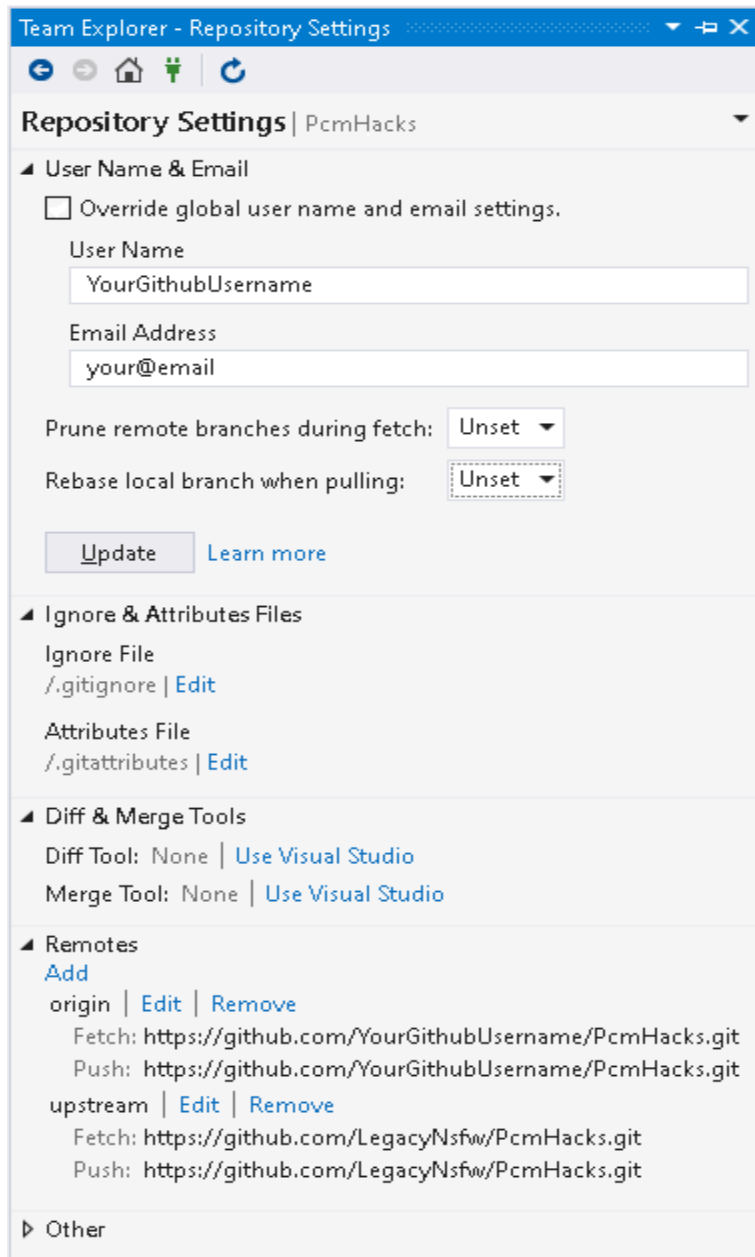
Under Solutions,

- Double Click: PcmApps.sln

Wait for it ... Don't rush it!

# Let's Git Configuration done

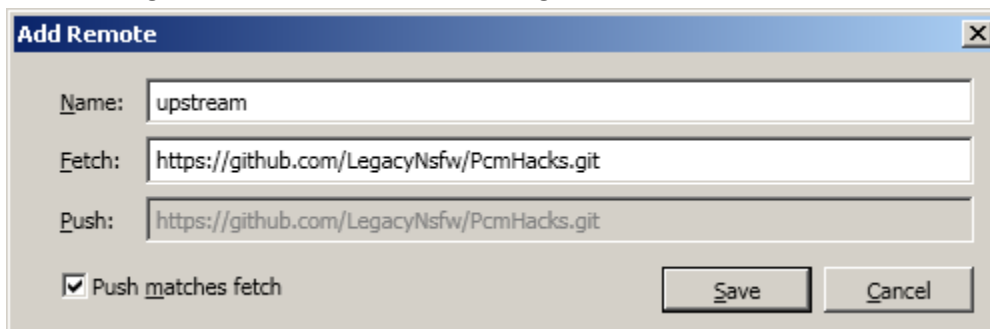
Create the upstream link to: <https://github.com/LegacyNsfw/PcmHacks.git>,



Open Team Explorer

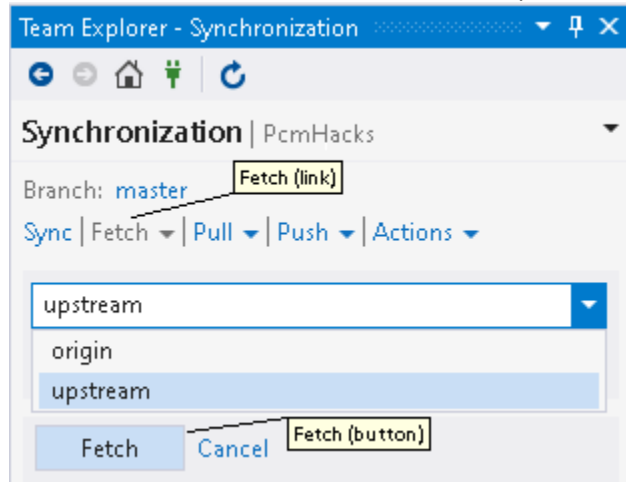
- Select: Home
- Select: Settings
- Select: Repository Settings
- Expand: Remotes
- Select: Add

Set the following values In the Add Remote dialog, and save.



The premise is to 'Push' to origin (Your Fork), and 'Fetch' from upstream (<https://github.com/LegacyNsfw/PcmHacks.git>)

Because Visual Studio does not show the upstream branches at first, let's fetch them,



Open Team Explorer

- Select: Home
- Select: Sync
- Select: Fetch (link)
- Select: upstream
- Select: Fetch (button)

That's it; you're all set up... You now have a local clone, of your fork, of the LegacyNsfw/PcmHacks repository.

Yup, that makes 3 copies!

1. The LegacyNsfw/PcmHacks repository, referred to as '**upstream**'
2. Your fork (on GitHub), referred to as '**origin**'
3. Your local repository, referred to as local or '**clone**'.

# So, you want to create a feature

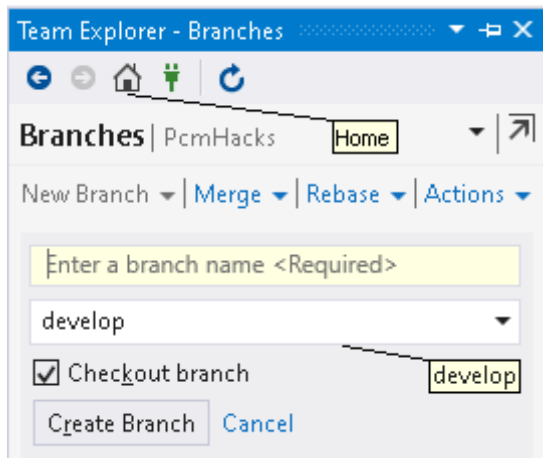
The basic premise is,

- NEVER make changes to the develop branch.
- Always create your own branch, based on develop.
  - Verify your changes. (Compare with Unmodified) (diff)
  - Stage the verified changes.
  - Commit the staged changes.
  - Push your YourFeature branch to your fork. (origin)
  - Login to GitHub and create a pull request.

When it gets merged (approved) ...

- Update your repositories.
  - Fetch upstream develop.
  - Merge upstream develop into develop. (Update your local repository)
  - Push develop to origin. (Update your fork)
  - Delete YourFeature branches. (On both, local and origin)

Create **YourFeature** branch,



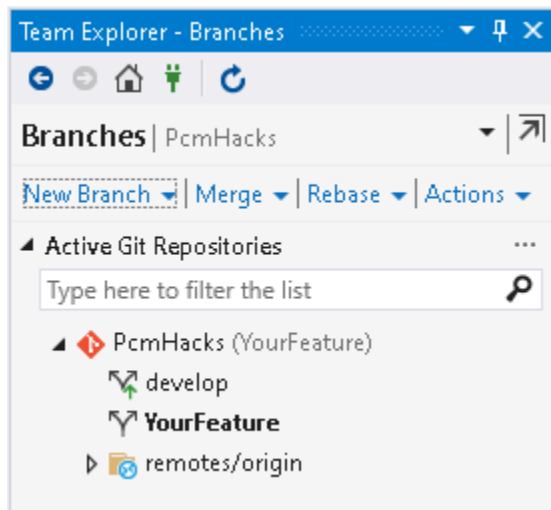
Open Team Explorer

- Select: Home
- Select: Branches
- Select: New Branch

Enter a branch name for YourFeature.

- Select: develop
- Check: Checkout branch
- Select: Create Branch

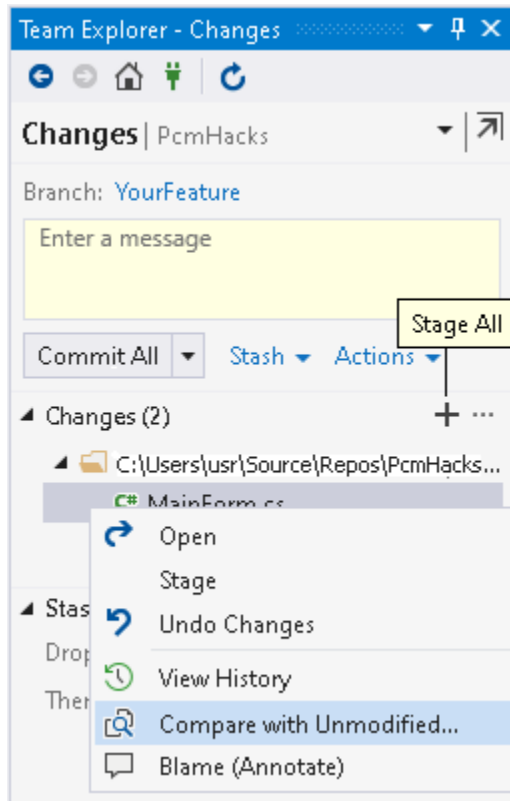
**YourFeature** branch is now the current branch.



NOTE: YourFeature branch is in bold type, which signifies it's the Current Branch.

Now create / edit your feature, when you are done, continue with the rest of this document.

Verify your changes with diff to ensure you stage only what you want,



Open Team Explorer

- Select: Home
- Select: Changes

If you made changes, you should have changes in the Changes list.

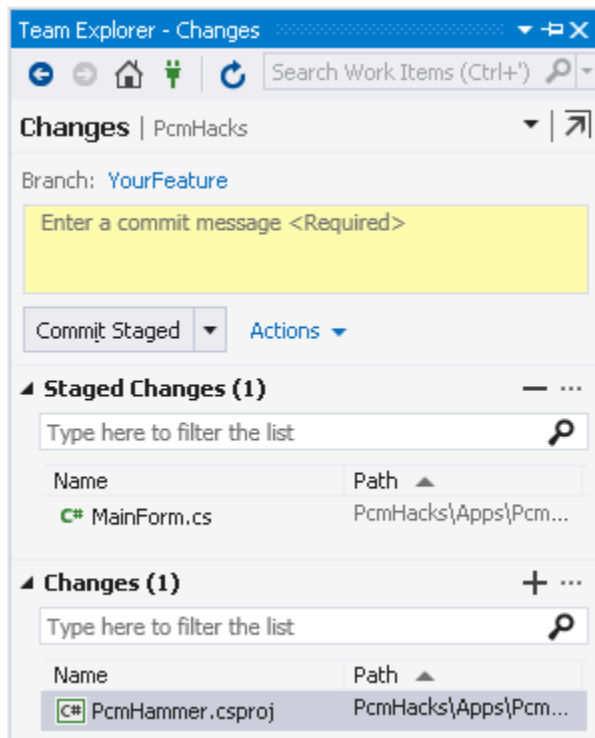
- Right-Click: A changed file
- Select: Compare with Unmodified...

Once happy with your changes, stage them,

- Right-Click: 'Each verified file'
- Select: Stage

I DO NOT recommend using: Stage All

Once you have staged what you want,

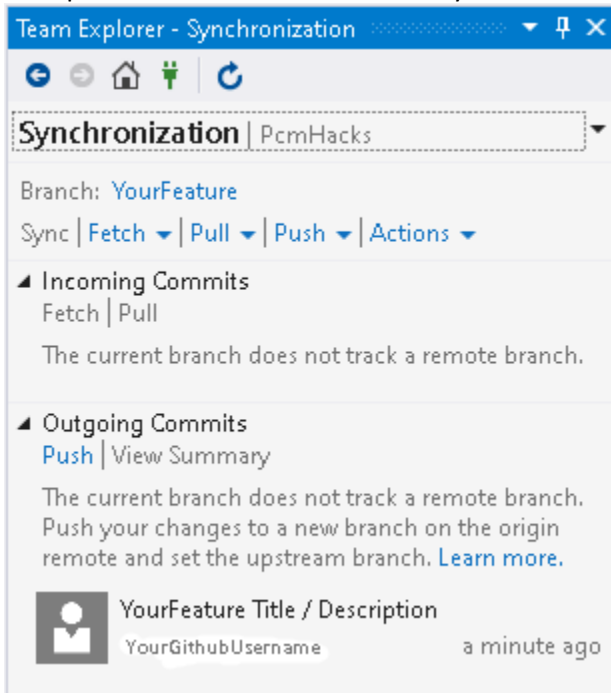


Enter a commit message. (A title / short description of your feature)

- Select: Commit Staged

That commits your changes to the local branch **YourFeature**.

Now push the YourFeature branch to your fork on GitHub (origin),



Open Team Explorer

- Select: Home
- Select: Sync

Under outgoing Commits,

- Select: push

Open your favorite internet browser and login to GitHub.

Navigate to your fork of the PcmHacks repository (**origin**),

- Select: Compare & pull request
- Add edit comments if so desire,
- Select: Create pull request

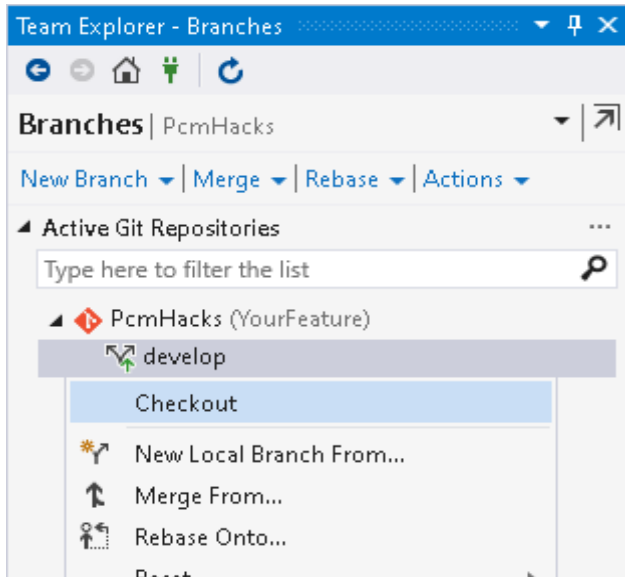
At this point if you find you have forgotten something or worse ... AND YourFeature Pull Request is still open, you can refollow the above “So you want to create a feature” steps with the following differences,

- Checkout YourFeature branch do not create a new one.
  - You do not need to do a Pull Request.
- It will automatically associate it with YourFeature Pull Request when you Push YourFeature branch.

When **YourFeature** gets approved and merged, or whenever the PcmHacks (**upstream**) repository is updated.

Update your repositories, first the local, then your fork.

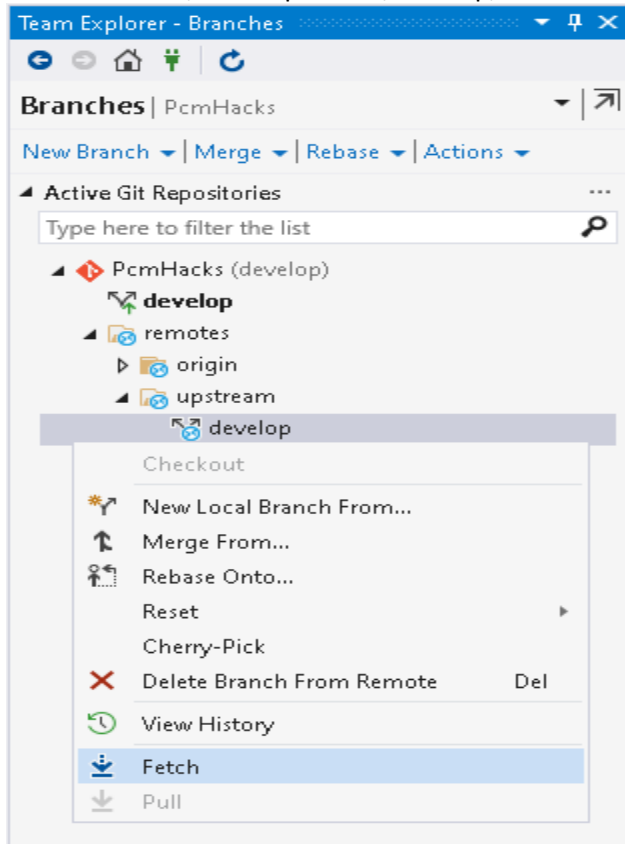
Open the PcmHacks Solution,



Open Team Explorer

- Select: Home
- Select: Branches
- Right-Click: develop
- Select: checkout

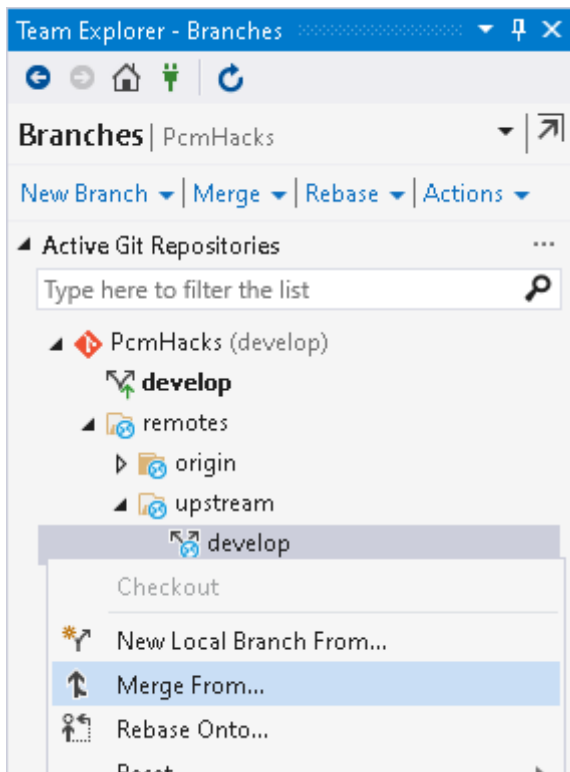
When it's done, fetch upstream/develop,



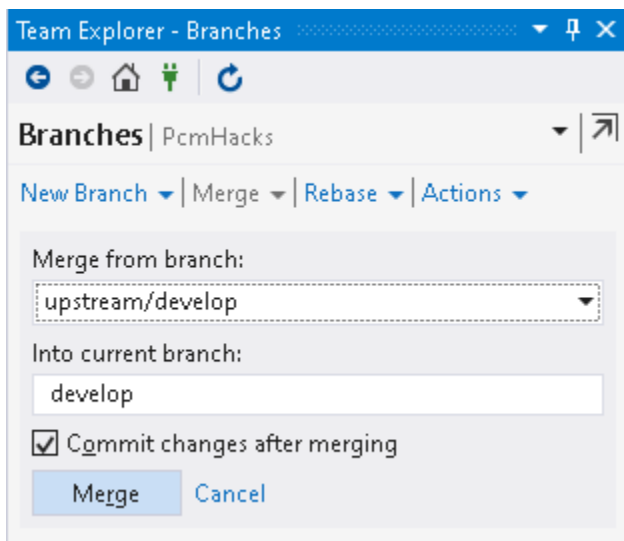
- Expand: remotes
- Expand: upstream
- Right-Click: develop
- Select: Fetch



When it's done, merge it into the local clone,



- Right-Click: develop
- Select: Merge From...



- Merge from branch: **upstream/develop**
- Into current branch: **develop**
- Check: Commit changes after merging
- Select: Merge

Answer yes, if asked.

That updates your local repository.

Update your fork (origin), open Team Explorer

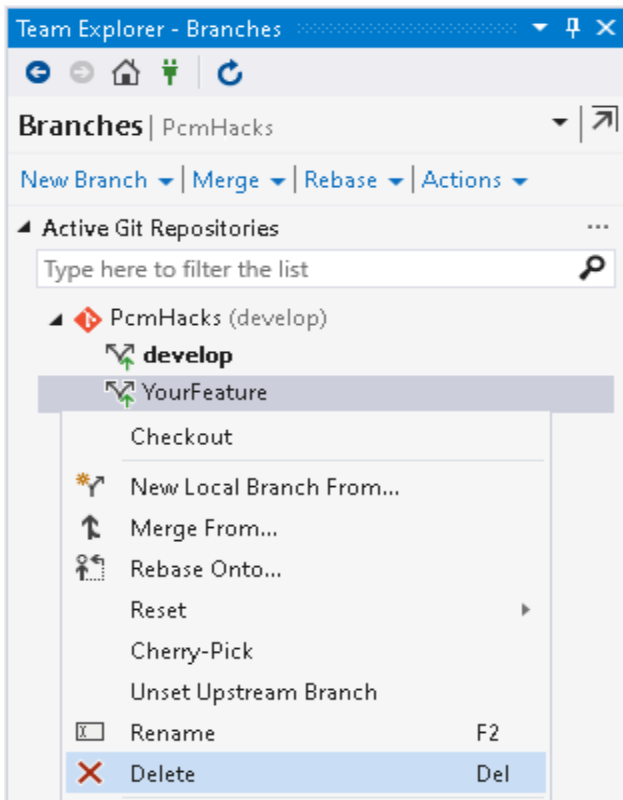
- Select: Home
- Select: Sync

Under Outgoing Commits,

- Select: Push

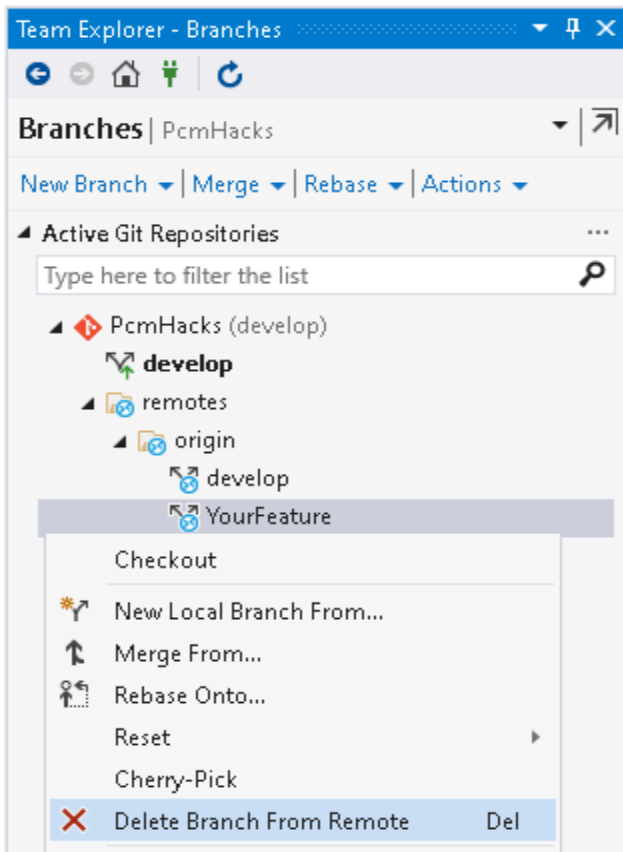
That updates your fork.

Cleanup the branches, delete **YourFeature** branch,



Open Team Explorer

- Select: Home
- Select: Branches
- First from local,
- Right-Click: **YourFeature**
- Select: Delete



And from your fork (origin)

- Expand: Remotes
- Expand: origin
- Right-Click: YourFeature
- Select: Delete Branch From Remote

Answer Yes if asked.

Now, all 3 repositories are equal.