Pull Changes

When you want to retrieve changes from the server, you will have to **pull** those changes down. You can pull changes from your personal GitHub repository. If you forked another repository, you can also pull changes from the original repository (called the *upstream* repository).

IMPORTANT NOTE

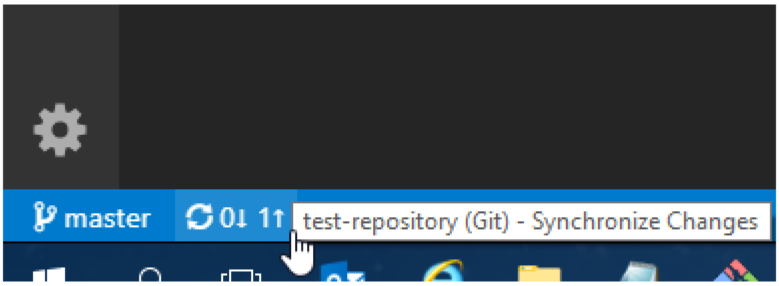
When working with a forked repository, after pulling from the original upstream repository, you should **push** the changes up to your fork. This ensures that the upstream changes are also reflected in your personal server repository,

Pulling changes in Visual Studio Code

Pulling from your server repository

Visual Studio Code has an interesting tool for pulling changes down from the remote repository. Instead of simply pulling changes, it actually *synchronizes* the local and remote repositories. This means that it pushes any commit changes up, and also pulls down any changes from the server that may exist.

To pull your changes down, use the **Synchronize Changes** button on the blue bar at the bottom of the window:

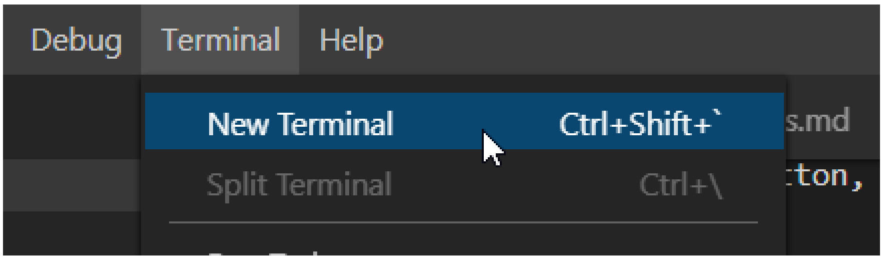


Once you click that button, your server changes should exist in your local clone! You can go to your repository on GitHub and make sure it matches your local repository.

Pulling from the original upstream repository

The easiest way to pull from the upstream repository is via the command line.

To open a command prompt within Visual Studio Code, select "Terminal" in the file menu, and the click "New Terminal":



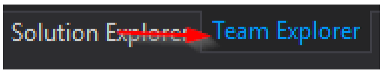
That should open a Git Bash terminal in the proper folder. Then, follow the instructions for pulling from the original upstream repository via the command line.

Pulling changes in Visual Studio

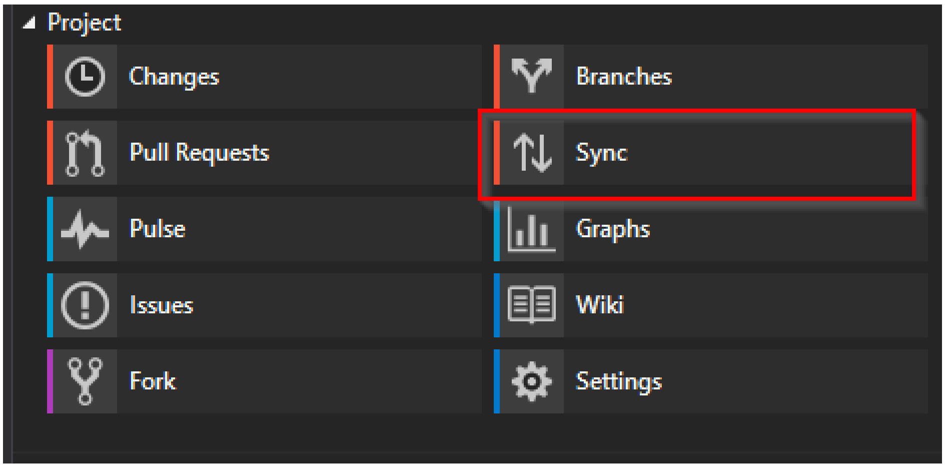
Pulling from your server repository

Visual Studio has some nice built-in tools for pulling changes from the remote repository.

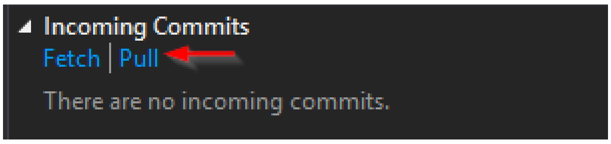
1. Open the **Team Explorer** pane



2. Under "Project", click **Sync**



3. Under "Incoming Commits", click **Pull**



There are no incoming commits.

   - *Note: To simply view incoming commits, but not merge them, you can click **Fetch***

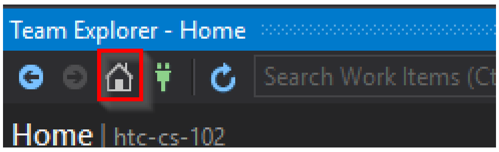
Once you pull, your server changes should exist in your local clone! You can go to your repository on GitHub and make sure it matches your local repository.

Pulling from the original upstream repository

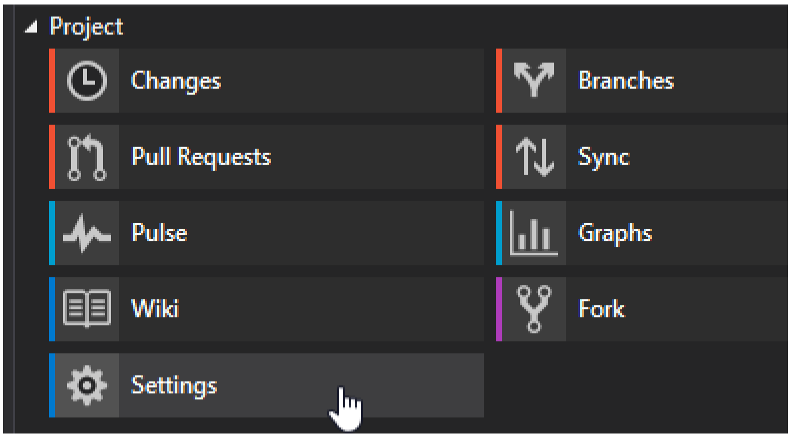
Adding the upstream remote

To keep your fork up-to-date with the latest changes from the original repository, you will need to add an **upstream remote**. This will allow you to pull when new material is added to the original repository.

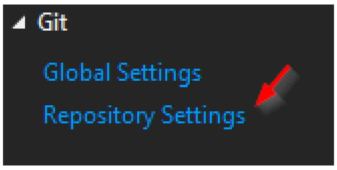
1. In the **Team Explorer** pane, click the "Home" icon to go home



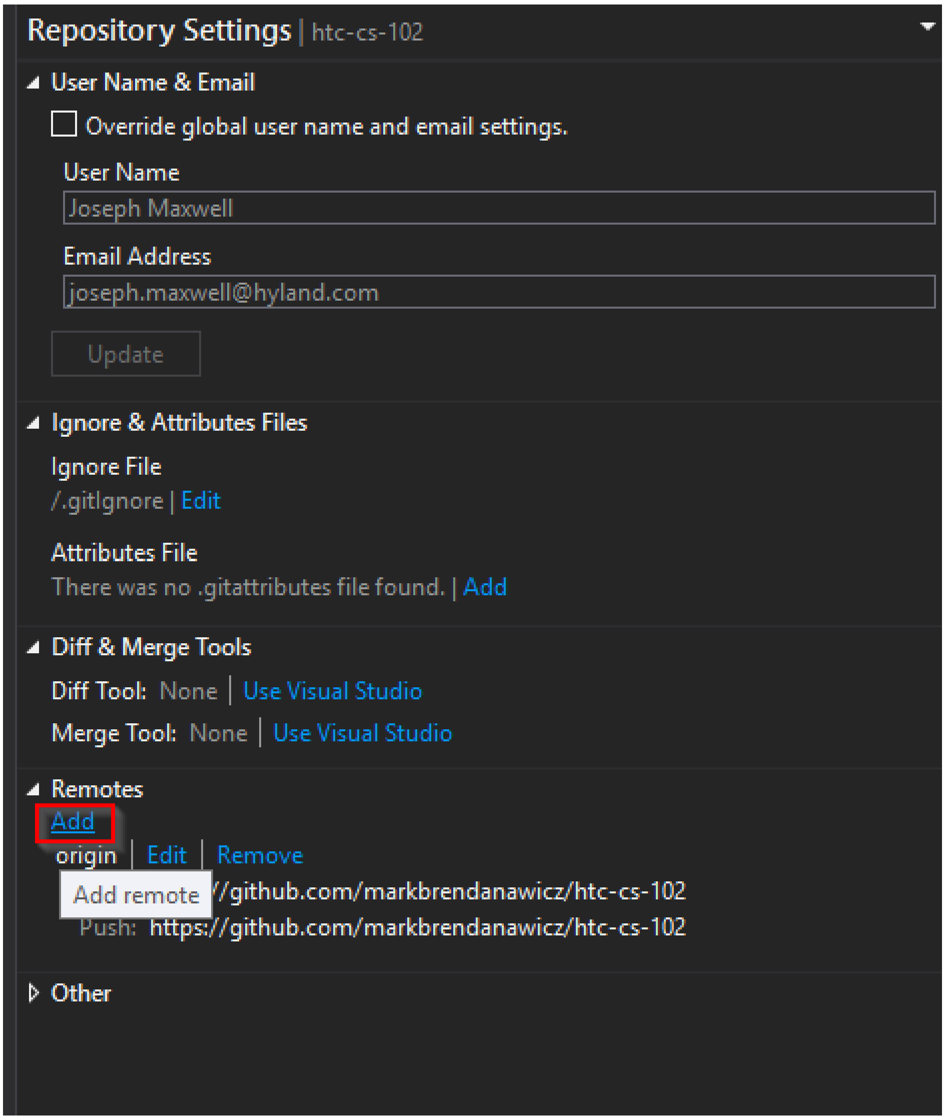
2. Under "Project", click **Settings**



3. Under "Git", click **Repository Settings**



4. In Repository Settings, under "Remotes", click **Add**



Repository Settings | htc-cs-102

User Name & Email

□ Override global user name and email settings.

User Name

Joseph Maxwell

Email Address

joseph.maxwell@hyland.com

Update

Ignore & Attributes Files

Ignore File

/.gitIgnore | Edit

Attributes File

There was no .gitattributes file found. | Add

Diff & Merge Tools

Diff Tool: None | Use Visual Studio

Merge Tool: None | Use Visual Studio

Remotes

Add

origin | Edit | Remove

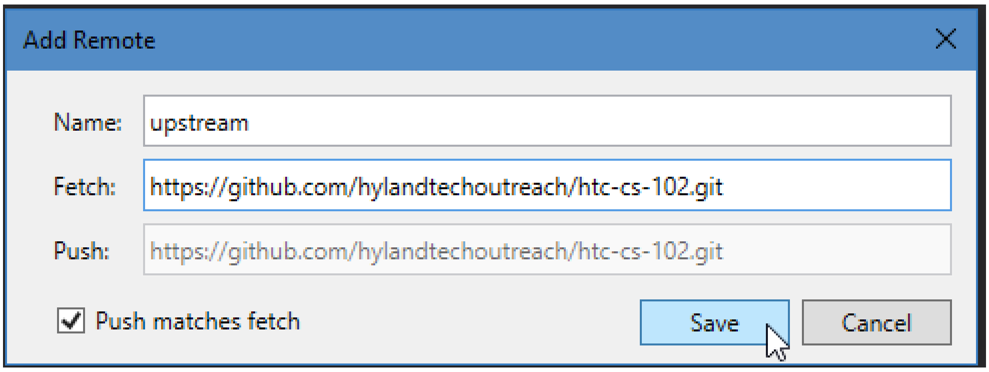
Add remote //github.com/markbrendanawicz/htc-cs-102

Push: https://github.com/markbrendanawicz/htc-cs-102

Other

5. In the "Add Remote" dialog, enter "upstream" for *Name*
6. Enter the URL for the upstream repository for *Fetch*

7. Click **Save**



Add Remote

Name: upstream

Fetch: https://github.com/hylandtechoutreach/htc-cs-102.git

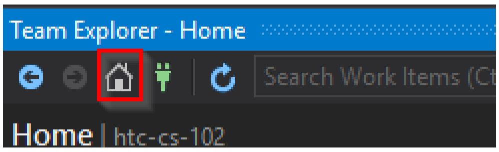
Push: https://github.com/hylandtechoutreach/htc-cs-102.git

☑ Push matches fetch

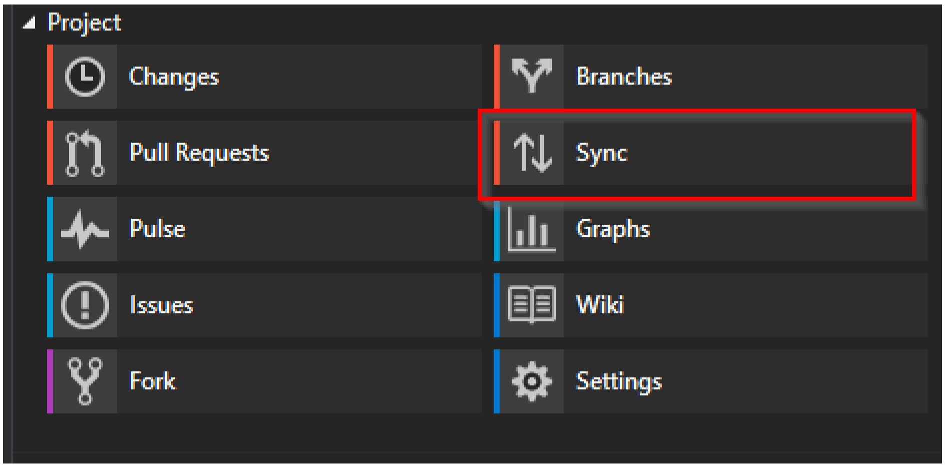
Now your **upstream remote** is all set up! In order to receive the latest changes from the original repository, you will have to **fetch** and **merge** from the upstream.

Fetching from the upstream remote

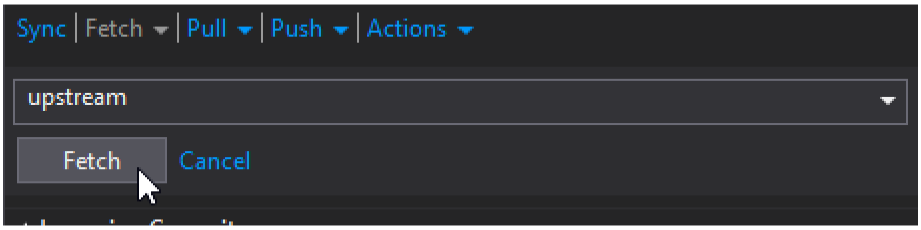
1. In the **Team Explorer** pane, click the "Home" icon to go home



2. Under "Project", click **Sync**

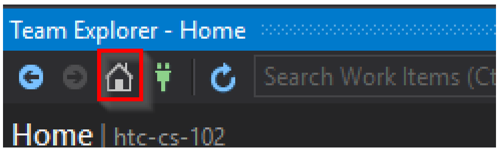


3. In Synchronization, at the top of the pane, click **Fetch**
4. Click the dropdown and select "upstream"
5. Click **Fetch**

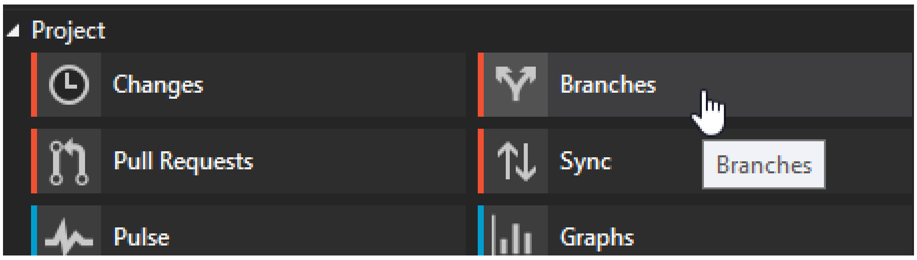


Merging from the upstream remote

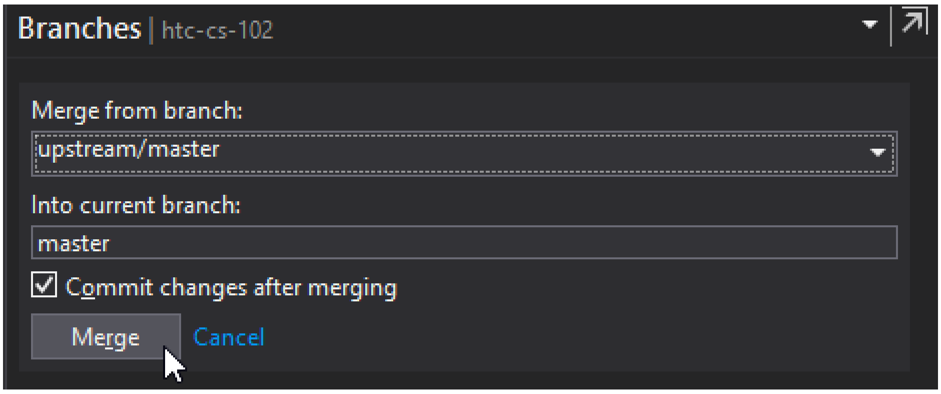
1. In the **Team Explorer** pane, click the "Home" icon to go home



2. Under "Project", click **Branches**



3. In Branches, at the top of the pane, click **Merge**
4. In the "Merge from branch" dropdown, select "upstream/master"
5. Click **Merge**



Merge from branch: upstream/master

Into current branch: master

☑ Commit changes after merging

Now you have the latest changes from the original upstream repository!

Pulling changes via the Command Line

Pulling from your server repository

If you're using the command line, pulling from your server repository is simple:

1. Change your directory to the folder that contains your repository (replace `/path/to/repository/` with your path)

```
cd /path/to/repository/
```

2. Run the `pull` command

```
git pull
```

That's it! Now your local clone repository should have all the latest changes from your GitHub repository.

Pulling from the original upstream repository

Pulling changes from the upstream repository is also fairly simple, assuming you do not have any conficting changes in your local repository.

If you are in the proper directory, all you have to do is run `git pull` with the URL of the upstream repository. Replace `UPSTREAMREPOSITORYURL` with the url of the original repository:

```
git pull https://github.com/hylandtechoutreach/UPSTREAMREPOSITORYURL.git
```

Now your local branch should contain all the latest changes from the upstream repository!