Back Ground

We use version control to save work and coordinate code changes across team. Version control keeps a history of code development so that we can review and even rollback to any version of code with ease.

We have two types of version control systems. They are Central version control and Distributed version control. Git is a distributed version control and widely used in many projects. Entire Git work flow can be handled by below mediums.

* Using web portal
* Using command prompt
* Using visual studio

Note : Handling Git work flow using visual studio is much easier and developer friendly

Prerequisites

* Distributed version control basics
* Git Basics
* Visual Studio basics

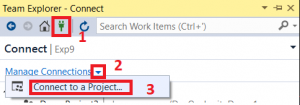
Note : Visual studio already have features to handle work flow for Git and VSTS projects. This blog covers basic features which will help to developers to kick start using git with visual studio

GIT Work Flow

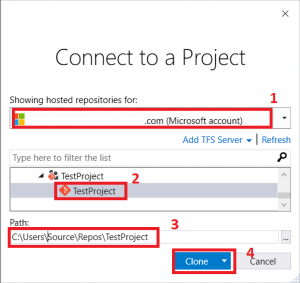
* Clone: To get copy of source code from remote git repository to local git repository
* Create Branch : Create branch for the changes of planned work (Feature/Bug fix)
* Commit changes : To move the code changes to local git repository
* Push Changes : Move the changes (all your commits )to remote git repository
* Pull changes : Get the all commits (all your team commits) from remote git repository

1. Clone:

* + In ***Team Explorer***, open the Connect page by selecting the Connect icon, and then choose Manage Connections, ***Connect to Project***.

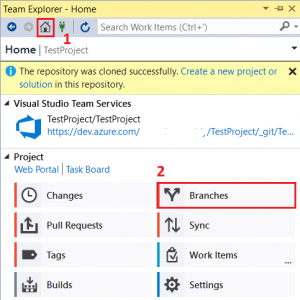


* + On the ***Connect to a Project dialog***, select the repo you want to clone from the list.
  + Verify the location of the cloned repo on your PC and select ***Clone***.

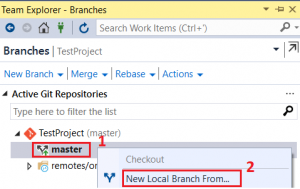


2. Create Branch:

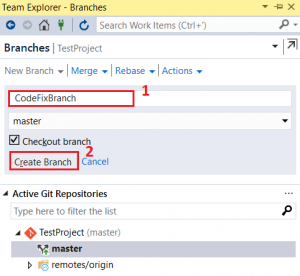
* + - In Team Explorer, select the Home icon and choose Branches



* + - Right click the master branch and choose ***New Local Branch From***

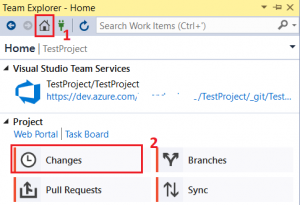


* + - Choose a descriptive branch name for your work and select ***Create Branch***

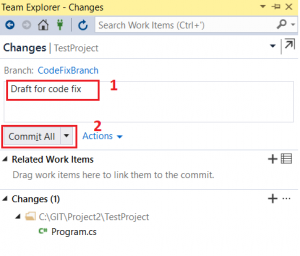


3. Commit Changes:

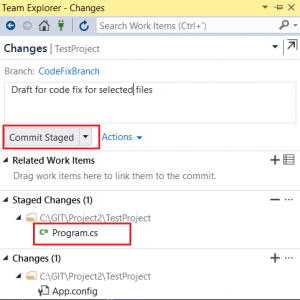
* + You make commits in your local Git repository to save your changes on that branch  
    Make changes to your files in the cloned repo.
  + From the Team Explorer Home view, you can open up Visual Studio solutions in the repo. Git keeps track of changes made to your code both inside and outside of Visual Studio.
  + When you are done with the your changes, save them in Git using a ***commit***.
  + Open up the Changes view from Team Explorer by selecting the Home icon and choose ***Changes***.



* + Enter a message describing the commit and select ***Commit All***.

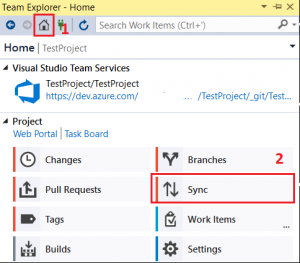


* + If you have multiple files and you don’t want to commit them all, you can right-click each desired file and choose ***Stage***. When you have staged all the files you would like to commit, click ***Commit Staged***.

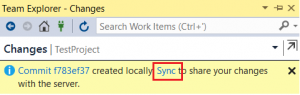


4. Push Changes :

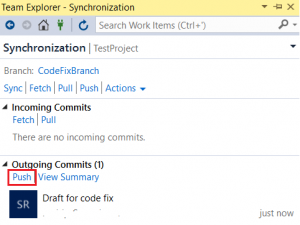
* + When you are ready to share your changes with the team, you can push those changes (all your commits) so that others can reach them. You can only push changes after you add commits to a branch.
  + Open up the Synchronization view in Team Explorer by selecting the Home icon and choosing ***Sync***.



* + You can also navigate to the Synchronization view from the Changes view by choosing Sync immediately after making a commit.

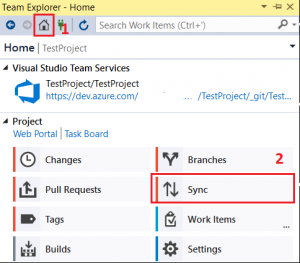


* + Select ***Push***to share your commit with the remote repository.



5. Pull Changes :

* + To get changes from others and keep your code up to date, you can pull commits made by others and merge them into your branch. Git is very good about merging multiple changes even in the same file, but sometimes you might have to resolve merge conflict. It’s a good idea to pull your branches regularly to keep them up to date with the changes from others. This makes sure that your feature branches from your main branch are using the latest version of the code.
  + Open up the Synchronization view in Team Explorer by selecting the Home icon and choosing ***Sync***.



* + You can download the latest changes to the branch you are on using the ***Pull***link. (There are two Pull links, one near the top and one in the Incoming Commits section. You can use either one as they both do the same thing).

