What is Git?

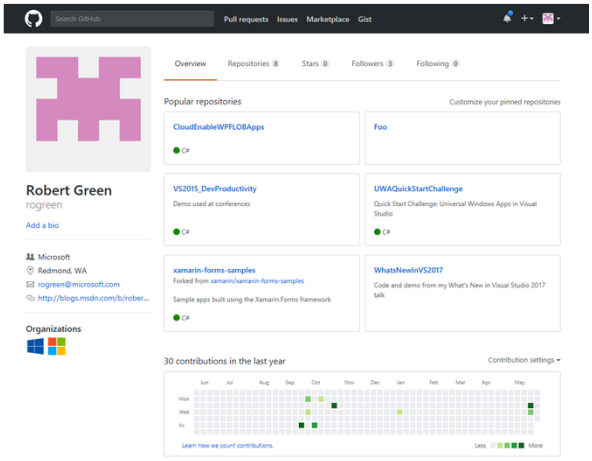
- Distributed version control system
  - Your local copy of code is a complete version control repository
  - Commit your work locally, and then sync your copy of the repository with the copy on the server
  - Differs from centralized version control where clients must synchronize code with a server before creating new versions of code
- Created in 2006 to manage the Linux kernel
- Most commonly used version control system today
  - Quickly becoming the standard for version control

Benefits of Git

- Everyone has their own local copy of code
  - Work on your own branch
  - Work offline
  - Faster since you don't have to communicate with the server
- Everyone has a backup of the entire project
- Everyone maintains the full history of the project
- Everyone can experiment with new additions before committing to master

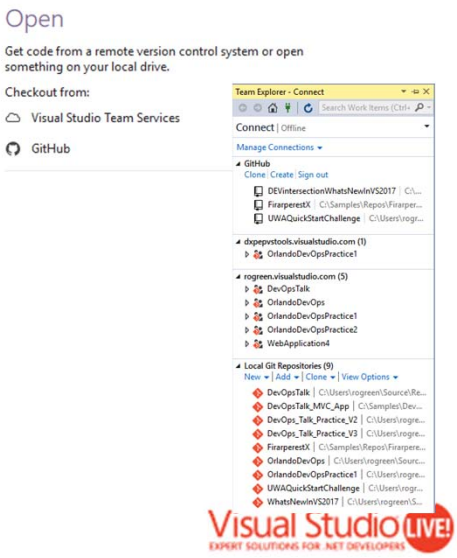
What is GitHub?

- GitHub is a version control repository stored on the Web
- Use it to share code with others and find/contribute to code written by others



Git vs GitHub vs VSTS

- Git is the plumbing
- GitHub is a Web based repo
- VSTS is that plus project management / DevOps tools
- Visual Studio has great support for both VSTS and GitHub
- You can use Git or TFVC with a VSTS project



Git Started

- Create a new repo
- Clone an existing repo
- Save work with commits
- Share code with push
- Update code with fetch and pull
- Create work in branches
- Create a pull request
- Resolve merge conflicts

Git repository (aka repo)

- A folder that you've told Git to help you track file changes in.
- A Git repo contains every version of every file saved in the repo.
- You can have any number of repos on your computer, each stored in their own folder.

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Create a repo

- Two ways to create a repo
  - Create a repo from an folder on your computer and then associate it with a shared repo
  - Clone an existing repo, which creates a new repo in a new folder on your computer and configures it for sharing

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Save work with commits

- Git does not automatically snapshot your code as you make edits to files in your repo.
- You must tell Git exactly which changes you want to add to the next snapshot by staging those changes.
- After staging your changes, create a commit to save the snapshot to your repo.

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Create a new repo
Save work with commits

DEMO

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Share code with push

- Share changes made in commits and branches using the push command.
- Push your branches to the remote repository.
  - Git takes the commits and adds them to an existing branch on the remote or creates a new branch with the same commits as your local branch.
- If there are conflicts between your local commits on the commits on the remote branch, you must first resolve these conflicts before you can push your changes.
  - You should pull the changes from others first, resolve the conflicts and commit the changes, then re-attempt the push.

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Share code with push

Clone an existing repo

DEMO

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Update code with fetch and pull

- Fetch downloads the changes from your remote repo but does not apply them to your code.
- Merge applies changes taken from fetch to a branch on your local repo.
- Pull does a fetch and then a merge.
- If there is a merge conflict between a commit you haven't pushed yet and a commit you are merging or pulling, you'll need to resolve those conflicts before you finish updating your code.

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Update code with fetch and pull

Use Visual Studio and GitHub

DEMO

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Create work in branches

- Create new branches to isolate changes for a feature or a bug fix from your master branch and other work. .
- Git does not create multiple copies of your source when working with branches—it uses the history information stored in commits to recreate the files on a branch when you start working on it.
- Isolating work in branches makes it very simple to change what you are working on by simply changing your current branch.

Create a pull request

- The pull request is the collaborative process that lets the rest of the team discuss changes in a branch and agree to merge them once everyone approves.
- Use pull requests to get early feedback from others on work in progress, even if you're not ready to merge the changes into another branch.
- You must resolve merge conflicts before you can commit a pull request.

Resolve merge conflicts

- When you merge one branch into another, file changes from commits in one branch can conflict with the changes the other.
- Git attempts to resolve these changes by using the history in your repo to determine what the merged files should look like.
- When it isn't clear how to merge changes, it halts the merge and tells you which files conflict.

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Create work in branches
Create a pull request
Resolve merge conflicts

DEMO

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Add existing code to GitHub
Use Visual Studio and VSTS

DEMO

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Resources

- Git and Team Services docs
  https://www.visualstudio.com/en-us/docs/git/overview
- Ry's Git Tutorial
  https://www.amazon.com/Rys-Git-Tutorial-Ryan-Hodson-ebook/dp/B00QFIA5OC
- Git Succinctly and GitHub Succintly
  https://www.syncfusion.com/resources/techportal/ebooks

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS