Pentalog

Git – Config & How-to

DATA ANALYSIS
UX/UI
FRONT-END
TECHNOLOGY
NEAR/OFFSHORE
AGILITY
BACK-END
SPRINT
KANBAN
CAMPAIGNS
GROWTH HACKING
SCRUM
BACKLOG
DEVOPS
DESIGN
SEO
CONTINUOUS INTEGRATION
MOBILE
QA
AUTOMATION
RESPONSIVE
UNIT TESTING

What is Git

- Git is a version control system created by Linus Torvalds in 2005
- It helps developers collaborate and work together on a project
- It has become the standard version control system used by software developers
- It can be difficult to learn, but you won’t regret learning it

How to use Git

- Git is mainly used from the command line. Because of this, it is more difficult to get started with it
- There are some user interfaces for git that makes it more user friendly
- Some IDEs also contain support for using Git, including Visual Studio
- In this course, we will concentrate on using Git from Visual Studio, and we will only use a subset of its functionalities

Git terminology

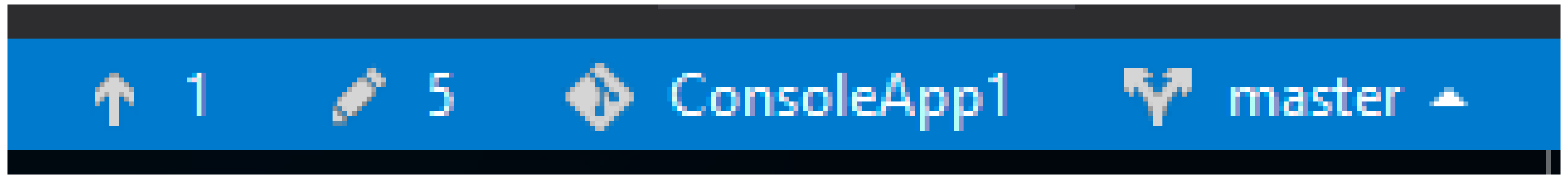
- Git is a distributed version control system. Each developer has his own copy of the source code, including all the history
- A project’s source code including the entire history is called a repository
- A developer can create a git repository in any folder on his computer
- A set of code changes to the code (adding, modifying or deleting files) is called a commit
- Each commit has a parent, and this chain of commits is the history of the project

Hosting a Git repository

- Although a Git repository can be created and used locally, it can also be hosted in a remote location
- There are several websites that offer hosting for Git repositories. In this course, we will be using GitHub
- A Git repository can have a remote location associated with it. This is called a remote
- After a developer commits some changes to a local repository, he can upload those changes to the remote location. This operation is called pushing
- A developer can also receive changes made from a remote repository. This operation is called pulling

Git integration

- Visual Studio can use the Git source control system
- To work with Git, we use the Team Explorer panel
- Shortcuts to different Git panels are displayed in the right side of the Visual Studio status bar

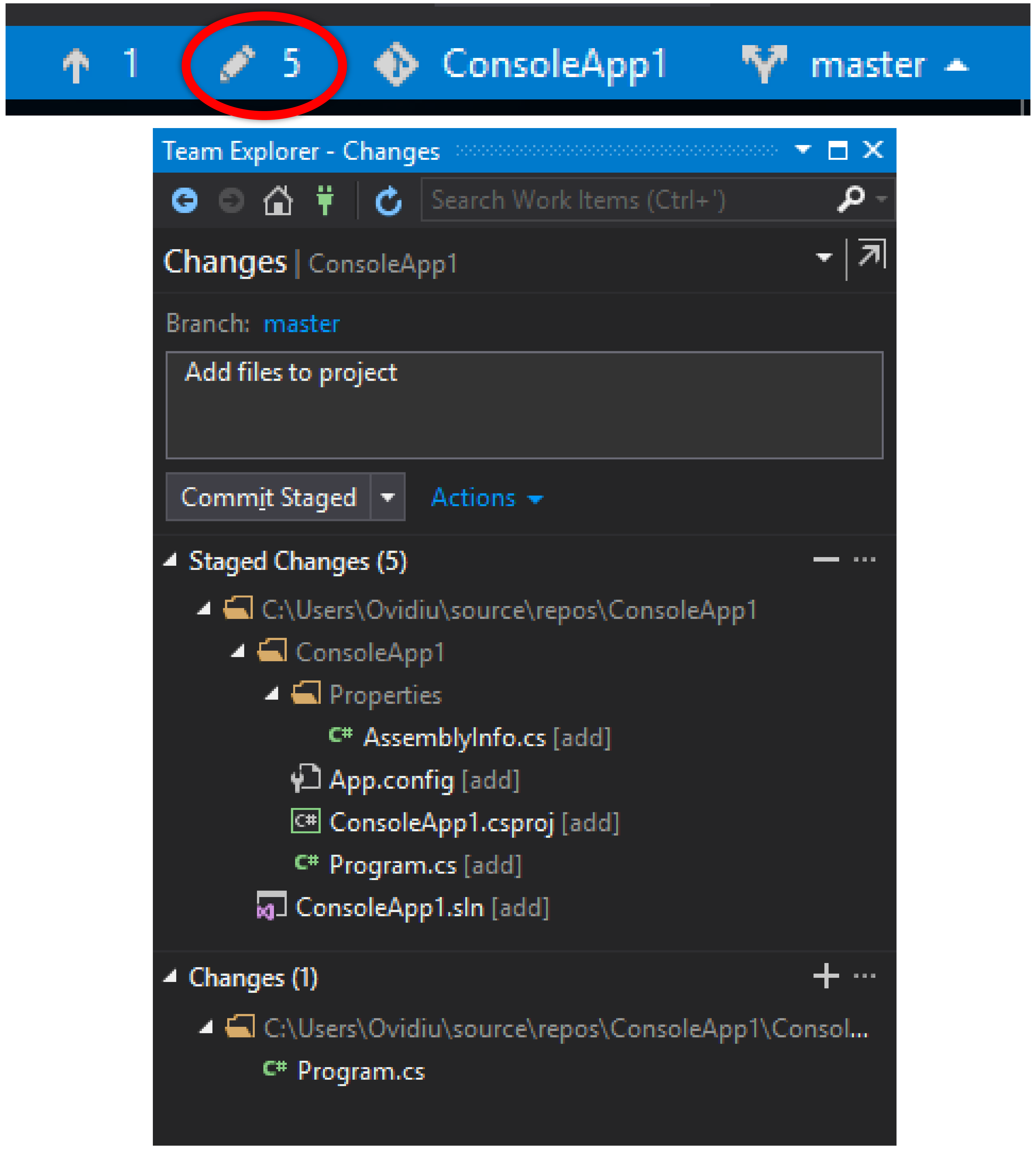


Initializing a repository for a project

- In order to use git for a project, we need to create a repository for it
- When we are creating the project in Visual Studio, we can use the “Create new Git repository checkbox”
- If we have created a project without checking that option, we can still initialize a repository from the “Add to Source Control” button in Visual Studio’s status bar
- Once the repository has been initialized, the project’s code history will be stored in the repository. We can commit changes to the repository and view its history.
- After the repository is initialized, we can also push it to a remote location, such as GitHub

Reviewing and committing changes

- We can see the changes we have made to the project files in the Changes section of Team Explorer
- The changes section allows us to view the files we changed, stage or unstage the changed files, enter a description for the commit we are making, and finally make the commit
- When committing changes, Git requires us to enter a message that describes the changes we made to the files



Branch: master

Add files to project

Staged Changes (5)

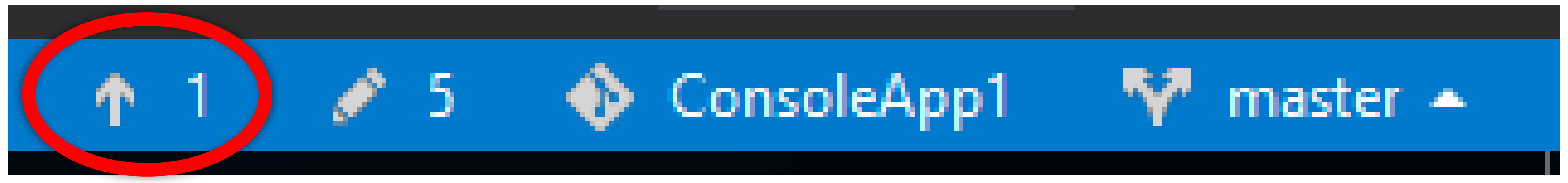
```
C:\Users\Ovidiu\source\repos\ConsoleApp1
  ConsoleApp1
    Properties
      AssemblyInfo.cs [add]
    App.config [add]
    ConsoleApp1.csproj [add]
    Program.cs [add]
  ConsoleApp1.sln [add]
```

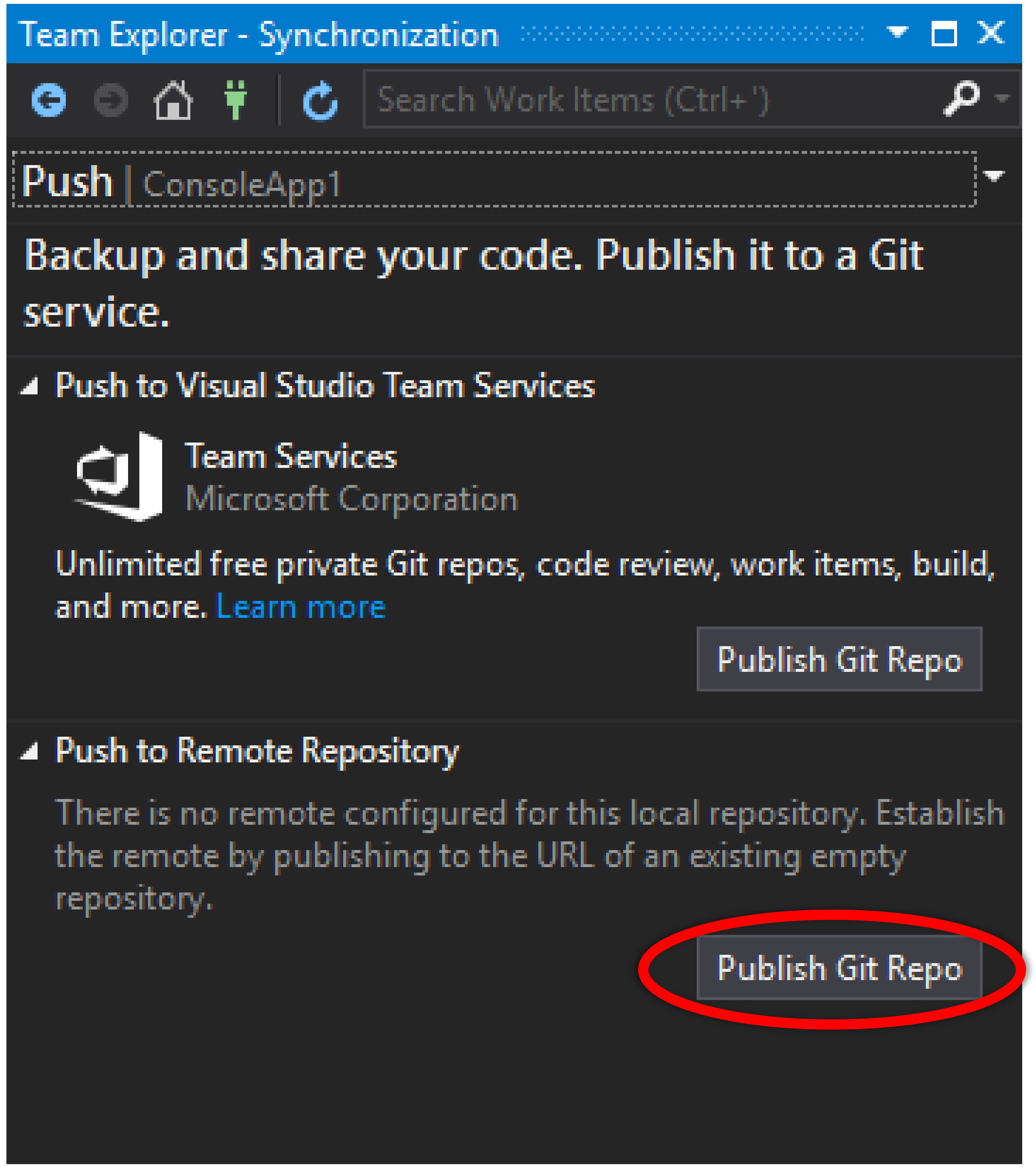
Changes (1)

```
C:\Users\Ovidiu\source\repos\ConsoleApp1\Consol...
  Program.cs
```

Publish or sync changes

- The Sync section allows us to publish our changes to a remote repository, and update our local repository with changes made remotely
- If we don’t have a remote repository set up, Visual Studio prompts us to setup a remote repository using a URL





Backup and share your code. Publish it to a Git service.

Push to Visual Studio Team Services

Team Services
Microsoft Corporation

Unlimited free private Git repos, code review, work items, build, and more. Learn more

Push to Remote Repository

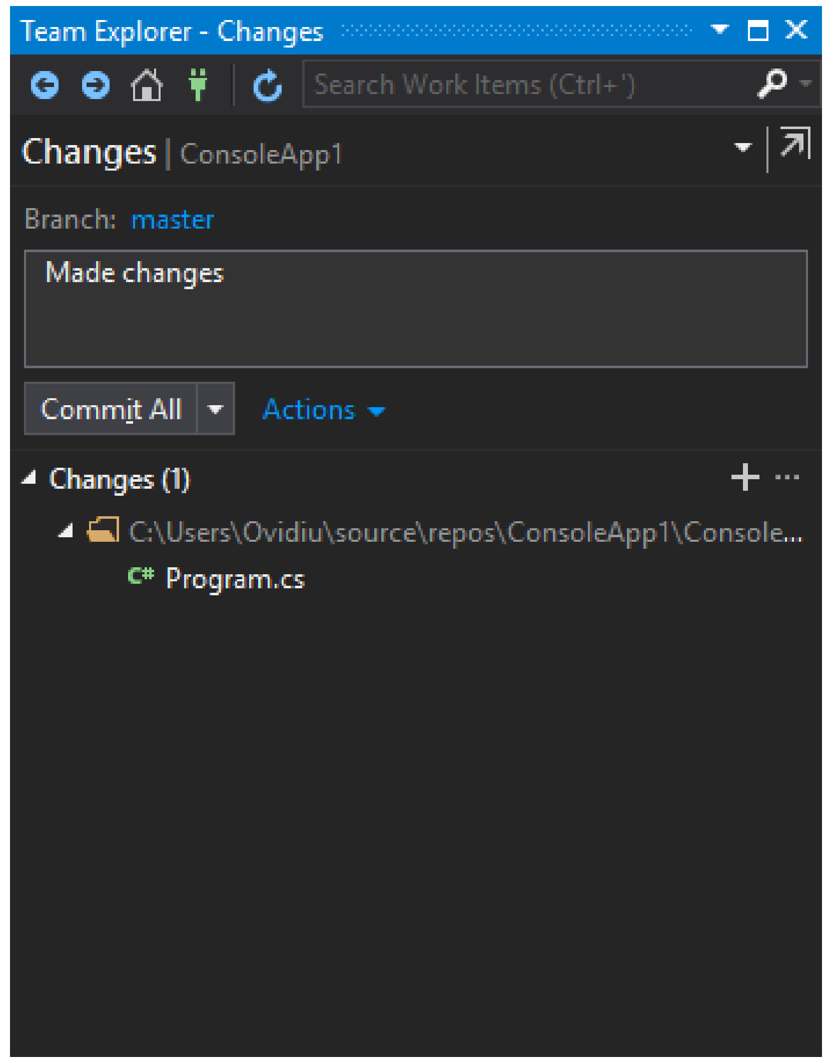
There is no remote configured for this local repository. Establish the remote by publishing to the URL of an existing empty repository.

Publish to GitHub

- In order to publish our project to GitHub, we must first create an empty repository: https://github.com/new
- The repository needs to be empty, so don’t add a readme file, a .gitignore file or a license file
- Once our repository is created, we can copy the URL, which looks like this: https://github.com/oviradoi/HelloWorld.git
- In Visual Studio, we must use the Publish Git Repo button and paste the GitHub URL in the box that appears
- After we login to GitHub, Visual Studio will publish the project for us

Project workflow

- When we make modifications to our project, we must first commit our changes, then push them to GitHub
- We can stage and commit the changes in the Changes section
- After that, we can push our changes to GitHub in the Synchronization section

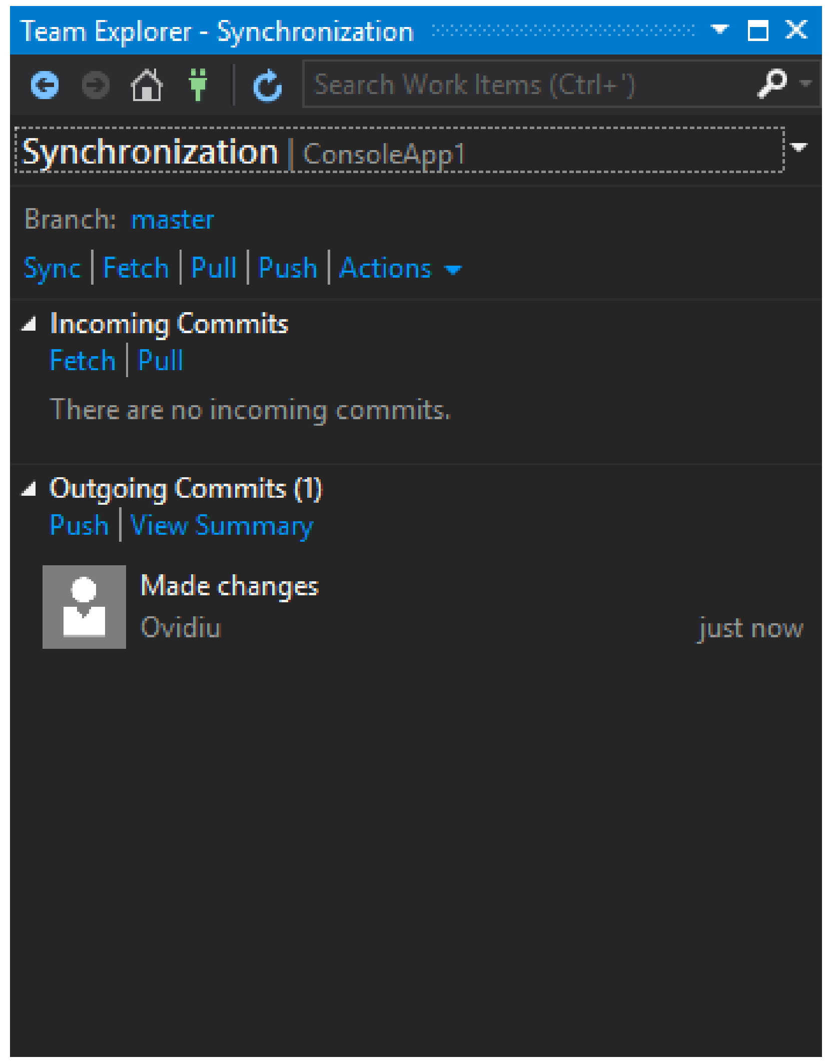


Branch: master

Made changes

C:\Users\Ovidiu\source\repos\ConsoleApp1\Console...

Program.cs


Team Explorer - Synchronization
Search Work Items (Ctrl+')
Synchronization | ConsoleApp1
Sync | Fetch | Pull | Push | Actions
Incoming Commits
Fetch | Pull
Outgoing Commits (1)
Push | View Summary

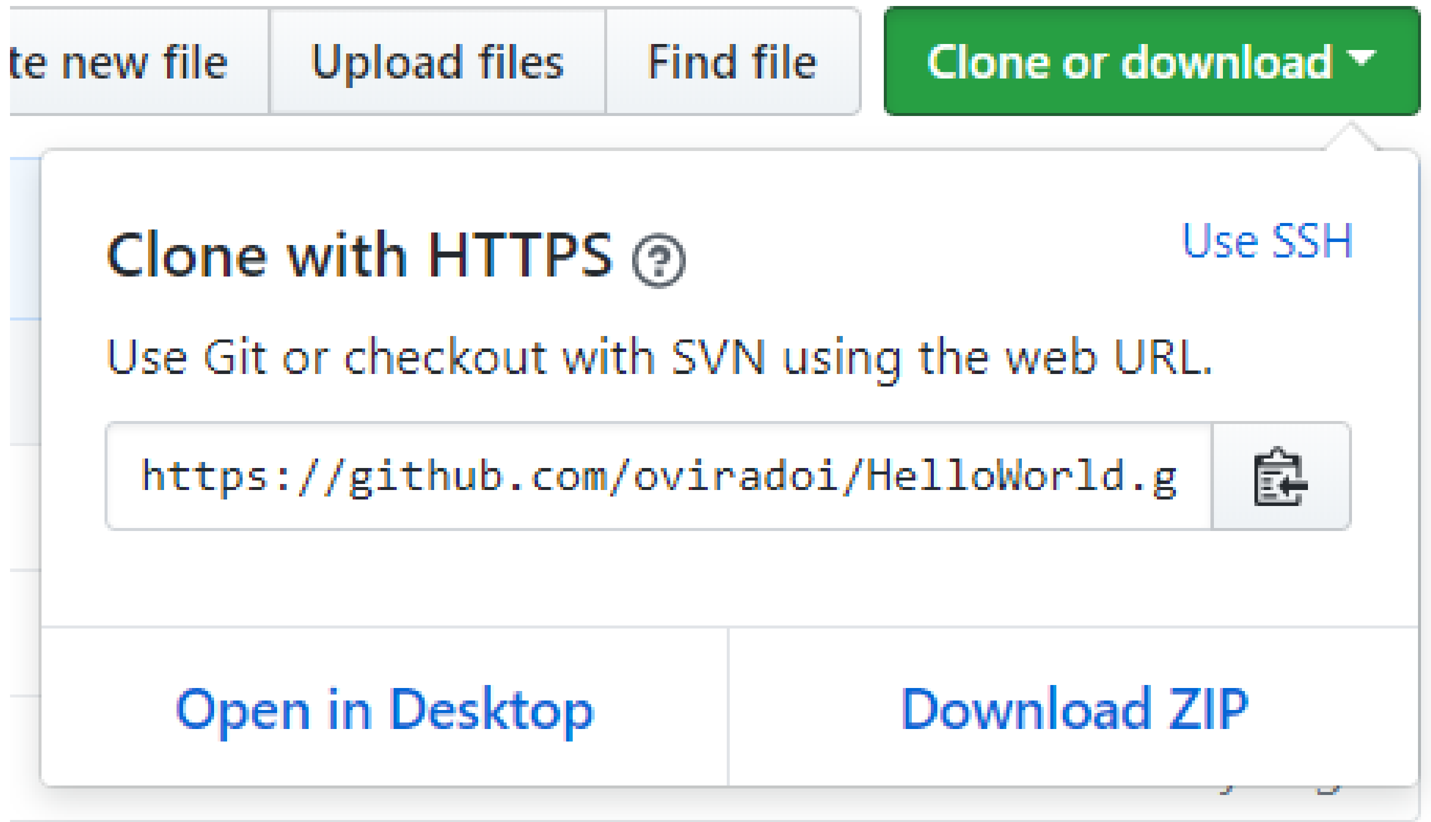
Branch: master

There are no incoming commits.

Made changes
Ovidiu just now

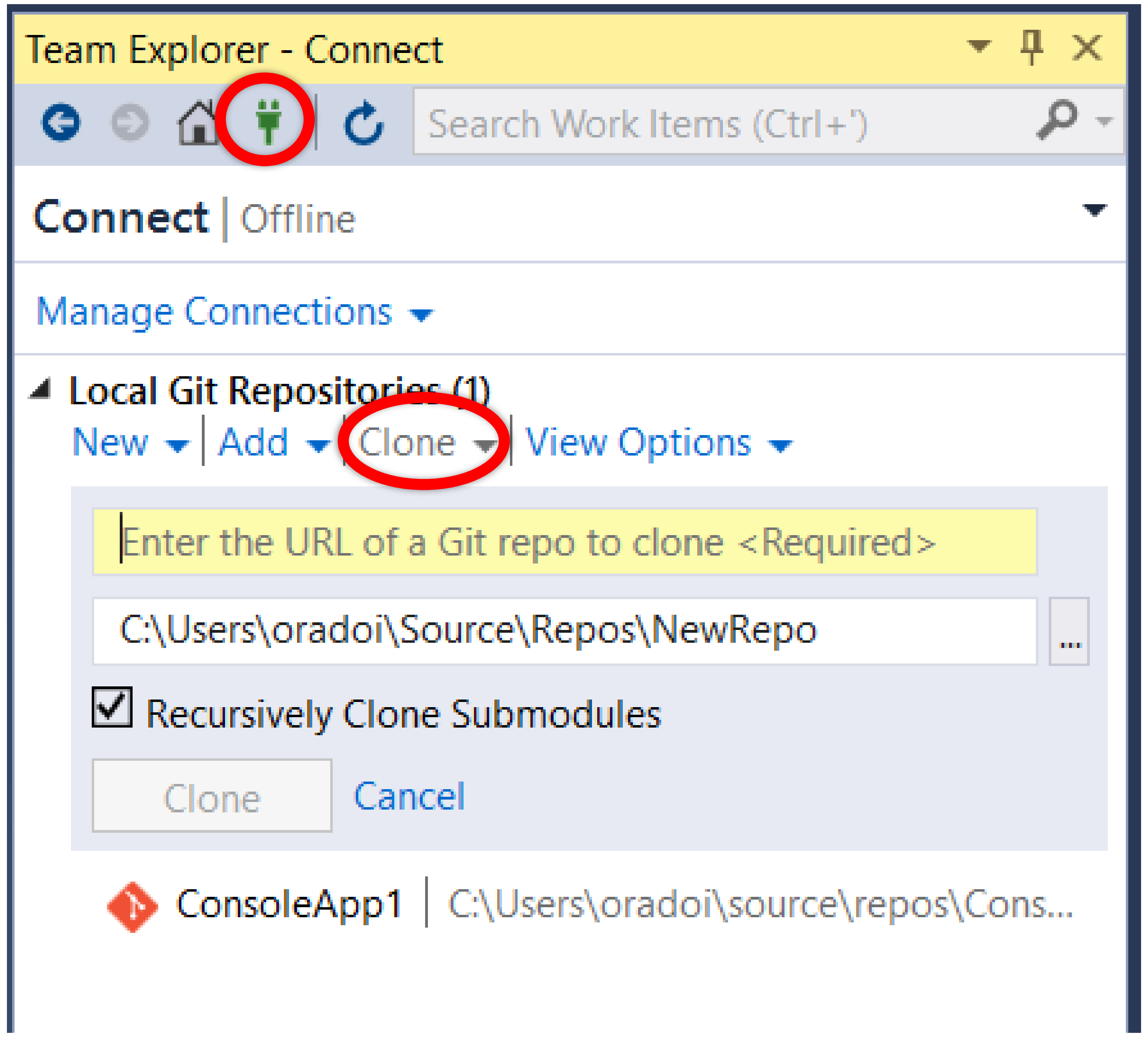
Clone from GitHub

- In order to clone a project from GitHub, we need to get the repository URL using the Clone or download button
- In Visual Studio, in the Team Explorer pane, we click on the Manage Connections button, and then on the Clone button
- We paste the URL in the next text box, and then the repository will be cloned



Use Git or checkout with SVN using the web URL.

https://github.com/oviradoi/HelloWorld.g



Enter the URL of a Git repo to clone <Required>

C:\Users\oradoi\Source\Repos\NewRepo

☑ Recursively Clone Submodules

ConsoleApp1 | C:\Users\oradoi\source\repos\Cons...

Pushing to a non-empty repo

- On GitHub, if you add a readme file, a gitignore file or a license file, then the repository will not be empty
- If you have initialized a repository in Visual Studio and you push it to a non-empty GitHub repository, then you will get a conflict because the two repositories have different files
- To fix this problem, the easiest way is to delete the GitHub repository, and create another empty one (no readme, no gitignore, no license)
- If the repository on GitHub is not empty and contains code that you don’t want to delete, then you have to clone it locally and open it with Visual Studio, and then modify the code, commit the code, and push it back to GitHub. This will work because the local and remote repositories contain the same history that you will add to

Using Download as zip instead of cloning

- If you want to download a project from GitHub, the correct way is to clone it. GitHub also has a “Download as zip” button that you can use to get the code.
- Cloning a project will download the whole repository, including its history. It will also remember the location of the project on GitHub. This allows you to modify the code and push it back to GitHub
- Downloading the code as a zip file will only download the latest version of the project. It will also not remember where the project came from. You will be able to open the project with Visual Studio, run it and modify it. But you will not be able to push the changes back to GitHub.
- You should use Clone when you are downloading your own projects, and you want to modify them. You should use Download as zip only when you are downloading somebody else’s code and you don’t want to modify it and push the modifications back to GitHub

Homework

- Initialize a repository for the Guess a number game project in Visual Studio
- Commit all the changes you made to the code files
- Create a GitHub repository for the Guess a number game
- Upload the repository you created locally to the repository on GitHub

Reference

- https://docs.microsoft.com/en-us/azure/devops/repos/git/creatingrepo?view=vsts&tabs=visual-studio
- https://docs.microsoft.com/en-us/azure/devops/repos/git/share-your-code-in-git-vs-2017?view=vsts
- https://docs.microsoft.com/en-us/azure/devops/repos/git/commits?view=vsts&tabs=visual-studio
- https://docs.microsoft.com/en-us/azure/devops/repos/git/pushing?view=vsts&tabs=visual-studio
- https://try.github.io/