Git and Visual Studio

# Introduction

Git is the most popular to manage code today. As a code management system, git helps us manage code as we continue to develop it. Managing code means keeping track of changes in a way that allows us to recover previous versions and gain insights into how the code has evolved over time. This can be particularly useful for teams working on shared software projects.

Git manages code in what are called repositories. A repository is a collection of related code. Each repository stores a full set of changes that have been applied to the code over time. Repositories are managed through a rich set of command-line instructions. These include

git clone: to install a copy of an existing repository, fetched from a cloud-based service like GitHub or Bitbucket

git init: to start a brand new repository on your hard drive

git add: to add the changes you’ve been working on to something called the staging area, which is a place where code sits right before you commit it to become part of the repository’s content and history.

git commit: to commit the changes you made to the repository’s content and history so that it becomes a permanent, retrievable piece of it.

git push: to publish your recent changes to a cloud-based service like GitHub or Bitbucket

git pull: to pull the latest edition of a repository from GitHub or Bitbucket onto your own machine.

Speaking of your own machine, we should distinguish between the working directory and the repository. The working directory is where you are working on your code, applying changes like adding, deleting, and modifying code. The repository is usually – but not always – a subfolder of your working directory called .git that you don’t modify directly but instead issue git commands against and serves as the permanent history of everything you’ve done to the code.

In my professional software work, I prefer to issue commands from the command line to manage my code using git. However, many software development tools – including Visual Studio – have built-in tools for managing code with git. This note set will introduce you to those tools.

# Working with git and GitHub in Visual Studio

We’ll learn by example.

## Creating a new repository

Create a new repository for your work in this course by selecting File >> New Repository. You’ll be greeted with this dialog.

A screenshot of a computer

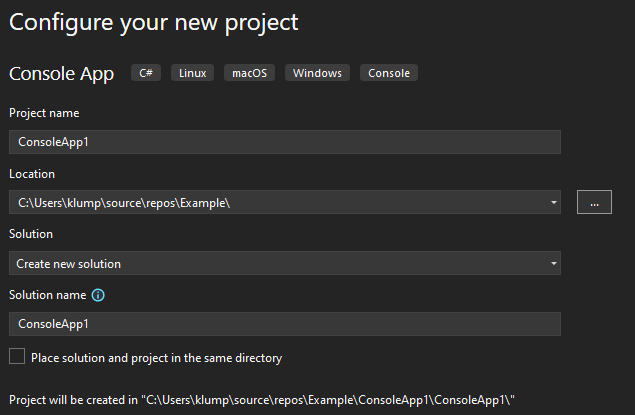
Description automatically generated

Fill in the path where you want to house the repository. Remember that this path will basically serve as your working directory. In the example, I have indicated that I want to house the new repository in c:\users\klump\source\repos\example. I’ve also indicated that I want to use the default .gitignore template. The .gitignore file tells the git tool what kinds of files and which folders to ignore when committing changes to the repository. While you can manually edit that .gitignore file, and while every repository should have one, we’ll take advantage of Visual Studio’s help here and just use the default one Microsoft supplies. Finally, we’ll leave the License template value as None, but we could decide that we want to use a particular license to establish the legal terms of distributing our software if we wished.

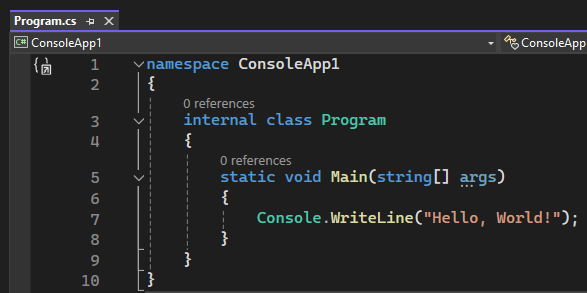
In the bottom half of the dialog, I’ve also specified that I want to link the new local repository with one in the cloud, specifically at GitHub. I already have used git and GitHub on my machine, so it filled in the name of the Account and the owner. If you’ve never used git and GitHub with Visual Studio before, you need to click the down arrow on the Account dropdown and choose “Add an account”, in which case you’ll be take to a login screen for logging into the GitHub account you want to use. (Of course, this assumes you already have a GitHub account you want to use. If you don’t already have one, please go to github.com to create one.) You see from the dialog that I have chosen to call the new repository at GitHub that will be linked to this new local one “Example20250112”. You’ll see that I’ve also chosen to describe it using a brief but descriptive phrase.

Press “Create and Push”. Press “Yes” to on the ensuing dialog to continue.

Now create a new Visual Studio project to add to the repository. Select File >> New >> Project. Select Console App (.NET Framework) from the “Recent project templates folder”, or search for “Console App” in the search bar and then select “Console App (.NET Framework). Set the location to match where you created the repository.



The Program.cs file will appear.



This is code we can commit to our local repository and then push to the remote one. Choose git >> Commit or Stash. The Solution Explorer will transform to show this:

A screenshot of a computer program

Description automatically generated

Type a descriptive comment into the text area and then click “Commit All” The top of the Solution Explorer will then say something like this, which indicates that a new version of the code has been added to the repository. A “version” – also called a snapshot – is a collection of all the code files at a particular point int time. Each snapshot is identified by a unique hexadecimal id.

A screenshot of a computer

Description automatically generated

When you want to push the contents of your local repository to GitHub so that others can pull them, choose Git >> Push from the main menu. When you do, you should see this message at the top of the Solution Explorer:

A screenshot of a computer

Description automatically generated

If there are other people working on your same project, they might have committed changes locally and pushed them to GitHub asynchronously from you. So, if you’re working in a team, it would be better to choose Git >> Sync (Pull then Push). This pulls their recent changes into your repository so that your subsequent push will include their changes in addition to yours.

## Using an existing repository from GitHub

Select Git >> Clone an existing repository. Type the URL of the repository into the text field. For example, the Workspace of in-class examples this semester is https://github.com/klumpra/cpsc23000sp2025.git. Paste or type that address into the text box. Also choose a folder to house it.

A screenshot of a computer

Description automatically generated

## Using an existing repository that is already on your machine

Choose Git >> Local Repositories and, if you don’t see the name of the repository you want to open already in the menu, scroll down to “Open local repositories …”. Navigate to the folder where that repository is located and select it.

Alternatively, if all you’re interested in is just creating a new project that will be part of an existing repository, create the new project in the folder where the repository is housed. For example, I have the workspace for in-class examples repository housed in the folder C:\Users\klump\Dropbox\2025Spring\cpsc23000\workspace. Choose File >> New >> Project and situate it in that folder.

# Summary

While it is always a good idea to master the command line and use it for things like git, Visual Studio’s support for git and GitHub is really good. In this chapter, we learned how to use Visual Studio’s built-in git tools.