Module 01 Lab 02 Worksheet

Introduction to Version Control

# Overview

Regardless of what programming language or development environment you use, you should know at least the basics of version control. Sometimes known as source control, version control works by tracking the changes to your project folder as you add, modify or delete code. This now only lets you keep a backup of your code, but also protects you from yourself. For example, if you accidentally delete a large block of code, you can recover it *even if you've saved the modified file*.

VS Code has built-in integration with the command line version control utility git. This makes it easy to commit and track your changes, as well as rollback your project code if you happen to program yourself into a corner.

|  |
| --- |
| **NOTE**: This lab assumes you've done Lab 01 in this module and have set up your developer workspace. |

|  |  |
| --- | --- |
| * Start VS Code. * Close any project folder that may still be loaded, if necessary. * Click on the Explorer icon. * Select Open Folder. * In the Open/Save dialog, click on New Folder. * Create a new, empty folder called git-lab. * Click on Open. | 0 |
| * Create a new file called file01.txt. Add the line   This is the first file in my new project.   * Save the file. * Click on the Git icon in the Side Bar.   Is this workspace currently under version control? (5 pts.)  no | 5 |
| Click on Initialize git repository.  How does the Git icon change? (5 pts.)  First a clock appears then is replaced by a number 1. | 5 |
| Notice that file01.txt is now listed in the Git window. Note also that the file name is marked with a 'U'. Hover your mouse over the 'U'. What does the tooltip say?  untracked |  |
| A *git repository* is a local folder in your project that holds copies of your files and file changes. Git takes a 'snapshot' of your current project folder. That process is called *committing changes* or just a *commit*.  Our changes go through three steps:   1. *Untracked* (git doesn't know about them yet.) 2. *Staged* (we're getting the changes ready for the commit.) 3. *Committed* (our changes have been recorded.)   Whenever you commit a set of changes, you will be prompted to add a short note describing those changes. Ex. 'fixed password bug' | 0 |
| To the right of file01.txt, you'll see a plus sign. Hover your mouse over it. What does the tooltip say? (5 pts.)  Stage Changes | 5 |
| Hover your mouse over Changes and you'll also see a plus sign to the right. What does the tooltip for this one say? (5 pts.)  Stage all Changes | 5 |
| You can stage a single changed file or all of them if you wish. Since we have only one file to commit, click on the Stage All button. (The plus sign to the right of Changes)  Describe at least three things that changed. (5 pts.)  The plus signs changed to minus signs.  The U changed to an A.  There is now a “CHANGES” part under the STAGED CHANGES area. | 5 |
| In the message box above Staged Changes, enter the text  First commit for Module 1, lab 02.  Hover your mouse over the checkmark symbol at the top (DON'T click on it yet!). What does the tooltip say? (5 pts.)  Commit | 5 |
| Double-check your commit message, then go ahead and click on the checkmark  Describe at least two things that changed.  The file was removed.  The STAGED CHANGES area was removed. | 5 |
| Go back to your File Explorer.  Add a new line to file01.txt - This is the second line of this file.  Save the change.  What happens to the Git icon? (5 pts.)  A number 1 apears.  Explain this change. (5 pts.)  This is telling me that one file has been changed and needs committing. | 10 |
| Add a new file to your project called file02.txt.  What happens to the Git icon? (5 pts.)  A number 2 appears | 5 |
| In file02.txt, add the line  All your base belong to us  Save your changes.  Click on the Git icon.  What is the status of file01.txt? (5 pts.)  Modified  What is the status of file02.txt? (5 pts.)  Untracked | 10 |
| Enter the commit message  Added file02.txt  And click on the checkmark to stage and commit your changes in one step.  How does the Git icon change? (5 pts.)  The number 2 is removed. | 5 |
| Go back to the Explorer view.  In file01.txt, change the first line to read  The town where I grew up has a zip code of E-I-E-I-O.  Save your changes.  Go to the Git view.  In the Changes section, you will see file01.txt has been modified. To the right of the file name, you will also see that there is a backward-curving arrow. Hover your mouse over it. What is it labelled? (5 pts.)  Discard changes  There is another similar arrow to the right of Changes. What is it labelled? (5 pts.)  Discard all Changes | 10 |
| You haven't committed the changes to the file yet, so you can undo them even if you've saved the file. Also, you can undo a single file change or all *uncommitted* changes if you wish.  Click on the backward arrow to the right of file01.txt in the Git window.  When asked if you are sure you want to clean changes in file01.txt, click on Clean Changes.  Look at file01.txt. What is the first line in the file now? (5 pts.)  This is the first file in my new project. | 5 |
| Create a new folder in your project called docs.  Does the Git icon change? (5 pts.)  no | 5 |
| Inside docs, add two files called doc01.txt and doc02.txt. You can put text in them if you want but it's okay to leave them empty for now.  What happens to the Git icon? (5 pts.)  A number 2 appears | 5 |
| Go ahead and commit your changes with the message  Added docs folder | 0 |
| In the Explorer view, delete the following:   * The docs folder * file01.txt   What does the Git icon look like now? (5 pts.)  A number 3 appears | 5 |
| Go to the Git view. There is a 'D' next to the three changes. What does the tooltip for this say? (5 pts.)  Deleted  What else is different about the names? (5 pts.)  The file names have a line through them | 10 |
| Go ahead and click on the backward arrow to the right of Changes to Clean All.  Accept the changes when prompted.  Go back to the Explorer view.  What happened to your project? (5 pts.)  All the files came back | 5 |
| Open the Integrated Terminal. At the command prompt, enter the command  git log  Copy and paste the output here. (5 pts.)  commit 00264df58073e37c50332bcf61d797e20e83f18a (HEAD -> master)  Author: Chris Schmidlin <chris.schmidlin@gmail.com>  Date: Sun Aug 11 13:19:02 2019 -0500  Added docs folder  commit 94ea4dd85c28c8aa844aba60d737eba0812f270b  Author: Chris Schmidlin <chris.schmidlin@gmail.com>  Date: Sun Aug 11 13:15:06 2019 -0500  Added file02.txt  commit 73ccc125d4c67b37248686170a0c4442e6f09886  Author: Chris Schmidlin <chris.schmidlin@gmail.com>  Date: Sun Aug 11 13:12:34 2019 -0500  First Commit for Module 1, lab 2. | 5 |
| While still in the terminal, enter the command  git status  Copy and paste the output here. (5 pts.)  On branch master  nothing to commit, working tree clean | 5 |
| **Total** | 110 |

# Summary

This lab introduced you to the basics of version control using git and Visual Studio Code. Version control is a valuable programming tool because it helps keep your project files and folders safe.

Though git is a command line utility, we used the Visual Studio Code graphical interface to do all of our basic version control tasks. We can also access the git command directly via the Integrated Terminal in Visual Studio Code if desired.