PLEASE READ WHOLE DOCUMENT BEFORE STARTING, SOME PROBLEMS RELY ON OTHER SOLUTIONS TO MOVE FORWARD.

These are just the steps Microsoft provided on GitHub for GVFS, but my sub bullet points are some problems I ran into and some solutions I found while working on this project. I have never worked on a project like this so my explanations have some detail and are also to help those like me who don’t have much experience. Finally, I would like to encourage anyone and everyone who is interested in this project to give it a try.

**Building GVFS**

* Installing Visual Studio 2017 Community Edition or higher
  + When installing this software the only problem I ran into was not seeing the .Net Framework 3.5 development tools so I chose to install later .Net versions. (After completely installing Visual Studio I found out the .Net Framework 3.5 development tools are installed by default, or at least for this project.)
* Installing InnoSetup 5.5.9 or later
  + As it says, installing it to its default location is the best option.
* Create a folder to clone into, e.g C:\Repos\GVFS
  + I think using the example provided makes it easier to follow.
* Clone this repo into the src subfolder, e.g. C:\Repos\GVFS\src
  + Once again, using the example makes it easier to follow.
  + Also, make sure to clone correctly using the clone link on GitHub. The reason I mention this is because I had never used Git before and learned it to run this project. I thought downloading the zip file would work just fine but that was not the case, so I had to go back and delete the files I had created and remake them, this time cloning it using the link on GitHub.
* Open src\GVFS.sln in Visual Studio. Do not upgrade any projects.
  + The problem I ran into in this step was when opening the solution, I told it to use the latest version of .Net Framework on my Visual Studio which was .Net Framework 4.6. Doing this caused it to not build correctly since it was not the original .Net framework the project used. Had to close it and open it again.
* Build GVFS.sln
  + Once I reopened the solution, it once again asked if it should use the most current framework and I chose the option to use the .Net framework that the project originally used. When I did this, it was able to build correctly.

**Testing GVFS**

* GVFS requires Windows 10 Creators Update (Windows 10 version 1703) or later
  + To find what version you have do: Windows+r and type “winver” and if it’s not version 1703 or later you will need to update. To update, on Windows 10, you can do: Start Menu > Settings > Update and Security > Windows Update. You will see a message about updating to Windows 10 Creators Update and click on “Yes, show me how” and download it that way.
* Install GVFS-enabled Git for Windows (2.13.0.gvfs.1 or later) from <https://github.com/Microsoft/git/releases>
  + Make sure to get it from this website and that the format is correct, X.XX.X.gvfs.X.
* Install GVFS from your build output
  + If you named the files using this example, then GVFS should be in the directory mentioned.
* GVFS will work with any git service that supports the GVFS [protocol](https://github.com/Microsoft/GVFS/blob/master/Protocol.md). For now, that means you'll need to create a repo in Team Services (<https://www.visualstudio.com/team-services/>), and push some contents to it.
  + You will have to add the file mentioned manually in Team Services.
* gvfs clone <URL of repo you just created>
  + After making a mistake the first time, I cloned it correctly at this step.
* cd <root>\src
* Run git commands as you normally would
* gvfs unmount when done

**Creating skeleton for python port of GVFS**

* When creating this python port, I first had to know exactly what all the commands and their arguments were. To do this I used GitBash and then put them all into a word document so I could use them as a reference later.
* To start getting an idea of how to make the skeleton I was introduced to a python package called argparse. I learned about argparse and did a few examples to try and understand it better.
* I was also introduced to another python package called click and did some examples.
* After doing skeleton examples from both packages, I chose to use click to make the GVFS skeleton,
  + I chose to use click because I felt it was going to be easier for me as a beginner and probably for other beginners who try working on this project.
  + I also chose it because it will allow people who have not encountered or known about click to pick it up.
* Finally, once I had made the skeleton using click, I made sure to try and format it in the way I thought was most understandable and left some comments on things that could possibly be added.

**Used Software and Programs**

* Visual Studio
* InnoSetup
* GitBash
* Git
* Command Line
* PyCharm