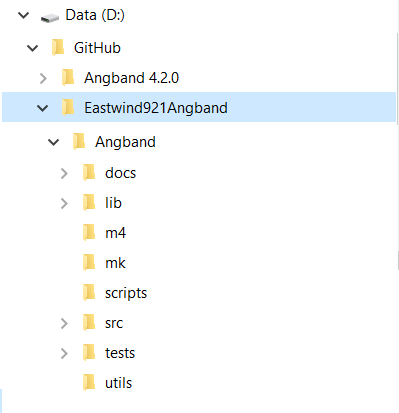
These instructions were developed for building and developing Angband using Visual Studio 2019 Community Edition (the free version of VS). As of the writing of these instructions, Angband 4.2.0 is the supported version, but the ‘master’ version contains additional changes and fixes, some of which are needed to build with VS. Those will be noted below.

1. Install Visual Studio Community 2019 from Microsoft.
   1. I didn’t keep a record of my install experience, so the following is from memory.
   2. I think I chose the “Desktop development with C++” workload.
   3. I don’t think I needed or chose any optional components during the install.
   4. There was some welcome spam to click through.
   5. After VS was installed I ran it. If the top menu bar wasn’t displayed I turned it on.
   6. Under Extensions->Manage Extensions I searched for Git. There are several extensions that support Git, the one I chose was “GitHub Extension for Visual Studio”, created by GitHub, It is version 2.10.8.8132 and was free.
2. Register with GitHub and fork yourself
   1. It’s easiest to register outside of VS, using a web browser pointed at github.com.
   2. Once you have github credentials and they are validated via email, login via a web browser.
   3. Point your browser at the master angband respository: github.com/angband/angband.
   4. Click the ‘fork’ button in the upper right (the button looks like it’s been split in two with a number on the right half). After a moment, this creates a private fork off master which you will need.
3. Run VS and connect to GitHub
   1. Start VS, there should be a window open on the left.
   2. The left hand pane of VS has tabs at the bottom, one of them should be Team Explorer, click on the tab. If the tab isn’t there for you yet, use the menu bar View->Team Explorer to bring up team explorer in that window (and add the tab for it).
   3. You should see a section named simply “GitHub”. In the GitHub section, click Sign In and enter your GitHub credentials and get yourself signed into GitHub from within VS and connected.
4. Set up an empty directory structure somewhere on your computer where you will build & develop Angband.
   1. Start by making an empty folder somewhere convenient to you. This will be your “solution folder” or “solution directory”.
      1. You can set the name of the solution folder to something that denotes not only that it’s for Angband but also which version of Angband. For example, I have one named Angband\_4.2.0 and another named “Angband\_Master” and a third where I’m doing development for my personal variant. I *was* able to make things work with spaces in these folder names, but if you can avoid any spaces in the path or folder name you may avoid some hassles in the future, so it might be wise to use underscores instead of spaces in the names.
   2. Create a second folder as a subfolder within your solution folder, and name it Angband. This will be your “project folder” or “project directory”.
      1. The checked-in VS files have a dependency on the project directory being named Angband, so use exactly that.
      2. For VS development, it’s best to stick with the VS standard organization of project directories being in an immediate sub-folder of the solution folder rather than mixing everything into one folder. Separating them allows you to later create other related projects underneath the same solution, for example for testing or tools or scripts that you use with Angband but aren’t part of the game itself.
      3. This second folder must be empty for Git Clone to work.
5. Clone Angband
   1. In VS, in the Team Explorer window, in addition to the GitHub section where you logged in, you should also see a “Local Git Repositories” section, with New, Add, Clone and View Options pulldowns. Click the Clone pulldown.
   2. In the (yellow) text box, where it says ‘enter the URL of a git repository’, enter the URL of your fork of master (don’t enter master!). For example, my Git ID is Eastwind921, so my fork was automatically named Eastwind921/angband. Therefore to clone my fork I entered:

<http://github.com/Eastwind921/angband> in the yellow text box.

* 1. In the text box below the URL, enter the path to your project directory, which is the lower of the two folders you created in step III c, the one that must be empty.
  2. I left Recursively Clone Submodules checked, but I don’t know if there are any submodules for angband.
  3. Click the Clone button and wait for it to download everything. Status shows in the VS Output window. Once everything is downloaded, close VS.
  4. At this point your solution folder should contain only your project folder, named Angband, and the Angband project folder should contain a number of files and directories, including the src and lib folders. These must be directly within the project folder as shown in the image below. In this example, I located my solution folder, named “Eastwind921Angband”, and highlighted in blue, next to another solution folder named “Angband 4.2.0”, both under a D:\GitHub folder I use to contain all my GitHub projects. The project folder named Angband is below it, and all the Angband code is below that.



* 1. The VS solution file (.sln filetype) is not checked into the GitHub repository in the location where you need it to be to use it. Copy (or move) it from src\win\vs2019\Angband.sln to your solution directory (Eastwind921Angband in the example).
  2. You can change the filename of the solution file to match the name you used on your solution folder; doing so will help you later if you have multiple instances of VS running working on multiple solutions at once, you’ll be able to tell from within VS which one is which.
  3. Move or copy 3 other files, Angband.vcxproj, Angband.vcxproj.filters and Angband.vcxproj.user from the src\win\vs2019 folder to your Angband project directory (Eastwind921Angband\Angband). Don’t rename any of these files.
  4. Reopen VS with the solution file by double clicking on the solution file or by running VS and using the File->Open->Project/Solution menu choice.

1. Configuration changes before building
   1. The provided solution files have a number of settings changes ‘baked in’. These override default VS build settings, compiler and linker options, etc, as needed to build Angband. There shouldn’t be any additional project configuration changes necessary unless something changes in the future.
   2. One thing to double-check is in the toolbar immediately below the VS menus (this is the ‘standard’ toolbar, in case yours isn’t visible). It has a pair of pulldowns where you set the configuration you are building. It should already be set to Debug and x86. The x64 build doesn’t work, and would be slightly larger and very slightly slower if it did.
2. Code fixes before building
   1. If you are trying to build an older version of Angband, including 4.2.0, you will need to apply some fixes to get it to build and run. These have been fixed in the angband master as change 616ba762 (“Fixes for Visual Studio Compilation by Gordon and Eastwind”) and 5d2e4f54 (Changes for VS 2019 compilation). The diffs are not large if you’d rather apply them by hand to your copy than try to sync them down using GitHub. No attempt has been made to build versions prior to 4.2.0, additional fixes might be required.
3. Build
   1. Hit the F7 Function Key to build.
   2. The code should compile and link without errors, but with a number of warnings.
4. Post Build Step
   1. To run, Angband needs to be able to find and load various text files contained under the lib folder. The approach used here is to copy the lib folder to the built executable (versus moving the executable). The reason for that is by default VS supports both Debug and Release configurations, and copying the lib directory to the executable’s location allows multiple configurations to coexist, whereas doing it the other way would cause a collision of the executable files from Release and Debug, and building one configuration would overwrite the executable built by the other configuration.
   2. The provided solution files automate this step completely by doing two xcopy commands after every successful build. You can see the output from these commands in the output window. The first time you build everything will get copied (2 files from the first xcopy and about 343 from the second). After that, if nothing has been updated under ‘lib’, then you’ll see “0 files copied” (twice) as the xcopy commands are set up to only copy updated files. If you build successfully and then hit ‘F7’ again, you’ll see that nothing needed to be built but the post-build step will run again, copying 0 files.
   3. If you make changes to lib data, you must use ‘F7’ to “build” your change, which causes xcopy to refresh the copy from lib, so that running Angband will pick up the changes. Often you will change both, and the automation keeps you from having to remember to copy the text files to the executable.
5. Run
   1. Hit ‘F5’ to run what you built under the debugger.
   2. In case you don’t want to run under the debugger, the Angband.exe executable is located in the Debug directory under your solution folder. (There’s another Debug directory under your project folder, that contains your intermediate build results files)