

Project Minetest Setup Guide

Originally Written By: Miranda Craghead

Updated By: Danny Moolchand

Updated Again By: Luke Herczeg (with help from Jahdiel Suarez)

Updated Once More By: Nathan Whelden

Table of Contents

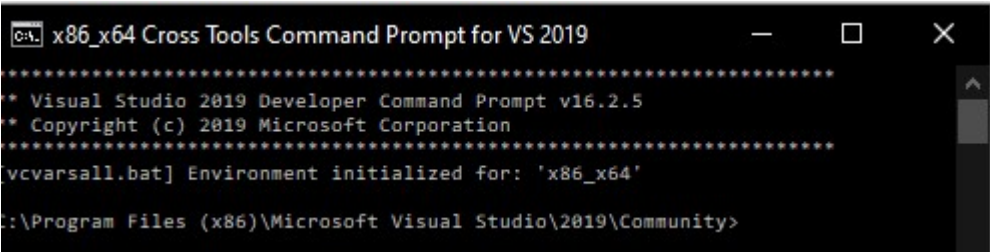
Step 1: Install Minetest Locally	3
Step 2: System Setup	3
Step 3a: SpiderMonkey DLL Linking.....	5
Step 3b: Visual Studio Setup	7
Step 4: Run The Game!	10
Appendix A: Previously Encountered Errors & Solutions.....	14
Appendix B: Helpful Notes.....	14

Step 1: Install Minetest Locally

- 1) Create a folder named DIR
 - a. I put it in my C driver under the path `C:\users\<name>`
 - b. I got several errors from paths having spaces, so DO NOT choose paths with spaces
 - i. If you can't avoid this (I.E. Your computer name has a space in it – like Danny's PC) refer to the error section at the end for a solution.
- 2) Clone the Minetest game from the GitHub master branch
 - a. Open the windows command terminal and cd into DIR directory
 - b. Type **git clone <https://github.com/cacticouncil/minetest.git>**
 - c. Took a few minutes

Step 2: System Setup

- 1) Visual Studio 2017/2019
 - a. URL to older editions of visual studios:
https://my.visualstudio.com/Downloads?q=visual%20studio%202017&wt.mc_id=o~msft~vscom~older-downloads
 - b. I kept the original paths for visual studios, but this is up to you
 - i. Although I kept the defaults, I got an error as my visual studios path had spaces in it. Check the error section at the bottom for a solution.
 - ii. Note I solved the solution in the visual studios command terminal, which can be found by searching command in the windows search box and looking for the name provided below.



```
C:\> x86_x64 Cross Tools Command Prompt for VS 2019

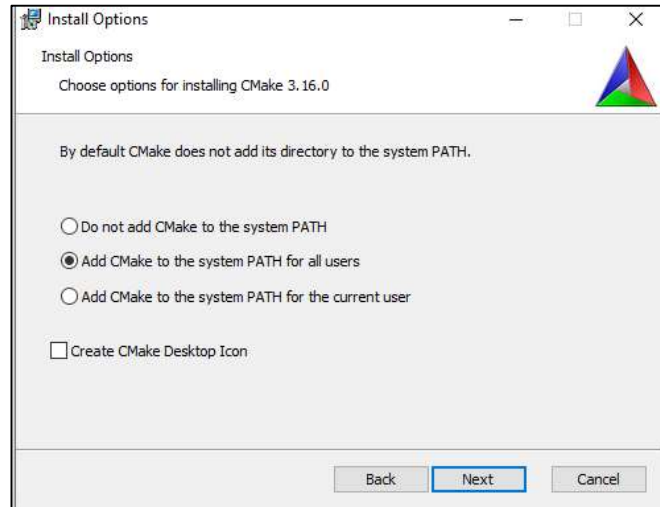
*****
* Visual Studio 2019 Developer Command Prompt v16.2.5
* Copyright (c) 2019 Microsoft Corporation
*****

[vvarsall.bat] Environment initialized for: 'x86_x64'

C:\Program Files (x86)\Microsoft Visual Studio\2019\Community>
```

2) CMake

- a. Link to download for x64 [cmake-3.16.0-rc3-win64-x64.msi](#)
- b. Had compatibility errors with other window versions (X32, X84)
- c. Make sure you add CMake to the system path either for all users, or just you

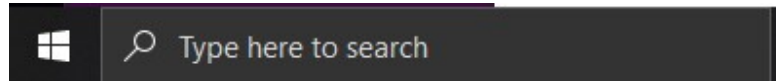


3) Vcpkg

- a. URL: <https://github.com/Microsoft/vcpkg>
- b. Open the windows command line terminal and power shell
 - i. I ran the command line terminal from my user/name folder
 - ii. The DIR/minetest directory exists in my user folder
- c. In the command line terminal type:
git clone https://github.com/Microsoft/vcpkg.git
 - i. PAUSE ANTIVIRUS BEFORE INSTALLING
 1. I had various problems installing vcpkg due to my antivirus that disappeared after pausing during instillation.
- d. In the PowerShell commands
 - i. **cd vcpkg**
 - ii. **.\bootstrap-vcpkg.bat**
 - iii. **.\vcpkg integrate install**
 - iv. **.\vcpkg install sdl2 curl**
- e. In the command line terminal type **vcpkg install irrlicht zlib curl[winssl] openal-soft libvorbis libogg sqlite3 freetype luajit --triplet x64-windows**
 - i. This step can take some time 10+ minutes
- f. TURN ANTIVIRUS BACK ON


Step 3a: SpiderMonkey DLL Linking

- 1) Open CMake GUI
 - a. CMake GUI should appear if you search cmake in your Windows search box



- 2) Configure Cmake
 - a. You may need to create a build folder inside minetest folder

Where is the source code:	<input type="text" value="C:/Users/miranda/DIR/minetest"/>	<input data-bbox="1317 617 1487 644" type="button" value="Browse Source..."/>
Where to build the binaries:	<input type="text" value="C:/Users/miranda/DIR/minetest/build"/>	<input data-bbox="1317 674 1487 701" type="button" value="Browse Build..."/>



Specify the generator for this project

Optional platform for generator (if empty, generator uses: Win32)

Optional toolset to use (argument to -T)

☐ Use default native compilers

☐ Specify native compilers

☒ Specify toolchain file for cross-compiling

☐ Specify options for cross-compiling

- b. Click configure
 - i. If you aren't prompted here under files click delete cache then re-click configure
 - ii. Select the vcpkg toolchain file
e.g. D:/vcpkg/scripts/buildsystems/vcpkg.cmake



Specify the Toolchain file

- c. Assuming no errors were generated from configure hit generate
 - i. Various warnings are expected

- ii. If errors are generated click **file** at the top left corner and select **delete cache**. After this try reconfiguring, and generating. Hopefully the errors will be gone.

1. The solution above worked for this error

```

Configure Generate Open Project Current Generator: Visual Studio 15 2017

Selecting Windows SDK version 10.0.17763.0 to target Windows 10.0.18362.
*** Will build version 5.1.0-dev ***
Detecting GMP from system failed.
Using bundled mini-gmp library.
Using bundled JSONCPP library.
Using LuaJIT provided by system.
CMake Warning (dev) at C:/Users/Miranda Craghead/vcpkg/installed/x86-windows/share
if given arguments:

```

2. If CMake can't find SQLITE3, choose "Advanced" box on the right top corner, then specify the location of SQLITE3_INCLUDE_DIR and SQLITE3_LIBRARY manually

```

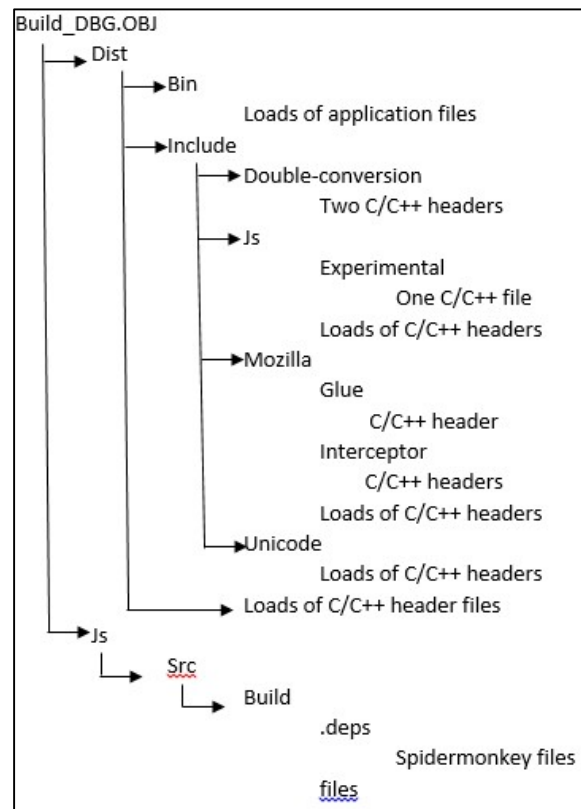
1 CMake Warning at src/CMakeLists.txt:52 (message):
2   cURL is required to load the server list
3
4 CMake Error at C:/Program Files/CMake/share/cmake-3.15/Modules/FindPackageHandleStandardArgs.cmake:137 (message):
5   Could NOT find SQLite3 (missing: SQLITE3_LIBRARY SQLITE3_INCLUDE_DIR)

```

- d. Hit open project

- 3) Link the SpiderMonkey dll

- a. Download SpiderMonkey **build_DBG.OBJ** folder from [here!](#)
 - i. I saved it under the path C:\Users\<name> after extract all/unzipping it
 - ii. The download takes a decent amount of time
 - iii. Note several team members experienced errors with not getting the entire install for the build_DBG.OBJ, causing errors. File structure provided below to eliminate this error.



- 4) Copy these files to `C:\DIR\minetest\bin\Debug`
 - a. `C:\build_DBG.OBJ\dist\bin\mozglue.dll`
 - b. `C:\build_DBG.OBJ\js\src\build\mozjs-71a1.lib`
 - c. `C:\build_DBG.OBJ\js\src\build\mozjs-71a1.dll`
 - d. `C:\vcpkg\installed\x64-windows\bin\zlib1.dll`
- 5) Replace '`C:\build_DBG.OBJ\dist\include\mozconf.h`' with this [edited version](#).
- 6) Finally, to get minimal development test mode to run change line 9 of `mod.js` in the directory `\DIR\minetest\mods\simple_wasm`

From: `var buffer = new Uint8Array(%wb[the default path to the mod.wasm file]wb%)`

To: `var buffer = new Uint8Array(%wb[the absolute path to the mod.wasm file]wb%)`

```
mods > simple_wasm > JS mod.js > utf8ToString
1 function utf8ToString(mem, pos) {
2   let s = "";
3   for (i = pos; mem[i]; i++) {
4     s += String.fromCharCode(mem[i]);
5   }
6   return s;
7 }
8
9 var buffer = new Uint8Array(%wbC:\Users\Natha\DIR\minetest\mods\luke_mod_wasm\mod.wasmwb%);
10 var m = new WebAssembly.Instance(new WebAssembly.Module(buffer));
11 var memArray = new Uint8Array(m.exports.memory.buffer);
12 var ex = m.exports;
```

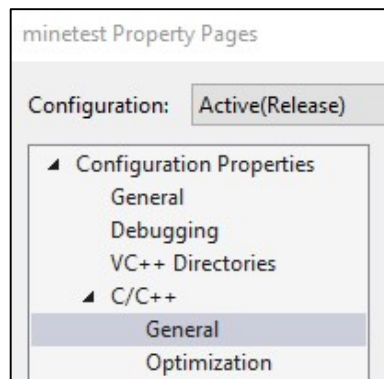
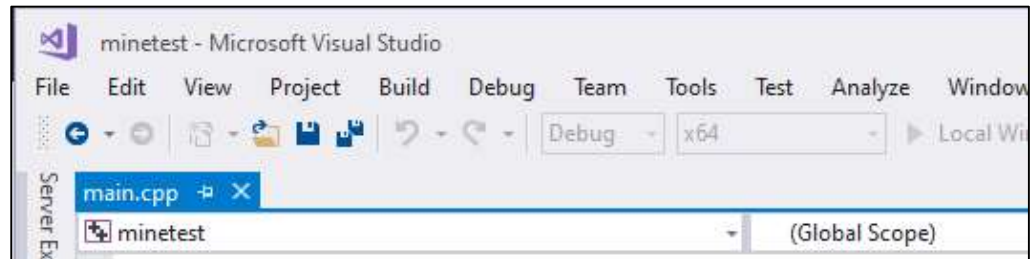
Step 3b: Visual Studio Setup

1. Visual Studios steps
 - i. Replace `main.cpp` with the updated `main.cpp` under the Source Files. Link:
https://cacticouncil.slack.com/files/UM999UGL9/FPD1TDGHK/main.cpp?origin_team=TOWG WK859&origin_channel=CFFQURZA9

- b. **IMPORTANT:** Comment Out lines 140-141. These lines attempt to reinitialize the SpiderMonkey engine twice when starting the Minimal Version of the Minetest Game. Comment them out so the game can launch.


```
136 int main(int argc, char *argv[])
137 {
138
139     //This initializes spidermonkey, the game tries to do it twice so we remove this for the minimal version to work
140     //int sm_status = sm_hello_world();
141     //std::cout << "SpiderMonkey Status: " << sm_status << std::endl;
142
143
144     int retval;
145     debug_set_exception_handler();
```

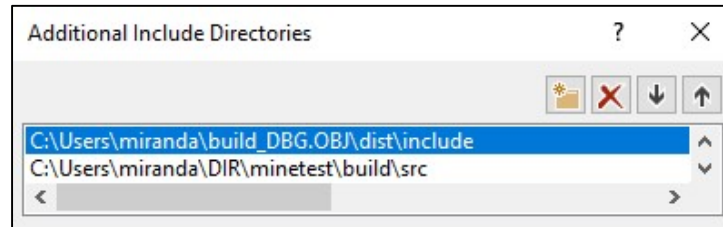
- i. Make sure you are on Debug x64 before attempting to link, otherwise you will experience errors.
1. If you do link in the wrong setting you can try to relink in Debug x64. If the build still generates errors exit Visual Studios and go back to the prior step of Configure CMake.



- ii. At the upper right corner in the solution explore right click minetest and select properties
- iii. On the left option menu **Go to C/C++ → General → Additional Include Directories**
1. Click on **Additional Include Directories** to highlight, then click the down arrow to the right and select edit



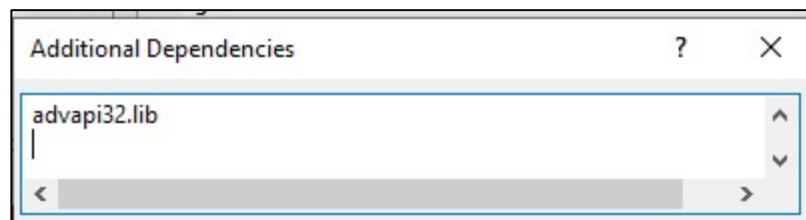
2. Add your path to **build_DBG.OBJ\dist\include** by clicking  and then pasting the path.



3. Click ok then apply
 - a. Note clicking ok twice will close the window and won't save your changes. **You need to click apply on the minetest Property Pages.**

iv. Go to **Linker → Input → Additional Dependencies**

1. Click on **Additional Dependencies**, then click the down arrow and select edit.
2. Scroll to the bottom of the list and then click enter to add a new space



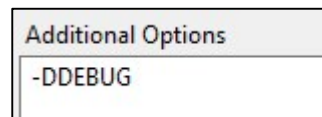
3. Paste **mozjs-71a1.lib** Then click ok and apply.

v. Go to **Linker → General → Additional Library Directories**

1. Click **Additional Library Directories**, then click the down arrow and select edit.
2. Paste your path to **build_DBG.OBJ\js\src\build** and then click ok and apply.

vi. Go to **C/C++ → Command Line → Additional Options**

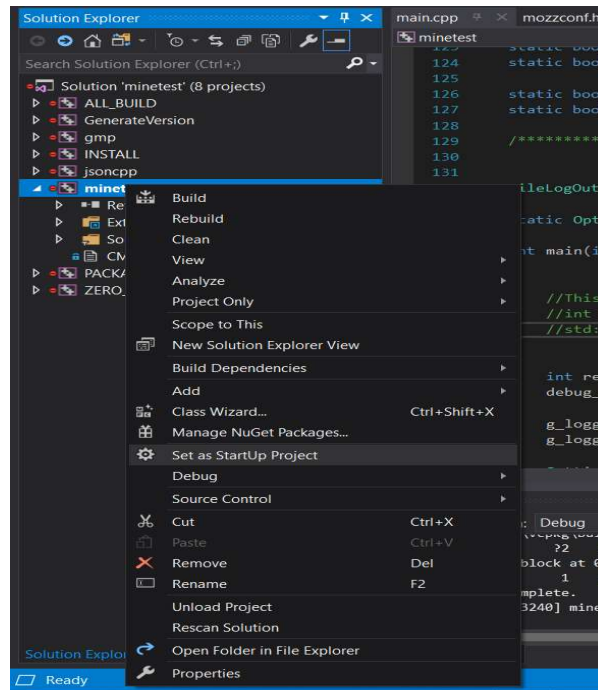
1. Replace **%(AdditionalOptions) /FD** with **-DDEBUG** or just write **-DDEBUG** if the box is empty



2. Click apply then ok to exit the minetest Property page

Step 4: Run The Game!

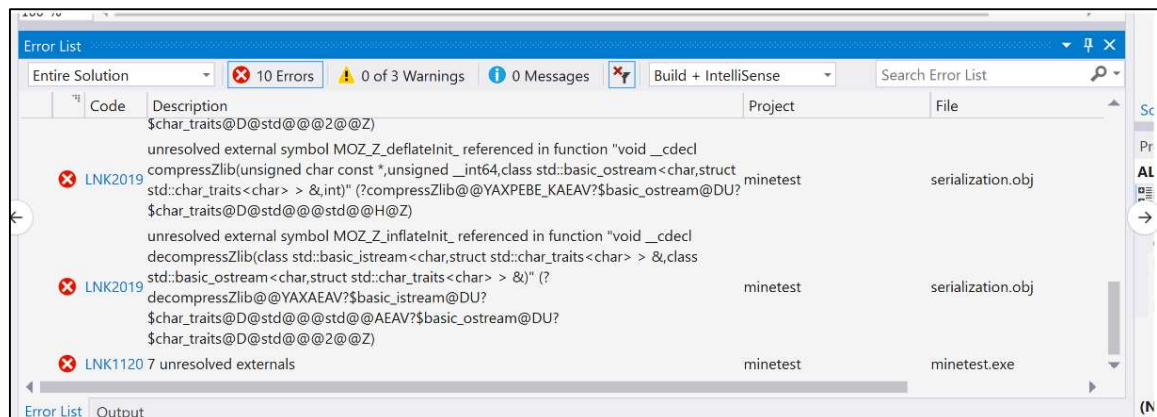
- 1) Right Click on **minetest** in the Solution Explorer and click on **Set As Startup Project**



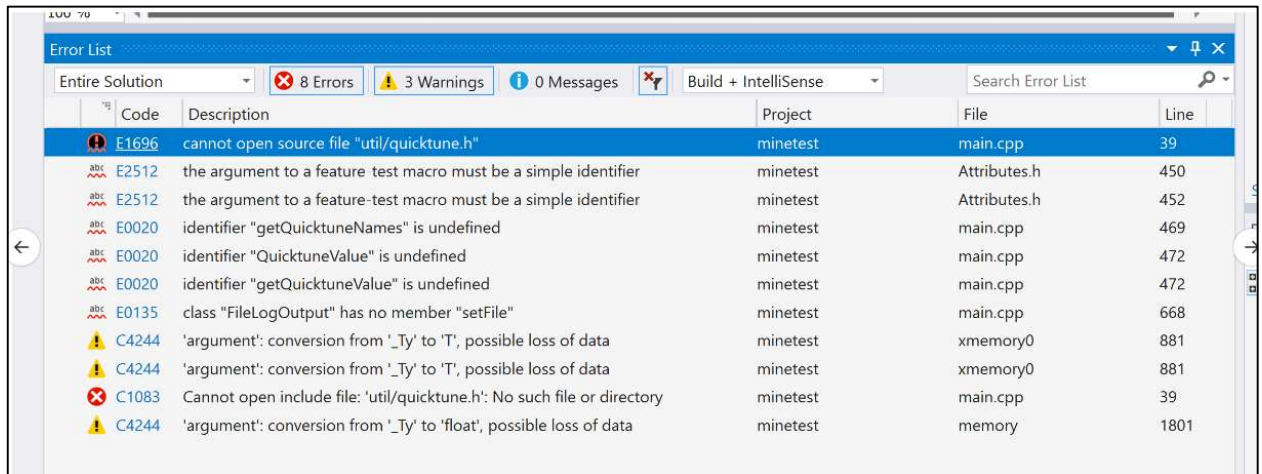
- 2) Run Local Windows Debugger (Running the Debug version from following the steps above)
 - a. If the game builds successfully go to file path **/DIR/minetest/bin/Debug** and select **minetest** to open the game



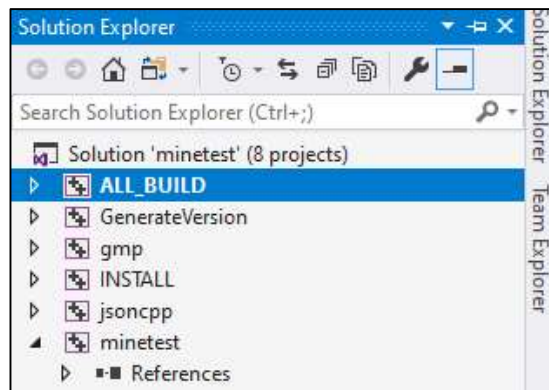
- i. If your build results in errors, such as these below the SpiderMonkey dll didn't link successfully



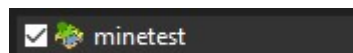
1. **Solution 1:** go back to #3 - link the SpiderMonkey dll and repeat the steps
2. **Solution 2:** close Visual Studios and go at Step 3 SpiderMonkey dll Linking.
 - a. If you get errors in CMake at the top left corner click **file** → **delete cache** and then try **configure** again.
 - b. Solution 2 is also relevant if you get errors that make no sense such as the ones below.



- 3) Building **release x64** minetest
 - a. Follow the steps in step 3 SpiderMonkey dll linking for #2 Configure CMake.
 - b. Once the visual studios project is open
 - i. Right click **ALL_BUILD** and select build



- ii. Assuming no errors go to **DIR/minetest/bin/Release** and select minetest to open the game



[OPTIONAL] IF YOU WANT TO PLAY THE NON-DEV VERSION OF THE GAME:

1. Clone the `minetest_game` from https://github.com/minetest/minetest_game.git to `DIR\minetest\games`
2. Navigate to `DIR\minetest\games\minetest_game\mods\creative`
3. Edit `init.lua` by commenting out lines 35 and 38:

```
33 function minetest.is_creative_enabled(name)
34     if name == "" then
35         --return old_is_creative_enabled(name)
36     end
37     return minetest.check_player_privs(name, {creative = true})
38     --old_is_creative_enabled(name)
39 end
```

By default the minetest engine comes with a minimal, lightweight version of the game called “devtest” for development. If you want all the game files, you need to download the full version.



If the game works the first time you open a 'devtest' world, and then you go back to the main menu and try to open the 'devtest' world again, it will FAIL, and **that is okay**. The current structure only allows you to initialize WASM mods one time per run of the game.

If the game works (and runs in 'devtest' mode), **YOU'VE DONE IT ALL!!!**

Now, take a deep breath and go for a walk.

Then, see the [**Minetest Modding Guide**](#) so that you can make your own mods, both with and without WASM!

Appendix A: Previously Encountered Errors & Solutions

- Refer to the link below to resolve errors involving spaces in path names
 - <https://manjusullad.wordpress.com/2016/01/06/how-can-i-find-create-short-path-for-windows-folders-directories/>
 - One thing to note, if you get an error while trying to set a short path name it's probably because the name you are setting is too long. If I recall correctly this command lets you set the path to up to 8 characters.
- Documentation providing information on bugs we encountered
 - https://bugzilla.mozilla.org/show_bug.cgi?id=1579743

Appendix B: Helpful Notes

- CMAKE is just used to generating the build files. When you run it, you do not need to run it again. To reopen the project, just open Visual Studio and click on **minetest.sln**