

Building wolfSSH on Windows:

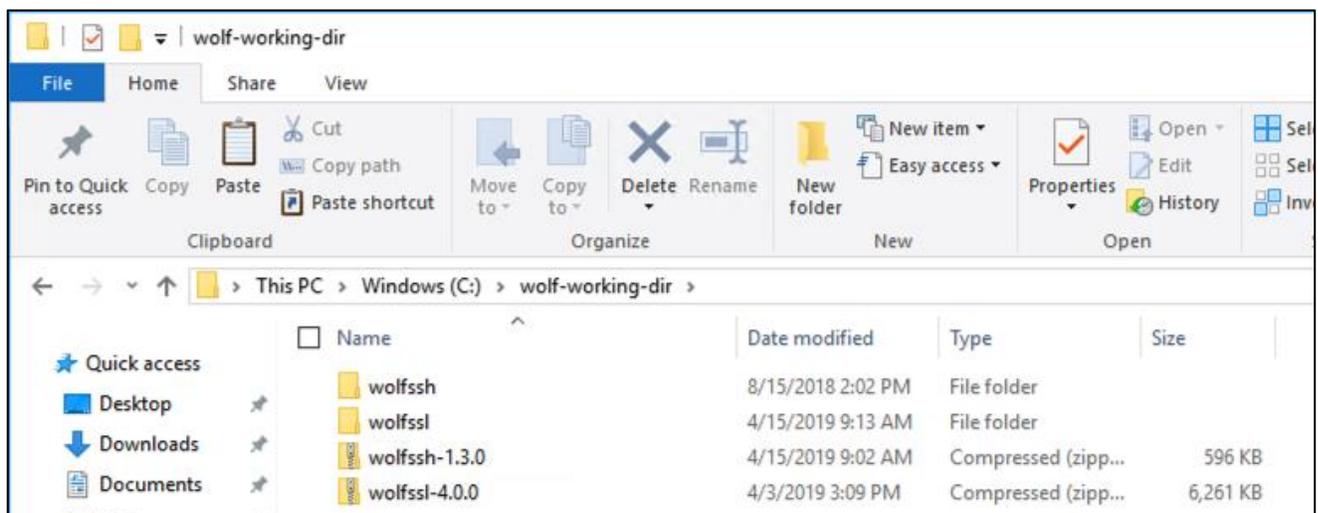
Step 1) Download wolfssl-4.0.0.zip (or later version) and wolfssh-1.3.0.zip (or later version) from the download page at <https://www.wolfssl.com/download/>

NOTE: From here forward this guide will simply refer to the current versions of wolfssl and wolfssh with x.x.x to denote current version IE wolfssl-x.x.x for wolfssl-4.0.0 at the time this guide was written.

Step 2) Make a directory, C:\wolf-working-dir (or whatever you wish to call it)

Step 3) Place both zip archives in this directory and unzip them so you end up with a directory setup as seen in the screenshot below:

CRITICAL NOTE! Rename both directories to simply wolfssl and wolfssh respectively from wolfssl-x.x.x/ and wolfssh-x.x.x/ once unzipped. This is critical for the include paths set in the wolfSSH Visual Studio project or it will not be able to find the wolfSSL headers out of the box. You can alternatively update the project include paths if you wish to keep the versioning in the directory names but for starting out it is recommended to rename them so you have a directory setup like the below screen-shot before getting started:

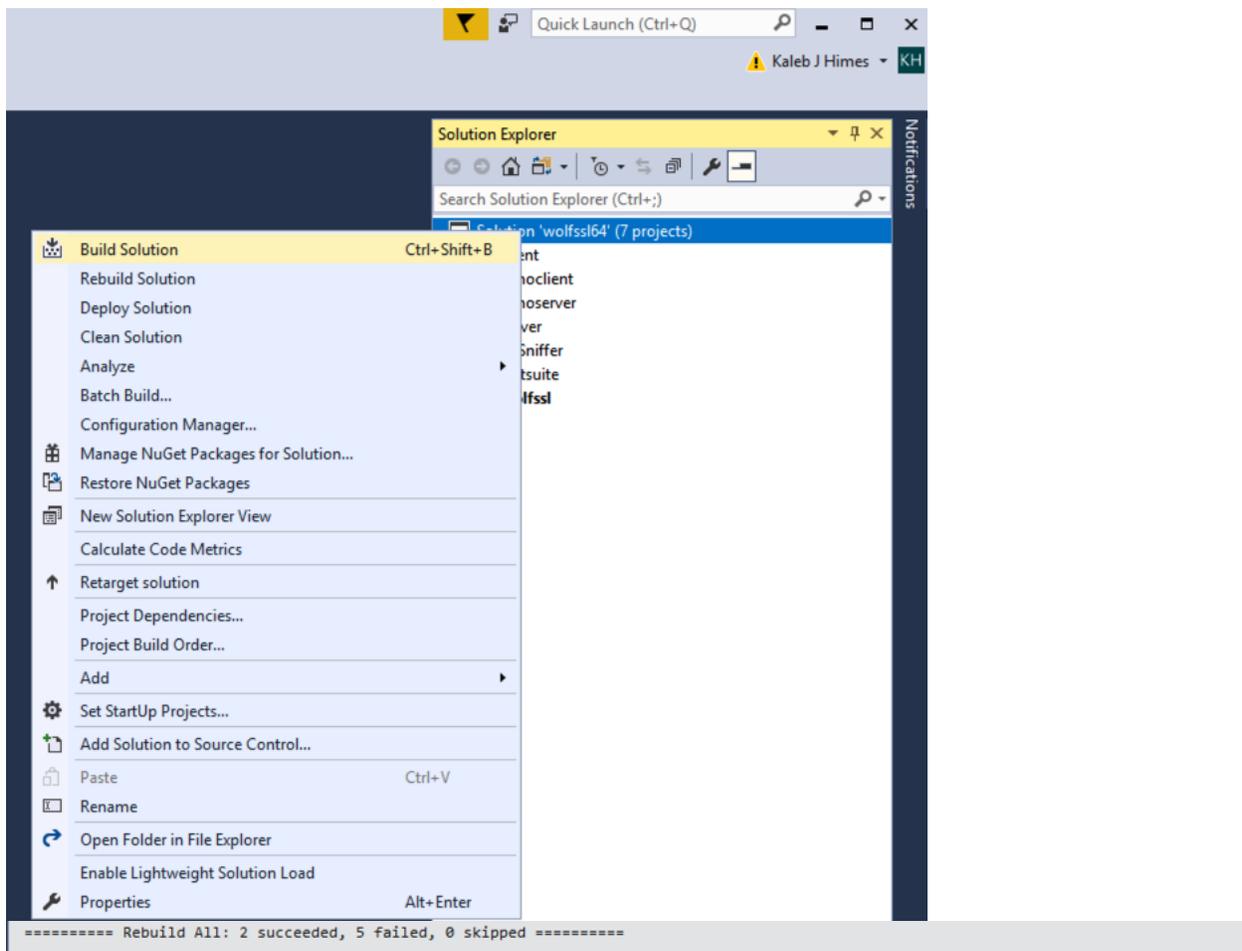


Step 4) Copy the file C:\wolf-working-dir\wolfssh\ide\winvs\user_settings.h

Step 5) Paste that file (and choose to replace the file in the destination) to C:\wolf-working-dir\wolfssl\IDE\WIN\user_settings.h

Step 6) Open the solution file in visual studio for wolfssl C:\wolf-working-dir\wolfssl\wolfssl64.sln NOTE: Do NOT use wolfssl.sln, specifically use the wolfssl64.sln. Allow Visual Studio to re-target the solution to your SDK (click accept or OK when the popup windows asks to re-target the solutions)

Step 7) In visual studio use the Solution explorer to right-click and select “Build Solution” (See screen shot below). You should see 2 succeed and 5 failed or 1 succeeded and 6 failed (depending on future configuration of the sniffer). This is expected since we are building for SSH and only using the wolfCrypt portion of the library the user_settings.h header that we copied over in Step 4 above turned off all TLS support. The examples are expected to fail while the library (at a minimum) should build successfully.



Now close and exit Visual Studio

Step 8) Browse to C:\wolf-working-dir\wolfssh\ide\winvs\ and double-click wolfssh.sln to open in Visual Studio. Again you will see a popup asking to retarget to your SDK, click “OK” to allow Visual Studio to automatically retarget for your version of the SDK that is installed on your system.

Step 9) Right click and select “Build Solution” from the Solution explorer window, same as was done when building wolfssl64.sln in step 7. You should see 6 succeeded, 0 failed in the output window in Visual studio.

If you have any questions or concerns with this guide please shoot an email to support@wolfssl.com or contact us via the Zendesk portal at <https://wolfssl.zendesk.com> anytime.

Thank you for evaluating wolfSSH and we look forward to hearing from you if you have any questions. Best of luck in all your testing!

- The wolfSSL Team