HomeBridge on Windows 10 64 Bit.

This guide is a work in progress. Additional clarifications may be added in the future. These instructions have been tested on Windows 10 64-bit, but may also work for earlier versions of windows.

I. Contents

I. Installing Homebridge

- There are a number of steps in this guide that say to use Windows Command Prompt. **Do NOT try to use Windows PowerShell** instead. The install will fail if Windows PowerShell is used instead of Command Prompt.

- You should perform all the installs identified in this guide from the Windows user account that you will use to run homebridge.

    - This is because, by default, certain HomeBridge modules are installed in the logged-in user's account at: C:\Users\ *_your user name_* \AppData\Roaming\npm\node_modules and will not be accessible from other Windows accounts. I've found it helpful to set up a specific Windows "user" just for the running of HomeBridge (e.g., set up a new user under the login name "HomeBridge" or something like that, and then do the installation when logged in as that user). This is particularly important if you want to auto-start HomeBridge at Windows Boot time.

- Once homebridge is installed, you will have to install plugins and then edit HomeBridge's config.json file for those plugins. On Windows 10, HomeBridge's config.json file should be placed in the following folder by default:

    C:\Users\ *_your user name_* \.homebridge

A. Install Bonjour SDK for Developers

Download Bonjour for Developers 3.0.0.10 from: https://developer.apple.com/bonjour/) then select the "Bonjour SDK for Windows" and then "Bonjour SDK for Windows v3.0" to install.

- You'll need to sign up for a free Apple developer account.

- Bonjour developer SDK sometimes doesn't set its environment variable correctly, so check its installation following instructions below. This is particularly relevant if you see an error saying like "Cannot open include file: 'dns_sd.h':" when you try to do the install at #8, below.

B. Install Nodejs 64-bit

These Windows 64 bit install instructions have been tested with Nodejs v8.9.3 so its recommended that you use that. Earlier version of NodeJS may work but haven't been tested.

Download Nodejs v8.9.3 64BIT from: https://nodejs.org/dist/v8.9.4/node-v8.9.4-x64.msi

- Leave all the defaults as-is when installing.

32-bit option: If installing on a 32 bit version of Windows, download and install: https://nodejs.org/dist/v8.9.4/node-v8.9.4-x86.msi instead of the 64-bit version.

C. Install Windows Build Tools

Open a Windows Command Prompt in Administrative mode. To do that, Click on the Windows Menu Bar start icon and start typing "cmd". When you see the application "Command Prompt" appear in the search results, right click on it and choose “Run as Administrator." Do NOT use Windows PowerShell for this -- for some reason, this step fails if you use PowerShell instead of Command Prompt.

From the Windows Command Prompt, execute the command:

- npm install -g windows-build-tools

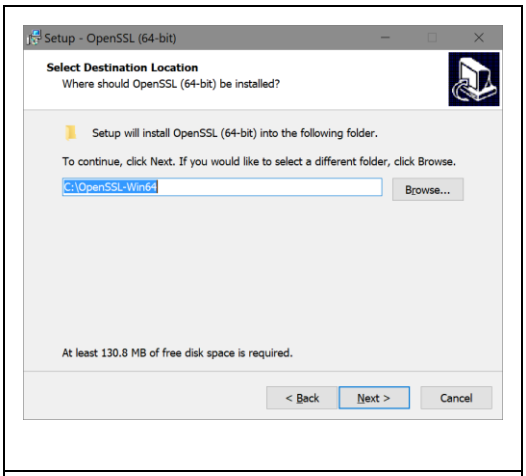
Be patient - this one takes a while. At times, it will look like nothing is happening or that it may be done installing. If you don’t see the command prompt, just continue to wait.

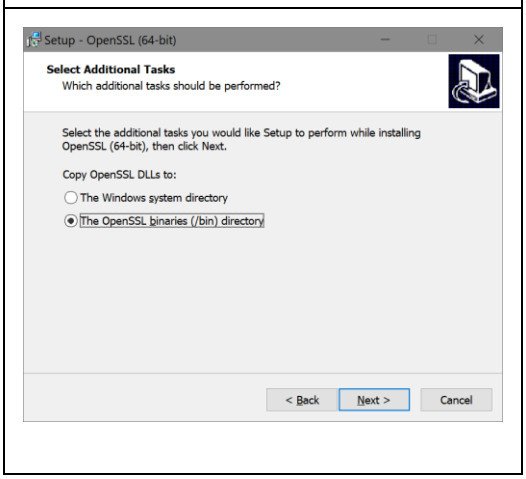
D. Install OpenSSL 64-bit

Download Open SSL 1.0.2n 64 BIT from: https://slproweb.com/download/Win64OpenSSL-1_0_2n.exe

32-bit option: If installing on a 32 bit version of Windows, download and install: http://slproweb.com/download/Win32OpenSSL-1_0_2n.exe instead of the 64-bit version.

- You **must** use the 1.0.x series of OpenSSL and **not use** the later 1.1.x series. This is because a library file has been renamed in the 1.1.x versions which causes compile errors during the installation of homebridge.





E. Install Homebridge

After the Windows Build tools install, then install HomeBridge from a **newly opened** Windows Command Prompt using the command:

- npm install -g homebridge

For this step, DO NOT re-use the command prompt that you used in step "C". If you re-use the same command prompt window, this step may fail!

You will likely see a lot of compiler warning messages (yellow text) during this installation. They can be ignored.

F. Test Homebridge

Open a new Windows Command Prompt and enter the command:

homebridge

You should see some text displayed and a QR code. If so, HomeBridge installed correctly.

II. Optional

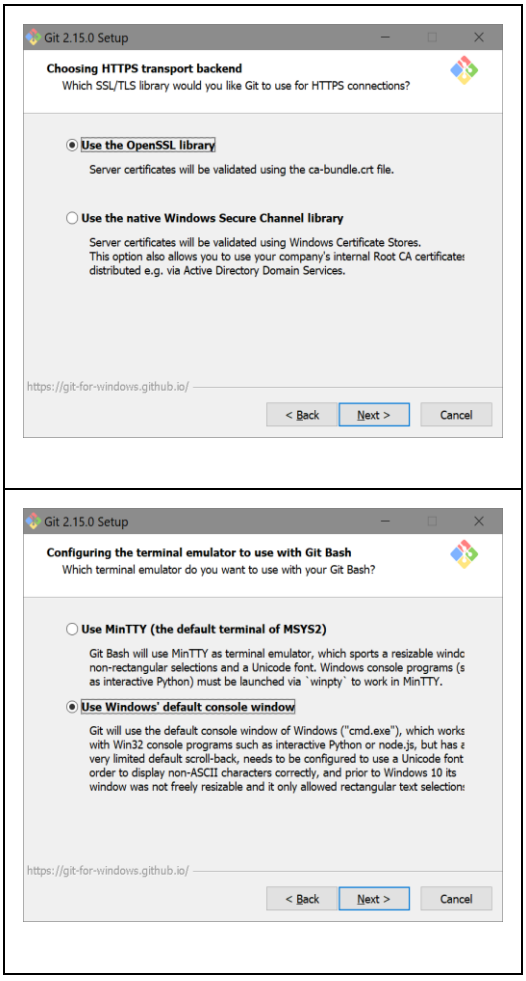
A. Install GIT 64-bit

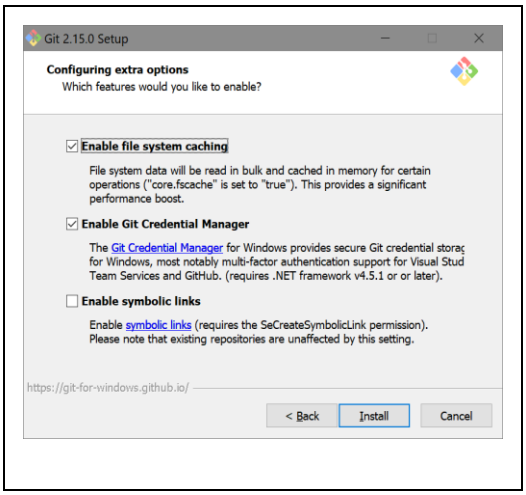
Git is needed if you want to install plugins directly from a github repository. Most users will not need to do this, but if you do

Download Git 2.15.1.2 64 bit from: https://github.com/git-for-windows/git/releases/download/v2.15.1.windows.2/Git-2.15.1.2-64-bit.exe

Then during it's install:

- Leave all defaults as-is except selected "use Windows' default console window" instead of MinTTY when asked.





III. Auto-Start Homebridge and Windows Startup Using Task Scheduler

** This section to be expanded (maybe, eventually!).

A. Create an Auto-Login Account

- To auto-start Homebridge at system startup, you need to first set up a Windows auto-login account
- First, add a new user using the conventional Windows account creation tools. For purposes of this explanation, it is assumed the added user is named “HomebridgeUser” set up as a local user.
- These instructions have been tested with the new “HomeBridgeUser” account having Administrative rights – setting up the account as a “Standard” user is currently untested.

B. Use NetplWiz to Auto-Login

- Then, use the netplzwiz command line tool to set up Windows to automatically login to “HomebridgeUser” at system startup.
- Instructions for doing so are explained here: https://www.lifewire.com/how-do-i-auto-login-to-windows-2626066

C. Use Task Scheduler to Start Homebridge

Then, you set up a task in Windows Task Scheduler with a Trigger set to "At log on" of the "Homebridge" user and set the "Start a program" actions to start homebridge.

** Additional clarification to be added **

Note that when a program is started with Task Scheduler, it is started with a below-normal priority. To reset the priority to “Normal”, follow the instructions here: https://bdbits.wordpress.com/2010/04/29/setting-a-scheduled-task-process-priority/

D. Auto-Lock the HomebridgeUser Account After Startup

You may want to include a second Task Scheduler task to immediately lock the HomebridgeUser account after homebridge startup. This will allow HomeBridge to run in its account, but returns to the login screen for "regular" use of the computer by others. To do this, set a second scheduled task that also Triggers "At log on" of the "HomebridgeUser" user with a "Start a program" action set as follows:

Program/Script: C:\Windows\System32\rundll32.exe Add arguments (optional): user32.dll,LockWorkStation Start in (optional): C:\Windows\System32

IV. Advice on Common Problems:

A. Using OpenSSL Versions 1.1.x instead of 1.0.x

The “ed25519” module (which is automatically installed during the “npm install -g homebridge) requires a file libeay32.lib which is normall at: C:\OpenSSL-Win64/lib/libeay32.lib. However, in Version 1.1.x of OpenSSL, this has been named libcrypto.lib which prevents installation of the ed25519 module.

- Solution (currently untested): make a copy of libcrypto.lib and rename to libeay32.lib so you now have both a libeay32.lib and libcrypto.lib file in C:\OpenSSL-Win64\lib\

B. Loss of Data Warning c42444 – Conversion from crypto_int64 to unsigned char

During installation of the "ed25519" module (which is automatically installed during the "npm install -g homebridge) stage, you may get numerous compiler warnings along the lines of:

```
..\src\ed25519\sc_muladd.c(367): warning C4244: '=': conversion from 'crypto_int64' to
'unsigned char', possible loss of data
[C:\Users\XXXXX\AppData\Roaming\npm\node_modules\ed25519\build\ed25519.vcxproj]
```

These warnings are generated by the MicroSoft VisualStudio compiler and can be ignored.

C. Bonjour SDK Is missing / Can't Open Include File "dns_sd.h"

If you get a "fatal error C1083: Cannot open include fine: 'dnssd.h'" message during the "npm -g install homebridge" stage (Section I, above), it may be a Bonjour SDK install error. Occasionally, the Bonjour Developer SDK fails to set its environment variable correctly. To check this:

1. From a Windows "cmd" window, enter the command "SET BONJOUR_SDK_HOME"
2. Windows should now display the BONJOUR_SDK_HOME environment variable showing the path to the BONJOUR developer SDK.
3. Does it match the path to where you have installed Bonjour? Typically, you should see the following (which is where the Bonjour Developer SDK should be installed): BONJOUR_SDK_HOME=C:\Program Files\Bonjour SDK\
4. If the path points to the "D:" drive or somewhere else, then you need to manually set/fix the environment variable.
5. To set / fix the variable, from the "Control Panel" search for "Environment". Then Click "Edit System Environment Variables"
6. The "System Properties" dialog box will appear with the "Advanced" tab selected. From that dialog, select the "Environmental Variables" button.
7. In the "System variables" section, edit / create the "BONJOUR_SDK_HOME" variable and set it to "C:\Program Files\Bonjour SDK"