**HomeBridge on Windows 10 64 Bit.**

This guide is a work in progress. Additional clarifications to be added to be added (expected update by Jan. 30, 2018).

Contents

[I. Installing Homebridge 1](#_Toc502124423)

[A. Install GIT 2.15.1.2 64BIT 1](#_Toc502124424)

[B. Install Visual Studio Community 2](#_Toc502124425)

[C. Install Python 2.7.14 3](#_Toc502124426)

[D. Install OpenSSL 1.0.2n 4](#_Toc502124427)

[E. Install Nodejs 5](#_Toc502124428)

[F. Install Bonjour SDK for Developers 5](#_Toc502124429)

[G. Install Windows Build Tools 5](#_Toc502124430)

[H. Install Homebridge 6](#_Toc502124431)

[II. Auto-Start Homebridge and Windows Startup Using Task Scheduler 6](#_Toc502124432)

[A. Create an Auto-Login Account 6](#_Toc502124433)

[B. Use NetplWiz to Auto-Login 6](#_Toc502124434)

[C. Use Task Scheduler to Start Homebridge 6](#_Toc502124435)

[D. Auto-Lock the HomebridgeUser Account After Startup 6](#_Toc502124436)

[III. Advice on Common Problems: 7](#_Toc502124437)

[A. Using OpenSSL Versions 1.1.x instead of 1.0.x 7](#_Toc502124438)

[B. Loss of Data Warning c42444 – Conversion from crypto\_int64 to unsigned char 7](#_Toc502124439)

[C. Bonjour SDK Is missing / Can’t Open Include File “dns\_sd.h” 7](#_Toc502124440)

# Installing Homebridge

## Install GIT 2.15.1.2 64BIT

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.

|  |
| --- |
|  |
|  |
|  |

## Install Visual Studio Community

Download Visual Studio Community 2017 from: <https://www.visualstudio.com/products/visual-studio-community-vs>

* Select “Desktop development with C++”, “Nodejs” development and “Python” development.

## Install Python 2.7.14

Download Python 2.7.14 64 BIT from: <https://www.python.org/ftp/python/2.7.14/python-2.7.14.amd64.msi>

* Select “Install for all users” and leave default installation directory as: “C:\Python27\”
* \*\* You need this even though “Python development” was selected for Visual Studio 2017!

|  |
| --- |
|  |
|  |
|  |
|  |

## Install OpenSSL 1.0.2n

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

Currently recommend that you stick with the 1.0.x series of OpenSSL and not use the later 1.1.x series. A library file has been renamed in the 1.1.x versions which causes compile errors during the installation of the referenced ed25519 module. But if you must use Open SSL 1.1.x (e.g., because you have it installed on your system for other purposes), then see instructions in Section III.A, below.

* In the “Copy OpenSSL DLLs” dialog box, select “The OpenSSL binaries (/bin) directory”

|  |
| --- |
|  |
|  |

## Install Nodejs

These Windows 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/v8.9.3/node-v8.9.3-x64.msi>

* Leave all the defaults alone when installing.

## 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](https://github.com/nfarina/homebridge/pull/8), below.

## Install Windows Build Tools

Right-click the Windows MenuBar start icon and choose “Command Prompt (Admin)”, then execute the command:

* npm install -g windows-build-tools

(Be patient - this one takes a while)

## Install Homebridge

After the Windows Build tools install, then install homebridge from the command shell using the command:

* npm install -g homebridge

# Auto-Start Homebridge and Windows Startup Using Task Scheduler

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

## 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.

## Use NetplWiz to Auto-Login

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

## 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 \*\*

## 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 |
|  |

# Advice on Common Problems:

## 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\

## 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.

## 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"