

# Build Domoticz from source

**Note: This guide will also work for most Debian based Linux systems.**

There are two ways to install Domoticz:

- [Using the precompiled/binary distribution](#)
- Compiling Domoticz from source (Not that difficult if you **strictly** follow this page)

## Contents

- [1 Preparation](#)
- [2 Compiling](#)
  - [2.1 Build & Install Boost Libraries](#)
  - [2.2 Build Support for OpenZWave 1.6+](#)
  - [2.3 Domoticz Source](#)
- [3 Make Domoticz startup at boot time \(Also needed for Web Update\)](#)
- [4 Updating Domoticz](#)
  - [4.1 Binary install update](#)
  - [4.2 Source Update](#)
  - [4.3 Source Update OpenZwave](#)
  - [4.4 Notes](#)
- [5 Additional information](#)

## Preparation

Make sure your system is been setup and have set to the correct date time !

For Ubuntu/Debian systems issue:

```
sudo dpkg-reconfigure tzdata
```

For Raspberry Pi Users:

Log in to your Raspberry with the "pi" user, open the terminal window, and issue:

```
sudo raspi-config
```

Using Raspi-config you can set your timezone, finish (reboot) and login again. You can find various tutorials on how to do this elsewhere on the internet, or you can check out the tutorial on this wiki: [Initial Raspberry Pi Setup](#)

# Compiling

*It's recommended to use GCC 6 or higher (we require `-std=c++14`)  
Mac OS X with clang++ / Xcode 5.1 (requires `-std=c++14 -stdlib=libc++`)*

```
sudo apt-get install make gcc g++ libssl-dev git libcurl4-gnutls-dev libusb-  
dev libsqlite3-dev python3-dev zlib1g-dev libcereal-dev liblua5.3-dev uthash-  
dev
```

**If your system has limited memory (Less than 2GB of memory), you (might) need to create a swap file for compilation.**

This can be done by following this tutorial: <https://digitizer.com/create-swap-file-ubuntu-linux/>

**(You do not need to enable the swap file on boot-time!)**

Change to your source folder (could be your home folder)

**You need CMake version 3.16.0 or higher**

To compile/install CMake issue the following commands:

**Remove any previous CMake installation**

```
sudo apt remove --purge --auto-remove cmake
```

**Go to the official CMake site and download the latest source:**

<http://www.cmake.org/download>

**The following will assume version 3.19.3 is the latest version**

```
wget https://github.com/Kitware/CMake/releases/download/v3.19.3/cmake-  
3.19.3.tar.gz  
tar -xzvf cmake-3.19.3.tar.gz  
rm cmake-3.19.3.tar.gz  
cd cmake-3.19.3  
./bootstrap  
make  
sudo make install  
cd ..  
rm -Rf cmake-3.19.3
```

**Verify you are on the correct version with**

```
cmake --version
```

*Results of `cmake --version`:*

```
cmake version 3.16.xx
```

CMake suite maintained and supported by Kitware ([kitware.com/cmake](http://kitware.com/cmake)).

## Build & Install Boost Libraries

If this is a clean system, continue at *Now build and install the Boost library*.

Its recommended that you use the latest Boost libraries - at the moment we verified version 1.75. For this we need to compile/install them by issuing the following commands: These instructions can also be used on a Linux system.

*[This first step is necessary because you will get linking errors if you don't remove the old Boost library]*

```
sudo apt remove --purge --auto-remove libboost-dev libboost-thread-dev
libboost-system-dev libboost-atomic-dev libboost-regex-dev libboost-chrono-
dev
```

*Now build and install the Boost library. (Be patience with the 'tar' command as this could take minutes!)*

```
mkdir boost
cd boost
wget
https://boostorg.jfrog.io/artifactory/main/release/1.76.0/source/boost\_1\_76\_0
.tar.gz
tar xzf boost_1_76_0.tar.gz
cd boost_1_76_0/
./bootstrap.sh
./b2 stage threading=multi link=static --with-thread --with-system
sudo ./b2 install threading=multi link=static --with-thread --with-system
cd ../../
sudo rm -Rf boost/
```

## Build Support for OpenZWave 1.6+

If you need support for ZWave, you need to compile open-zwave before compiling domoticz. Make sure to follow the below steps, you will end up with a folder layout like:

- open-zwave-read-only
- dev-domoticz

For the first time, you need to clone the code (copy) from GitHub into your system:

```
git clone https://github.com/domoticz/open-zwave open-zwave-read-only
```

Refresh the source and build OpenZWave  
Initial build:

RPI-3: 4m.30s  
RPI-4: 2m.25s  
Odroid HC1/2/XU4: 1m.49s

```
cd open-zwave-read-only
git pull
make
sudo make install
```

## Domoticz Source

For the first time, you need to checkout the source from GitHub onto your system:

```
git clone https://github.com/domoticz/domoticz.git dev-domoticz
```

### Building Domoticz:

Change to the Domoticz directory, refresh the source and and start building

```
cd dev-domoticz
git pull
cmake -DCMAKE_BUILD_TYPE=Release CMakeLists.txt
make
```

On a raspberry PI compilation could take a long time. Suggested is to use a swap file (see above) if you have less then 4GB of memory

Initial build:  
RPI-4: 28m.40s

# Make Domoticz startup at boot time (Also needed for Web Update)

```
sudo cp domoticz.sh /etc/init.d
sudo chmod +x /etc/init.d/domoticz.sh
sudo update-rc.d domoticz.sh defaults
```

*Edit the startup script and change the USERNAME, DAEMON and DAEMON\_ARGS parameters to reflect your current settings*

## Updating Domoticz

### Binary install update

Use the Web Interface to update Domoticz (Setup->Check for Updates)

If this fails for any reason (e.g. partial download giving segmentation fault) then you should manually update by:

**Stable:**

```
cd domoticz
./updaterelease
```

**Beta:**

```
cd domoticz
./updatebeta
```

## Source Update

Login to your Raspberry Pi with the 'pi' user and issue:

```
cd dev-domoticz
git pull
make
```

## Source Update OpenZwave

Repeat the steps of the OpenZwave build without the clone part

## Notes

Always create a backup of your Domoticz environment before updating! For Beta updates an automatic backup of the complete domoticz environment is created in `domoticz/backups/domoticz_backup_yyyymmdd_hhmmss.tar.gz` by the update script.

Always refresh the [browser cache](#) after an update!

# Additional information

(Alternative way is to make use of Vagrant, see the [Build Domoticz from Source with Vagrant](#) page)

Some extra information not directly needed to build domoticz from source but potential helpful to some can be found [in the Q&A](#)