

Open62541-1.3.pdf Setup Journal

I am trying to follow the [Open62541 documentation](#) to do a basic setup of an Open62541 server.

2.1 Building The Library

2.1.1 Building with CMake on Ubuntu or Debian

Skipped this section because I am using Windows.

2.1.2 Building with CMake on Windows

Here we explain the build process for Visual Studio (2013 or newer). To build with MinGW, just replace the compiler selection in the call to CMake.

- Download and install
 - Python 2.7.x (Python 3.x works as well): <https://python.org/downloads>
 - CMake: <http://www.cmake.org/cmake/resources/software.html>
 - Microsoft Visual Studio: <https://www.visualstudio.com/products/visual-studio-community-vs>

- I confirmed Python is installed (3.10.0)
- I confirmed Cmake is installed (3.26.0)
- I confirmed Visual Studio is installed (2019 16.11.24)

```
PS C:\> cmake --version
cmake version 3.26.0-rc4

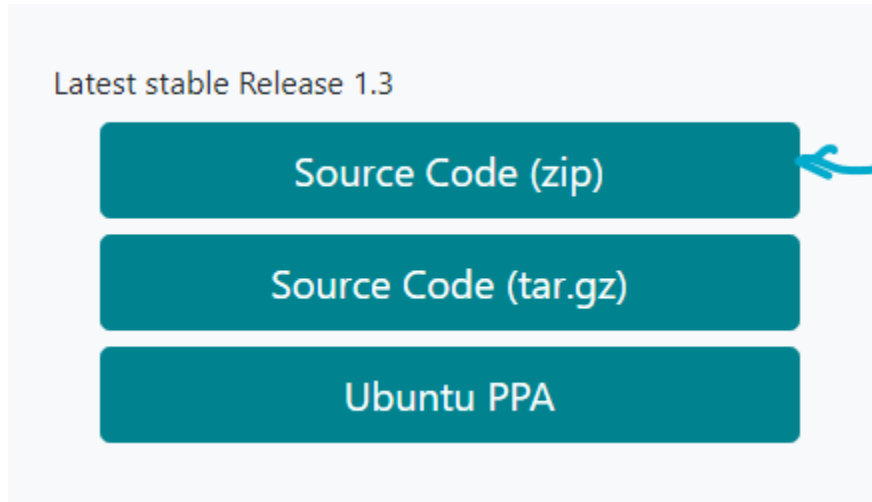
CMake suite maintained and supported by Kitware (kitware.com/cmake).
PS C:\> python --version
Python 3.10.0
```

Visual Studio

Microsoft Visual Studio Community 2019
Version 16.11.24
© 2021 Microsoft Corporation.
All rights reserved.
Installed products:

- Download the open62541 sources (using git or as a zipfile from github)

I downloaded the sources [from here using this link](#):



Extracted to C:\open62541-1.3\open62541-1.3 on my Windows 10 computer:

Name	Date modified	Type	Size
.azure-pipelines	4/4/2022 2:33 PM	File folder	
.github	4/4/2022 2:33 PM	File folder	
arch	4/4/2022 2:33 PM	File folder	
deps	4/4/2022 2:33 PM	File folder	
doc	4/4/2022 2:33 PM	File folder	
examples	4/4/2022 2:33 PM	File folder	
include	4/4/2022 2:33 PM	File folder	
plugins	4/4/2022 2:33 PM	File folder	
src	4/4/2022 2:33 PM	File folder	
tests	4/4/2022 2:33 PM	File folder	
tools	4/4/2022 2:33 PM	File folder	
.clang-format	4/4/2022 2:33 PM	CLANG-FORMAT ...	2 KB
.clang-tidy	4/4/2022 2:33 PM	CLANG-TIDY File	1 KB
.cquery	4/4/2022 2:33 PM	CQUERY File	1 KB
.dockerignore	4/4/2022 2:33 PM	DOCKERIGNORE F...	1 KB
.editorconfig	4/4/2022 2:33 PM	Editor Config Sour...	1 KB
.gitignore	4/4/2022 2:33 PM	Git Ignore Source ...	1 KB
.gitmodules	4/4/2022 2:33 PM	Text Document	1 KB
appveyor.yml	4/4/2022 2:33 PM	Yaml Source File	2 KB
AUTHORS	4/4/2022 2:33 PM	File	1 KB
azure-pipelines.yml	4/4/2022 2:33 PM	Yaml Source File	1 KB
CHANGELOG	4/4/2022 2:33 PM	File	6 KB
CMakeLists.txt	4/4/2022 2:33 PM	Text Source File	77 KB
CODE_OF_CONDUCT.md	4/4/2022 2:33 PM	Markdown Source...	4 KB
CONTRIBUTING.md	4/4/2022 2:33 PM	Markdown Source...	23 KB
cppcheck-suppressions.txt	4/4/2022 2:33 PM	Text Source File	1 KB
CPPLINT.cfg	4/4/2022 2:33 PM	CFG File	1 KB
FEATURES.md	4/4/2022 2:33 PM	Markdown Source...	10 KB
ISSUE_TEMPLATE.md	4/4/2022 2:33 PM	Markdown Source...	2 KB
LICENSE	4/4/2022 2:33 PM	File	17 KB
LICENSE-CC0	4/4/2022 2:33 PM	File	7 KB
README.md	4/4/2022 2:33 PM	Markdown Source...	16 KB

- Open a command shell (cmd) and run

```

cd <path-to>\open62541
mkdir build
cd build
<path-to>\cmake.exe .. -G "Visual Studio 14 2015"
:: You can use use cmake-gui for a graphical user-interface to select features

```

I ran the above commands with the following output:

```

PS C:\> cd c:\open62541-1.3\open62541-1.3
PS C:\open62541-1.3\open62541-1.3> mkdir build

Directory: C:\open62541-1.3\open62541-1.3

Mode                LastWriteTime         Length Name
----                -
d-----           2/25/2023   3:58 PM         build

PS C:\open62541-1.3\open62541-1.3> cd build
PS C:\open62541-1.3\open62541-1.3\build> cmake.exe .. -G "Visual Studio 16 2019"
-- Selecting Windows SDK version 10.0.19041.0 to target Windows 10.0.18363.
-- The C compiler identification is MSVC 19.29.30148.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: C:/Program Files (x86)/Microsoft Visual Studio/2019/Community/VC/Tools/MSVC/14.29.30133/bin/Hostx64/x64/cl.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Found Python3: C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe (found version "3.10.0") found components: Interpreter
-- Found Git: C:/Program Files/Git/cmd/git.exe (found version "2.29.2.windows.2")
fatal: not a git repository (or any of the parent directories): .git
CMake Warning at tools/cmake/SetGitBasedVersion.cmake:23 (message):
  Failed to determine the version from git information. Using defaults.
Call Stack (most recent call first):
  CMakeLists.txt:51 (set_open62541_version)

-- open62541 Version: v1.2.2-unknown
-- CMAKE_BUILD_TYPE not given; setting to 'Debug'
-- The selected architecture is: win32
-- Could NOT find Sphinx (missing: SPHINX_EXECUTABLE)
-- Could NOT find LATEX (missing: LATEX_COMPILER)
-- Configuring done (5.1s)
-- Generating done (0.5s)
-- Build files have been written to: C:/open62541-1.3/open62541-1.3/build
PS C:\open62541-1.3\open62541-1.3\build> |

```

It looks like it was overall successful, but I noted a few warnings.

- CMake Warning at tools/cmake/SetGitBasedVersion.cmake:23 (message):
 - Failed to determine the version from git information. Using defaults.
 - Call Stack (most recent call first):
 - CMakeLists.txt:51 (set_open62541_version)
- fatal: not a git repository (or any of the parent directories): .git
- -- Could NOT find Sphinx (missing: SPHINX_EXECUTABLE)
- -- Could NOT find LATEX (missing: LATEX_COMPILER)

2.1.3 - 2.1.6 Building on OS X, OpenBSD, Docker

Skipped these sections because I am using Windows.

2.2 Build Options

```

PS C:\open62541-1.3\open62541-1.3\build> cmake-gui.exe .. -G "Visual Studio 16 2019"

```

Where is the source code:

Preset:

Where to build the binaries:

Search: Grouped Advanced

Name	Value
BUILD_SHARED_LIBS	<input type="checkbox"/>
CLANG_FORMAT_EXE	CLANG_FORMAT_EXE-NOTFOUND
CMAKE_BUILD_TYPE	Debug
CMAKE_CONFIGURATION_TYPES	Debug;Release;MinSizeRel;RelWithDebInfo
CMAKE_INSTALL_PREFIX	C:/Program Files (x86)/open62541
UA_ARCHITECTURE	win32
UA_BUILD_EXAMPLES	<input type="checkbox"/>
UA_BUILD_TOOLS	<input type="checkbox"/>
UA_BUILD_UNIT_TESTS	<input type="checkbox"/>
UA_ENABLE_AMALGAMATION	<input type="checkbox"/>
UA_ENABLE_DA	<input checked="" type="checkbox"/>
UA_ENABLE_DIAGNOSTICS	<input type="checkbox"/>
UA_ENABLE_DISCOVERY	<input type="checkbox"/>
UA_ENABLE_ENCRYPTION	OFF
UA_ENABLE_ENCRYPTION_TPM2	OFF
UA_ENABLE_HISTORIZING	<input type="checkbox"/>
UA_ENABLE_JSON_ENCODING	<input type="checkbox"/>
UA_ENABLE_METHODCALLS	<input checked="" type="checkbox"/>
UA_ENABLE_PUBSUB	<input type="checkbox"/>
UA_ENABLE_PUBSUB_DELTAFRAMES	<input type="checkbox"/>
UA_ENABLE_PUBSUB_ETH_UADP	<input type="checkbox"/>
UA_ENABLE_PUBSUB_INFORMATIONMODEL	<input type="checkbox"/>
UA_ENABLE_PUBSUB_INFORMATIONMODEL_ME...	<input type="checkbox"/>
UA_ENABLE_SUBSCRIPTIONS	<input checked="" type="checkbox"/>
UA_ENABLE_SUBSCRIPTIONS_EVENTS	<input checked="" type="checkbox"/>
UA_FORCE_WERROR	<input checked="" type="checkbox"/>
UA_LOGLEVEL	300
UA_MULTITHREADING	0
UA_NAMESPACE_ZERO	REDUCED

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 16 2019

I left all the options at the default and hit "Configure", and then "Generate":

File Tools Options Help

Where is the source code:

Preset:

Where to build the binaries:

Search: Grouped Advanced

Name	Value
BUILD_SHARED_LIBS	<input type="checkbox"/>
CLANG_FORMAT_EXE	CLANG_FORMAT_EXE-NOTFOUND
CMAKE_BUILD_TYPE	Debug
CMAKE_CONFIGURATION_TYPES	Debug;Release;MinSizeRel;RelWithDebInfo
CMAKE_INSTALL_PREFIX	C:/Program Files (x86)/open62541
UA_ARCHITECTURE	win32
UA_BUILD_EXAMPLES	<input type="checkbox"/>
UA_BUILD_TOOLS	<input type="checkbox"/>
UA_BUILD_UNIT_TESTS	<input type="checkbox"/>
UA_ENABLE_AMALGAMATION	<input type="checkbox"/>
UA_ENABLE_DA	<input checked="" type="checkbox"/>
UA_ENABLE_DIAGNOSTICS	<input type="checkbox"/>
UA_ENABLE_DISCOVERY	<input type="checkbox"/>
UA_ENABLE_ENCRYPTION	OFF
UA_ENABLE_ENCRYPTION_TPM2	OFF
UA_ENABLE_HISTORIZING	<input type="checkbox"/>
UA_ENABLE_JSON_ENCODING	<input type="checkbox"/>
UA_ENABLE_METHODCALLS	<input checked="" type="checkbox"/>
UA_ENABLE_PUBSUB	<input type="checkbox"/>
UA_ENABLE_PUBSUB_DELTAFRAMES	<input type="checkbox"/>
UA_ENABLE_PUBSUB_ETH_UADP	<input type="checkbox"/>
UA_ENABLE_PUBSUB_INFORMATIONMODEL	<input type="checkbox"/>
UA_ENABLE_PUBSUB_INFORMATIONMODEL_ME...	<input type="checkbox"/>
UA_ENABLE_SUBSCRIPTIONS	<input checked="" type="checkbox"/>
UA_ENABLE_SUBSCRIPTIONS_EVENTS	<input checked="" type="checkbox"/>
UA_FORCE_WERROR	<input checked="" type="checkbox"/>
UA_LOGLEVEL	300
UA_MULTITHREADING	0
UA_NAMESPACE_ZERO	REDUCED

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 16 2019

```

Selecting Windows SDK version 10.0.19041.0 to target Windows 10.0.18363.
fatal: not a git repository (or any of the parent directories): .git

CMake Warning at tools/cmake/SetGitBasedVersion.cmake:23 (message):
  Failed to determine the version from git information. Using defaults.
Call Stack (most recent call first):
  CMakeLists.txt:51 (set_open62541_version)

open62541 Version: v1.2.2-unknown
The selected architecture is: win32
Could NOT find Sphinx (missing: SPHINX_EXECUTABLE)
Could NOT find LATEX (missing: LATEX_COMPILER)
Configuring done (1.5s)
    
```

As you can see it outputs the same warnings as noted above. Then I clicked Generate and it output the following:

```
| Generating done (1.5s)
```

2.2.1 - 2.1.7 Build Options

I skipped these sections since I am just sticking with the default options for now just to get it working.

2.3 Building the Examples

Make sure that you can build the shared library as explained in the previous steps. Even easier way to build the examples is to install `open62541` in your operating system (see *Installing open62541*).

Then the compiler should automatically find the includes and the shared library.

```
cp /path-to/examples/tutorial_server_firststeps.c . # copy the example server
gcc -std=c99 -o server tutorial_server_firststeps.c -lopen62541
```

The way this is worded leaves me feeling unclear.

The phrase “easier way” sounds to me like you’re about to propose an easier way to “build the shared library as explained in the previous steps”. In which case I would respond, “cool, but I already built the shared library in the previous steps, so I don’t need an easier way to do that.”

Did you mean to say this?:

“Now that you’ve built the library in the previous steps, you need to link it into your c project using gcc. The hard way is to link it manually using compiler options. The easy way is to install `open62541` in your operating system, so that the compiler will automatically find the includes and the shared library. Once you have installed `open62541` in your operating system, you can easily build the `tutorial_server_firststeps` example using the following commands:”

It sounds like I need to jump forward to 3.1, “install `open62541` in [my] operating system” and then jump back to 2.3 and try building the example.

3.1 Manual Installation

You can install open62541 using the well known *make install* command. This allows you to use pre-built libraries and headers for your own project.

I am new to *make install*, let's give it a try.

To override the default installation directory use `cmake -DCMAKE_INSTALL_PREFIX=/some/path`. Based on the SDK Features you selected, as described in [Build Options](#), these features will also be included in the installation. Thus we recommend to enable as many non-experimental features as possible for the installed binary.

I'm just going to stick with the defaults to increase my chance of getting this working.

The recommended cmake options for a default installation are:

```
git submodule update --init --recursive
mkdir build && cd build
cmake -DBUILD_SHARED_LIBS=ON -DCMAKE_BUILD_TYPE=RelWithDebInfo -DUA_NAMESPACE_
→ZERO=FULL ..
make
sudo make install
```

I don't know where these commands are supposed to be run. Should this be run in the root directory of the open62541 source code?

Why does it say to make a build directory if we already made one in 2.1.2?

I'll try running it in the root directory:

```
PS C:\open62541-1.3\open62541-1.3> git submodule update --init --recursive
fatal: not a git repository (or any of the parent directories): .git
```

As expected, the git command does nothing, since I downloaded a .zip of the project. 2.1.2 says I could download the source code as .zip, so I'm assuming that git cloning is not required.

```
PS C:\open62541-1.3\open62541-1.3> mkdir build
mkdir : An item with the specified name C:\open62541-1.3\open62541-1.3\build already exists.
At line:1 char:1
+ mkdir build
+ ~~~~~
+ CategoryInfo          : ResourceExists: (C:\open62541-1.3\open62541-1.3\build:String) [New-Item], IOException
+ FullyQualifiedErrorId : DirectoryExist,Microsoft.PowerShell.Commands.NewItemCommand

PS C:\open62541-1.3\open62541-1.3> cd build
PS C:\open62541-1.3\open62541-1.3\build> |
```

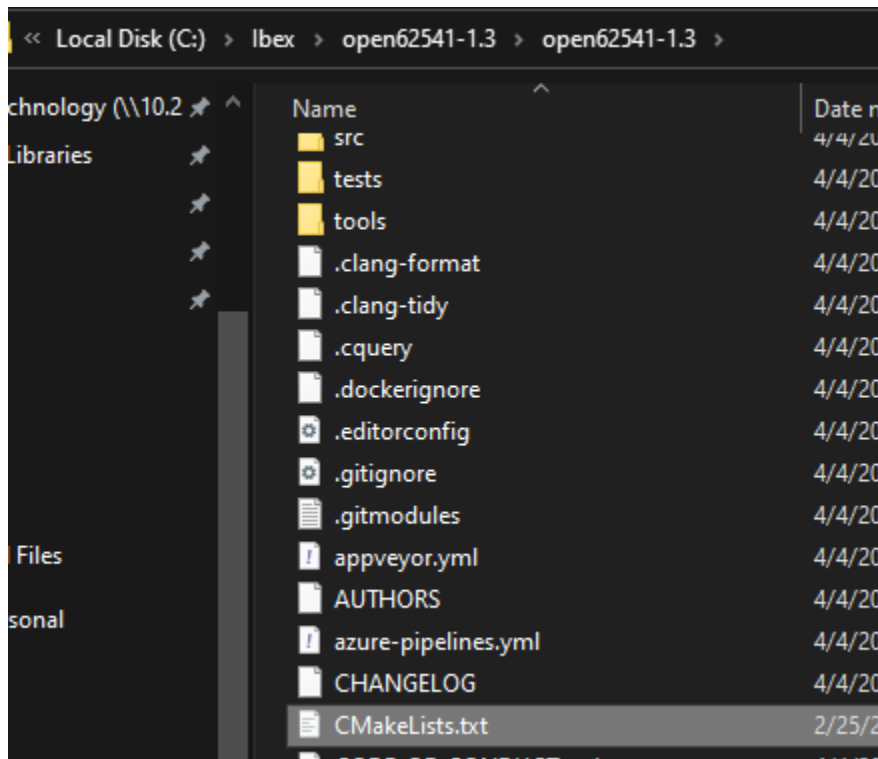
As expected, mkdir build fails because we already made a build directory in 2.1.2. I did it anyway just to show that I am trying to follow the documentation as exactly as possible.

Now to try the cmake command:

```
PS C:\open62541-1.3\open62541-1.3\build> cmake -DBUILD_SHARED_LIBS=ON -DCMAKE_BUILD_TYPE=RelWithDebInfo -DUA_NAMESPACE_Z
ERO=FULL
CMake Warning:
  No source or binary directory provided. Both will be assumed to be the
  same as the current working directory, but note that this warning will
  become a fatal error in future CMake releases.

CMake Error: The source directory "C:/open62541-1.3/open62541-1.3/build" does not appear to contain CMakeLists.txt.
Specify --help for usage, or press the help button on the CMake GUI.
```

Looks like it failed, and it needs to be run in a directory that contains CMakeLists.txt. Now I am even more confused why the previous two commands said to make a build folder and run cmake there, since a newly created folder is obviously to be empty, and cmake will therefore fail since there is no CMakeLists.txt file in an empty folder.



Looks like the root folder of the source code has the CMakeLists.txt that cmake is looking for, lets try running cmake there:

```

PS C:\open62541-1.3\open62541-1.3\build> cd ..
PS C:\open62541-1.3\open62541-1.3> cmake -DBUILD_SHARED_LIBS=ON -DCMAKE_BUILD_TYPE=RelWithDebInfo -DUA_NAMESPACE_ZERO=FULL
CMake Warning:
  No source or binary directory provided. Both will be assumed to be the
  same as the current working directory, but note that this warning will
  become a fatal error in future CMake releases.

-- Building for: Visual Studio 16 2019
-- Selecting Windows SDK version 10.0.19041.0 to target Windows 10.0.18363.
-- The C compiler identification is MSVC 19.29.30148.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: C:/Program Files (x86)/Microsoft Visual Studio/2019/Community/VC/Tools/MSVC/14.29.30133
/bin/Hostx64/x64/cl.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Found Python3: C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe (found version "3.10.0") found compon
ents: Interpreter
-- Found Git: C:/Program Files/Git/cmd/git.exe (found version "2.29.2.windows.2")
fatal: not a git repository (or any of the parent directories): .git
CMake Warning at tools/cmake/SetGitBasedVersion.cmake:23 (message):
  Failed to determine the version from git information. Using defaults.
Call Stack (most recent call first):
  CMakeLists.txt:51 (set_open62541_version)

-- open62541 Version: v1.2.2-unknown
-- The selected architecture is: win32
CMake Error at CMakeLists.txt:1170 (message):
  File
  C:/open62541-1.3/open62541-1.3/deps/ua-nodeset/Schema/Opc.Ua.NodeSet2.xml
  not found. You probably need to initialize the git submodule for
  deps/ua-nodeset or set open62541_NODESET_DIR.

-- Configuring incomplete, errors occurred!

```

It looks like the command failed with these warnings and errors:

- CMake Warning:
 - No source or binary directory provided. Both will be assumed to be the same as the current working directory, but note that this warning will become a fatal error in future CMake releases.
- fatal: not a git repository (or any of the parent directories): .git
 - Call Stack (most recent call first): CMakeLists.txt:51 (set_open62541_version)
- CMake Error at CMakeLists.txt:1170 (message):
 - File
 - C:/open62541-1.3/open62541-1.3/deps/ua-nodeset/Schema/Opc.Ua.NodeSet2.xml
 - not found. You probably need to initialize the git submodule for
 - deps/ua-nodeset or set open62541_NODESET_DIR.

Based on these warnings, it sounds like git clone is necessary after all, and downloading a .zip will not work. So I will try again starting with git clone instead of downloading a zip.

Cloned repo to fresh clean directory:

```
PS C:\open62541-1.3\open62541-1.3> cd C:\
PS C:\> mkdir open62541-1.3-git

Directory: C:\

Mode                LastWriteTime         Length Name
----                -
d-----           2/25/2023   5:16 PM         open62541-1.3-git

PS C:\> cd open62541-1.3-git
PS C:\open62541-1.3-git> git clone https://github.com/open62541/open62541.git
Cloning into 'open62541'...
remote: Enumerating objects: 66628, done.
remote: Counting objects: 100% (207/207), done.
remote: Compressing objects: 100% (128/128), done.
remote: Total 66628 (delta 89), reused 136 (delta 76), pack-reused 66421
Receiving objects: 100% (66628/66628), 49.13 MiB | 1.09 MiB/s, done.
Resolving deltas: 100% (49898/49898), done.
Updating files: 100% (2027/2027), done.
PS C:\open62541-1.3-git> |
```

Switched to latest stable release branch 1.3:

```
PS C:\open62541-1.3-git> dir

Directory: C:\open62541-1.3-git

Mode                LastWriteTime         Length Name
----                -
d-----           2/25/2023   5:18 PM         open62541

PS C:\open62541-1.3-git> cd open62541
PS C:\open62541-1.3-git\open62541> git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
PS C:\open62541-1.3-git\open62541> git checkout 1.3
Switched to a new branch '1.3'
Branch '1.3' set up to track remote branch '1.3' from 'origin'.
PS C:\open62541-1.3-git\open62541> git status
On branch 1.3
Your branch is up to date with 'origin/1.3'.

nothing to commit, working tree clean
```

Re-ran the commands from 2.1.2 to run cmake:


```
PS C:\open62541-1.3-git\open62541\build> cmake .. -G "Visual Studio 16 2019"
-- Selecting Windows SDK version 10.0.19041.0 to target Windows 10.0.18363.
-- open62541 Version: v1.3.5
-- The selected architecture is: win32
-- Found Sphinx: C:/Users/User/AppData/Local/Programs/Python/Python310/Scripts/sphinx-build.exe
-- Could NOT find LATEX (missing: LATEX_COMPILER)
-- Configuring done (1.4s)
-- Generating done (1.4s)
-- Build files have been written to: C:/open62541-1.3-git/open62541/build
```

Ok great, let's try fixing the missing LATEX_COMPILER. There is a [Python library called LaTeXCompiler](#).

```
PS C:\open62541-1.3-git\open62541\build> pip install LaTeXCompiler
Collecting LaTeXCompiler
  Downloading LaTeXCompiler-1.0.tar.gz (10 kB)
  Preparing metadata (setup.py) ... done
Collecting tqdm
  Downloading tqdm-4.64.1-py2.py3-none-any.whl (78 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.5/78.5 kB 483.9 kB/s eta 0:00:00
```

However installing it did not resolve the warning:

```
Successfully installed LaTeXCompiler-1.0 tqdm-4.64.1
PS C:\open62541-1.3-git\open62541\build> cmake .. -G "Visual Studio 16 2019"
-- Selecting Windows SDK version 10.0.19041.0 to target Windows 10.0.18363.
-- open62541 Version: v1.3.5
-- The selected architecture is: win32
-- Could NOT find LATEX (missing: LATEX_COMPILER)
-- Configuring done (1.3s)
-- Generating done (1.3s)
-- Build files have been written to: C:/open62541-1.3-git/open62541/build
```

I also tried installing python libraries PyLatex and Latex, those did not resolve the warning. So I am at a dead end with that warning and will move on to the next step.

Ran Cmake GUI with default options:

Where is the source code:

Preset:

Where to build the binaries:

Search: Grouped Advanced

Name	Value
BUILD_SHARED_LIBS	<input type="checkbox"/>
CLANG_FORMAT_EXE	CLANG_FORMAT_EXE-NOTFOUND
CMAKE_BUILD_TYPE	Debug
CMAKE_CONFIGURATION_TYPES	Debug;Release;MinSizeRel;RelWithDebInfo
CMAKE_INSTALL_PREFIX	C:/Program Files (x86)/open62541
UA_ARCHITECTURE	win32
UA_BUILD_EXAMPLES	<input type="checkbox"/>
UA_BUILD_TOOLS	<input type="checkbox"/>
UA_BUILD_UNIT_TESTS	<input type="checkbox"/>
UA_ENABLE_AMALGAMATION	<input type="checkbox"/>
UA_ENABLE_DA	<input checked="" type="checkbox"/>
UA_ENABLE_DIAGNOSTICS	<input type="checkbox"/>
UA_ENABLE_DISCOVERY	<input type="checkbox"/>
UA_ENABLE_ENCRYPTION	OFF
UA_ENABLE_ENCRYPTION_TPM2	OFF
UA_ENABLE_HISTORIZING	<input type="checkbox"/>
UA_ENABLE_JSON_ENCODING	<input type="checkbox"/>
UA_ENABLE_METHODCALLS	<input checked="" type="checkbox"/>
UA_ENABLE_PUBSUB	<input type="checkbox"/>
UA_ENABLE_PUBSUB_DELTAFRAMES	<input type="checkbox"/>
UA_ENABLE_PUBSUB_ETH_UADP	<input type="checkbox"/>
UA_ENABLE_PUBSUB_INFORMATIONMODEL	<input type="checkbox"/>
UA_ENABLE_PUBSUB_INFORMATIONMODEL_ME...	<input type="checkbox"/>
UA_ENABLE_SUBSCRIPTIONS	<input checked="" type="checkbox"/>
UA_ENABLE_SUBSCRIPTIONS_EVENTS	<input checked="" type="checkbox"/>
UA_FORCE_WERROR	<input checked="" type="checkbox"/>
UA_LOGLEVEL	300
UA_MULTITHREADING	0
UA_NAMESPACE_ZERO	REDUCED

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 16 2019

```

Selecting Windows SDK version 10.0.19041.0 to target Windows 10.0.18363.
open62541 Version: v1.3.5
The selected architecture is: win32
Could NOT find LATEX (missing: LATEX_COMPILER)
Configuring done (1.4s)
Generating done (0.9s)
    
```


OK now we're back to 3.1, now with a cloned git repo and the Sphinx library. Let's see if anything works better.

The recommended cmake options for a default installation are:

```
git submodule update --init --recursive
mkdir build && cd build
cmake -DBUILD_SHARED_LIBS=ON -DCMAKE_BUILD_TYPE=RelWithDebInfo -DUA_NAMESPACE_
  ↳ZERO=FULL ..
make
sudo make install
```

First try the git command:

```
PS C:\open62541-1.3-git\open62541\build> git submodule update --init --recursive
Submodule 'deps/mdnsd' (https://github.com/Pro/mdnsd.git) registered for path '../deps/mdnsd'
Submodule 'deps/mqtt-c' (https://github.com/LiamBindle/MQTT-C.git) registered for path '../deps/mqtt-c'
Submodule 'deps/ua-nodeset' (https://github.com/OPCFoundation/UA-Nodeset) registered for path '../deps/ua-nodeset'
Cloning into 'C:/open62541-1.3-git/open62541/deps/mdnsd'...
Cloning into 'C:/open62541-1.3-git/open62541/deps/mqtt-c'...
Cloning into 'C:/open62541-1.3-git/open62541/deps/ua-nodeset'...
Submodule path '../deps/mdnsd': checked out '3151afe5899dba5125dffa9f4cf3ae1fe2edc0f0'
Submodule path '../deps/mqtt-c': checked out 'f69ce1e7fd54f3b1834c9c9137ce0ec5d703cb4d'
Submodule path '../deps/ua-nodeset': checked out 'f71b3f411d5cb16097c3ae0c744f67ad45535fffb'
```

It looks like it successfully set up some submodule repos this time around.

Now the mkdir build command makes sense, because the git initialization cleared the build directory we made in step 2.1.2.

```
PS C:\open62541-1.3-git\open62541\build> mkdir build

Directory: C:\open62541-1.3-git\open62541\build

Mode                LastWriteTime         Length Name
----                -
d-----           2/25/2023   5:44 PM         build

PS C:\open62541-1.3-git\open62541\build> cd build
```

Great, the mkdir and cd commands worked. Let's try the next command cmake:

```
PS C:\open62541-1.3-git\open62541\build\build> cmake -DBUILD_SHARED_LIBS=ON -DCMAKE_BUILD_TYPE=RelWithDebInfo -DUA_NAMESPACE_ZERO=FULL ..
-- Selecting Windows SDK version 10.0.19041.0 to target Windows 10.0.18363.
-- open62541 Version: v1.3.5
-- The selected architecture is: win32
-- Could NOT find LATEX (missing: LATEX_COMPILER)
-- Configuring done (2.45s)
-- Generating done (0.8s)
-- Build files have been written to: C:/open62541-1.3-git/open62541/build
```

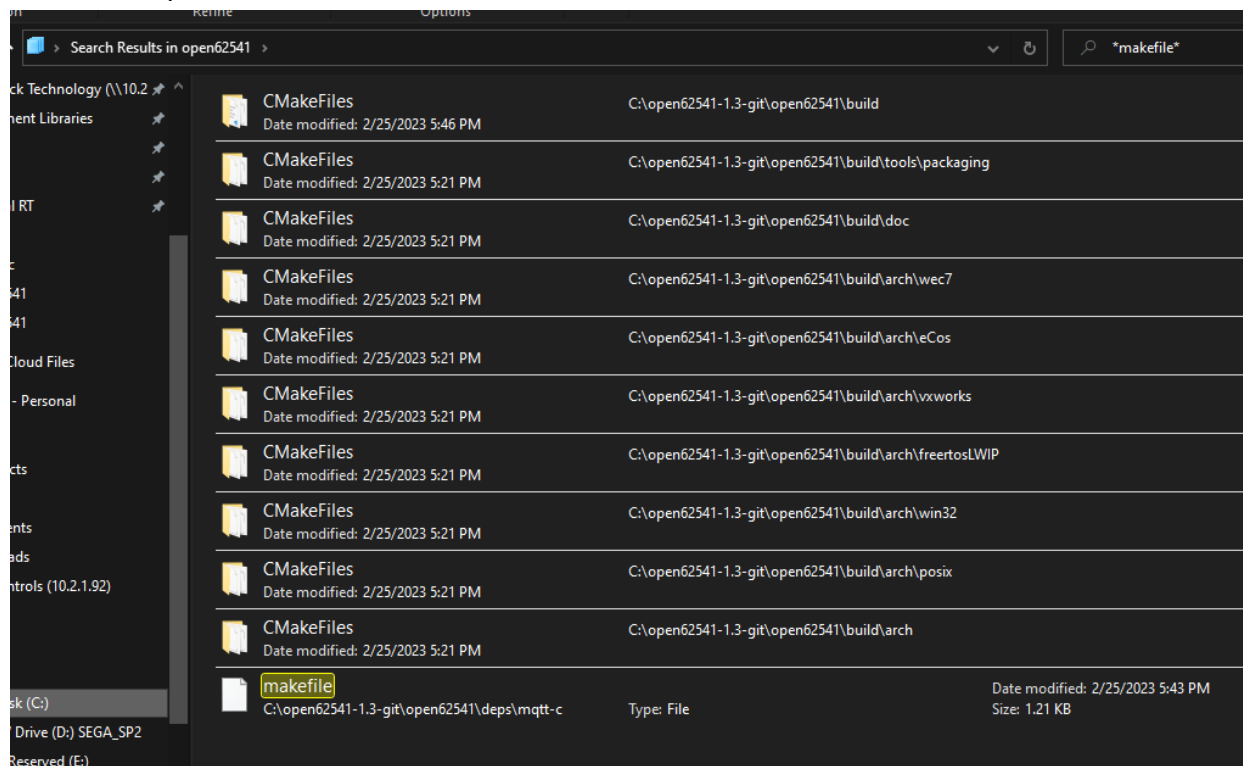
OK the cmake line worked this time!

Lets try the next command “make”:

```
PS C:\open62541-1.3-git\open62541\build\build> make
make.exe": *** No targets specified and no makefile found. Stop.
```

It fails.

Looks like the documentation forgot to navigate us to the location of the makefile. Let’s run a Windows Explorer search and see if we can find it ourselves:



Ok, looks like the only one is for a dependency, and its not in the build subfolder. That seems strange. Well lets try running make there:

```
PS C:\open62541-1.3-git\open62541\build\build> make
make.exe": *** No targets specified and no makefile found. Stop.
PS C:\open62541-1.3-git\open62541\build\build> cd ..
PS C:\open62541-1.3-git\open62541\build> make
make.exe": *** No targets specified and no makefile found. Stop.
PS C:\open62541-1.3-git\open62541\build> cd ..
PS C:\open62541-1.3-git\open62541> make
make.exe": *** No targets specified and no makefile found. Stop.
PS C:\open62541-1.3-git\open62541> cd deps
PS C:\open62541-1.3-git\open62541\deps> cd mqtt-c
PS C:\open62541-1.3-git\open62541\deps\mqtt-c> make
mkdir -p bin
gcc -Wextra -Wall -std=gnu99 -Iinclude -Wno-unused-parameter -Wno-unused-variable -Wno-duplicate-decl-specifier tests.c
src/mqtt.c src/mqtt_pal.c -lcmocka -lws2_32 -o bin/tests
tests.c:4:20: fatal error: cmocka.h: No such file or directory
#include <cmocka.h>
          ^
compilation terminated.
make.exe": *** [bin/tests] Error 1
```

It failed.

Well that is seeming like a dead end. So let's try the next command "make install" back in the build directory. I dropped the "sudo" because I'm on Windows:

```
PS C:\open62541-1.3-git\open62541\build> make install
make.exe": *** No rule to make target 'install'. Stop.
```

OK that failed too.

I see some files in the build directory that look relevant, like cmake_install.cmake and INSTALL.vcxproj. Since we are trying to "install" the libraries on our operating system, those seem like relevant files. But I'm not sure what to do with them.

Conclusion

I think this project is at a dead end for now. I can't proceed with 2.3 "Building the examples" until I "install open62541 in [my] operating system."