Adding a Library:
https://youtu.be/abuCXC3t6eQ

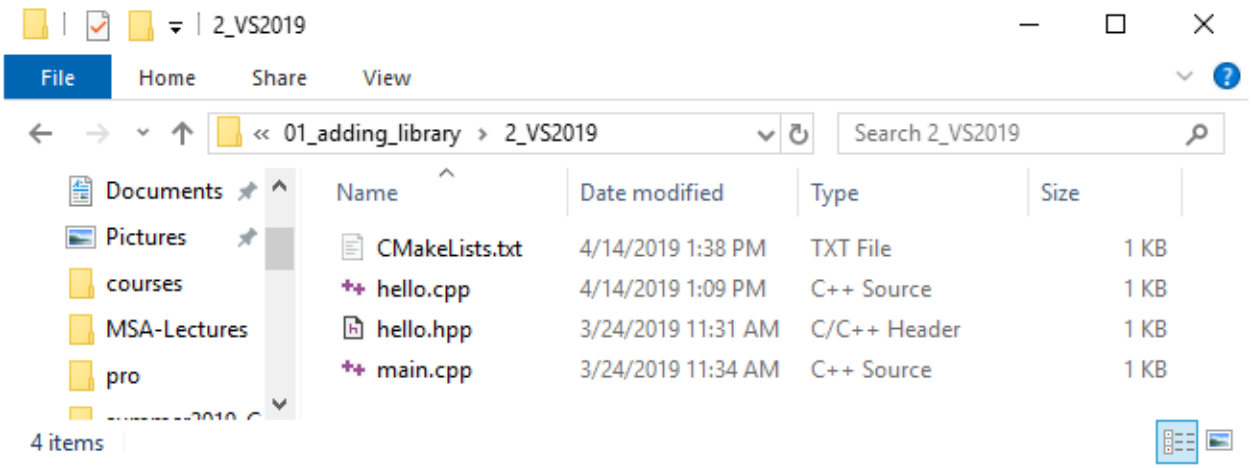
Demo of VS-CMake-integration:
https://youtu.be/OSvAeb99YcM?t=614

CMake Projects in VS
https://docs.microsoft.com/en-us/cpp/build/cmake-projects-in-visual-studio?view=vs-2019

VS-2019 CMake Project Settings UI
https://devblogs.microsoft.com/cppblog/introducing-the-new-cmake-project-settings-ui/

VS-Integration of CMake (Project Setup)
https://youtu.be/SYcTXK4q2Hg

- **Local** directory-path :
  - D:\dev\build\2_cmake\01_adding_library\2_VS2019



- **CMakeLists.txt:**

```
cmake_minimum_required(VERSION 3.10.2)
project(MyProject VERSION 1.0.0)

add_library(
    say-hello STATIC
    hello.hpp
    hello.cpp
)

# (<executable> <source>)
add_executable(josh main.cpp)
```

```
# (<executable-linking-into> <link-interface-mode> <name-of-library-linking-into-executable>)
target_link_libraries(josh PRIVATE say-hello)
```

- **main.cpp**

```
#include "hello.hpp"
int main()
{
	hello::say_hello();
}
```

- **hello.cpp**

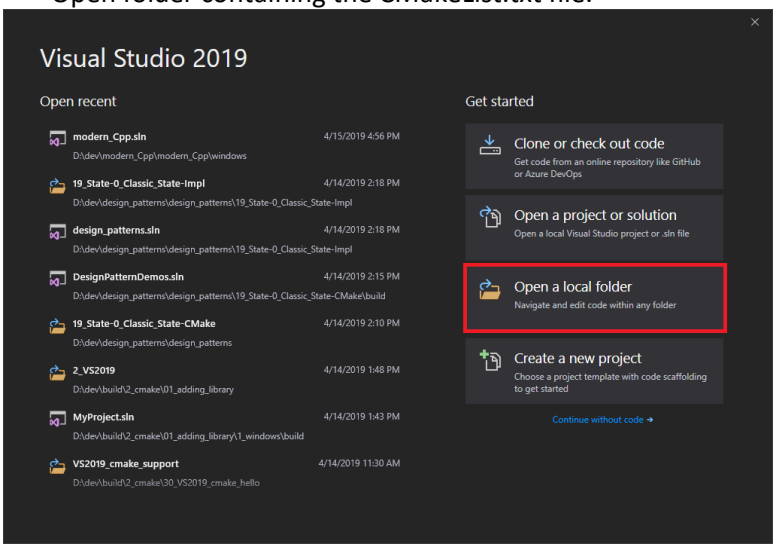
```
#include <iostream>
#include "hello.hpp"
void hello::say_hello()
{
	std::cout << "Hello\n";
}
```

- **hello.h**

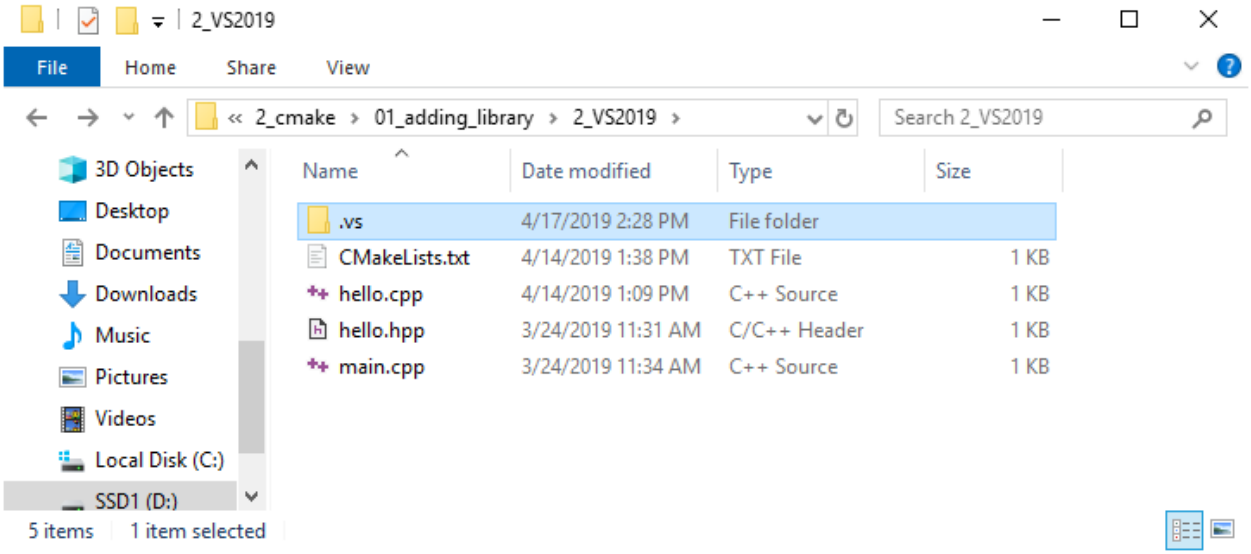
```
#ifndef HELLO_HPP_INCLUDED
#define HELLO_HPP_INCLUDED
namespace hello
{
	void say_hello();
}
#endif
```

Open folder containing CMakeLists.txt

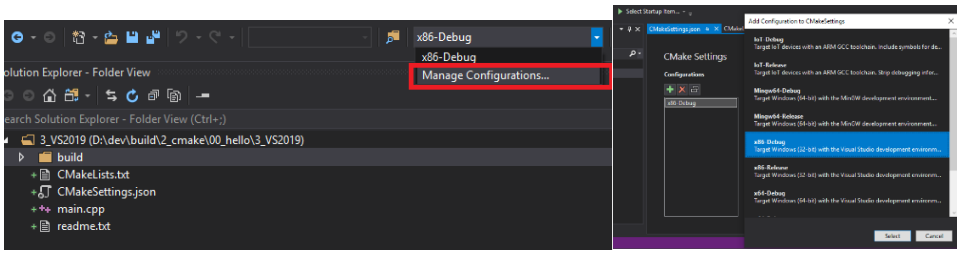
- Open folder containing the CMakeList.txt file.



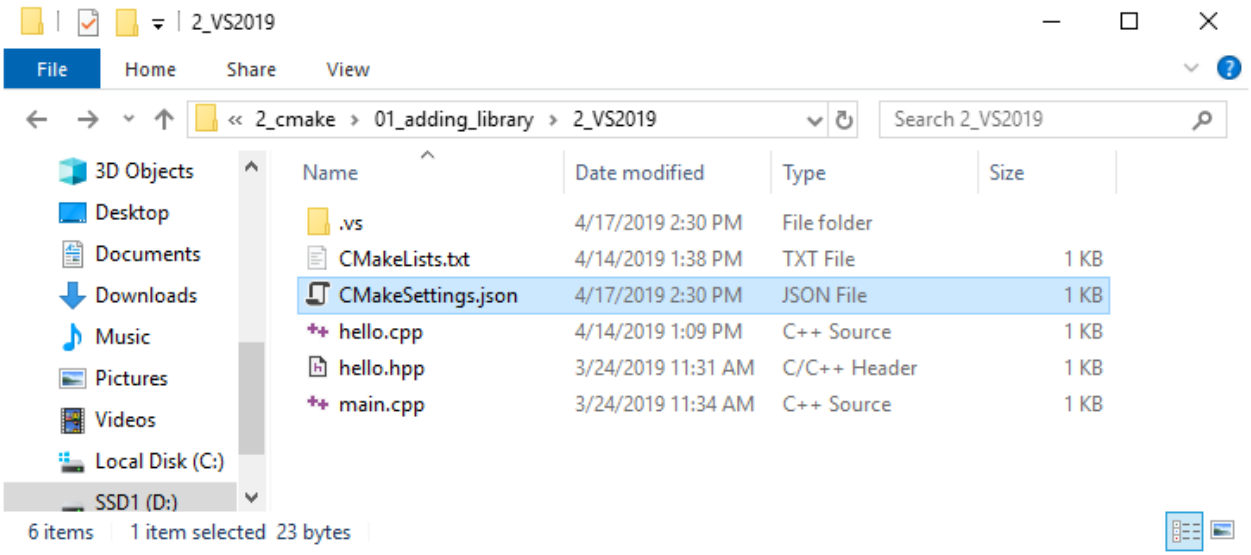
Opening a directory containing the CMakeLists.txt file results in the .vs folder being created:



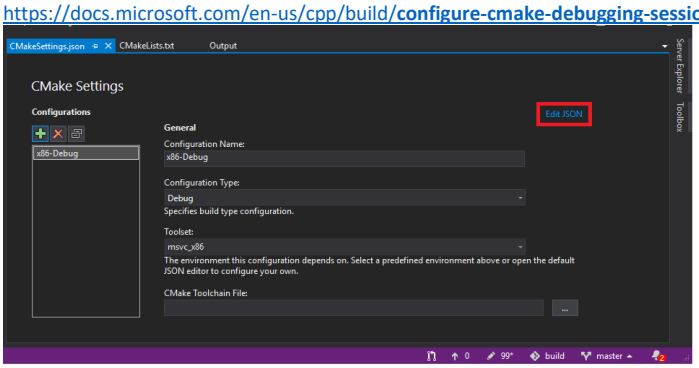
Change to x86-debug:



By selecting "Manage Connections" the CMakeSettings.json file is created:



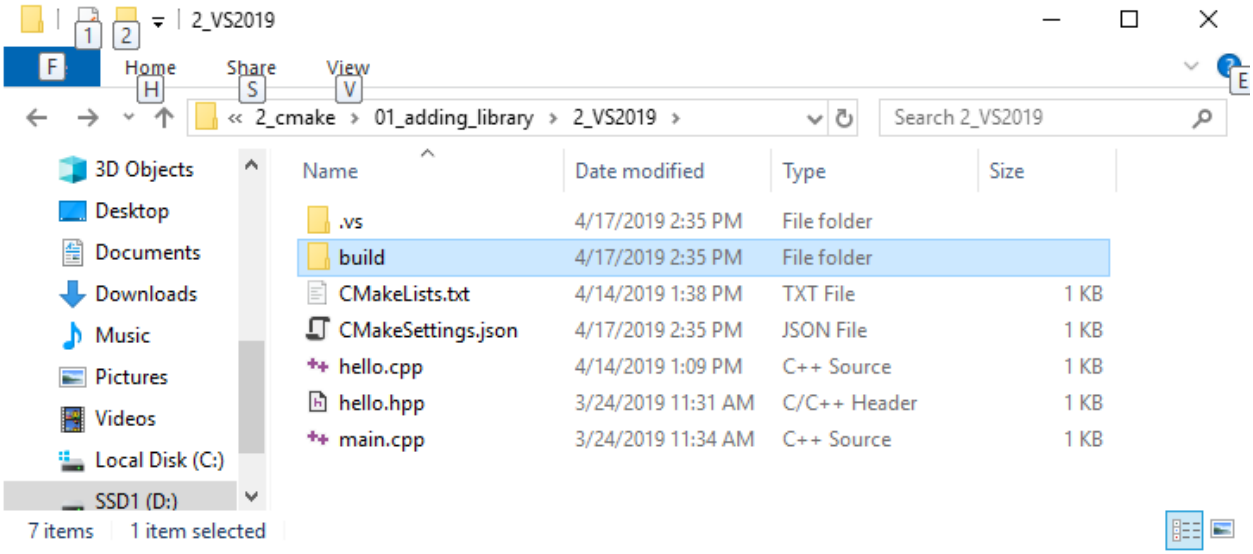
Edit the JSON file:
https://docs.microsoft.com/en-us/cpp/build/**configure-cmake-debugging-sessions**?view=vs-2019

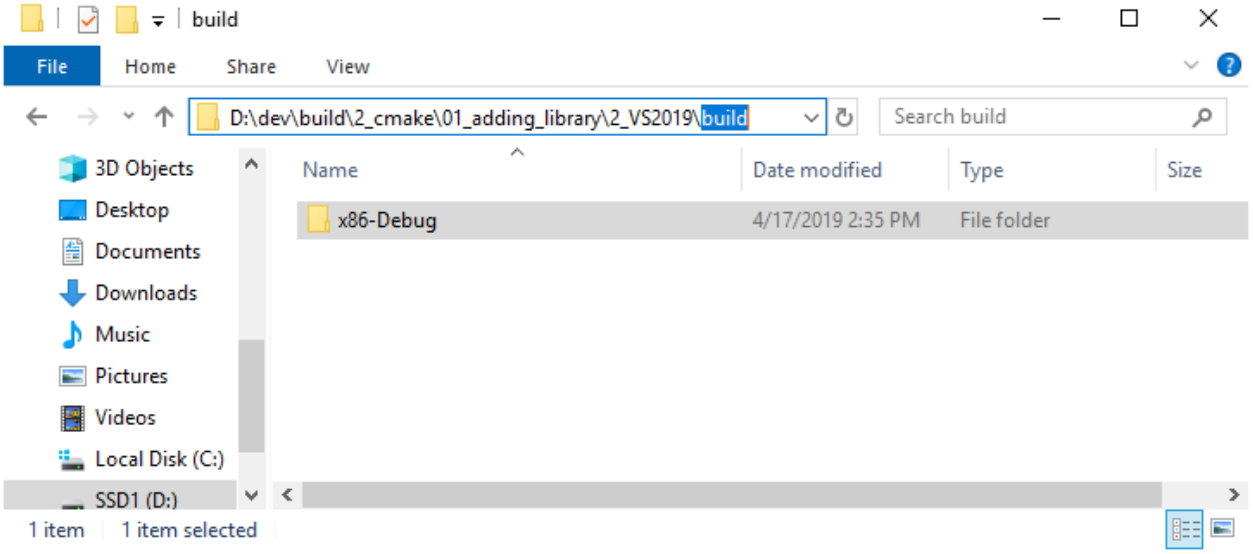


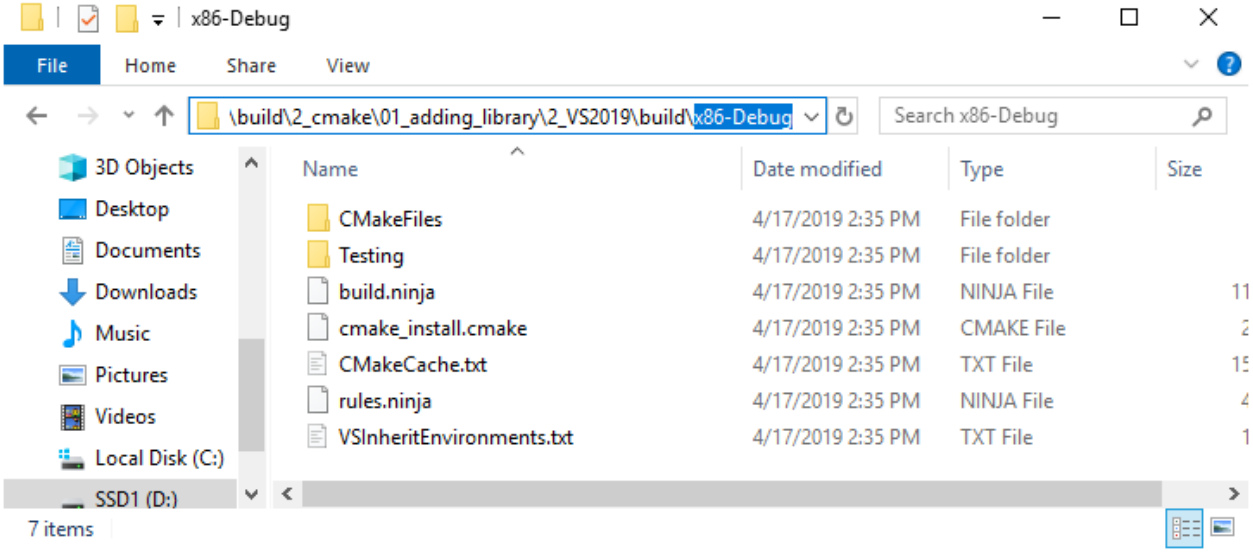
Change the **buildRoot** and **installRoot**:

```
{
  "configurations": [
    {
      "name": "x86-Debug",
      "generator": "Ninja",
      "configurationType": "Debug",
      "buildRoot": "${workspaceRoot}\\build\\${name}",
      "installRoot": "${workspaceRoot}\\install\\${name}",
      "cmakeCommandArgs": "",
      "buildCommandArgs": "-v",
      "ctestCommandArgs": "",
      "inheritEnvironments": [ "msvc_x86" ],
      "variables": []
    }
  ]
}
```

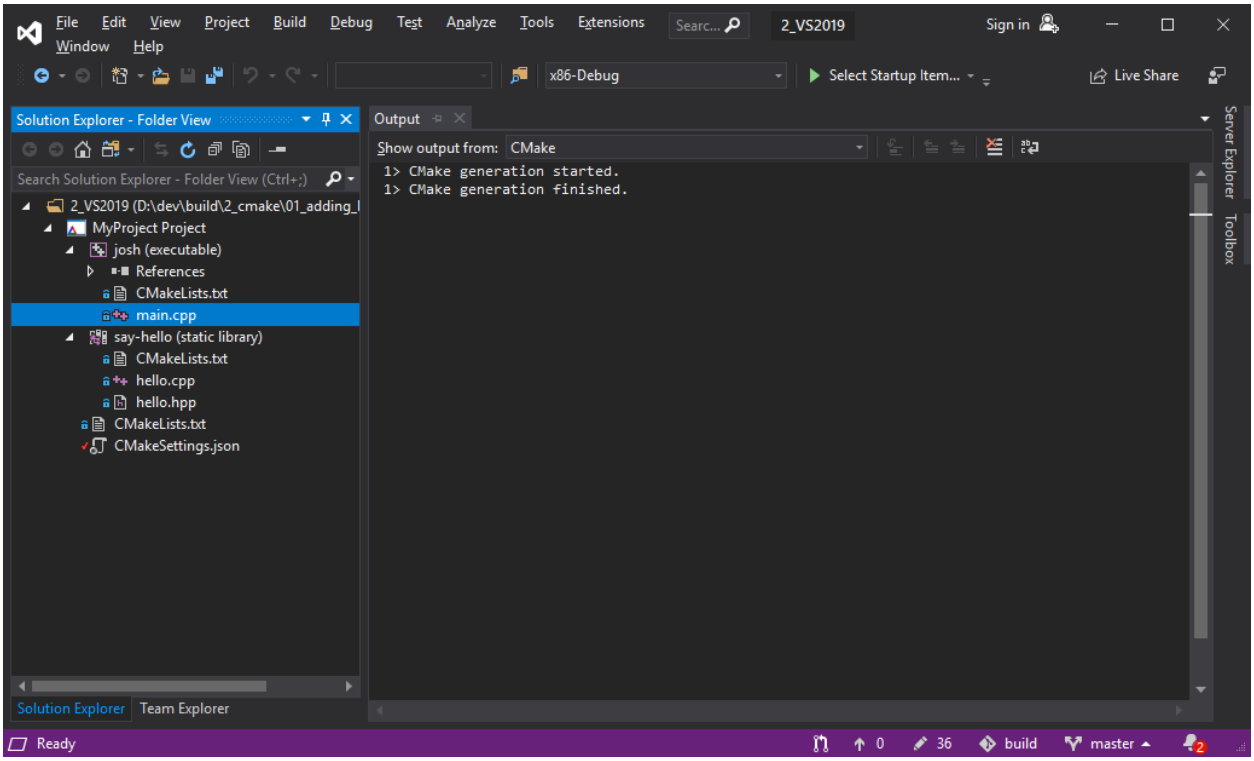
By saving the edit to CMakeSettings.json the build directory is generated automatically (before building):


2_VS2019
1
2
F
Home
H
Share
S
View
V
E
« 2_cmake › 01_adding_library › 2_VS2019 ›
Search 2_VS2019
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
SSD1 (D:)
Name
Date modified
Type
Size
.vs 4/17/2019 2:35 PM File folder
build 4/17/2019 2:35 PM File folder
CMakeLists.txt 4/14/2019 1:38 PM TXT File 1 KB
CMakeSettings.json 4/17/2019 2:35 PM JSON File 1 KB
hello.cpp 4/14/2019 1:09 PM C++ Source 1 KB
hello.hpp 3/24/2019 11:31 AM C/C++ Header 1 KB
main.cpp 3/24/2019 11:34 AM C++ Source 1 KB
7 items
1 item selected

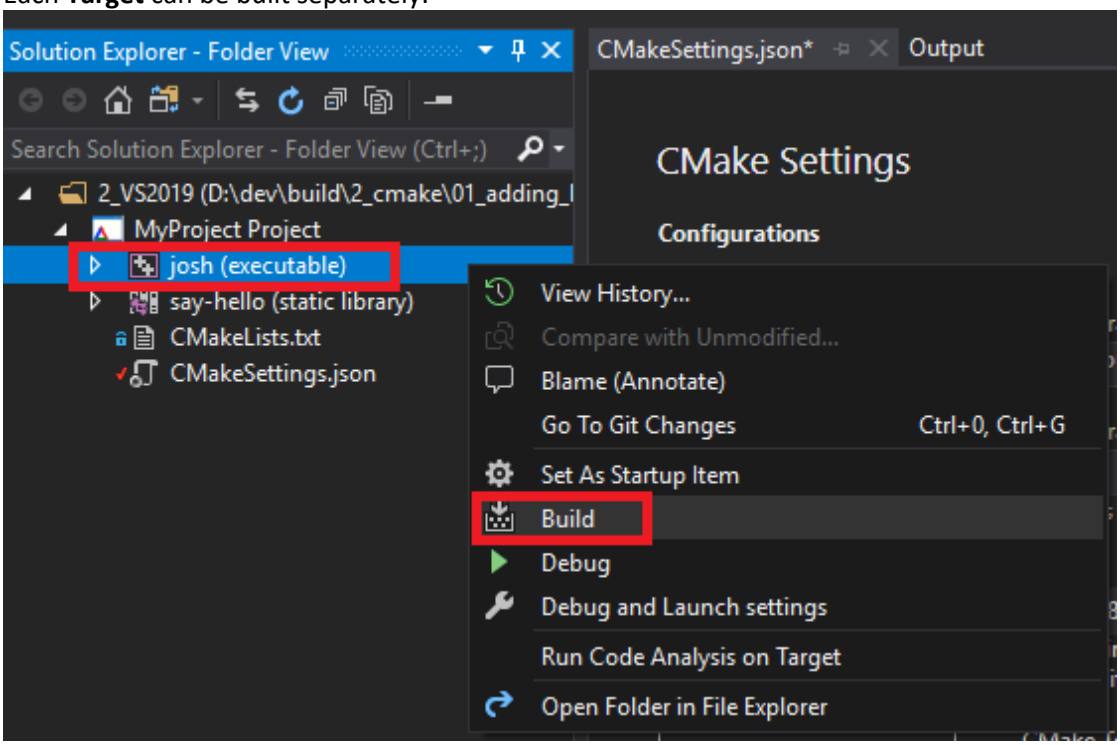

build
File
Home
Share
View
D:\dev\build\2_cmake\01_adding_library\2_VS2019\build
Search build
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
SSD1 (D:)
Name
Date modified
Type
Size
x86-Debug 4/17/2019 2:35 PM File folder
1 item
1 item selected


x86-Debug
File
Home
Share
View
\build\2_cmake\01_adding_library\2_VS2019\build\x86-Debug
Search x86-Debug
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
SSD1 (D:)
Name
Date modified
Type
Size
CMakeFiles 4/17/2019 2:35 PM File folder
Testing 4/17/2019 2:35 PM File folder
build.ninja 4/17/2019 2:35 PM NINJA File
cmake_install.cmake 4/17/2019 2:35 PM CMAKE File
CMakeCache.txt 4/17/2019 2:35 PM TXT File
rules.ninja 4/17/2019 2:35 PM NINJA File
VSInheritEnvironments.txt 4/17/2019 2:35 PM TXT File
7 items

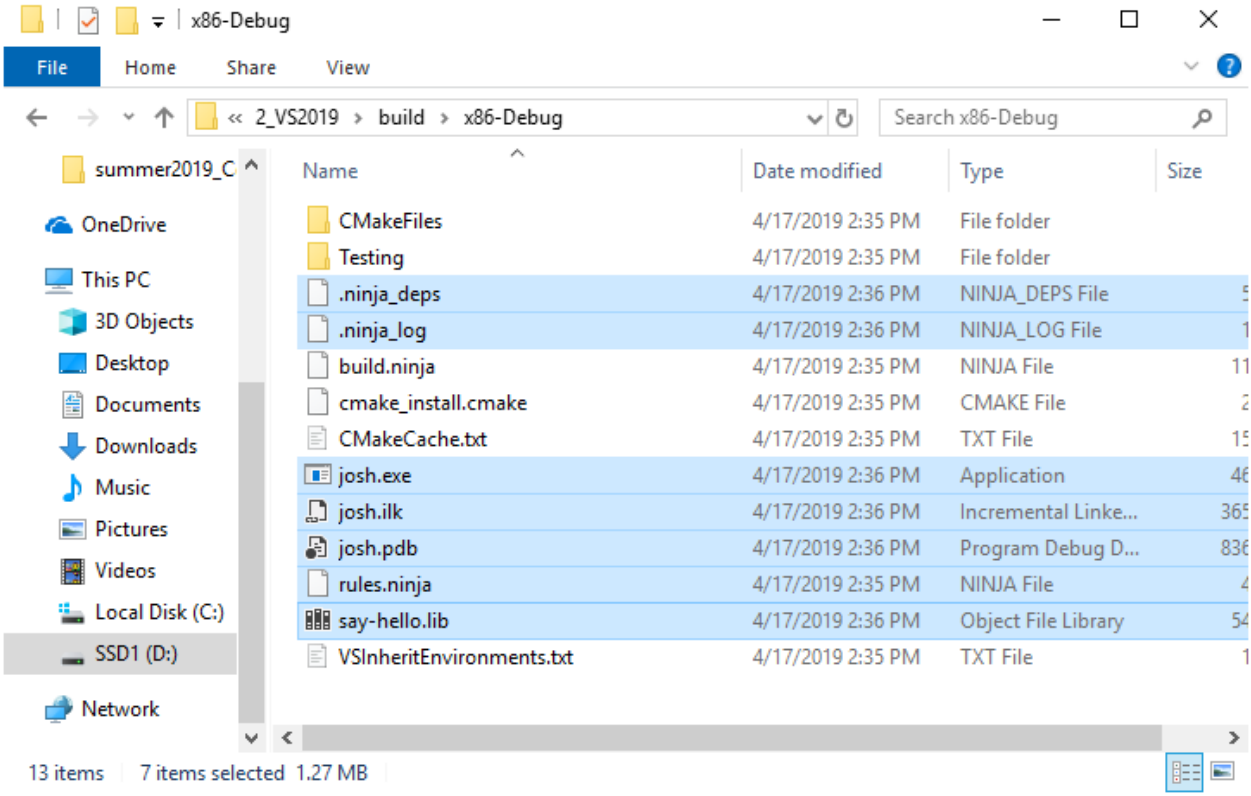
CMakeView:



Each **Target** can be built separately:



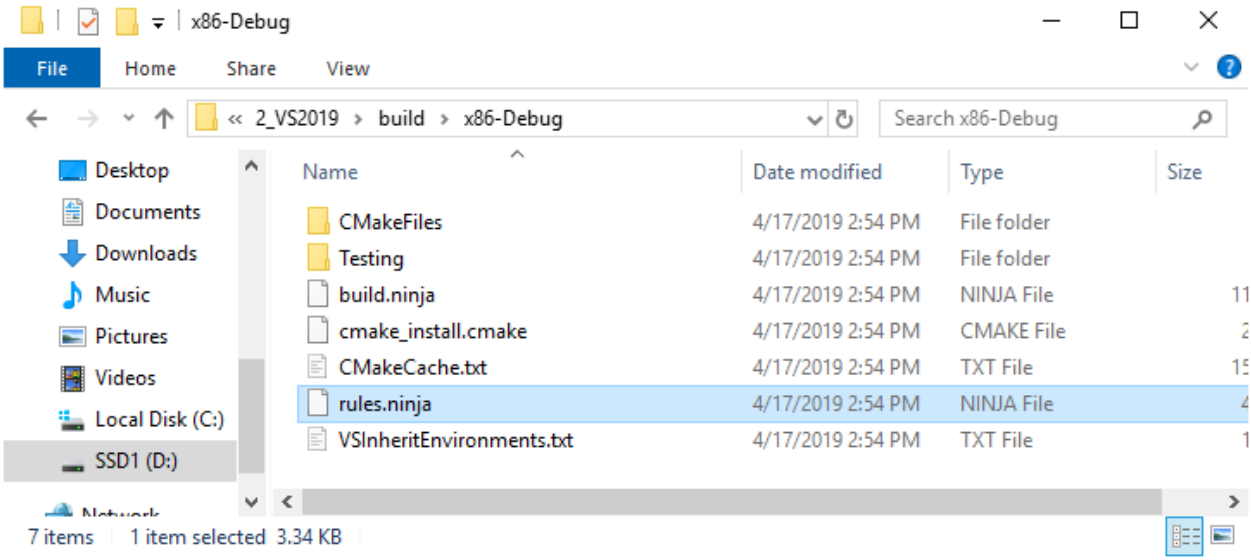
Building the executable target results in the following in the build directory:



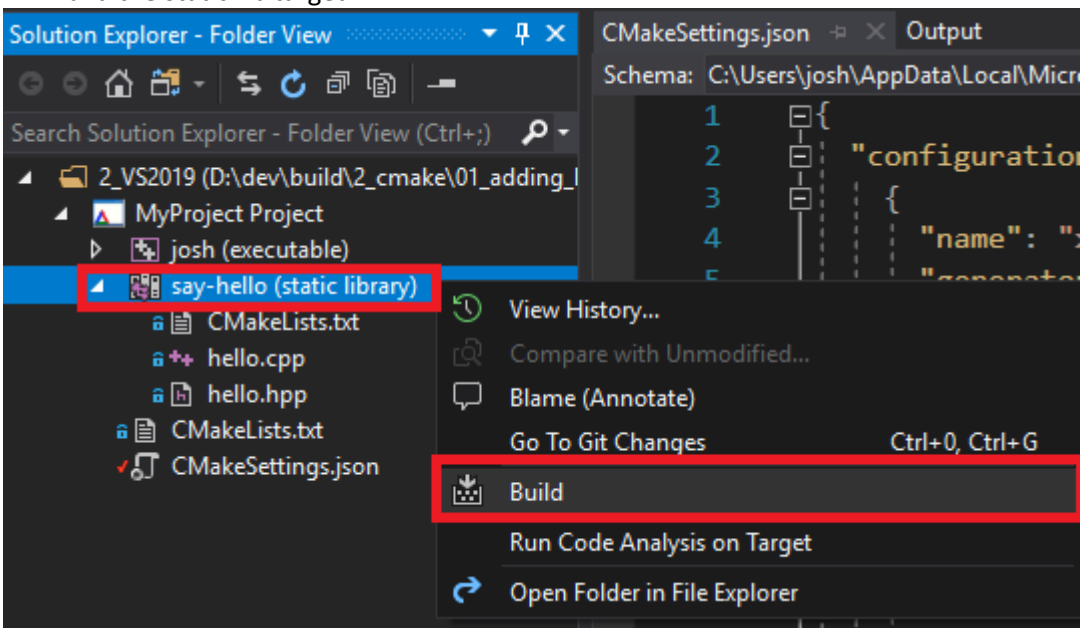
NOTE: rules.ninja was generated in previous step.

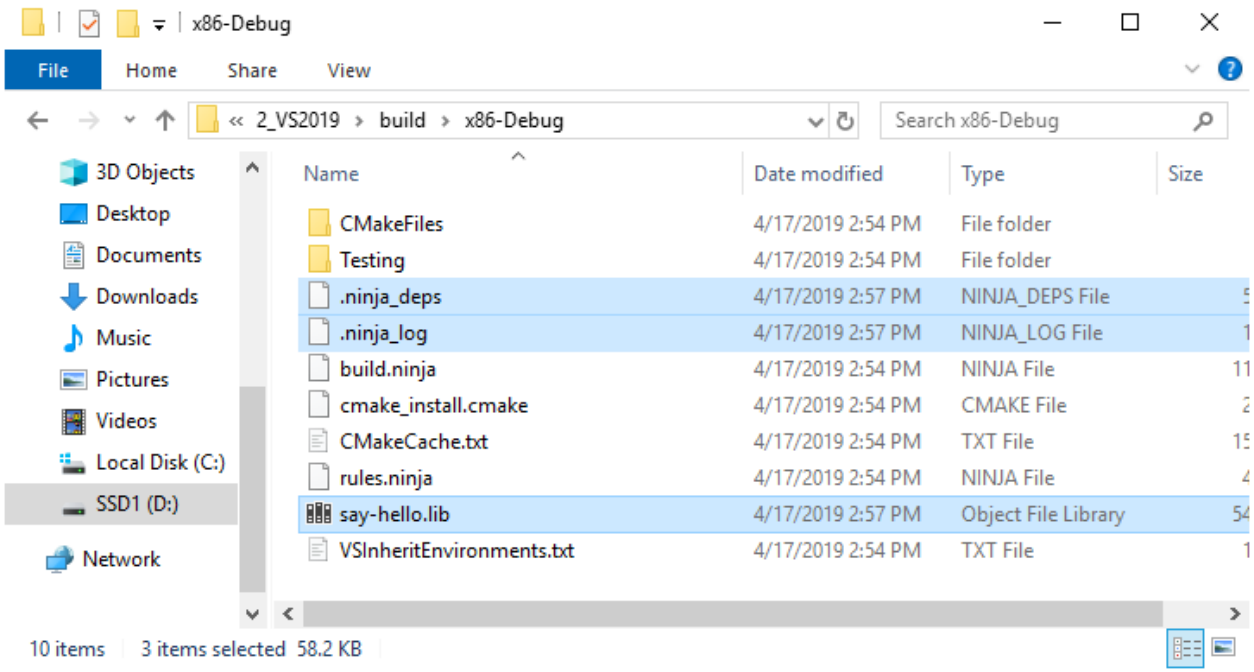
- Apparently, building the executable target automatically builds the static-lib target since we link to it in CMakeLists.txt.
- For completeness, the following step deletes all the generated files, and then first builds the static-lib target, followed by the executable target
    - To show a clear picture of the exact files generated from building the static-lib target

- Delete the built files (actually, stepped through all of the steps from the beginning):

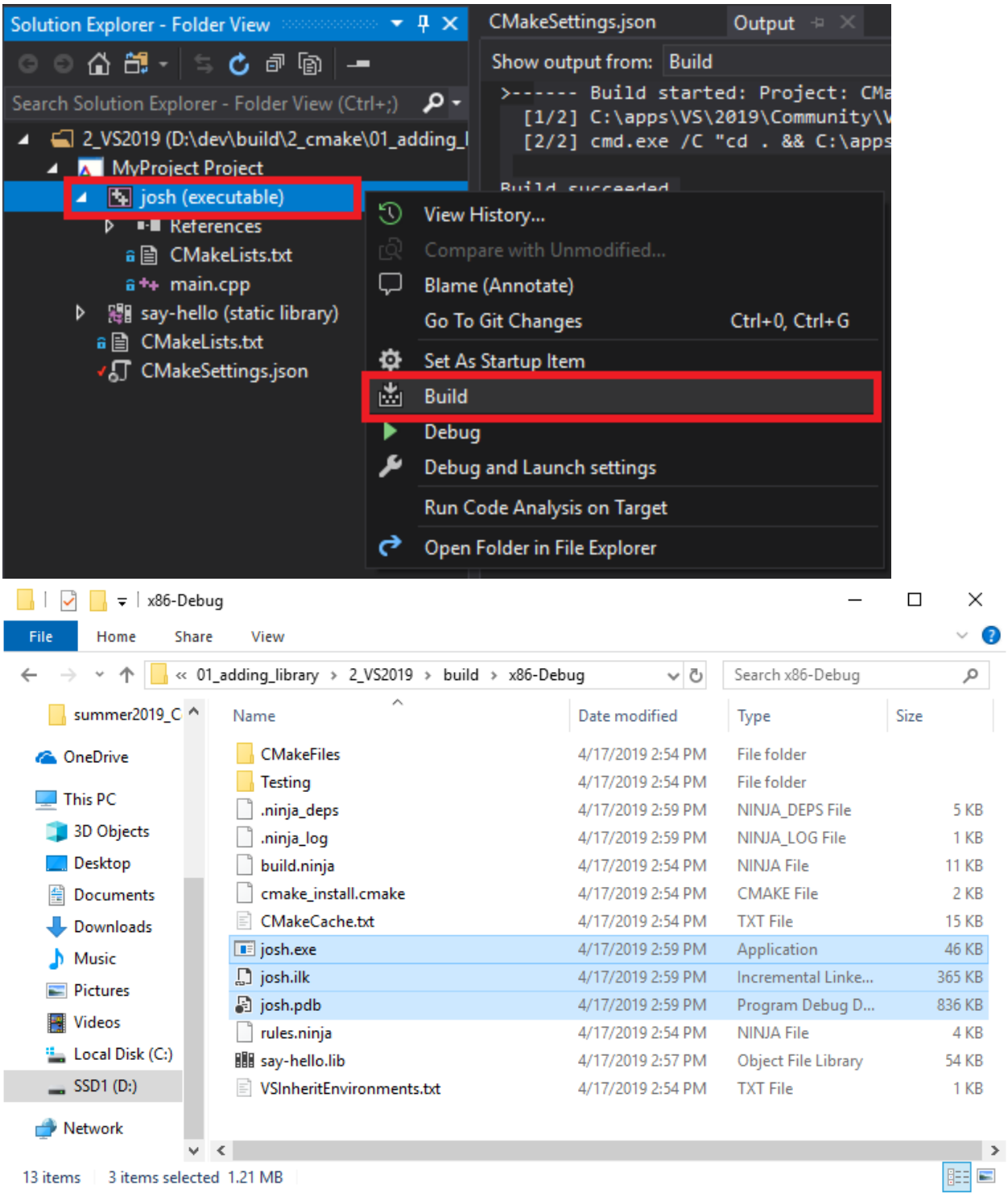


- Build the static-lib target

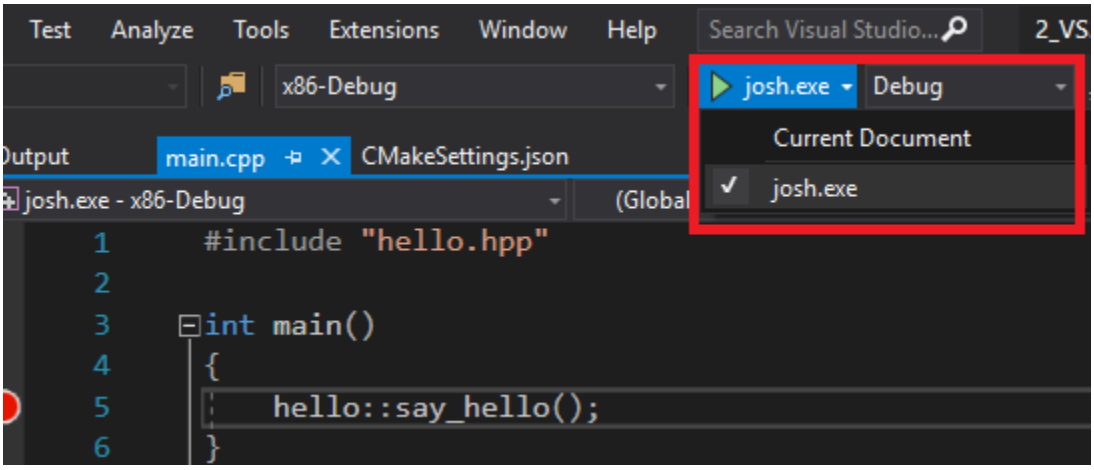


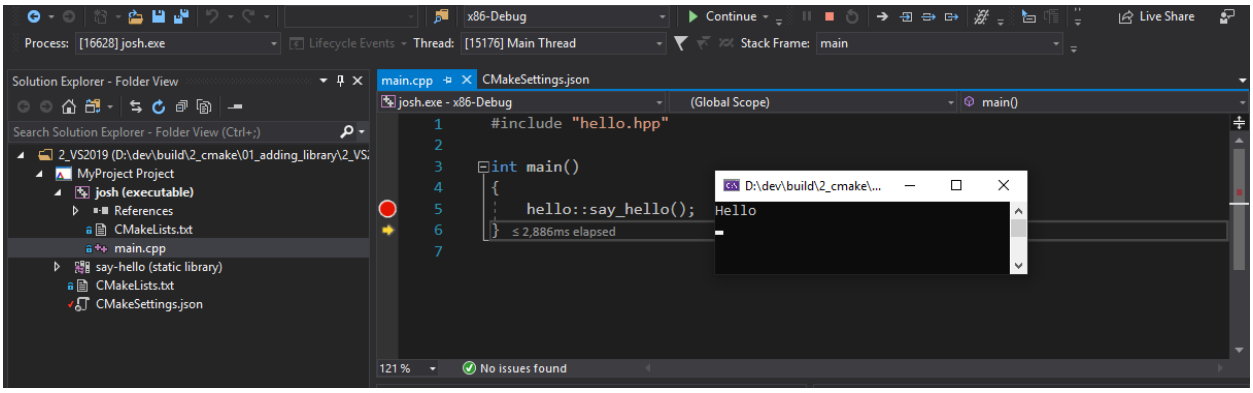


- Build the executable target



- Run the executable


Test
Analyze
Tools
Extensions
Window
Help
Search Visual Studio...
2_VS
x86-Debug
josh.exe
Debug
Current Document
josh.exe
main.cpp
CMakeSettings.json
josh.exe - x86-Debug
#include "hello.hpp"
int main()
{
hello::say_hello();
}


x86-Debug
Continue
Live Share
Process: [16628] josh.exe
Thread: [15176] Main Thread
Stack Frame: main
Solution Explorer - Folder View
Search Solution Explorer - Folder View (Ctrl+;)
2_VS2019 (D:\dev\build\2_cmake\01_adding_library\2_VS
MyProject Project
josh (executable)
References
CMakeLists.txt
main.cpp
say-hello (static library)
CMakeLists.txt
CMakeSettings.json
main.cpp
CMakeSettings.json
josh.exe - x86-Debug
(Global Scope)
main()
#include "hello.hpp"
int main()
{
hello::say_hello();
}
≤ 2,886ms elapsed
D:\dev\build\2_cmake\...
Hello
121 %
No issues found