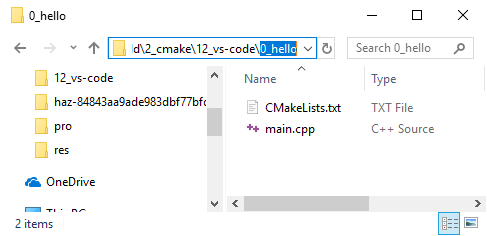
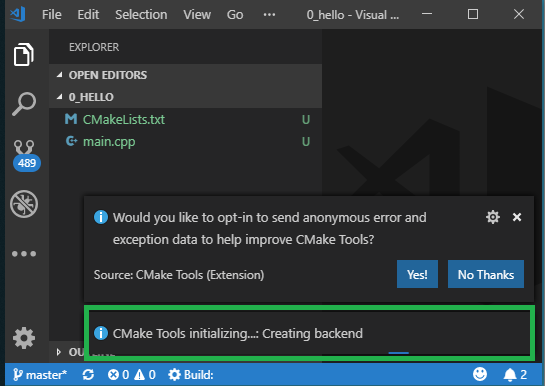
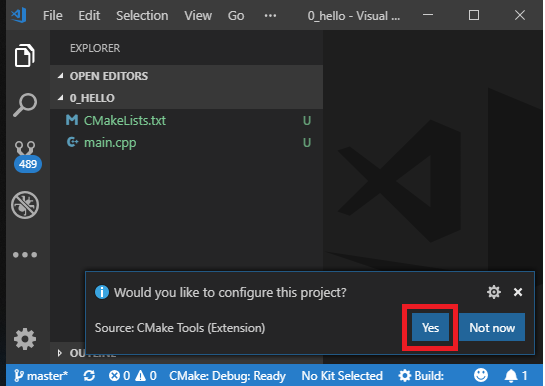
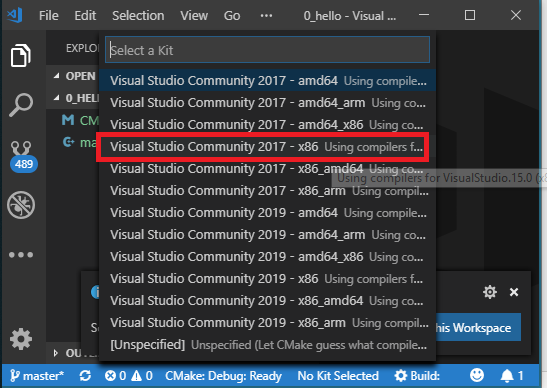
Install CMake and CMake-Tools extensions in VS-Code.

Open a directory with a CMake based project.



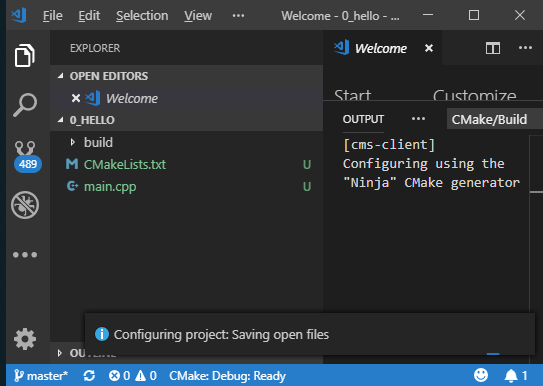


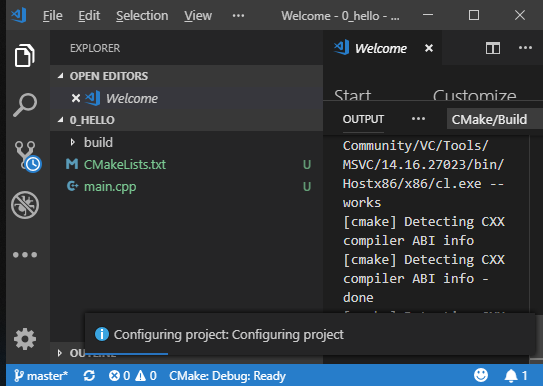




If the configure option doesn’t start automatically, then enter the following in the Command pallet (**cntrl**+**shift**+**p**)

CMake Configure





[cms-client] Configuring using the "Ninja" CMake generator

[cms-client] Configuring using the "Ninja" CMake generator

[cmake] The C compiler identification is MSVC 19.16.27030.1

[cmake] The CXX compiler identification is MSVC 19.16.27030.1

[cmake] Check for working C compiler: C:/apps/VS/2017/Community/VC/Tools/MSVC/14.16.27023/bin/Hostx86/x86/cl.exe

[cmake] Check for working C compiler: C:/apps/VS/2017/Community/VC/Tools/MSVC/14.16.27023/bin/Hostx86/x86/cl.exe -- works

[cmake] Detecting C compiler ABI info

[cmake] Detecting C compiler ABI info - done

[cmake] Detecting C compile features

[cmake] Detecting C compile features - done

[cmake] Check for working CXX compiler: C:/apps/VS/2017/Community/VC/Tools/MSVC/14.16.27023/bin/Hostx86/x86/cl.exe

[cmake] Check for working CXX compiler: C:/apps/VS/2017/Community/VC/Tools/MSVC/14.16.27023/bin/Hostx86/x86/cl.exe -- works

[cmake] Detecting CXX compiler ABI info

[cmake] Detecting CXX compiler ABI info - done

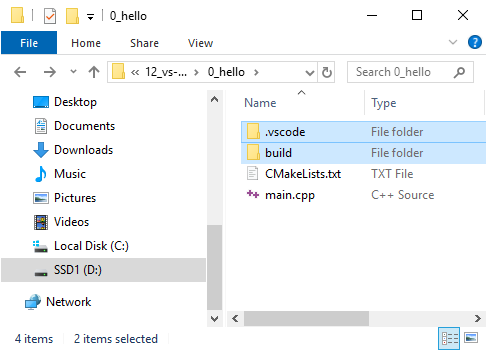
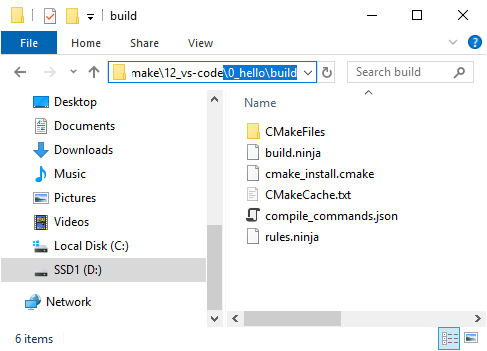
[cmake] Detecting CXX compile features

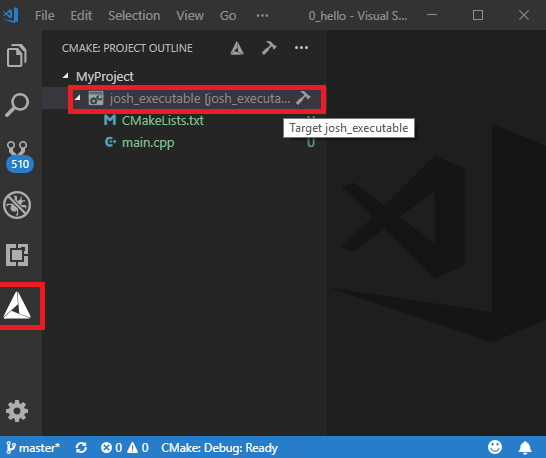
[cmake] Detecting CXX compile features - done

[cmake] Configuring done

[cmake] Generating done

After configuring the project the build and .vs directories are created.

Select the CMake icon to see CMake view.

Contents of **CMakeLists.txt**:

cmake\_minimum\_required(VERSION 3.10)

project(MyProject VERSION 1.0.0)

add\_executable(josh\_executable main.cpp)

Contents of **main.cpp**

#include <iostream>

auto main() -> int

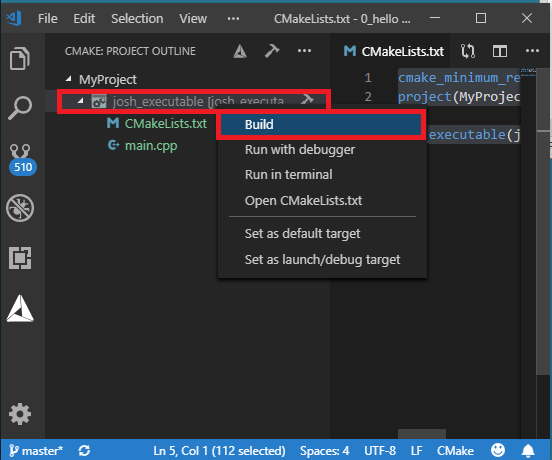
{

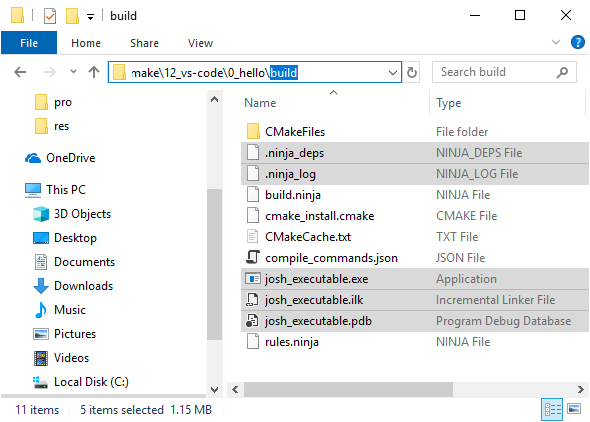
std::cout << "Hello!\n";

return(0);

}

You can build individual targets.





Run executable with debugger

