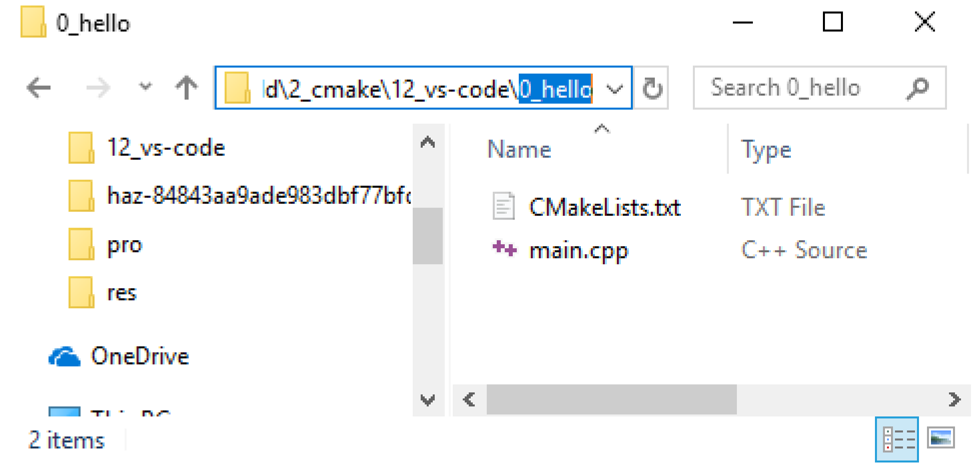
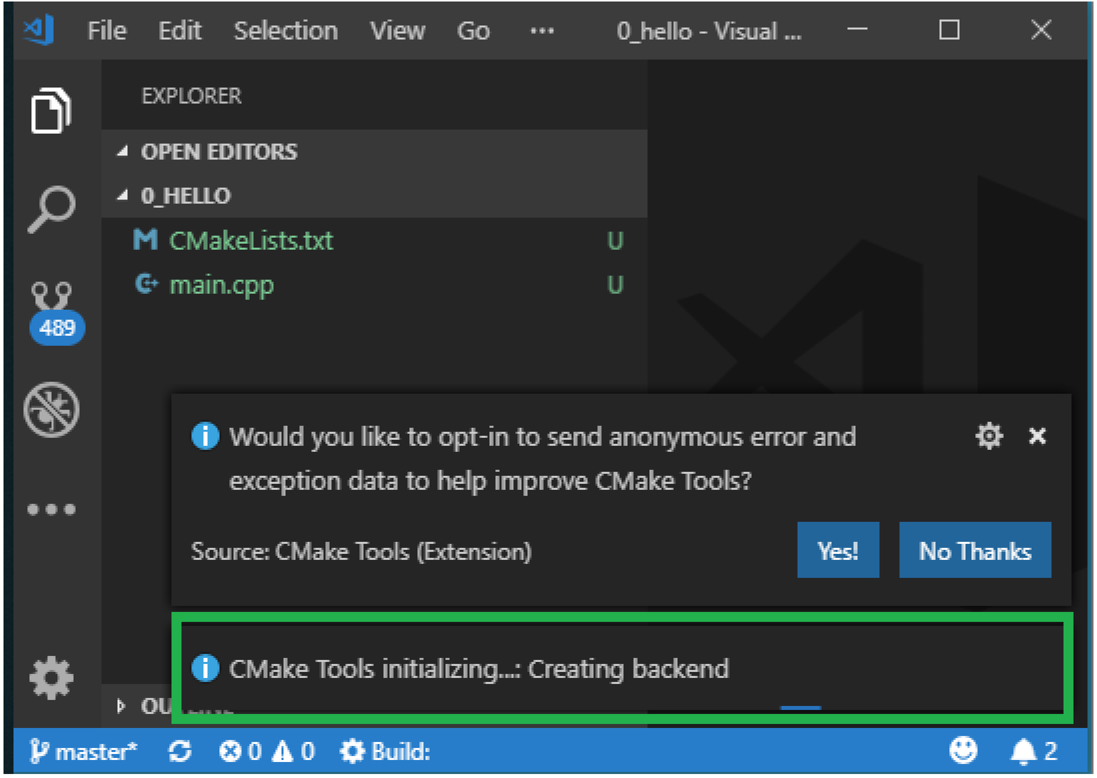
Install CMake and CMake-Tools extensions in VS-Code.

Open a directory with a CMake based project.





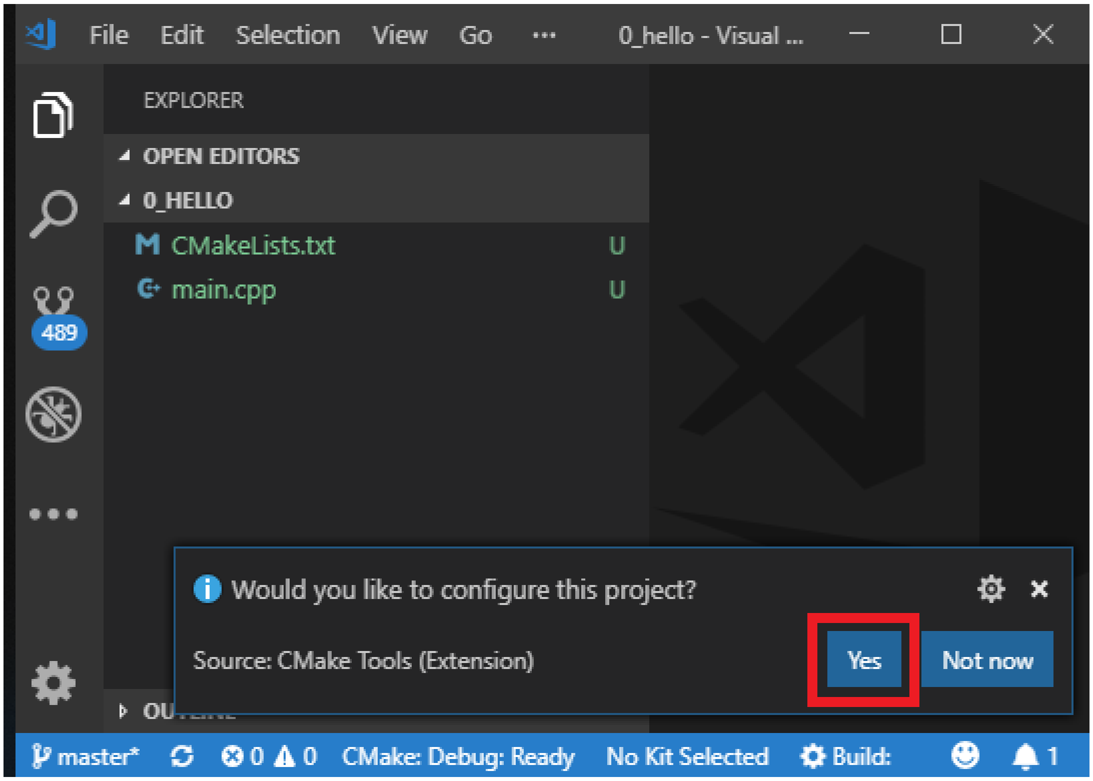
Would you like to opt-in to send anonymous error and exception data to help improve CMake Tools?

Source: CMake Tools (Extension)

Yes!

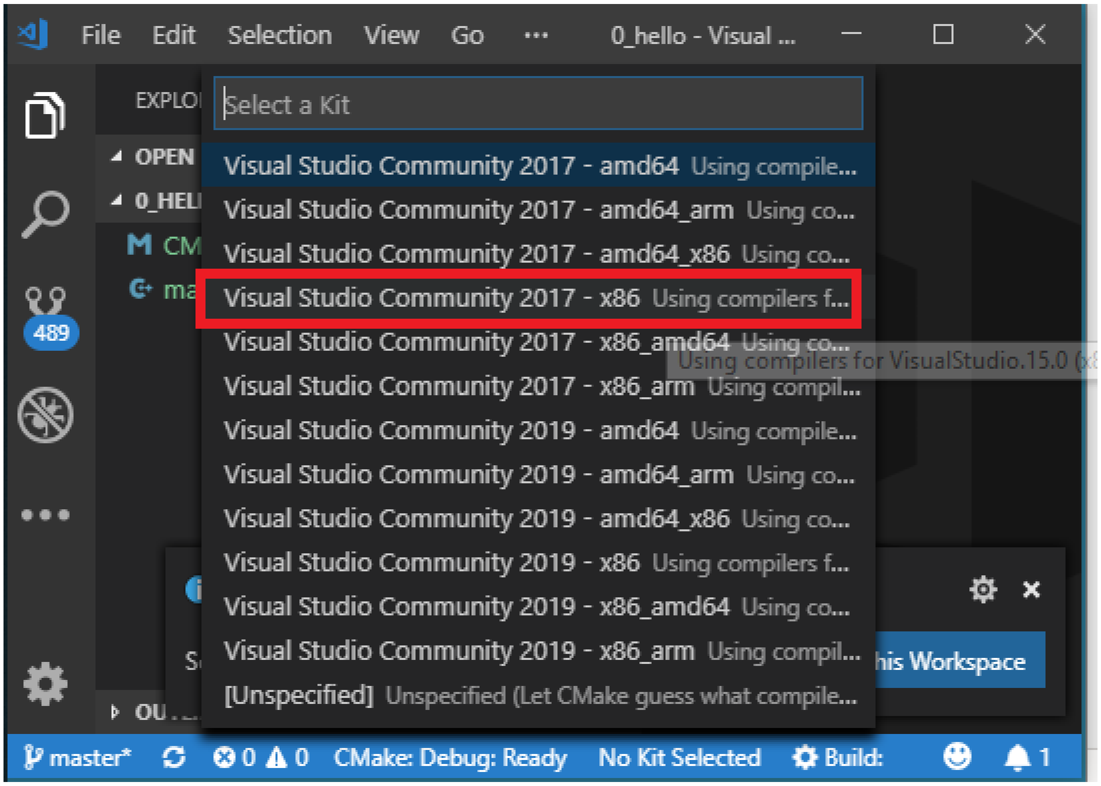
No Thanks

CMake Tools initializing...: Creating backend



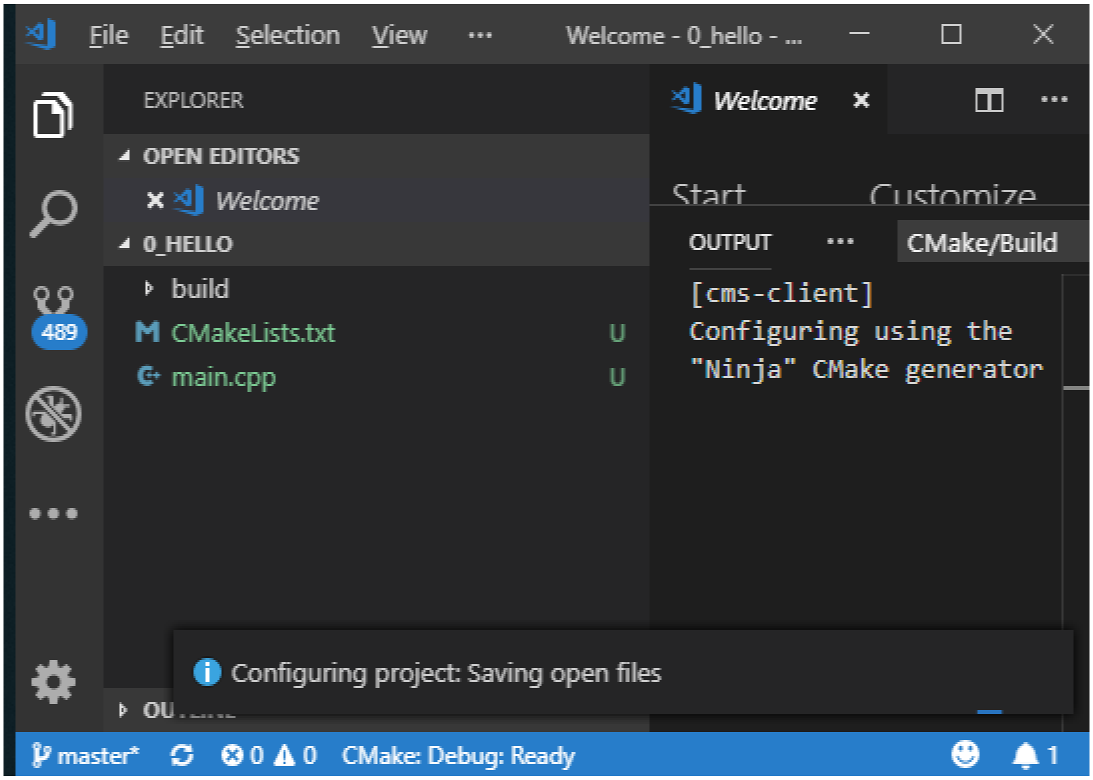
Would you like to configure this project?

Source: CMake Tools (Extension)



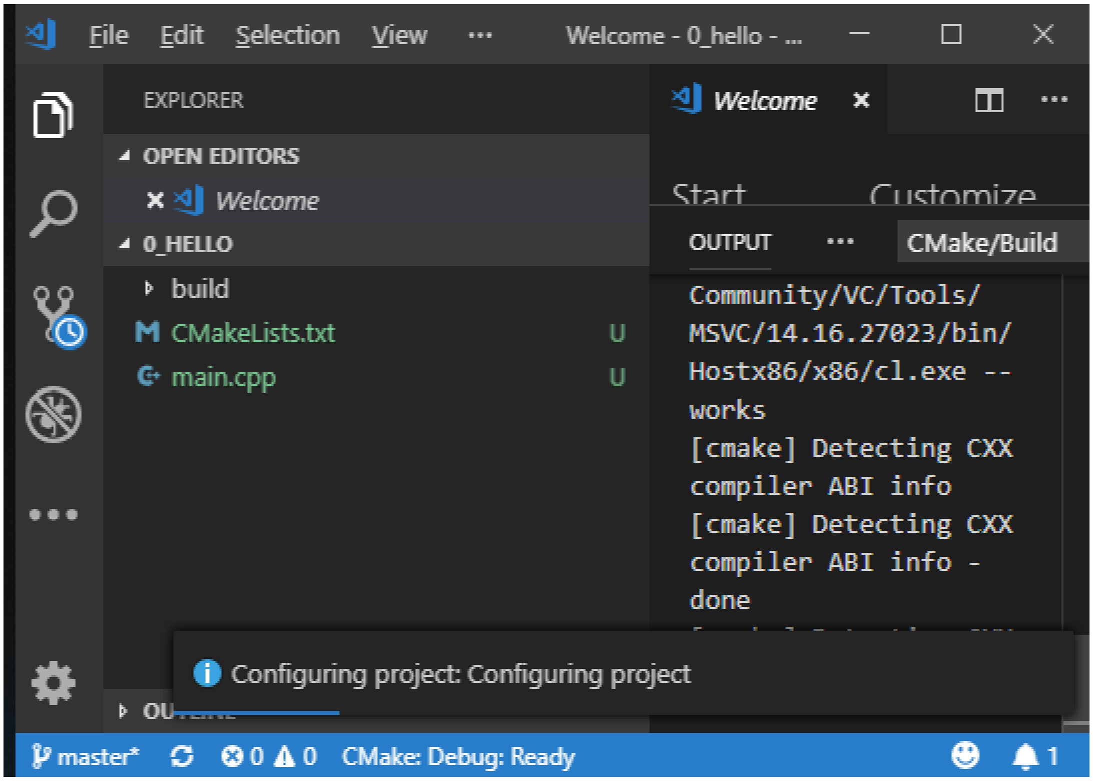
File
Edit
Selection
View
Go
0_hello - Visual ...
Select a Kit
Visual Studio Community 2017 - amd64 Using compile...
Visual Studio Community 2017 - amd64_arm Using co...
Visual Studio Community 2017 - amd64_x86 Using co...
Visual Studio Community 2017 - x86 Using compilers f...
Visual Studio Community 2017 - x86_amd64 Using co...
Using compilers for VisualStudio.15.0 (x8
Visual Studio Community 2017 - x86_arm Using compil...
Visual Studio Community 2019 - amd64 Using compile...
Visual Studio Community 2019 - amd64_arm Using co...
Visual Studio Community 2019 - amd64_x86 Using co...
Visual Studio Community 2019 - x86 Using compilers f...
Visual Studio Community 2019 - x86_amd64 Using co...
Visual Studio Community 2019 - x86_arm Using compil...
[Unspecified] Unspecified (Let CMake guess what compile...
his Workspace
489
master*
0 0
CMake: Debug: Ready
No Kit Selected
Build:
1

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



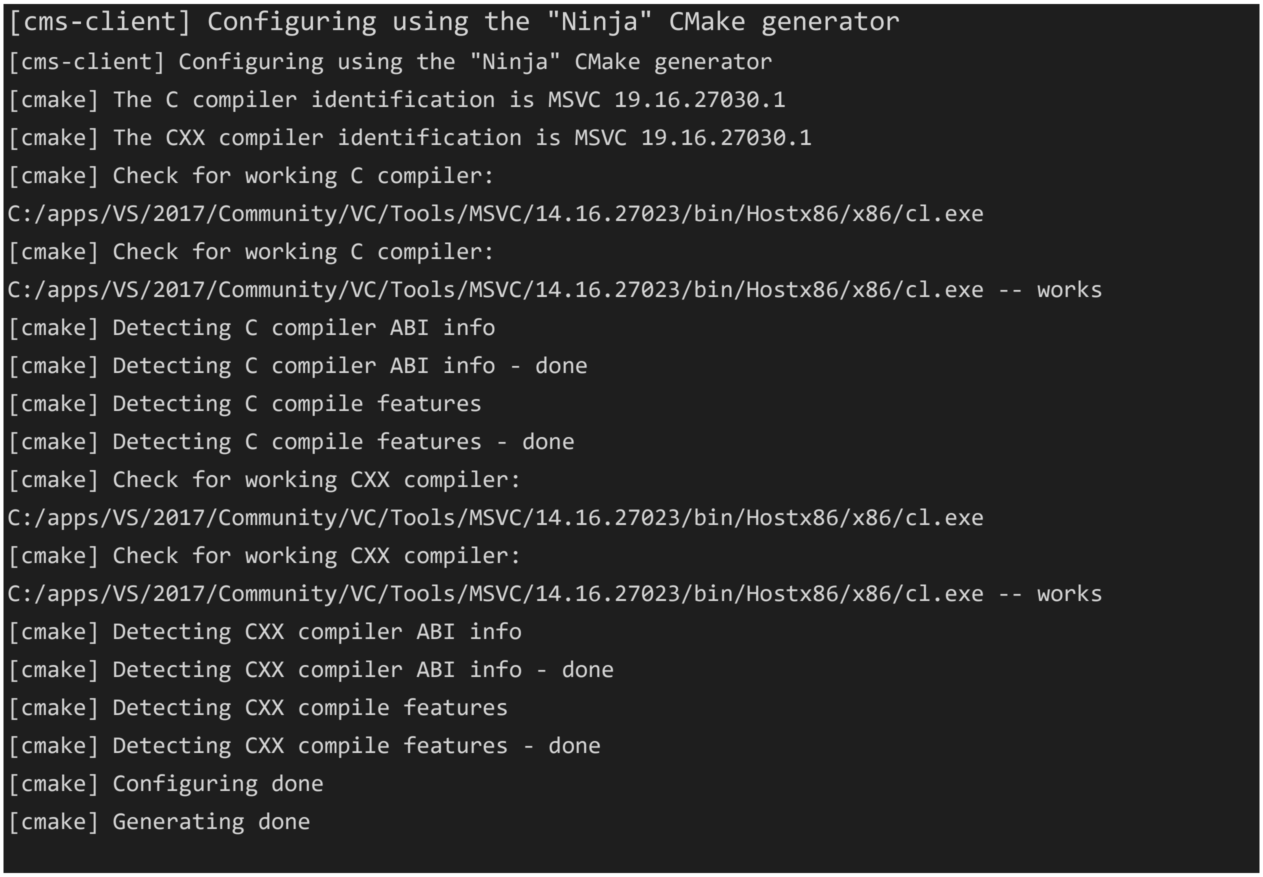
```
[cms-client]
Configuring using the
"Ninja" CMake generator
```

Configuring project: Saving open files



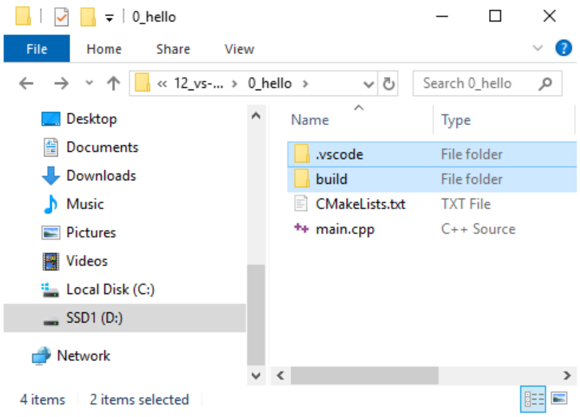
```
Community/VC/Tools/
MSVC/14.16.27023/bin/
Hostx86/x86/cl.exe --
works
[cmake] Detecting CXX
compiler ABI info
[cmake] Detecting CXX
compiler ABI info -
done
```

Configuring project: Configuring project



```
[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.

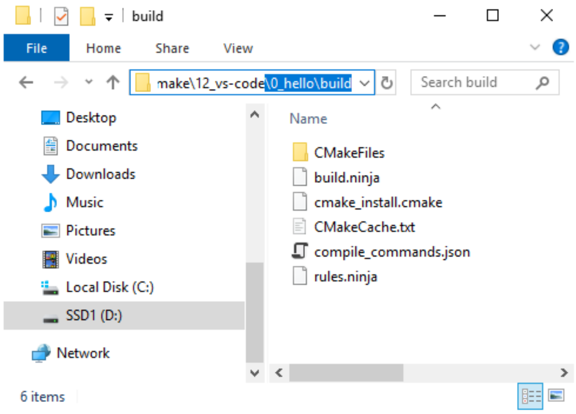


0_hello

« 12_vs-... › 0_hello ›

Name	Type
.vscode	File folder
build	File folder
CMakeLists.txt	TXT File
main.cpp	C++ Source

4 items   2 items selected



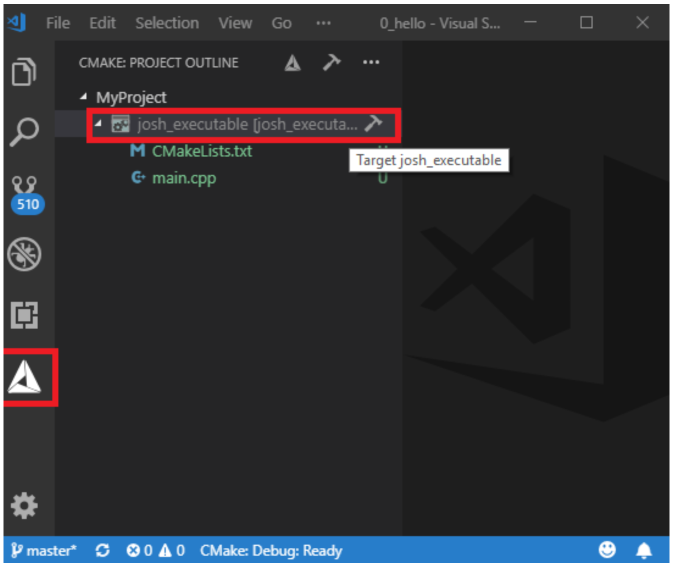
build

make\12_vs-code\0_hello\build

Name
CMakeFiles
build.ninja
cmake_install.cmake
CMakeCache.txt
compile_commands.json
rules.ninja

6 items

Select the CMake icon to see CMake view.



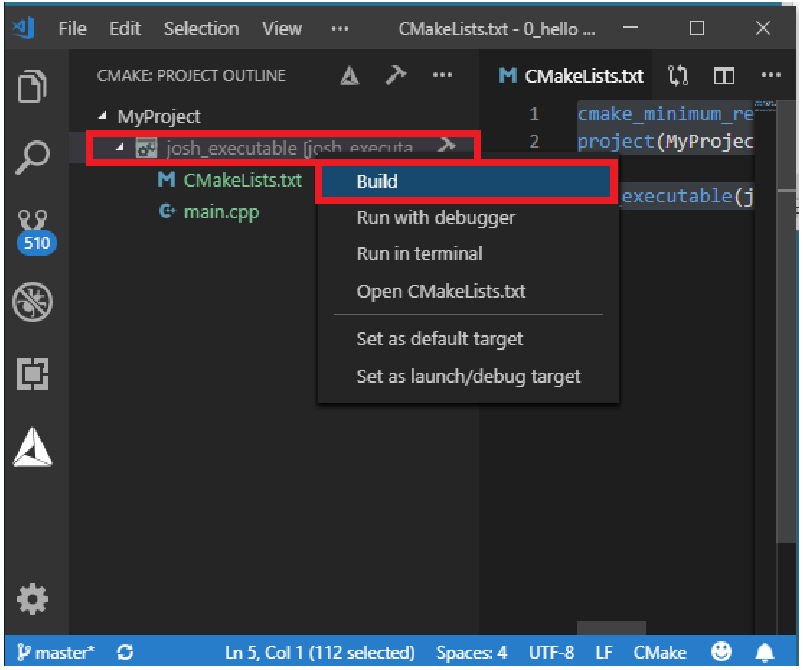
<u>Contents of **CMakeLists.txt**:</u>

```
cmake_minimum_required(VERSION 3.10)
project(MyProject VERSION 1.0.0)

add_executable(josh_executable
main.cpp)
```

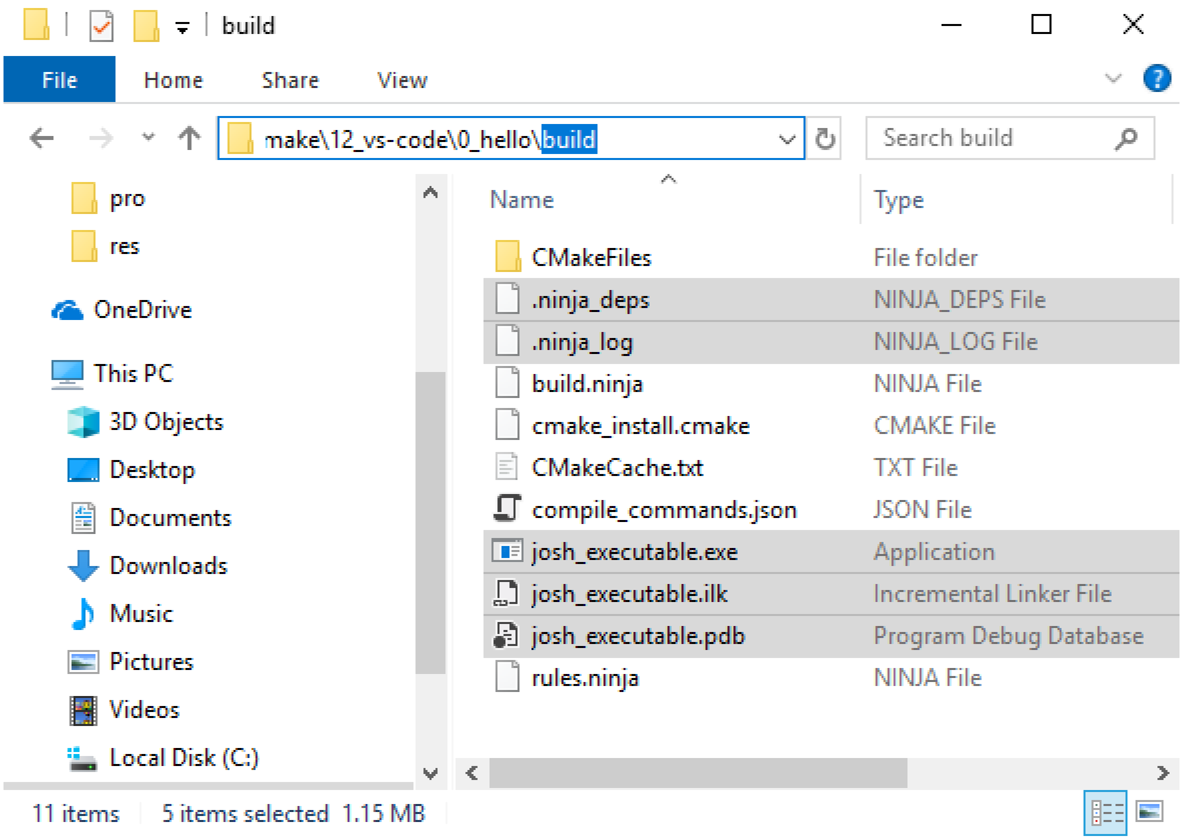
<u>Contents of **main.cpp**</u>

```
#include <iostream>
auto main() -> int
{
        std::cout << "Hello!\n";
        return(0);
}
```

You can build individual targets.



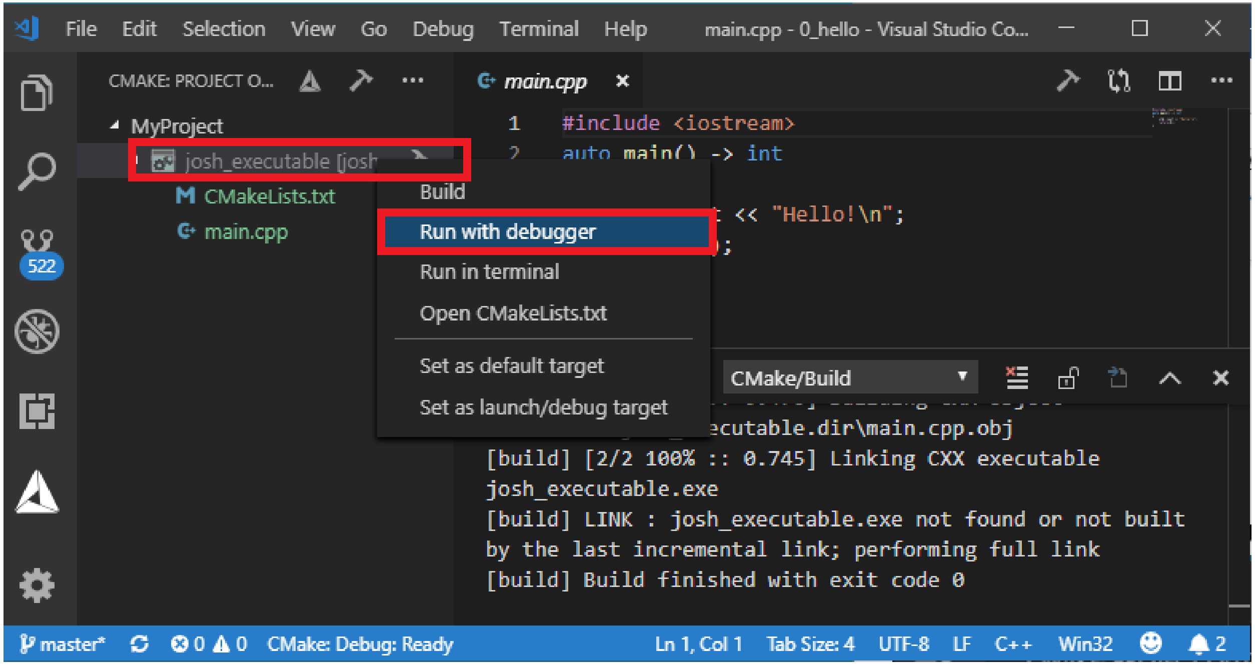
```
cmake_minimum_re
project(MyProjec

executable(j
```



Name	Type
CMakeFiles	File folder
.ninja_deps	NINJA_DEPS File
.ninja_log	NINJA_LOG File
build.ninja	NINJA File
cmake_install.cmake	CMAKE File
CMakeCache.txt	TXT File
compile_commands.json	JSON File
josh_executable.exe	Application
josh_executable.ilk	Incremental Linker File
josh_executable.pdb	Program Debug Database
rules.ninja	NINJA File

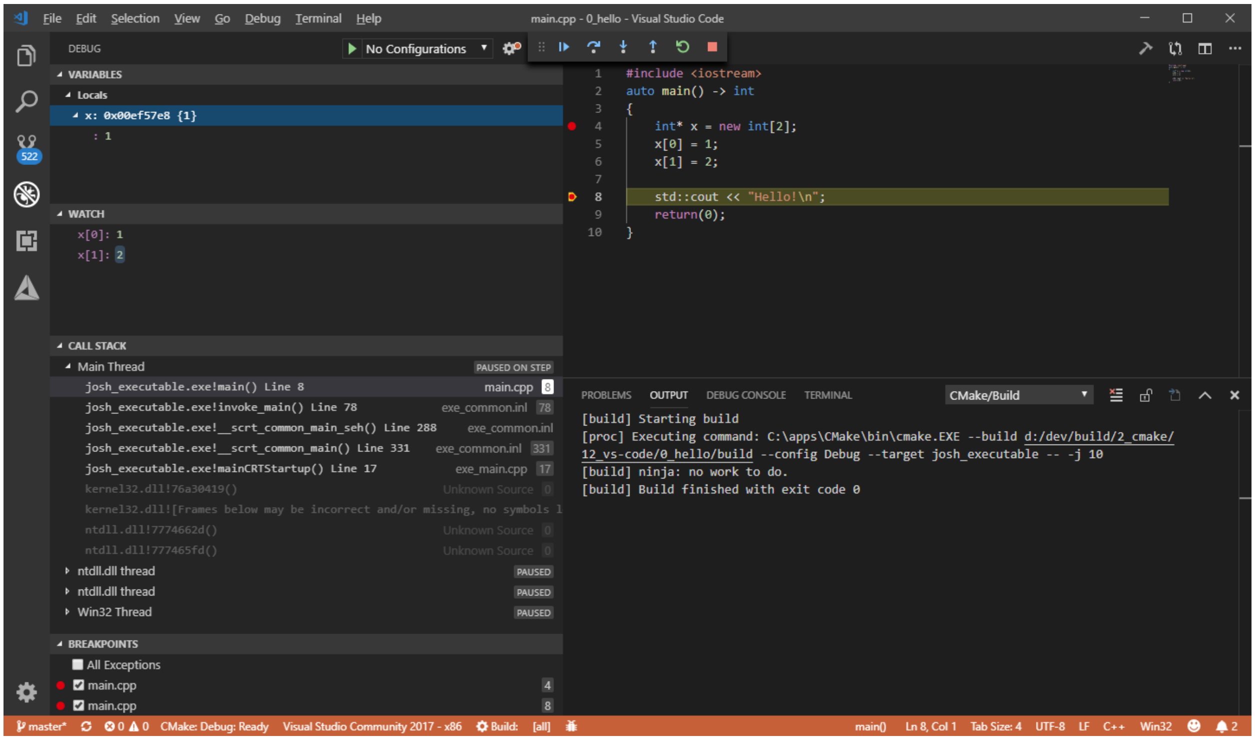
11 items 5 items selected 1.15 MB

Run executable with debugger



```
#include <iostream>
auto main() -> int
: << "Hello!\n";
;
```

```
ecutable.dir\main.cpp.obj
[build] [2/2 100% :: 0.745] Linking CXX executable
josh_executable.exe
[build] LINK : josh_executable.exe not found or not built
by the last incremental link; performing full link
[build] Build finished with exit code 0
```



main.cpp - 0_hello - Visual Studio Code

No Configurations

VARIABLES

Locals

```
x: 0x00ef57e8 {1}
  : 1
```

WATCH

```
x[0]: 1
x[1]: 2
```

CALL STACK

Main Thread PAUSED ON STEP

```
josh_executable.exe!main() Line 8                        main.cpp 8
josh_executable.exe!invoke_main() Line 78                exe_common.inl 78
josh_executable.exe!__scrt_common_main_seh() Line 288    exe_common.inl
josh_executable.exe!__scrt_common_main() Line 331        exe_common.inl 331
josh_executable.exe!mainCRTStartup() Line 17             exe_main.cpp 17
kernel32.dll!76a30419()                                  Unknown Source 0
kernel32.dll![Frames below may be incorrect and/or missing, no symbols l
ntdll.dll!7774662d()                                     Unknown Source 0
ntdll.dll!777465fd()                                     Unknown Source 0
```

ntdll.dll thread PAUSED

ntdll.dll thread PAUSED

Win32 Thread PAUSED

BREAKPOINTS

- □ All Exceptions
- ● ☑ main.cpp 4
- ● ☑ main.cpp 8

```cpp
#include <iostream>
auto main() -> int
{
    int* x = new int[2];
    x[0] = 1;
    x[1] = 2;

    std::cout << "Hello!\n";
    return(0);
}
```

CMake/Build

```
[build] Starting build
[proc] Executing command: C:\apps\CMake\bin\cmake.EXE --build d:/dev/build/2_cmake/
12_vs-code/0_hello/build --config Debug --target josh_executable -- -j 10
[build] ninja: no work to do.
[build] Build finished with exit code 0
```

master* ⊗ 0 ⚠ 0 CMake: Debug: Ready Visual Studio Community 2017 - x86 Build: [all] main() Ln 8, Col 1 Tab Size: 4 UTF-8 LF C++ Win32 2