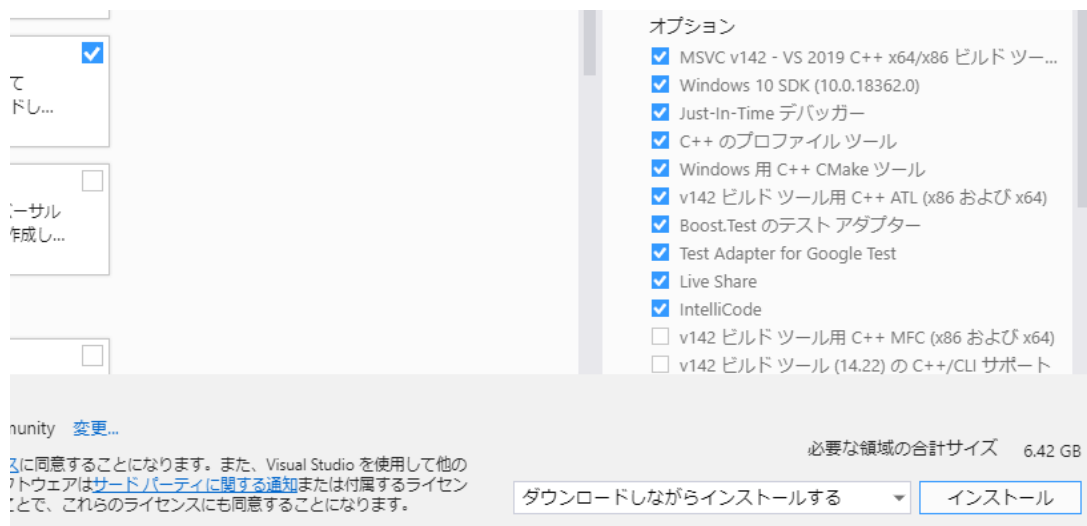


CMake + NMakeのインストール、使用手順

本文書では Visual Studio Community 2019 のインストーラによって、関連ビルドツールである CMake および NMake をインストールする手順および使用手順を説明する。

インストール手順

1. CMake、NMake は、Visual Studio Community 2019 のインストール手順において、[C++によるデスクトップ開発]のチェックボックスをONにすることによって、デフォルトでインストール対象として選択される。個別に確認するには、ウィンドウ右側のオプション項目で、[MSVC v142 - VS 2019 C++ x64/x86 ビルドツール] (C++ コンパイラやNMakeに対応) 、 [Windows用 C++ CMake ツール] のチェックボックスがONになっていることを確認する。



インストール後のCMake, NMakeの動作確認

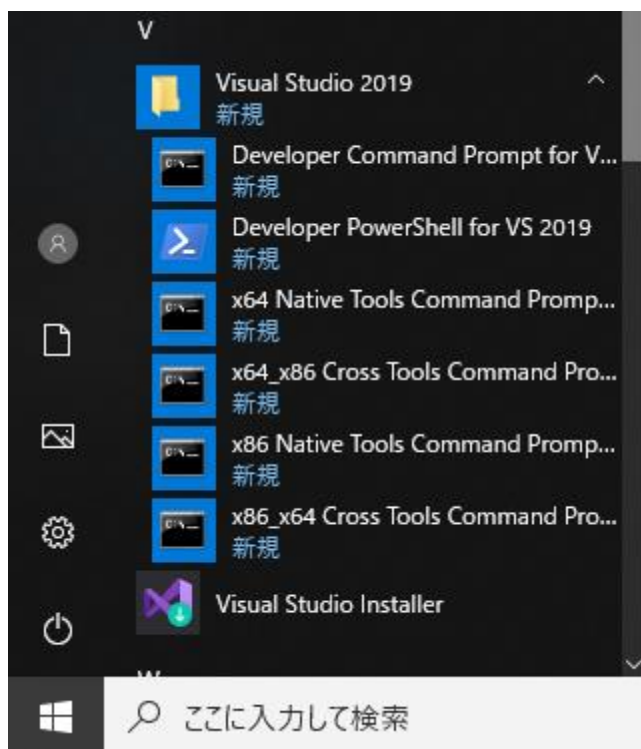
1. テキストエディタで次の内容のファイルを作成し、 C:¥tmp¥test.cpp に保存する。

```
#include <iostream>
int main()
{
    std::cout << "Hello" << std::endl;
    return 0;
}
```

2. テキストエディタで次の内容のファイルを作成し、 C:¥tmp¥CMakeLists.txt に保存する

```
cmake_minimum_required(VERSION 2.8.12)
project(test)
add_executable(test test.cpp)
```

3. タスクバーの検索で"command"（または "tools"）を検索する。これにより [x64 Native Tools Command Prompt for VS 2019]という項目が見つかる。



4. 項目 [x64 Native Tools Command Prompt for VS 2019] をクリックしてコマンドプロンプトを開く。このコマンドプロンプトでは NMake 等を利用するための設定が読み込まれている。

```
*****
** Visual Studio 2019 Developer Command Prompt v16.2.3
** Copyright (c) 2019 Microsoft Corporation
*****
[vcvarsall.bat] Environment initialized for: 'x64'
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community>_
```

5. 次のコマンドにより、カレントディレクトリを ソースファイルがあるディレクトリ C:

¥tmp に変更する

cd C:¥tmp

```
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community>cd C:¥tmp
C:¥tmp>
```

6. 次のコマンドにより、ビルド用のディレクトリ build を作成する。

mkdir build

```
C:¥tmp>mkdir build
C:¥tmp>cd build
C:¥tmp¥build>_
```

7. 次のコマンドにより、カレントディレクトリを build に変更する。

cd build

8. 次のコマンドにより、CMakeLists.txt から NMake 用のMakefileを生成する。

cmake .. -G "NMake Makefiles"

```

C:\tmp\build>cmake .. -G "NMake Makefiles"
-- The C compiler identification is MSVC 19.22.27905.0
-- The CXX compiler identification is MSVC 19.22.27905.0
-- Check for working C compiler: C:/Program Files (x86)/Microsoft Visual Studio/2019/Community/VC/Tools/MSVC/14.22.27905/bin/Hostx64/x64/cl.exe
-- Check for working C compiler: C:/Program Files (x86)/Microsoft Visual Studio/2019/Community/VC/Tools/MSVC/14.22.27905/bin/Hostx64/x64/cl.exe -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: C:/Program Files (x86)/Microsoft Visual Studio/2019/Community/VC/Tools/MSVC/14.22.27905/bin/Hostx64/x64/cl.exe
-- Check for working CXX compiler: C:/Program Files (x86)/Microsoft Visual Studio/2019/Community/VC/Tools/MSVC/14.22.27905/bin/Hostx64/x64/cl.exe -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: C:/tmp/build
C:\tmp\build>

```

9. 次のコマンドによりソースファイルのビルドを行う。

nmake

```

C:\tmp\build>nmake

Microsoft(R) Program Maintenance Utility Version 14.22.27905.0
Copyright (C) Microsoft Corporation. All rights reserved.

Scanning dependencies of target test
[ 50%] Building CXX object CMakeFiles/test.dir/test.cpp.obj
test.cpp
[100%] Linking CXX executable test.exe
[100%] Built target test
C:\tmp\build>

```

10. ビルドした test.exe を実行する。

```

C:\tmp\build>test.exe
Hello

```

以上