

# CMake Bonus

---

作者：F74054122 林家緯

## 功能

---

- 使用 CMake 把自訂 library 鏈接到程式
- 可使用 CMake GUI 選擇 STATIC / SHARED build方式

## 執行環境

---

- Microsoft Visual Studio 2017
- CMake GUI 3.14.4

## 目錄結構

---

```
/
|--/ext           # 外部 Library (NANA, FMT)
|--/square        # 自訂 Library
|  |--CMakeLists.txt
|  |--square.cpp
|  |--square.h
|--/src           # 主程式
   |--main.cpp
```

## 自訂 Library

---

### 程式碼部分

```
-----square.h-----
#ifndef SQUARE_H_
#define SQUARE_H_

int square(int num);

#endif

-----square.cpp-----
int square(int num)
{
    return num * num;
}
```

## CMakeLists

---

```

# CMake configuration for square
# Author: Robin Lin
# This is a simple library used to calculate square of a double number

cmake_minimum_required(VERSION 3.12 FATAL_ERROR)

# Add building options in CMake GUI
option(BUILD_STATIC_SQUARE "Build static library for square library" ON)
option(BUILD_SHARED_SQUARE "Build shared library for square library" OFF)

if(BUILD_STATIC_SQUARE)
    add_library(square STATIC square.cpp square.h) # STATIC build
endif()

if(BUILD_SHARED_SQUARE)
    set(CMAKE_WINDOWS_EXPORT_ALL_SYMBOLS ON) # Must set for Visual Studio to build lib
    add_library(square SHARED square.cpp square.h) # SHARED build
endif()

```

## 主程式

### main.cpp

```

#include <cstdlib>
#include <nana/gui.hpp>
#include <nana/gui/widgets/label.hpp>
#include <nana/gui/widgets/button.hpp>
#include <nana/gui/widgets/textbox.hpp>
#include <fmt/format.h>
// Include custom library
#include "square.h"
int main()
{
    using namespace nana;

    //Define a form.
    form fm;

    textbox text = nana::textbox(fm);

    //Define a button and answer the click event.
    button btn{fm, u8"算!"};
    btn.events().click([&fm, &text]{
        auto val = atoi(text.caption().c_str());

        auto msg = msgbox(fm, u8"計算結果");

        // Use custom library here
        (msg << fmt::format("{0}*{0}={1}", val, square(val))).show();
    });
}

```

```

});

fm.events().unload([&fm] {

    auto msg = msgbox(fm, u8"重要問題?", msgbox::yes_no);
    msg<<u8"助教男的帥女的美?";
    do {
        if(msg.show()==msgbox::pick_yes)
            break;
        auto msg2 = msgbox(fm, u8"訊息");
        msg2.icon(msgbox::icon_information);
        (msg2<<u8"對自己要有信心!").show();
    } while(true);

});

//Layout management
fm.div("vert <<<<weight=80% text><><>weight=24<>button><><>");
fm["text"]<<text;
fm["button"] << btn;
fm.collocate();

//Show the form
fm.show();

//Start to event loop process, it blocks until the form is closed.
exec();
}

```

## 主要 CMakeLists

```

cmake_minimum_required(VERSION 3.4)

project(NANATest)

file(GLOB SOURCE src/*.cpp)
file(GLOB HEADER src/*.h)

# Set dll place
set(CMAKE_RUNTIME_OUTPUT_DIRECTORY "${CMAKE_CURRENT_BINARY_DIR}")

set(NANA_PATH "${CMAKE_CURRENT_LIST_DIR}/ext/nana" CACHE PATH "")
set(FMT_PATH "${CMAKE_CURRENT_LIST_DIR}/ext/fmt" CACHE PATH "")
set(SQUARE_PATH "${CMAKE_CURRENT_LIST_DIR}/square" CACHE PATH "")

if (NOT EXISTS "${NANA_PATH}/CMakeLists.txt")
    message(FATAL_ERROR "Please specify valid nana path!")
endif ()

if (NOT EXISTS "${FMT_PATH}/CMakeLists.txt")
    message(FATAL_ERROR "Please specify valid fmt path!")
endif ()

```

```
if (NOT EXISTS "${SQUARE_PATH}/CMakeLists.txt")
    message(FATAL_ERROR "Please specify valid square path!")
endif ()

# Add include file square.h
include_directories("${SQUARE_PATH}")

add_subdirectory(${NANA_PATH})
add_subdirectory(${FMT_PATH})
add_subdirectory(${SQUARE_PATH})

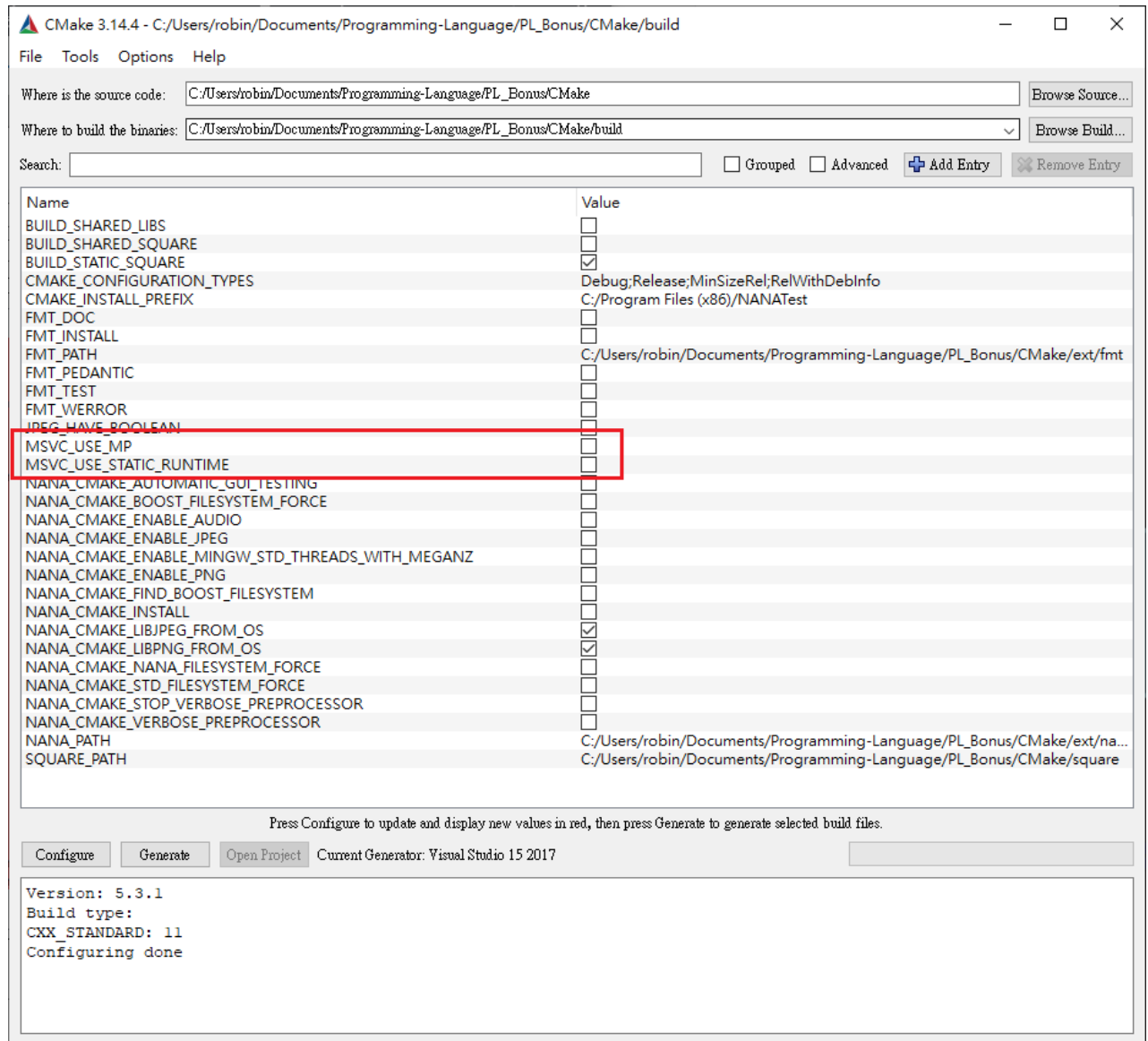
add_executable(NANATest ${SOURCE} ${HEADER})
target_link_libraries(NANATest nana)
target_link_libraries(NANATest fmt)
target_link_libraries(NANATest square)
```

## 使用說明

---

### 1. 打開 CMake GUI

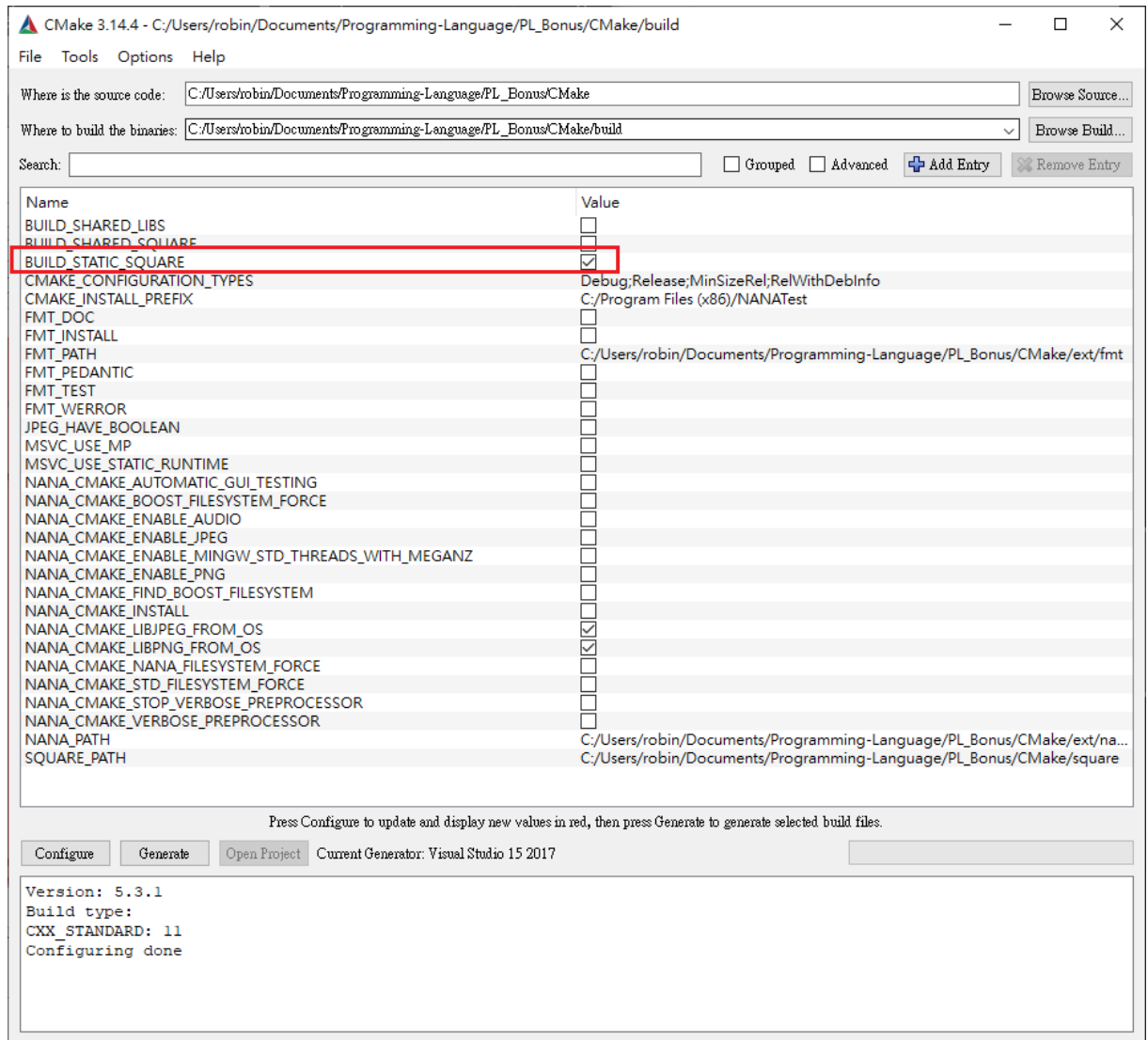
- 設定 source code 目錄
- 設定 binary 目錄
- 進行 configure
  - 選擇 Visual Studio 15 2017
  - 選擇 x64
- 取消勾選 MSVC\_USE\_MP MSVC\_USE\_STATIC\_RUNTIME



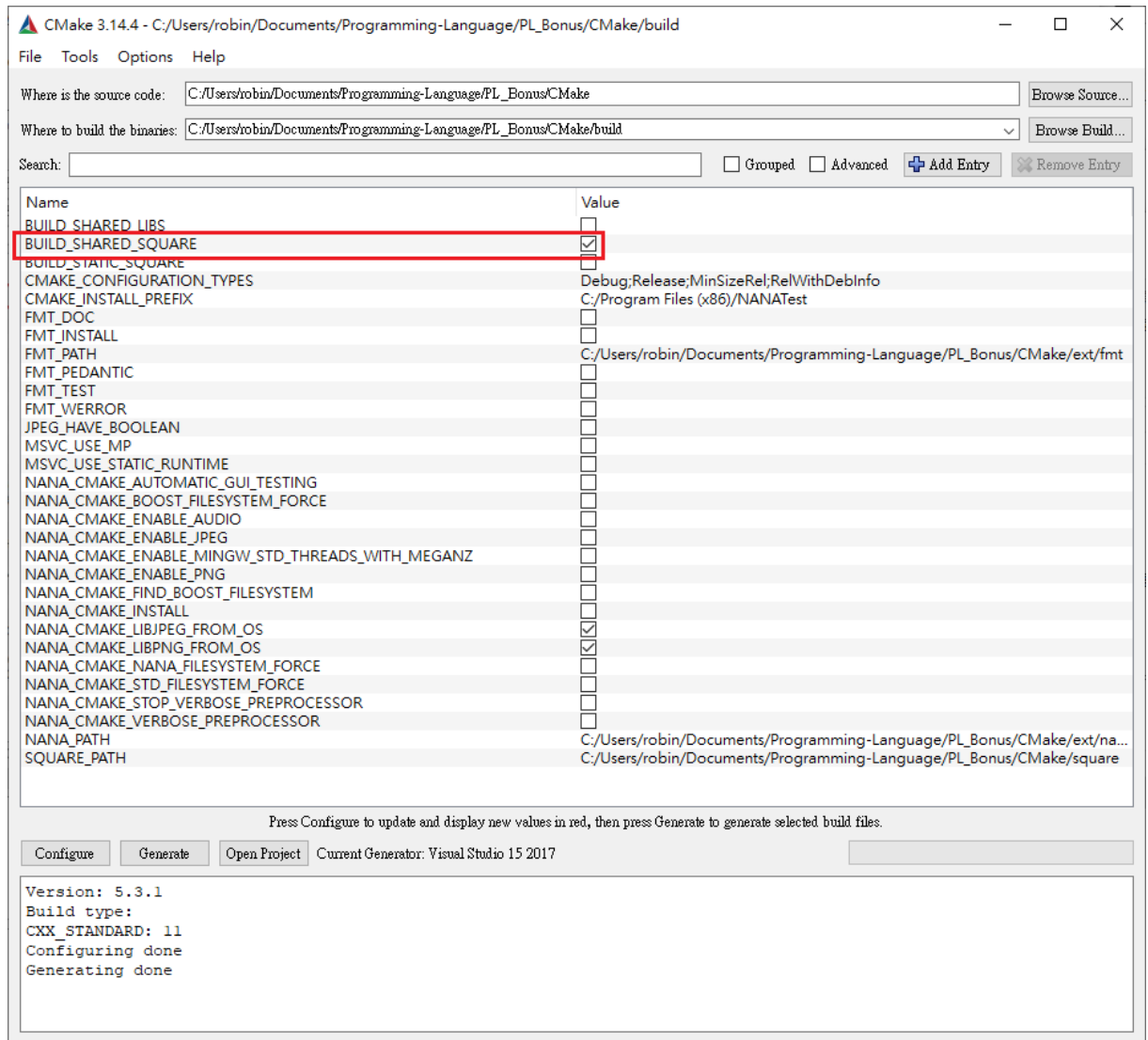
## 2. 選擇 Build 方式

### • STATIC BUILD

- 選擇 BUILD\_STATIC\_SQUARE (default)



- SHARED BUILD
  - 選擇 BUILD\_SHARED\_SQUARE



### 3. 建立 Solution file

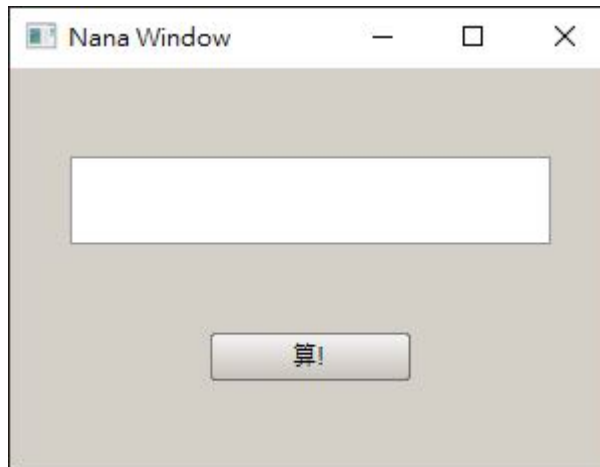
- 點擊 Generate
- 點擊 Open Project

### 4. 使用 Visual Studio 編譯

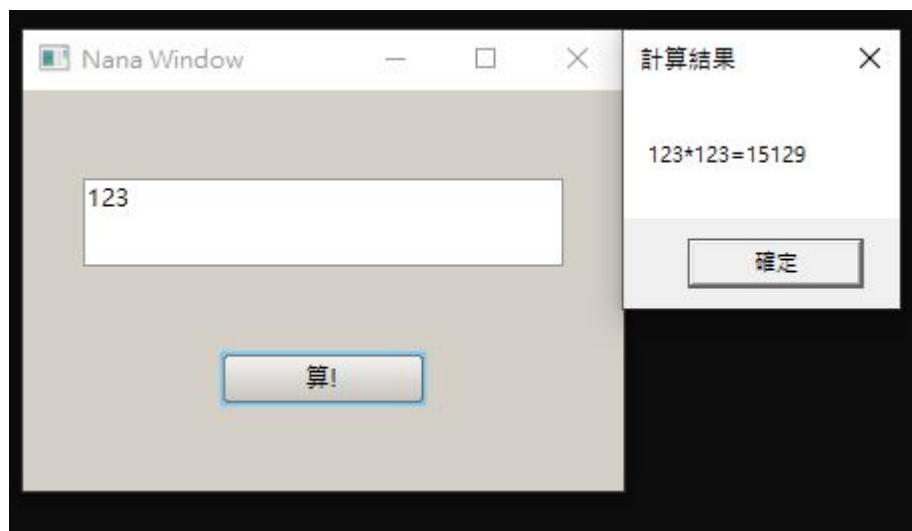
- 點擊 本機 windows 偵錯工具

### 5. 執行

1. 輸入一個數字



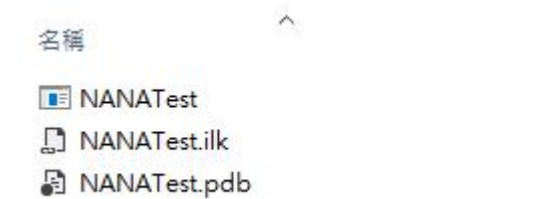
2. 返回數字平方後的結果



## STATIC BUILD vs. SHARED BUILD

觀察執行檔目錄 `Debug/` 可發現 STATIC BUILD 和 SHARED BUILD 的不同

- STATIC BUILD 只有單純的執行檔



- SHARED BUILD 則多出 dll 檔

