


## 在VS2022 MFC程序中调用CUDA代码

 MSVC 同时被 3 个专栏收录 ▾

0 订阅    42 篇文章    订阅专栏

### 在VS2022 MFC程序中调用CUDA函数

#### Pre: 安装好CUDA后VS中该有的效果

首先，假设你已经安装好了CUDA，并且成功集成在VS2022中（即新建项目有CUDA项目的选项，如下图所示）。

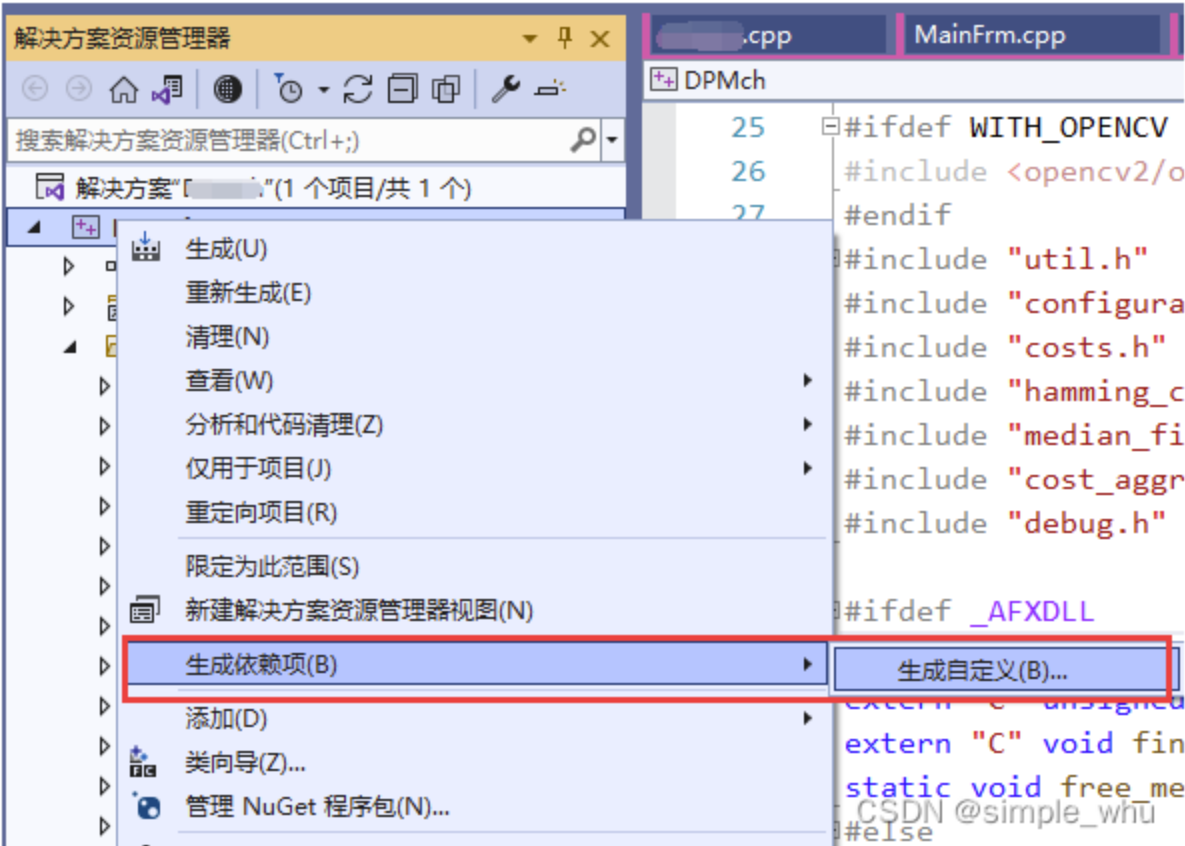


你已经有一个MFC项目和一份CUDA项目，且都可以成功编译运行。  
目标是，在MFC项目中调用CUDA项目中的函数：例如，想在MFC程序中点击某个按钮以调用GPU处理程序。  
那么，除了新增一个按钮控件并添加处理程序之外，还要将CUDA函数集成到MFC项目中。

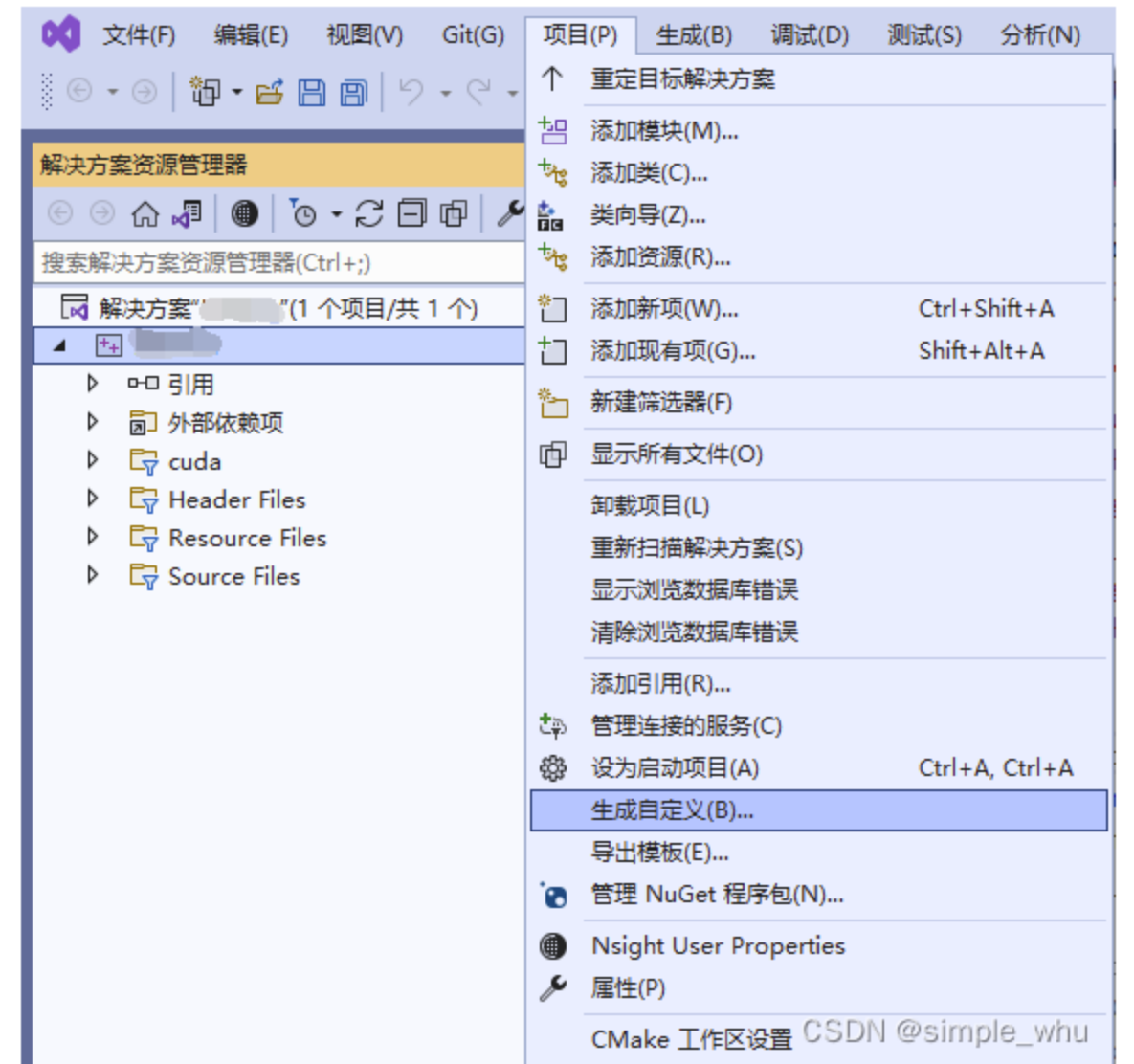
#### 将CUDA函数集成到MFC项目中

1. 为项目添加CUDA配置

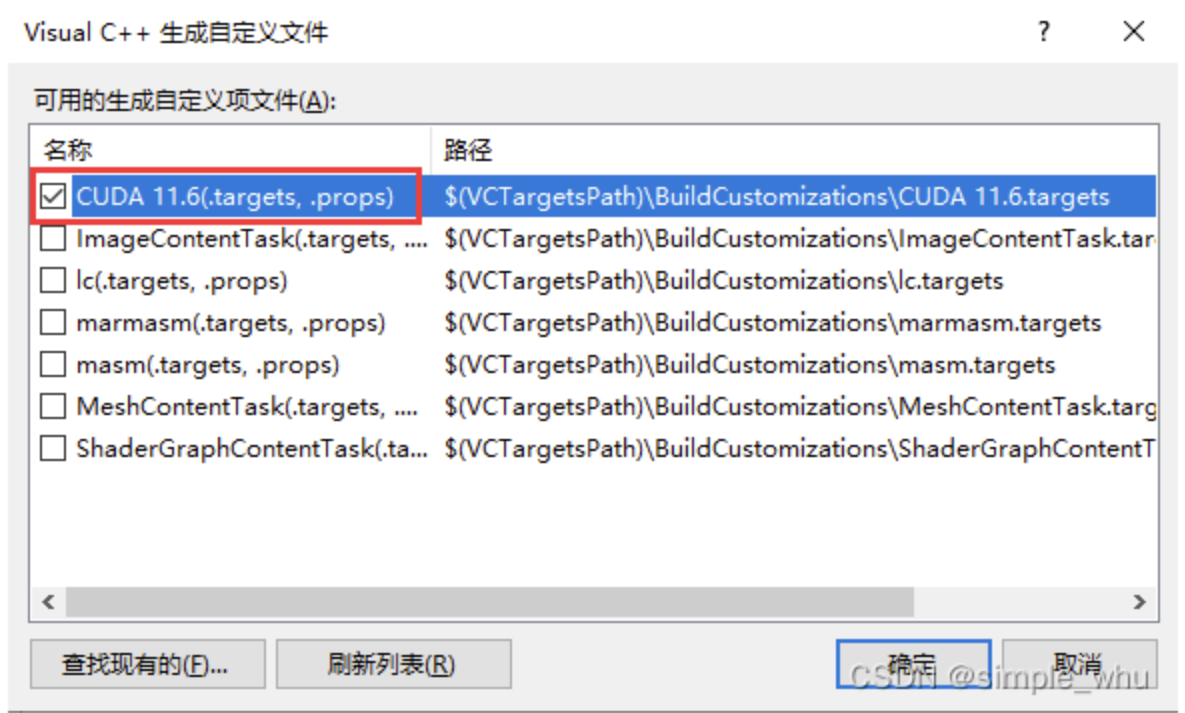
方法一：右键单击MFC项目，选中“生成依赖项—生成自定义”，如图所示。



方法二：单击菜单栏，选择“生成自定义”，如图所示。



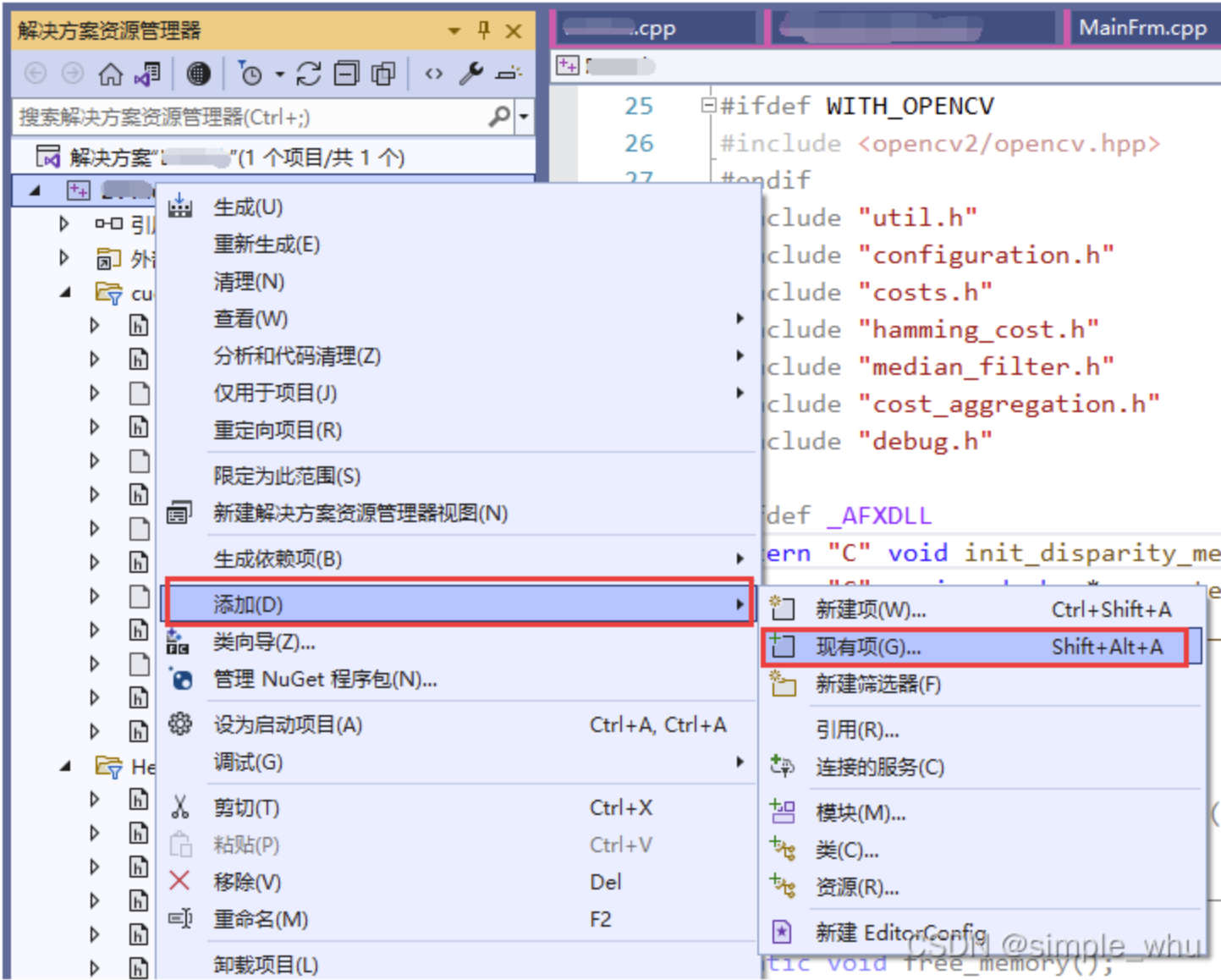
均会弹出“Visual C++生成自定义文件”对话框。如下图所示，勾选CUDA那一项，点击“确定”即可。



2. 把cuda代码添加到项目中

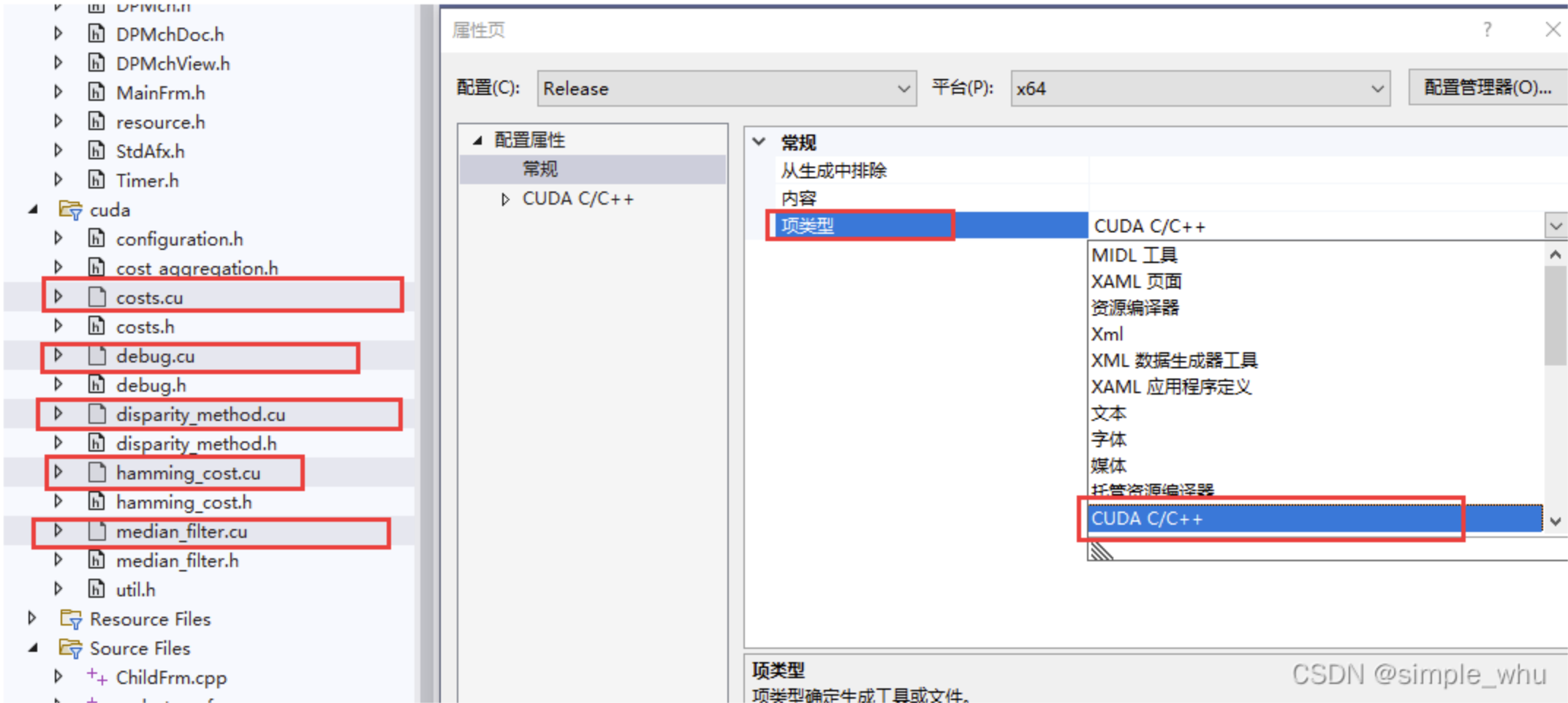
首先，将所有cuda项目的源代码复制到MFC项目文件夹下，并全部添加到MFC项目中。

- 添加方式1：可以如下图所示，右键单击项目，选择“添加-现有项”，在弹出的文件选择对话框中选中要添加的文件（可多选）。
- 添加方式2：在文件资源管理器中选中要添加的文件（可多选），拖动到 解决方案资源管理器 的目标项目中。

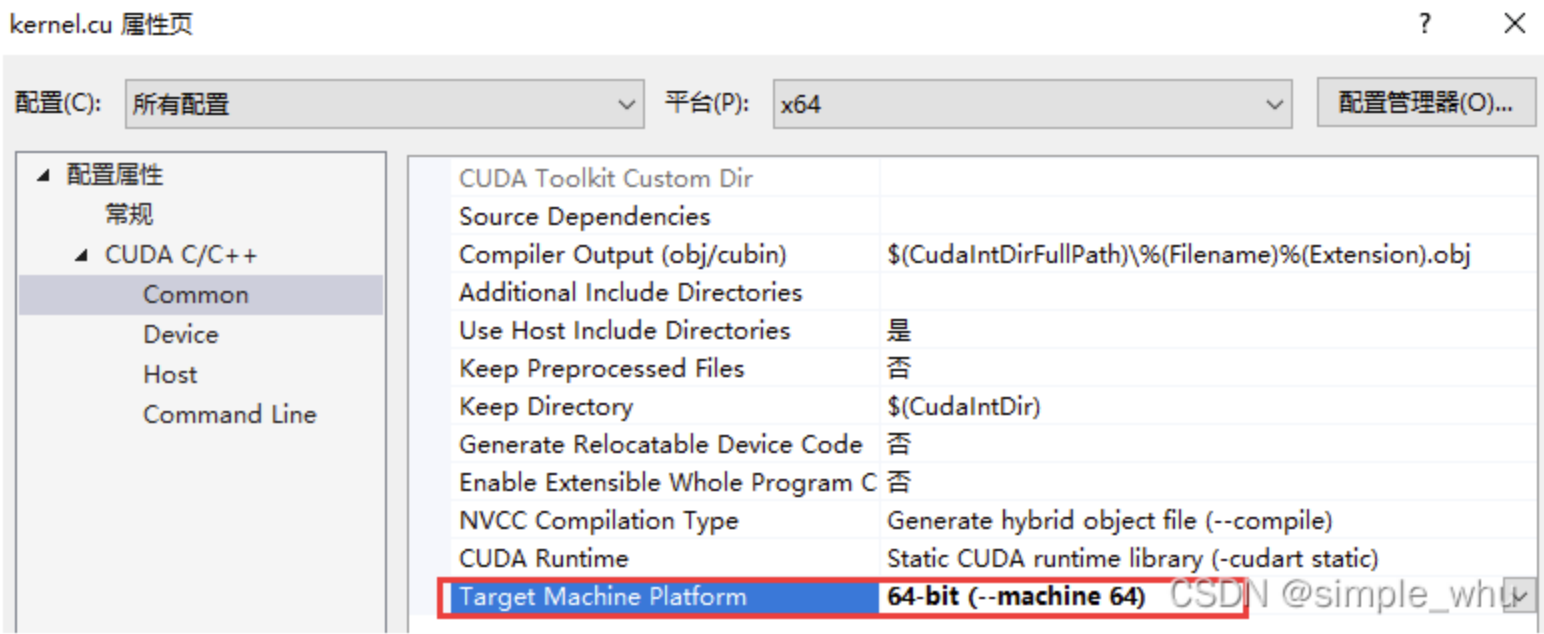


添加完成后，在 解决方案资源管理器 中选中这些 .cu 代码，单击右键菜单选中“属性”，如下图所示，选择“项类型”为“CUDA C/C++”（这样，在生成过程会调用nvcc编译器将它们编成.obj文件）；

如果无法选择‘项类型’为‘CUDA C/C++’，则有可能是第一步没做好。



并且，选择生成64位平台代码，如下图所示。



3. 导出想调用的cuda函数定义

假定要调用的GPU处理函数为 `void cuda_function_tobecalled(int number);`，声明在'do\_somethings.cu'中。

对代码作如下编辑：

```
1 extern "C" void cuda_function_tobecalled(int n);// 在函数声明前添加'extern "C" '以导出定义
2 省略部分
3
4 void cuda_function_tobecalled(int n)
5 {
6     // 省略, gpu运算代码
7 }
8
```

4. 创建CUDA的调用接口 (非必须)

如果不想使用cuda\_transfer这一套, 可以直接在调用 cuda\_function\_tobecalled 的函数的定义之前添加上 extern "C" 语句导入定义。

添加一个 cuda\_transfer.h 和一个 cuda\_transfer.cpp 到项目中。

```
1 // cuda_transfer.h
2 // 不能在这里包含cuda相关的头文件
3 int cuda_function_caller(int n, double& time_elapsed);
4
```

```
1 // cuda_transfer.cpp
2 // 不能在这里包含cuda相关的头文件
3 #include "cuda_transfer.h"
4 #include<chrono>
5 #include<iostream>
6 using namespace std;
7
8 extern "C" cuda_function_tobecalled(int n);// 将要被调用的cuda函数, 在它声明的位置被导出, 在这里被导入。
9
10
11 void cuda_function_caller(int n, double& time_elapsed)
12 {
13     chrono::time_point<chrono::high_resolution_clock> _start, _end;
14     _start = chrono::high_resolution_clock::now();
15     cuda_function_tobecalled(n);
16     _end = chrono::high_resolution_clock::now();
17     time_elapsed = chrono::duration_cast<chrono::microseconds>(_end - _start).count();//计时结束
18 }
19
```

后续其他cuda\_transfer.h可以被其他文件包含, 方便使用。  
至此, 配置已完成, 生成MFC应用程序, 即可运行。

给出没做第四步的示例[VS2022工程](#)