CUDA Installation

Jen-Hao Chen

March 28, 2016

CUDA Installation

Goal :

- **Win 10**

  + Visual Studio community 2013

  + CUDA 7.5

- **OR : Win 7 + VS 2010 + CUDA 6.5**

1. System Requirements

To use CUDA on your system, you will need the following installed:

- A CUDA-capable GPU
- A supported version of Microsoft Windows
- A supported version of Microsoft Visual Studio
- the NVIDIA CUDA Toolkit

1. System Requirements

Table 1. Windows Operating System Support in CUDA 7.5

Operating System	Native x86_64	Cross (x86_32 on x86_64)
Windows 10	YES	YES
Windows 8.1	YES	YES
Windows 7	YES	YES
Windows Server 2012 R2	YES	YES
Windows Server 2008 R2	YES	YES

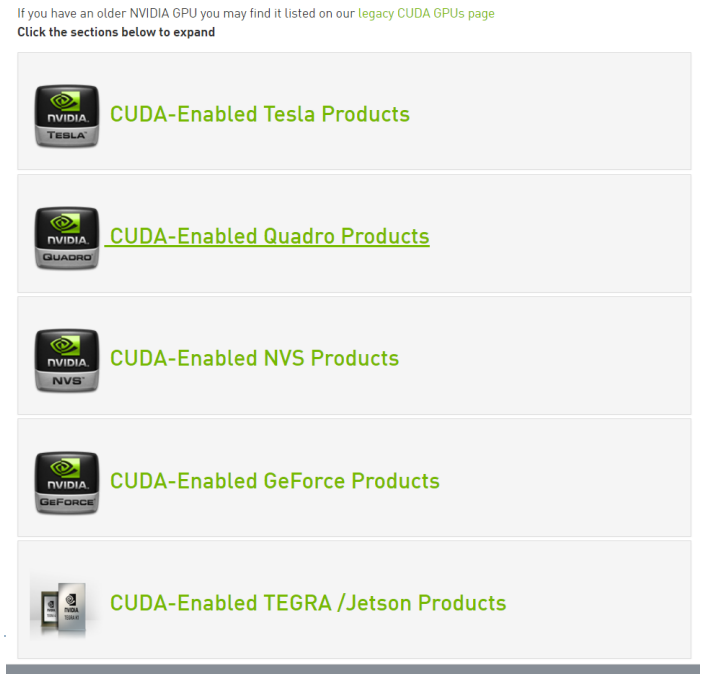
Table 2. Windows Compiler Support in CUDA 7.5

Compiler	IDE	Native x86_64	Cross (x86_32 on x86_64)
Visual C++ 12.0	Visual Studio 2013	YES	YES
	Visual Studio Community 2013	YES	NO
Visual C++ 11.0	Visual Studio 2012	YES	YES
Visual C++ 10.0 **DEPRECATED**	Visual Studio 2010	YES	YES

2. Installing CUDA Development Tools

Verify the system has a CUDA-capable GPU

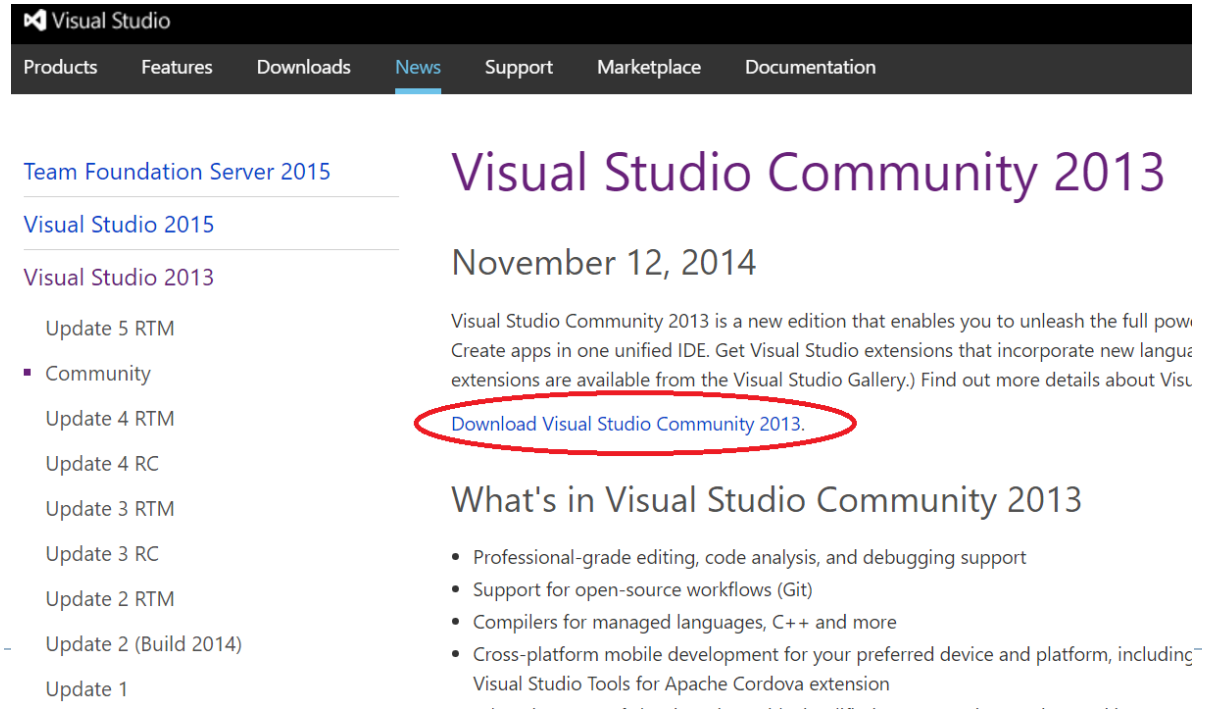
- If you have an NVIDIA card that is listed in http://developer.nvidia.com/cuda-gpus, that GPU is CUDA-capable.



2. Installing CUDA Development Tools

Step.1 Install Visual Studio community 2013 (free)

- Download from https://www.visualstudio.com/en-us/news/vs2013-community-vs.aspx



2. Installing CUDA Development Tools

Step.2 Download driver that supports CUDA 7.5

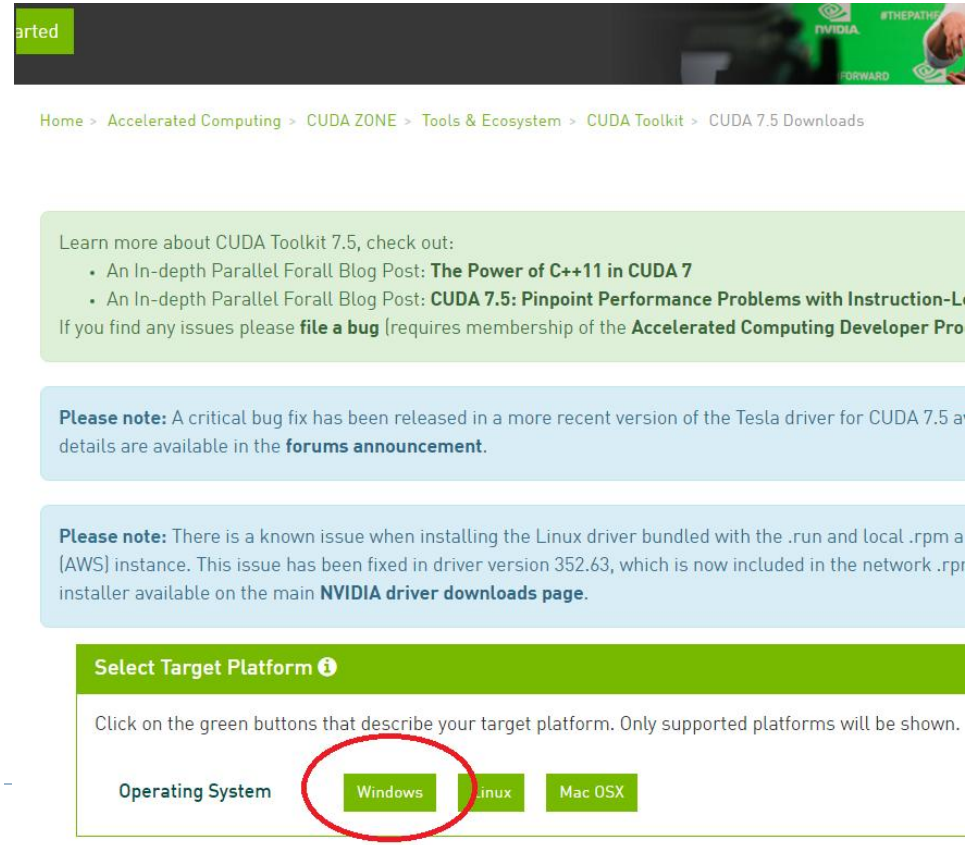
- Search keywords <GPU name, driver and CUDA 7.5>

  or go to

2. Installing CUDA Development Tools

Step.3 Download and install CUDA 7.5

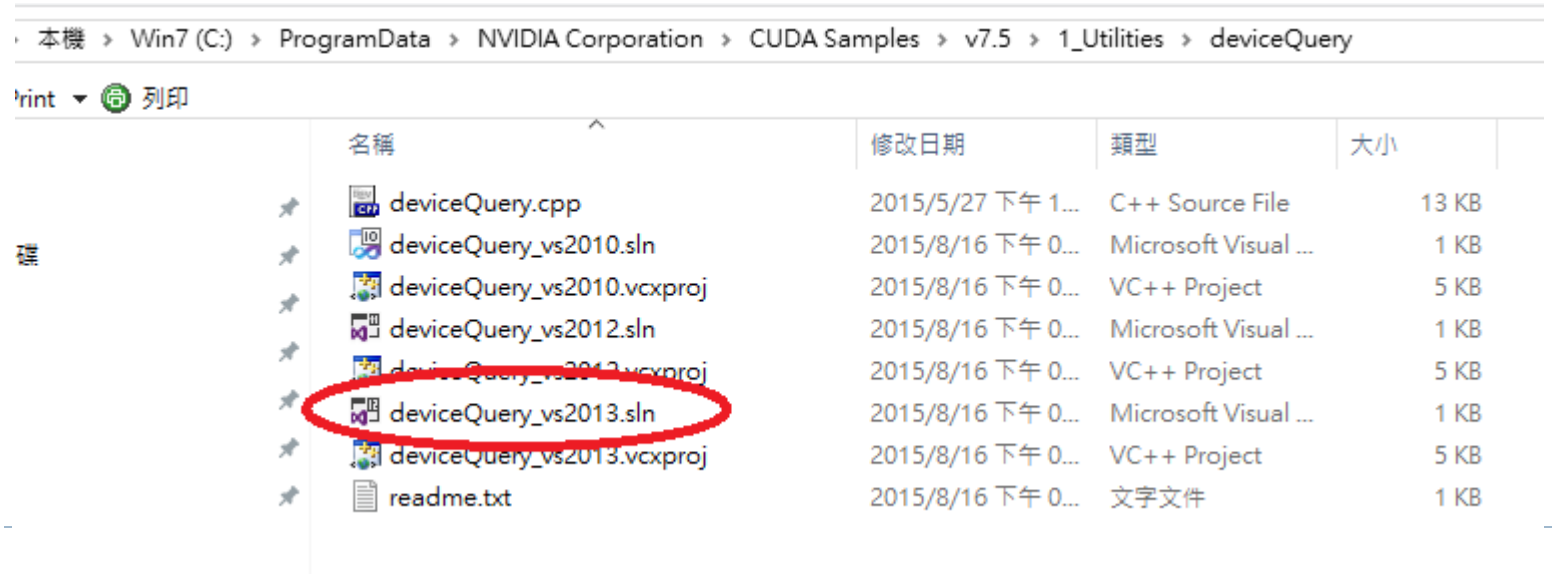
- https://developer.nvidia.com/cuda-downloads



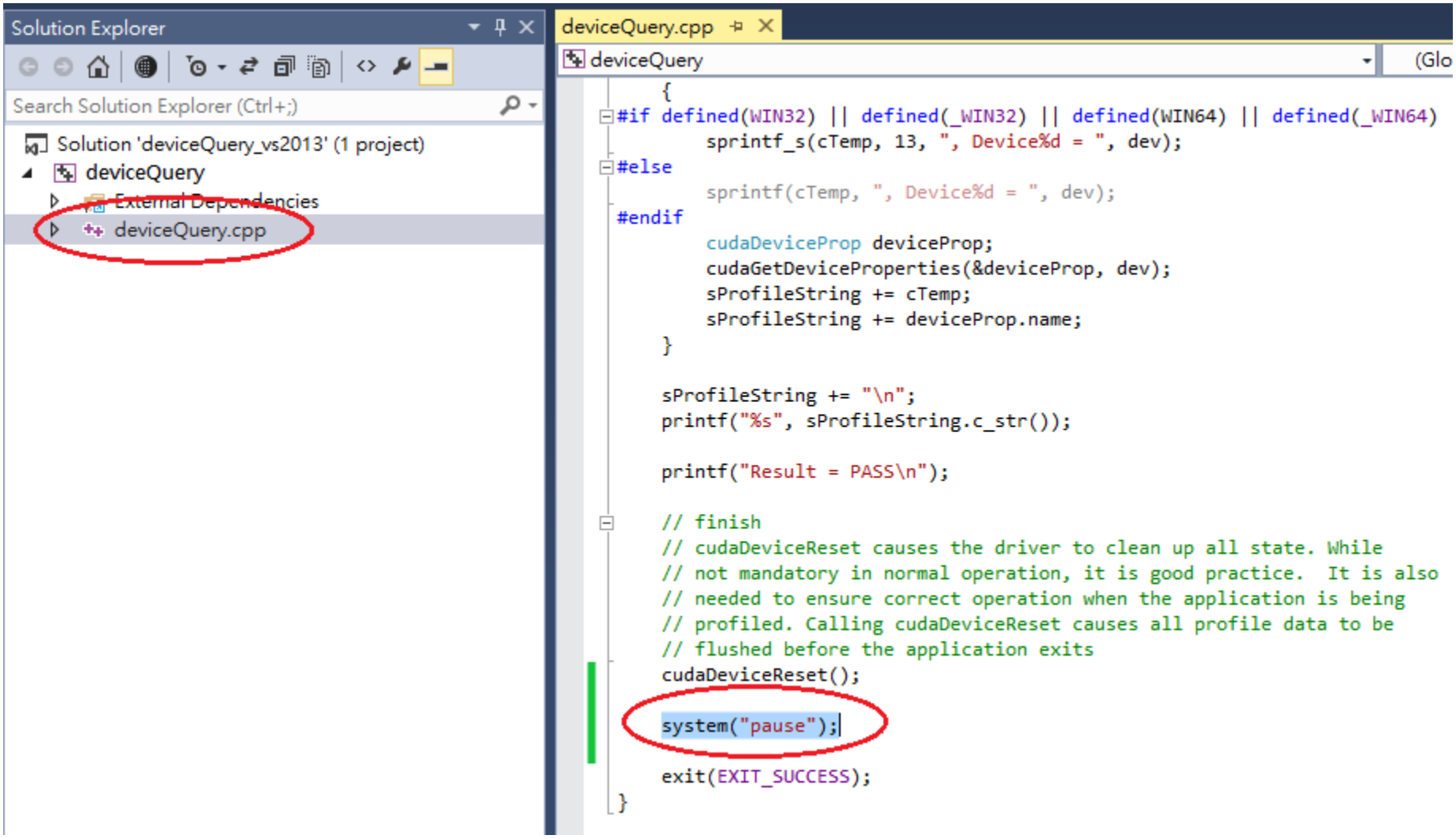
3. Verify the Installation

Running the Compiled Examples

- To verify a correct configuration of the hardware and software, it is highly recommended that you run the **deviceQuery** program located at **C:\ProgramData\NVIDIA Corporation\CUDA Samples\v7.5\1_Utilities\deviceQuery**



3. Verify the Installation



Valid Results from deviceQuery CUDA Sample (Quadro K600)

```
 CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 1 CUDA Capable device(s)

Device 0: "Quadro 600"
  CUDA Driver Version / Runtime Version          7.5 / 7.5
  CUDA Capability Major/Minor version number:    2.1
  Total amount of global memory:                 1024 MBytes (1073741824 bytes)
  ( 2) Multiprocessors, ( 48) CUDA Cores/MP:     96 CUDA Cores
  GPU Max Clock rate:                            1280 MHz (1.28 GHz)
  Memory Clock rate:                             800 Mhz
  Memory Bus Width:                              128-bit
  L2 Cache Size:                                 131072 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(65536), 2D=(65536, 65535), 3D=(2048, 2048, 2048)
  Maximum Layered 1D Texture Size, (num) layers  1D=(16384), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(16384, 16384), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 32768
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  1536
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (65535, 65535, 65535)
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 1 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Disabled
  CUDA Device Driver Mode (TCC or WDDM):         WDDM (Windows Display Driver Model)
  Device supports Unified Addressing (UVA):      Yes
  Device PCI Domain ID / Bus ID / location ID:   0 / 1 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 7.5, CUDA Runtime Version = 7.5, NumDevs = 1, Device0 = Quadro 600
Result = PASS
```

Valid Results from deviceQuery CUDA Sample (Quadro M4000)

```
 CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 1 CUDA Capable device(s)

Device 0: "Quadro M4000"
  CUDA Driver Version / Runtime Version          7.5 / 7.5
  CUDA Capability Major/Minor version number:    5.2
  Total amount of global memory:                 8192 MBytes (8589934592 bytes)
  (13) Multiprocessors, (128) CUDA Cores/MP:     1664 CUDA Cores
  GPU Max Clock rate:                            773 MHz (0.77 GHz)
  Memory Clock rate:                             3005 Mhz
  Memory Bus Width:                              256-bit
  L2 Cache Size:                                 2097152 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(65536), 2D=(65536, 65536), 3D=(4096, 4096, 4096)
  Maximum Layered 1D Texture Size, (num) layers  1D=(16384), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(16384, 16384), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  2048
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 2 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Disabled
  CUDA Device Driver Mode (TCC or WDDM):         WDDM (Windows Display Driver Model)
  Device supports Unified Addressing (UVA):      Yes
  Device PCI Domain ID / Bus ID / location ID:   0 / 1 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >

deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 7.5, CUDA Runtime Version = 7.5, NumDevs = 1, Device0 = Quadro M4000
Result = PASS
```

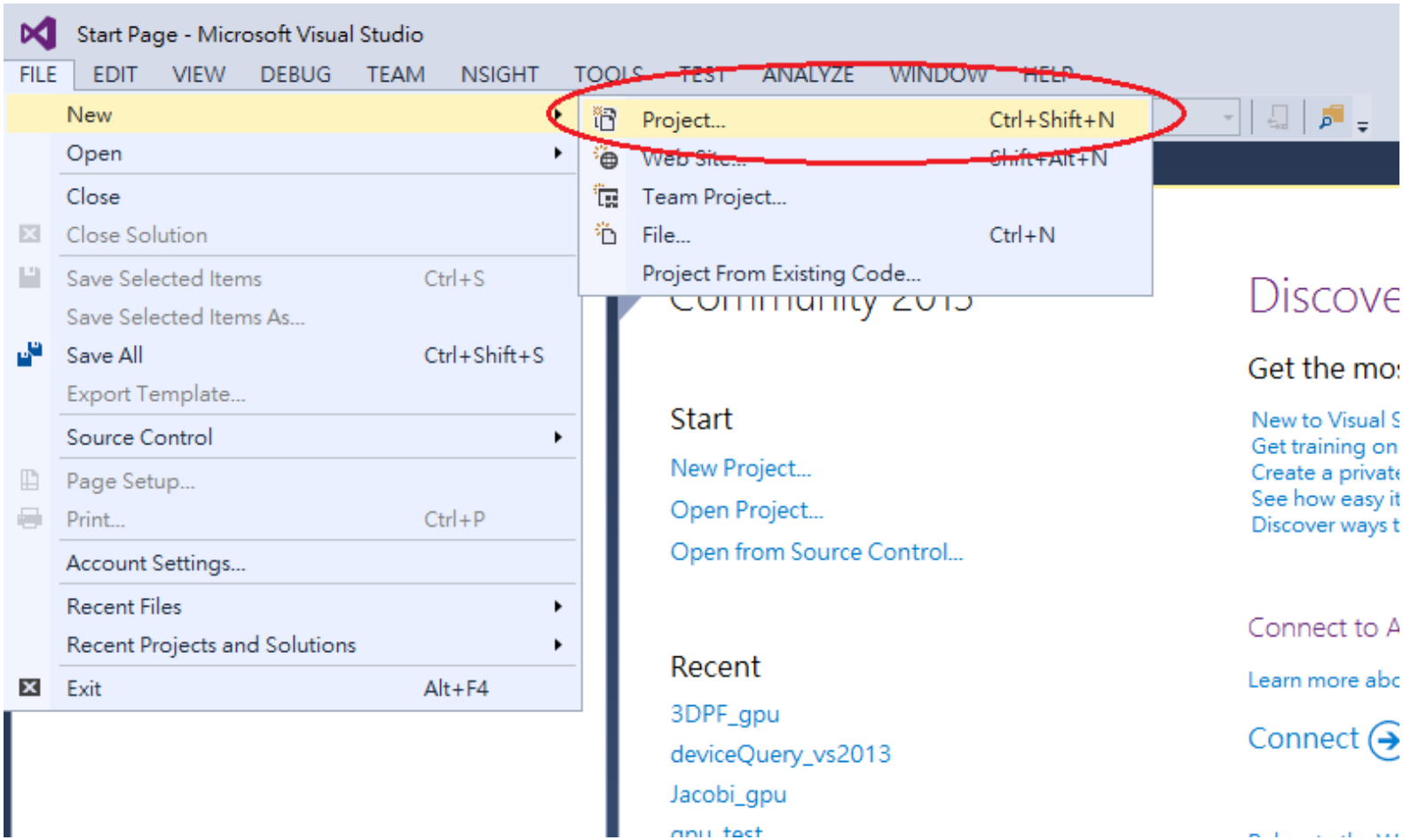
Compiling CUDA Programs

Build Customizations for New Projects

- When creating a new CUDA application, the Visual Studio project file must be configured to include CUDA build customizations. To accomplish this, click File-> New | Project... NVIDIA-> CUDA->, then **select a template** for your CUDA Toolkit version. For example, selecting the **"CUDA 7.5 Runtime" template** will configure your project for use with the CUDA 7.5 Toolkit. The new project is technically a C++ project (.vcxproj) that is preconfigured to use NVIDIA's Build Customizations. All standard capabilities of Visual Studio C++ projects will be available.

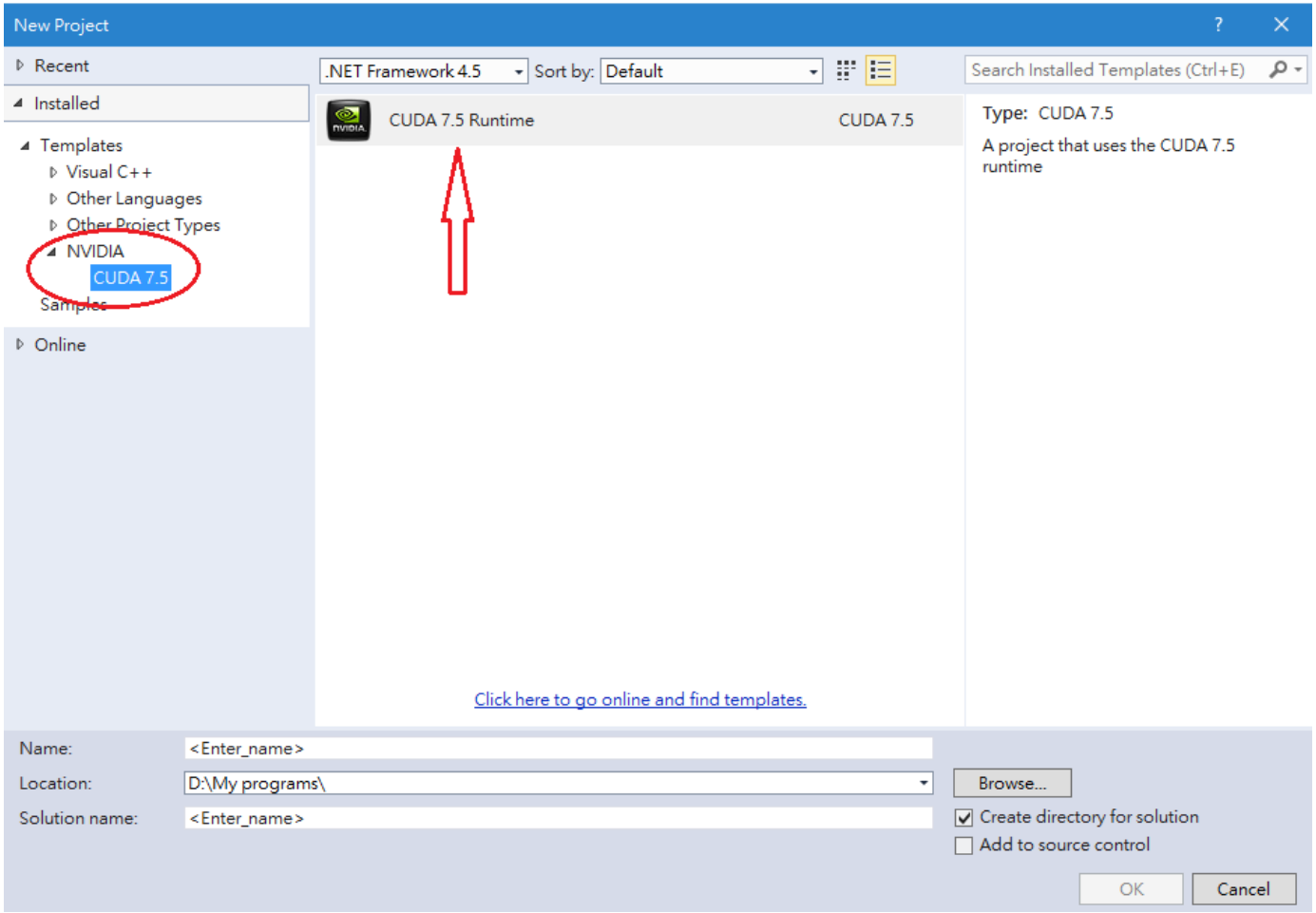
Compiling CUDA Programs

Build Customizations for New Projects



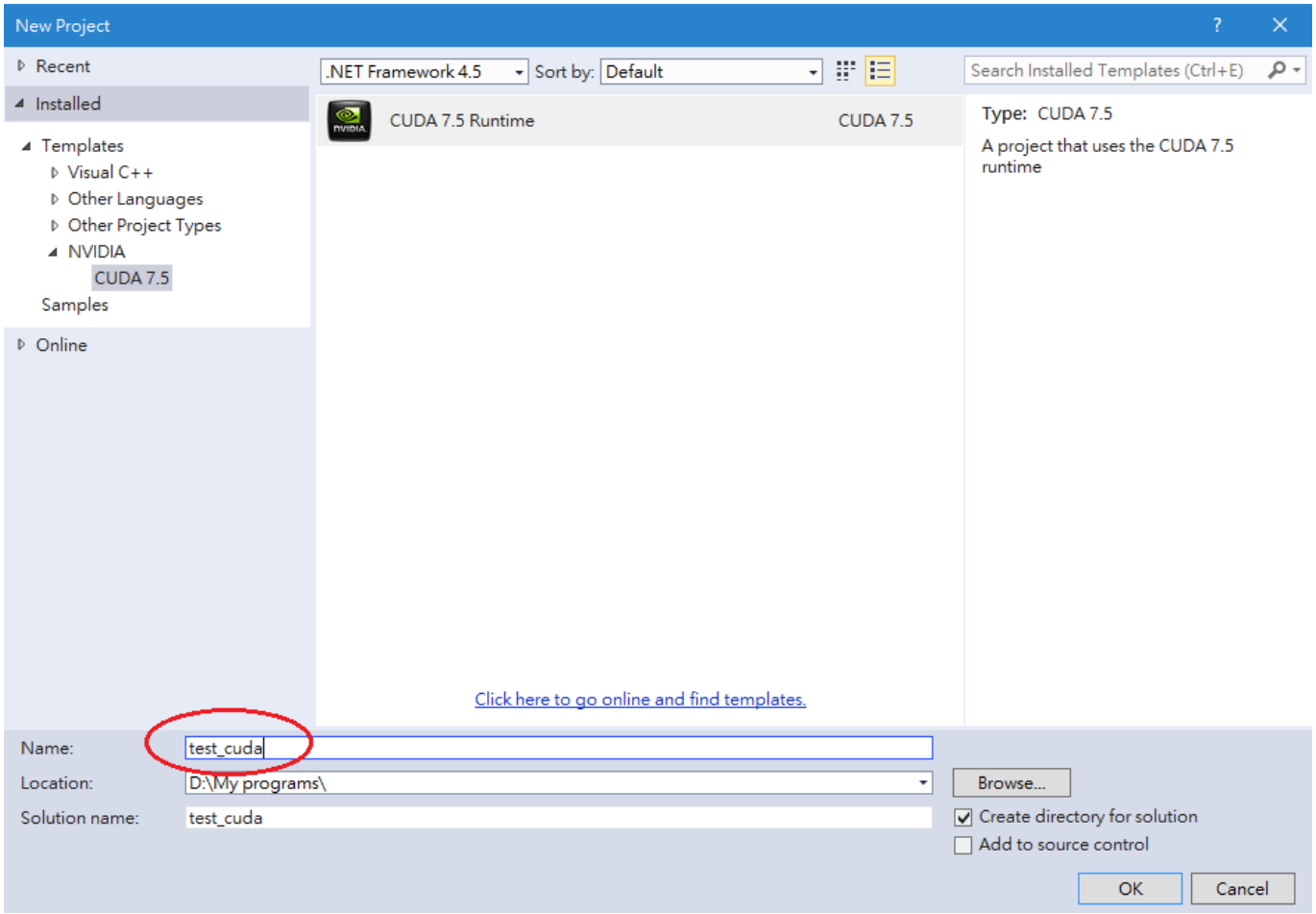
Compiling CUDA Programs

Build Customizations for New Projects



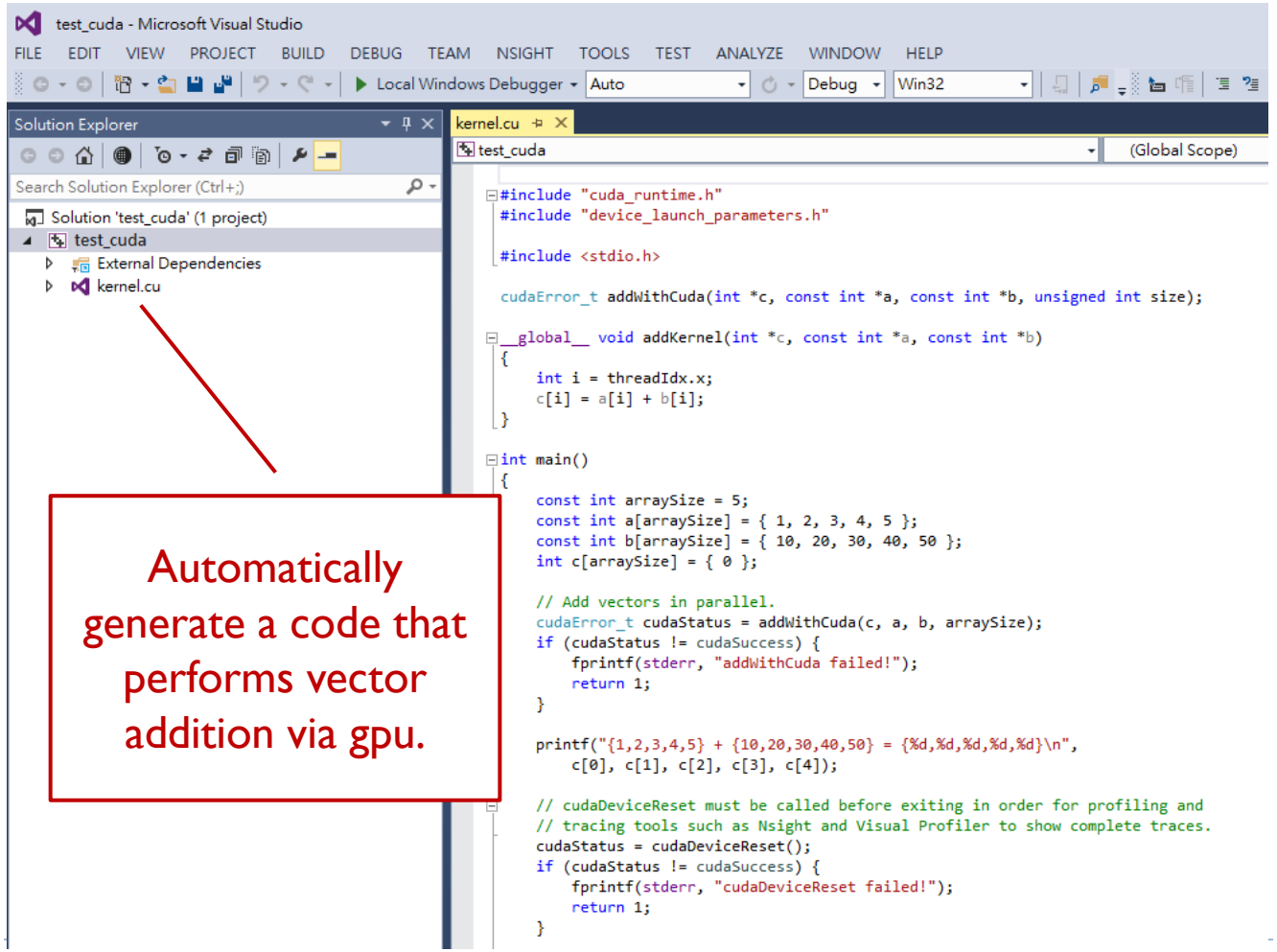
Compiling CUDA Programs

Build Customizations for New Projects



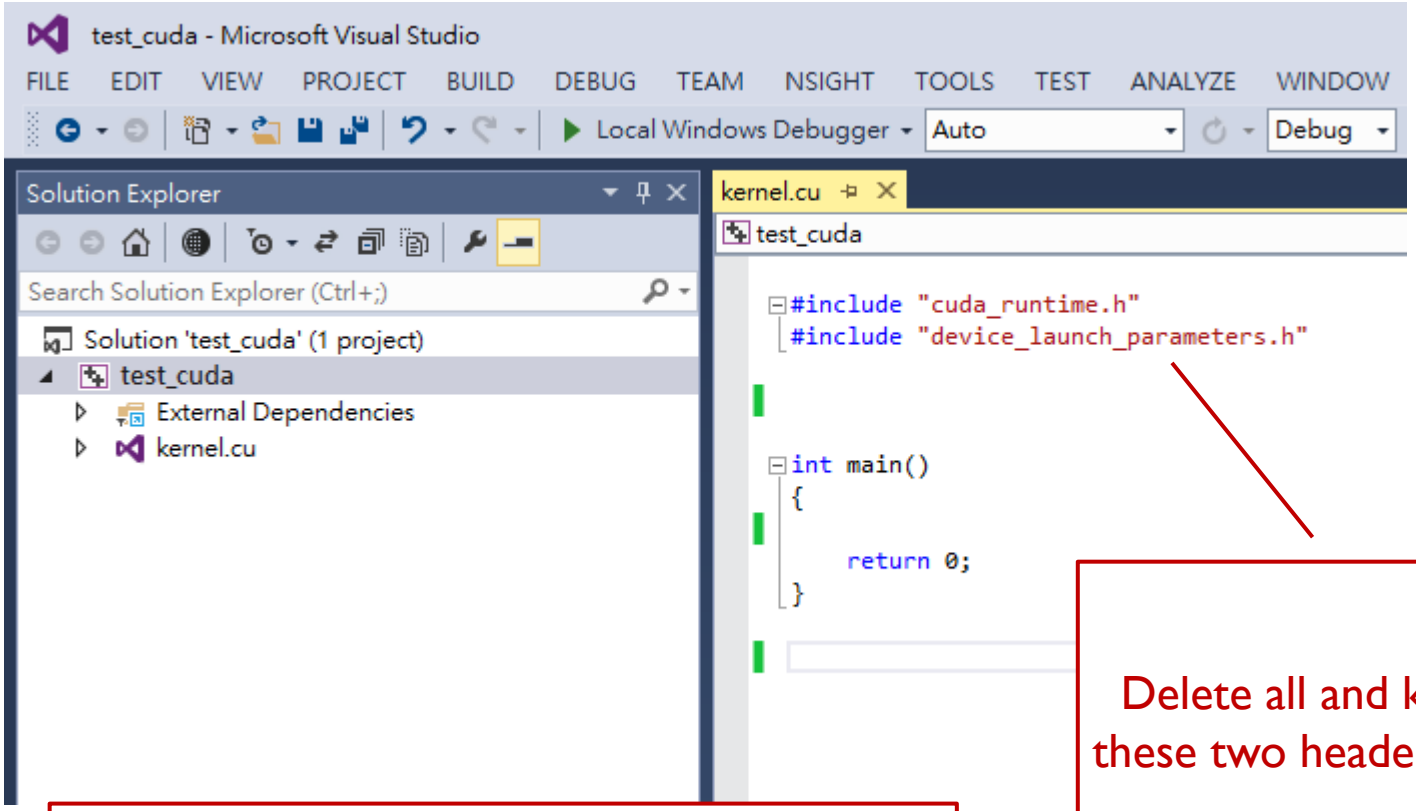
Compiling CUDA Programs

Build Customizations for New Projects



Compiling CUDA Programs

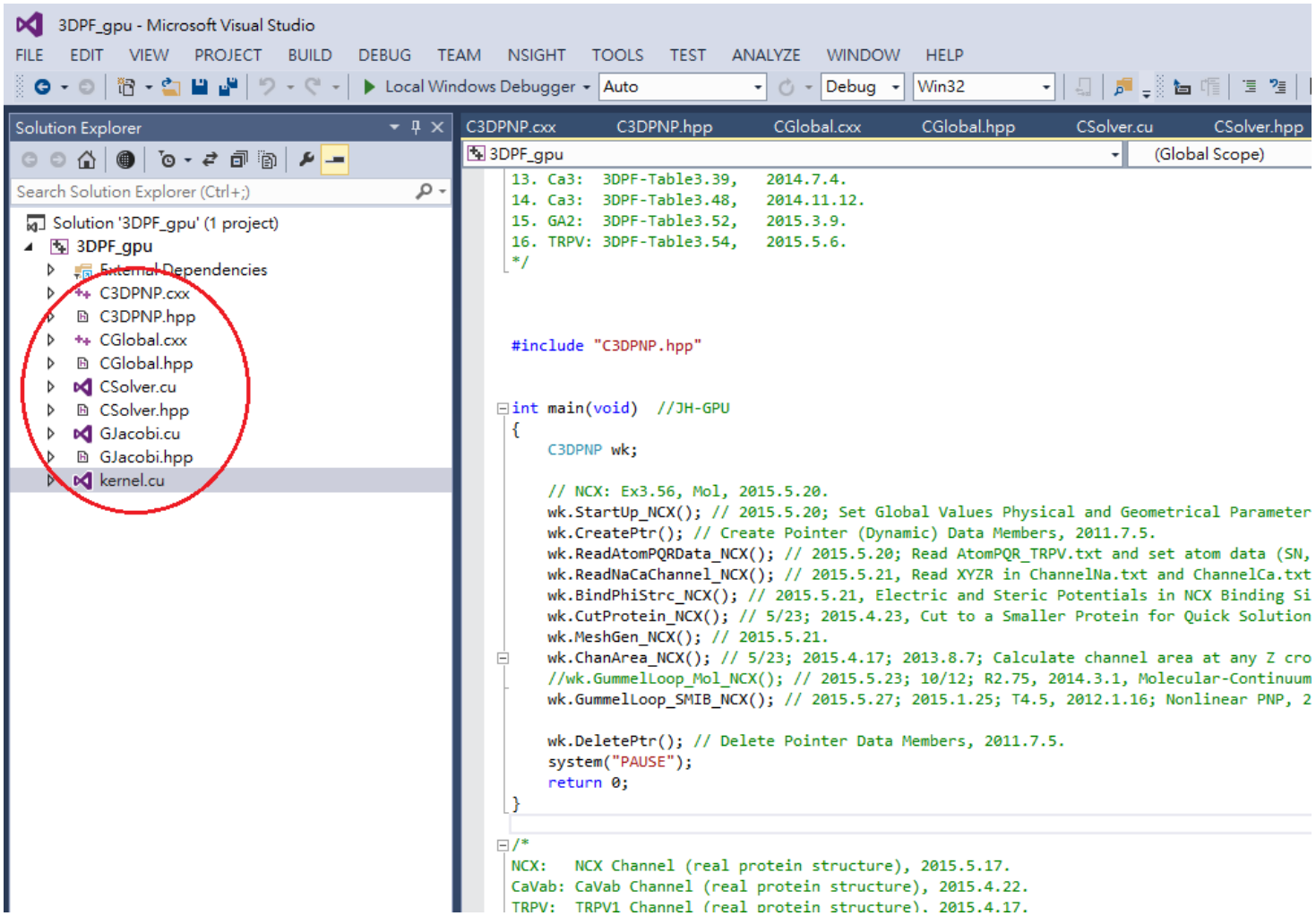
Build Customizations for New Projects



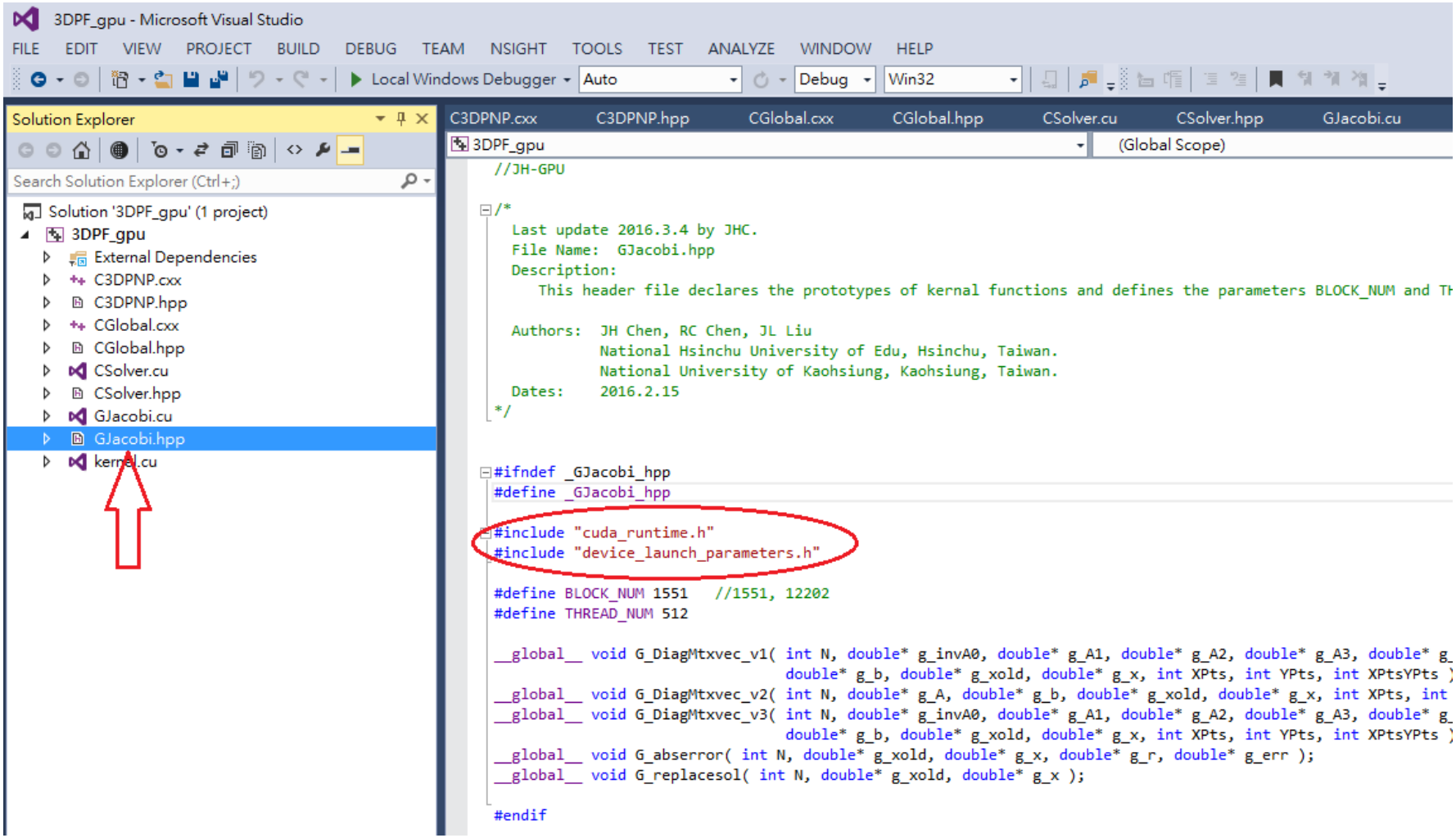
Delete all and keep these two header files.

Remember that these two header files should be put somewhere that gpu codes can see.

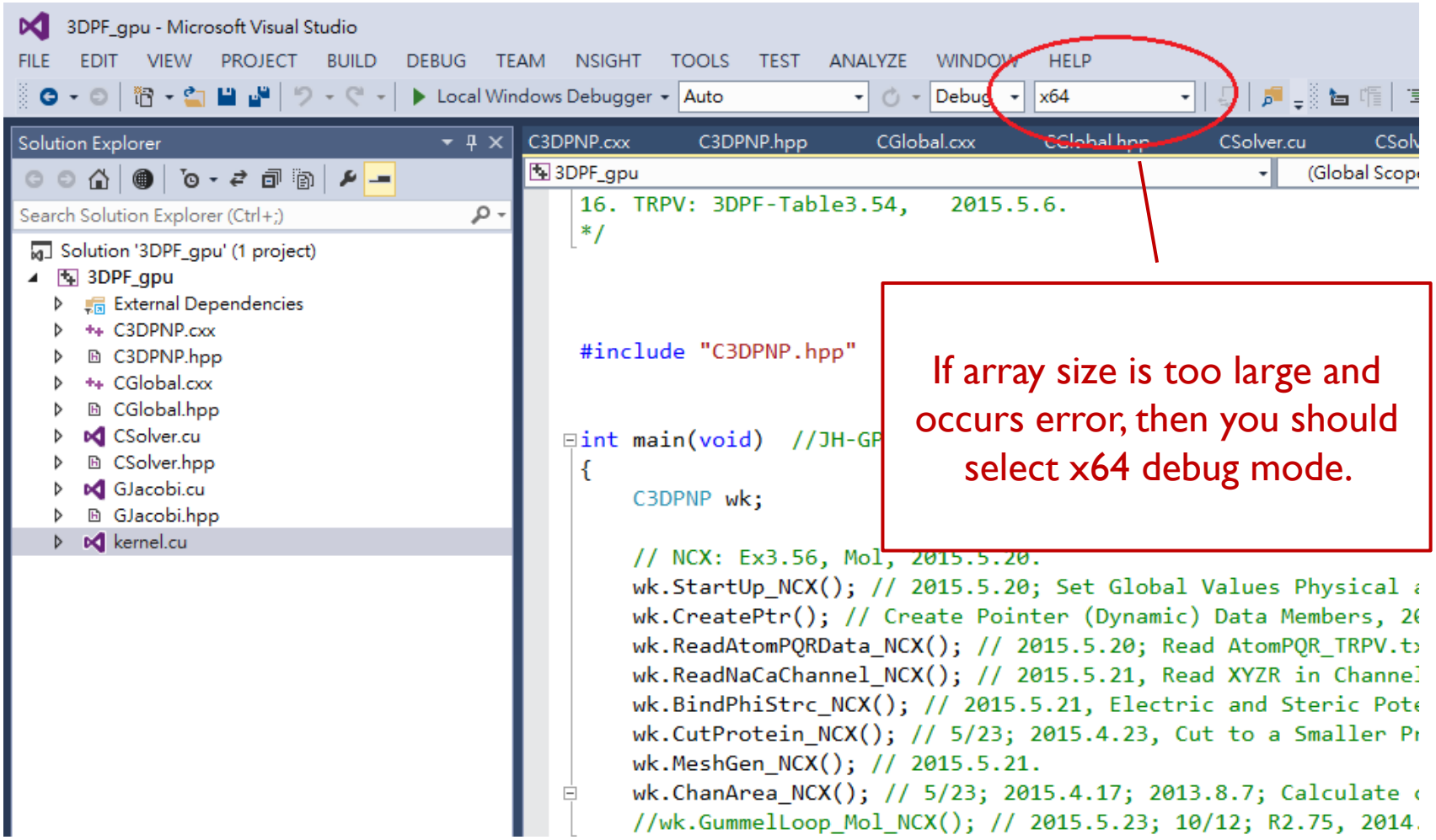
An example: 3DPF_gpu



An example: 3DPF_gpu



An example: 3DPF_gpu



Reference

- NVIDIA **CUDA Installation Guide for Microsoft Windows**
- Microsoft Visual Studio 2013 語言套件 **https://www.microsoft.com/zh-tw/download/details.aspx?id=40783**