

Create CUDA Project For CSC3150 Assignment 4

Environment:

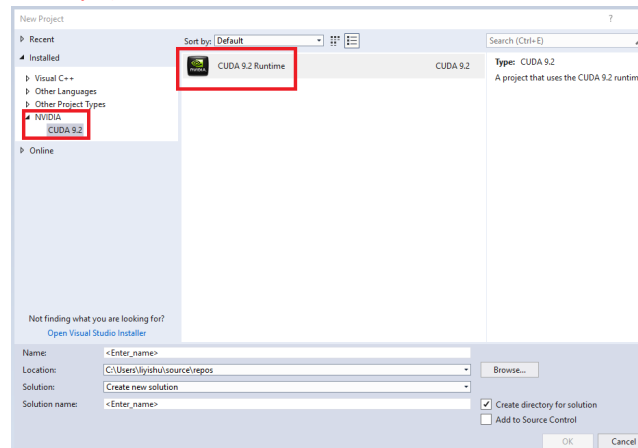
- Win 10
- VS2017
- CUDA 9.2
- NVIDIA GeForce GTX 1060
- Compute capacity: 6.1

Configuration steps:

1. Launch VS2017

2. Create CUDA project

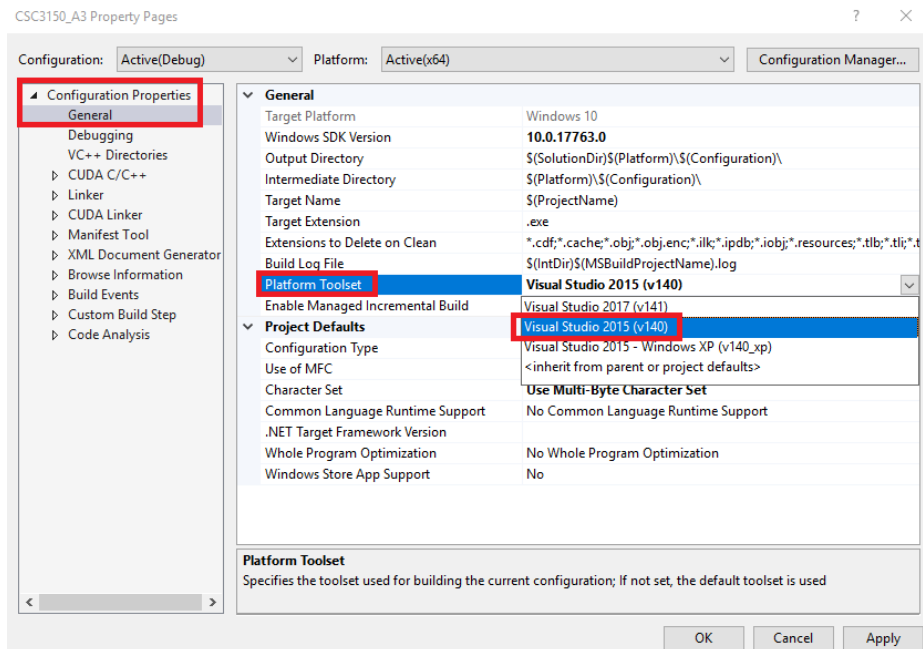
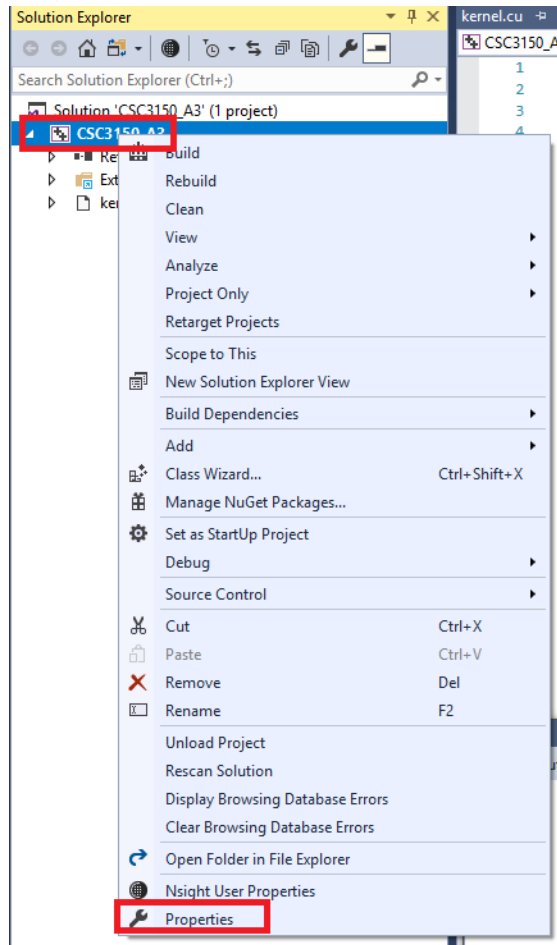
(If you're using PCs in TC301, change the project storing location in this step. Please ensure storing your CUDA project in D disk, rather than C disk, where will auto delete users' data every 30 days.)



3. There will be a “kernel.cu” automatically created.

```
kernel.cu → X
%| CSC3150_A3 | (Global S
1
2 #include "cuda_runtime.h"
3 #include "device_launch_parameters.h"
4
5 #include <stdio.h>
6
7 cudaError_t addWithCuda(int *c, const int *a, const int *b, unsigned
8
9
10 __global__ void addKernel(int *c, const int *a, const int *b)
11 {
12     int i = threadIdx.x;
13     c[i] = a[i] + b[i];
14 }
15
16 int main()
17 {
18     const int arraySize = 5;
19     const int a[arraySize] = { 1, 2, 3, 4, 5 };
20     const int b[arraySize] = { 10, 20, 30, 40, 50 };
21     int c[arraySize] = { 0 };
22
23     // Add vectors in parallel.
24     cudaError_t cudaStatus = addWithCuda(c, a, b, arraySize);
25     if (cudaStatus != cudaSuccess) {
26         fprintf(stderr, "addWithCuda failed!");
27         return 1;
28     }
29
30     printf("{1,2,3,4,5} + {10,20,30,40,50} = {%d,%d,%d,%d,%d}\n",
31           c[0], c[1], c[2], c[3], c[4]);
```

4. Select and right click the project, open its properties. Update the platform toolkit as 'Visual Studio 2015(v140)'

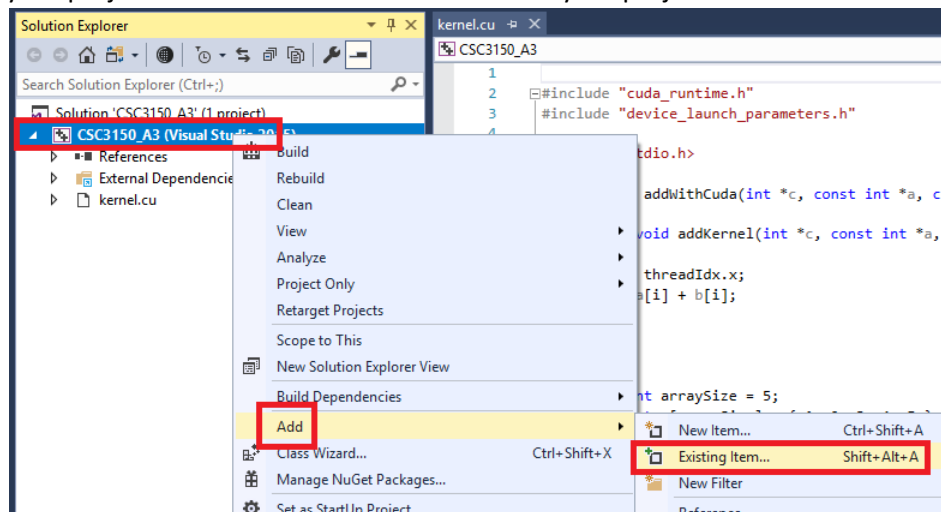


5. Press “Ctrl”+”F5” to run the project. If the calculation results being displayed correctly in console, it means your CUDA project settings is correct.

6. Copy provided source code into your project folder.
(‘main.cu’, ‘user_program.cu’, ‘file_system.cu’, ‘file_system.h’, ‘data.bin’)

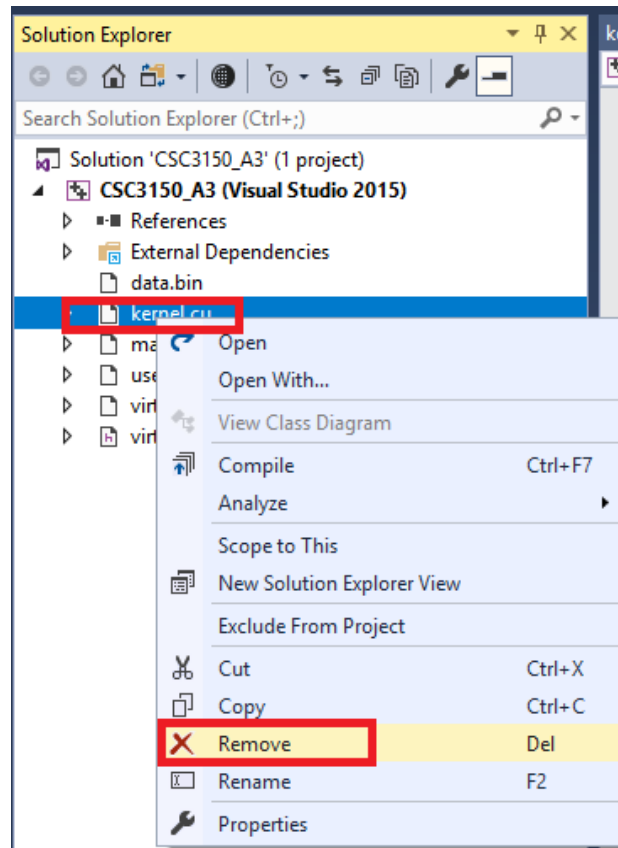
CSC3150_A4	10/28/2019 3:31 PM
CSC3150_A4.vcxproj.user	10/28/2019 3:31 PM
data.bin	1/1/2015 11:31 PM
file_system	10/28/2019 2:42 PM
file_system	10/28/2019 2:41 PM
kernel	10/28/2019 3:31 PM
main	10/28/2019 2:39 PM
user_program	10/28/2019 3:09 PM

7. In Visual Studio, select and right click your project, add exiting items (the source codes) into your project. And remove the ‘kernel.cu’ from your project.



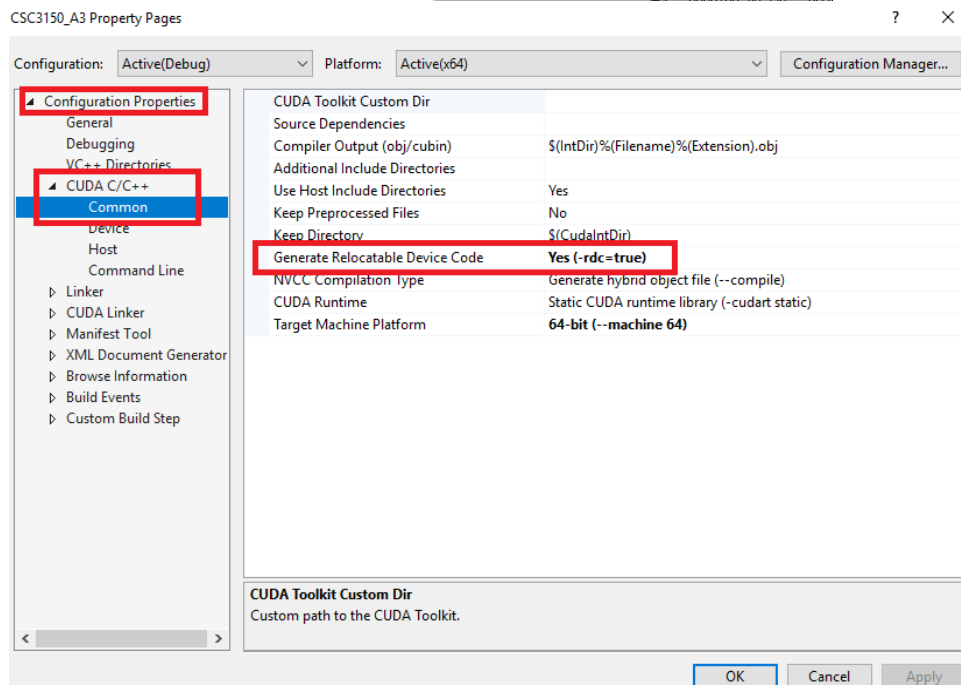
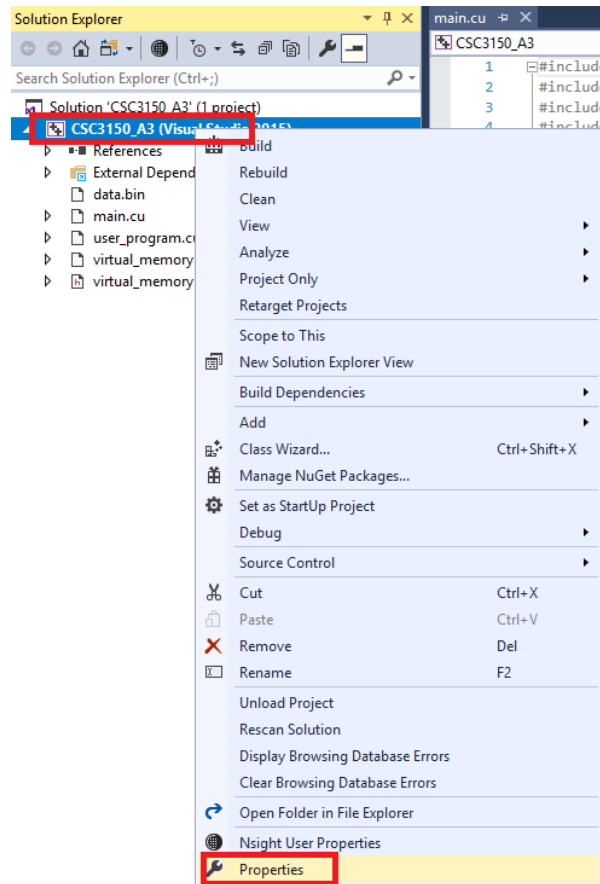
Name

- CSC3150_A4
- CSC3150_A4.vcxproj.user
- data.bin
- file_system
- file_system
- kernel
- main
- user_program



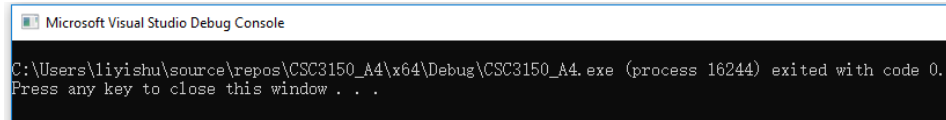
- Solution 'CSC3150_A4' (1 project)
 - CSC3150_A4
 - References
 - External Dependencies
 - data.bin
 - file_system.cu
 - file_system.h
 - main.cu
 - user_program.cu

8. Select the project and right click it. Open its properties, allow the CUDA files to generate relocatable device code.



9. For those three cuda files (.cu), right click to compile them. (Or use 'Ctrl'+F7'). Then press "Ctrl"+"F5" to run your program.

For the provided template, you will get nothing being displayed in output as below:



```
Microsoft Visual Studio Debug Console

C:\Users\liyishu\source\repos\CSC3150_A4\x64\Debug\CSC3150_A4.exe (process 16244) exited with code 0.
Press any key to close this window . . .
```

10. In "file_system.cu", start your programming.