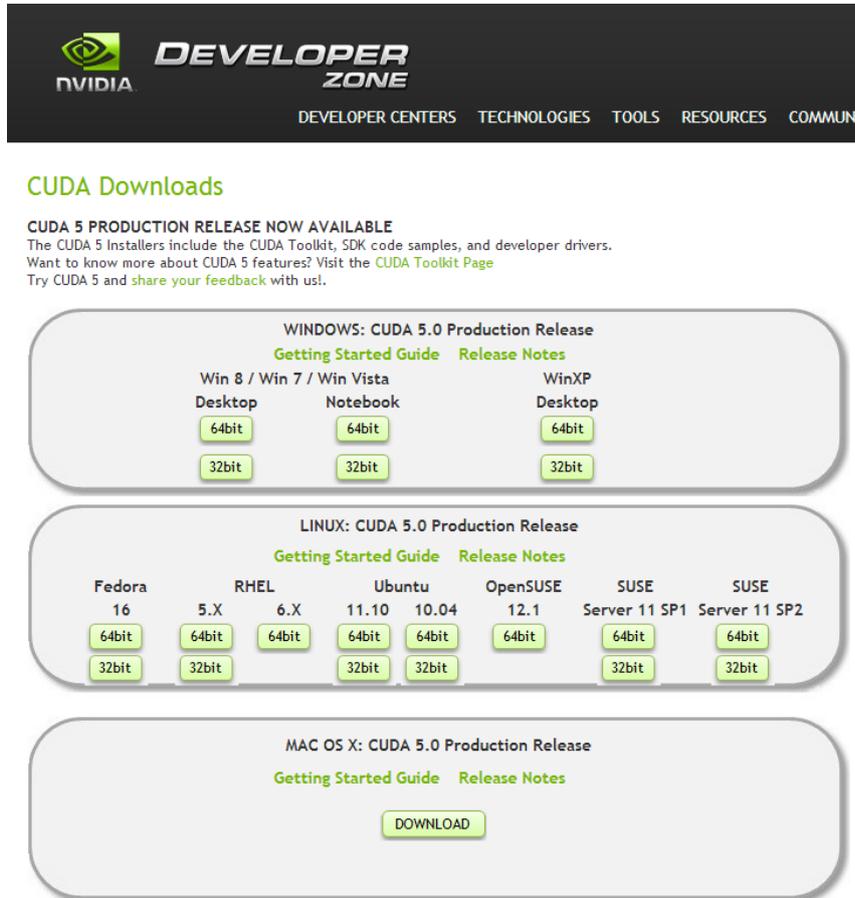


CUDA를 사용한 GPU 프로그래밍 가이드

1. CUDA 툴킷 설치

- ① NVIDIA CUDA ZONE 접속(<https://developer.nvidia.com/cuda-downloads>)

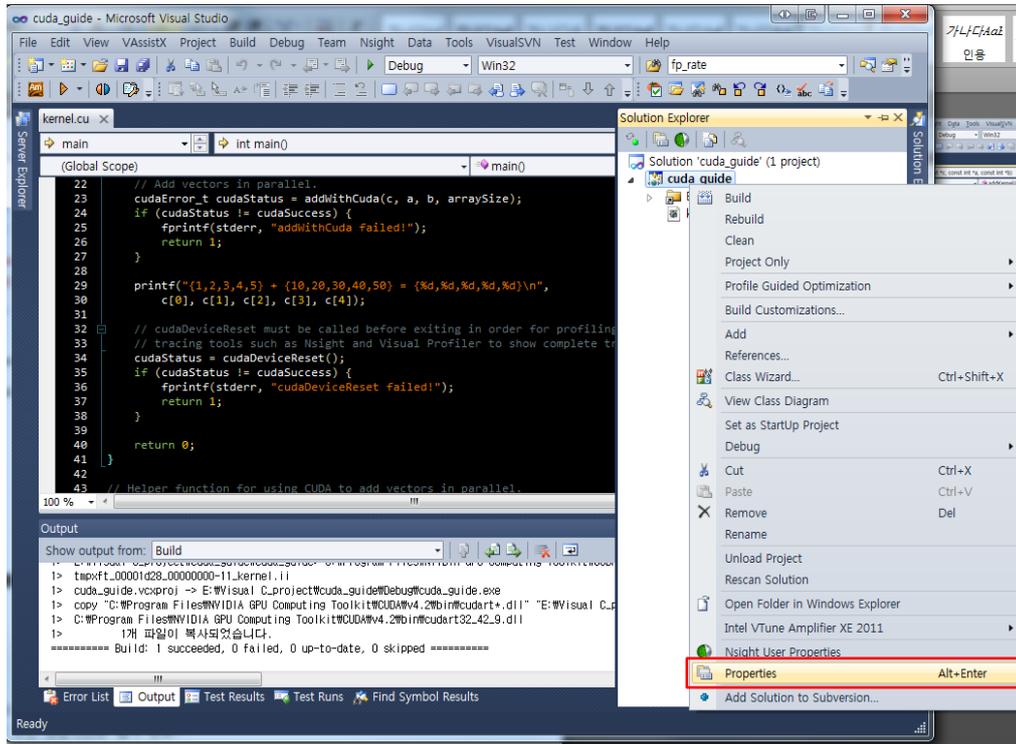


위의 페이지에서 본인 OS에 맞는 현재(2012년 12월) 기준 최신 버전인 CUDA 5.0의 툴킷을 다운로드 받는다. CUDA를 위해 설치해야 할 것은 아래의 총 4가지이다. 다운로드 받아서 설치하면 아래의 1), 2), 3)이 설치 된다.

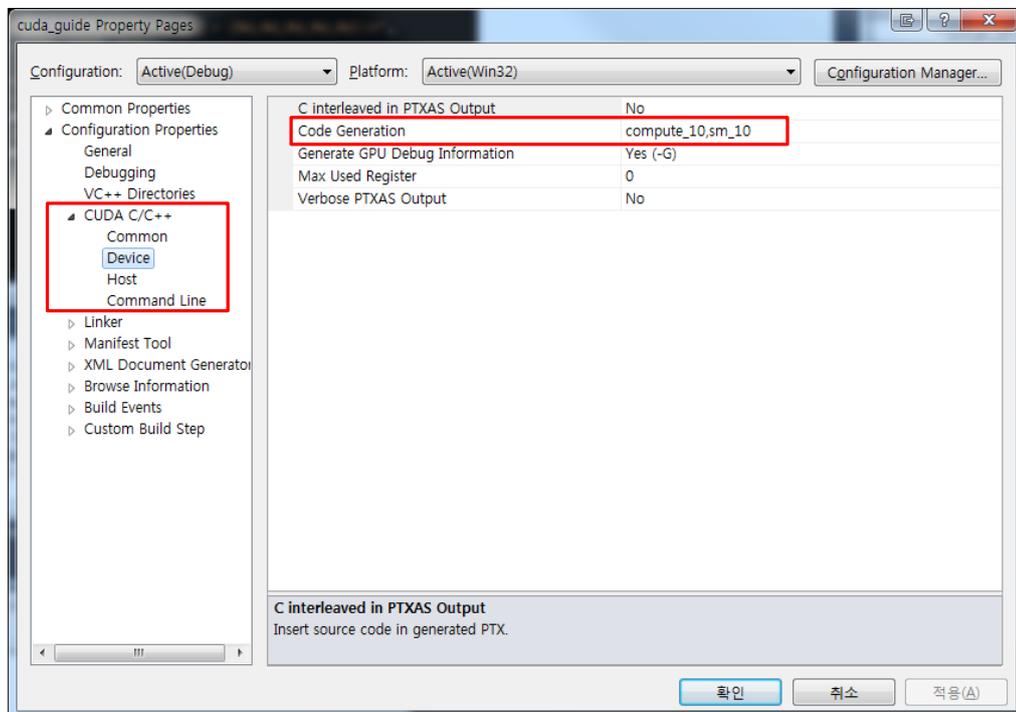
주의사항은 Visual studio 2010이 이미 설치되어 있는 상태에서 설치해야 Add-on 형식으로 Visual studio의 메뉴에 CUDA 메뉴가 추가 된다.

- 1) CUDA Toolkit
- 2) SDK
- 3) Developer drivers(GPU 드라이버)
- 4) 컴파일러(Visual studio 2010 강추 – 여러모로 CUDA와 호환 되는 면이 많이 있음)

④ 본인의 GPU 종류에 따른 컴파일 옵션 설정.



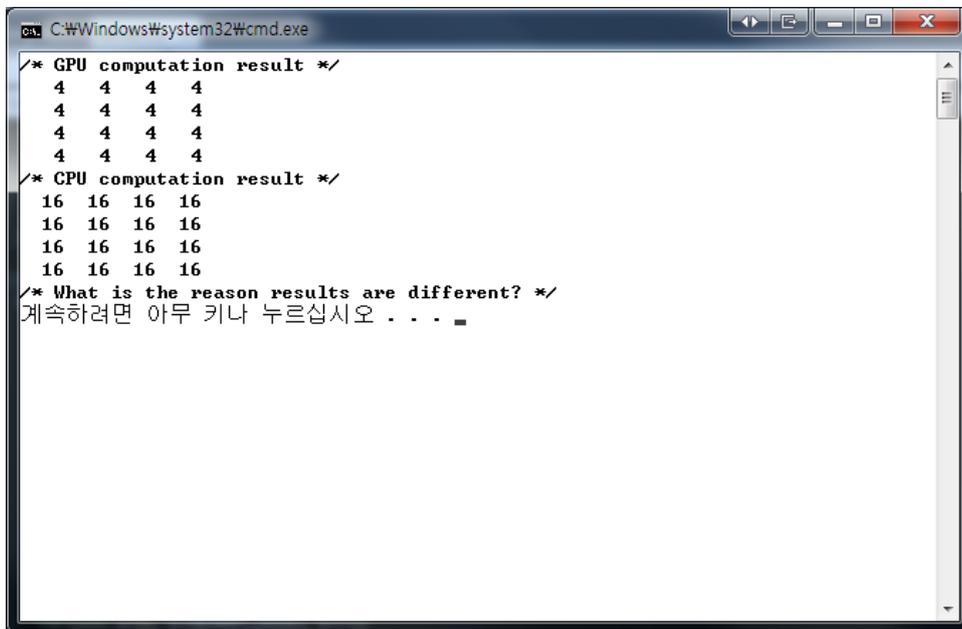
위 그림과 같이 Solution explorer 에서 소스코드가 포함 된 프로젝트 이름 위에서 우클릭 후 'Properties' 클릭.



왼쪽의 Configuration Properties 탭에서 'CUDA C/C++' 메뉴 → 'Device' → 'Code Generation' 선택하여 'compute_10,sm_10'으로 되어 있는 것을 본인 GPU의 Compute capability(<https://developer.nvidia.com/cuda-gpus>)를 찾아 버전을 입력한다.

예를 들어, 본인 GPU의 Compute capability가 3.0이라면, 'compute_30,sm_30'으로 변경하면 해당 GPU의 하드웨어 리소스를 모두 활용하도록 컴파일 할 수 있다. 기본적으로 더 높은 버전은 하위버전을 포함하게 되므로, 가장 낮은 버전인 1.0이 디폴트 옵션으로 잡혀 있는 것이다. 같은 소스코드를 컴파일 해도 Compute capability에 따라 성능 차이는 크게 날 수 있다.

- ⑤ 첨부한 'kernel.cu' 파일을 프로젝트에 추가하고 컴파일하여 실행 해 본다. 아래와 같은 화면이 나오면 문제없이 실행 된 것이다. Matrix 곱 연산을 구현한 CPU 코드와 GPU 커널 코드를 각각 분석하여 왜 결과가 다른지 알아 보는 것이 기본 과제!



```
C:\Windows\system32\cmd.exe
/* GPU computation result */
4 4 4 4
4 4 4 4
4 4 4 4
4 4 4 4
/* CPU computation result */
16 16 16 16
16 16 16 16
16 16 16 16
16 16 16 16
/* What is the reason results are different? */
계속하려면 아무 키나 누르십시오 . . . .
```