Getting Started

NVIDIA CUDA C

Installation and Verification on Microsoft Windows XP, Windows Vista, and Windows 7

Table of Contents

Chapter 1.
Introduction

CUDA—Supercomputing on Desktop Systems

NVIDIA® CUDA™ is a general purpose parallel computing architecture introduced by NVIDIA. It includes the CUDA Instruction Set Architecture (ISA) and the parallel compute engine in the GPU. To program to the CUDA architecture, developers can, today, use C, one of the most widely used high-level programming languages, which can then be run at great performance on a CUDA enabled processor.

The CUDA architecture and its associated software were developed with several design goals in mind:

❑ Provide a small set of extensions to standard programming languages, like C, that enable a straightforward implementation of parallel algorithms. With CUDA and C for CUDA, programmers can focus on the task of parallelization of the algorithms rather than spending time on their implementation.

❑ Support heterogeneous computation where applications use both the CPU and GPU. Serial portions of applications are run on the CPU, and parallel portions are offloaded to the GPU. As such, CUDA can be incrementally applied to existing applications. The CPU and GPU are treated as separate devices that have their own memory spaces. This configuration also allows simultaneous computation on both the CPU and GPU without contention for memory resources.

CUDA-enabled GPUs have hundreds of cores that can collectively run thousands of computing threads. Each core has shared resources, including registers and memory. The on-chip shared memory allows parallel tasks running on these cores to share data without sending it over the system memory bus.

This guide will show you how to install and check the correct operation of CUDA Development Tools.

System Requirements

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

❑ CUDA-enabled GPU

❑ Device driver

❑ CUDA software (available at no cost from http://www.nvidia.com/cuda)

❑ Microsoft Visual Studio 2005 or 2008, or the corresponding versions of Microsoft Visual C++ Express

About This Document

This document is intended for readers familiar with Microsoft Windows XP, Microsoft Windows Vista, or Microsoft Windows 7 operating systems and the Microsoft Visual Studio environment. You do not need previous experience with CUDA or experience with parallel computation.

Chapter 2.
Installing the CUDA Development Tools

The installation of CUDA on a system running Microsoft Windows consists of four simple steps:

❑ Verify the system has a CUDA-enabled GPU

❑ Download the CUDA software

❑ Install the driver for Windows XP, Windows Vista or Windows 7 (if necessary)

❑ Install the CUDA software

Test your installation by compiling and running one of the sample programs in the CUDA software to validate that the hardware and software are running correctly and communicating with each other.

Verify You Have a CUDA-Enabled GPU

Many of the NVIDIA products today contain CUDA-enabled GPUs. These include:

❑ NVIDIA GeForce® 8, 9, and 200 series GPUs

❑ NVIDIA Tesla™ computing solutions

❑ Many of the NVIDIA Quadro® products

An up-to-date list of CUDA-enabled GPUs can be found on the NVIDIA CUDA Web site at http://www.nvidia.com/object/cuda_learn_products.html. The Release Notes for the CUDA Toolkit also contain a list of supported products.

To verify which video adapter your Windows system uses, open the Control Panel (**Start➔Control Panel)** and double click on **System**. In the **System Properties** window that opens, click the **Hardware** tab, then **Device Manager**. Expand the **Display adapters** entry. There you will find the vendor name and model of your graphics card.

Download CUDA Software

The CUDA driver is integrated in the NVIDIA graphics driver and is available from the main CUDA download site at http://www.nvidia.com/object/cuda_get.html.

Choose the platform you are using and download the driver, the CUDA Toolkit, and the CUDA SDK.

Install the CUDA Driver for Microsoft Windows

As mentioned in the previous section, the CUDA driver is integrated in the NVIDIA graphics driver. To use the CUDA Toolkit, you must have at least the version of the NVIDIA ForceWare® graphics driver specified in the CUDA Toolkit release notes. In most cases, if you are running a recent NVIDIA graphics adapter that has support for CUDA, you already have installed the CUDA driver.

To identify the version of your NVIDIA driver, look in the NVIDIA Control Panel. Open the NVIDIA Control Panel by right clicking on the desktop and selecting **NVIDIA Control Panel**. Click the **System Information** button in the lower left corner of the main panel to display a dialog box that specifies the version of the driver installed on your system.

Note: New versions of the CUDA Software can require updates of the driver, so always verify that you are running the right release of the driver for the version of CUDA you are using.

CUDA Software

To run CUDA programs, you will need the following CUDA software:

❑ The CUDA Toolkit

❑ The CUDA SDK

The CUDA Toolkit contains the tools needed to compile and build a CUDA application in conjunction with Microsoft Visual Studio. It includes tools, libraries, header files, and other resources.

The CUDA SDK (software development kit) includes sample projects that have all the necessary project configuration and build files to perform one-click builds using Microsoft Visual Studio.

Both software packages are available for 32-bit Windows (called x86 on the download site) and 64-bit Windows (called x86-64 on the download site). Download instructions appear in an earlier section of this chapter.

Before installing these packages, you should read the Release Notes bundled with each, as these notes provide details on installation and software functionality.

Installing the Software

Follow these few steps for a successful installation:

1. Download the NVIDIA CUDA software from http://www.nvidia.com/object/cuda_get.html and save the installer to your desktop.

2. Uninstall previous versions of the NVIDIA CUDA Toolkit and NVIDIA CUDA SDK if they have previously been installed. You can uninstall the NVIDIA CUDA Toolkit through the **Start** menu: **Start→All Programs→NVIDIA Corporation→CUDA Toolkit→Uninstall CUDA**. Uninstalling the CUDA SDK uses the same sequence.

3. Install the latest version of the NVIDIA CUDA Toolkit by following the installation notes in the NVIDIA CUDA Toolkit release notes. The CUDA Toolkit is installed by default under `C:\CUDA`. Several environment variables are defined with the toolkit installation:

   ➢ `CUDA_BIN_PATH` (defaults to either `C:\CUDA\bin` or `C:\CUDA\bin64`) contains the compiler executables and runtime libraries.

   ➢ `CUDA_INC_PATH` (defaults to `C:\CUDA\include`) contains the include files needed to compile CUDA programs.

   ➢ `CUDA_LIB_PATH` (defaults to either `C:\CUDA\lib` or `C:\CUDA\lib64`) contains the libraries needed for linking CUDA codes.

   In addition to these directories, the CUDA Toolkit installation also includes a documentation directory (`C:\CUDA\doc`) containing the CUDA Programming Guide, compiler guide, and guides for the CUDA implementation of the `BLAS` and `FFT` libraries.

4. Install the latest version of the NVIDIA CUDA SDK by following the installation notes in the NVIDIA CUDA SDK release notes. The CUDA SDK is installed in `C:\Documents and Settings\All Users\Application Data\NVIDIA Corporation\NVIDIA GPU Computing SDK` and contains source code for many example problems and templates for Microsoft Visual Studio.

Verify the Installation

The version of the CUDA Toolkit can be checked by running "nvcc –V" in a Command Prompt window. A Command Prompt window can be obtained through the **Start** menu as follows: **Start➔All Programs➔Accessories➔Command Prompt.**

In the CUDA SDK, NVIDIA includes sample programs that come in both source and compiled form. To verify a correct configuration of the hardware and software, it is highly recommended that you run the **bandwidthTest** program located in **C:\Documents and Settings\All Users\Application Data\NVIDIA Corporation\NVIDIA GPU Computing SDK\C\bin\win32\Release**, presuming that you used the default installation directory structure. (On 64-bit versions of Windows, the directory name ends with **\win64\Release**.)

If CUDA is installed and configured correctly, the output should look similar to Figure 2.


```
C:\Documents and Settings\All Users\Application ...
Running on......
        device 0:Quadro NVS 140M
Quick Mode
Host to Device Bandwidth for Pageable memory
.
Transfer Size (Bytes)    Bandwidth(MB/s)
 33554432                1739.7

Quick Mode
Device to Host Bandwidth for Pageable memory
.
Transfer Size (Bytes)    Bandwidth(MB/s)
 33554432                472.7

Quick Mode
Device to Device Bandwidth
.
Transfer Size (Bytes)    Bandwidth(MB/s)
 33554432                4227.9

&&&& Test PASSED

Press ENTER to exit...
```
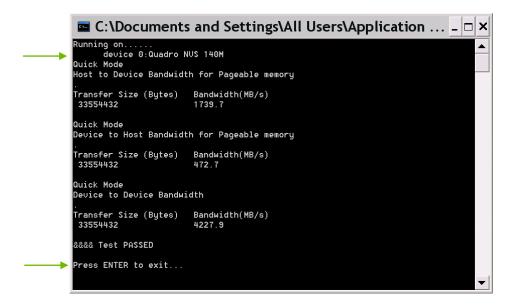
Figure 1.   Valid Results from Sample CUDA **bandwidthTest** Program

Note that your device name (second line) and the bandwidth numbers will vary from system to system. The important items are the second line, which confirms a CUDA device was found, and the second-to-last line, which confirms that all necessary tests passed.

Should the tests not pass, make sure you do have an NVIDIA GPU on your system that supports CUDA and make sure it is properly installed.

To see a graphical representation of what CUDA can do, run the sample Particles executable in **NVIDIA GPU Computing SDK\C\bin\win32\Release** (or …**\win64\Release** on 64-bit Windows).

Chapter 3.
Compiling a CUDA Program

The project files in the CUDA SDK have been designed to provide simple, one-click builds of the programs, which include all source code. To build the 32-bit or 64-bit Windows projects (for release, debug, or emulated release and debug—called **emurelease** and **emudebug**, respectively), use the provided ***.sln** solution files for Microsoft Visual Studio 2005 and ***_vc90.sln** for Microsoft Visual Studio 2008. (Likewise for the corresponding versions of Microsoft Visual C++ Express Edition.) You can use either the solution files located in each of the examples directories in **NVIDIA GPU Computing SDK\C\src** or the global solution files **Release.sln** or **Release_vc90.sln** located in **NVIDIA GPU Computing SDK\C\src**.

Compiling Sample Projects

The **bandwidthTest** project is a good sample project to build and run. It is located in the **C:\Documents and Settings\All Users\Application Data\NVIDIA Corporation\NVIDIA GPU Computing SDK\C\src\bandwidthTest** directory. The output is placed in **NVIDIA GPU Computing SDK\C\bin\win32\Debug**. (As mentioned previously, the **\win32** segment of this address will be **\win64** on 64-bit versions of Windows.) This location presumes that you used the default installation directory structure.

Build the program using the appropriate solution file and run the executable. If all works correctly, the output should be similar to Figure 1.

Sample Projects

The sample projects come in four configurations: debug and release (where release contains no debugging information), and emulated versions of both. The emulated versions are for developing and running CUDA software in the absence of a CUDA GPU.

A few of the example projects require some additional setup. The **simpleD3D9** example requires the system to have a Direct3D SDK installed and the Visual C++ directory paths (located in **Tools➔Options**...) properly configured. Consult the Direct3D documentation for additional details.

Most samples link to a utility library called **cutil** whose source code is in **NVIDIA GPU Computing SDK\C\common\src**. The **release** and **emurelease** versions of these samples link to **cutil32.lib** (or **cutil64.lib**) and dynamically load **cutil32.dll** (or **cutil64.dll**). The **debug** and **emudebug** versions of these

samples link to **cutil32D.lib** and dynamically load **cutil32D** (or their 64-bit equivalents on 64-bit versions of Windows).

To build the Win32 release and/or debug configurations of the **cutil** library, use the solution files located in **NVIDIA GPU Computing SDK\C\common**. The output of the compilation process should be placed in **NVIDIA GPU Computing SDK\C\common\lib**:

- ❑ **cutil32.lib and cutil32D.lib** (or **cutil64.lib** and **cutil64D.lib**) are the release and debug import libraries.

- ❑ **cutil32.dll** and **cutil32D.dll** (or **cutil64.dll** and **cutil64D.dll**) are the release and debug dynamic-link libraries, which also are copied to **NVIDIA GPU Computing SDK\C\bin\win32\[release|emurelease]** and **NVIDIA GPU Computing SDK\C\bin\win32\[debug|emudebug]** respectively. (Substitute **\win64 for \win32 on 64-bit Windows**.)

What's Next?

Now that you have CUDA-enabled hardware and the software installed, you can examine and enjoy the numerous included programs. To begin using CUDA to accelerate the performance of your own applications, consult the *CUDA Programming Guide*, located in the directory where you installed the CUDA Toolkit (by default, **C:\CUDA\doc**).

For tech support on programming questions, consult and participate in the bulletin board and mailing list at http://forums.nvidia.com/index.php?showforum=71.