NVIDIA CUDA GETTING STARTED GUIDE FOR MICROSOFT WINDOWS

DU-05349-001_v5.0 | October 2012

Installation and Verification on Windows

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1. INTRODUCTION

CUDA™ is a parallel computing platform and programming model invented by NVIDIA. It enables dramatic increases in computing performance by harnessing the power of the graphics processing unit (GPU).

CUDA was 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 C/C++, 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 the CPU and GPU without contention for memory resources.

CUDA-capable GPUs have hundreds of cores that can collectively run thousands of computing threads. These cores have shared resources including a register file and a shared 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 the CUDA development tools.

1.1 System Requirements

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

- CUDA-capable GPU
- Microsoft Windows XP, Vista, 7, or 8 or Windows Server 2003 or 2008
- NVIDIA CUDA Toolkit (available at no cost from http://www.nvidia.com/content/cuda/cuda-downloads.html)
- Microsoft Visual Studio 2008 or 2010, or a corresponding version of Microsoft Visual C++ Express

1.2 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 CUDA DEVELOPMENT TOOLS

The installation of CUDA development tools on a system running the appropriate version of Windows consists of a few simple steps:

- Verify the system has a CUDA-capable GPU.
- Download the NVIDIA CUDA Toolkit.
- Install the NVIDIA CUDA Toolkit.
- Test that the installed software runs correctly and communicated with the hardware.

2.1 Verify You Have a CUDA-Capable GPU

To verify that your GPU is CUDA-capable, 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. If it is an NVIDIA card that is listed in http://www.nvidia.com/object/cuda_gpus.html, your GPU is CUDA-capable.

The Release Notes for the CUDA Toolkit also contain a list of supported products.

2.2 Download the NVIDIA CUDA Toolkit

The NVIDIA CUDA Toolkit is available at http://www.nvidia.com/content/cuda/cuda-downloads.html.

Choose the platform you are using and download the NVIDIA CUDA Toolkit

The NVIDIA CUDA Toolkit contains the driver and tools needed to create, build and run a CUDA application as well as libraries, header files, CUDA samples source code, and other resources.

2.3 Install the CUDA Software

Before installing the toolkit, you should read the *Release Notes*, as they provide details on installation and software functionality.

Install the CUDA Toolkit by executing the Toolkit installer and following the on-screen prompts.

The driver and toolkit must be installed for CUDA to function. If you have not installed a stand-alone driver, install the driver from the NVIDIA CUDA Toolkit.

You can choose what to install from the following packages:

1. 

   If you want to install the CUDA Driver for new hardware, and have already installed the CUDA Driver before, you can launch the CUDA Driver installer from the Start Menu under:

   `NVIDIA Corporation\CUDA Toolkit\v5.0`, or

   `NVIDIA Corporation\CUDA Toolkit\v5.0 (64 bit)`

   CUDA Driver

   The CUDA Driver installation can be done silently or by using a GUI. A silent installation of the driver is done by enabling that feature when choosing what to install.

   - Silent: Only the display driver will be installed.
   - GUI: A window will appear after the CUDA Toolkit installation if you allowed it at the last dialog with the full driver installation UI. You can choose which features you wish to install.

2. CUDA Toolkit

   The CUDA Toolkit installation defaults to `C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v#.#`, where `#.#` is version number 3.2 or higher. This directory contains the following:

   `Bin\`
   the compiler executables and runtime libraries

   `Include\`
   the header files needed to compile CUDA programs

   `Lib\`
   the library files needed to link CUDA programs

   `Doc\`
   the *CUDA C Programming Guide*, *CUDA C Best Practices Guide*, documentation for the CUDA libraries, and other CUDA Toolkit-related documentation

   Note: CUDA Toolkit versions 3.1 and earlier installed into `C:\CUDA` by default, requiring prior CUDA Toolkit versions to be uninstalled before the installation of new

versions. Beginning with CUDA Toolkit 3.2, multiple CUDA Toolkit versions can be installed simultaneously.

3. CUDA Samples

   The CUDA Samples contain source code for many example problems and templates with Microsoft Visual Studio 2008 and 2010 projects.

   For Windows XP, the samples can be found here:

   ```
   C:\Documents and Settings\All Users\Application Data\NVIDIA Corporation\CUDA Samples\v5.0
   ```

   For Windows Vista, Windows 7, and Windows Server 2008, the samples can be found here:

   ```
   C:\ProgramData\NVIDIA Corporation\CUDA Samples\v5.0
   ```

The NVIDIA CUDA Toolkit installer only installs Visual Studio project templates for toolkit version 5.0 and higher. Installing NVIDIA® Nsight™, Visual Studio Edition will install Visual Studio project templates for toolkit versions earlier than CUDA 5.0.

2.4 Verify the Installation

Before continuing, it is important to verify that the CUDA programs can find and communicate correctly with the CUDA-capable hardware. To do this, you need to compile and run some of the included sample programs.

2.4.1 Running the Compiled Examples

The version of the CUDA Toolkit can be checked by running `nvcc -V` in a **Command Prompt** window. You can display a **Command Prompt** window by going to:

Start > All Programs > Accessories > Command Prompt

CUDA Samples include sample programs in both *source* and *compiled* form. To verify a correct configuration of the hardware and software, it is highly recommended that you run the `deviceQuery` program located here:

Windows XP:

```
C:\Documents and Settings\All Users\Application Data\NVIDIA Corporation\CUDA Samples\v5.0\C\bin\win32\Release
```

Windows Vista, Windows 7, Windows 8, Windows Server 2003, and Windows Server 2008:

```
C:\ProgramData\NVIDIA Corporation\CUDA Samples\v5.0\C\bin\win32\Release
```

This assumes 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 1 Valid Results from Sample CUDA Device Query Program.

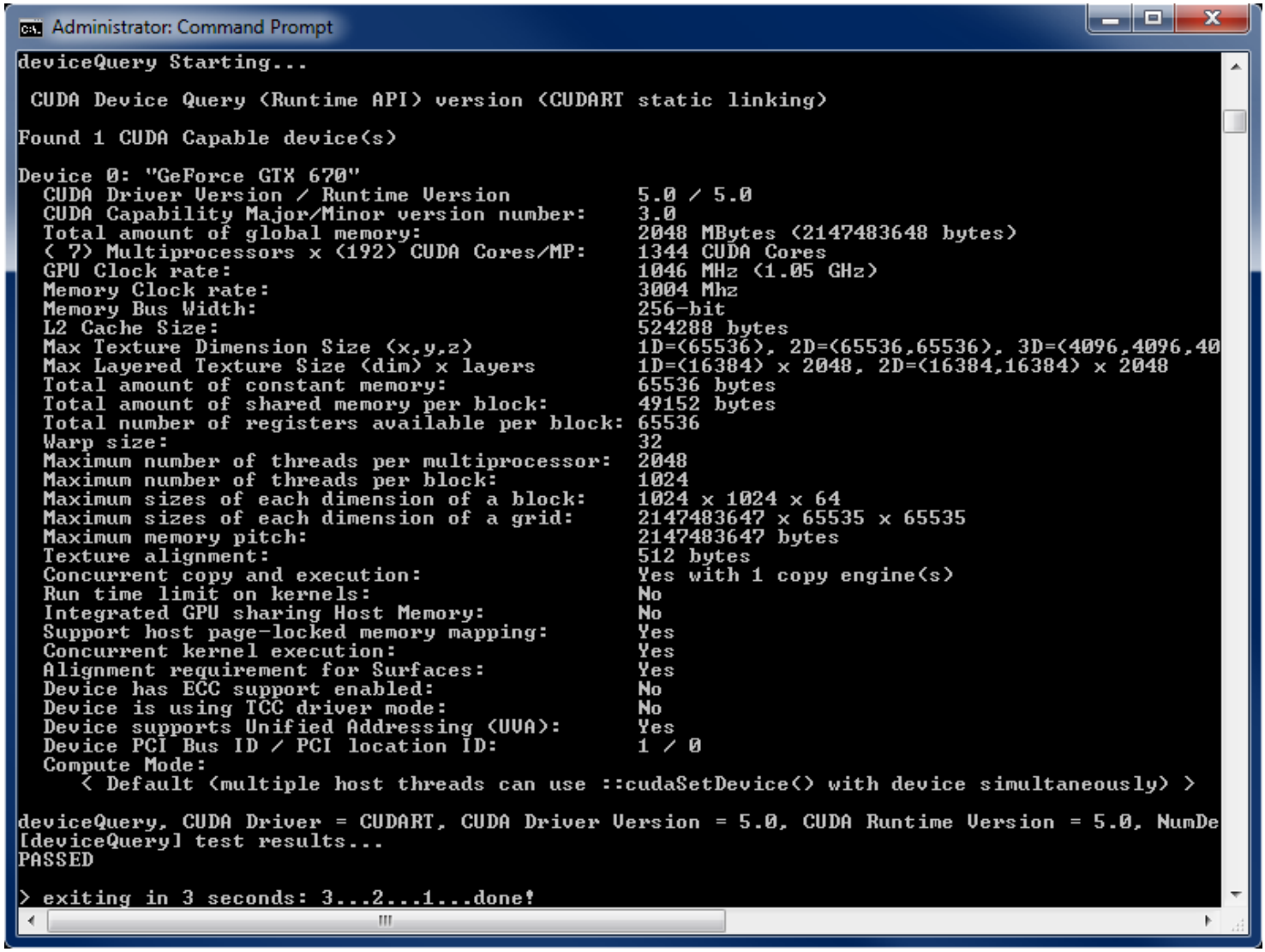

Figure 1 Valid Results from Sample CUDA Device Query Program

The exact appearance and the output lines might be different on your system. The important outcomes are that a device was found, that the device(s) match what is installed in your system, and that the test passed.

If a CUDA-capable device and the CUDA Driver are installed but `deviceQuery` reports that no CUDA-capable devices are present, ensure the deivce and driver are properly installed.

Running the `bandwidthTest` program, located in the same directory as `deviceQuery` above, ensures that the system and the CUDA-capable device are able to communicate correctly. The output should resemble Figure 2 Valid Results from Sample CUDA Bandwidth Test Program.

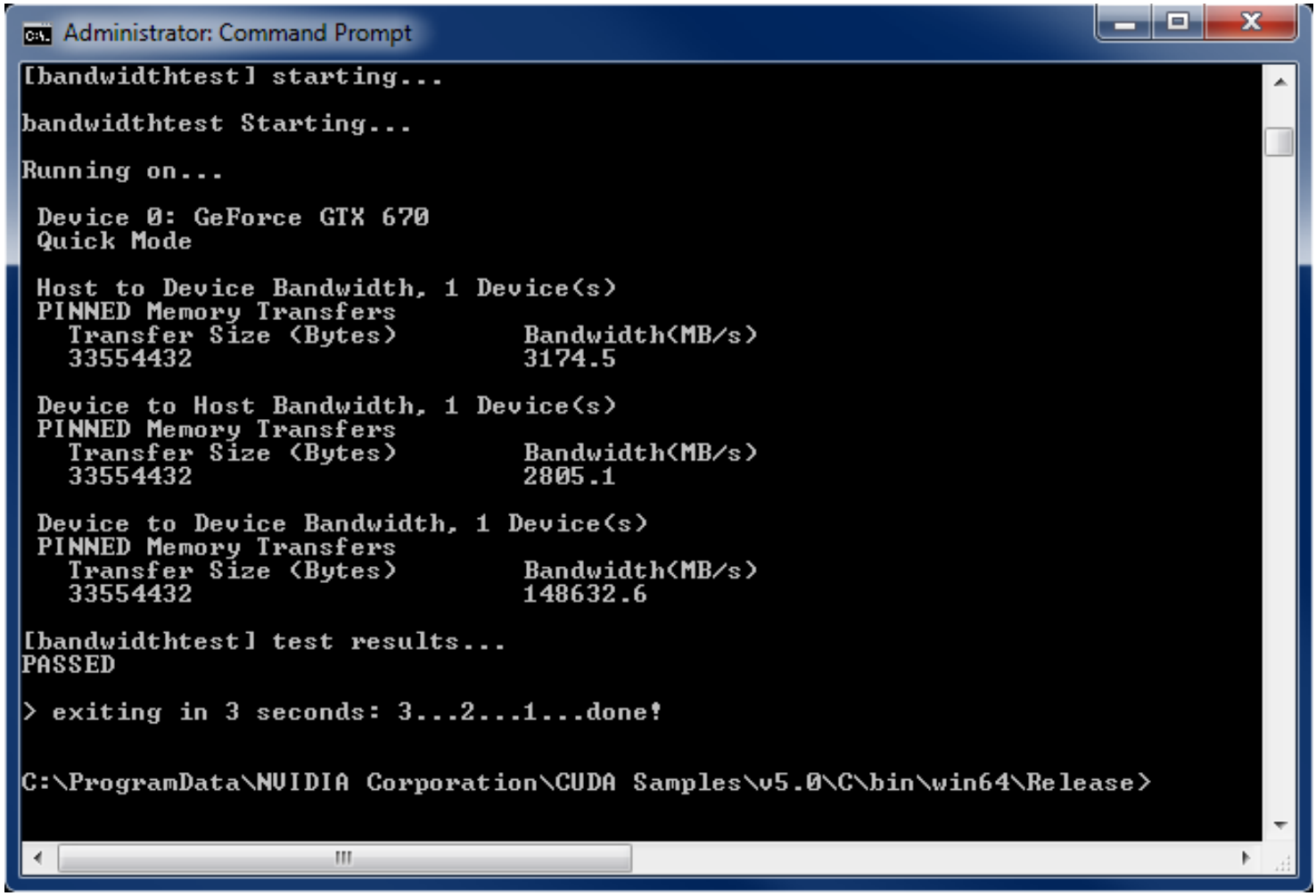

Figure 2 Valid Results from Sample CUDA Bandwidth Test Program

The device name (second line) and the bandwidth numbers 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.

If the tests do not pass, make sure you do have a CUDA-capable NVIDIA GPU on your system and make sure it is properly installed.

To see a graphical representation of what CUDA can do, run the sample Particles executable in:

- For Windows XP:

  ```
  c:\Documents and Settings\All Users\Application Data\CUDA Samples\v5.0\C\bin
  \win32\Release
  ```

  (or …`\win64\Release` on 64-bit Windows)
- For Windows Vista, Windows 7, Windows 8, Windows Server 2003, and Windows Server 2008:

  ```
  C:\ProgramData\NVIDIA Corporation\CUDA Samples\v5.0\C\bin\win32\Release
  ```

  (or …`\win64\Release` on 64-bit Windows)

Chapter 3. COMPILING CUDA PROGRAMS

The project files in the CUDA Samples have been designed to provide simple, one-click builds of the programs that include all source code. To build the 32-bit or 64-bit Windows projects (for release or debug mode), use the provided `*.sln` solution files for Microsoft Visual Studio 2008 or 2010 (and 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

```
CUDA Samples\v5.0\C\<category>\<sample_name>
```

or the global solution files `Samples*.sln` located in

```
CUDA Samples\v5.0\C
```

CUDA Samples are organized according to `<category>`. Each sample is organized into one of the following folders: (`0_Simple`, `1_Utilities`, `2_Graphics`, `3_Imaging`, `4_Finance`, `5_Simulations`, `6_Advanced`, `7_CUDALibraries`).

3.1 Compiling Sample Projects

The `bandwidthTest` project is a good sample project to build and run. It is located in the `NVIDIA Corporation\CUDA Samples\v5.0\C\1_Utilities \bandwidthTest` directory.

The output is placed in `CUDA Samples\C\v5.0\bin\win32\Release`. (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 2 Valid Results from Sample CUDA Bandwidth Test Program.

3.2 Sample Projects

The sample projects come in two configurations: debug and release (where release contains no debugging information).

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.

These sample projects also make use of the `$CUDA_PATH` environment variable to locate the CUDA Toolkit and a `.rules` file to locate and configure the `nvcc` compiler. The environment variable is set automatically and the `.rules` file is installed automatically as part of the CUDA Toolkit installation process. The `.rules` file is installed into `$VisualStudioInstallDir\VC\VCProjectDefaults`. You can reference this `.rules` file from your Visual Studio project files when building your own CUDA applications.

3.3 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 5.0 Runtime" template will configure your project for use with the CUDA 5.0 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.

To specify a custom CUDA Toolkit location, under **CUDA C/C++**, select **Common**, and set the **CUDA Toolkit Custom Dir** field as desired. Note that the selected toolkit must match the version of the Build Customizations.

3.4 Build Customizations for Existing Projects

When adding CUDA acceleration to existing applications, the relevant Visual Studio project file must be updated to include CUDA build customizations. For Visual Studio 2010, this can be done using one of the following two methods:

1. Open the Visual Studio 2010 project, right click on the project name, and select **Build Customizations...**, then select the CUDA Toolkit version you would like to target.
2. Alternatively, you can configure your project always to build with the most recently installed version of the CUDA Toolkit. First add a CUDA build customization to your project as above. Then, right click on the project name and select **Properties**. Under **CUDA C/C++**, select **Common**, and set the **CUDA Toolkit Custom Dir** field to `$(CUDA_PATH)`.

While Option 2 will allow your project to automatically use any new CUDA Toolkit version you may install in the future, selecting the toolkit version explicitly as in Option 1 is often better in practice, because if there are new CUDA configuration options added to the build customization rules accompanying the newer toolkit, you would not see those new options using Option 2.

Note for advanced users: If you wish to try building your project against a newer CUDA Toolkit without making changes to any of your project files, go to the Visual Studio

2010 command prompt, change the current directory to the location of your project, and execute a command such as the following:

```
msbuild <projectname.extension> /t:Rebuild /p:CudaToolkitDir="drive:/path/to/
new/toolkit/"
```

Chapter 4. ADDITIONAL CONSIDERATIONS

Now that you have CUDA-capable 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 C Programming Guide*, located in the CUDA Toolkit documentation directory.

A number of helpful development tools are included in the CUDA Toolkit or are available for download from the NVIDIA Developer Zone to assist you as you develop your CUDA programs, such as NVIDIA® Nsight™ Visual Studio Edition, NVIDIA Visual Profiler, and cuda-memcheck.

For technical support on programming questions, consult and participate in the developer forums at http://developer.nvidia.com/cuda/.