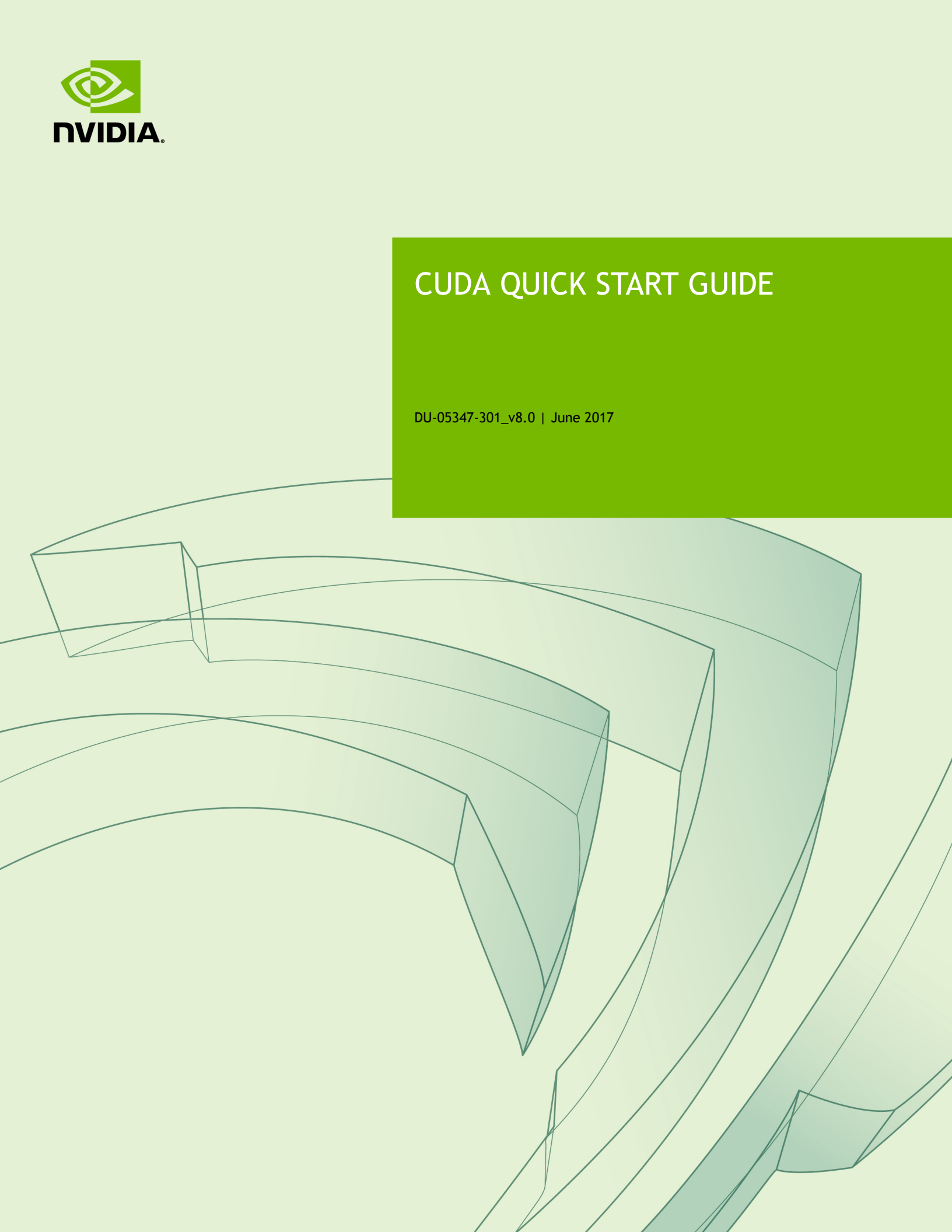


CUDA QUICK START GUIDE

DU-05347-301_v8.0 | June 2017

TABLE OF CONTENTS

Chapter 1. Introduction 1
Chapter 2. Windows 2
2.1. Network Installer 2
2.2. Local Installer 4
Chapter 3. Mac OSX 6
3.1. Network Installer 6
3.2. Local Installer 7
Chapter 4. Linux 8
4.1. Linux x86_64 8
4.1.1. Redhat / CentOS 8
4.1.1.1. RPM Installer 8
4.1.1.2. Runfile Installer 9
4.1.2. Fedora 9
4.1.2.1. RPM Installer 10
4.1.2.2. Runfile Installer 10
4.1.3. SUSE Linux Enterprise Server 11
4.1.3.1. RPM Installer 11
4.1.3.2. Runfile Installer 11
4.1.4. OpenSUSE 12
4.1.4.1. RPM Installer 12
4.1.4.2. Runfile Installer 13
4.1.5. Ubuntu 13
4.1.5.1. Debian Installer 14
4.1.5.2. Runfile Installer 14
4.2. Linux armhf 15
4.2.1. L4T 15
4.2.1.1. Debian Installer 15
4.3. Linux aarch64 15
4.3.1. L4T 15
4.3.1.1. Debian Installer 16
4.4. Linux POWER8 16
4.4.1. Ubuntu 16
4.4.1.1. Debian Installer 16
4.4.2. Redhat / CentOS 17
4.4.2.1. RPM Installer 17

Chapter 1. INTRODUCTION

This guide covers the basic instructions needed to install CUDA and verify that a CUDA application can run on each supported platform.

These instructions are intended to be used on a clean installation of a supported platform. For questions which are not answered in this document, please refer to the Windows Installation Guide, Mac Installation Guide, and Linux Installation Guide.

The CUDA installation packages can be found on the CUDA Downloads Page.

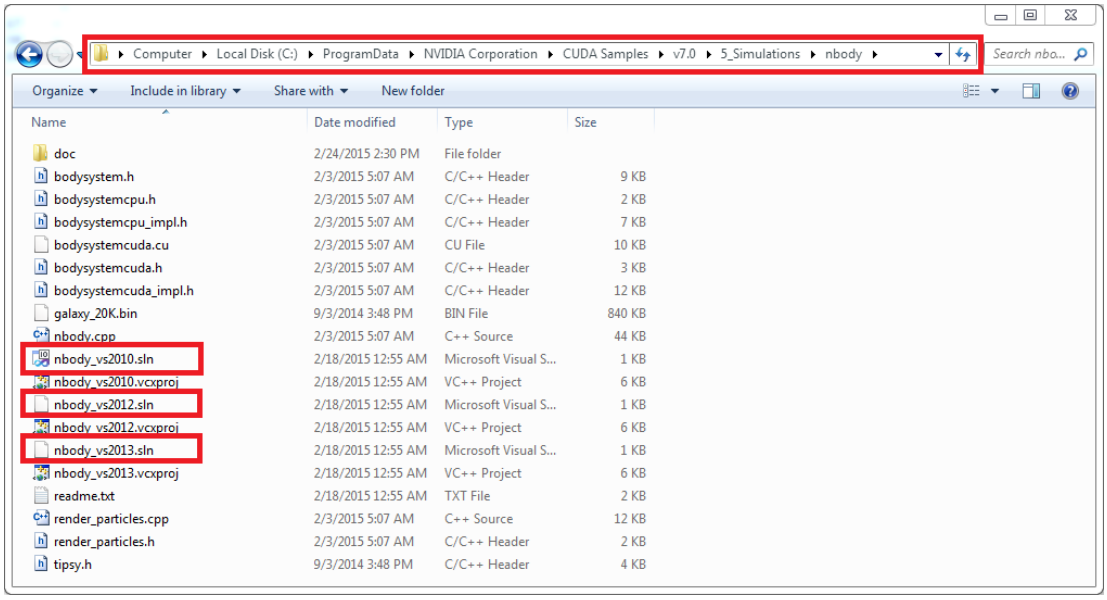
Chapter 2. WINDOWS

When installing CUDA on Windows, you can choose between the Network Installer and the Local Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. For more details, refer to the Windows Installation Guide.

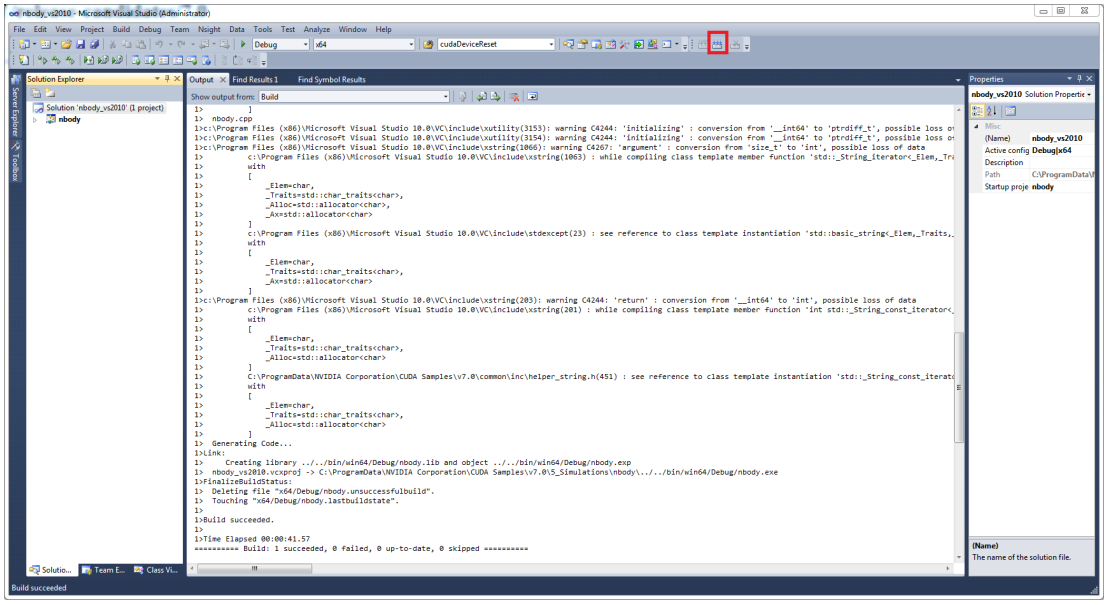
2.1. Network Installer

Perform the following steps to install CUDA and verify the installation.

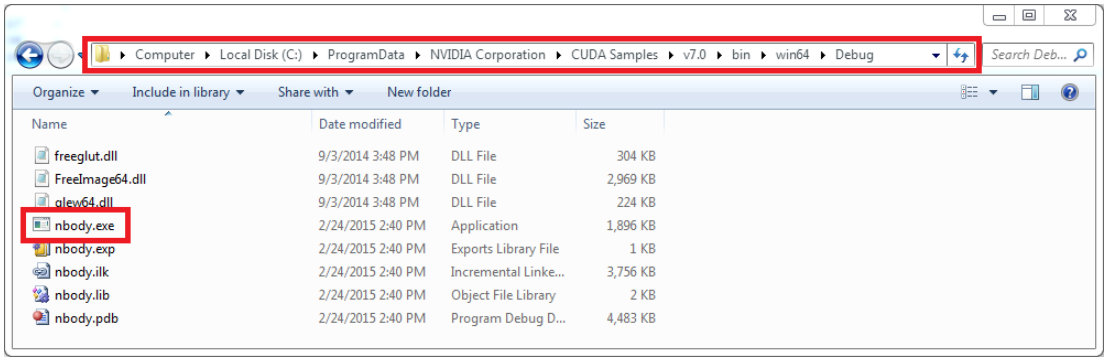
1. Launch the downloaded installer package.
2. Read and accept the EULA.
3. Select "next" to download and install all components.
4. Once the download completes, the installation will begin automatically.
5. Once the installation completes, click "next" to acknowledge the Nsight Visual Studio Edition installation summary.
6. Click "close" to close the installer.
7. Navigate to the CUDA Samples' nbody directory.
8. Open the nbody Visual Studio solution file for the version of Visual Studio you have installed.



9. Open the "Build" menu within Visual Studio and click "Build Solution".



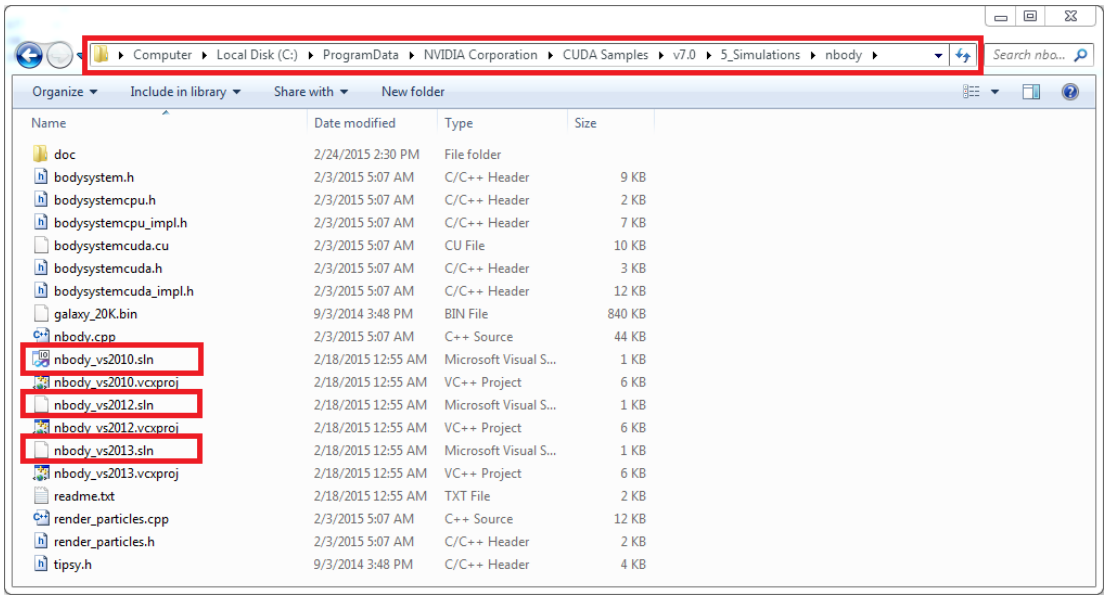
10. Navigate to the CUDA Samples' build directory and run the nbody sample.



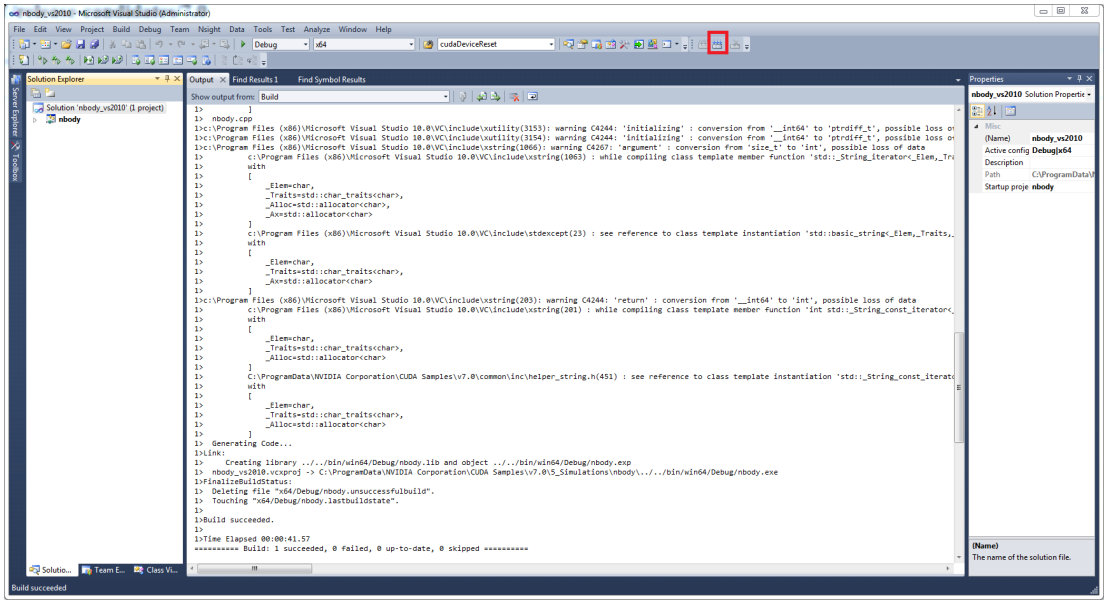
2.2. Local Installer

Perform the following steps to install CUDA and verify the installation.

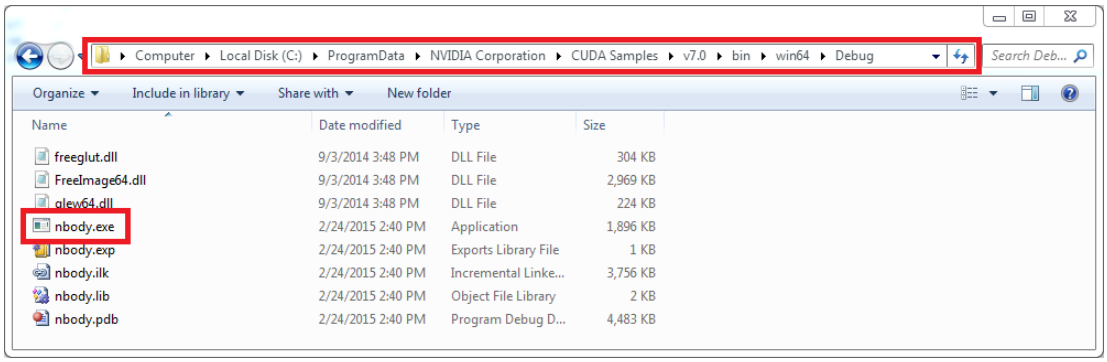
1. Launch the downloaded installer package.
2. Read and accept the EULA.
3. Select "next" to install all components.
4. Once the installation completes, click "next" to acknowledge the Nsight Visual Studio Edition installation summary.
5. Click "close" to close the installer.
6. Navigate to the CUDA Samples' nbody directory.
7. Open the nbody Visual Studio solution file for the version of Visual Studio you have installed.



8. Open the "Build" menu within Visual Studio and click "Build Solution".



9. Navigate to the CUDA Samples' build directory and run the nbody sample.



Chapter 3. MAC OSX

When installing CUDA on Mac OSX, you can choose between the Network Installer and the Local Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. For more details, refer to the Mac Installation Guide.

3.1. Network Installer

Perform the following steps to install CUDA and verify the installation.

1. Launch the installer.
2. Read and accept the EULA.
3. Select "next" to download and install all components.
4. Once the downloads and installations complete, click "next" to move to the install finished screen.
5. Click "close" to close the installer.
6. Open a terminal.
7. Set up the development environment by modifying the PATH and DYLD_LIBRARY_PATH variables:

```
$ export PATH=/Developer/NVIDIA/CUDA-8.0/bin${PATH:+:${PATH}}
$ export DYLD_LIBRARY_PATH=/Developer/NVIDIA/CUDA-8.0/lib\
                           ${DYLD_LIBRARY_PATH:+:${DYLD_LIBRARY_PATH}}
```

8. Install Xcode via the App Store.
9. Install Xcode command-line tools:

```
$ xcode-select --install
```

10. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

3.2. Local Installer

Perform the following steps to install CUDA and verify the installation.

1. Launch the installer.
2. Read and accept the EULA.
3. Select "next" to install all components.
4. Once the installations complete, click "next" to move to the install finished screen.
5. Click "close" to close the installer.
6. Open a terminal.
7. Set up the development environment by modifying the PATH and DYLD_LIBRARY_PATH variables:

```
$ export PATH=/Developer/NVIDIA/CUDA-8.0/bin${PATH:+:${PATH}}
$ export DYLD_LIBRARY_PATH=/Developer/NVIDIA/CUDA-8.0/lib\
                           ${DYLD_LIBRARY_PATH:+:${DYLD_LIBRARY_PATH}}
```

8. Install Xcode via the App Store.
9. Install Xcode command-line tools:

```
$ xcode-select --install
```

10. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

Chapter 4. LINUX

CUDA on Linux can be installed using an RPM, Debian, or Runfile package, depending on the platform being installed on.

4.1. Linux x86_64

For development on the x86_64 architecture. In some cases, x86_64 systems may act as host platforms targeting other architectures. See the Linux Installation Guide for more details.

4.1.1. Redhat / CentOS

When installing CUDA on Redhat or CentOS, you can choose between the Runfile Installer and the RPM Installer. The Runfile Installer is only available as a Local Installer. The RPM Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. In the case of the RPM installers, the instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

4.1.1.1. RPM Installer

Perform the following steps to install CUDA and verify the installation.

1. Install EPEL to satisfy the DKMS dependency by following the instructions at EPEL's website.
2. Install the repository meta-data, clean the yum cache, and install CUDA:

```
$ sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
$ sudo yum clean expire-cache
$ sudo yum install cuda
```

3. Reboot the system to load the NVIDIA drivers.
4. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

5. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

4.1.1.2. Runfile Installer

Perform the following steps to install CUDA and verify the installation.

1. Disable the Nouveau drivers:
    1. Create a file at **/etc/modprobe.d/blacklist-nouveau.conf** with the following contents:

    ```
    blacklist nouveau
    options nouveau modeset=0
    ```

    2. Regenerate the kernel initramfs:

    ```
    $ sudo dracut --force
    ```

2. Reboot into runlevel 3 by temporarily adding the number "3" and the word "nomodeset" to the end of the system's kernel boot parameters.
3. Run the installer silently to install with the default selections (implies acceptance of the EULA):

```
sudo sh cuda_<version>_linux.run --silent
```

4. Create an xorg.conf file to use the NVIDIA GPU for display:

```
$ sudo nvidia-xconfig
```

5. Reboot the system to load the graphical interface.
6. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

7. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

4.1.2. Fedora

When installing CUDA on Fedora, you can choose between the Runfile Installer and the RPM Installer. The Runfile Installer is only available as a Local Installer. The RPM Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. In the case of the RPM installers, the

instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

4.1.2.1. RPM Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the RPMFusion free repository to satisfy the Akmods dependency:

```
$ su -c 'dnf install --nogpgcheck http://download1.rpmfusion.org/free/
fedora/rpmfusion-free-release-$(rpm -E %fedora).noarch.rpm'
```

2. Install the repository meta-data, clean the dnf cache, and install CUDA:

```
$ sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
$ sudo dnf clean expire-cache
$ sudo dnf install cuda
```

3. Reboot the system to load the NVIDIA drivers.
4. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

5. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

4.1.2.2. Runfile Installer

Perform the following steps to install CUDA and verify the installation.

1. Disable the Nouveau drivers:
    1. Create a file at **/usr/lib/modprobe.d/blacklist-nouveau.conf** with the following contents:

```
blacklist nouveau
options nouveau modeset=0
```

    2. Regenerate the kernel initramfs:

```
$ sudo dracut --force
```

2. Reboot into runlevel 3 by temporarily adding the number "3" and the word "nomodeset" to the end of the system's kernel boot parameters.
3. Run the installer silently to install with the default selections (implies acceptance of the EULA):

```
sudo sh cuda_<version>_linux.run --silent
```

4. Create an xorg.conf file to use the NVIDIA GPU for display:

```
$ sudo nvidia-xconfig
```

5. Reboot the system to load the graphical interface.

6. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

7. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

4.1.3. SUSE Linux Enterprise Server

When installing CUDA on SUSE Linux Enterprise Server, you can choose between the Runfile Installer and the RPM Installer. The Runfile Installer is only available as a Local Installer. The RPM Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. In the case of the RPM installers, the instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

4.1.3.1. RPM Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the repository meta-data, refresh the Zypper cache, and install CUDA:

```
$ sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
$ sudo zypper refresh
$ sudo zypper install cuda
```

2. Add the user to the video group:

```
$ sudo usermod -a -G video <username>
```

3. Reboot the system to load the NVIDIA drivers.
4. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

5. Install a writable copy of the samples then build and run the vectorAdd sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/0_Simple/vectorAdd
$ make
$ ./vectorAdd
```

4.1.3.2. Runfile Installer

Perform the following steps to install CUDA and verify the installation.

1. Reboot into runlevel 3 by temporarily adding the number "3" and the word "nomodeset" to the end of the system's kernel boot parameters.

2. Run the installer silently to install with the default selections (implies acceptance of the EULA):

```
sudo sh cuda_<version>_linux.run --silent
```

3. Create an xorg.conf file to use the NVIDIA GPU for display:

```
$ sudo nvidia-xconfig
```

4. Reboot the system to load the graphical interface.
5. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

6. Install a writable copy of the samples then build and run the vectorAdd sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/0_Simple/vectorAdd
$ make
$ ./vectorAdd
```

4.1.4. OpenSUSE

When installing CUDA on OpenSUSE, you can choose between the Runfile Installer and the RPM Installer. The Runfile Installer is only available as a Local Installer. The RPM Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. In the case of the RPM installers, the instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

4.1.4.1. RPM Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the repository meta-data, refresh the Zypper cache, and install CUDA:

```
$ sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
$ sudo zypper refresh
$ sudo zypper install cuda
```

2. Add the user to the video group:

```
$ sudo usermod -a -G video <username>
```

3. Reboot the system to load the NVIDIA drivers.
4. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

5. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

4.1.4.2. Runfile Installer

Perform the following steps to install CUDA and verify the installation.

1. Disable the Nouveau drivers:
    1. Create a file at **/etc/modprobe.d/blacklist-nouveau.conf** with the following contents:

        ```
        blacklist nouveau
        options nouveau modeset=0
        ```
    2. Regenerate the kernel initrd:

        ```
        $ sudo /sbin/mkinitrd
        ```
2. Reboot into runlevel 3 by temporarily adding the number "3" and the word "nomodeset" to the end of the system's kernel boot parameters.
3. Run the installer silently to install with the default selections (implies acceptance of the EULA):

    ```
    sudo sh cuda_<version>_linux.run --silent
    ```
4. Create an xorg.conf file to use the NVIDIA GPU for display:

    ```
    $ sudo nvidia-xconfig
    ```
5. Reboot the system to load the graphical interface.
6. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

    ```
    $ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
    $ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
    ```
7. Install a writable copy of the samples then build and run the nbody sample:

    ```
    $ cuda-install-samples-8.0.sh ~
    $ cd ~/NVIDIA_CUDA-8.0_Samples/5_Simulations/nbody
    $ make
    $ ./nbody
    ```

4.1.5. Ubuntu

When installing CUDA on Ubuntu, you can choose between the Runfile Installer and the Debian Installer. The Runfile Installer is only available as a Local Installer. The Debian Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. In the case of the Debian installers, the instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

4.1.5.1. Debian Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the repository meta-data, update the apt-get cache, and install CUDA:

```
$ sudo dpkg --install cuda-repo-<distro>-<version>.<architecture>.deb
$ sudo apt-get update
$ sudo apt-get install cuda
```

2. Reboot the system to load the NVIDIA drivers.
3. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

4. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

4.1.5.2. Runfile Installer

Perform the following steps to install CUDA and verify the installation.

1. Disable the Nouveau drivers:
    1. Create a file at **/etc/modprobe.d/blacklist-nouveau.conf** with the following contents:

```
blacklist nouveau
options nouveau modeset=0
```

    2. Regenerate the kernel initramfs:

```
$ sudo update-initramfs -u
```

2. Reboot into runlevel 3 by temporarily adding the number "3" and the word "nomodeset" to the end of the system's kernel boot parameters.
3. Run the installer silently to install with the default selections (implies acceptance of the EULA):

```
sudo sh cuda_<version>_linux.run --silent
```

4. Create an xorg.conf file to use the NVIDIA GPU for display:

```
$ sudo nvidia-xconfig
```

5. Reboot the system to load the graphical interface.
6. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

7. Install a writable copy of the samples then build and run the nbody sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

4.2. Linux armhf

For development on the armhf architecture.

4.2.1. L4T

When installing CUDA on L4T on ARMv7, you must use the Debian Installer. The Debian Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. The instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

4.2.1.1. Debian Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the repository meta-data, update the apt-get cache, and install CUDA:

```
$ sudo dpkg --install cuda-repo-<distro>-<version>.<architecture>.deb
$ sudo apt-get update
$ sudo apt-get install cuda-toolkit-8-0
```

2. Reboot the system to load the NVIDIA drivers.
3. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

4. Install a writable copy of the samples then build and run the vectorAdd sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/0_Simple/vectorAdd
$ make
$ ./vectorAdd
```

4.3. Linux aarch64

For development on the aarch64 architecture.

4.3.1. L4T

When installing CUDA on L4T on aarch64, you must use the Debian Installer. The Debian Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is

a stand-alone installer with a large initial download. The instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

4.3.1.1. Debian Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the repository meta-data, update the apt-get cache, and install CUDA:

```
$ sudo dpkg --install cuda-repo-<distro>-<version>.<architecture>.deb
$ sudo apt-get update
$ sudo apt-get install cuda-toolkit-8-0
```

2. Reboot the system to load the NVIDIA drivers.
3. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

4. Install a writable copy of the samples then build and run the vectorAdd sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/0_Simple/vectorAdd
$ make
$ ./vectorAdd
```

4.4. Linux POWER8

For development on the POWER8 architecture.

4.4.1. Ubuntu

When installing CUDA on Ubuntu on POWER8, you must use the Debian Installer. The Debian Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. The instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

4.4.1.1. Debian Installer

Perform the following steps to install CUDA and verify the installation.

1. Install the repository meta-data, update the apt-get cache, and install CUDA:

```
$ sudo dpkg --install cuda-repo-<distro>-<version>.<architecture>.deb
$ sudo apt-get update
$ sudo apt-get install cuda
```

2. Reboot the system to load the NVIDIA drivers.
3. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

4. Install a writable copy of the samples then build and run the vectorAdd sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/0_Simple/vectorAdd
$ make
$ ./vectorAdd
```

4.4.2. Redhat / CentOS

When installing CUDA on Redhat on POWER8, you must use the RPM Installer. The RPM Installer is available as both a Local Installer and a Network Installer. The Network Installer allows you to download only the files you need. The Local Installer is a stand-alone installer with a large initial download. The instructions for the Local and Network variants are the same. For more details, refer to the Linux Installation Guide.

4.4.2.1. RPM Installer

Perform the following steps to install CUDA and verify the installation.

1. Install EPEL to satisfy the DKMS dependency by following the instructions at EPEL's website.
2. Install the repository meta-data, clean the yum cache, and install CUDA:

```
$ sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
$ sudo yum clean expire-cache
$ sudo yum install cuda
```

3. Reboot the system to load the NVIDIA drivers.
4. Set up the development environment by modifying the PATH and LD_LIBRARY_PATH variables:

```
$ export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64\
                         ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
```

5. Install a writable copy of the samples then build and run the vectorAdd sample:

```
$ cuda-install-samples-8.0.sh ~
$ cd ~/NVIDIA_CUDA-8.0_Samples/0_Simple/vectorAdd
$ make
$ ./vectorAdd
```

Notice

Trademarks

Copyright