

# CUDA QUICK START GUIDE

DU-05347-301\_v9.1 | April 2018



### TABLE OF CONTENTS

Chapter 1. Introduction1
Chapter 2. Windows 2
2.1. Network Installer 2
2.2. Local Installer
Chapter 3. Mac OSX6
3.1. Network Installer
3.2. Local Installer7
Chapter 4. Linux
4.1. Linux x86_648
4.1.1. Redhat / CentOS
4.1.1.1. RPM Installer
4.1.1.2. Runfile Installer
4.1.2. Fedora
4.1.2.1. RPM Installer 10
4.1.2.2. Runfile Installer10
4.1.3. SUSE Linux Enterprise Server11
4.1.3.1. RPM Installer 11
4.1.3.2. Runfile Installer11
4.1.4. OpenSUSE12
4.1.4.1. RPM Installer 12
4.1.4.2. Runfile Installer13
4.1.5. Ubuntu
4.1.5.1. Debian Installer14
4.1.5.2. Runfile Installer14
4.2. Linux POWER
4.2.1. Ubuntu
4.2.1.1. Debian Installer15
4.2.2. Redhat
4.2.2.1. RPM Installer

# 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.

						X
🗿 🕞 🛛 📕 🕨 Computer 🕨 Local D	isk (C:) 🕨 ProgramData 🕨 N	VIDIA Corporation 🕨	CUDA Samples 🕨	v7.0 • 5_Simulations • nbo	ody 🕨 👻 🍫 Search nbo.	λ
Organize 👻 Include in library 👻	Share with 🔻 New fold	er			:= ▼ 🔳	(?
Name	Date modified	Туре	Size			
퉬 doc	2/24/2015 2:30 PM	File folder				
b bodysystem.h	2/3/2015 5:07 AM	C/C++ Header	9 KB			
bodysystemcpu.h	2/3/2015 5:07 AM	C/C++ Header	2 KB			
bodysystemcpu_impl.h	2/3/2015 5:07 AM	C/C++ Header	7 KB			
bodysystemcuda.cu	2/3/2015 5:07 AM	CU File	10 KB			
h bodysystemcuda.h	2/3/2015 5:07 AM	C/C++ Header	3 KB			
bodysystemcuda_impl.h	2/3/2015 5:07 AM	C/C++ Header	12 KB			
galaxy_20K.bin	9/3/2014 3:48 PM	BIN File	840 KB			
nbody.cpp	2/3/2015 5:07 AM	C++ Source	44 KB			
🖉 nbody_vs2010.sln	2/18/2015 12:55 AM	Microsoft Visual S	1 KB			
nbody_vs2010.vcxproj	2/18/2015 12:55 AM	VC++ Project	6 KB			
nbody_vs2012.sln	2/18/2015 12:55 AM	Microsoft Visual S	1 KB			
nbody vs2012.vcxproj	2/18/2015 12:55 AM	VC++ Project	6 KB			
nbody_vs2013.sln	2/18/2015 12:55 AM	Microsoft Visual S	1 KB			
nbody_vs2013.vcxproj	2/18/2015 12:55 AM	VC++ Project	6 KB			
readme.txt	2/18/2015 12:55 AM	TXT File	2 KB			
😬 render_particles.cpp	2/3/2015 5:07 AM	C++ Source	12 KB			
h] render_particles.h	2/3/2015 5:07 AM	C/C++ Header	2 KB			
n tipsy.h	9/3/2014 3:48 PM	C/C++ Header	4 KB			

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

ordi nebody v2010 - Microsoft Visual Studio (Administration)	_ D X
File Edit View Project Build Debug Team Nsight Data Tools Test Analyze Window Help	
記・記・22月2月  おね2月1ク・ページージージーション bebug ・ 1x4 ・ 122 codaDeviceReset ・ 122 音道22 美国盤ロ・11日世話。	
1919 % % % M M M G G G G G G G G G G G G G G	
Solution Explorer     P X     Output X     Find Results1 Find Symbol Results	Properties 👻 🕈 🗙
Show output from: Build · V 40 A R	nbody_vs2010 Solution Propertie -
Solution / hbody vs2010 (L project) 1> 1	🔁 24 🔟
3 Body 1 Body 2 Body 1 Body 1 Body 2 Body 1 Body 2 Body 1 Body 2 Body 3 Body <p< th=""><th><ul> <li>A filesce</li> <li>Antropy adody u-2020</li> <li>Antropy adody u-2020</li> <li>Antropy adody u-2020</li> <li>Path</li> <li>C/Program/Deta/V</li> <li>Startup proje ebody</li> </ul></th></p<>	<ul> <li>A filesce</li> <li>Antropy adody u-2020</li> <li>Antropy adody u-2020</li> <li>Antropy adody u-2020</li> <li>Path</li> <li>C/Program/Deta/V</li> <li>Startup proje ebody</li> </ul>
D Device and	
15	
1>Time Elapsed 00:00:41.57 	(Name)
	The name of the solution file.
Build succeeded	4

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

						- 0	23
🚰 💽 🛛 🚺 🕨 Computer 🕨 Local D	isk (C:) 🕨 ProgramData 🕨 N	VIDIA Corporation 🕨	CUDA Samples	▶ v7.0 ▶ bin ▶ win64 ▶ Debug	<b>-</b> <sup>4</sup> ∳	Search De	b 🔎
Organize 👻 Include in library 💌	Share with 🔻 New fold	er			1		0
Name	Date modified	Туре	Size				
freeglut.dll	9/3/2014 3:48 PM	DLL File	304 KB				
FreeImage64.dll	9/3/2014 3:48 PM	DLL File	2,969 KB				
alew64.dll	9/3/2014 3:48 PM	DLL File	224 KB				
🔲 nbody.exe	2/24/2015 2:40 PM	Application	1,896 KB				
🗐 nbody.exp	2/24/2015 2:40 PM	Exports Library File	1 KB				
剜 nbody.ilk	2/24/2015 2:40 PM	Incremental Linke	3,756 KB				
😪 nbody.lib	2/24/2015 2:40 PM	Object File Library	2 KB				
🐏 nbody.pdb	2/24/2015 2:40 PM	Program Debug D	4,483 KB				

### 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.

Ca Kate Computer + Local Disk (C:)	▶ ProgramData ▶ N	VIDIA Corporation 🕨	CUDA Samples	v7.0 • 5_Simulations	▶ nbody ▶	→ ↓ Search nbo ↓	ρ
	-						5
Organize   Include in library   Sha	re with 🔻 New fold	er				iii 🔹 🔟 🔞	1
Name	Date modified	Туре	Size				
\mu doc	2/24/2015 2:30 PM	File folder					
bodysystem.h	2/3/2015 5:07 AM	C/C++ Header	9 KB				
bodysystemcpu.h	2/3/2015 5:07 AM	C/C++ Header	2 KB				
bodysystemcpu_impl.h	2/3/2015 5:07 AM	C/C++ Header	7 KB				
bodysystemcuda.cu	2/3/2015 5:07 AM	CU File	10 KB				
bodysystemcuda.h	2/3/2015 5:07 AM	C/C++ Header	3 KB				
bodysystemcuda_impl.h	2/3/2015 5:07 AM	C/C++ Header	12 KB				
galaxy_20K.bin	9/3/2014 3:48 PM	BIN File	840 KB				
en nbody.cpp	2/3/2015 5:07 AM	C++ Source	44 KB				
😕 nbody_vs2010.sln	2/18/2015 12:55 AM	Microsoft Visual S	1 KB				
🔐 nbody_vs2010.vcxproj	2/18/2015 12:55 AM	VC++ Project	6 KB				
nbody_vs2012.sln	2/18/2015 12:55 AM	Microsoft Visual S	1 KB				
nbody vs2012.vcxproj	2/18/2015 12:55 AM	VC++ Project	6 KB				
nbody_vs2013.sln	2/18/2015 12:55 AM	Microsoft Visual S	1 KB				
Inbody_vs2013.vcxproj	2/18/2015 12:55 AM	VC++ Project	6 KB				
📄 readme.txt	2/18/2015 12:55 AM	TXT File	2 KB				
🐏 render_particles.cpp	2/3/2015 5:07 AM	C++ Source	12 KB				
h render_particles.h	2/3/2015 5:07 AM	C/C++ Header	2 KB				
h tipsy.h	9/3/2014 3:48 PM	C/C++ Header	4 KB				

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

oor mbody.v2010 - Microsoft Visual Studio (Administrator)	
File Edit View Project Build Debug Team Nsight Data Tools Test Analyze Window Help	
記・記・望」#  ※ね込り・ページー以  > Debug ・ #4 ・   図 cutaDexiceFeet ・   収 学品 32 次 影響 ロ・:   回 西 古 :	
111 % % % M M M G G E E G G E C 4 .	
🕌 Solution Explorer 🔍 🕂 X Output 🗴 Find Results 1 Find Symbol Results 🔹 🗸 -	Properties 👻 🕂 🗙
Show output from: Build - 🖓 🏟 🛸	nbody_vs2010 Solution Propertie •
<pre>View Weeks USB/0 (prote)</pre>	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
13bilid succeeded. D	
1)lme Labjed W0:0041.5/	(Name)
42 Soldela ₱) term E. ₱2 Cless V	The name of the solution file.
Build successed	

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

					_ Ο Σ
🔾 🛛 🚺 🕨 Computer 🕨 Lo	cal Disk (C:) 🔸 ProgramData 🕨 N	VIDIA Corporation 🕨	CUDA Samples	▶ v7.0 ▶ bin ▶ win64 ▶ Debug	✓ ✓ Search Deb
Organize 👻 Include in library	✓ Share with ▼ New fold	ler			i≡ <b>-</b> □ (
Name	Date modified	Туре	Size		
freeglut.dll	9/3/2014 3:48 PM	DLL File	304 KB		
FreeImage64.dll	9/3/2014 3:48 PM	DLL File	2,969 KB		
alew64.dll	9/3/2014 3:48 PM	DLL File	224 KB		
🖳 nbody.exe	2/24/2015 2:40 PM	Application	1,896 KB		
nbody.exp	2/24/2015 2:40 PM	Exports Library File	1 KB		
a) nbody.ilk	2/24/2015 2:40 PM	Incremental Linke	3,756 KB		
🐊 nbody.lib	2/24/2015 2:40 PM	Object File Library	2 KB		
📔 nbody.pdb	2/24/2015 2:40 PM	Program Debug D	4,483 KB		

# 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-9.1/bin${PATH:+:${PATH}}
$ export DYLD_LIBRARY_PATH=/Developer/NVIDIA/CUDA-9.1/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-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_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:

- 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-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_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-9.1/bin${PATH:+:${PATH}}
```

- \$ export LD\_LIBRARY\_PATH=/usr/local/cuda-9.1/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-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_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:

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

```
$ cuda-install-samples-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_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 standalone 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-9.1/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-9.1/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-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_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-9.1/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-9.1/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-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_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:

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

```
$ cuda-install-samples-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_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-9.1/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-9.1/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-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_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 standalone 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:

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

```
$ cuda-install-samples-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_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:

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

```
$ cuda-install-samples-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_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 standalone 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, install GPG key, update the apt-get cache, and install CUDA:

```
$ sudo dpkg --install cuda-repo-<distro>-<version>.<architecture>.deb
$ sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/
compute/cuda/repos/<distro>/<architecture>/7fa2af80.pub
$ 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:

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

```
$ cuda-install-samples-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_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-9.1/bin${PATH:+:${PATH}}
$ export LD_LIBRARY_PATH=/usr/local/cuda-9.1/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-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_Samples/5_Simulations/nbody
$ make
$ ./nbody
```

```
4.2. Linux POWER
```

For development on the IBM Power platform (POWER8 and POWER9 architectures).

### 4.2.1. Ubuntu

When installing CUDA on Ubuntu on POWER, 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
```

- 2. Reboot the system to load the NVIDIA drivers.
- 3. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:

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

```
$ cuda-install-samples-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_Samples/0_Simple/vectorAdd
$ make
$ ./vectorAdd
```

### 4.2.2. Redhat

When installing CUDA on Redhat on POWER, 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 standalone 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.

#### 4.2.2.1. 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:

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

```
$ cuda-install-samples-9.1.sh ~
$ cd ~/NVIDIA_CUDA-9.1_Samples/0_Simple/vectorAdd
$ make
$ ./vectorAdd
```

#### Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

#### Trademarks

NVIDIA and the NVIDIA logo are trademarks or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

#### Copyright

© 2015-2018 NVIDIA Corporation. All rights reserved.

