**[CUDA Quick Start Guide](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "abstract)**

Minimal first-steps instructions to get CUDA running on a standard system.

**[1. Introduction](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "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](http://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/) and [Linux Installation Guide](http://docs.nvidia.com/cuda/cuda-installation-guide-linux/).

The CUDA installation packages can be found on the [CUDA Downloads Page](https://developer.nvidia.com/cuda-downloads/).

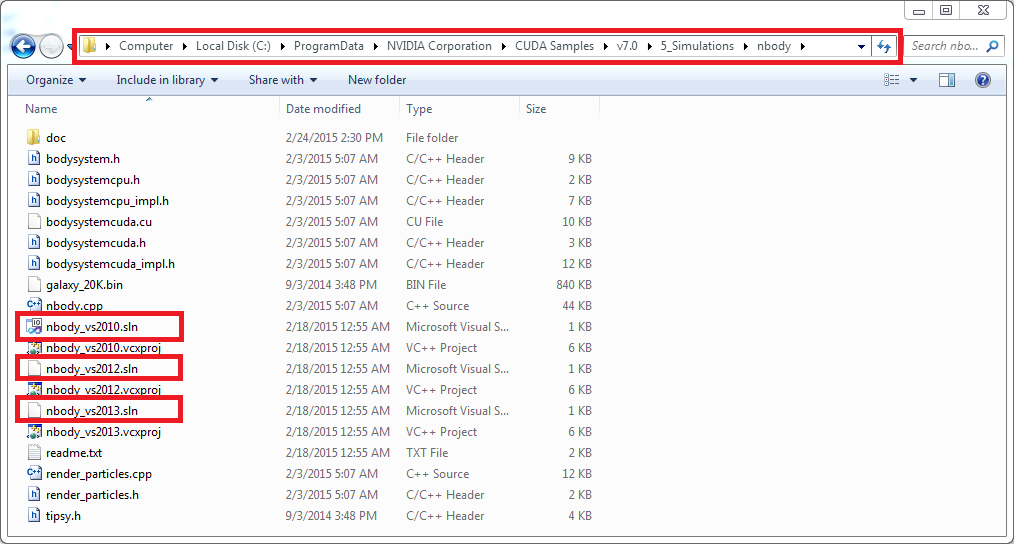
**[2. Windows](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "windows)**

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](http://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/).

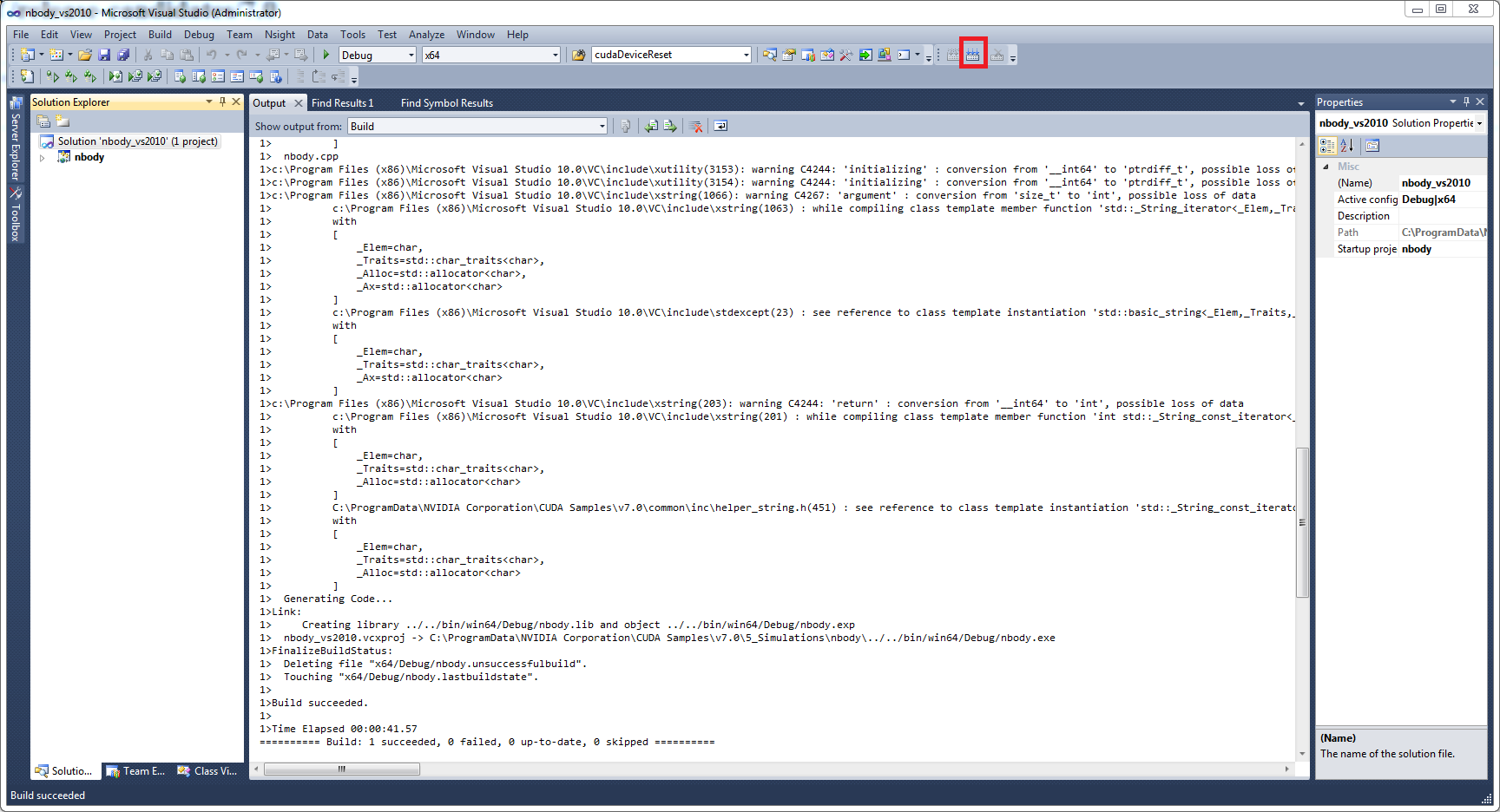
**[2.1. Network Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "windows-network)**

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

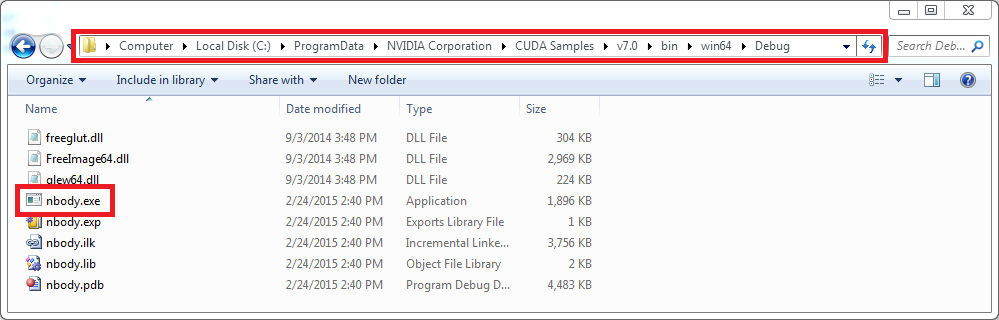
1. Launch the downloaded installer package.
2. Read and accept the EULA.
3. Select "next" to download and install all components.
4. Once the download completes, the installation will begin automatically.
5. Once the installation completes, click "next" to acknowledge the Nsight Visual Studio Edition 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.



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



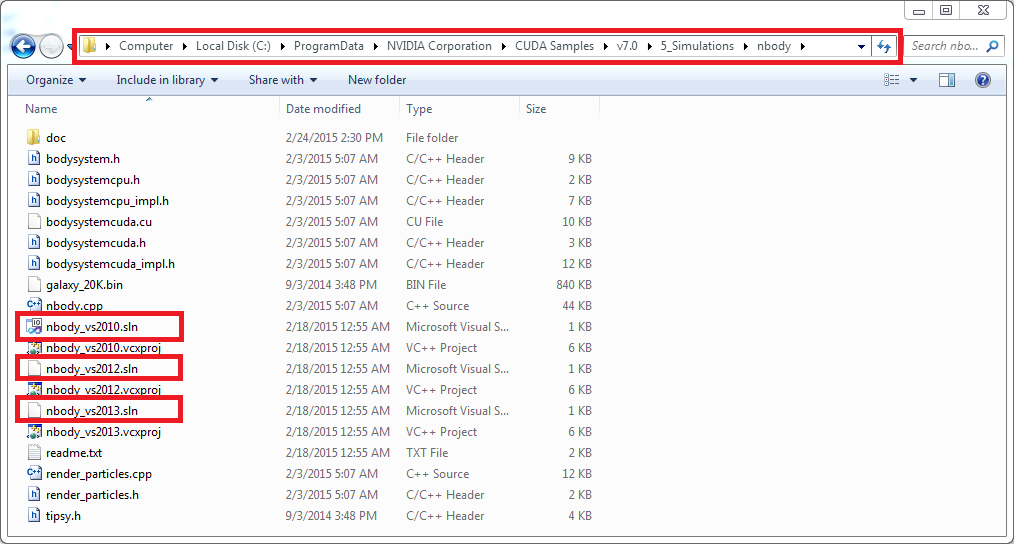
1. Navigate to the CUDA Samples' build directory and run the nbody sample.



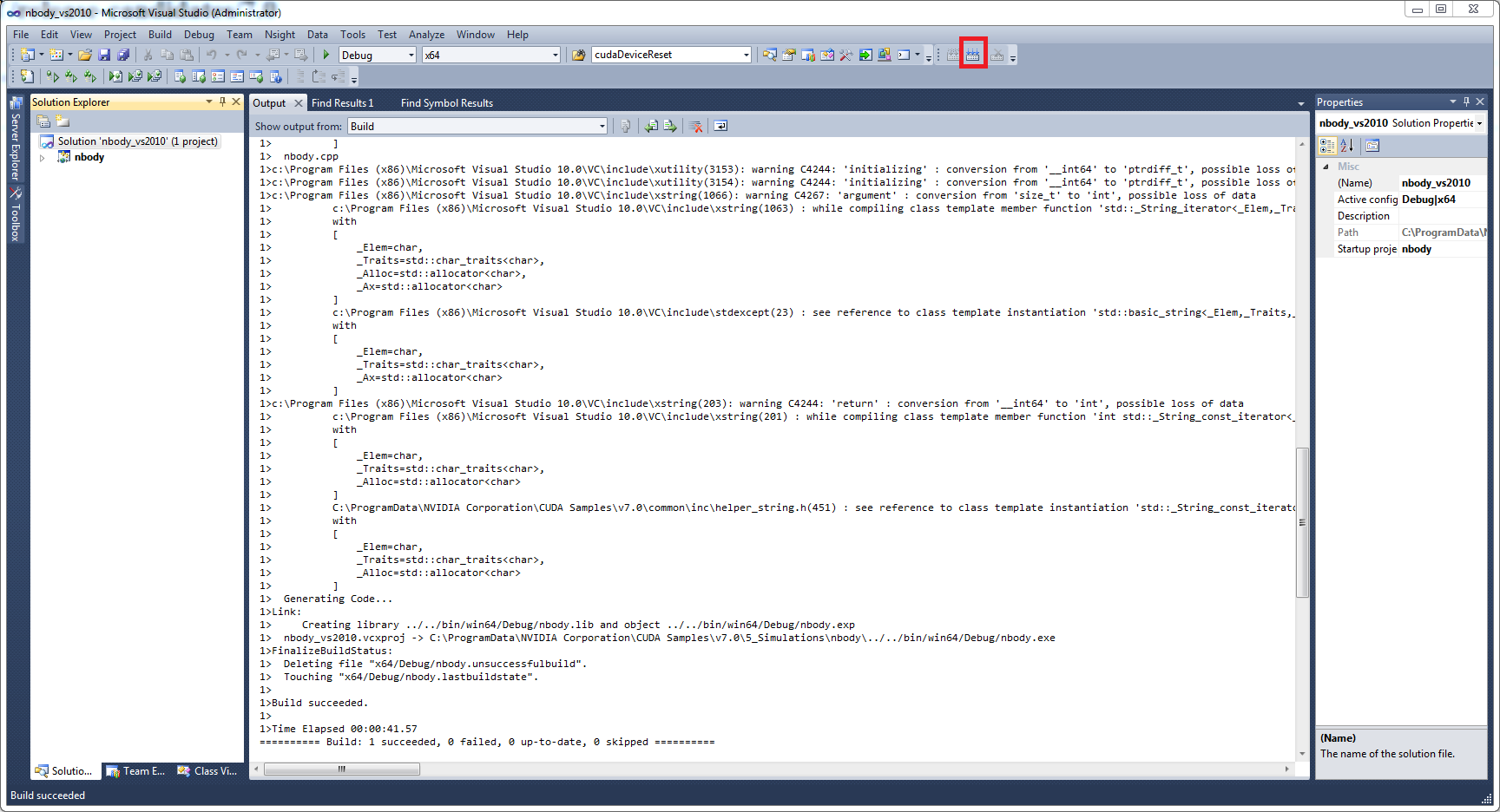
**[2.2. Local Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "windows-local)**

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

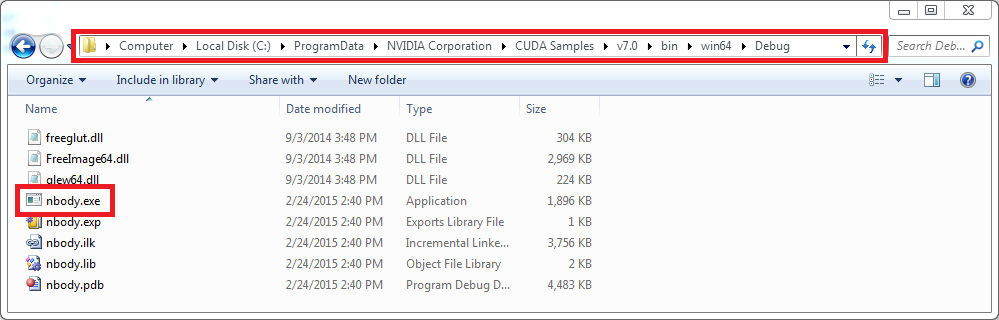
1. Launch the downloaded installer package.
2. Read and accept the EULA.
3. Select "next" to install all components.
4. Once the installation completes, click "next" to acknowledge the Nsight Visual Studio Edition installation summary.
5. Click "close" to close the installer.
6. Navigate to the CUDA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



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



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



**[3. Linux](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "linux)**

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

**[3.1. Linux x86\_64](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "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](http://docs.nvidia.com/cuda/cuda-installation-guide-linux/) for more details.

**[3.1.1. Redhat / CentOS](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "redhat-x86_64)**

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](http://docs.nvidia.com/cuda/cuda-installation-guide-linux/).

**[3.1.1.1. RPM Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "redhat-x86_64-rpm)**

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](https://fedoraproject.org/wiki/EPEL" \t "_blank).
2. **Enable optional repos**:

On **RHEL 7 Linux** only, execute the following steps to enable optional repositories.

* + **On x86\_64 workstation:**

**$** subscription-manager repos --enable=rhel-7-workstation-optional-rpms

* + **On POWER9 system:**

**$** subscription-manager repos --enable=rhel-7-for-power-9-optional-rpms

* + **On x86\_64 server:**

**$** subscription-manager repos --enable=rhel-7-server-optional-rpms

1. Install the repository meta-data, clean the yum cache, and install CUDA:
2. **$** sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
3. **$** sudo yum clean expire-cache

**$** sudo yum install cuda

1. Reboot the system to load the NVIDIA drivers.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the nbody sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/5\_Simulations/nbody
4. **$** make

**$** ./nbody

**[3.1.1.2. Runfile Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "redhat-x86_64-run)**

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:
   2. blacklist nouveau

options nouveau modeset=0

* 1. Regenerate the kernel initramfs:

**$** sudo dracut --force

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

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

**$** sudo nvidia-xconfig

1. Reboot the system to load the graphical interface.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the nbody sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/5\_Simulations/nbody
4. **$** make

**$** ./nbody

**[3.1.2. Fedora](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "fedora-x86_64)**

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](http://docs.nvidia.com/cuda/cuda-installation-guide-linux/).

**[3.1.2.1. RPM Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "fedora-x86_64-rpm)**

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'

1. Install the repository meta-data, clean the dnf cache, and install CUDA:
2. **$** sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
3. **$** sudo dnf clean expire-cache

**$** sudo dnf install cuda

1. Reboot the system to load the NVIDIA drivers.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the nbody sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/5\_Simulations/nbody
4. **$** make

**$** ./nbody

**[3.1.2.2. Runfile Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "fedora-x86_64-run)**

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:
   2. blacklist nouveau

options nouveau modeset=0

* 1. Regenerate the kernel initramfs:

**$** sudo dracut --force

* 1. Run the below command:

**$** sudo grub2-mkconfig -o /boot/grub2/grub.cfg

* 1. Reboot the system.

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

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

**$** sudo nvidia-xconfig

1. Reboot the system to load the graphical interface.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the nbody sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/5\_Simulations/nbody
4. **$** make

**$** ./nbody

**[3.1.3. SUSE Linux Enterprise Server](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "sles-x86_64)**

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](http://docs.nvidia.com/cuda/cuda-installation-guide-linux/).

**[3.1.3.1. RPM Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "sles-x86_64-rpm)**

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

1. Install the repository meta-data, refresh the Zypper cache, and install CUDA:
2. **$** sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
3. **$** sudo SUSEConnect --product PackageHub/15/x86\_64
4. **$** sudo zypper refresh

**$** sudo zypper install cuda

1. Add the user to the video group:

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

1. Reboot the system to load the NVIDIA drivers.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the vectorAdd sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/0\_Simple/vectorAdd
4. **$** make

**$** ./vectorAdd

**[3.1.3.2. Runfile Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "sles-x86_64-run)**

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

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

**$** sudo nvidia-xconfig

1. Reboot the system to load the graphical interface.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the vectorAdd sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/0\_Simple/vectorAdd
4. **$** make

**$** ./vectorAdd

**[3.1.4. OpenSUSE](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "suse-x86_64)**

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](http://docs.nvidia.com/cuda/cuda-installation-guide-linux/).

**[3.1.4.1. RPM Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "suse-x86_64-rpm)**

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

1. Install the repository meta-data, refresh the Zypper cache, and install CUDA:
2. **$** sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
3. **$** sudo zypper refresh

**$** sudo zypper install cuda

1. Add the user to the video group:

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

1. Reboot the system to load the NVIDIA drivers.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the nbody sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/5\_Simulations/nbody
4. **$** make

**$** ./nbody

**[3.1.4.2. Runfile Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "suse-x86_64-run)**

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:
   2. blacklist nouveau

options nouveau modeset=0

* 1. Regenerate the kernel initrd:

**$** sudo /sbin/mkinitrd

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

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

**$** sudo nvidia-xconfig

1. Reboot the system to load the graphical interface.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the nbody sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/5\_Simulations/nbody
4. **$** make

**$** ./nbody

**[3.1.5. WSL](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "wsl-x86_64-rpm)**

These instructions must be used if you are installing in a WSL environment. Do not use the Ubuntu instructions in this case.

1. **Install repository meta-data**

**$** sudo dpkg -i cuda-repo-<distro>\_<version>\_<architecture>.deb

1. **Install the CUDA public GPG key**

When installing using the local repo:

**$** sudo apt-key add /var/cuda-repo-<version>/7fa2af80.pub

When installing using the network repo:

**$** sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/cuda/repos/<distro>/<architecture>/7fa2af80.pub

Pin file to prioritize CUDA repository:

**$** wget https://developer.download.nvidia.com/compute/cuda/repos/<distro>/<architecture>/cuda-<distro>.pin

**$** sudo mv cuda-<distro>.pin /etc/apt/preferences.d/cuda-repository-pin-600

1. **Update the Apt repository cache and install CUDA**
2. **$** sudo apt-get update

**$** sudo apt-get install cuda

**[3.1.6. Ubuntu](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "ubuntu-x86_64)**

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](http://docs.nvidia.com/cuda/cuda-installation-guide-linux/).

**[3.1.6.1. Debian Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "ubuntu-x86_64-deb)**

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

**$** sudo apt-get install cuda

1. Reboot the system to load the NVIDIA drivers.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the nbody sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/5\_Simulations/nbody
4. **$** make

**$** ./nbody

**[3.1.6.2. Runfile Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "ubuntu-x86_64-run)**

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:
   2. blacklist nouveau

options nouveau modeset=0

* 1. Regenerate the kernel initramfs:

**$** sudo update-initramfs -u

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

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

**$** sudo nvidia-xconfig

1. Reboot the system to load the graphical interface.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the nbody sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/5\_Simulations/nbody
4. **$** make

**$** ./nbody

**[3.1.7. Debian](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "debian-x86_64)**

When installing CUDA on Debian 10, 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. For more details, refer to the [Linux Installation Guide](http://docs.nvidia.com/cuda/cuda-installation-guide-linux/).

**[3.1.7.1. Debian Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "debian-x86_64-deb)**

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:
2. **$** sudo dpkg -i cuda-repo-<distro>\_<version>\_<architecture>.deb
3. **$** sudo apt-key adv --fetch-keys https://developer.download.nvidia.com/compute/cuda/repos/debian10/x86\_64/7fa2af80.pub
4. **$** sudo add-apt-repository contrib
5. **$** sudo apt-get update

**$** sudo apt-get -y install cuda

1. Reboot the system to load the NVIDIA drivers.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the nbody sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/5\_Simulations/nbody
4. **$** make

**$** ./nbody

**[3.1.7.2. Runfile Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "debian-x86_64-run)**

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:
   2. blacklist nouveau

options nouveau modeset=0

* 1. Regenerate the kernel initramfs:

**$** sudo update-initramfs -u

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

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

**$** sudo nvidia-xconfig

1. Reboot the system to load the graphical interface.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the nbody sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/5\_Simulations/nbody
4. **$** make

**$** ./nbody

**[3.2. Linux POWER8](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "linux-ppc64le)**

For development on the POWER8 architecture.

**[3.2.1. Ubuntu](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "ubuntu-ppc64le)**

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](http://docs.nvidia.com/cuda/cuda-installation-guide-linux/).

**[3.2.1.1. Debian Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "ubuntu-ppc64le-deb)**

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:
2. **$** sudo dpkg --install cuda-repo-<distro>-<version>.<architecture>.deb
3. **$** sudo apt-get update

**$** sudo apt-get install cuda

1. Reboot the system to load the NVIDIA drivers.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the vectorAdd sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/0\_Simple/vectorAdd
4. **$** make

**$** ./vectorAdd

**[3.2.2. Redhat / CentOS](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "redhat-ppc64le)**

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](http://docs.nvidia.com/cuda/cuda-installation-guide-linux/).

**[3.2.2.1. RPM Installer](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "redhat-ppc64le-rpm)**

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](https://fedoraproject.org/wiki/EPEL" \t "_blank).
2. Install the repository meta-data, clean the yum cache, and install CUDA:
3. **$** sudo rpm --install cuda-repo-<distro>-<version>.<architecture>.rpm
4. **$** sudo yum clean expire-cache

**$** sudo yum install cuda

1. Reboot the system to load the NVIDIA drivers.
2. Set up the development environment by modifying the PATH and LD\_LIBRARY\_PATH variables:
3. **$** export PATH=/usr/local/cuda-11.1/bin${PATH:+:${PATH}}
4. **$** export LD\_LIBRARY\_PATH=/usr/local/cuda-11.1/lib64\

${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

1. Install a writable copy of the samples then build and run the vectorAdd sample:
2. **$** cuda-install-samples-11.1.sh ~
3. **$** cd ~/NVIDIA\_CUDA-11.1\_Samples/0\_Simple/vectorAdd
4. **$** make

**$** ./vectorAdd

**[Notices](https://docs.nvidia.com/cuda/cuda-quick-start-guide/index.html" \l "notices-header)**

**Notice**

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation (“NVIDIA”) makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer (“Terms of Sale”). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND 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. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

**VESA DisplayPort**

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

**HDMI**

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

**OpenCL**

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

**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**

© 2015-2020 NVIDIA Corporation. All rights reserved.

This product includes software developed by the Syncro Soft SRL (http://www.sync.ro/).