Virtual GPU Software Client Licensing

User Guide

DU-07757-001 _v14.0 through 14.4 | December 2022

Table of Contents

Chapter 1. Introduction to NVIDIA vGPU Software Licensing 1
1.1. NVIDIA vGPU Software Licensed Products 1
1.2. How NVIDIA vGPU Software Licensing Is Enforced 2
1.2.1. Software Enforcement of NVIDIA vGPU Software Licensing 4
1.2.2. EULA-Only Enforcement of NVIDIA vGPU Software Licensing 6
1.3. Display Resolutions for Physical GPUs 6
Chapter 2. Configuring a Licensed Client of NVIDIA License System 8
2.1. Configuring a Licensed Client on Windows 8
2.2. Configuring a Licensed Client on Linux 10
Chapter 3. Licensing NVIDIA vGPU Software from the Legacy License Server 13
3.1. Licensing NVIDIA vGPU from the Legacy License Server 13
3.1.1. Licensing an NVIDIA vGPU on Windows 13
3.1.2. Licensing an NVIDIA vGPU on Linux 15
3.2. Licensing a Physical GPU from the Legacy License Server 17
3.2.1. Licensing a Physical GPU for vWS on Windows 17
3.2.2. Removing a vWS License from a Physical GPU on Windows 19
3.2.3. Licensing a Physical GPU on Linux 19
3.3. Licensing NVIDIA vGPU Software on Linux by Using NVIDIA X Server Settings 22
3.3.1. Enabling License Management in NVIDIA X Server Settings 23
3.3.2. Licensing an NVIDIA vGPU on Linux by Using NVIDIA X Server Settings 23
3.3.3. Licensing a Physical GPU for vWS or vCS on Linux by Using NVIDIA X Server Settings 24
Chapter 4. Advanced Topics 27
4.1. Setting the Client Host Identifier for License Checkouts 27
4.2. Licenses Obtained After Boot 28
4.3. Operating with Intermittent Connectivity to the License Server 28
4.4. Applying Windows License Settings Through the Windows Registry 28
Chapter 5. Troubleshooting 33
5.1. Known Issues 33
5.2. Verifying the NVIDIA vGPU Software License Status of a Licensed Client 33
5.3. Licensing Event Logs 34
5.4. Troubleshooting Steps 36

List of Figures

Figure 1. Software Enforcement of NVIDIA vGPU Software Licensing 5

Figure 2. Managing vGPU licensing in NVIDIA Control Panel 14

Figure 3. Managing vWS Licensing in NVIDIA Control Panel 18

Figure 4. Configuring vGPU Licensing Through Windows Registry Settings 32

List of Tables

Table 1. NVIDIA vGPU Software Licensed Products 1

Table 2. NVIDIA vGPU Software License Enforcement by Deployment Type 2

Table 3. Licensing Registry Settings 29

Chapter 1. Introduction to NVIDIA vGPU Software Licensing

The following products are available as licensed products on NVIDIA GPUs that support NVIDIA vGPU software:

- NVIDIA® RTX Virtual Workstation (vWS)
- NVIDIA Virtual Compute Server (vCS)
- NVIDIA Virtual PC (vPC)
- NVIDIA Virtual Applications (vApps)

This guide describes these licensed products and how to enable and use them on supported hardware.

1.1. NVIDIA vGPU Software Licensed Products

The available NVIDIA vGPU software licensed products support different classes of NVIDIA vGPU software deployments.

Table 1. NVIDIA vGPU Software Licensed Products

NVIDIA vGPU Software Licensed Product	Target Users	Supported NVIDIA vGPU Software Deployments
vApps	Users of PC-level applications and server-based desktops that use Citrix Virtual Apps and Desktops, VMware Horizon, RDSH, or other app streaming or session-based solutions	▶ A-series NVIDIA vGPUs ▶ GPU pass through ▶ Microsoft DDA ▶ VMware vDGA ▶ Bare metal
vPC	Users of business virtual desktops who require a great user experience with PC	▶ B-series NVIDIA vGPUs

NVIDIA vGPU Software Licensed Product	Target Users	Supported NVIDIA vGPU Software Deployments
	applications for Windows, web browsers, and high-definition video	▶ Microsoft RemoteFX vGPU ▶ VMware vSGA
vCS	Users of compute-intensive virtual servers for artificial intelligence (AI), deep learning, or high-performance computing (HPC) workloads	▶ C-series NVIDIA vGPUs ▶ GPU pass through ▶ Microsoft DDA ▶ VMware vDGA
vWS	Users of mid-range and high-end workstations who require access to remote professional graphics applications with full performance on any device anywhere	▶ Q-series NVIDIA vGPUs ▶ C-series NVIDIA vGPUs ▶ B-series NVIDIA vGPUs ▶ GPU pass through ▶ Microsoft DDA ▶ VMware vDGA ▶ Bare metal

Note:

- Microsoft DDA and VMware vDGA are functionally equivalent to GPU pass through.
- Microsoft RemoteFX vGPU and VMware vSGA are graphics virtualization technologies.
- Microsoft DDA requires Windows Server 2016 with the Hyper-V role.

1.2. How NVIDIA vGPU Software Licensing Is Enforced

Depending on the NVIDIA vGPU software deployment, licensing is enforced either through software or only through the end-user license agreement (EULA).

Table 2. NVIDIA vGPU Software License Enforcement by Deployment Type

NVIDIA vGPU Software Deployment	Required NVIDIA vGPU Software License	Enforcement
A-series NVIDIA vGPU	vApps	Software See Note (1).

NVIDIA vGPU Software Deployment	Required NVIDIA vGPU Software License	Enforcement
B-series NVIDIA vGPU	vPC or vWS	Software See Note (2).
C-series NVIDIA vGPU	vCS or vWS	Software See Note (3).
Q-series NVIDIA vGPU	vWS	Software See Note (4).
GPU pass through for workstation or professional 3D graphics	vWS	Software
GPU pass through for compute-intensive virtual servers	vCS	Software See Note (5).
GPU pass through for PC-level applications	vApps	EULA only
Microsoft DDA for workstation or professional 3D graphics	vWS	Software
Microsoft DDA for compute-intensive virtual servers	vCS	Software See Note (5).
Microsoft DDA for PC-level applications	vApps	EULA only
VMware vDGA for workstation or professional 3D graphics	vWS	Software
VMware vDGA for compute-intensive virtual servers	vCS	Software See Note (5).
VMware vDGA for PC-level applications	vApps	EULA only
Bare metal for workstation or professional 3D graphics	vWS	Software
Bare metal for PC-level applications	vApps	EULA only
Microsoft RemoteFX vGPU	vPC	EULA only
VMware vSGA	vPC	EULA only

Note:

1. For A-series NVIDIA vGPU deployments, one vApps license is required for each concurrent user, but licensing for only one of these users is enforced through software. Licensing for the remaining users is enforced through the EULA.
2. The vWS license edition includes the vPC license edition: virtual GPUs that require a vPC license can also be used with a vWS license.
3. For C-series NVIDIA vGPU deployments, one license per vGPU assigned to a VM is enforced through software. This license is valid for up to 10 vGPU instances on a single GPU or for the assignment to a VM of one vGPU that is assigned all the physical GPU's frame buffer. When multiple C-series vGPUs are assigned to a single VM, one license for each vGPU assigned to the VM is required. One license is enforced through software. The remaining licenses are enforced through the EULA.
4. When multiple Q-series vGPUs are assigned to a single VM, only one license for the VM is required, **not** one license for each vGPU assigned to the VM.
5. For GPU pass through, Microsoft DDA, and VMware vDGA deployments for compute-intensive virtual servers, one license per VM is enforced through software. This license is

valid for the assignment of one physical GPU to a VM. One additional license is required for each additional physical GPU assigned to a VM. Any additional licenses are enforced through the EULA.

1.2.1. Software Enforcement of NVIDIA vGPU Software Licensing

When a Virtual Machine (VM) that is running NVIDIA vGPU software is first booted, the virtual GPU or physical GPU assigned to the VM operates at full capability. When licensing is enforced through software, the performance of the virtual GPU or physical GPU is degraded over time if the VM fails to obtain a license.

How the performance of an unlicensed vGPU or physical GPU is degraded depends on how much time has elapsed since the VM was booted and the NVIDIA vGPU software graphics driver was loaded.

Elapsed Time	Performance Degradation
20 minutes	▶ Frame rate is capped at 15 frames per second. ▶ The performance of applications and processes that use CUDA is degraded.
24 hours	▶ Frame rate is capped at 3 frames per second. ▶ CUDA stops working and CUDA API function calls fail. ▶ GPU resource allocations for a vGPU are limited, which will prevent some applications from rendering or running correctly, or will cause them to report errors when started.

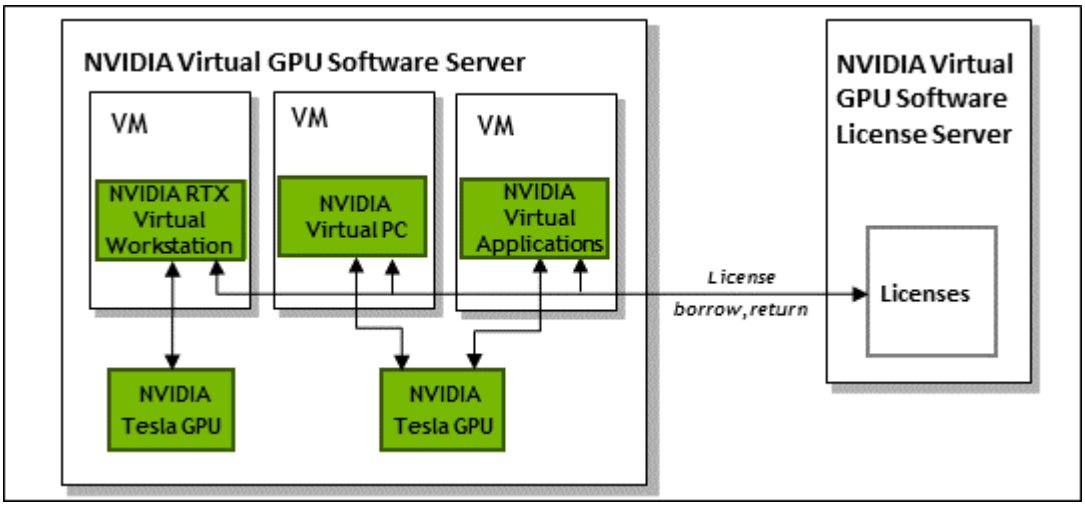
A VM obtains a license over the network from an NVIDIA vGPU software license server. The license is "checked out" or "borrowed" when the VM is booted. The VM retains the license until it is shut down. It then releases the license back to the license server.

Note: Information on setting up and running the NVIDIA vGPU software license server is provided in the documentation for the supported license servers:

- For NVIDIA License System, refer to *NVIDIA License System User Guide*.
- For the legacy license server, refer to *Virtual GPU License Server User Guide*.

Figure 1 illustrates how NVIDIA vGPU software licensing is enforced through software.

Figure 1. Software Enforcement of NVIDIA vGPU Software Licensing



Additional Considerations for NVIDIA vGPU Deployments

For NVIDIA vGPU deployments, the NVIDIA vGPU software automatically selects the correct type of license based on the vGPU type.

After you license NVIDIA vGPU, the VM that is set up to use NVIDIA vGPU is capable of running the full range of DirectX and OpenGL graphics applications.

Licensing settings persist across reboots and need only be modified if the license server address changes, or the VM is switched to running GPU pass through.

Additional Considerations for Physical GPU Deployments

The type of license required depends on how the physical GPU is deployed.

- In GPU pass-through mode on Windows, or in a bare-metal deployment on Windows or Linux, a physical GPU requires a vWS license.
- In GPU pass-through mode on Linux, a physical GPU requires a vWS license or a vCS license.

You can also operate a physical GPU in pass-through mode or a bare-metal deployment at reduced capability with a vApps license. vPC is not available on GPUs running in pass-through mode or bare-metal deployments.

1.2.2. EULA-Only Enforcement of NVIDIA vGPU Software Licensing

When licensing is enforced only through the EULA, no licenses are checked out from the license server. You **must** comply with the terms of the EULA by ensuring that **all** the licenses required for your NVIDIA vGPU software deployment have been purchased.

The `nvidia-smi -q` command indicates that the product is licensed even when no license is checked out from the license server.

If licensing for **all** your NVIDIA vGPU software deployments is enforced only through the EULA, you do not need to set up and run the NVIDIA vGPU software license server.

1.3. Display Resolutions for Physical GPUs

The display resolutions supported by a physical GPU depend on the NVIDIA GPU architecture and the NVIDIA vGPU software license that is applied to the GPU.

vWS Physical GPU Resolutions

GPUs that are licensed with a vWS license support a maximum combined resolution based on the number of available pixels, which is determined by the NVIDIA GPU architecture. You can choose between using a small number of high resolution displays or a larger number of lower resolution displays with these GPUs.

The following table lists the maximum number of displays per GPU at each supported display resolution for configurations in which all displays have the same resolution.

NVIDIA GPU Architecture	Available Pixels	Display Resolution	Displays per GPU
Pascal and later	66355200	7680×4320	2
		5120×2880 or lower	4
Maxwell	35389440	5120×2880	2
		4096×2160 or lower	4

The following table provides examples of configurations with a mixture of display resolutions.

NVIDIA GPU Architecture	Available Pixels	Available Pixel Basis	Maximum Displays	Sample Mixed Display Configurations
Pascal and later	66355200	2 7680×4320 displays	4	1 7680×4320 display plus 2 5120×2880 displays

NVIDIA GPU Architecture	Available Pixels	Available Pixel Basis	Maximum Displays	Sample Mixed Display Configurations
				1 7680×4320 display plus 3 4096×2160 displays
Maxwell	35389440	4 4096×2160 displays	4	1 5120×2880 display plus 2 4096×2160 displays

Note: You cannot use more than four displays even if the combined resolution of the displays is less than the number of available pixels from the GPU. For example, you cannot use five 4096×2160 displays with a GPU based on the NVIDIA Pascal architecture even though the combined resolution of the displays (44236800) is less than the number of available pixels from the GPU (66355200).

vApps or vCS Physical GPU Resolutions

GPUs that are licensed with a vApps or a vCS license support a single display with a fixed maximum resolution. The maximum resolution depends on the following factors:

- NVIDIA GPU architecture
- The NVIDIA vGPU Software license that is applied to the GPU
- The operating system that is running in the on the system to which the GPU is assigned

License	NVIDIA GPU Architecture	Operating System	Maximum Display Resolution	Displays per GPU
vApps	Pascal or later	Linux	2560×1600	1
	Pascal or later	Windows	1280×1024	1
	Maxwell	Windows and Linux	2560×1600	1
vCS	Pascal or later	Linux	4096×2160	1

Chapter 2. Configuring a Licensed Client of NVIDIA License System

A client with a network connection obtains a license by leasing it from a NVIDIA License System service instance. The service instance serves the license to the client over the network from a pool of floating licenses obtained from the NVIDIA Licensing Portal. The license is returned to the service instance when the licensed client is shut down.

Before configuring a licensed client, ensure that the following prerequisites are met:

- The NVIDIA vGPU software graphics driver is installed on the client.
- The client configuration token that you want to deploy on the client has been created from the NVIDIA Licensing Portal or the DLS as explained in *NVIDIA License System User Guide*.
- Ports 443 and 80 in your firewall or proxy must be open to allow HTTPS traffic between a service instance and its the licensed clients. These ports must be open for both CLS instances and DLS instances.

Note: For DLS releases **before** DLS 1.1, ports 8081 and 8082 were also required to be open to allow HTTPS traffic between a DLS instance and its licensed clients. Although these ports are no longer required, they remain supported for backward compatibility.

The graphics driver creates a default location in which to store the client configuration token on the client.

The process for configuring a licensed client is the same for CLS and DLS instances but depends on the OS that is running on the client.

2.1. Configuring a Licensed Client on Windows

Perform this task from the client.

1. **Physical GPUs only:** Add the `FeatureType` DWord (`REG_DWORD`) registry value to the Windows registry key `HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\GridLicensing`.

   Note:
   - If you're licensing an NVIDIA vGPU, the `FeatureType` DWord (`REG_DWORD`) registry value is **not** required. NVIDIA vGPU software automatically selects the correct type of license based on the vGPU type.
   - If you are upgrading an existing driver, this value is already set.

   Set this value to the feature type of a GPU in pass-through mode or a bare-metal deployment:

   - 0: NVIDIA Virtual Applications
   - 2: NVIDIA RTX Virtual Workstation

2. **Optional:** If you want store the client configuration token in a custom location, add the `ClientConfigTokenPath` String (`REG_SZ`) registry value to the Windows registry key `HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\GridLicensing`.

   Set the value to the full path to the folder in which you want to store the client configuration token for the client. You can use the syntax `\\`*`fully-qualified-domain-name`*`\`*`share-name`* for the path to the folder. By default, the client searches for the client configuration token in the `%SystemDrive%:\Program Files\NVIDIA Corporation\vGPU Licensing\ClientConfigToken` folder.

   By specifying a shared network drive mapped on the client, you can simplify the deployment of the same client configuration token on multiple clients. Instead of copying the client configuration token to each client individually, you can keep only one copy in the shared network drive.

3. **Since 14.3:** If a non-transparent proxy server is configured between your licensed client and a CLS instance, provide the address and port number of the proxy server.

   Note: Authenticated non-transparent proxy servers are **not** supported.

   Provide this information by adding the following registry values to the Windows registry key `HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\GridLicensing`.

   `ProxyServerAddress` String `(REG_SZ)`
   The address of the proxy server. The address can be a fully qualified domain name such as `iproxy1.example.com`, or an IP address such as `10.31.20.45`.

   `ProxyServerPort` String `(REG_SZ)`
   The port number of the proxy server.

4. If you are storing the client configuration token in a custom location, create the folder in which you want to store the client configuration token.

   If the folder is a shared network drive, ensure that the following conditions are met:

   - The folder is mapped locally on the client to the path specified in the `ClientConfigTokenPath` registry value.

- The COMPUTER object has the rights to access the folder on the shared network drive. The COMPUTER object requires these rights because the license service runs before any user logs in.

If you are storing the client configuration token in the default location, omit this step. The default folder in which the client configuration token is stored is created automatically after the graphics driver is installed.

5. Copy the client configuration token to the folder in which you want to store the client configuration token.

   Ensure that this folder contains only the client configuration token that you want to deploy on the client and no other files or folders. If the folder contains more than one client configuration token, the client uses the newest client configuration token in the folder.

   - If you want to store the client configuration token in the default location, copy the client configuration token to the `%SystemDrive%:\Program Files\NVIDIA Corporation\vGPU Licensing\ClientConfigToken` folder.
   - If you want to store the client configuration token in a custom location, copy the token to the folder that you created in the previous step.
6. Restart the `NvDisplayContainer` service.

The NVIDIA service on the client should now automatically obtain a license from the CLS or DLS instance.

After a Windows licensed client has been configured, options for configuring licensing for a network-based license server are no longer available in **NVIDIA Control Panel**.

2.2. Configuring a Licensed Client on Linux

Perform this task from the client.

1. As root, open the file `/etc/nvidia/gridd.conf` in a plain-text editor, such as `vi`.

```
$ sudo vi /etc/nvidia/gridd.conf
```

Note: You can create the `/etc/nvidia/gridd.conf` file by copying the supplied template file `/etc/nvidia/gridd.conf.template`.

2. Add the `FeatureType` configuration parameter to the file `/etc/nvidia/gridd.conf` on a new line as `FeatureType="value"`.

   value depends on the type of the GPU assigned to the licensed client that you are configuring.

GPU Type	Value
NVIDIA vGPU	1. NVIDIA vGPU software automatically selects the correct type of license based on the vGPU type.
Physical GPU	The feature type of a GPU in pass-through mode or a bare-metal deployment:

GPU Type	Value
	▶ 0: NVIDIA Virtual Applications ▶ 2: NVIDIA RTX Virtual Workstation ▶ 4: NVIDIA Virtual Compute Server

This example shows how to configure a licensed Linux client for NVIDIA Virtual Compute Server.

```
# /etc/nvidia/gridd.conf.template - Configuration file for NVIDIA Grid Daemon
…
# Description: Set Feature to be enabled
# Data type: integer
# Possible values:
# 0 => for unlicensed state
# 1 => for NVIDIA vGPU
# 2 => for NVIDIA RTX Virtual Workstation
# 4 => for NVIDIA Virtual Compute Server
FeatureType=4
...
```

3. **Since 14.3:** If a non-transparent proxy server is configured between your licensed client and a CLS instance, provide the address and port number of the proxy server.

   Note: Authenticated non-transparent proxy servers are **not** supported.

   Provide this information by adding the following configuration parameters to the file `/etc/nvidia/gridd.conf` on separate lines.

   ```
   ProxyServerAddress=address
   ProxyServerPort=port
   ```

   address
   The address of the proxy server. The address can be a fully qualified domain name such as `iproxy1.example.com`, or an IP address such as `10.31.20.45`.

   port
   The port number of the proxy server.

   This example sets the address of a proxy server to `10.31.20.45` and the port number to `3128`.

   ```
   # /etc/nvidia/gridd.conf.template - Configuration file for NVIDIA Grid Daemon
   …
   ProxyServerAddress=10.31.20.45
   ProxyServerPort=3128
   ...
   ```

4. Save your changes to the `/etc/nvidia/gridd.conf` file and close the file.
5. If you are storing the client configuration token in a custom location, create the directory in which you want to store the client configuration token.

   If the directory is a shared network directory, ensure that it is mounted locally on the client at the path specified in the `ClientConfigTokenPath` configuration parameter.

   If you are storing the client configuration token in the default location, omit this step. The default directory in which the client configuration token is stored is created automatically after the graphics driver is installed.

6. Copy the client configuration token to the directory in which you want to store the client configuration token.

   Ensure that this directory contains only the client configuration token that you want to deploy on the client and no other files or directories. If the directory contains more than one client configuration token, the client uses the newest client configuration token in the directory.

   - If you want to store the client configuration token in the default location, copy the client configuration token to the `/etc/nvidia/ClientConfigToken` directory.
   - If you want to store the client configuration token in a custom location, copy the token to the directory that you created in the previous step.

7. Ensure that the file access modes of the client configuration token allow the owner to read, write, and execute the token, and the group and others only to read the token.

   a). Determine the current file access modes of the client configuration token.

   ```
   # ls -l client-configuration-token-directory
   ```

   b). If necessary, change the mode of the client configuration token to 744.

   ```
   # chmod 744 client-configuration-token-directory/client_configuration_token_*.tok
   ```

   client-configuration-token-directory
   The directory to which you copied the client configuration token in the previous step.

8. Restart the `nvidia-gridd` service.

The NVIDIA service on the client should now automatically obtain a license from the CLS or DLS instance.

After a Linux licensed client has been configured, options for configuring licensing for a network-based license server are no longer available in **NVIDIA X Server Settings**.

Chapter 3. Licensing NVIDIA vGPU Software from the Legacy License Server

How to license NVIDIA vGPU software depends on whether NVIDIA vGPU software is being used for a vGPU or a physical GPU.

3.1. Licensing NVIDIA vGPU from the Legacy License Server

How to license NVIDIA vGPU depends on the guest OS that is running in the VM.

3.1.1. Licensing an NVIDIA vGPU on Windows

Perform this task from the guest VM to which the vGPU is assigned.

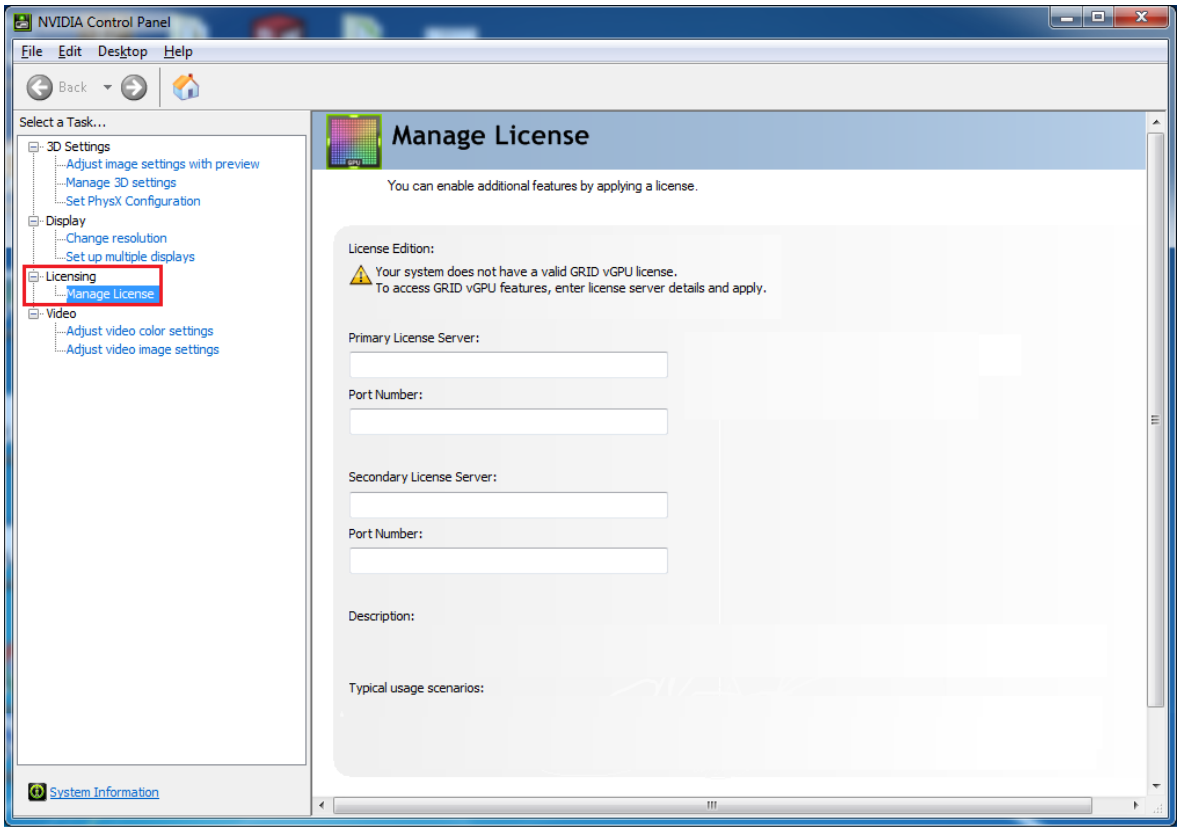
The **NVIDIA Control Panel** tool that you use to perform this task detects that a vGPU is assigned to the VM and, therefore, provides no options for selecting the license type. After you license the vGPU, NVIDIA vGPU software automatically selects the correct type of license based on the vGPU type.

1. Open **NVIDIA Control Panel**:
   - Right-click on the Windows desktop and select **NVIDIA Control Panel** from the menu.
   - Open **Windows Control Panel** and double-click the **NVIDIA Control Panel** icon.
2. In **NVIDIA Control Panel**, select the **Manage License** task in the **Licensing** section of the navigation pane.

   Note: If the **Licensing** section and **Manage License** task are not displayed in **NVIDIA Control Panel**, the system has been configured to hide licensing controls in **NVIDIA Control Panel**. For information about registry settings, refer to Applying Windows License Settings Through the Windows Registry.

   The **Manage License** task pane shows that NVIDIA vGPU is currently unlicensed.

Figure 2. Managing vGPU licensing in **NVIDIA Control Panel**



3. In the **Primary License Server** field, enter the address of your primary NVIDIA vGPU software License Server.

   The address can be a fully-qualified domain name such as `gridlicense1.example.com`, or an IP address such as `10.31.20.45`.

   If you have only one license server configured, enter its address in this field.
4. Leave the **Port Number** field under the **Primary License Server** field unset.
   The port defaults to `7070`, which is the default port number used by NVIDIA vGPU software License Server.
5. In the **Secondary License Server** field, enter the address of your secondary NVIDIA vGPU software License Server.

   If you have only one license server configured, leave this field unset.

   The address can be a fully-qualified domain name such as `gridlicense2.example.com`, or an IP address such as `10.31.20.46`.
6. Leave the **Port Number** field under the **Secondary License Server** field unset.
   The port defaults to `7070`, which is the default port number used by NVIDIA vGPU software License Server.
7. Click **Apply** to assign the settings.

The system requests the appropriate license for the current vGPU from the configured license server.

The vGPU within the VM should now operate at full capability without any performance degradation over time for as long as the vGPU is licensed.

If the system fails to obtain a license, see Troubleshooting for guidance on troubleshooting.

3.1.2. Licensing an NVIDIA vGPU on Linux

1. As root, open the file `/etc/nvidia/gridd.conf` in a plain-text editor, such as `vi`.

```
[nvidia@localhost ~]$ sudo vi /etc/nvidia/gridd.conf
```

> **Note:** You can create the `/etc/nvidia/gridd.conf` file by copying the supplied template file `/etc/nvidia/gridd.conf.template`.

2. Set `ServerAddress` to the address of your primary NVIDIA vGPU software License Server.

   The address can be a fully-qualified domain name such as `gridlicense1.example.com`, or an IP address such as `10.31.20.45`.

   If you have only one license server configured, set `ServerAddress` to the address of the server.

3. **Optional:** Set `ServerPort` to the port number of your primary NVIDIA vGPU software License Server.

   If you do not set `ServerPort`, it will default to `7070`, which is the default port number that is used by the NVIDIA vGPU software License Server.

4. Set `BackupServerAddress` to the address of your secondary NVIDIA vGPU software License Server.

   The address can be a fully-qualified domain name such as `gridlicense2.example.com`, or an IP address such as `10.31.20.46`.

   If you have only one license server configured, do not set `BackupServerAddress`.

5. **Optional:** Set `BackupServerPort` to the port number of your secondary NVIDIA vGPU software License Server.

   If you do not set `BackupServerPort`, it will default to `7070`, which is the default port number that is used by the NVIDIA vGPU software License Server.

   If you have only one license server configured, do not set `BackupServerPort`.

6. Set `FeatureType` to 1, to license vGPU.

> **Note:** You do not need to specify the type of the license. NVIDIA vGPU software automatically selects the correct type of license based on the vGPU type.

7. Save your changes to the `/etc/nvidia/gridd.conf` file.
8. Restart the `nvidia-gridd` service:

```
[nvidia@localhost ~]$  sudo service nvidia-gridd restart
```

   The `nvidia-gridd` service requests the appropriate license for the current vGPU from the configured license server.

9. **Optional:** Confirm that the service has obtained a license by examining the log messages written to the system event log in the `/var/log` directory, for example, `/var/log/syslog` or `/var/log/messages`:

   ```
   [nvidia@localhost ~]$ sudo grep gridd /var/log/system-event-log
   ```

   For example:

   ```
   [nvidia@localhost ~]$ sudo grep gridd /var/log/messages
   …
   Aug 5 15:40:06 localhost nvidia-gridd: Started (4293)
   Aug 5 15:40:24 localhost nvidia-gridd: License acquired successfully.
   ```

The vGPU within the VM should now operate at full capability without any performance degradation over time for as long as the vGPU is licensed.

`gridd.conf` File for NVIDIA vGPU

The following example shows a `gridd.conf` file for NVIDIA vGPU in which options are set as follows:

- `ServerAddress` is set to `gridlicense1.example.com`.
- `BackupServerAddress` is set to `gridlicense2.example.com`.
- `ServerPort` and `BackupServerPort` are both set to `7070`.
- `FeatureType` is set to 1.

```
# /etc/nvidia/gridd.conf.template - Configuration file for NVIDIA Grid Daemon

# This is a template for the configuration file for NVIDIA Grid Daemon.
# For details on the file format, please refer to the nvidia-gridd(1)
# man page.

# Description: Set License Server Address
# Data type: string
# Format:  "<address>"
ServerAddress= gridlicense1.example.com

# Description: Set License Server port number
# Data type: integer
# Format:  <port>, default is 7070
ServerPort=7070

# Description: Set Backup License Server Address
# Data type: string
# Format:  "<address>"
BackupServerAddress= gridlicense2.example.com

# Description: Set Backup License Server port number
# Data type: integer
# Format:  <port>, default is 7070
BackupServerPort=7070

# Description: Set Feature to be enabled
# Data type: integer
# Possible values:
#    0 => for unlicensed state
#    1 => for GRID vGPU
#    2 => for Quadro Virtual Datacenter Workstation
#    4 => for NVIDIA Virtual Compute Server
FeatureType=1
```

```
# Description: Parameter to enable or disable Grid Licensing tab in nvidia-settings
# Data type: boolean
# Possible values: TRUE or FALSE, default is FALSE
#EnableUI=TRUE

# Description: Set license borrow period in minutes
# Data type: integer
# Possible values: 10 to 10080 mins(7 days), default is 1440 mins(1 day)
#LicenseInterval=1440

# Description: Set license linger period in minutes
# Data type: integer
# Possible values: 0 to 10080 mins(7 days), default is 0 mins
#LingerInterval=10
```

If the system fails to obtain a license, see Troubleshooting for guidance on troubleshooting.

3.2. Licensing a Physical GPU from the Legacy License Server

How to license a physical GPU depends on the guest OS that is running in the VM.

3.2.1. Licensing a Physical GPU for vWS on Windows

Perform this task from the guest VM to which the GPU is passed through or from the bare-metal host.

The **NVIDIA Control Panel** tool that you use to perform this task detects that a GPU is passed through to the VM or deployed to the bare-metal host and, therefore, provides the options for selecting between the vWS and vWS license types.

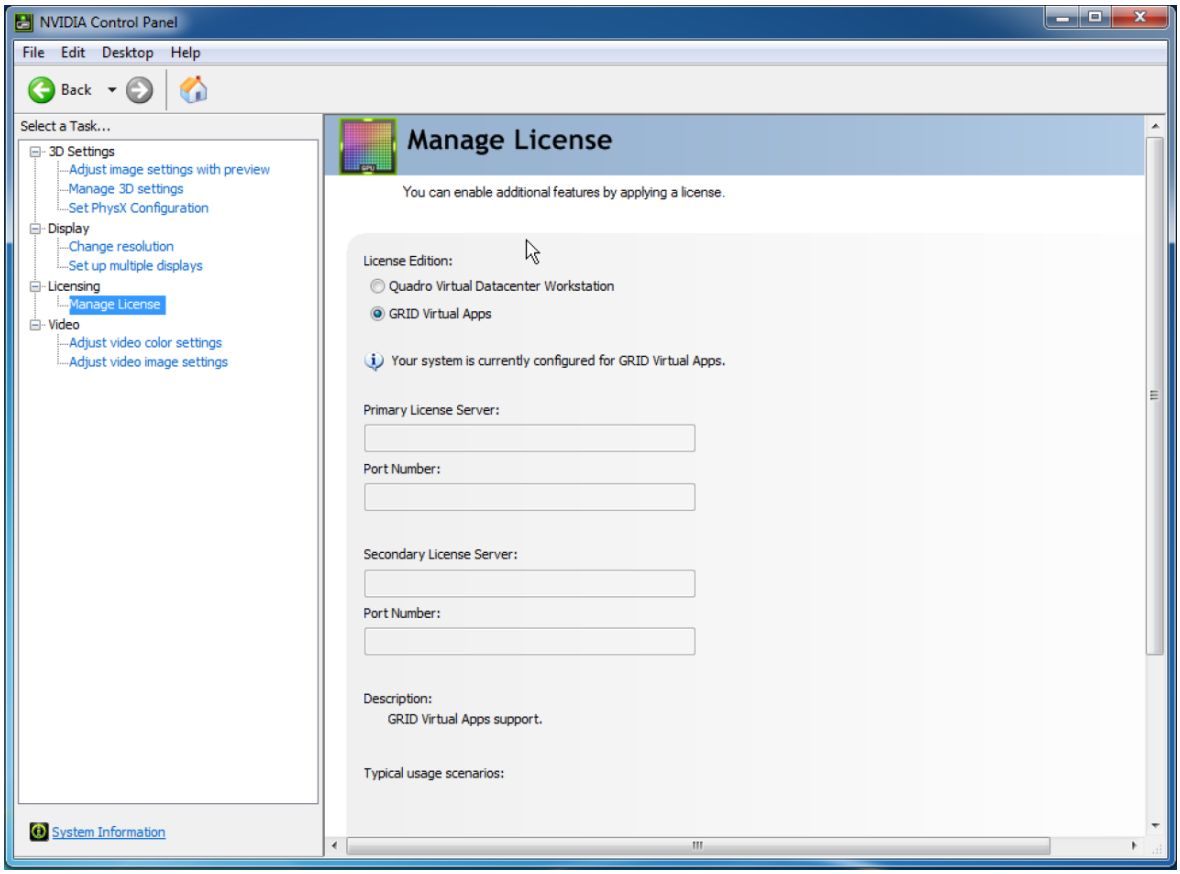
1. Open **NVIDIA Control Panel**:
   - Right-click on the Windows desktop and select **NVIDIA Control Panel** from the menu.
   - Open **Windows Control Panel** and double-click the **NVIDIA Control Panel** icon.
2. In **NVIDIA Control Panel**, select the **Manage License** task in the **Licensing** section of the navigation pane.

   Note: If the **Licensing** section and **Manage License** task are not displayed in **NVIDIA Control Panel**, the GPU or driver you are using do not support NVIDIA vGPU software licensed features.

   The **Manage License** task pane shows the current License Edition being used.

   The default is vApps.

Figure 3. Managing vWS Licensing in **NVIDIA Control Panel**



3. Select **vWS**.
4. In the **Primary License Server** field, enter the address of your primary NVIDIA vGPU software License Server.

   The address can be a fully-qualified domain name such as `gridlicense1.example.com`, or an IP address such as `10.31.20.45`.

   If you have only one license server configured, enter its address in this field.
5. Leave the **Port Number** field under the **Primary License Server** field unset.
   It will default to `7070`, which is the default port number used by NVIDIA vGPU software License Server.
6. In the **Secondary License Server** field, enter the address of your secondary NVIDIA vGPU software License Server.

   If you have only one license server configured, leave this field unset.

   The address can be a fully-qualified domain name such as `gridlicense2.example.com`, or an IP address such as `10.31.20.46`.
7. Leave the **Port Number** field under the **Secondary License Server** field unset.
   The port defaults to `7070`, which is the default port number used by NVIDIA vGPU software License Server.
8. Click **Apply** to assign the settings.

The **License Edition** section will indicate that your system is running in vWS mode. The system will also request the appropriate license for the current GPU from the configured license server. If unsuccessful, the system warns you that it could not obtain a license. If necessary, refer to Troubleshooting for guidance on troubleshooting.

Once configured in **NVIDIA Control Panel**, licensing settings persist across reboots and need only be modified if the license server address changes, or the VM is switched to running NVIDIA vGPU.

3.2.2. Removing a vWS License from a Physical GPU on Windows

1. Open **NVIDIA Control Panel**.
2. In the **Manage License** task, select **vApps**.
3. Click **Apply**.

The setting does not take effect until the next time the system is shut down or rebooted. vWS features remain available until then.

3.2.3. Licensing a Physical GPU on Linux

Licensing of a physical GPU in pass-through mode or a bare-metal deployment for vWS or vCS is enforced through software. To enable you to specify the correct NVIDIA vGPU software licensed product, options are provided for specifying vWS and vCS.

Note: Licensing of a physical GPU in pass-through mode or a bare-metal deployment for vApps is enforced through the EULA only. Therefore, no option is provided to specify the vApps license type.

1. As root, open the file `/etc/nvidia/gridd.conf` in a plain-text editor, such as `vi`.

```
[nvidia@localhost ~]$ sudo vi /etc/nvidia/gridd.conf
```

Note: You can create the `/etc/nvidia/gridd.conf` file by copying the supplied template file `/etc/nvidia/gridd.conf.template`.

2. Set `ServerAddress` to the address of your primary NVIDIA vGPU software License Server.

   The address can be a fully-qualified domain name such as `gridlicense1.example.com`, or an IP address such as `10.31.20.45`.

   If you have only one license server configured, set `ServerAddress` to the address of the server.
3. **Optional:** Set `ServerPort` to the port number of your primary NVIDIA vGPU software License Server.

   If you do not set `ServerPort`, it will default to `7070`, which is the default port number that is used by the NVIDIA vGPU software License Server.
4. Set `BackupServerAddress` to the address of your secondary NVIDIA vGPU software License Server.

The address can be a fully-qualified domain name such as `gridlicense2.example.com`, or an IP address such as `10.31.20.46`.

If you have only one license server configured, do not set `BackupServerAddress`.

5. **Optional:** Set `BackupServerPort` to the port number of your secondary NVIDIA vGPU software License Server.

   If you do not set `BackupServerPort`, it will default to `7070`, which is the default port number that is used by the NVIDIA vGPU software License Server.

   If you have only one license server configured, do not set `BackupServerPort`.

6. Set `FeatureType` to the integer that specifies the NVIDIA vGPU software licensed product that you want to run on the GPU.

NVIDIA vGPU Software Licensed Product	**FeatureType**
vWS	2
vCS	4

> **Note:** Ensure that you set `FeatureType` to the licensed product that you want, which might be different from the type of license that will be acquired from the license server. For example, the NVIDIA Virtual Compute Server product requires a vCS license, but can be used with a vWS license. If you want to use NVIDIA Virtual Compute Server, you must set `FeatureType` to 4 even if you have only vWS licenses.

7. Save your changes to the `/etc/nvidia/gridd.conf` file.
8. Restart the `nvidia-gridd` service:

```
[nvidia@localhost ~]$ sudo service nvidia-gridd restart
```

   The service should automatically obtain a license.

9. **Optional:** Confirm that the service has obtained a license by examining the log messages written to the system event log in the `/var/log` directory, for example, `/var/log/syslog` or `/var/log/messages`:

```
[nvidia@localhost ~]$ sudo grep gridd /var/log/system-event-log
```

   For example:

```
[nvidia@localhost ~]$ sudo grep gridd /var/log/messages
…
Aug 5 17:12:15 localhost nvidia-gridd: Started (10430)
Aug 5 17:12:34 localhost nvidia-gridd: License acquired successfully.
```

Once configured in `gridd.conf`, licensing settings persist across reboots and need only be modified if the license server address changes, or the VM is switched to running NVIDIA vGPU.

`gridd.conf` File for vWS

The following example shows a `gridd.conf` file for vWS in which options are set as follows:

- `ServerAddress` is set to `gridlicense1.example.com`.
- `BackupServerAddress` is set to `gridlicense2.example.com`.
- `ServerPort` and `BackupServerPort` are both set to `7070`.

- `FeatureType` is set to 2.

```
# /etc/nvidia/gridd.conf.template - Configuration file for NVIDIA Grid Daemon

# This is a template for the configuration file for NVIDIA Grid Daemon.
# For details on the file format, please refer to the nvidia-gridd(1)
# man page.

# Description: Set License Server Address
# Data type: string
# Format:  "<address>"
ServerAddress= gridlicense1.example.com

# Description: Set License Server port number
# Data type: integer
# Format:  <port>, default is 7070
ServerPort=7070

# Description: Set Backup License Server Address
# Data type: string
# Format:  "<address>"
BackupServerAddress= gridlicense2.example.com

# Description: Set Backup License Server port number
# Data type: integer
# Format:  <port>, default is 7070
BackupServerPort=7070

# Description: Set Feature to be enabled
# Data type: integer
# Possible values:
#    0 => for unlicensed state
#    1 => for GRID vGPU
#    2 => for Quadro Virtual Datacenter Workstation
#    4 => for NVIDIA Virtual Compute Server
FeatureType=2

# Description: Parameter to enable or disable Grid Licensing tab in nvidia-settings
# Data type: boolean
# Possible values: TRUE or FALSE, default is FALSE
#EnableUI=TRUE

# Description: Set license borrow period in minutes
# Data type: integer
# Possible values: 10 to 10080 mins(7 days), default is 1440 mins(1 day)
#LicenseInterval=1440

# Description: Set license linger period in minutes
# Data type: integer
# Possible values: 0 to 10080 mins(7 days), default is 0 mins
#LingerInterval=10
```

`gridd.conf` File for vCS

The following example shows a `gridd.conf` file for vCS in which options are set as follows:

- `ServerAddress` is set to `gridlicense1.example.com`.
- `BackupServerAddress` is set to `gridlicense2.example.com`.
- `ServerPort` and `BackupServerPort` are both set to `7070`.
- `FeatureType` is set to 4.

```
# /etc/nvidia/gridd.conf.template - Configuration file for NVIDIA Grid Daemon
```

```
# This is a template for the configuration file for NVIDIA Grid Daemon.
# For details on the file format, please refer to the nvidia-gridd(1)
# man page.

# Description: Set License Server Address
# Data type: string
# Format:  "<address>"
ServerAddress= gridlicense1.example.com

# Description: Set License Server port number
# Data type: integer
# Format:  <port>, default is 7070
ServerPort=7070

# Description: Set Backup License Server Address
# Data type: string
# Format:  "<address>"
BackupServerAddress= gridlicense2.example.com

# Description: Set Backup License Server port number
# Data type: integer
# Format:  <port>, default is 7070
BackupServerPort=7070

# Description: Set Feature to be enabled
# Data type: integer
# Possible values:
#    0 => for unlicensed state
#    1 => for GRID vGPU
#    2 => for Quadro Virtual Datacenter Workstation
#    4 => for NVIDIA Virtual Compute Server
FeatureType=4

# Description: Parameter to enable or disable Grid Licensing tab in nvidia-settings
# Data type: boolean
# Possible values: TRUE or FALSE, default is FALSE
#EnableUI=TRUE

# Description: Set license borrow period in minutes
# Data type: integer
# Possible values: 10 to 10080 mins(7 days), default is 1440 mins(1 day)
#LicenseInterval=1440

# Description: Set license linger period in minutes
# Data type: integer
# Possible values: 0 to 10080 mins(7 days), default is 0 mins
#LingerInterval=10
```

3.3. Licensing NVIDIA vGPU Software on Linux by Using NVIDIA X Server Settings

By default, the **Manage License** option is not available in **NVIDIA X Server Settings** and you must enable this option if you want to manage licensing by using **NVIDIA X Server Settings**. However, this option must be disabled with Red Hat Enterprise Linux 6.8 and 6.9 or CentOS 6.8 and 6.9.

3.3.1. Enabling License Management in NVIDIA X Server Settings

By default, the **Manage License** option is not available in **NVIDIA X Server Settings**. Before using **NVIDIA X Server Settings** to license NVIDIA vGPU software, you must enable this option.

Note: Do **not** enable the **Manage License** option with Red Hat Enterprise Linux 6.8 and 6.9 or CentOS 6.8 and 6.9. To prevent a segmentation fault in DBus code from causing the `nvidia-gridd` service from exiting, the GUI for licensing must be disabled with these OS versions.

If you are licensing a physical GPU for vCS, you **must** use the configuration file `/etc/nvidia/gridd.conf`.

This task requires `sudo` privileges.

1. If **NVIDIA X Server Settings** is running, shut it down.
2. If the `/etc/nvidia/gridd.conf` file does not already exist, create it by copying the supplied template file `/etc/nvidia/gridd.conf.template`.
3. As root, edit the `/etc/nvidia/gridd.conf` file to set the `EnableUI` option to `TRUE`.
4. Start the `nvidia-gridd` service.

```
# sudo service nvidia-gridd start
```

When **NVIDIA X Server Settings** is restarted, the **Manage License** option is now available.

3.3.2. Licensing an NVIDIA vGPU on Linux by Using NVIDIA X Server Settings

Perform this task from the guest VM to which the vGPU is assigned.

The **NVIDIA X Server Settings** tool that you use to perform this task detects that a vGPU is assigned to the VM and, therefore, provides no options for selecting the license type. After you license the vGPU, NVIDIA vGPU software automatically selects the correct type of license based on the vGPU type.

Ensure that the **Manage License** option is enabled as explained in Enabling License Management in NVIDIA X Server Settings.

Note: Do **not** enable the **Manage License** option with Red Hat Enterprise Linux 6.8 and 6.9 or CentOS 6.8 and 6.9. To prevent a segmentation fault in DBus code from causing the `nvidia-gridd` service from exiting, the GUI for licensing must be disabled with these OS versions.

If you do not want to or cannot enable the **Manage License** option, use the the configuration file `/etc/nvidia/gridd.conf` as explained in Licensing an NVIDIA vGPU on Linux.

1. Start **NVIDIA X Server Settings** by using the method for launching applications provided by your Linux distribution.

   For example, on Ubuntu Desktop, open the **Dash**, search for **`NVIDIA X Server Settings`**, and click the **NVIDIA X Server Settings** icon.
2. In the **NVIDIA X Server Settings** window that opens, click **Manage GRID License**.

   The **License Edition** section of the **NVIDIA X Server Settings** window shows that NVIDIA vGPU is currently unlicensed.
3. In the **Primary Server** field, enter the address of your primary NVIDIA vGPU software License Server.

   The address can be a fully-qualified domain name such as `gridlicense1.example.com`, or an IP address such as `10.31.20.45`.

   If you have only one license server configured, enter its address in this field.
4. Leave the **Port Number** field under the **Primary Server** field unset.
   The port defaults to `7070`, which is the default port number used by NVIDIA vGPU software License Server.
5. In the **Secondary Server** field, enter the address of your secondary NVIDIA vGPU software License Server.

   If you have only one license server configured, leave this field unset.

   The address can be a fully-qualified domain name such as `gridlicense2.example.com`, or an IP address such as `10.31.20.46`.
6. Leave the **Port Number** field under the **Secondary Server** field unset.
   The port defaults to `7070`, which is the default port number used by NVIDIA vGPU software License Server.
7. Click **Apply** to assign the settings.
   The system requests the appropriate license for the current vGPU from the configured license server.

The vGPU within the VM should now operate at full capability without any performance degradation over time for as long as the vGPU is licensed.

If the system fails to obtain a license, refer to Troubleshooting for guidance on troubleshooting.

3.3.3. Licensing a Physical GPU for vWS or vCS on Linux by Using **NVIDIA X Server Settings**

Perform this task from the guest VM to which the GPU is passed through or from the bare-metal host.

The **NVIDIA X Server Settings** tool that you use to perform this task detects that the GPU is passed through to the VM or deployed to the bare-metal host and, therefore, provides options for selecting between the following NVIDIA vGPU software licensed products:

- vWS
- NVIDIA Virtual Compute Server

- vApps

Ensure that the **Manage License** option is enabled as explained in Enabling License Management in NVIDIA X Server Settings.

Note: Do **not** enable the **Manage License** option with Red Hat Enterprise Linux 6.8 and 6.9 or CentOS 6.8 and 6.9. To prevent a segmentation fault in DBus code from causing the `nvidia-gridd` service from exiting, the GUI for licensing must be disabled with these OS versions.

If you do not want to or cannot enable the **Manage License** option, use the configuration file `/etc/nvidia/gridd.conf`.

If you are licensing a physical GPU for vCS, you **must** use the configuration file `/etc/nvidia/gridd.conf`.

For instructions, refer to Licensing a Physical GPU on Linux.

1. Start **NVIDIA X Server Settings** by using the method for launching applications provided by your Linux distribution.

   For example, on Ubuntu Desktop, open the **Dash**, search for **`NVIDIA X Server Settings`**, and click the **NVIDIA X Server Settings** icon.
2. In the **NVIDIA X Server Settings** window that opens, click **Manage GRID License**.

   The **License Edition** section of the **NVIDIA X Server Settings** window shows the current License Edition being used.

   The default is vApps.
3. Select the NVIDIA vGPU software licensed product that you want.
   - **vWS**
   - **NVIDIA Virtual Compute Server**

   Note: Ensure that you select the licensed product that you want, which might be different from the type of license that will be acquired from the license server. For example, the NVIDIA Virtual Compute Server product requires a vCS license, but can be used with a vWS license. If you want to use NVIDIA Virtual Compute Server, you must select **NVIDIA Virtual Compute Server** even if you have only vWS licenses.
4. In the **Primary Server** field, enter the address of your primary NVIDIA vGPU software License Server.

   The address can be a fully-qualified domain name such as `gridlicense1.example.com`, or an IP address such as `10.31.20.45`.

   If you have only one license server configured, enter its address in this field.
5. Leave the **Port Number** field under the **Primary Server** field unset.
   The port defaults to `7070`, which is the default port number used by NVIDIA vGPU software License Server.
6. In the **Secondary Server** field, enter the address of your secondary NVIDIA vGPU software License Server.

   If you have only one license server configured, leave this field unset.

The address can be a fully-qualified domain name such as `gridlicense2.example.com`, or an IP address such as `10.31.20.46`.

7. Leave the **Port Number** field under the **Secondary Server** field unset.
   The port defaults to `7070`, which is the default port number used by NVIDIA vGPU software License Server.
8. Click **Apply** to assign the settings.

   The **License Edition** section indicates that your system is running in one of the following modes, depending on the license type that you selected:

   - vWS
   - NVIDIA Virtual Compute Server

   The system also requests the appropriate license for the current GPU from the configured license server. If unsuccessful, the system warns you that it could not obtain a license. If necessary, refer to Troubleshooting for guidance on troubleshooting.

Once configured in **NVIDIA X Server Settings**, licensing settings persist across reboots and need only be modified if the license server address changes, or the VM is switched to running NVIDIA vGPU.

Chapter 4. Advanced Topics

This chapter discusses advanced topics and settings for NVIDIA vGPU software licensing.

4.1. Setting the Client Host Identifier for License Checkouts

If a VM is shut down abruptly and then restarted after a change to its network configuration, two licenses might be checked out for the same VM. You can avoid this situation by setting the client host identifier for license checkouts.

By default, the license server selects the first valid MAC address it finds to identify the VM for license checkouts. If a VM that has a license checked out is shut down abruptly, it might not release the license back to the license server. If the network configuration of the VM is changed after the shutdown and the VM is restarted, the license server might use a different MAC address to identify the VM, which will cause two licenses to be checked out for the same VM. Setting the client host identifier avoids this situation because the license server always uses identifier that you set to identify the VM.

1. Set `LicenseHostId` to the MAC address of a network interface controller (NIC) on the VM.
   - On Windows, add the following Windows registry key setting:
     ```
     [HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global
     \GridLicensing]"LicenseHostId"=vm-nic-mac-address
     ```
   - On Linux, add the following entry to `/etc/nvidia/gridd.conf`:
     ```
     LicenseHostId==vm-nic-mac-address
     ```

   vm-nic-mac-address
   The MAC address of the network interface controller (NIC) on the VM that the license server will use to identify the VM for license checkouts. Specify **only** the numbers and letters in the address **without** any spaces or punctuation, for example, **`00005E0053FF`**.
2. Apply the changed setting to the VM.
   - If the vGPU or physical GPU assigned to the VM has already been licensed, restart the VM.
   - Otherwise, license the vGPU or physical GPU assigned to the VM as explained in the following topics:
     - Configuring a Licensed Client of NVIDIA License System

- Licensing NVIDIA vGPU Software from the Legacy License Server

4.2. Licenses Obtained After Boot

Under normal operation, an NVIDIA vGPU software license is obtained by a platform during boot, before user login and launch of applications. If a license is not available, the system will periodically retry its license request to the license server. During this time, the vGPU or GPU initially operates at full capability but its performance is degraded over time if the system fails to obtain a license.

The performance of an unlicensed vGPU or GPU is degraded as described in Software Enforcement of NVIDIA vGPU Software Licensing. Furthermore, vWS features described in Display Resolutions for Physical GPUs and vCS features are not available to applications started while the performance of a physical GPU is degraded.

On Windows, unavailability of a license is indicated by a pop-up window or by the license status on the **Licensing** tab of the **NVIDIA Control Panel**. On Linux, unavailability of a license is indicated by log messages.

If the performance of a vGPU or GPU has been degraded, the full capability of the vGPU or GPU is restored when a license is acquired. However, application software launched while the performance was degraded might need to be restarted to recognize and use the licensed features.

4.3. Operating with Intermittent Connectivity to the License Server

To check out a license, vWS, vPC, and vApps clients require connectivity to a license server when booting.

After they have booted, clients may operate without connectivity to the license server for a period of up to 1 day, after which time the client will be warned of license expiration.

4.4. Applying Windows License Settings Through the Windows Registry

NVIDIA vGPU software licensing settings can be controlled via the Windows Registry, removing the need for manual interaction with **NVIDIA Control Panel**. Settings are stored in this registry key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\GridLicensing
```

Registry values are summarized in Table 3.

Table 3. Licensing Registry Settings

Name	Type	Description
Legacy license server only: `ServerAddress`	String (`REG_SZ`)	License server address
Legacy license server only: `BackupServerAddress`	String (`REG_SZ`)	Secondary license server address
Legacy license server only: `ServerPort`	String (`REG_SZ`)	License server port number (Optional: defaults to 7070)
Legacy license server only: `BackupServerPort`	String (`REG_SZ`)	Secondary license server port number (Optional: defaults to 7070)
`FeatureType`	DWord (`REG_DWORD`)	The feature type of a GPU in pass-through mode or a bare-metal deployment: 0: vApps 2: vWS All other values reserved. Do **not** change the value of this registry key in a VM configured with NVIDIA vGPU.
`NvCplDisableManageLicensePage`	DWord (`REG_DWORD`)	If this registry key is absent, licensing controls are shown in NVIDIA Control Panel for GPUs supporting licensing. When present: 0: Show licensing controls in NVIDIA Control Panel 1: Hide licensing controls in NVIDIA Control Panel All other values reserved.
`LicenseInterval`	DWord (`REG_DWORD`)	An integer in the range 10-10080 that specifies the period of time in minutes for which a license can be borrowed after it is checked out After this period has elapsed, the client must obtain a new license from the server. The default is 1440 minutes, which corresponds to a period of 1 day.

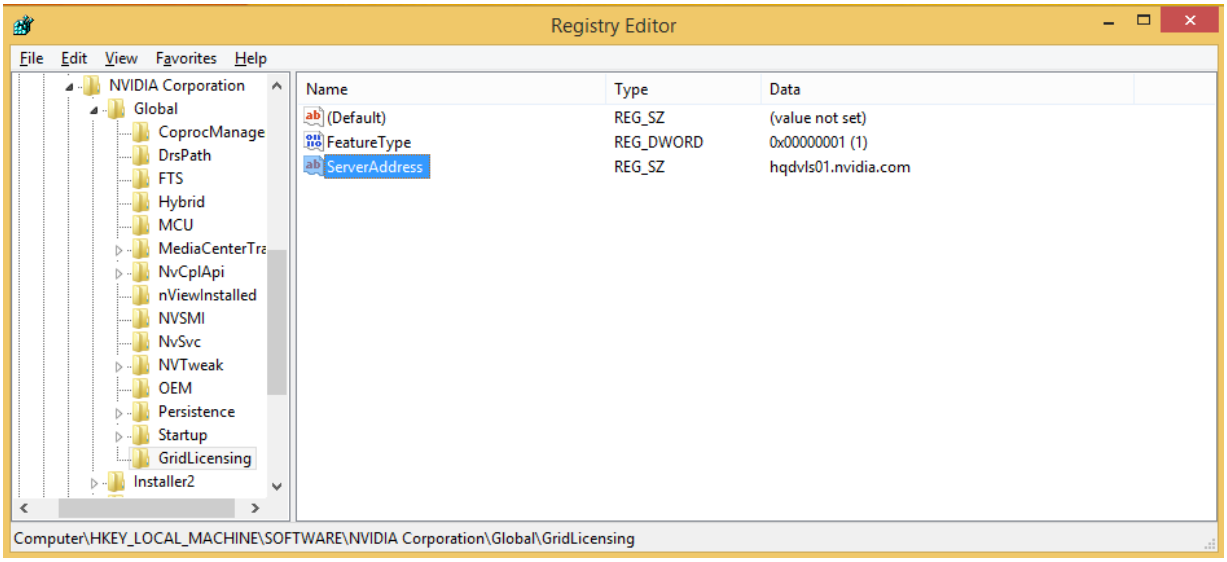
Name	Type	Description
`LingerInterval`	DWord (`REG_DWORD`)	An integer in the range 0–10080 that specifies the period of time in minutes for which a license will remain checked out against a client VM after the VM is shut down. During this period, a VM that is rebooting can reclaim the same license after the reboot is complete. After this period has elapsed, the license is freed and available for use by other clients. The default is 0 minutes, which instantly frees licenses from a VM that is shut down cleanly. If you oversubscribe your licenses and want to protect against accidental loss of a license during a reboot, increase the `LingerInterval` to a value that reflects the amount of time required for a reboot.
`DisableExpirationPopups`	DWord (`REG_DWORD`)	Disables or enables pop-up notifications for NVIDIA vGPU software license expiration. 0: Enable license expiration pop-up notifications 1: Disable license expiration pop-up notifications (Optional: defaults to 0)
`DisableSpecificPopups`	DWord (`REG_DWORD`)	If this registry key is absent, pop-up notifications are enabled for changes in NVIDIA vGPU software license state to the following states: ▶ Unlicensed ▶ Requesting license ▶ License acquired ▶ License request failed ▶ License renewal failed ▶ License expired (if `DisableExpirationPopups` is 0)

Name	Type	Description
		When present: 0: Disable none of the pop-up notifications for license state changes 1: Disable pop-up notifications for changes to the following license states: ▶ Requesting license ▶ License acquired ▶ License renewal failed (Optional: defaults to 0)
`EnableLogging`	DWord (`REG_DWORD`)	If this registry key is absent, significant licensing events are logged. When present: 0: Disable logging of significant licensing events 1: Enable logging of significant licensing events (Optional: defaults to 1)
`LicenseHostId`	String (`REG_SZ`)	If this registry key is absent, the license server selects the first valid MAC address it finds to identify the VM for license checkouts. When present, this registry key must be the MAC address of the network interface controller (NIC) on the VM that the license server will use to identify the VM for license checkouts. Specify **only** the numbers and letters in the address **without** any spaces or punctuation, for example, **`00005E0053FF`**. Setting this registry key prevents two license checkouts by a single VM if the VM is shut down abruptly and then restarted after a change to its network configuration.
NVIDIA License System only: `ClientConfigTokenPath`	String (`REG_SZ`)	The full path to the folder in which you want to store the client configuration token for the client.

Name	Type	Description
		Set this registry key **only** if you want store the client configuration token in a custom location.
Since 14.3, NVIDIA License System only: `ProxyServerAddress`	String `(REG_SZ)`	Proxy server address. Set this registry key **only** if a non-transparent proxy server is configured between your licensed client and a CLS instance.
Since 14.3, NVIDIA License System only: `ProxyServerPort`	String `(REG_SZ)`	Proxy server port number. Set this registry key **only** if a non-transparent proxy server is configured between your licensed client and a CLS instance.

Figure 4 shows an example of configuring virtual GPU licensing settings in the registry. Note it is sufficient to simply set the license server address in `ServerAddress`. Leave `FeatureType` at its preset value.

Figure 4. Configuring vGPU Licensing Through Windows Registry Settings



Chapter 5. Troubleshooting

This chapter describes basic troubleshooting steps.

5.1. Known Issues

Before troubleshooting or filing a bug report, review the release notes that accompany each driver release, for information about known issues with the current release, and potential workarounds.

5.2. Verifying the NVIDIA vGPU Software License Status of a Licensed Client

After configuring a client with an NVIDIA vGPU software license, verify the license status by displaying the licensed product name and status.

To verify the license status of a licensed client, run `nvidia-smi` with the `-q` or `--query` option. If the product is licensed, the expiration date is shown in the license status.

```
nvidia-smi -q
==============NVSMI LOG==============

Timestamp                           : Wed Mar 31 01:49:28 2020
Driver Version                      : 440.88
CUDA Version                        : 10.0

Attached GPUs                       : 1
GPU 00000000:00:08.0
    Product Name                    : Tesla T4
    Product Brand                   : Grid
    Display Mode                    : Enabled
    Display Active                  : Disabled
    Persistence Mode                : N/A
    Accounting Mode                 : Disabled
    Accounting Mode Buffer Size     : 4000
    Driver Model
        Current                     : WDDM
        Pending                     : WDDM
    Serial Number                   : 0334018000638
    GPU UUID                        : GPU-ba2310b6-95d1-802b-f96f-5865410fe517
    Minor Number                    : N/A
    VBIOS Version                   : 90.04.21.00.01
    MultiGPU Board                  : No
    Board ID                        : 0x8
```

```
    GPU Part Number                 : 699-2G183-0200-100
    Inforom Version
        Image Version               : G183.0200.00.02
        OEM Object                  : 1.1
        ECC Object                  : 5.0
        Power Management Object     : N/A
    GPU Operation Mode
        Current                     : N/A
        Pending                     : N/A
    GPU Virtualization Mode
        Virtualization mode         : Pass-Through
    vGPU Software Licensed Product
        Product Name                : NVIDIA Virtual Compute Server
        License Status              : Licensed (Expiry: 2021-11-13 18:29:59 GMT)
    …
    …
```

5.3. Licensing Event Logs

Significant licensing events, for example, acquisition of a license, return of a license, expiration of a license, or failure to acquire a license, are logged to help troubleshoot issues. Error events are logged with an error code to help you to determine the underlying cause of the error. License acquisition events are logged with the name and version of the licensed product, for example, `GRID-Virtual-WS,2.0`.

Windows Licensing Event Logs

On Windows, licensing events are logged in the plain-text file `%SystemDrive%\Users\Public\Documents\NvidiaLogging\Log.NVDisplay.Container.exe.log`.

The log file is rotated when its size reaches 16 MB. A new log file is created and the old log file is renamed to `Log.NVDisplay.Container.exe.log1`. Each time the log file is rotated, the number in the file name of each existing old log file is increased by 1. The oldest log file is deleted when the number of log files exceeds 16.

Logging is enabled by default. To disable logging, create the Windows Registry key `EnableLogging (DWORD)` with the value zero at the path `HKLM\SOFTWARE\NVIDIA Corporation\Global\GridLicensing`.

Linux Licensing Event Logs

On Linux, logging is enabled by default. The licensing events are logged in the standard activity log in the `/var/log` directory. The name of the log file in this directory depends on the Linux distribution, for example:

- For Red Hat Enterprise Linux and CentOS distributions, the file name is `messages`.
- For Ubuntu distributions, the file name is `syslog`.

You can use the standard features of the operating system to rotate and purge these logs.

Log Messages for Acquisition and Return of a License and Transient Network Failure

This example shows log messages for the following sequence of events:

1. Acquisition of a license when a VM is started
2. Return of a license when the VM is shut down
3. Failure to communicate with the license server when the VM is started when the license server is unreachable
4. Acquisition of a license after communications with the license server are restored

```
Mon Oct 22 20:53:29 2018:<1>:License acquired successfully. (Info: http://
gridlicense1.example.com:7070/request; GRID-Virtual-WS,2.0)
Mon Oct 22 20:54:22 2018:<1>:End Logging
Mon Oct 22 20:54:24 2018:<0>:License returned successfully
Mon Oct 22 20:54:24 2018:<0>:End Logging
Mon Oct 22 20:55:36 2018:<1>:Failed server communication. Please ensure your system
 has network connectivity to the license server
Mon Oct 22 20:55:49 2018:<1>:Failed server communication. Please ensure your system
 has network connectivity to the license server
Mon Oct 22 20:56:01 2018:<1>:Failed server communication. Please ensure your system
 has network connectivity to the license server
Mon Oct 22 20:56:19 2018:<1>:Failed server communication. Please ensure your system
 has network connectivity to the license server
Mon Oct 22 20:56:31 2018:<1>:Failed server communication. Please ensure your system
 has network connectivity to the license server
Mon Oct 22 20:56:44 2018:<1>:Failed server communication. Please ensure your system
 has network connectivity to the license server
Mon Oct 22 20:56:47 2018:<1>:License acquired successfully. (Info: http://
gridlicense1.example.com:7070/request; GRID-Virtual-WS,2.0)
Mon Oct 22 21:36:11 2018:<1>:End Logging
Mon Oct 22 21:36:12 2018:<0>:License returned successfully
Mon Oct 22 21:36:12 2018:<0>:End Logging
```

Log Messages for Failure to Acquire a License

This example shows log messages for the failure to acquire a license because no licenses of the requested type are available.

```
Mon Oct 22 22:26:36 2018:<1>:Error : Server URL : http://
gridlicense1.example.com:7070/request -
[1,7E2,2,0[70000030,0,702C3]]
Insufficient count for the requested feature.
Mon Oct 22 22:28:15 2018:<1>:Error : Server URL : http://
gridlicense1.example.com:7070/request -
[1,7E2,2,0[70000030,0,702C3]]
Insufficient count for the requested feature.
Mon Oct 22 22:29:55 2018:<1>:Error : Server URL : http://
gridlicense1.example.com:7070/request -
[1,7E2,2,0[70000030,0,702C3]]
Insufficient count for the requested feature.
Mon Oct 22 22:31:34 2018:<1>:Error : Server URL : http://
gridlicense1.example.com:7070/request -
[1,7E2,2,0[70000030,0,702C3]]
Insufficient count for the requested feature.
```

5.4. Troubleshooting Steps

If an NVIDIA vGPU software system fails to obtain a license, troubleshoot the failure as follows:

- Examine the log of significant licensing events for information about the cause of the failure. For details, see Licensing Event Logs.
- Check that the license server address and port number are correctly configured.
- Run a network ping test from the NVIDIA vGPU software system to the license server address to verify that the system has network connectivity to the license server.
- Verify that the date and time are configured correctly on the NVIDIA vGPU software system. If the time is set inaccurately or is adjusted backwards by a large amount, the system may fail to obtain a license.
- Verify that the license server in use has available licenses of the type required by the NVIDIA vGPU software feature the NVIDIA vGPU software system is configured to use.

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, the NVIDIA logo, NVIDIA GRID, NVIDIA GRID vGPU, NVIDIA Maxwell, NVIDIA Pascal, NVIDIA Turing, NVIDIA Volta, GPUDirect, Quadro, and Tesla 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

© 2013-2022 NVIDIA Corporation & affiliates. All rights reserved.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com