

NVIDIA Virtual GPU Software Client Licensing User Guide

Release 21.0

NVIDIA Corporation

Nov 05, 2026

Contents

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 . . . 3
1.2.1.1 Additional Considerations for NVIDIA vGPU Deployments . . . 4
1.2.1.2 Additional Considerations for Physical GPU Deployments . . . 4
1.2.2 EULA-Only Enforcement of NVIDIA vGPU Software Licensing . . . 5
1.3 Display Resolutions for Physical GPUs . . . 5
1.3.1 NVIDIA RTX Virtual Workstation Physical GPU Resolutions . . . 5
1.3.2 NVIDIA Virtual Applications or NVIDIA vGPU for Compute Physical GPU Resolutions 6

2 Configuring a Licensed Client of NVIDIA License System 7
2.1 Configuring a Licensed Client on Windows with Default Settings . . . 8
2.2 Configuring a Licensed Client on Linux with Default Settings . . . 8

3 Advanced Topics 11
3.1 Configuring a Licensed Client with Custom Settings . . . 11
3.1.1 Configuring a Licensed Client on Windows with Custom Settings . . . 11
3.1.2 Configuring a Licensed Client on Linux with Custom Settings . . . 13
3.1.3 Enabling License Management in NVIDIA X Server Settings . . . 17
3.2 Configuring a Proxy Server Between a Licensed Client and a Service Instance . . . 17
3.2.1 Proxy Server Requirements and Firewall Rules for a Service Instance . . . 17
3.2.1.1 Proxy Server Requirements for a Service Instance . . . 17
3.2.1.2 Non-Transparent Proxy Server Support . . . 18
3.2.1.3 Firewall Rules for a Service Instance . . . 18
3.2.2 Generating an Encrypted Credentials File . . . 18
3.2.2.1 Generating an Encrypted Credentials File on Windows . . . 19
3.2.2.2 Generating an Encrypted Credentials File on Linux . . . 20
3.3 Setting the Client Host Identifier for License Checkouts . . . 20
3.4 Licenses Obtained After Boot or User Login . . . 21
3.5 Operating with Intermittent Connectivity to the License Server . . . 21
3.6 Windows Registry Settings for NVIDIA vGPU Software Licensing . . . 21
3.6.1 Registry Key Values for Licensing Settings . . . 22
3.6.2 Registry Key Value for Managing Licensing Controls in NVIDIA Control Panel . . . 24

4 Troubleshooting 27
4.1 Known Issues . . . 27
4.2 Verifying the NVIDIA vGPU Software License Status of a Licensed Client . . . 27
4.3 Licensing Event Logs . . . 28
4.3.1 Windows Licensing Event Logs . . . 29
4.3.2 Linux Licensing Event Logs . . . 29
4.3.3 Log Messages for Acquisition and Return of a License and Transient Network Failure 29
4.3.4 Log Messages for Failure to Acquire a License . . . 30

4.4 Troubleshooting Steps 30
4.5 Maximum Allowed Network Adapters on a VM Exceeded 31
4.5.1 Windows: Recovering from a Failure Caused by Too Many Network Adapters 31
4.5.2 Linux: Recovering from a Failure Caused by Too Many Network Adapters 31

5 Notices **33**

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 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 1.1: 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 applications for Windows, web browsers, and high-definition video	▶ B-series NVIDIA vGPUs ▶ Microsoft RemoteFX vGPU
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 ▶ 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 is a graphics virtualization technology.
- 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 1.2: NVIDIA vGPU Software License Enforcement by Deployment Type

NVIDIA vGPU Software Deployment	Required NVIDIA vGPU Software License	Enforcement
A-series NVIDIA vGPU	vApps	Software[1]
B-series NVIDIA vGPU	vPC or vWS	Software[2]
Q-series NVIDIA vGPU	vWS	Software[3]
GPU pass through for workstation or professional 3D graphics	vWS	Software
GPU pass through for PC-level applications	vApps	EULA only
Microsoft DDA for workstation or professional 3D graphics	vWS	Software
Microsoft DDA for PC-level applications	vApps	EULA only
VMware vDGA for workstation or professional 3D graphics	vWS	Software
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

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 if the VM fails to obtain a license within 20 minutes and is further degraded if the VM still hasn't obtained a license after 24 hours.

If users report that the performance of a virtual GPU or physical GPU is degraded, determine whether the associated VM has obtained a license as explained in *Troubleshooting*.

A VM obtains a license over the network from a service instance of the NVIDIA License System. When a VM obtains a license, the license is "checked out" or "borrowed." The VM releases the license back to the service instance when the license is no longer required. You can control when a license is checked out and released back to the service instance through the setting of a Windows registry key value or Linux configuration parameter.

[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] 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.

- By default, the license is checked out when the VM is booted and released when the VM is shut down.
- If the Windows registry key value or Linux configuration parameter is set, the license is checked out when a user logs in to the VM and released when the user logs out of the VM.

For information about the Windows registry key value for controlling this behavior, refer to *Windows Registry Settings for NVIDIA vGPU Software Licensing*.

Note

For information about setting up and running NVIDIA License System, refer to NVIDIA License System User Guide.

Figure 1.1 illustrates how NVIDIA vGPU software licensing is enforced through software.

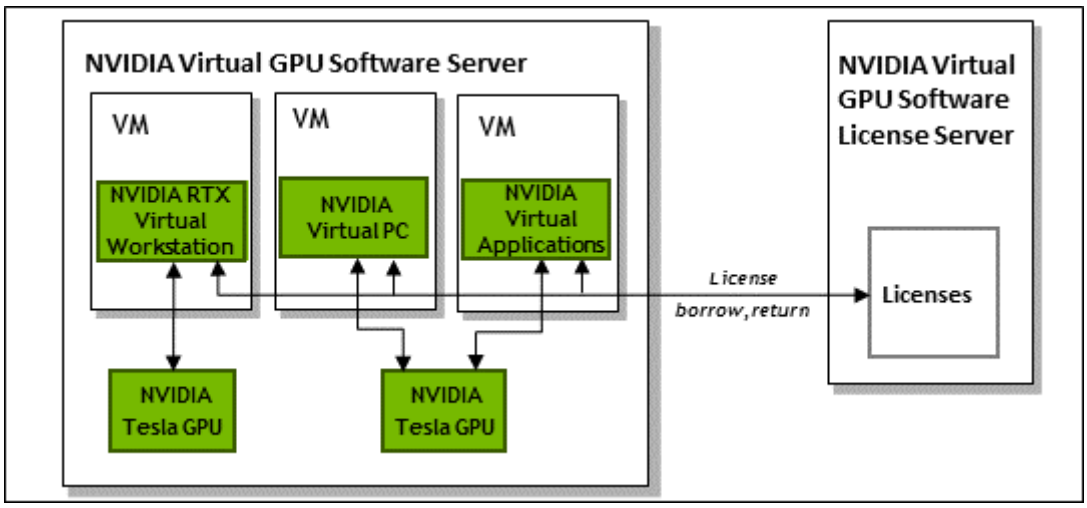


Figure 1.1: Software Enforcement of NVIDIA vGPU Software Licensing

1.2.1.1 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.

1.2.1.2 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.

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 vGPU software license that is applied to the GPU.

1.3.1. NVIDIA RTX Virtual Workstation Physical GPU Resolutions

GPUs that are licensed with an NVIDIA RTX Virtual Workstation license support a maximum combined resolution based on the number of available pixels. 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.

Available Pixels	Display Resolution	Displays per GPU
66355200	7680×4320	2
	5120×2880 or lower	4

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

Available Pixels	Available Pixel Basis	Maximum Displays	Sample Mixed Display Configurations
66355200	2 7680×4320 displays	4	1 7680×4320 display plus 2 5120×2880 displays
			1 7680×4320 display plus 3 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 even though the combined resolution of the displays (44236800) is less than the number of available pixels from the GPU (66355200).

1.3.2. NVIDIA Virtual Applications or NVIDIA vGPU for Compute Physical GPU Resolutions

GPUs that are licensed with an NVIDIA Virtual Applications or an NVIDIA vGPU for Compute license support a single display with a fixed maximum resolution. The maximum resolution depends on the following factors:

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

License	Operating System	Maximum Display Resolution	Displays per GPU
NVIDIA Virtual Applications	Linux	2560×1600	1
	Windows	1280×1024	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 no longer requires the license.

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

- If you are using a Delegated License Service (DLS) instance to serve licenses, ensure that you are using at minimum release 3.4 of the DLS virtual appliance. Earlier releases of the DLS virtual appliance are **incompatible** with this release of NVIDIA vGPU software.

  For more information about the prerequisites for using this release of NVIDIA vGPU software, refer to NVIDIA License System User Guide. These requirements are met automatically for Cloud License Service (CLS) instances, which are hosted on the NVIDIA Licensing Portal.
- 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 Generating a Client Configuration Token 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 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 NVIDIA vGPU software graphics driver creates a default location in which to store the client configuration token on the client. If you want to use this location for the client configuration token and, on Windows, are configuring the client with NVIDIA vGPU, you can configure the client with default settings. Otherwise, you must configure the client with custom settings as explained in *Configuring a Licensed Client with Custom Settings*.

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 with Default Settings

Perform this task from the client.

1. Copy the client configuration token to the `%SystemDrive%\Program Files\NVIDIA Corporation\vGPU Licensing\ClientConfigToken` folder.
2. Restart the `NvDisplayContainer` service.

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

2.2. Configuring a Licensed Client on Linux with Default Settings

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

> **Note**
>
> You can also perform this step from NVIDIA X Server Settings. Before using NVIDIA X Server Settings to perform this step, ensure that this option has been enabled as explained in *Enabling License Management in NVIDIA X Server Settings*.

This example shows how to configure a licensed Linux client for NVIDIA RTX Virtual Workstation.

```
# /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 vGPU for Compute
FeatureType=2
...
```

3. Copy the client configuration token to the `/etc/nvidia/ClientConfigToken` directory.
4. 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.
    1. Determine the current file access modes of the client configuration token.

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

    2. 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.
5. Save your changes to the `/etc/nvidia/gridd.conf` file and close the file.
6. Restart the `nvidia-gridd` service.

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

Chapter 3. Advanced Topics

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

3.1. Configuring a Licensed Client with Custom Settings

NVIDIA License System provides custom settings for the following configuration properties of a licensed client:

- The feature type of a physical GPU
- The directory in which to store the client configuration token
- Details for a proxy server between a licensed client and a service instance

If you want to use the default directory in which to store the client configuration token and, on Windows, are configuring the client with NVIDIA vGPU, follow the instructions for a simplified configuration in *Configuring a Licensed Client of NVIDIA License System*.

3.1.1. Configuring a Licensed Client on Windows with Custom Settings

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\SYSTEM\CurrentControlSet\Services\nvlddmkm\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.
   - You can also perform this step from NVIDIA Control Panel.

   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. 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\SYSTEM\CurrentControlSet\Services\nvlddmkm\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. 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 NVIDIA vGPU software graphics driver is installed.

4. 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.

5. If you want the licensed client to check out a license when a user logs in to the client, add the `EnableLicenseOnLogin` DWord (`REG_DWORD`) registry value to the Windows registry key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global\GridLicensing` and set this registry value to `1`.

   By default, a licensed client checks out a license when the client is booted.

6. If a non-transparent proxy server is configured between your licensed client and a service instance, provide the information about the proxy server that the licensed client requires.

   Provide this information by adding the following registry values to the Windows registry key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global\GridLicensing`.

1. For all non-transparent proxy servers, provide the address and port number of the proxy server in the following Windows registry values.

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.

2. If necessary, provide the credentials of the user that will log in to the proxy server.

   This information is required for proxy servers that use the following authentication methods:

   - Basic authentication
   - Microsoft Windows Challenge/Response (Microsoft NTLM) authentication for a client that is not a member of an Active Directory domain `ProxyUserName` String `(REG_SZ)` : The username of the user that will log in to the proxy server.

ProxyCredentialsFilePath String (REG_SZ)
The full path to the file that contains the encrypted credentials of the user that will log in to the proxy server, for example, `C:\Program Files\NVIDIA Corporation\vGPU Licensing\ProxySettings\proxy-credentials.dat`.

This file is generated as explained in *Generating an Encrypted Credentials File*.

7. Restart the `NvDisplayContainer` service.

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

3.1.2. Configuring a Licensed Client on Linux with Custom Settings

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

Note

You can also perform this step from NVIDIA X Server Settings. Before using NVIDIA X Server Settings to perform this step, ensure that this option has been enabled as explained in *Enabling License Management in NVIDIA X Server Settings*.

This example shows how to configure a licensed Linux client for NVIDIA RTX Virtual Workstation.

```
# /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 vGPU for Compute
FeatureType=2
...
```

3. If you want store the client configuration token in a custom location, add the `ClientConfigTokenPath` configuration parameter to the file `/etc/nvidia/gridd.conf` on a new line as `ClientConfigTokenPath="<path>"`

<path>
The full path to the directory in which you want to store the client configuration token for the client. By default, the client searches for the client configuration token in the `/etc/nvidia/ClientConfigToken/` directory.

By specifying a shared network directory that is mounted locally 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 directory.

This example shows how to configure a licensed Linux client to search for the client configuration token in the `/mnt/nvidia/ClientConfigToken/` directory. This directory is a mount point on the client for a shared network directory.

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

(continues on next page)

(continued from previous page)

```
ClientConfigTokenPath=/mnt/nvidia/ClientConfigToken/
...
```

4. 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 NVIDIA vGPU software graphics driver is installed.

5. 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.

6. 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.

   1. Determine the current file access modes of the client configuration token.

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

   2. 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.

7. If you want the licensed client to check out a license when a user logs in to the client, add the `EnableLicenseOnLogin` configuration parameter to the file `/etc/nvidia/gridd.conf` on a new line as `EnableLicenseOnLogin=TRUE`.

   By default, a licensed client checks out a license when the client is booted.

8. If a non-transparent proxy server is configured between your licensed client and a service instance, provide the information about the proxy server that the licensed client requires.

   1. For all non-transparent proxy servers, provide the address and port number of the proxy server.

      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
...
```

2. If necessary, provide the credentials of the user that will log in to the proxy server.

   This information is required for proxy servers that use basic authentication.

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

```
ProxyUserName=<domain>\<username>
ProxyCredentialsFilePath=<path>
```

<domain>
The domain to which the user belongs, for example, `example.com`.

<username>
The username of the user that will log in to the proxy server, for example, `clsuser`.

<path>
The full path to the file that contains the encrypted credentials of the user that will log in to the proxy server, for example, `/etc/nvidia/proxy-credentials.dat`.

This file is generated as explained in *Generating an Encrypted Credentials File*.

This example sets the domain and username of the user that will log in to the proxy server to `example.com\clsuser` and the path to the file that contains the encrypted credentials of the user to `/etc/nvidia/proxy-credentials.dat`.

```
# /etc/nvidia/gridd.conf.template - Configuration file for NVIDIA
 ↪Grid Daemon
…
ProxyUserName=example.com\clsuser
ProxyCredentialsFilePath=/etc/nvidia/proxy-credentials.dat
...
```

9. Save your changes to the `/etc/nvidia/gridd.conf` file and close the file.
10. Restart the `nvidia-gridd` service.

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

3.1.3. 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.

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.

3.2. Configuring a Proxy Server Between a Licensed Client and a Service Instance

3.2.1. Proxy Server Requirements and Firewall Rules for a Service Instance

To enable communication between a licensed client and a service instance through a proxy server, the proxy server must meet certain requirements. To enable communication through a firewall, firewall rules that allow traffic on specific URLs through specific ports must be defined.

The processes for configuring a proxy server and defining firewall rules are separate from the process for configuring a service instance. Use the standard interfaces of the proxy server and the firewall that you are using to perform these processes.

> **Note**
>
> **Only** HTTP proxy servers are supported. HTTPS proxy servers are **not** supported.

3.2.1.1 Proxy Server Requirements for a Service Instance

NVIDIA License System supports transparent proxy servers and non-transparent proxy servers.

- **A transparent proxy server** identifies itself to the server and does **not** modify client requests and responses.
- **A non-transparent proxy server** does not reveal the IP address of the client and modifies client requests and responses.

Any proxy server between a licensed client and a service instance must allow programmatic calls to the URL through which the service instance is accessed.

Service Instance Type	URL
CLS instance	`api.cls.licensing.nvidia.com`
DLS instance	The IP address or, if defined, the fully qualified domain name or the CNAME of the VM on which the DLS virtual appliance is installed

3.2.1.2 Non-Transparent Proxy Server Support

NVIDIA License System supports both authenticated and unauthenticated non-transparent proxy servers.

The following authenticated proxy servers are supported:

- Squid

The following authentication methods are supported for authenticated proxy servers:

- Basic
- Microsoft Windows Challenge/Response (Microsoft NTLM) (Windows clients only)
- Kerberos (only for clients that are a member of an Active Directory domain)

3.2.1.3 Firewall Rules for a Service Instance

To enable communication between a licensed client and a service instance through a firewall, firewall rules that allow traffic on the URLs through the ports specified in the following table must be defined.

CLS URL	DLS URL	Port	Traffic
`api.cls.licensing.nvidia.com`	The IP address or, if defined, the fully qualified domain name or the CNAME of the VM on which the DLS virtual appliance is installed	443	▶ Licensing operations, namely, the borrowing, renewal, and return of a license. ▶ Licensed client authentication
`api.licensing.nvidia.com`	The IP address or, if defined, the fully qualified domain name or the CNAME of the VM on which the DLS virtual appliance is installed	80	License return from a Windows licensed client that has not been shut down cleanly

3.2.2. Generating an Encrypted Credentials File

Some authentication methods require a licensed client to provide user credentials when the client authenticates with a proxy server. To enable the client to provide these credentials securely without input from a user, you must generate a file that contains these credentials in an encrypted form that the client can read.

The following authentication methods require an encrypted credentials file:

- Basic authentication
- Microsoft Windows Challenge/Response (NTLM) authentication for a client that is not a member of an Active Directory domain

How to generate an encrypted credentials file depends on the OS that client is running. For detailed instructions, refer to the following topics:

- *Generating an Encrypted Credentials File on Windows*
- *Generating an Encrypted Credentials File on Linux*

3.2.2.1 Generating an Encrypted Credentials File on Windows

Perform this task in a Windows PowerShell window as the Administrator user on the client.

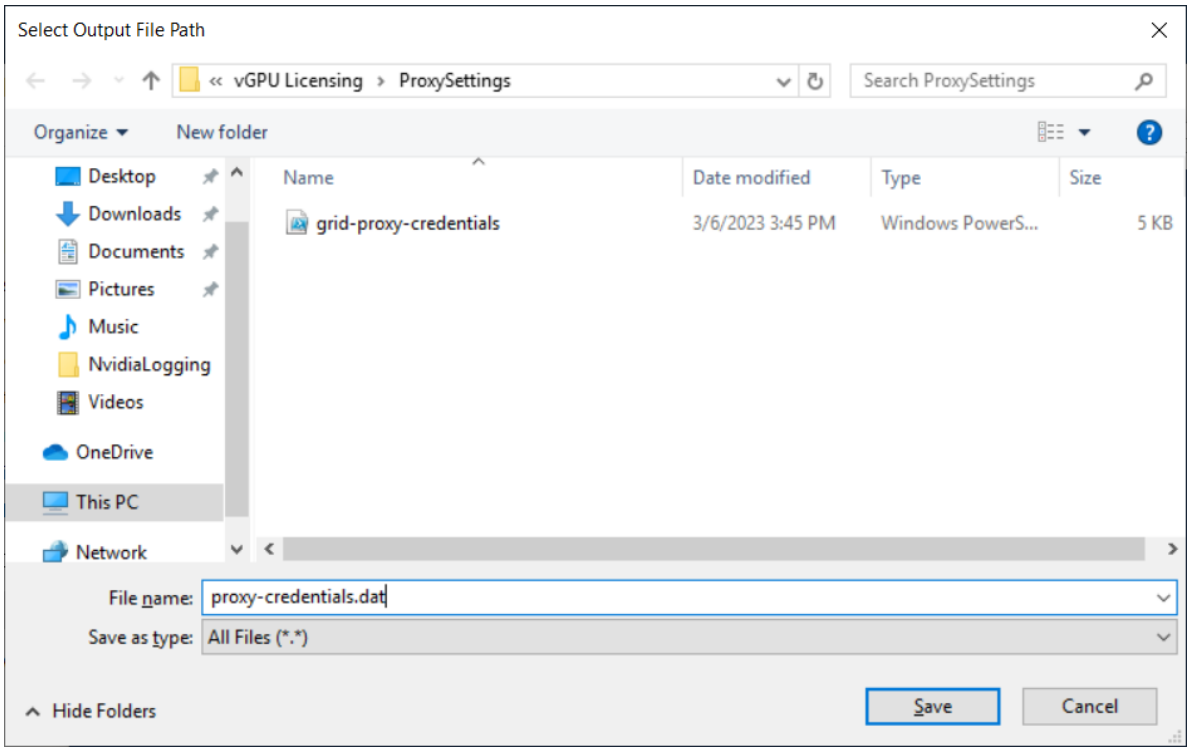
1. Change to the `C:\Program Files\NVIDIA Corporation\vGPU Licensing\ProxySettings` folder.

```
PS C:\> cd "C:\Program Files\NVIDIA Corporation\vGPU Licensing\
↪ProxySettings"
```

2. Run the `grid-proxy-credentials` Windows PowerShell script.

```
PS C:\> .\grid-proxy-credentials.ps1
```

3. In the Select Output File Path window that opens, navigate to the directory in which you want to generate the credentials file, enter the file name, and click **Save**.



4. When prompted in the Windows PowerShell window, specify the password of the user that will log in to the proxy server when the licensed client requests a license.

Provide the path to this file when configuring a licensed client that will use the file as explained in *Configuring a Licensed Client on Windows with Default Settings*.

3.2.2.2 Generating an Encrypted Credentials File on Linux

Perform this task in a Linux command shell on the client.

1. Run the `grid-proxy-credentials.sh` command.

```
# /usr/lib/nvidia/grid-proxy-credentials.sh -o <output-file-path>
```

<output-file-path>
The full path to the credentials file that you are generating. Ensure that the directory in the path exists.

> **Tip**
>
> To get help information for this command, type `/usr/lib/nvidia/grid-proxy-credentials.sh --help`.

This example creates the credentials file `/etc/nvidia/proxy-credentials.dat`.

```
# /usr/lib/nvidia/grid-proxy-credentials.sh -o /etc/nvidia/proxy-
↪credentials.dat
```

2. When prompted, specify the password of the user that will log in to the proxy server when the licensed client requests a license.

Provide the path to this file when configuring a licensed client that will use the file as explained in *Configuring a Licensed Client on Linux with Default Settings*.

3.3. 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\SYSTEM\CurrentControlSet\Services\nvlddmkm\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 *Configuring a Licensed Client of NVIDIA License System*.

3.4. Licenses Obtained After Boot or User Login

Under normal operation, an NVIDIA vGPU software license is obtained by a platform during boot or at user login, before 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* 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.

3.5. 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 or when a user logs in to the client.

After they have obtained a license, 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.

3.6. Windows Registry Settings for NVIDIA vGPU Software Licensing

NVIDIA vGPU software licensing settings are controlled through the Windows Registry.

3.6.1. Registry Key Values for Licensing Settings

Licensing settings are stored in this registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global\
↪GridLicensing
```

Registry values are summarized in Table 3.1.

Table 3.1: Licensing Registry Settings

Name	Type	Description
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.
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.
EnableLicenseOnLogin	DWord (REG_DWORD)	Controls whether a licensed client checks out a license when the client is booted or when a user logs in to the client. If this registry key is absent, a licensed client checks out a license when the client is booted. When present: 0: A licensed client checks out a license when the client is booted. 1: A licensed client checks out a license when a user logs in to the client. (Optional: defaults to 0)

continues on next page

Table 3.1 – continued from previous page

Name	Type	Description
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) 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)

continues on next page

Table 3.1 – continued from previous page

Name	Type	Description
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.
ClientConfigTokenPath	String (REG_SZ)	The full path to the folder in which you want to store the client configuration token for the client. Set this registry key **only** if you want store the client configuration token in a custom location.
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 service instance.
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 service instance.

3.6.2. Registry Key Value for Managing Licensing Controls in NVIDIA Control Panel

Licensing controls in NVIDIA Control Panel are controlled through the NvCplDisableManageLicensePage DWord (REG_DWORD) registry value in this Windows registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\nvlddmkm\NVControlPanel2\
↪RegisteredServers\Overrides\{DFA226E7-D28D-407D-95ED-5A79D9745BB5}
```

If this registry key value is absent, licensing controls are shown in NVIDIA Control Panel for GPUs supporting licensing.

If this registry key value is present:

- 0: Show licensing controls in NVIDIA Control Panel
- 1: Hide licensing controls in NVIDIA Control Panel
- All other values: reserved

Chapter 4. Troubleshooting

This chapter describes basic troubleshooting steps.

4.1. 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.

4.2. 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 from the licensed client, **not** the hypervisor host. If the product is licensed, the expiration date is shown in the license status.

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

Timestamp                                 : Wed Nov 23 10:52:59 2022
Driver Version                            : 525.60.06
CUDA Version                              : 12.0

Attached GPUs                             : 2
GPU 00000000:02:03.0
    Product Name                          : NVIDIA A2-8Q
    Product Brand                         : NVIDIA RTX Virtual Workstation
    Product Architecture                  : Ampere
    Display Mode                          : Enabled
    Display Active                        : Disabled
    Persistence Mode                      : Enabled
    MIG Mode
        Current                           : Disabled
        Pending                           : Disabled
    Accounting Mode                       : Disabled
```

(continues on next page)

(continued from previous page)

```
    Accounting Mode Buffer Size           : 4000
    Driver Model
        Current                           : N/A
        Pending                           : N/A
    Serial Number                         : N/A
    GPU UUID                              : GPU-ba5b1e9b-1dd3-11b2-be4f-
↪98ef552f4216
    Minor Number                          : 0
    VBIOS Version                         : 00.00.00.00.00
    MultiGPU Board                        : No
    Board ID                              : 0x203
    Board Part Number                     : N/A
    GPU Part Number                       : 25B6-890-A1
    Module ID                             : N/A
    Inforom Version
        Image Version                     : N/A
        OEM Object                        : N/A
        ECC Object                        : N/A
        Power Management Object           : N/A
    GPU Operation Mode
        Current                           : N/A
        Pending                           : N/A
    GSP Firmware Version                  : N/A
    GPU Virtualization Mode
        Virtualization Mode               : VGPU
        Host VGPU Mode                    : N/A
    vGPU Software Licensed Product
        Product Name                      : NVIDIA RTX Virtual Workstation
        License Status                    : Licensed (Expiry: 2022-11-23
↪10:41:16 GMT)
    …
    …
```

4.3. Licensing Event Logs

Significant licensing events 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`.

Examples of significant licensing events that are logged are as follows:

- Acquisition of a license
- Return of a license
- Expiration of a license
- Failure to acquire a license
- License state changes between the unlicensed restricted state (20 mins), unlicensed state (24 hours), and licensed state

4.3.1. 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 `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\nvlddmkm\Global\GridLicensing`.

License state changes are reported in Event Viewer under **Windows Logs** in the **Application** category. To open Event Viewer, press **Windows**+**R**, type `eventvwr.msc`, and press **ENTER**.

4.3.2. 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.

4.3.3. 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
```

(continues on next page)

(continued from previous page)

```
↪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
```

4.3.4. 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.
```

4.4. 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.

4.5. Maximum Allowed Network Adapters on a VM Exceeded

If a VM is configured with too many network adapters, the VM might fail to acquire or return a license. The exact number of network adapters allowed on a VM depends on your system. In testing, it was verified that a VM with **40** network adapters could successfully acquire a license.

If a VM fails to acquire or return a license because the VM is configured with too many network adapters, the following error message is written to the licensing event log file:

```
Failed to update local trusted store - Maximum buffer size exceeded
```

For information about this log file, refer to *Licensing Event Logs*.

How to recover from such a failure depends on the guest OS that is running in the VM as explained in the following topics:

- *Windows: Recovering from a Failure Caused by Too Many Network Adapters*
- *Linux: Recovering from a Failure Caused by Too Many Network Adapters*

4.5.1. Windows: Recovering from a Failure Caused by Too Many Network Adapters

If a Windows VM fails to acquire or return a license because the VM is configured with too many network adapters, reduce the number of network adapters on the VM.

1. Stop the `NvDisplayContainer` service.
2. Remove some network adapters from the VM.
3. Delete all files in the `%SystemDrive%\Program Files\NVIDIA Corporation\vGPU Licensing\TrustedStorage` folder, which contains the trusted storage database files.
4. Start the `NvDisplayContainer` service.

4.5.2. Linux: Recovering from a Failure Caused by Too Many Network Adapters

If a Linux VM fails to acquire or return a license because the VM is configured with too many network adapters, reduce the number of network adapters on the VM.

1. Stop the `nvidia-gridd` service.

   ```
   # sudo service nvidia-gridd stop
   ```

2. Remove some network adapters from the VM.
3. Delete all files in the `/var/lib/nvidia/vGPULicensing/` directory, which contains the trusted storage database files.

   ```
   # sudo \rm -rf /var/lib/nvidia/vGPULicensing/*
   ```

4. Start the `nvidia-gridd` service.

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

Chapter 5. Notices

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

©2013-2026, NVIDIA Corporation