RealityServer® Web Services 3.1 Installation Guide

Document version 1.28
December 8, 2010

Copyright Information

The information contained in this document is subject to change without notice. NVIDIA Corporation and its employees shall not be responsible for incidental or consequential damages resulting from the use of this material or liable for technical or editorial omissions made herein.

NVIDIA, the NVIDIA logo, and DiCE, imatter, iray, mental cloud, mental images, mental matter, mental mesh, mental mill, mental queue, mental ray, Metanode, MetaSL, neuray, Phenomenon, RealityDesigner, RealityPlayer, RealityServer, rendering imagination visible, Shape-By-Shading, and SPM, are trademarks and/or registered trademarks of NVIDIA Corporation. Other product names mentioned in this document may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Table of Contents

System Requirements

Memory

Memory requirements greatly depend on the size of the 3D scene, and the number of different scenes concurrently loaded. Also renderers need different amounts of memory.

All hosts forming one RealityServer® cluster should have the same amount of memory. 2 GB is a typical minimum. 8 GB is better. Some large scenes with several hundreds of millions of triangles require 32 GB if the RT_BSP renderer is used.

Processors

RealityServer should run on a machine using 64-bit processors. RealityServer will make use of multiple processor and cores found on a machine.

The Developer Edition does not require licenses, but is limited to 16 cores and up to 4 GPUs.

32-bit CPUs are supported on Windows, using a separate executable, but can run into memory limitations with large scenes.

Graphics Card

The usage of CUDA capable NVIDIA graphics cards is highly recommended when using the iray renderer. If no such graphics card is found iray will fall back to CPU based rendering. An OpenGL capable graphics card is necessary when using the GPU or SKETCH renderer.

The following graphics cards are supported:

- NVIDIA Quadro® 6000
- NVIDIA Quadro® 5000
- NVIDIA Quadro® 4000
- NVIDIA Quadro® FX 5800
- NVIDIA Quadro® FX 5600
- NVIDIA Quadro® FX 5500
- NVIDIA Quadro® FX 4800
- NVIDIA Quadro® FX 4700 X2
- NVIDIA Quadro® FX 4600
- NVIDIA Quadro® FX 3800
- NVIDIA Quadro® FX 3700
- NVIDIA Quadro Plex® 2200 S4
- NVIDIA Quadro Plex® 2200 D2

- NVIDIA Tesla® C1060
- NVIDIA Tesla® C2050
- NVIDIA Tesla® C2070
- NVIDIA Tesla® M1060
- NVIDIA Tesla® M2050
- NVIDIA Tesla® M2070
- NVIDIA Tesla® S1070
- NVIDIA Tesla® S2050

The Developer Edition of RealityServer also supports NVIDIA Geforce® graphics cards if they support shader model 3 but they might only support a subset of the rendering capabilities of RealityServer.

Please visit `www.mentalimages.com` for an updated lists of supported cards.

It is necessary to use the Cg and CUDA libraries that come with RealityServer. The `LD_LIBRARY_PATH` (Linux) or `PATH` (Windows) environment variable must include the directory where the libraries are installed. This is done automatically if the supplied startup scripts such as `realityserver` are used.

Recent NVIDIA® drivers including OpenGL must be installed on the system already. (They are provided by NVIDIA, not by the operating system.)

The following drivers are recommended:

	Windows	Linux
Quadro	265.90	260.19.21
Tesla	263.06	260.19.21

Linux systems must have an accessible X display server (`xhost +`).

In case of problems, use the `-gputrace` command-line option to enable additional diagnostic messages.

ECC Functionality on Fermi based GPUs

The new NVIDIA Fermi® architecture supports new ECC functionality which is enabled by default on all Fermi cards and drivers. This functionality consumes additional GPU memory and potentially reduces overall performance for programs which do not require ECC functionality such as RealityServer. For maximum compatibility we strongly recommend disabling ECC functionality on Fermi hardware before using RealityServer.

Linux

The ECC functionality can be configured on Linux utilising the 'nvidia-smi' program included with the relevant driver packages. To determine your current ECC configuration run the command:

```
nvidia-smi --report-ecc-config
```

You should see output similar to the following:

```
ECC configuration for GPU 0:
        Current: 1
        After reboot: 1
ECC configuration for GPU 1:
        Current: 1
        After reboot: 1
```

This indicates that ECC is currently enabled for these GPUs. To disable ECC please use the following commands:

```
nvidia-smi --ecc-config=0 --gpu=0
nvidia-smi --ecc-config=0 --gpu=1
```

Note that you will need to run this command for each GPU in your system with the relevant ID. After configuring your ECC options please reboot your machine to make them active.

Windows

Under Windows you configure the ECC functionality from the 'NVIDIA Control Panel' usually available by right clicking on your desktop and selecting the 'NVIDIA Control Panel' item. You may need to enable the Advanced Configuration mode if asked.

In the panel on the left labelled 'Select a Task...', navigate to the 'Workstation' section and select 'Change ECC state', from this item you can deselect ECC for the GPUs in your system. After changing the ECC state please reboot your system for the changes to take effect.

Note that the required commands and descriptions change from time to time with updated drivers, please refer to the relevant NVIDIA documentation for further information on how to disable ECC functionality on your GPU.

Linux irqpoll Kernel Option

On certain machines and configurations (in particular machines with 8 or more GPUs) you may see messages such as this during the use of RealityServer when rendering with iray:

```
Disabling IRQ #194
irq 178: nobody cared (try booting with the "irqpoll" option)
```

These messages are usually visible on the console as well as in the '/var/log/messages' output on most Linux distributions. If you encounter these messages it is very likely that you will also encounter severely degraded performance with GPU accelerated iray rendering. If, and only if you encounter this message, you should enable the Linux Kernel 'irqpoll' option to prevent these issues.

The process will differ depending on your specific distribution and boot loader being used. For example on CentOS 5.5 you will need to edit your GRUB bootloader configuration file in '/boot/grub/menu.lst'. Here is an example of an entry which adds the necessary 'irqpoll' option during startup.

```
title CentOS (2.6.18-194.8.1.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-194.8.1.el5 ro
root=/dev/VolGroup00/LogVol00 irqpoll
```

You may then need to re-install your boot loader using the standard process. Some distributions include graphical tools for making changes to the boot configuration. Please refer to your specific distributions documentation for instructions on how to add boot time kernel options.

Windows Remote Desktop

Please note that starting RealityServer from within a Windows Remote Desktop session is likely to prevent the GPU functionality of RealityServer from working correctly. As Windows Remote Desktop utilises its own display driver which does not provide 3D acceleration, RealityServer is not able to access the underlying GPU correctly.

For situations requiring the use of Windows Remote Desktop it is recommended to either utilise the RealityService program to run RealityServer as a Windows service or alternatively switch to remote desktop software which does not impose its own graphics driver.

Operating System

Supported operating systems are Linux and Windows on x86 processors for both 32 bit and 64 bit platforms.

On Linux at least SuSE® Version 9.3, or Red Hat® Enterprise Linux® 4 should be used. Both 32-bit and 64-bit versions are available. An X11 server must be installed if GPU rendering is to be used. On Linux RealityServer has an external dependency on libstdc++.so.5 . This library exists on RHEL4 (CentOS4) per default but needs to be installed as compatiblity library on more recent Linux distributions (compat-libstdc++ package).

The Windows support includes Windows XP, Vista and Windows 7 in 32-bit and 64-bit versions.

Network

For clustering Gigabit Ethernet is recommended. Slower networks work, but can greatly slow down multi-hosted rendering, especially at startup and for some time after loading a large 3D world.

Selection and configuration of the Ethernet switch is critical:

- In the default network configuration RealityServer may generate a large amount of multicast traffic. Without care, this traffic might be forwarded to other hosts in the net, thus consuming a lot of the available bandwidth. Better switches allow enabling a feature called *multicast snooping* or *multicast spoofing*. This will allow the switch to forward multicast traffic only to hosts that have registered interest in those multicast groups. For this reason, this feature should be switched on if it is available. Please refer to the user manual of your switch for instructions to do this.

  Note that RealityServer 3.0 supports other modes of networking which only generate low bandwidth multicast traffic or no multicast at all. See the reference manual for details.

- In general, a RealityServer setup with several hosts may generate a large amount of network traffic. Consumer grade network switches are usually not able to handle this amount of traffic, and might cause network packet loss, resulting in low performance and bad scalability (adding more hosts to the network does not increase throughput as expected).

The Cisco 3750G and 2960G switches, and the Foundry SuperX switch have been found to work well.

Note that RealityServer's load balancing and resource allocation strategies assume that all memory, all CPU cycles, and the entire network bandwidth are at its disposal. It will dynamically ramp up its usage to optimally use all available resources. It is generally not a good idea to run other large or interactive programs (or a second RealityServer) on a RealityServer host.

RealityServer Web Services Installation

RealityServer Web Services is supplied as a simple archive. There is no need to install anything. You can simply unpack the archive to a directory of your choice and use the supplied shell scripts in the unpacked directory to start RealityServer. Then please connect with a browser to port 8080 on the machine where RealityServer was started.

Removal of RealityServer is done by simply deleting the unpacked directory.

Please see the README file for details of the contents of that directory.

Firewalls

Operation of the RealityServer involves network connections. On Windows, there are often firewalls installed (such as the generic Windows firewall) that block network connections, in a way that makes it impossible for the blocked program to detect or react to the problem. RealityServer cannot be run on a machine that requires this kind of firewall protection unless the firewall is manually instructed to allow to open the port listening for incoming connections on the configured HTTP port.

Network Buffers

For maximum performance of the HTTP server on some Linux systems the possible size of network buffers must be increased because the default values are too low (256 kB). Please use the following lines to increase them:

```
sysctl -w net.core.rmem_max = 8388608
sysctl -w net.core.wmem_max = 8388608
```

These settings are reset when rebooting. To make them permanent, add these lines to the file */etc/sysctl.conf*:

```
net.core.wmem_max = 8388608
net.core.rmem_max = 8388608
```

Software Protection Manager (SPM)

See the separate SPM installation guide. The software protection manager manages the software licenses that are required by RealityServer. The RealityServer will not run if no licenses are available.

The RealityServer Developer Edition does not need an SPM installation.

RealityServer Web Services Networking Setup

When RealityServer is used in networked mode, multiple RealityServers started on different hosts will work together in storing data, handling user requests and rendering images. Those hosts will form a so-called cluster. All hosts in the cluster are equivalent and do the same work and offer the same services.

It is not necessary to perform any kind of setup. When a RealityServer starts, it automatically searches its local multicast group, and finds a cluster to join. This is also true if a RealityServer on a host dies for any reason, including power loss, and rejoins the network later. No host list is needed, everything is fully automatic.

However, this means that in any network of hosts that share a multicast group, only one RealityServer network can exist, because they will all find each other and form a single network.

When starting up RealityServers in a network all RealityServer hosts must have completed startup before handling user requests.

Network Setup

For correct startup of the networked discovery, these prerequisites must be met:

- The network setup must allow unicast and also multicast traffic to travel between all the hosts to be connected. (See the System Requirements section.)
- Each host must be able to identify its own IP address. On Linux systems check the file `/etc/hosts`. It should contain a line of the form *own-ip-address own-host-name*. Issuing the command *hostname* on the command line should give you this hostname, too. If it returns "localhost", the setting is wrong.
- For maximum performance, the possible size of network buffers must be increased. On Linux systems, the default values are too low (256 kB). Please use the following lines to increase them:

  ```
  sysctl -w net.core.rmem_max = 8388608
  sysctl -w net.core.wmem_max = 8388608
  ```

  These settings are reset when rebooting. To make them permanent, add these lines to the file */etc/sysctl.conf*:

  ```
  net.core.wmem_max = 8388608
  net.core.rmem_max = 8388608
  ```