
DCV

User Guide

Version **2016.0**

Copyright

Copyright © 2010-2017, NICE s.r.l.

Copyright © 2004-2009, IBM Corporation

All right reserved.

We Would Like to Hear from You

You can help us make this document better by telling us what you think of the content, organization, and usefulness of the information. If you find an error or just want to make a suggestion for improving this document, please address your comments to [<documentation@nice-software.com>](mailto:documentation@nice-software.com). Please send only comments regarding NICE documentation.

For product support, contact [<helpdesk@nice-software.com>](mailto:helpdesk@nice-software.com).

Although the information in this document has been carefully reviewed, NICE s.r.l. ("NICE") does not warrant it to be free of errors or omissions. NICE reserves the right to make corrections, updates, revisions, or changes to the information in this document.

UNLESS OTHERWISE EXPRESSLY STATED BY NICE, THE PROGRAM DESCRIBED IN THIS DOCUMENT IS PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL NICE BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION ANY LOST PROFITS, DATA, OR SAVINGS, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS PROGRAM.

Document Redistribution and Translation

This document is protected by copyright and you may not redistribute or translate it into another language, in part or in whole, without the express written permission of NICE s.r.l.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States or other countries or both: IBM, IBM logo.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is the registered trademark of Linus Torvalds in the United States and other countries.

Microsoft, Windows, Internet Explorer, DirectX, Direct3D are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Mac and OS X are a trademark of Apple Inc., registered in the U.S. and other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

VNC is a registered trademark of RealVNC Ltd. in the United States and other countries.

OpenGL is a registered trademark of SGI.

NVIDIA, GRID vGPU, NVIDIA GRID, NVIDIA GRID vGPU, Quadro, Tesla, vGPU are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and/or other countries.

Other products or services mentioned in this document are identified by the trademarks or service marks of their respective owners.

Last Update

September 21, 2017 (rev. 17066)

Latest Version

<https://www.nice-software.com/download/dcv>

Contents

Welcome	v
About This Guide	v
Save this Document	v
Who Should Read This Guide	v
Learn About NICE Products	vi
World Wide Web	vi
NICE DCV Documentation	vi
Get Technical Support	vii
NICE Support Contacts	vii
1. Prerequisites	1
End Station Requirements	1
Network Requirements	2
Additional Requirements for NICE External Rendering Server	2
2. Installing DCV End Station	5
Installing Windows End Stations	5
Installing Mac End Stations	6
Installing Linux End Stations	6
Using Portable End Stations	7
Using Windows Portable End Station	7
Using Linux Portable End Station	7
Using Mac Portable End Station	7
Advanced Installations	8
Windows Unattended Installation	8
Installing Linux End Stations from RPMs	8
Configuring Proxy Support	9
Configuring Encryption Support	9
3. Using Desktop Cloud Visualization	11
Using DCV on an End Station	11
Using DCV on a Windows End Station	11
Using DCV on a Mac End Station	11
Using DCV on a Linux End Station	11
Using DCV from a plain VNC client	12
Using DCV Console	12
Using DCV in a Remote Desktop Connection	16
Using DCV on an Application host	19
Command Line Tools	19
The DCV Config utility	20
Running an Application with DCV on Linux	21
Running an Application with DCV Rendering Server	22
A. dcv.conf	23
[Application]	23
external_rendering_server	23
local_display_list	24
local_multisample	24

print_dcv_version	24
target_fps	25
use_nvidia_grid	25
use_sample_queries	26
[Endstation]	26
console_shortcut	26
enable_console	26
encryption	27
proxy_port	27
proxy_server	27
proxy_type	28
tip_timeout	28
[License]	29
license_file	29
use_platform_product	29
[Logging]	30
log_dir	30
log_file	30
log_level	31
[Remotization]	31
allow_udp	31
blacklist_vchannels	32
codec_qu	32
codec_tcp	32
default_proxy_port	33
default_proxy_server	33
default_proxy_type	34
enable_quality_update	34
encryption	34
host	35
image_quality	35
quality_update_interval	35
quality_update_quality	36
tcp_max_port	36
tcp_start_port	36
udp_quality	37
udp_start_port	37
use_nvidia_hw_encoder	37
use_udp	38
[RenderingServer]	38
agent_max_port	38
agent_start_port	38
host	39
local_display_list	39
port	39
user	40
[Wrapper]	40
disable_driver_unload	40

About This Guide

This document describes the installation, configuration, and operation of NICE Desktop Cloud Visualization (DCV).

NICE DCV provides high-performance visualization of complex OpenGL and DirectX applications across low-bandwidth, high-latency networks to remote or distributed locations.

Save this Document

Retain this document with your original system. This document emphasizes recent system information and does not include complete information about previous releases.

Who Should Read This Guide

This guide is intended for:

- system administrators who need to install and configure NICE DCV;
- users who want to remotely access their OpenGL and DirectX applications using NICE DCV.

Learn About NICE Products

World Wide Web

You can find the latest information about NICE DCV on its web site

<https://www.nice-software.com/dcv>.

For more information about other NICE products and about the professional services provided by NICE you can refer to the company's web site <https://www.nice-software.com>.

Report problems accessing the aforementioned web sites to [<helpdesk@nice-software.com>](mailto:helpdesk@nice-software.com).

NICE DCV Documentation

The latest NICE DCV documentation is available at

<https://www.nice-software.com/download/dcv>.

Get Technical Support

Contact NICE or your DCV reseller for technical support.

NICE Support Contacts

Use one of the following to contact NICE technical support.

Email

[<helpdesk@nice-software.com>](mailto:helpdesk@nice-software.com)

World Wide Web

<http://support.nice-software.com/>

Phone

+39 0141 901516

Mail

NICE Support
c/o NICE s.r.l.
Via Milliavacca, 9
14100 Asti
Italy

When contacting NICE, please include your company's full name.

Prerequisites

This chapter lists the hardware and software prerequisites for installing Desktop Cloud Visualization.

End Station Requirements

Table 1.1. End Station Requirements

End Station Requirements	
Operating system	• Red Hat Enterprise Linux 5.x, 6.x, 7.x 32/64-bit
	• SUSE Enterprise Server 11 SP2 32/64-bit
	• Microsoft Windows 7 32/64-bit
	• Microsoft Windows 8, 8.1 32/64-bit
	• Microsoft Windows 10 32/64-bit
	• Mac OS X Snow Leopard (10.6), Lion (10.7), Mountain Lion (10.8), Mavericks (10.9), Yosemite (10.10)

The DCV installer includes and automatically installs RealVNC Visualization Edition 4.6.x (Viewer).

Portable end station packages which do not require installation are available. See the section called “Using Portable End Stations” for more information.

The application host and its end stations can run different operating systems.

Different versions of DCV Server and DCV End Station are compatible but some features may not be available when not using the latest version.

DCV is compatible with many *plain-VNC* clients from third parties. When using such clients, 3D images are delivered to the clients using the standard VNC protocol and therefore with reduced performance.

Network Requirements

DCV is a network aware technology and may require extra firewall configuration. Your firewall should allow traffic on the interval of TCP ports between 7300 and 7399. DCV uses ports in this interval to establish a connection for the 3D channel.

The ports can be changed in the `dcv.conf` configuration file (see Appendix A, `dcv.conf`).

RealVNC requires its own ports to open 2D channel. In a standard configuration your firewall should also allow traffic on:

- (Windows) TCP port 5900
 - (Linux) the interval of TCP ports between 5900 and 5999
- Please refer to RealVNC documentation for further information.

Table 1.2. Summary of default firewall configuration

Host	Process	Rule
Application Host	RealVNC Server	(Linux) Allow incoming TCP connections on 5900-5999 ports from end stations (Windows) Allow incoming TCP connections on 5900 port from end stations
	OpenGL processes (Linux) / <code>dcvd</code> process (Windows)	(Linux) Allow incoming TCP connections on 7300-7399 ports from end stations (Windows) Allow incoming TCP connections on 7300 port from end stations



Note

In addition to local firewall configuration check no external firewall is blocking the communication between end stations and application host.

Additional Requirements for NICE External Rendering Server

When using DCV with NICE External Rendering Server configuration:

- the 3D channel uses connections from the end station to the *rendering host* (instead of the application host);
- the DCV rendering server uses TCP connections from the *application host* to the *rendering host* (port: 2007);
- the OpenGL applications use TCP connections from the *application host* to the *rendering host* (ports: starting from 34567, one port for each DCV Rendering Agent).

The network between the rendering server and application hosts does not need to be exposed to other hosts and therefore the suggested network topology is:

- **Host-Only Network:** an isolated network between rendering host and application hosts;
- **Public Network:** a network reachable to rendering hosts, application hosts and end stations.

Table 1.3. Summary of default firewall configuration for External 3D Rendering

Host	Network	Process	Rule
Rendering Host	Public	DCV Rendering Agents	Allow incoming TCP connections on 7300-7399 ports from end stations
	Host-Only	DCV Rendering Server	Allow incoming TCP connections on 2007 port from application hosts (See Note)
	Host-Only	DCV Rendering Agents	Allow incoming TCP connections on 34567 port and above (one port for each agent) from application hosts (See Note)
Application Host	Public	RealVNC Server	(Windows) Allow incoming TCP connections on 5900 port from end stations
	Host-Only	OpenGL Applications	Allow outgoing TCP connections to rendering host on 2007 port (See Note)



Note

Since the Host-Only Network is an isolated network it is suggested to shut down on both application host and rendering host the firewall on the Host-Only network interfaces.

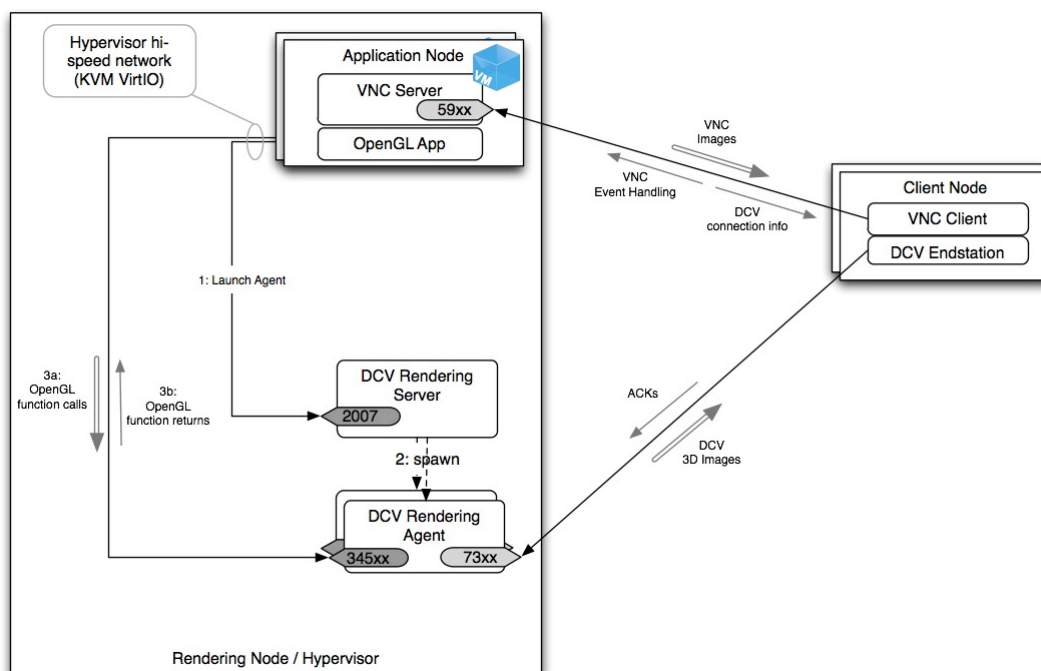


Figure 1.1. DCV with External 3D Rendering

Figure 1.1, “DCV with External 3D Rendering” illustrates the typical deployment when DCV works in External 3D Rendering configuration.

Installing DCV End Station

The installation of the DCV End Station requires administrative access to the client host. In case administrative privileges are not available, portable packages which do not require any installation can be used. See the section called “Using Portable End Stations” for more information.

See the following sections for OS-specific installation instructions.

Installing Windows End Stations



Important

DCV End Station embeds a component called RealVNC Visualization Edition Viewer.

Different versions of VNC on the same system may conflict with each other. It is suggested to uninstall any VNC component from the system before installing DCV.

To install the DCV End Station perform the following steps:

1. download the DCV End Station MSI archive for your system architecture from <https://www.nice-software.com/dcv>
2. navigate to the folder containing the installation files
3. run the installer as Administrator and follow the displayed instructions
4. log off and log in again. This action updates the system configuration so that you can start using the software.



Note

To generate logs of the installation process, open a Command Prompt as Administrator and launch the installer in the following way:

```
msiexec /i <path-to-msi> /log C:\install.log
```

Installing Mac End Stations

To install the DCV End Station, perform the following steps:

1. download from <https://www.nice-software.com/dcv> the DCV End Station DMG archive
2. open the installer DMG
3. drag and drop the DCV End Station application from the DMG to the Applications folder
4. to always open `.vnc/` `.dcv` files using the DCV End Station press `CMD+I` on a `.vnc/` `.dcv` file, select `Open with DCV End Station` and press `Change all`.

To access the `dcv.conf` configuration file right click on the DCV End Station application, select `Show Package Contents` and navigate to the `Contents/Resources/conf` directory.



Note

In case of upgrade the full application contents, including the `dcv.conf` file, will be replaced with the defaults. Remember to backup the configuration in case you changed it.

Installing Linux End Stations

To install the DCV End Station, perform the following steps:

1. download the DCV End Station `.run` archive from <https://www.nice-software.com/dcv>
2. from a terminal go to the directory containing the installation files
3. run the installer as the `root` user and follow the instructions:

```
# sh nice-dcv-endstation-2016.0-xxx.run
```

Using Portable End Stations

DCV Portable End Station is a bundled version of DCV End Station including everything is needed to remotely connect to a DCV Server. It does not require an installation making it easy to use from any client computer.



Note

USB remotization requires to install drivers on the system and will not be available when using a portable end station.

Using Windows Portable End Station

To use the DCV Portable End Station, perform the following steps:

1. download the DCV Portable End Station ZIP archive from <https://www.nice-software.com/dcv>
2. expand the archive to a local folder or a USB stick and open its contents
3. double click on `niceviewer` to start the DCV End Station.
4. optionally, double click on `setup-niceviewer.bat` to always open `.vnc/` `.dcv` files with this DCV Portable End Station

Using Linux Portable End Station

To use the DCV Portable End Station, perform the following steps:

1. download the DCV Portable End Station `tar.gz` archive from <https://www.nice-software.com/dcv>
2. expand the archive to a local folder with:

```
$ tar xvzf nice-dcv-portable-2016.0-xxx.tar.gz
```

3. to launch the DCV End Station run:

```
$ niceviewer
```

Using Mac Portable End Station

The DCV End Station provided in the DMG package can be used directly without installing it.

Advanced Installations

Windows Unattended Installation

To deploy the NICE DCV End Station in a unattended mode you can use the Windows tool `msiexec`, with the following switches:

```
C:\> msiexec /QN /I nice-dcv-endstation-2016.0-xxx-Release.msi
```

To remove the DCV End Station you can use the Windows tool `msiexec`, with the following switches:

```
C:\> msiexec /QN /X nice-dcv-endstation-2016.0-xxx-Release.msi
```

or the WMI command-line utility, executing:

```
C:\> wmic product where name="NICE Desktop Cloud Visualization Endstation"  
call uninstall /nointeractive
```

At the end of the both installation and uninstallation processes, the machine will reboot.

Installing Linux End Stations from RPMs

To install a DCV End Station, perform the following steps:

1. download the DCV End Station RPM archive for your system architecture from <https://www.nice-software.com/dcv>
2. from a terminal go to the directory containing the installation files
3. as the `root` user, install the appropriate packages depending on your distribution:

- RHEL 32 bit:

```
# yum install --nogpgcheck vnc-VE4_6_3-x86_linux.rpm \  
nice-dcv-endstation-2016.0-xxx.i686.rpm
```

- RHEL 64 bit:

```
# yum install --nogpgcheck vnc-VE4_6_3-x64_linux.rpm \  
nice-dcv-endstation-2016.0-xxx.x86_64.rpm
```

- SLES 32 bit:

```
# zypper install -f vnc-VE4_6_3-x86_linux.rpm \  
nice-dcv-endstation-2016.0-xxx.i686.rpm
```

- SLES 64 bit:


```
# zypper install -f vnc-VE4_6_3-x64_linux.rpm \
nice-dcv-endstation-2016.0-xxx.x86_64.rpm
```

Configuring Proxy Support

DCV End Station supports connection to a server behind an HTTP or SOCKS5 proxy.

To enable proxy support you need to specify the proxy type, server address and port in the `dcv.conf` file. Please refer to [Appendix A, `dcv.conf`](#) for more information.

In the following `dcv.conf` excerpt, the end station is configured to connect to an HTTP proxy running on port 3128 (the default port for Squid server) on node `proxy.intranet`.

```
[Endstation]
proxy_port=3128
proxy_server=proxy.intranet
proxy_type=HTTP
```

In the following example, the end station connects to a SOCKS5 proxy running on port 1080 (the default port for Antinat server) on node `proxy.intranet`.

```
[Endstation]
proxy_port=1080
proxy_server=proxy.intranet
proxy_type=SOCKS5
```

To enable the proxy support in RealVNC please refer to the RealVNC documentation available here: <http://www.realvnc.com/products/vnc/documentation/4.6/docs/Chapter3.html> in the "Configuring VNC Viewer before you connect" paragraph.

Configuring Encryption Support

DCV supports encryption of the 3D channel between the end station and the server.

By default, an end station uses the server encryption preferences. This chapter explains how the end station can be configured to tighten the encryption rules if supported by the server.

You can configure DCV End Station using the `dcv.conf`.

The encryption setting in the `[Endstation]` section supports the following values:

- `Server (default)`: uses the server's encryption preferences.
- `PreferOff`: use an unencrypted connection if possible.
- `PreferOn`: use an encrypted connection if possible.
- `AlwaysOn`: do not connect to servers that do not use at least 128-bit encrypted connections.
- `AlwaysMaximum`: do not connect to servers that do not use 256-bits encrypted connections.

By default RealVNC communication between client and host computer is encrypted using industry-standard 128-bit AES. For more information on how to request a different encryption option, see

Chapter 2 "Getting Connected", "Step 4: Request an encryption option", <http://www.realvnc.com/products/vnc/documentation/4.6/docs/ae1014953.html>.

For more information on RealVNC security, see Chapter 7 "Securing Connections", <http://www.realvnc.com/products/vnc/documentation/4.6/docs/Chapter7.html>.

Using Desktop Cloud Visualization

This chapter describes how to run remote OpenGL applications with DCV and use the DCV components both on application hosts and end stations, both on Linux and Windows.

Using DCV on an End Station

Using DCV on a Windows End Station

To start the end station on Windows, click `Start > All Programs > NICE > DCV > DCV Viewer`.

Using DCV on a Mac End Station

Go to the Applications folder and launch the `DCV End Station`.

Using DCV on a Linux End Station

To start the end station on Linux, click on the programs menu and select `Internet > VNC Viewer`.

To launch the end station from command line, use the `vncviewer` command.

This is the syntax of the `vncviewer` command is:

```
vncviewer [<servername[:portnum]>] [<other viewer options>] [-h]
```

Refer to the RealVNC documentation for more information about the `vncviewer` command.



Important

Make sure that the `vncviewer` program used is the one provided by RealVNC Visualization Edition, the DCV component will not be loaded by other VNC clients.

After connection to a remote server a tool-tip will pop-up showing the availability of the DCV Console as in Figure 3.1, “Example of a Linux Desktop at connection”.

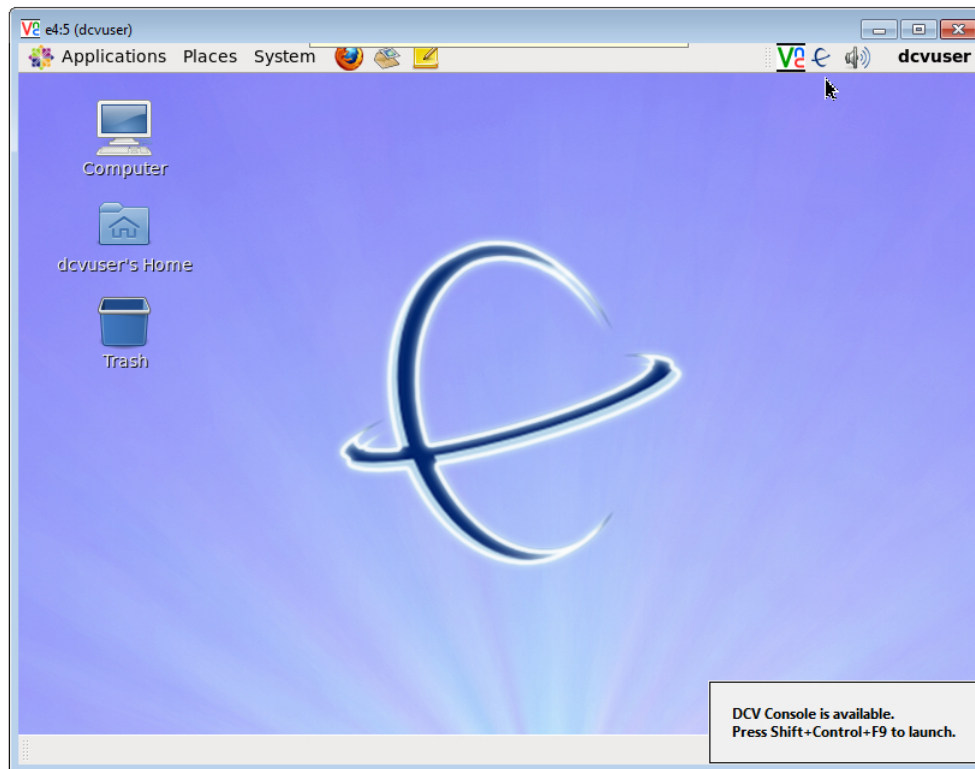


Figure 3.1. Example of a Linux Desktop at connection

Using DCV from a plain VNC client

On some clients it may be not possible to install the DCV End Station but a plain RealVNC Viewer may be available.

Starting from DCV 2012.2, the DCV Server automatically detects plain VNC clients (without DCV End Station) and sends the 3D images on their VNC connection. The DCV Config tool reports the number of currently connected plain VNC clients.

This feature is useful to enable access from iOS devices such as iPhones or iPads using the standard RealVNC client available from the Apple App Store.



Important

When connecting from a DCV End Station double check the DCV Console tool-tip shows in the bottom right corner. In case it does not show, the DCV End Station may not be working properly and your connection may be not optimized.

Using DCV Console

The DCV Console provides a graphical interface to control and monitor the DCV connection and to tweak its settings. The same interface is available both on Windows and Linux end stations.

Figure 3.2, “DCV Console main window” shows the main window of the DCV Console.

The DCV Console can be started by pressing the default <Control><Shift>F9 keystroke combination in the VNC Viewer window. The keystroke combination can be configured in the `dcv.conf` file by setting the `console_shortcut` property. Please refer to [Appendix A, `dcv.conf`](#) for more information.



Note

The keystroke combination is <CMD><Fn>F9 it is not possible to configure the keystroke combination on Mac.

Normally a user just needs to operate on the quality slider to adapt DCV to his current needs.

An in depth description of the available settings follows.



Figure 3.2. DCV Console main window

Quality slider

Controls the balance between image compression and image quality. Choosing a high value improves the quality of the images, but requires more data to be sent for each image, which can reduce responsiveness or lower the frame rate. Conversely a low value can increase performance at the expense of image quality.

By changing the Quality slider you may notice a change in the bandwidth usage and/or FPS. Adjust it to find the value which works best in your environment. The best value may change depending on the application, the image complexity and the network conditions.

Starting from DCV 2013, you can easily switch to lossless compression by setting the quality slider to Best Quality as shown in [Figure 3.3, “DCV Console main window showing lossless compression”](#). Due to the higher bandwidth usage of lossless compression, this setting is suggested only in LAN environments.

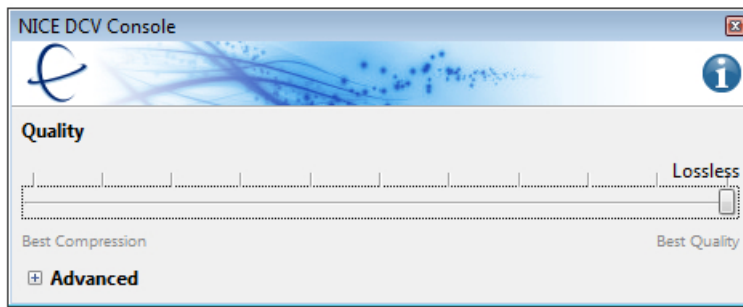


Figure 3.3. DCV Console main window showing lossless compression

About

Clicking on the information icon on the top right shows the About dialog which contains information on the DCV End Station version, the DCV Server version and hostname, the DCV protocol, the codecs in use, the connection type (Direct or via a Proxy) and the used encryption.

These information are available after the first OpenGL is launched.

Figure 3.4, “DCV Console About window” shows the About window of the DCV Console.

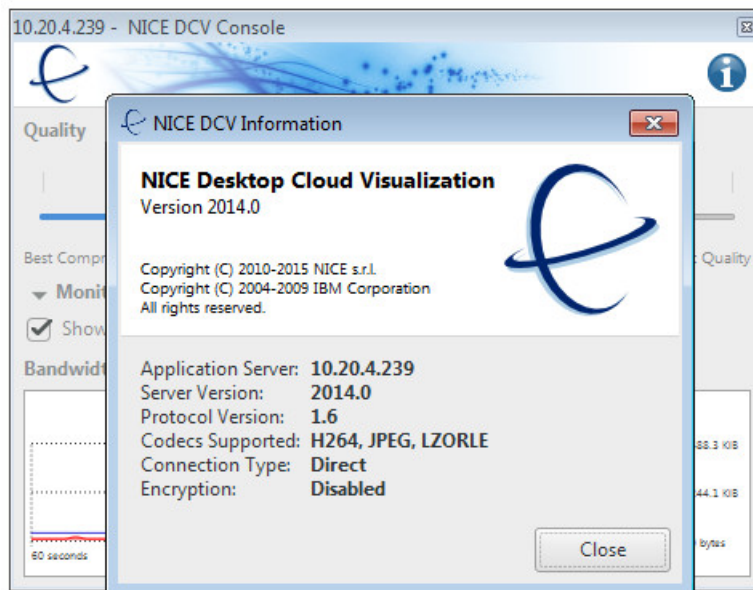


Figure 3.4. DCV Console About window

Advanced settings

By clicking on the Advanced expander you can access more settings and check network connection statistics.

Figure 3.5, “DCV Console advanced window” shows the advanced settings.

Show frames per second

If checked shows the frames per second received by the end station. Each 3D window will display an overlay FPS number in the bottom right corner.

Bandwidth usage

DCV Console provides statistics on the network link condition between the server and the end station.

A sliding graph shows the network bandwidth usage by DCV in the last minute. The red line shows the instant value, the blue line is the average in the last minute.

In addition to the current value and the average value, the graph box also shows the peak bandwidth from the time the end station connected.

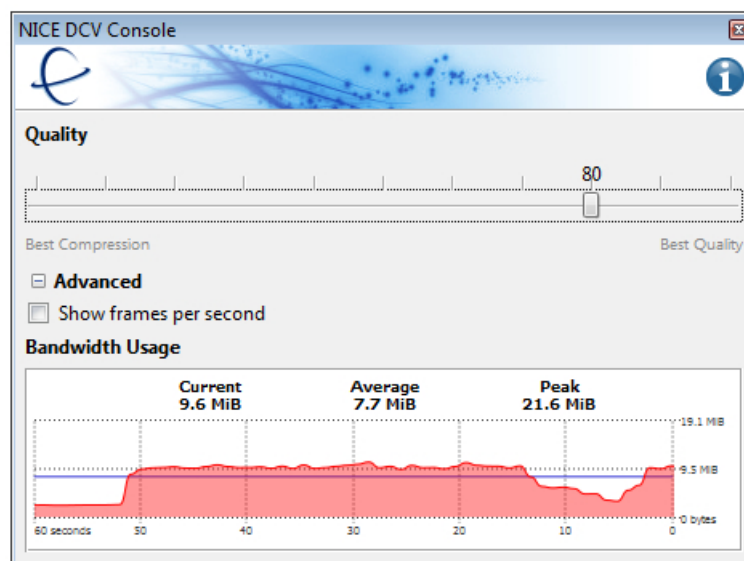


Figure 3.5. DCV Console advanced window

Using DCV in a Remote Desktop Connection

To start using DCV in a Remote Desktop Connection you just need to use a Remote Desktop Client and connect to the remote Application Host. Figure 3.6, “Remote Desktop Client connection dialog” shows the default connection dialog of the Remote Desktop Client.

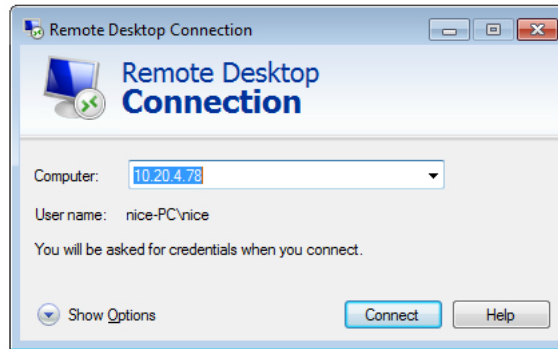


Figure 3.6. Remote Desktop Client connection dialog

When connected to a remote desktop you can launch the DCV Test application to check Desktop Cloud Visualization is working.

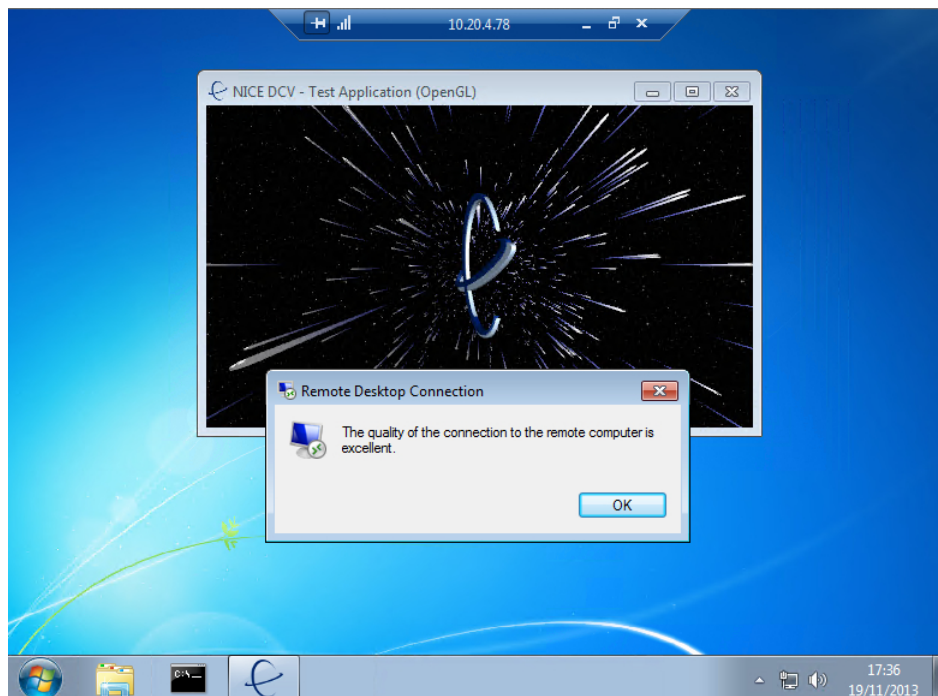


Figure 3.7. DCV Test application running in a Remote Desktop Connection

You can check the RD Client version and supported RD Protocol by opening the About dialog.



Figure 3.8. Remote Desktop Client about dialog

For optimal performance, both the client and server must support Remote Desktop Protocol 8.0 or above.

If RDP 8.0 is enabled, the connection quality button will be displayed in the connection bar.

Click the connection quality button to open an information dialog box that resembles the following. See Figure 3.9, “Check RDP 8.0 is enabled”.

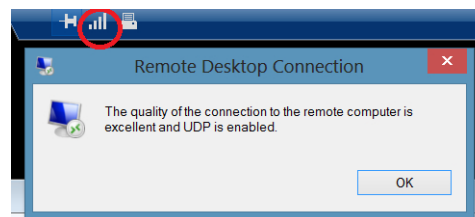


Figure 3.9. Check RDP 8.0 is enabled



Note

When using DCV inside a Remote Desktop Connection the delivery of 3D images is delegated to RDP while Desktop Cloud Visualization takes care of the GPU virtualization for OpenGL calls.

When using Remote Desktop Client, the DCV Console is not available and Quality settings is delegated to the RDP configuration.

Remote Desktop Client is available on many platforms. Figure 3.10, “Microsoft Remote Desktop for Android” shows DCV Test application running inside a Microsoft Remote Desktop for Android.

Check on Microsoft web site for more information on how to download the appropriate client for your platform.



Figure 3.10. Microsoft Remote Desktop for Android

Using DCV on an Application host

On an application host working with DCV revolves around the following activities:

- administering DCV (reserved to users with administrator privileges);
- enabling and configuring a DCV session;
- executing OpenGL applications.

To achieve these tasks DCV provides both command line and GUI utilities.

Command Line Tools

`dcdadmin.exe` (Windows with External Rendering Server) / `dcdadmin` (Linux)

The `dcdadmin` command allows an administrator to manage DCV on a system. By running `dcdadmin enable` the DCV libraries will be deployed on the system while issuing `dcdadmin disable` will undeploy them. The DCV libraries are deployed automatically by the installer, so the use of the `dcdadmin` command is required only when upgrading the graphic drivers or when performing system administration tasks. The `dcdadmin status` command allows a user to inspect if DCV components are deployed on the system.

`dcd.exe` (Windows with External Rendering Server) / `dcd` (Linux)

The `dcd` command allows a user to switch on and off DCV for the current desktop session. When `dcd on` is issued, DCV becomes active and all OpenGL applications subsequently started will make use of DCV. In a similar way DCV can be deactivated with the `dcd off` command. OpenGL applications making already use of DCV will not be impacted by the `dcd off` command. Finally `dcd status` provides information about the activation state of DCV on the current desktop.

On Linux the `dcd` command optionally accepts a VNC X display number as an argument so that the action is applied to the specified display instead of the current one. For instance `dcd on :2.0` will activate DCV on the VNC X display :2.0.

`dcdtest.exe` (Windows) / `dcdtest` (Linux)

The `dcdtest` command starts a simple test application showing NICE and DCV logos rotating over a star field.

Press spacebar to pause the animation.

`dcdscrn` (Linux)

The `dcdscrn` (*DCV SmartCard Run*) command allows to run applications which use the standard PC/SC interface so that they can use the DCV Smart Card Remotization.

To launch an application with `dcdscrn` type:

```
dcdscrn <app> [app arguments]
```

For more information please refer to the the section called “Smart Card Remotization”.

`dcdstartx` (Linux)

The `dcdstartx` (*DCV Start X*) command allows to start a minimal X11 server showing the DCV logo. It provides a minimal setup, preventing user logins on the graphic console and takes care of enabling access to local users by calling the `dcdxgrantaccess` command.

For more information please refer to the the section called “Grant Local Connections to the 3D Rendering Display”.

The DCV Config utility

The DCV Config utility provides a graphical interface to check DCV status. On Linux and on Windows with External Rendering Server it allows to switch DCV on and off for the current session and to list the applications remotized through DCV.

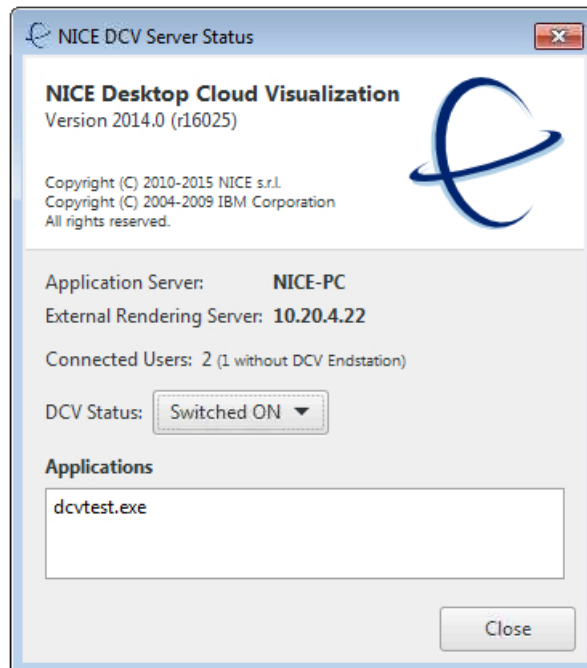


Figure 3.11. DCV Config showing External Rendering Server

It also reports the number of connected users and how many of them are not using a DCV End Station.

On Windows the DCV Config utility starts up automatically at login if DCV is enabled. On Linux the DCV Config utility is started automatically the first time `dcv on` is executed (this usually occurs while executing the `xstartup` file).

The DCV Config utility is accessible by clicking on the DCV icon in the system tray area. On Linux, in case the tray area is not available, the DCV Config utility will show its main window at startup.

To avoid the DCV Config utility to be shown at session startup and to be added to the tray area, use `dcv on --nowin`.

The `DCV Server` shows the hostname of the server DCV is running on, and on Windows with External Rendering Server, the default External Rendering Server.

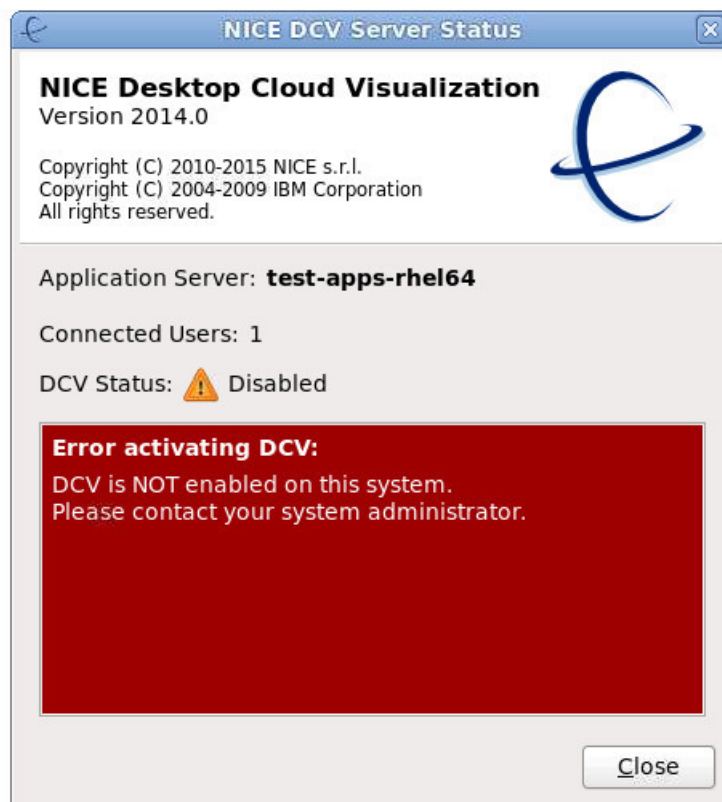


Figure 3.12. DCV Config diagnostic messages

The DCV Config utility also provides the following diagnostic information in case of problems:

- DCV is installed but not enabled on the system (see Figure 3.12, “DCV Config diagnostic messages”);
- DCV cannot find a valid license;
- DCV cannot start because a system problem occurred.

Running an Application with DCV on Linux

This section describes the different ways to run applications with DCV:

Display Isolation Mode

To run an application with DCV in *Display Isolation Mode*, a standalone VNC server must be started on the application host:

1. log in to the application host and run the following command:

```
vncserver
```

for more information on how to use the `vncserver` command, refer to

```
man vncserver
```

The VNC server will execute the `xstartup` file to setup an isolated desktop session.

The DCV installer adds automatically the `dcv on` command at the end of the `/etc/vnc/xstartup` script so that the new session is ready to use OpenGL applications with DCV.

Once the VNC server is up and running you can connect to it using the `vncviewer` and then to start an application you can perform the same steps described for the [the section called “Native Display Mode \(System Console\)”](#).

Native Display Mode (System Console)

To run an application with DCV in *Native Display mode*, the VNC module must be loaded in the X server, so that it is possible to connect to the host using the `vncviewer` client.

Once the VNC session is active, to start an OpenGL application making use of DCV, you can follow these steps:

1. check in the DCV Config that DCV is active. Click on the `Switch on` button if DCV is not already active;
2. start the OpenGL application;
3. after few seconds the new application will appear in the DCV Config application list.

Running an Application with DCV Rendering Server

To run an application with DCV Rendering Server, ensure the RealVNC Visualization Edition Server is running on the application host machine, so that it is possible to connect to the host using the `vncviewer` client.

Once the VNC session is active, to start an OpenGL application making use of DCV, you can follow these steps:

1. check in the DCV Config that DCV is active. Click on the `Switch on` button if DCV is not already active;
2. start the OpenGL application;
3. after few seconds the new application will appear in the DCV Config application list.

Appendix A. dcv.conf

You can customize DCV via the `dcv.conf` configuration file located in the `conf` directory under the DCV installation.

This file specifies rules for different DCV components.

It is a text file with a basic structure composed of "sections" and "properties" based on the INI format.

The section name appears on a line by itself, in square brackets. All properties after the section declaration are associated with that section. There is no explicit "end of section" delimiter; sections end at the next section declaration, or the end of the file. Sections can not be nested.

Each property has a key and a value, delimited by an equals sign. The key appears to the left of the equals sign. Properties may be grouped into arbitrarily named sections.

Lines starting with a hash mark or semi-colon and empty lines are ignored.

Each entry in `dcv.conf` has one of the following forms:

- `[section]`
- `KEY=VALUE`

[Application]

List of `[Application]` settings.

external_rendering_server

Syntax

```
external_rendering_server = host[:port]
```

Description

Specifies the hostname or IP address followed by the port of the external rendering server that the OpenGL applications must use to delegate 3D rendering.

The value of the `external_rendering_server` parameter can be overridden by setting the environment variable `DCV_RENDERING_SERVER`.

Valid Values

Hostname or IP address followed by the port of a DCV rendering server. Port may be omitted.

Default Value

No external rendering server. In case just host is specified the default port is 2007.



Note

The communication between OpenGL applications and the DCV rendering server must go through a high bandwidth/low latency network connection.

local_display_list

Syntax

```
local_display_list = :display[.screen] ...
```

Description

List of local 3D accelerated X displays and screens used by DCV for OpenGL rendering. The value of the `local_display_list` parameter can be overridden by setting the environment variable `RVN_LOCAL_DISPLAY`.

Valid Values

Space separated list of local 3D accelerated X displays.

Default Value

:0.0

local_multisample

Syntax

```
local_multisample = integer
```

Description

Specifies the number of samples to request. Use 0 to turn off multisampling, use -1 to choose the highest supported value.

The value of the `local_multisample` parameter can be overridden by setting the environment variable `RVN_LOCAL_MULTISAMPLE`.

Valid Values

- -1: highest supported multisampling;
- 0: turn multisampling off;
- greater than 0: the multisampling number to use.

Default Value

0



Note

Increasing the number of multisampling requires more computing power and decreases OpenGL rendering performances.

print_dcv_version

Syntax

```
print_dcv_version = boolean
```

Description

Specifies if DCV should print a message with the DCV version when an OpenGL application starts.

The value of the `print_dcv_version` parameter can be overridden by setting the environment variable `DCV_PRINT_VERSION`.

Valid Values

true | false

Default Value

false

target_fps

Syntax

`target_fps = integer`

Description

Specifies the frame rate at which frames are grabbed and sent on the network. The Target FPS can be reached or not depending on:

- Whether or not the application is generating enough frames on server side;
- The time spent to generate compressed the frames;
- The available bandwidth permits to send the frames;
- The end station is able to decompressed the frames.

The value of the `target_fps` parameter can be overridden by setting the environment variable `RVN_TARGET_FPS`.

Valid Values

- -1 , 0: disables the limit;
- greater than 0: sets the limit to that number.

Default Value

25

use_nvidia_grid

Syntax

`use_nvidia_grid = boolean`

Description

Specifies to use the NVIDIA GRID technology.

The value of the `use_nvidia_grid` parameter can be overridden by setting the environment variable `DCV_USE_NVIDIA_GRID`.

Valid Values

true | false

Default Value

true

use_sample_queries

Syntax

`use_sample_queries = boolean`

Description

Specifies whether DCV should use GL sample queries or not.

The value of the `use_sample_queries` parameter can be overridden by setting the environment variable `RVN_USE_SAMPLE_QUERIES`.

Valid Values

true | false

Default Value

true

[Endstation]

List of [Endstation] settings.

console_shortcut

Syntax

`console_shortcut = should be something like "<Control><Shift>F9"`

Description

Sets the shortcut to launch the console

The value of the `console_shortcut` parameter can be overridden by setting the environment variable `RVN_CONSOLE_SHORTCUT`.

Valid Values

Valid modifiers are Control, Shift and Alt. These values must be contained between < and > characters.

Default Value

<Control><Shift>F9



Note

See that in order to set these characters in an environment variable you need to escape them with ^ on Windows and with \ on Linux.

enable_console

Syntax

`enable_console = boolean`

Description

Specifies whether to load and use the client-side console.

The value of the `enable_console` parameter can be overridden by setting the environment variable `DCV_ENDSTATION_ENABLE_CONSOLE`.

Valid Values

true | false

Default Value

true

encryption

Syntax

`encryption = server | alwaysmaximum | alwayson | preferon | preferoff`

Description

The endstation encryption mode.

The value of the `encryption` parameter can be overridden by setting the environment variable `DCV_ENDSTATION_ENCRYPTION`.

Valid Values

`server` uses the server's encryption preferences `alwaysmaximum` to use always encryption with a 256 bits encryption key `alwayson` to connect to servers that use at least 128-bit encrypted connections `preferon` to use 128 bits encryption if the endstation allows the server to decide or if sets `preferon` `preferoff` to not use encryption

Default Value

server

proxy_port

Syntax

`proxy_port = integer`

Description

Specify the TCP port number of the HTTP or SOCKS5 proxy server that the DCV end station must use to connect to the DCV server.

The value of the `proxy_port` parameter can be overridden by setting the environment variable `DCV_PROXY_PORT`.

Valid Values

A valid TCP port number.

Default Value

8123

proxy_server

Syntax

`proxy_server = hostname | IP address`

Description

Specify the hostname or IP address of the HTTP or SOCKS5 proxy server that the DCV endstation must use to connect to the DCV server.

The value of the `proxy_server` parameter can be overridden by setting the environment variable `DCV_PROXY_SERVER`.

Valid Values

Hostname or IP address of a HTTP or SOCKS5 proxy server. Empty string to directly connect to the DCV server without using any proxy server.

Default Value

No proxy server.



Note

Proxy servers that require authentication are not supported. This may be the subject of a future release.

proxy_type

Syntax

```
proxy_type = HTTP | SOCKS5
```

Description

Specify the type of proxy server that the DCV end station must use to connect to the DCV server.

The value of the `proxy_type` parameter can be overridden by setting the environment variable `DCV_PROXY_TYPE`.

Valid Values

HTTP to use a HTTP proxy or SOCKS5 to use a SOCKS5 proxy

Default Value

HTTP

tip_timeout

Syntax

```
tip_timeout = milliseconds
```

Description

Sets the time the console shortcut tip will be present.

The value of the `tip_timeout` parameter can be overridden by setting the environment variable `DCV_ENDSTATION_TIP_TIMEOUT`.

Valid Values

Any positive integer.

Default Value

10000

[License]

List of [License] settings.

license_file

Syntax

```
license_file = filename | port@host
```

Description

Specifies a demo license file or RepriseLM-based license used by DCV.

The value of the `license_file` parameter can be overridden by setting the environment variable `DCV_LICENSE_FILE`.

Valid Values

The value for *license_file* can be either of the following:

- The full path name to the license file.

Linux example:

```
license_file = /opt/nice/dcv/license/license.lic
```

Windows example:

```
license_file = C:\Program Files\DCV\license\license.lic
```

- For a permanent license, the name of the license server host and TCP port number used by the `rlm` daemon, in the format *port@host*.

For example:

```
license_file = 5053@licserver
```

The port number must be the same as that specified in the `HOST` line of the license file.

Default Value

If you installed DCV with a default installation, the license file is expected to be in the `license` directory under DCV installation path.

use_platform_product

Syntax

```
use_platform_product = ...
```

Description

Force the use of the old platform-specific products (`dcv_linux`, `dcv_windows`, `dcv_windows_vm`)

The value of the `use_platform_product` parameter can be overridden by setting the environment variable `DCV_LICENSE_USE_PLATFORM_PRODUCT`.

Valid Values

...

Default Value

false

[Logging]

List of [Logging] settings.

log_dir

Syntax

`log_dir = directory`

Description

Specifies the fullpath of the directory where DCV will create log files
The value of the `log_dir` parameter can be overridden by setting the environment variable `DCV_LOG_DIR`.

Valid Values

A path where the user running the DCV server can write.

log_file

Syntax

`log_file = filepath`

Description

Specifies the full file path prefix used by DCV to create log file to which DCV messages are logged into. Each log file will be created in the form: *log_file.pid*, where *pid* is an integer identifying the OpenGL process using DCV.

For example:

```
log_file = /tmp/dcv.log
```

will make DCV create log files as:

```
/tmp/dcv.log.1234
```

The value of the `log_file` parameter can be overridden by setting the environment variable `DCV_LOG_FILE`.

Valid Values

A file path where the user running the DCV application can write.

log_level

Syntax

log_level = *string*

Description

Specifies the logging level of DCV messages.

For example:

```
log_level = INFO
```

The value of the log_level parameter can be overridden by setting the environment variable DCV_LOG_LEVEL.

Valid Values

The log levels in order from highest to lowest are:

- error
- warning
- info
- debug

The most important DCV messages are at the error or warning level. Messages at the info and debug level are only useful for debugging.

Default Value

warning

[Remotization]

List of [Remotization] settings.

allow_udp

Syntax

allow_udp = ...

Description

Whether to allow to use UDP.

The value of the allow_udp parameter can be overridden by setting the environment variable DCV_ALLOW_UDP.

Valid Values

true | false

Default Value

false

blacklist_vchannels

Syntax

`blacklist_vchannels = :backends`

Description

List of DCV Virtual Channel backends to blacklist. When a vchannel is blacklisted its backend is not started and end stations will not be able to use it.

The value of the `blacklist_vchannels` parameter can be overridden by setting the environment variable `DCV_BLACKLIST_VCHANNELS`.

Valid Values

List of backends separated by ":" on Linux and ";" on Windows

codec_qu

Syntax

`codec_qu = string`

Description

Specifies the codec DCV should use for static image quality enhancement.

See also the `enable_update_quality` setting for enabling the static image quality enhancement.

The value of the `codec_qu` parameter can be overridden by setting the environment variable `RVN_CODEC_QU`.

Valid Values

Supported codecs:

- `jpeg`: lossy JPEG compression;
- `rle`: lossless run-length encoding.
- `zrle`: lossless run-length encoding.
- `lzorle`: lossless run-length encoding.

Default Value

`lzorle`

codec_tcp

Syntax

`codec_tcp = string`

Description

Specifies the codec DCV should use on TCP connections.

The value of the `codec_tcp` parameter can be overridden by setting the environment variable `RVN_CODEC_TCP`.

Valid Values

Supported codecs:

- h264: lossy H264 compression;
- jpeg: lossy JPEG compression;
- rle: lossless run-length encoding.
- zrle: lossless run-length encoding.
- lzo: lossless run-length encoding.

Default Value

h264

default_proxy_port

Syntax

`default_proxy_port = integer`

Description

Specify the TCP port number of the HTTP or SOCKS5 proxy server that the DCV end station will use as a default to connect to the DCV server. The proxy properties on the end station will override these defaults.

The value of the `default_proxy_port` parameter can be overridden by setting the environment variable `DCV_DEFAULT_PROXY_PORT`.

Valid Values

A valid TCP port number.

Default Value

8123

default_proxy_server

Syntax

`default_proxy_server = hostname | IP address`

Description

Specify the hostname or IP address of the HTTP or SOCKS5 proxy server that the DCV endstations will use as a default to connect to the DCV server. The proxy properties on the end station will override these defaults.

The value of the `default_proxy_server` parameter can be overridden by setting the environment variable `DCV_DEFAULT_PROXY_SERVER`.

Valid Values

Hostname or IP address of a HTTP or SOCKS5 proxy server. Do not specify this setting if you want to directly connect to the DCV server without using any proxy server.

Default Value

No proxy server.

**Note**

Proxy servers that require authentication are not supported. This may be the subject of a future release.

default_proxy_type

Syntax

```
default_proxy_type = HTTP | SOCKS5
```

Description

Specify the type of proxy server that the DCV end station will use as a default to connect to the DCV server. The proxy properties on the end station will override these defaults.

The value of the `default_proxy_type` parameter can be overridden by setting the environment variable `DCV_DEFAULT_PROXY_TYPE`.

Valid Values

HTTP to use a HTTP proxy or SOCKS5 to use a SOCKS5 proxy

Default Value

HTTP

enable_quality_update

Syntax

```
enable_quality_update = boolean
```

Description

Specify if DCV must send a higher quality image (the *quality update*) when the 3D image is not changing.

See also the `quality_update_quality` setting for tweaking the quality level of the static images.

The value of the `enable_quality_update` parameter can be overridden by setting the environment variable `RVN_ENABLE_QUALITY_UPDATE`.

Valid Values

true | false

Default Value

true

encryption

Syntax

```
encryption = alwaysmaximum | alwayson | preferon | preferoff
```

Description

The server application encryption mode.

The value of the `encryption` parameter can be overridden by setting the environment variable `DCV_ENCRYPTION`.

Valid Values

alwaysmaximum to use always encryption with a 256 bits encryption key alwayson to use always encryption with a 128 bits encryption key preferon to use 128 bits encryption if the endstation allows the server to decide or if sets preferon preferoff to not use encryption

Default Value

preferoff

host

Syntax

`host = host`

Description

Specifies the hostname or IP address of the external rendering server that the OpenGL applications must use to delegate 3D rendering. In case it is specified it must be a valid address reachable from the end station.

The value of the `host` parameter can be overridden by setting the environment variable `RVN_RENDERING_HOST`.

Valid Values

Hostname or IP address of a DCV rendering server.

image_quality

Syntax

`image_quality = integer`

Description

Specify the quality level of dynamic images when using TCP connections. Higher values correspond to higher image quality and more data transfer. Lower values reduce quality and reduce bandwidth usage.

The value of the `image_quality` parameter can be overridden by setting the environment variable `RVN_IMAGE_QUALITY`.

Valid Values

1-100

Default Value

80

quality_update_interval

Syntax

`quality_update_interval = integer`

Description

Interval in milliseconds before a quality update is sent after a dynamic quality frame was sent.

The value of the `quality_update_interval` parameter can be overridden by setting the environment variable `RVN_QUALITY_UPDATE_INTERVAL`.

Valid Values

Values greater than zero are valid.

Default Value

300

quality_update_quality

Syntax

`quality_update_quality = integer`

Description

Specify the quality level of static images. Higher values correspond to higher image quality and more data transfer. Lower values reduce quality and reduce bandwidth usage. See also the `enable_update_quality` setting for enabling the static image quality enhancement.

The value of the `quality_update_quality` parameter can be overridden by setting the environment variable `RVN_QUALITY_UPDATE_QUALITY`.

Valid Values

1-100

Default Value

95

tcp_max_port

Syntax

`tcp_max_port = integer`

Description

End of the range of TCP ports used by DCV to listen to end stations.

The value of the `tcp_max_port` parameter can be overridden by setting the environment variable `RVN_MINI_MAX_PORT`.

Valid Values

A valid TCP port number greater than `tcp_start_port`.

Default Value

7399

tcp_start_port

Syntax

`tcp_start_port = integer`

Description

Start of the range of TCP ports used by DCV to listen to end stations.

The value of the `tcp_start_port` parameter can be overridden by setting the environment variable `RVN_MINI_START_PORT`.

Valid Values

A valid TCP port number.

Default Value

7300

udp_quality

Syntax

`udp_quality = integer`

Description

Specify the quality level of dynamic images when using UDP connections. Higher values correspond to higher image quality and more data transfer. Lower values reduce quality and reduce bandwidth usage.

The value of the `udp_quality` parameter can be overridden by setting the environment variable `RVN_UDP_QUALITY`.

Valid Values

1-100

Default Value

80

udp_start_port

Syntax

`udp_start_port = integer`

Description

Start of the range of UDP ports used by DCV to listen to end stations.

The value of the `udp_start_port` parameter can be overridden by setting the environment variable `RVN_UDP_START_PORT`.

Valid Values

A valid UDP port number.

Default Value

7300

use_nvidia_hw_encoder

Syntax

`use_nvidia_hw_encoder = boolean`

Description

Specifies to use the NVIDIA HW Encoder.

The value of the `use_nvidia_hw_encoder` parameter can be overridden by setting the environment variable `DCV_USE_NVIDIA_HW_ENCODER`.

Valid Values

true | false

Default Value

true

use_udp

Syntax

`use_udp = integer`

Description

Specify whether DCV should use TCP or UDP connections.

The value of the `use_udp` parameter can be overridden by setting the environment variable `RVN_USE_UDP`.

Valid Values

0 | 1 | 2

Default Value

0

[RenderingServer]

List of [RenderingServer] settings.

agent_max_port

Syntax

`agent_max_port = integer`

Description

Maximum value of the ports interval used by the rendering agents.

The value of the `agent_max_port` parameter can be overridden by setting the environment variable `SVN_SOCKETS_MAX_PORT`.

Valid Values

A TCP port number.

Default Value

60000

agent_start_port

Syntax

`agent_start_port = integer`

Description

Start of the ports interval used by the rendering agents.

The value of the `agent_start_port` parameter can be overridden by setting the environment variable `SVN_SOCKETS_PORT`.

Valid Values

A TCP port number.

Default Value

34567

host**Syntax**

`host = IP address`

Description

Specifies the network interface to which the rendering server should bind.

The value of the `host` parameter can be overridden by setting the environment variable `DCV_RENDERING_SERVER_HOST`.

Valid Values

The rendering host IP address to which application host are connected via a high bandwidth/low latency network.

local_display_list**Syntax**

`local_display_list = :display[.screen] ...`

Description

List of local 3D accelerated X displays and screens used by DCV for OpenGL rendering.

The value of the `local_display_list` parameter can be overridden by setting the environment variable `DCV_LOCAL_DISPLAY_LIST`.

Valid Values

Space separated list of local 3D accelerated X displays.

Default Value

`:0.0`

**Note**

Access to the selected displays must be granted to the user running the rendering server.

port**Syntax**

`port = integer`

Description

Specifies the TCP port where the rendering server should listen on.

The value of the `port` parameter can be overridden by setting the environment variable `DCV_RENDERING_SERVER_PORT`.

Valid Values

A free TCP port number above 1024.

Default Value

2007

user**Syntax**

`user = operating system username`

Description

Specifies the user account the rendering server is running as.

The value of the `user` parameter can be overridden by setting the environment variable `DCV_RENDERING_SERVER_USER`.

Valid Values

Any unprivileged operating system username.

Default Value

`dcv`

[Wrapper]

List of [Wrapper] settings.

disable_driver_unload**Syntax**

`disable_driver_unload = boolean`

Description

Specifies if explicit unload of the driver's GL library must be disabled.

The value of the `disable_driver_unload` parameter can be overridden by setting the environment variable `DCV_DISABLE_DRIVER_UNLOAD`.

Valid Values

`true` | `false`

Default Value

`true` - Unload GL library