NICE DCV

User Guide

NICE DCV: User Guide

Table of Contents

Getting Started with NICE DCV

NICE DCV is a high-performance remote display protocol. It lets you securely deliver remote desktops and application streaming from any cloud or data center to any device, over varying network conditions. By using NICE DCV with Amazon EC2, you can run graphics-intensive applications remotely on Amazon EC2 instances. You can then stream the results to more modest client machines, which eliminates the need for expensive dedicated workstations.

To use NICE DCV, install the NICE DCV server software on a server. The NICE DCV server software is used to create a secure session. You install and run your applications on the server. The server uses its hardware to perform the high-performance processing that the installed applications require. Your users access the application by remotely connecting to the session using a NICE DCV client application. When the connection is established, the NICE DCV server software compresses the visual output of the application and streams it back to the client application in an encrypted pixel stream. The client application receives the compressed pixel stream, decrypts it, and then outputs it to the local display.

Contents

Step 1: Get the NICE DCV Session Information

After the NICE DCV session is running on the NICE DCV server, you must have specific information to connect to it. Contact your NICE DCV administrator if you don't have the following information:

- The IP address or host name of the NICE DCV server
- The port that the NICE DCV server is configured to communicate over. By default, port 8443 is used by the NICE DCV server.
- The session ID
- A user name and password to connect to the NICE DCV host server

Step 2: Choose a NICE DCV Client

Next, choose the NICE DCV client that best meets your needs. NICE DCV offers the following clients:

- A Windows client
- A web browser client
- A Linux client
- A macOS client

For more information about the available clients, see NICE DCV clients (p. 2).

After you choose a NICE DCV client, you can use it to connect to, and interact with the NICE DCV session. For more information about using the NICE DCV clients to interact with sessions, see Using NICE DCV (p. 9).

NICE DCV clients

NICE DCV offers a Windows client, Linux client, web browser client, and macOS client. The clients offer similar feature sets, but there are some differences. Choose the NICE DCV client that meets your specific requirements.

Topics

Requirements

To use NICE DCV, ensure that the client computers meet the following minimum requirements. Bear in mind that your experience is dependent on the number of pixels that are streamed from the NICE DCV server to the NICE DCV client.

	Windows client	Web browser client	Linux client	macOS client
Software	The Windows client is supported on 32-bit and 64-bit versions of the following operating systems: • Windows 8.1 • Windows 10 • Windows 11 The client requires the following additional software: • .NET Framework 4.6.2 • Microsoft Visual C ++ Redistributable for Visual Studio. For more information and download instructions, see the Microsoft Support website.	The web browser client is supported on the latest three major versions of the following browsers, across all major desktop operating systems (Windows, macOS, and Linux): • Mozilla Firefox • Google Chrome • Microsoft Edge • Apple Safari The web browser client also requires WebGL and asm.js. **Note** The web browser client isn't supported on mobile operating systems, such as	The Linux client is supported on the following modern Linux operating systems: • RHEL 7.x and CentOS 7.x • RHEL 8.x and CentOS 8.x • SUSE Linux Enterprise 15.x • Ubuntu 18.04 and 20.04	macOS clients with Intel processors require macOS Mojave (10.14) or later. macOS clients with Apple M1 processors require macOS Big Sur (11).

	Windows client	Web browser client	Linux client	macOS client
		Android and iOS.		
Network	The client must connect to the NICE DCV server, and it must communicate over the required port. By default, this is port 8443.			

For more information about the NICE DCV server requirements, see NICE DCV server requirements in the *NICE DCV Administrator Guide*.

Supported features

The following table compares the features that are supported by the NICE DCV clients.

Feature	Windows client (p. 4)	Web browser client (p. 5)	Linux client (p. 6)	macOS client (p. 7)
Connect to Windows NICE DCV servers (p. 9)	✓	✓	✓	✓
Connect to Linux NICE DCV servers (p. 9)	✓	✓	✓	✓
QUIC UDP transport protocol (p. 9)	✓	✗	✓	✓
Manage streaming modes (p. 13)	✓	✓	✓	✓
Transfer files (p. 17)	✓	✓	✓	✓
Print from sessions (p. 19)	✓	✓ [1]	✓ [1]	✓ [1]
Copy and paste (p. 20)	✓	✓	✓	✓
Smart card support (p. 21)	✓	✗	✓	✓
USB remotization support (p. 25)	✓ (installable client)	✗	✗	✗
Connection file support (p. 33)	✓	✗	✓	✓
Stereo 2.0 audio playback	✓	✓	✓	✓
Surround sound audio playback	✓ (up to 7.1)	✗	✓ (up to 5.1)	✗
Stereo 2.0 audio recording	✓	✓	✓	✓
Touchscreen support	✓ (Windows 8.1 and later)	✓ [2]	✓	✗
Stylus support (on Linux and Windows 10 and Server 2019 servers)	✓ (Windows 10 and later)	✓ [3]	✓	✓
Gamepad support (on Windows 10 and Server 2016 and later servers)	✓	✗	✗	✗

Feature	Windows client (p. 4)	Web browser client (p. 5)	Linux client (p. 6)	macOS client (p. 7)
Multiple monitor support (p. 23)	✓	✓	✓	✓
Webcam support (p. 26)	✓	✓ [4]	✓	✓

[1] These clients support printing to a file only. They don't support printing to a local printer.

[2] Supported by Firefox, Edge, and Google Chrome.

[3] Supported in Chromium-based browsers only. This includes Google Chrome and Microsoft Edge version 79 and later. Tilt and pressure events aren't supported in other browsers.

[4] Supported in Chromium-based browsers only. This includes Google Chrome and Microsoft Edge version 79 and later. This doesn't include Firefox and Safari.

For more information about the NICE DCV server features, see NICE DCV server features in the *NICE DCV Administrator Guide*.

Windows client

The NICE DCV Windows client is supported on Windows computers only. The Windows client is a standalone application that runs on the Windows operating system.

For instructions on how to connect to a NICE DCV session using the Windows client, see Connecting to a NICE DCV session using the Windows client (p. 9).

The Windows client is available in two versions: an installable version and a portable version. Both versions have the same minimum system requirements and have the same features.

Contents

Installable Windows client

You can use an installation wizard to install the client. The wizard takes you through a series of steps where you can customize your client installation. Or, you can use the command line to perform an unattended installation. This second method uses default settings to automate the installation procedure.

Before using the wizard or the command line to install the client, make sure that your computer has the required software. For a complete list of required software, see Requirements (p. 2).

To install the Windows client using the installation wizard

1. Download the Windows client installer.

   Tip
   The latest packages page of the download website contains links that always point to the newest available version. You can use these links to automatically retrieve the newest NICE DCV packages.
2. Run the installer.

3. On the **Welcome** screen, choose **Next**.
4. On the **End-User License Agreement** screen, read the license agreement. If you accept the terms, select the **I accept the terms in the License Agreement** check box. Choose **Next**.
5. On the **Destination Folder** screen, choose **Next** to keep the default installation folder. To install the client in a different folder, change the destination path, and then choose **Next**.
6. (Optional) On the **Drivers Selection** screen, select **USB device remotization**. Then, choose **Will be installed on local hard drive**, **Next**. This installs the drivers required to support some specialized USB devices. These devices include 3D pointing devices and graphic tablets.

   Note
   Using specialized USB devices requires additional client and server configuration. For instructions, see Using USB remotization (p. 25).
7. On the **Ready to install** screen, choose **Install**.

To install the Windows client using an unattended installation

1. Download the Windows client installer.
2. Open a command prompt window and navigate to the folder where you downloaded the installer.
3. Run the unattended installer.

```
C:\> msiexec.exe /i nice-dcv-client-Release-2022.0-8145.msi /quiet /norestart /l*v
 dcv_client_install_msi.log
```

   To install all of the optional components, including the USB driver, include the `ADDLOCAL=ALL` option in the command.

```
C:\>  msiexec.exe /i nice-dcv-client-Release-2022.0-8145.msi ADDLOCAL=ALL /quiet /
norestart /l*v dcv_client_install_msi.log
```

Portable Windows client

The Windows client is also available in a portable version. You don't need to install the portable version on your computer. You can copy it to a USB drive and run it directly from the USB drive on any Windows computer that meets the minimum requirements.

To use the portable Windows client

1. Download the portable Windows client zip file.

   Tip
   The latest packages page of the download website contains links that always point to the newest available version. You can use these links to automatically retrieve the newest NICE DCV packages.
2. Extract the contents of the zip file.
3. To launch the client, open the extracted folder, navigate to `/bin/`, and double-click `dcvviewer.exe`.

Web browser client

The NICE DCV web browser client runs inside a web browser. You don't need to install the web client. The web browser client is supported on the following browsers across all major desktop operating systems (including Windows, macOS, and Linux):

Browser	Version
Google Chrome	Latest three major versions
Mozilla Firefox	Latest three major versions
Microsoft Edge	Latest three major versions
Apple Safari	Latest three major versions

For instructions on how to connect to a NICE DCV session using the web browser client, see Connecting to a NICE DCV session using the web browser client (p. 10).

WebCodecs

The web browser client can use WebCodecs to use video decoders that are already present in the browser. This can improve frame rate, because packets can be decoded by components of the browser. The NICE DCV web browser client will automatically use it if supported by the browser.

The use of WebCodecs is available on the following browsers:

- Google Chrome version 94 and later
- Microsoft Edge version 94 and later

All major operating systems are supported. This includes Windows, macOS, and Linux.

Limitations

The web browser client has the following limitations:

- It supports up to two screens with a maximum resolution of 1920x1080. The maximum resolution can be overriden on the server side. For more information, see Managing the NICE DCV Session Display Layout in the *NICE DCV Administrator Guide*.
- It uses the web browser's proxy configuration.

Linux client

The Linux client runs natively on the operating system. You can use it to connect to NICE DCV sessions that are hosted on Windows and Linux NICE DCV servers.

You install the Linux client on a Linux client computer using a software package. The software package installs all required packages and their dependencies, and performs the required client configuration.

For instructions on how to connect to a NICE DCV session using the Linux client, see Connecting to a NICE DCV session using the Linux client (p. 11).

To install the Linux client

1. The software packages are digitally signed with a secure GPG signature. To allow the package manager to verify the package signature, import the NICE GPG key. To do this, open a terminal window and import the NICE GPG key.

   - RHEL 7.x/8.x, CentOS 7.x/8.x, and SUSE Linux Enterprise 15

   ```
   $ sudo rpm --import https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY
   ```

- Ubuntu

  Download the GPG key.

  ```
  $ wget https://d1uj6qtbmh3dt5.cloudfront.net/NICE-GPG-KEY
  ```

  Install the GPG key.

  ```
  $ sudo apt-key add NICE-GPG-KEY
  ```

2. Download the appropriate client software package for your target operating system from the NICE DCV website.

   Tip
   The latest packages page of the download website contains links that always point to the newest available version. You can use these links to automatically retrieve the newest NICE DCV packages.

3. Install the Linux client. Enter the filename of the downloaded file to complete the following command.

   - RHEL 7.x and CentOS 7.x

     ```
     $  sudo yum install the downloaded .rpm file
     ```

   - RHEL 8.x and CentOS 8.x

     ```
     $  sudo yum install the downloaded .rpm file
     ```

   - Ubuntu 18.04

     ```
     $  sudo dpkg --install the downloaded .deb file
     ```

   - Ubuntu 20.04

     ```
     $  sudo dpkg --install the downloaded .deb file
     ```

   - SUSE Linux Enterprise 15

     ```
     $  sudo zypper install the downloaded .rpm file
     ```

macOS client

The NICE DCV macOS client is supported on Apple Mac computers only. The macOS client is a standalone application that runs on the macOS operating system.

The macOS client is installed using a `.dmg` software package.

For instructions on how to connect to a NICE DCV session using the macOS client, see Connecting to a NICE DCV session using the macOS client (p. 12).

To install the macOS client

1. Download the correct macOS client installer depending on your client computer.

   - macOS client for Intel processors

- macOS clients for Apple M1 processors

Tip
The latest packages page of the download website contains links that always point to the newest available version. You can use these links to automatically retrieve the newest NICE DCV packages.

2. Run the downloaded `.dmg` file.

   If you get an error stating that the application can't be installed because it's from an unidentified developer, see the Safely open apps on your Mac webpage.
3. Click and drag the `DCV.app` file to the **Applications** folder.
4. (Optional) For easy access, create a desktop shortcut or add the application to the dock.

Using NICE DCV

After you choose a NICE DCV client, you can use it to connect to and interact with a NICE DCV session.

Topics

Connecting to a NICE DCV session

After the NICE DCV session starts running on the NICE DCV server, you can connect to it using your preferred client. Ensure that you have the required information when connecting to your NICE DCV session. For more information, see Step 1: Get the NICE DCV Session Information (p. 1).

If you're connecting to a console session, contact your NICE DCV server administrator. With them, ensure that the session is started and confirm the server and session details. If you're connecting to a virtual session on a Linux NICE DCV server, you might need to start your own session. For more information, see Starting NICE DCV Sessions in the *NICE DCV Administrator Guide*.

Topics

Connecting to a NICE DCV session using the Windows client

The steps for connecting to a NICE DCV session are the same for both the installable and portable versions of the Windows client.

To connect to a session using the Windows client

1. Launch the Windows client.

2. Choose **Connections Settings**, configure your proxy settings as follows, and then choose **OK**.

   - To avoid connecting through a proxy, choose **Connect Directly**.
   - To connect to the NICE DCV server using your preconfigured operating system proxy settings, choose **Use system proxy**.
   - To connect to the NICE DCV server through a specific HTTP proxy server, choose **Get through web proxy**. Specify the IP address and communication port or the hostname of the proxy server. If the HTTP proxy server requires authentication, select the **Proxy server requiring password** check box and enter your user name and password.
   - To connect to the NICE DCV server through a specific SOCKS5 proxy server, choose **Get through SOCKSv5 proxy**. Specify the IP address and communication port or the hostname of the proxy server. If the SOCKSv5 proxy server requires authentication, select the **Proxy server requiring password** check box and enter your user name and password.
   - To select the transport protocol to use for data transport, choose the **Protocol** tab. By default, the client uses the QUIC protocol (based on UDP) for data transport if it's available. If it isn't available, the client uses the WebSocket protocol (based on TCP). This option is always available.

     QUIC is available only if the two following conditions are met. First, the NICE DCV server is configured to support it. Second, your network configuration supports UDP communication between the NICE DCV client and the NICE DCV server. Additionally, it's only supported for direct client-server communication where there are no intermediate proxies, gateways, or load balancers.

     You can force the client to use a data transport protocol by explicitly selecting it. To verify which protocol is in use, check the Streaming Modes dialog. Additionally, if the QUIC protocol is in use, "QUIC" appears in the titlebar.

     For more information and instructions, see Enable the QUIC UDP transport protocol in the *NICE DCV Administrator Guide*.

3. Specify the session details in the following format:

   ```
   server_hostname_or_IP:port#session_id
   ```

   In the following example, the command connects to a session that's named `my-session`. This session is hosted on a NICE DCV server with the hostname `my-dcv-server.com`. It's connected over port `8443`.

   ```
   my-dcv-server.com:8443#my-session
   ```

4. Choose **Connect**.
5. Enter your user name and password and choose **Login**.

   Note
   By default, the connection is terminated after three unsuccessful login attempts. To try again, restart the connection.

6. If you're prompted to verify the server's certificate, confirm the certificate's fingerprint with your NICE DCV administrator. If the fingerprint is valid, choose **Trust & Connect**.

Connecting to a NICE DCV session using the web browser client

The steps for connecting to a NICE DCV session are the same across all supported web browsers. The client connects to the NICE DCV server using your web browser's proxy settings. To connect using different proxy settings, see the documentation for your specific web browser.

Note
The web browser client doesn't support the QUIC (UDP) transport protocol.

To connect to your NICE DCV session using the web browser client

1. Open a web browser and enter the NICE DCV server URL in the following format:

   ```
   https://server_hostname_or_IP:port/#session_id
   ```

   In the following example, the URL connects to a session that's named `my-session`. This session is hosted on a NICE DCV server with the hostname `my-dcv-server.com`. It's connected over port `8443`.

   ```
   https://my-dcv-server.com:8443/#my-session
   ```

2. Enter your user name and password and choose **Login**.

   Note
   By default, the connection is terminated after three unsuccessful login attempts. To try again, restart the connection.

3. Your web browser might warn you that the server's certificate isn't trusted. If you're unsure about the authenticity of the certificate, confirm it with your NICE DCV administrator. Proceed if it's safe to do so.

   Note
   This step varies depending on the web browser that you're using.

Connecting to a NICE DCV session using the Linux client

The steps for connecting to a NICE DCV session are the same across all Linux clients.

To connect to a session using the Linux client

1. Launch the Linux client.
2. Choose **Connections Settings**, configure your proxy settings as follows, and then choose **Apply**.

   - To avoid connecting through a proxy, choose **Connect directly**.
   - To connect to the NICE DCV server using your preconfigured operating system proxy settings, choose **Use system proxy**.
   - To connect to the NICE DCV server through a specific HTTP proxy server, choose **Get through web proxy (HTTP)**. Specify the IP address or the hostname of the proxy server as well as the communication port. If the HTTP proxy server requires authentication, select the **Proxy server requiring password** check box and enter your user name and password.
   - To connect to the NICE DCV server through a specific HTTPS proxy server, choose **Get through web proxy (HTTPS)**. Specify the IP address or the hostname of the proxy server as well as the communication port. If the web proxy server requires authentication, select the **Proxy server requiring password** check box and enter your user name and password.
   - To select the transport protocol to use for data transport, choose the **Protocol** tab. By default, the client uses the QUIC protocol (based on UDP) for data transport if it's available. If it isn't available, the client uses the WebSocket protocol (based on TCP). This option is always available.

     QUIC is available only if the following two conditions are met. First, the NICE DCV server is configured to support it. Second, your network configuration supports UDP communication

between the NICE DCV client and the NICE DCV server. Additionally, it's only supported for direct client-server communication where there are no intermediate proxies, gateways, or load balancers.

You can force the client to use a data transport protocol by explicitly selecting it. To verify which protocol is in use, check the Streaming Modes dialog. Additionally, if the QUIC protocol is in use, "QUIC" appears in the titlebar.

For more information and instructions, see Enable the QUIC UDP transport protocol in the *NICE DCV Administrator Guide*.

3. Specify the session details in the following format:

```
server_hostname_or_IP:port#session_id
```

In the following example, the command connects to a session that's named `my-session`. This session is hosted on a NICE DCV server with the hostname `my-dcv-server.com`. It's connected over port `8443`.

```
my-dcv-server.com:8443#my-session
```

4. Choose **Connect**.
5. Enter your user name and password and choose **Login**.

   Note
   By default, the connection is terminated after three unsuccessful login attempts. To try again, restart the connection.

6. If you're prompted to verify the certificate on the server, confirm the fingerprint of the certificate with your NICE DCV administrator. If the fingerprint is valid, choose **Trust & Connect**.

Connecting to a NICE DCV session using the macOS client

To connect to a session using the macOS client

1. Launch the macOS client.

   If you get an error stating that the application can't be opened because it's from an unidentified developer, see the Safely open apps on your Mac webpage.

2. Choose **Connections Settings**, configure your proxy settings as follows, and then choose **Apply**.

   - To avoid connecting through a proxy, choose **Connect directly**.
   - To connect to the NICE DCV server using your preconfigured operating system proxy settings, choose **Use system proxy**.
   - To connect to the NICE DCV server through a specific HTTP proxy server, choose **Get through web proxy (HTTP)**. Specify the IP address or hostname of the proxy server as well as the communication port. If the HTTP proxy server requires authentication, select the **Proxy server requiring password** check box and enter your user name and password.
   - To connect to the NICE DCV server through a specific HTTPS proxy server, choose **Get through web proxy (HTTPS)**. Specify the IP address or hostname of the proxy server as well as the communication port. If the web proxy server requires authentication, select the **Proxy server requiring password** check box and enter your user name and password.
   - To select the transport protocol to use for data transport, choose the **Protocol** tab. By default, the client uses the QUIC protocol (based on UDP) for data transport if it's available. If it isn't available, the client uses the WebSocket protocol (based on TCP). This option is always available.

QUIC is available only if the following conditions are met. First, the NICE DCV server is configured to support it. Second, your network configuration supports UDP communication between the NICE DCV client and the NICE DCV server. Additionally, it's only supported for direct client-server communication where there are no intermediate proxies, gateways, or load balancers.

You can force the client to use a data transport protocol by explicitly selecting it. To verify which protocol is in use, check the Streaming Modes dialog. Additionally, if the QUIC protocol is in use, "QUIC" appears in the titlebar.

For more information, see Enable the QUIC UDP transport protocol in the *NICE DCV Administrator Guide*.

3. Specify the session details in the following format:

```
server_hostname_or_IP:port#session_id
```

In the following example, the command connects to a session that's named `my-session`. This session is hosted on a NICE DCV server with the host name `my-dcv-server.com`. It's connected over port `8443`.

```
my-dcv-server.com:8443#my-session
```

4. Choose **Connect**.
5. Enter your user name and password and choose **Login**.

   Note
   By default, the connection is terminated after three unsuccessful login attempts. To try again, restart the connection.

6. If you're prompted to verify the server's certificate, confirm the certificate's fingerprint with your NICE DCV administrator. If the fingerprint is valid, choose **Trust & Connect**.

Changing display resolution

By default, NICE DCV automatically adapts the display resolution of the remote machine to match the current size of the client. When the client window is resized, DCV requests the server to change its display resolution to a size that fits within the client window.

If you prefer a fixed resolution on the server, which does not change even when the client window is resized, select the **Display Resolution** menu and specify the desired resolution. If you decide to re-enable automatic resize, you can select **Adapt automatically** .

This functionality is available on the Windows client, web browser client, Linux client, and macOS client.

Managing streaming modes

NICE DCV uses an adaptive protocol that automatically optimizes the streaming mode depending on the network capabilities. However, you can specify whether you prefer to prioritize responsiveness or image quality. Prioritizing responsiveness reduces the image quality to improve the frame rate. Prioritizing image quality reduces the responsiveness to provide better image quality.

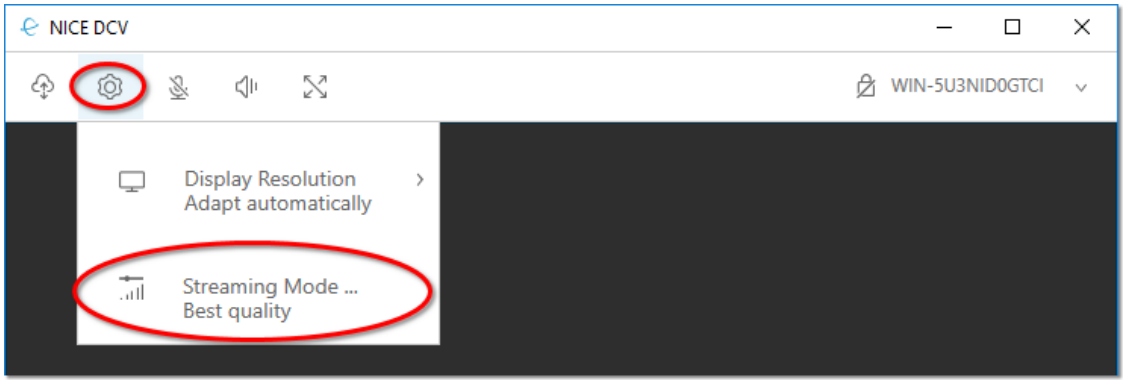
This functionality is available on the Windows client, web browser client, Linux client, and macOS client. The steps for setting the streaming mode depend on the client used.

Topics

Streaming modes on Windows, Linux, and macOS clients

To change the streaming mode on Windows, Linux, and macOS clients:

1. In the client, choose **Settings**, **Streaming Mode**.



2. In the Streaming Mode window, choose one of the following options:
   - **Best responsiveness** — This option prioritizes faster response times. It might result in lower image quality.
   - **Best quality** — This option prioritizes higher image quality. It might result in longer response times.
3. (Optional) For information about network performance, choose **Display Streaming Metrics**. For more information, see Streaming metrics (p. 14).
4. Close the **Streaming Mode** window.

Streaming metrics

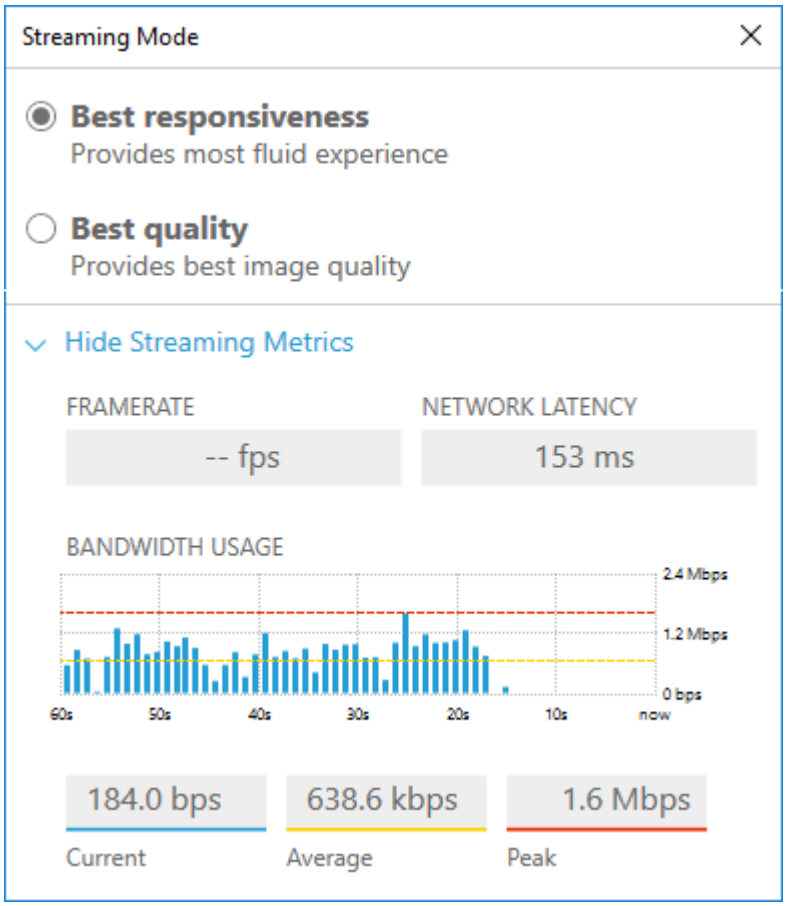
The streaming metrics can be used to evaluate your network performance and determine which streaming mode is suitable for your network conditions. To view the streaming metrics, choose **Settings**, **Streaming Mode**, **Display Streaming Metrics**.

The streaming metrics provide the following real-time information:

> **Note**
> Metrics are displayed for the current NICE DCV session connection.

- **Framerate**—Indicates the number of frames received from the NICE DCV server every second.
- **Network latency**—Indicates the amount of time (in milliseconds) it takes for a packet of data to be sent to the NICE DCV server and back to the client.
- **Bandwidth usage**—Indicates the amount of data being sent and received over the network connection. The red line shows the peak network throughput. The yellow line shows the average throughput. The blue line shows the current (real-time) throughput.

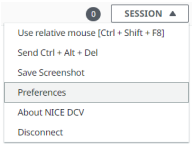
The following image shows example streaming metric data.



Streaming modes on Web browser client

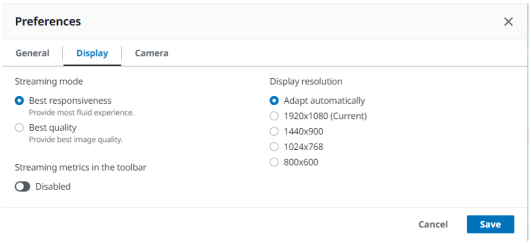
The steps for managing the streaming modes are the same across all supported web browsers.

1. In the client, choose **Session**, **Preferences**.



2. Under the **Display** tab, choose one of the following options from the **Streaming options** section:

   - **Best responsiveness** — This option prioritizes faster response times. It might result in lower image quality.
   - **Best quality** — This option prioritizes higher image quality. It might result in longer response times.



3. (Optional) For information about network performance, choose **Display Streaming Metrics**. For more information, see Streaming metrics (p. 16).
4. Save and close the **Preferences** modal.

Streaming metrics

The streaming metrics can be used to evaluate your network performance and determine which streaming mode is suitable for your network conditions.

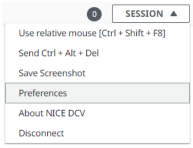
The streaming metrics provide the following real-time information:

> **Note**
> Metrics are displayed for the current NICE DCV session connection.

- **Framerate**—Indicates the number of frames received from the NICE DCV server every second.
- **Network latency**—Indicates the amount of time (in milliseconds) it takes for a packet of data to be sent to the NICE DCV server and back to the client.
- **Bandwidth usage**—Indicates the amount of data being sent and received over the network connection. The red line shows the peak network throughput. The yellow line shows the average throughput. The blue line shows the current (real-time) throughput.

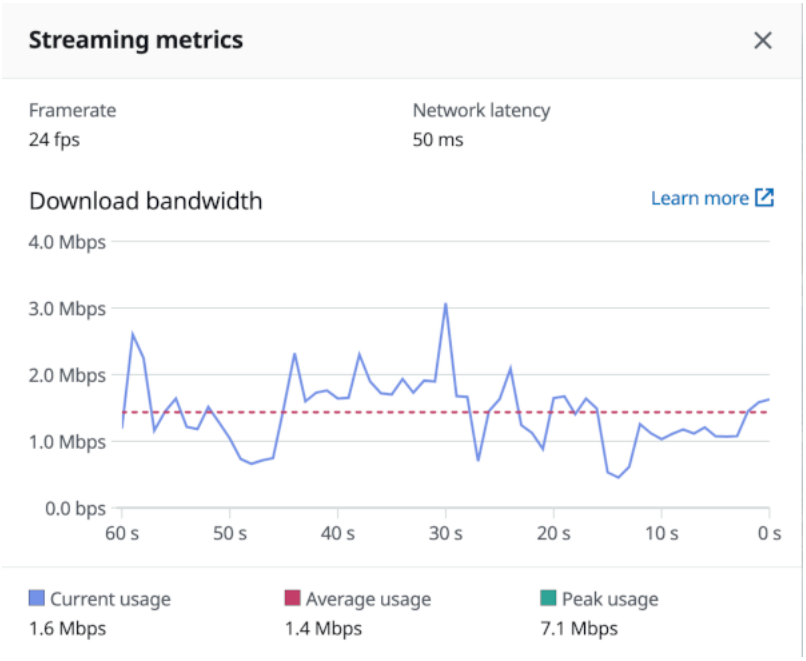
To view the streaming metrics:

1. In the client, choose **Session**, **Preferences**.



2. Under the **Display** tab, enable the toggle to show **Streaming metrics in the toolbar**.
3. Close the **Preferences** modal.
4. The streaming metrics are then displayed in the center of the client toolbar.

5. Click on the streaming metrics to see more detailed streaming data like in the following example.



6. (Optional) Close the **Metrics** modal.

Transferring files

You can use NICE DCV to upload files to and download files from NICE DCV session storage. For instructions on how to enable and configure session storage, see Enabling Session Storage in the *NICE DCV Administrator Guide*.

You must be authorized to use this feature. If you are not authorized, the functionality is not available in the client. For more information, see Configuring NICE DCV Authorization in the *NICE DCV Administrator Guide*.

This functionality is available on the Windows, web browser, Linux, and macOS clients.

Topics

Transferring files using Windows, Linux and macOs clients

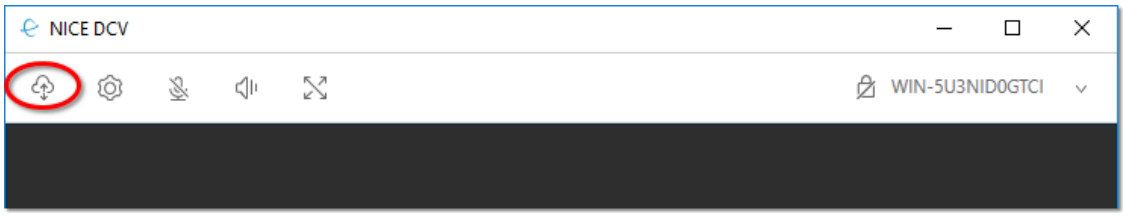
The steps for uploading, downloading, and renaming files are similar across Windows, Linux and macOS clients.

Downloading a file

If you're using the Windows client, the file is downloaded to your desktop. If you're using the Linux, or macOS client, the file is downloaded to your default Downloads folder.

To download a file from the session storage to your computer

1. In the client, choose **Storage**.



2. In the **File Storage** window, either select the file to download, or choose the down arrow next to the file and then choose **Download**.

Uploading a file

Files that you upload to a session are saved to a path that's specified by the NICE DCV server administrator.

To upload a file from your computer to the session storage

1. In the client, choose **Storage**.
2. (Optional) To upload the file to a new folder, choose **Create Folder**, enter a folder name, and then open the folder.
3. In the **File Storage** window, choose **Upload File**, browse to and select the file to upload, and then choose **Open**.

Renaming a file

You can change the name of the file in session storage.

To change the name of a file in session storage

1. In the client, choose **Storage**.
2. Choose the down arrow that's next to the file to rename, and choose **Rename**.
3. Enter the new file name and press **Enter**.

Transferring files using web browser

The steps for uploading, downloading, and renaming files are the same across all supported web browsers.

Downloading a file

In the web browser client the file is downloaded to your default Downloads folder.

To download a file from the session storage to your computer

1. In the client, choose **Storage**.

2. In the **File Storage** window, either select the file to download, or select a table row corresponding to the file to download and click on the **Actions** button and the **Download** option from there. You can also just click on the name of the file to download it.

Uploading a file

Files that you upload to a session are saved to a path that's specified by the NICE DCV server administrator.

To upload a file from your computer to the session storage

1. In the client, choose **Storage**.
2. (Optional) To upload the file to a new folder, choose **Create Folder**, enter a folder name, and then open the folder.
3. In the **File Storage** window, choose **Upload File**, browse to and select the file to upload, and then choose **Open**.

Renaming a file

You can change the name of the file in session storage.

To change the name of a file in session storage

1. In the client, choose **Storage**.
2. Select a table row corresponding to the item to rename and click on the **Actions** button and the **Rename** option from there.
3. Enter the new file name and press **Enter**.

Printing

You can use NICE DCV to print content from a NICE DCV session. The available printing devices depend on the client that you're using.

- **Windows client** — You can print to the physical printer that's connected to your client computer. Or, you can print to a `.PDF` document that uses the NICE DCV virtual printer.
- **Linux client** and **macOS client** — You can print to a `.PDF` document using the NICE DCV virtual printer.
- **Web browser client** — You can print to a `.PDF` document using the NICE DCV virtual printer.

When you print to the NICE DCV virtual printer, the content is exported to a printable file. You can download it to your local computer using the client and then print it using your local printer.

You must be authorized to use this feature. If you are not authorized, the functionality is not available in the client. For more information, see Configuring NICE DCV Authorization in the *NICE DCV Administrator Guide*.

To print content from the session

1. In the client, open the Print window.
2. In the Print window, choose one of the following printing devices and then choose **Print**.

   - (All clients connected to all Windows and Linux server) **DCV Printer** — Prints to the NICE DCV virtual printer

- (Windows client connected to Windows server only) ***<local-printer-name>*-DCV Redirected** — Prints to the local printer
- (Windows client connected to Linux server only) ***<local-printer-name>*-DCV-Redirected-*<connection-id>*** — Prints to the local printer

3. If you print to the NICE DCV virtual printer, a notification appears when the file is ready for download. In the top-right corner, choose **Notifications**, locate the Print notification in the list, and then choose **Download**.

   - If you're using the web browser client, after the download has completed, choose **Show in folder**.
   - If you're using the Windows client, the printer dialog is automatically opened when the file is downloaded.
   - If you're using the Linux or macOS clients, the downloaded file is automatically opened with the default associated application.

   Note
   The file is deleted from the NICE DCV server after you have downloaded it, and it's no longer available for download.

Copying and pasting

You can use NICE DCV to copy and paste text between your local computer and the NICE DCV session. You must be authorized to use this feature. If you are not authorized, the functionality is not available in the client. For more information, see Configuring NICE DCV Authorization in the *NICE DCV Administrator Guide*.

The type of content that can be copied and pasted, and the methods for copying and pasting differ between the Windows client, web browser client, Linux client, and macOS client.

Topics

Windows, Linux, and macOS clients

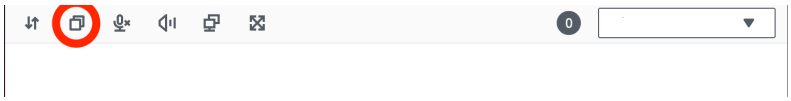
You can use the Windows, Linux, and macOS clients to copy and paste text and images between your local computer and the NICE DCV session. You can do this using the keyboard shortcuts and context (right-click) menu shortcuts. If you can't copy and paste, contact your NICE DCV server administrator to ensure that the permissions are properly configured.

Web browser client

You can use the web browser client to copy and paste text and images between your local computer and the NICE DCV session. Use keyboard shortcuts and context (right-click) menu to copy and paste text and images on Google Chrome and Microsoft Edge. Mozilla Firefox and Apple Safari do not support copying and pasting images, and require a different procedure to copy and paste text.

To copy text from the session in Mozilla Firefox or Apple Safari and paste on your local computer

1. In the web browser client, highlight the text to copy and choose **Clipboard**, **Copy to Local Device**.



The text is now placed in your computer's clipboard.

2. Paste the text using the paste keyboard shortcut or context menu shortcut.

To copy text from your local computer and paste in the session in Mozilla Firefox or Apple Safari

1. On your local computer, copy the text using the copy keyboard shortcut or context menu.
2. In the web browser client, choose **Clipboard**, **Paste to Remote Session**.
3. Paste the text using the host operating system's paste shortcuts.

Using a smart card

You can use NICE DCV to use one or more smart cards that's connected to your client computer. You can do this using the standard Personal Computer/Smart Card (PC/SC) interface, in a NICE DCV session. For each session, only one connected client can connect a smart card at a time. This is especially important in environments where multiple clients connect to the same session.

Smart card access is supported only with the Windows, Linux, and macOS clients. It's not supported with the web browser client.

You must be authorized to use this feature. If you are not authorized, the functionality is not available in the client. For more information, see Configuring NICE DCV Authorization in the *NICE DCV Administrator Guide*.

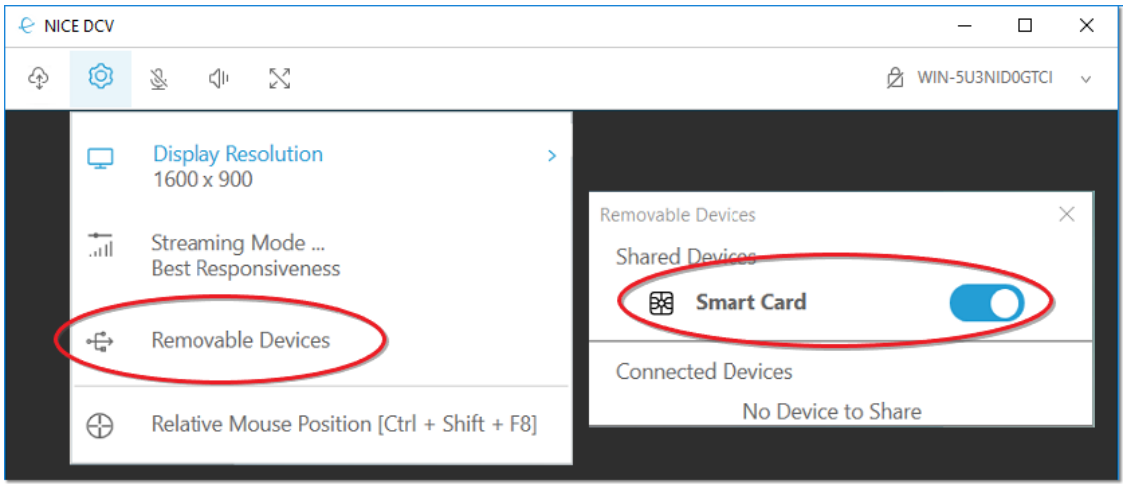
To use a smart card

1. Launch the client and connect to the NICE DCV session.
2. Connect the smart card to the session or release it.

   While your smart card is connected, no other clients who are connected to the session can connect a smart card. This is because only one client can connect a smart card at a time.

   After you're done using the smart card in the DCV session, release it. After it's released, other clients who are connected to the session can connect a smart card. The smart card is automatically released when you disconnect from the session.

   a. In the client, choose **Settings**, **Removable Devices**.
   b. To connect a smart card, enable the **Smart Card** toggle. To release control of the smart card, disable the **Smart Card** toggle.



3. (Optional) To have the NICE DCV server cache smart card data, enable the smart card caching feature. By default, smart card caching is disabled. With smart card caching enabled, the server caches the results of recent calls to the client's smart card. This helps to reduce the amount of traffic that is transferred between the client and the server and improves performance.

   You can't enable smart card caching if it's permanently disabled on the server. For more information, see Configuring Smart Card Caching in the *NICE DCV Administrator Guide*.

   To enable smart card caching, set and export the `DCV_PCSC_ENABLE_CACHE` environment variable. In the session, open a terminal window and run the following command:

   - Windows server

     To enable smart card caching for the current terminal window, run the following command.

     ```
     C:\> set DCV_PCSC_ENABLE_CACHE=1
     ```

     To enable smart card caching permanently for all applications on the server, run the following command.

     ```
     C:\> setx DCV_PCSC_ENABLE_CACHE 1
     ```

   - Linux server

     ```
     $  export DCV_PCSC_ENABLE_CACHE=1
     ```

     Note
     Make sure to run the following command in the same terminal that you intend to launch the application from (in step 4).

4. (Linux NICE DCV server only) Launch the required application with smart card support. In the session, open a terminal window, and launch the application using the `dcvscrun` command. For example, to launch `firefox` with smart card support, use the following command:

   ```
   $  dcvscrun firefox
   ```

Important
If you enabled smart card caching, run the following command in the same terminal that you set and exported the `DCV_PCSC_ENABLE_CACHE` environment variable in.

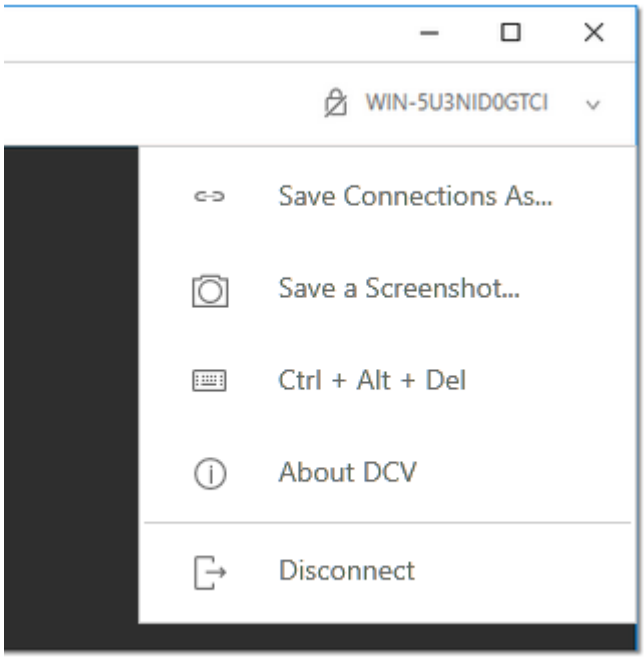
Saving a screenshot

You can use NICE DCV to save a screenshot of the NICE DCV session. This functionality is available on the Windows, web browser, Linux, and macOS clients. The steps for saving a screenshot are similar on all clients.

You must be authorized to use this feature. If you aren't authorized, the functionality isn't available in the client. For more information, see Configuring NICE DCV Authorization in the *NICE DCV Administrator Guide*. If you aren't authorized to save screenshots, the client also avoids the external tools that are running on your client computer to capture a screenshot of the NICE DCV client. Images that are obtained by these tools either show a black rectangle instead of the NICE DCV client window or only show the background desktop. This functionality is available only on Windows and macOS clients.

To save a screenshot

1. Launch the client, and connect to the NICE DCV session.
2. In the client, choose **Session**, **Save a Screenshot**.



3. Choose a location and the name for the screenshot file.

Using multiple screens

You can use the NICE DCV clients to extend the display for a session across multiple screens.

With the Windows, Linux, and macOS clients, the extended display matches your physical display layout and screen resolutions. For example, assume that you have three screens connected to your local computer. The server extends the display for a session across all three screens and matches the specific screen resolutions of your displays.

With the web browser client, the session display can be extended to up to two screens with 1920x1080 screen resolution. When the display is extended, the additional screen is opened in a new browser window. The second extends the display to the right of the original screen. Ensure that you position the screens accordingly.

You can also manually specify custom display layouts. For more information, see Managing the NICE DCV Session Display Layout in the *NICE DCV Administrator Guide*.

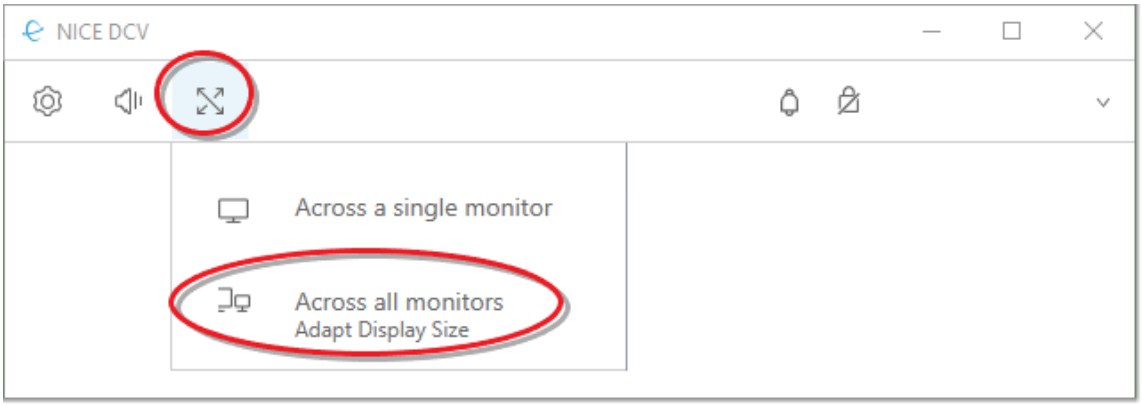
> **Note**
> If the requested layout isn't supported by the server, the layout might be adjusted to match the display limits of your server. If the layout can't be adjusted, the request fails and the changes aren't applied.

To extend the display

Do one of the following depending on your client.

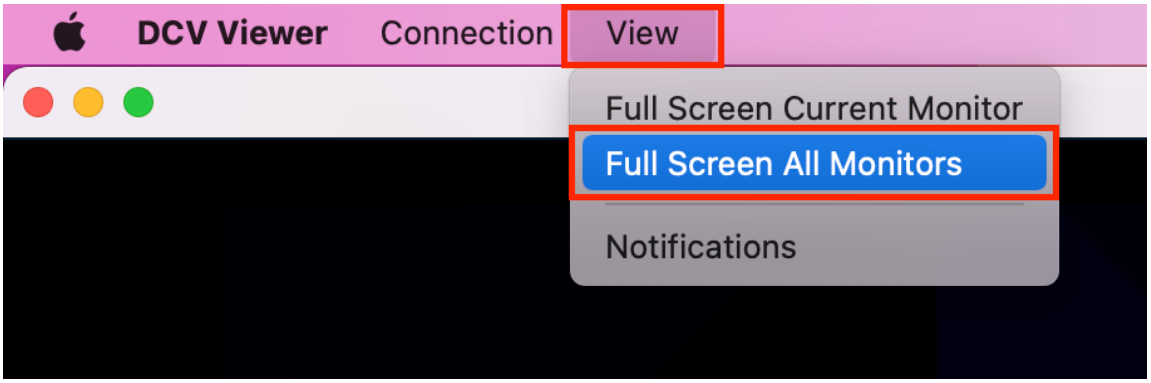
- Windows and Linux clients

  In the client, choose **Enter fullscreen**, **Across all monitors**.



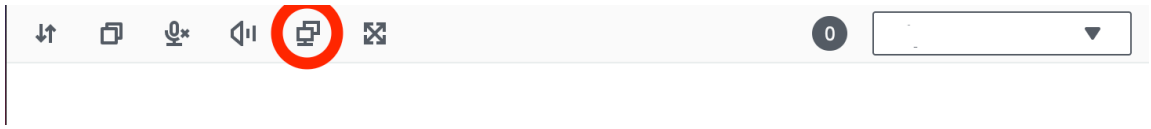
- macOS client

  In the client, choose **View**, **Full Screen All Monitors**.



- Web browser client

  In the client, choose **Multiscreen**.



After you extend the displays or enter fullscreen mode, a tab appears at the top-center edge of the screen. To exit fullscreen mode, choose the tab and then **Exit fullscreen**.

Using USB remotization

NICE DCV allows you to use specialized USB devices, such as 3D pointing devices and two-factor authentication USB dongles. These devices must be physically connected to your computer to interact with an application that's running on a NICE DCV server.

Graphic tablets, gamepads, and smart card readers are automatically supported by NICE DCV and do not require USB remotization to be used.

You must be authorized to use this feature. If you are not authorized, the functionality is not available in the client. For more information, see Configuring NICE DCV Authorization in the *NICE DCV Administrator Guide*.

> **Note**
> USB remotization is only supported in the installable Windows client. It isn't supported by the portable Windows client, web browser client, Linux client, or macOS client. Additional configuration might be required on the NICE DCV server. For more information and instructions, see Enabling USB Remotization in the *NICE DCV Administrator Guide*.

By default, the most commonly used USB devices are supported. For these devices, you can connect them to your computer and use them on the server without any additional configuration required. To use a USB device, connect it to your computer. In the client, choose **Settings**, and then move the slider next to the USB device in the list.

However, some specialized USB devices aren't supported in the default configuration. Unsupported devices don't appear in the **Settings** menu after they're connected. These devices must be added to the USB device *allow list* on the NICE DCV server before they can be used. After they're added to the allow list, they appear in the **Settings** menu in the client.

To use a device that must be added to the allow list on the NICE DCV server

1. Ensure that you installed the latest version of the Windows client and that you opted to install the USB remotization drivers. For more information, see Installable Windows client (p. 4).
2. Ensure that the USB device is connected to your computer and that you installed the required hardware drivers.
3. Navigate to `C:\Program Files (x86)\NICE\DCV\Client\bin\` and run `dcvusblist.exe`.
4. Open the context (right-click) menu for the USB device in the list and choose **Copy filter string**. Then, send the filter string to your NICE DCV server administrator.

   > **Note**
   > The NICE DCV server administrator adds the filter string for each USB device to the allow list. For more information, see Enabling USB Remotization in the *NICE DCV Administrator Guide*.

5. After the device is added to the allow list on the NICE DCV server, choose **Settings**. Then, move the slider next to the USB device to use it.

Using a webcam

With NICE DCV, you can use a webcam connected to your local client computer in a remote application that runs in a NICE DCV session. For each session, only one connected client can use a webcam at a time. This is especially important in environments where multiple clients connect to the same session.

Webcam functionality is supported with all NICE DCV clients. However, with the web browser client, webcam functionality is only supported with Chromium-based browsers, such as Google Chrome or Microsoft Edge. It isn't supported on Mozilla Firefox or Apple Safari.

Webcam functionality is supported on Windows NICE DCV servers only. It's not supported on Linux NICE DCV servers.

You must be authorized to use this feature. If you are not authorized, the functionality is not available in the client. For more information, see Configuring NICE DCV Authorization in the *NICE DCV Administrator Guide*.

If you have multiple webcams connected to your local client computer, you can select the webcam that you want to use. The selected camera is used automatically when the webcam is enabled using the webcam toolbar icon.

Topics

Using a webcam on Windows, Linux and macOS clients

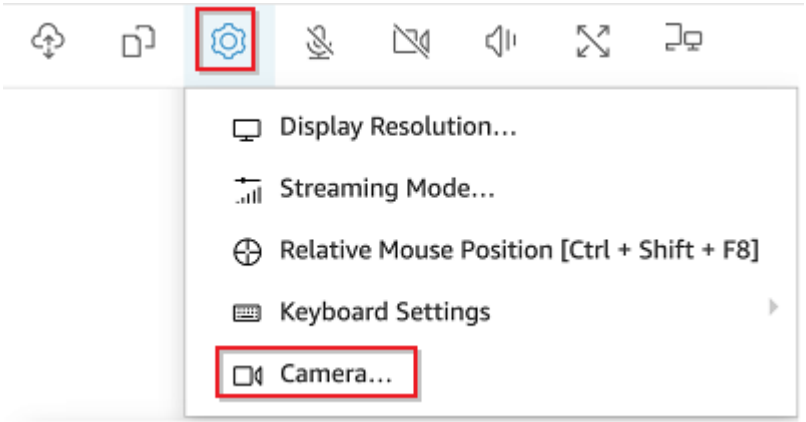
The steps for selecting the camera to use are similar across the Windows, Linux and macOS clients.

To select the webcam to use

1. Launch the client and connect to the NICE DCV session.
2. Do one of the following depending on your client.

   - Windows and Linux clients

     Choose **Settings**, **Camera** and then select the webcam to use.

   - macOS client

Choose **Connection**, **Camera Settings** and then select the webcam to use.

Note

- The camera menu items appears only if you're authorized to use a webcam in the session. If you don't see the camera menu items, you might not be authorized to use a webcam.
- You can't change the webcam selection while the webcam is in use, or while another client enabled a webcam in the session.

To start using your webcam in a session

You must first enable it. Use the webcam icon on the toolbar to enable or disable your webcam for use in the session. You can also use the icon to determine its current state. The webcam icon appears on the toolbar only if the following is the case:

- You're authorized to use a webcam.
- You have at least one webcam connected to your local computer.
- No other users enabled a webcam for use in the session.

Toolbar icon	Description
	Your webcam is disabled in the session. Other clients can enable a webcam for use in the session. Click the icon to enable your webcam in the session. If you didn't previously select the webcam to use, the default webcam is used.

Toolbar icon	Description
	Your webcam is enabled in the session, but it isn't in use. While your webcam is enabled, no other clients that are connected to the session can use a webcam. Click the icon to disable your webcam in the session.
	Your webcam is in use by a remote application in the NICE DCV session. No other clients can enable a webcam while your webcam is in use. Click the icon to disable your webcam in the session.

Troubleshooting

Topics

Webcam doesn't work on Windows 10

Windows 10 provides built-in privacy settings that manage access to the device camera. If you're running Windows 10 on your client computer, these privacy settings might prevent use of the webcam.

> **Note**
> If you're connecting to a Windows 2019 NICE DCV server, you might need to perform these steps on the NICE DCV server as well.

To modify the privacy settings on your computer, do the following:

1. Choose the search icon on the toolbar.
2. Enter `Settings` and press **Enter**.
3. In the left-hand panel, choose **Camera**.
4. For **Allow apps to access your camera**, switch the toggle to the **On** position.
5. You might need to restart your computer for the changes to take effect.

Client application says that the webcam is in use

Only one application can use the webcam at a time. If you're using the webcam in multiple applications, first close the applications where it's no longer needed.

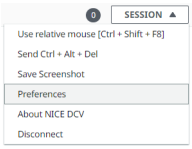
Using a webcam on the web browser client

Webcam functionality is only supported with Chromium-based browsers, such as Google Chrome or Microsoft Edge. It isn't supported on Mozilla Firefox or Apple Safari.

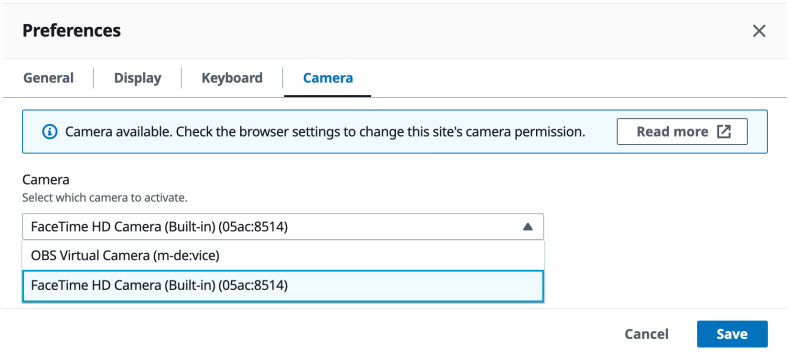
The steps for selecting the camera to use are the same across all supported web browsers..

To select the webcam to use

1. Launch the client and connect to the NICE DCV session.
2. In the client, choose **Session**, **Preferences**.



3. Under the **Camera** tab, select the camera to use.



4. Close the **Preferences** modal.

Note

- The camera menu items appears only if you're authorized to use a webcam in the session. If you don't see the camera menu items, you might not be authorized to use a webcam.
- You can't change the webcam selection while the webcam is in use, or while another client enabled a webcam in the session.
- If the camera permission settings have not been expressly granted or denied by the user, you're prompted to allow camera detection before being able to select the camera to use.
- In case the camera permission settings have been expressly granted or denied by the user, you would be able to change such setting following this procedure:

  1. At the top left of your browser window, click the area on the address bar left of the URL.
  2. In the popup window that opened, select the desired camera permission setting to be applied.

To start using your webcam in a session

You must first enable it. Use the webcam icon on the toolbar to enable or disable your webcam for use in the session. You can also use the icon to determine its current state. The webcam icon appears on the toolbar only if the following is the case:

- You're authorized to use a webcam.
- You have at least one webcam connected to your local computer.
- No other users enabled a webcam for use in the session.

Toolbar icon	Description
	Your webcam is disabled in the session. Other clients can enable a webcam for use in the session. Click the icon to enable your webcam in the session. If you didn't previously select the webcam to use, the default webcam is used.

Toolbar icon	Description
	Your webcam is enabled in the session, but it isn't in use. While your webcam is enabled, no other clients that are connected to the session can use a webcam. Click the icon to disable your webcam in the session.
	Your webcam is in use by a remote application in the NICE DCV session. No other clients can enable a webcam while your webcam is in use. Click the icon to disable your webcam in the session.

Troubleshooting

Client application says that the webcam is in use

Only one application can use the webcam at a time. If you're using the webcam in multiple applications, first close the applications where it's no longer needed.

Using accurate audio/video synchronization

The **Accurate Audio/Video synchronization** setting enables a mode that minimizes the time difference in audio and video playback. This mode improves lip sync. This mode is useful in workloads that require the video and audio to be accurately synchronized. This feature might introduce a lag in the perceived responsiveness of the remote system.

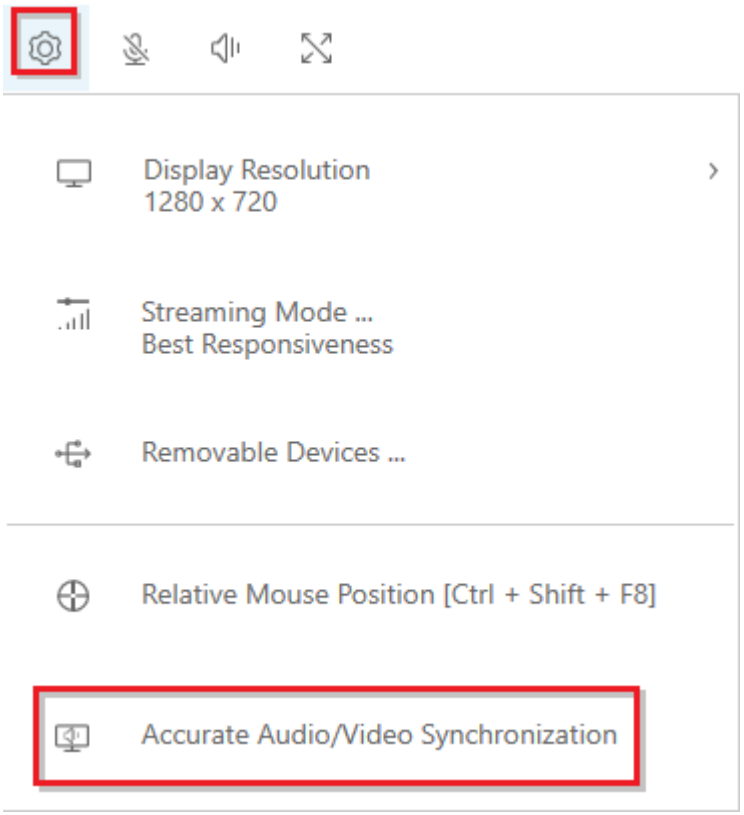
Accurate Audio/Video Synchronization functionality is supported on all native clients.

Accurate Audio/Video Synchronization functionality is supported on Windows and Linux servers with hardware GPU acceleration, and for console sessions only.

To enable or disable Audio/Video Synchronization

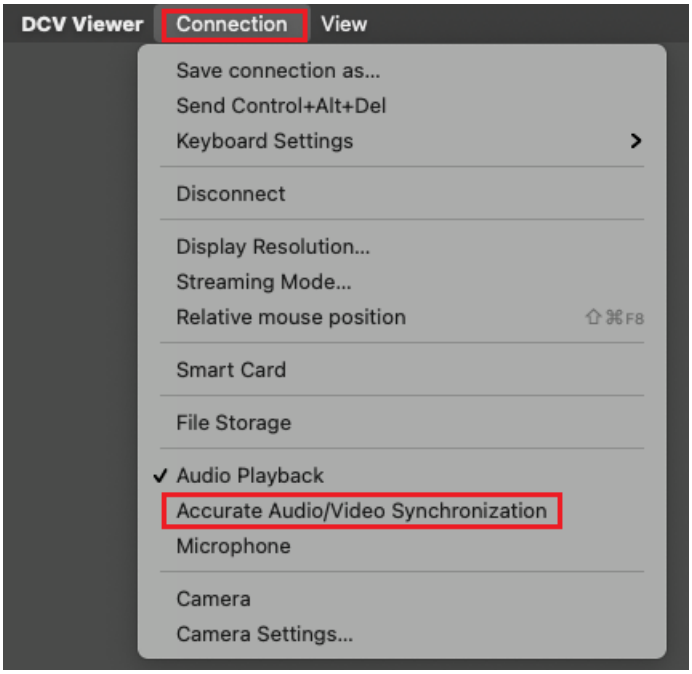
1. Launch the client and connect to the NICE DCV session.
2. Do one of the following depending on your client.
   - Windows, and Linux clients

     Choose **Settings**, **Accurate Audio/Video Synchronization**



- macOS client

  Choose **Connection**, **Accurate Audio/Video Synchronization**



Using high color accuracy

By default, NICE DCV uses YUV 4:2:0 chroma subsampling when compressing the display output and then updates the parts of the screen that are not changing over time to a full lossless RGB implementation. This default behavior aims to strike a balance between performance and image fidelity, though it may introduce chroma artifacts. By enabling the High color accuracy setting, the YUV chroma subsampling will be set to 4:4:4, thus increasing color fidelity. However this will increase network bandwidth and could affect performance of clients, especially at high resolution, because most client machines do not support HW accelerated decoding when using YUV 4:4:4.

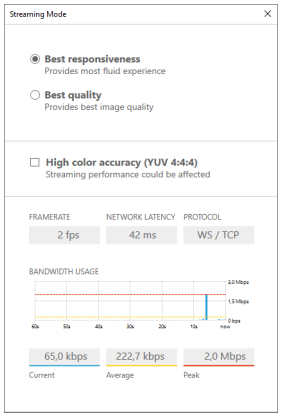
The steps for setting the high color accuracy depend on the client used.

Topics

High color accuracy on native clients

As long as you are using a NICE DCV Server and a NICE DCV Client both having version 2022.0 or later, please follow these steps to enable high color accuracy:

1. In the client, choose **Settings**, **Streaming Mode**.



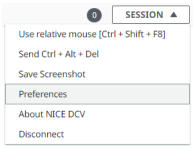
2. In the Streaming Mode window, the High color accuracy (YUV 4:4:4) checkbox allows to enable or disable the corresponding feature.
3. Close the **Streaming Mode** window.

High color accuracy on Web browser client

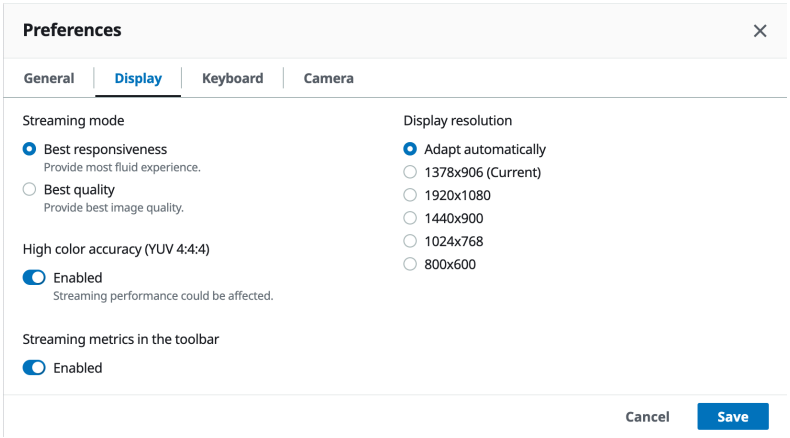
In order to use high color accuracy on Web browser client you need a NICE DCV Server with version 2022.0 or later, as well as a browser supporting the VideoDecoder interface of the Web Codecs API.

The steps for enabling the high color accuracy are the same across all supported web browsers.

1. In the client, choose **Session**, **Preferences**.



2. Under the **Display** tab, if the high color accuracy feature is available, the corresponding toggle will be visible and allows to specify whether to enable or disable the YUV chroma subsampling set to 4:4:4:



3. Save and close the **Preferences** modal.

Using a connection file

Using the Windows, Linux, or macOS native client, you can create a connection file that you can use to instantly connect to a NICE DCV session.

Contents

Creating the connection file

The connection file is a text-based file with a `.dcv` file extension. The format of the `.dcv` file is similar to that of an `.ini` file. The file includes `[groups]` followed by the parameters and their values. The groups and parameters take the following format:

```
[group_name]
parameter_name=parameter_value
```

For example:

```
[options]
fullscreen=true
```

For the Windows client, you can create a connection file for a specific NICE DCV session directly from the client. Or, alternatively, you can create a connection file from scratch using a text editor. For the Linux and macOS clients, you can only create a connection file from scratch using a text editor.

Note
The procedure for creating a connection file from scratch using a text editor is the same for the Windows, Linux, and macOS clients.

To create a connection file from the Windows client

1. Open the Windows client and connect to the server and session to create the file from.
2. Select the hostname for the NICE DCV server in the top-right corner and choose **Save Connection As**.
3. In the **Save As** window, enter a file name and destination folder, and choose **Save**.

By default, when you create a connection file using the Windows client, the file includes the `format`, `host`, `port`, `user`, and `proxytype` parameters. These parameters are required to connect to the session that the file was created from. You can manually customize or add parameters at any time by editing the file using a text editor.

To create a connection file from scratch using a text editor

1. Create a `.dcv` file with the following file name format: *`file_name`*`.dcv`
2. Open the `.dcv` file using your preferred text editor.
3. Add the `[version]` group and `format` parameter to the top of the file in the following format:

```
[version]
format=1.0
```

Important
If the `.dcv` file doesn't include the `[version]` group and `format` parameter, parsing fails.

4. Add the required parameter groups using the following format:

```
[group_name]
```

For more information about the parameter groups, see Supported parameters (p. 34).

5. Add the parameters and parameter values after the groups using the following format:

```
parameter_name=parameter_value
```

Note

- Parameter names are case sensitive.
- Don't enclose string parameter values in quotation marks.

For more information about the parameters and parameter values, see Supported parameters (p. 34).

6. Save the changes and close the `.dcv` file.

You can also use this procedure to add additional parameters to an existing connection file at any time.

Supported parameters

Currently, the `.dcv` file supports parameters in three parameter groups—`[version]`, `[connect]`, and `[options]`. The following tables list the groups and their available parameters.

Groups

[version] parameters

Important
This is a required group. If your `.dcv` file doesn't include this group, parsing fails.

The following table lists the parameters that can be specified in the `[version]` group.

Parameter	Type	Default value	Description	
format	string		**Important** This is a required parameter. The parameter value must be `1.0`. If your `.dcv` file doesn't include this parameter, parsing fails.	

[connect] parameters

The following table lists the parameters that can be specified in the `[connect]` group.

Parameter	Type	Default value	Description	
host	String		The hostname of the NICE DCV server hosting the session.	
port	Integer	8443	The port to use when connecting to the NICE DCV server.	
weburlpath	String		A custom path on the NICE DCV server for connection. For example, if you specify `customPath`, the client attempts to connect to `host:port/customPath`.	

Parameter	Type	Default value	Description	
sessionid	String		The ID of the NICE DCV session to connect to.	
authtoken	String		The authentication token to be used for the connection. If you specify an `authtoken`, you must also specify a `sessionid`. When using `authtoken`, you can omit the `user` and `password` parameters.	
user	String		The user name to use when connecting to the NICE DCV server.	
password	String		The password to use when connecting to the NICE DCV server. The password isn't encrypted.	
proxytype	String	SYSTEM	The proxy type to be used. Valid values include `HTTPS`, `HTTP`, `SOCKS5 | SOCKS`, `SYSTEM`, or `NONE | DIRECT`. If you specify `SYSTEM`, your computer's proxy settings are used.	
proxyhost	String		The address of the proxy server to be used if connecting through a proxy server.	
proxyport	Integer		The port to be used if connecting through a proxy server.	
proxyuser	String		The user name to be used for proxy authentication.	

Parameter	Type	Default value	Description	
proxypassword	String		The password to be used for proxy authentication. The password isn't encrypted.	
transport	String	`websocket`	The protocol to use for data transport. Specify `websocket` to use the WebSocket (TCP) protocol for data transport, or specify `quic` to use the QUIC (UDP) protocol for data transport. If you enable QUIC, the QUIC protocol is used for data transport and WebSocket is used for authentication traffic. If you enable WebSocket, the WebSocket protocol is used for both data transport and authentication traffic.	
webport	Integer	8443	The port to use for WebSocket (TCP) traffic.	
quicport	Integer	8443	The port to use for QUIC (UDP) traffic.	
certificatevalidationpolicy	String	ask-user	The policy for validating an untrusted certificate. Values include `strict`, `accept-untrusted`, and `ask-user`.	

`[options]` parameters

The following table lists the parameters that can be specified in the `[options]` group.

Parameter	Type	Default value	Description	
fullscreen	Boolean	false	Indicates whether the client starts in full screen mode.	
useallmonitors	Boolean	false	Indicates whether the client uses all monitors when starting full screen mode.	
promptreconnect	Boolean	true	Indicates whether the client prompts you to reconnect after you disconnect from a session. If the parameter is set to `true`, you're redirected to the sign-in screen when you disconnect. If the parameter is set to `false`, the client closes when you disconnect.	
enableyuv444decoding	Boolean	false	Indicates whether to enable the High color accuracy (YUV 4:4:4) (p. 32) when encoding dynamic video content.	

Running the connection file

To run `.dcv` connection file, navigate to the file and double-click it.

Or, specify the file path as an argument for the `dcvviewer` command.

- Windows client

```
C:\> dcvviewer.exe path\connection_file_name.dcv
```

- Linux and macOS client

```
$ dcvviewer path/connection_file_name.dcv
```

Set certificate validation policy

NICE DCV uses a secure TLS connection for communication between the server and client. The certificate validation policy determines how the NICE DCV client responds when a certificate can't be verified as trustworthy. Set one of the following options in the connection file:

- `Strict`: Prohibits the connection if there is any problem validating the TLS certificate.
- `Ask user`: Prompts the user to determine whether to trust the certificate when a certificate can't be verified.
- `Accept untrusted`: Connects to the server even if the TLS certificate is self signed and can't be validated by the client.

For information about editing the connection file, see ??? (p. 33).

Troubleshooting NICE DCV

This chapter describes how to identify and troubleshoot problems that you might have with NICE DCV client.

Topics

If you are a NICE DCV on-premises customer and you need additional help, contact your NICE DCV reseller to get the first level of support. If you are using NICE DCV on Amazon EC2, you can log a support ticket with AWS support. If you do not have an AWS support plan, you can seek help from the NICE DCV community by posting your question on the AWS forum.

Using the Log Files

Use the NICE DCV client log files to identify and troubleshoot problems with your NICE DCV client. Logs aren't enabled by default on Windows clients. After logs are enabled, the log files are stored in the following location on your NICE DCV client:

- Windows client

```
C:\ProgramData\client.log
```

  Note
  By default, the `ProgramData` folder might be hidden. If you don't see the `ProgramData` folder, set your file browser to show hidden items. Alternatively, enter `%programdata%` in the address bar and press **Enter**.

- Linux or macOS client

```
~/.local/share/NICE/dcvviewer/log/viewer.log
```

To enable NICE DCV to store log files on a Windows client

1. Navigate to the folder where the `dcvviewer.exe` file is located. (By default, this is `C:\Program Files (x86)\NICE\DCV\Client\bin\`.) Then, open a command prompt window.
2. Launch NICE DCV client using the command line interface.

   ```
   dcvviewer --log-level info --log-file-name C:/ProgramData/client.log
   ```

   Or add the following configuration to the connection file (p. 33):

   ```
   [debug]
   logfilename=C:/ProgramData/client.log
   loglevel=info
   ```

Release notes and document history for NICE DCV

This page provides the release notes and document history for NICE DCV.

Topics

NICE DCV release notes

This section provides an overview of the major updates, feature releases, and bug fixes for NICE DCV. All the updates are organized by release data. We update the documentation frequently to address the feedback that you send us.

Topics

- DCV 2017.0-5121 — March 18, 2018 (p. 58)
- DCV 2017.0-4334 — January 24, 2018 (p. 58)
- DCV 2017.0-4100 — December 18, 2017 (p. 58)

DCV 2022.0-12760— May 23, 2022

Build numbers	Changes and bug fixes
• nice-dcv-server: 12760 • nice-dcv-client (Windows): 8145 • nice-dcv-viewer (macOS): 4131 • nice-dcv-viewer (Linux): 4131 • nice-xdcv: 424 • nice-dcv-gl: 961 • nice-dcv-gltest: 291 • nice-dcv-simple-external-authenticator: 188	Fixes: • Fixed a problem preventing successful connection of the Web Client when specifying the web-url-path option.

DCV 2022.0-12627— May 19, 2022

Build numbers	Changes and bug fixes
• nice-dcv-server: 12627 • nice-dcv-client (Windows): 8145 • nice-dcv-viewer (macOS): 4131 • nice-dcv-viewer (Linux): 4131 • nice-xdcv: 424 • nice-dcv-gl: 961 • nice-dcv-gltest: 291 • nice-dcv-simple-external-authenticator: 188	Fixes: • Fixed some problems in the QUIC transport which could cause incorrect bandwidth estimation and visual artifacts. • Fixed a problem with the Audio service in the installer of the Windows server which could cause the update process to fail. • Fixed a problem with the USB handling in the installer of the Windows client which could cause the uninstall process to fail. • Fixed a problem when saving a screenshot in the macOS and Linux clients. • Updated the OpenSSL, zlib and gdk-pixbuf third party libraries.

DCV 2022.0-12123— March 23, 2022

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 12123 • nice-dcv-client (Windows): 7920 • nice-dcv-viewer (macOS): 3973	NICE DCV added the following features: • Added option to enable high color accuracy to the macOS and Linux clients.	Changes: • Improved bandwidth estimation and image quality when using the QUIC transport.

Build numbers	New features	Changes and bug fixes
• nice-dcv-viewer (Linux): 3973 • nice-xdcv: 424 • nice-dcv-gl: 961 • nice-dcv-gltest: 291 • nice-dcv-simple-external-authenticator: 188		Fixes: • Fixed visual artifacts in console sessions on Linux when using NVIDIA drivers 510.xx. • Fixed problem with DualShock 4 controllers connected via Bluetooth in the Windows native client. • Fixed possible crash in the macOS client when enabling the webcam.

DCV 2022.0-11954— February 23, 2022

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 11954 • nice-dcv-client (Windows): 7866 • nice-dcv-viewer (macOS): 3929 • nice-dcv-viewer (Linux): 3929 • nice-xdcv: 424 • nice-dcv-gl: 961 • nice-dcv-gltest: 291 • nice-dcv-simple-external-authenticator: 188	NICE DCV added the following features: • Game controller support for Windows Server and Windows native client. • The NICE DCV Web Client now leverages WebCodecs on browsers that support it. • Added option to enable high color accuracy to the Windows and Web clients. • Improved collaboration experience: users get notified when someone joins the session • Added CentOS 8 Stream to the list of supported Linux distributions.	Changes: • TLS certificates can now be updated without restarting the NICE DCV Server. • It is now possible to configure the NICE DCV Server to listen on a specific network interface or on specific IPv4 or IPv6 addresses. • The 'DCV Printer' is now automatically configured also on Linux systems. • The NICE DCV processes on Windows are now executed at higher priority. Fixes: • Fixed a crash on agent restart on Windows 2016 when using instances with a GPU. • Fixed a crash on Windows when logging out of a session while some USB devices are redirected from the NICE DCV Client. • Normalize user names that contain a Windows domain when performing authorization checks. • Improved relative mouse mode in the Windows Client.

Build numbers	New features	Changes and bug fixes
		• Fixed a problem with the synchronization of the CapsLock key.

DCV 2021.3-11591— December 20, 2021

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 11591 • nice-dcv-client (Windows): 7801 • nice-dcv-viewer (macOS): 3829 • nice-dcv-viewer (Linux): 3829 • nice-xdcv: 415 • nice-dcv-gl: 952 • nice-dcv-gltest: 284 • nice-dcv-simple-external-authenticator: 176	NICE DCV added the following features: • The user interface of the Web Client has been updated. • EC2 G5 and G5g instances are now supported. • Windows Server 2022 and Windows 11 are now supported operating systems.	• The init script for Linux virtual sessions does not load the user's bash profile anymore, thus avoiding recurring problems with environment variables overriding the system's default values. • The nice-dcv-ext-authenticator now requires Python 3.

DCV 2021.2-11445— November 18, 2021

Build numbers	Changes and bug fixes
• nice-dcv-server: 11445 • nice-dcv-client (Windows): 7792 • nice-dcv-viewer (macOS): 3797 • nice-dcv-viewer (Linux): 3797 • nice-xdcv: 411 • nice-dcv-gl: 946 • nice-dcv-gltest: 279 • nice-dcv-simple-external-authenticator: 160	Fixes: • Fixed a problem preventing the client from working correctly on macOS Monterey. • Improved security in the server on Windows. • Fixed a bug which could cause multi-monitor layouts to not be applied correctly, in particular when using the Web Client. • Fixed a problem which could cause the `Delete` key to not work correctly with some Windows applications. • Marked the Web client package on Linux as mutually exclusive with old versions of the server package, which included the Web client itself.

DCV 2021.2-11190— October 11, 2021

Build numbers	Changes and bug fixes
• nice-dcv-server: 11190	Fixes:

Build numbers	Changes and bug fixes
• nice-dcv-client (Windows): 7788 • nice-dcv-viewer (macOS): 3776 • nice-dcv-viewer (Linux): 3776 • nice-xdcv: 411 • nice-dcv-gl: 946 • nice-dcv-gltest: 279 • nice-dcv-simple-external-authenticator: 160	• Fixed a problem in the Windows client which prevented the user from dismissing the certificate validation dialog when connecting to a server with an expired certificate. • Fixed a problem with the middle click button on Stylus pens not working as expected on native clients. • Fixed a regression in Xdcv which prevented legacy X11 fonts to be loaded. • Fixed a problem in the macOS and Linux clients with keyboard combinations not working correctly when using a keyboard layout which uses dead keys.

DCV 2021.2-11135— September 24, 2021

Build numbers	Changes and bug fixes
• nice-dcv-server: 11135 • nice-dcv-client (Windows): 7781 • nice-dcv-viewer (macOS): 3740 • nice-dcv-viewer (Linux): 3740 • nice-xdcv: 408 • nice-dcv-gl: 944 • nice-dcv-gltest: 279 • nice-dcv-simple-external-authenticator: 160	Fixes: • Fixed a problem with QUIC packet size negotiation that can cause connectivity and performance problems when using a 2021.2 client to connect with an older server. • Fixed a bug with NVIDIA device selection that could cause NVENC encoder to fail. • Fixed problems on machines with Windows and a NVIDIA GPU that could cause compression artifacts and color accuracy artifacts. • Fixed a bug with modifier keys on Linux server which could cause some keyboard combinations to not work as expected. • Fixed a performance regression for macOS clients on machines with the M1 CPU. • Fixed a bug in the macOS client which would cause some keyboard combinations to not work as expected. • Fixed a problem with how touch events are handled in Linux virtual sessions that could cause termination of the session.

DCV 2021.2-11048— September 01, 2021

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 11048 • nice-dcv-client (Windows): 7774	NICE DCV added the following features:	Changes: • The NICE DCV web client is now a separate package

Build numbers	New features	Changes and bug fixes
• nice-dcv-viewer (macOS): 3690 • nice-dcv-viewer (Linux): 3690 • nice-xdcv: 406 • nice-dcv-gl: 944 • nice-dcv-gltest: 279 • nice-dcv-simple-external-authenticator: 160	• Web client clipboard improvements. With these improvements, you can now copy and paste PNG format images using the NICE DCV web client on Google Chrome and Microsoft Edge. • A screenshot blocking feature for the Windows and macOS clients. This feature adds an additional layer of security by preventing users from taking screenshots of NICE DCV session content. When enabled, any screenshots that a user capture result in a blank screen. • Streaming quality improvements. Streaming quality improved specifically through better “build-to-lossless” performance when using the QUIC protocol. • A `certificate-validation-policy` option to specify the behavior of your client was addded. You can use it when the server presents an untrusted X.509 certificate, such as a self-signed certificate. • The number of channels configured in the Audio Driver at run time can be changed. • The Pressure2K option was added to the `dcvinput` Xorg module. You can use this to change the pressure sensitivity range of the stylus from 0-65335 to 0-2048, for compatibility with applications, such as Mari and Nuke • Support for the experimental WebCodecs API on Google Chrome and Microsoft Edge was added. When you enable this API in the browser, the NICE DCV web client can use it to accelerate video decoding and deliver higher frame rates.	on Linux and an optional component in the Windows installer. With this change, customers can decide whether to deploy the web client. • The H.264 High Profile is now supported when the NVENC encoder is used. Using NVENC encoder with NVIDIA GPUs, you can reduce bandwidth usage while maintaining the same image quality. • NICE DCV server now uses all available GPUs for compression on machines with more than one GPU. • All Windows drivers shipped with NICE DCV are now WHQL certified. • OpenSSL was updated to version 1.1.1. • Xdcv was updated to version 1.20.13 of XServer. Fixes: • Fixed a problem with numpad keys on macOS clients. • Fixed an issue that prevented some USB devices (for example, gamepads) to be properly redirected to Windows servers. • Fixed a bug where modifier keys couldn't be properly released on disconnection. • Fixed a crash in the Linux native client when using Ubuntu 20.04 and Intel GPUs.

DCV 2021.1-10851— July 30, 2021

Build numbers	Changes and bug fixes
• nice-dcv-server: 10851 • nice-dcv-client (Windows): 7744 • nice-dcv-viewer (macOS): 3590 • nice-dcv-viewer (Linux): 3560 • nice-xdcv: 392 • nice-dcv-gl: 937 • nice-dcv-gltest: 275 • nice-dcv-simple-external-authenticator: 154	Changes: • We improved stability on the Windows, Linux, and macOS clients. Fixes: • Fixed a bug that caused screen flickering with AMD and NVIDIA graphic adapters on Windows servers. • Fixed a sporadic issue when connecting to a Linux server running multiple sessions. • Fixed bugs that were related to handling of non-western keyboard layouts on Linux server. • Fixed visual artifact on the connection window in the Windows client. • Fixed several bugs and improved device compatibility in the USB redirection driver on Windows.

DCV 2021.1-10598— June 10, 2021

Build numbers	Changes and bug fixes
• nice-dcv-server: 10598 • nice-dcv-client (Windows): 7713 • nice-dcv-viewer (macOS): 3473 • nice-dcv-viewer (Linux): 3473 • nice-xdcv: 392 • nice-dcv-gl: 937 • nice-dcv-gltest: 275 • nice-dcv-simple-external-authenticator: 154	• Fixed a problem in the Windows installer of the server to prefill the `session owner` field with the current user. • Improved the overall stability of the macOS and Linux clients.

DCV 2021.1-10557— May 31, 2021

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 10557 • nice-dcv-client (Windows): 7713 • nice-dcv-viewer (macOS): 3450	• NICE DCV added client option to enable accurate Audio/Video synchronization when connecting to a server with a GPU.	• Reduced CPU usage on Windows server hosts without a GPU.

Build numbers	New features	Changes and bug fixes
• nice-dcv-viewer (Linux): 3454 • nice-xdcv: 392 • nice-dcv-gl: 937 • nice-dcv-gltest: 275 • nice-dcv-simple-external-authenticator: 154	• NICE DCV added support for microphone on Linux console sessions.	• Fixed a problem with reading `.dcv` connection files in the macOS and Linux clients. • Added fallback to software decoding for macOS machines that don't support hardware accelerated decoding. • Added support for macOS client to read CA certificates that are stored in the system keychain.

DCV 2021.0-10242— April 12, 2021

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 10242 • nice-dcv-client (Windows): 7643 • nice-dcv-viewer (macOS): 3186 • nice-dcv-viewer (Linux): 3294 • nice-xdcv: 380 • nice-dcv-gl: 912 • nice-dcv-gltest: 266 • nice-dcv-simple-external-authenticator: 134	• Added webcam redirection support for Windows NICE DCV servers. • Added printer redirection support for Linux NICE DCV servers. • Added support for M1 processors on macOS clients. • Added multi-monitor display support for macOS clients.	• Optimized GPU and CPU resource usage on Linux servers and on Amazon EC2 instances with an NVIDIA GPU. • Added support for GPU accelerated video encoding using AMD GPUs on Amazon EC2 G4ad instances for Linux NICE DCV servers. • Optimized audio processing to reduce audio latency • Changed the default for clients to the QUIC protocol if the protocol is enabled on the server. • Added a new get-screenshot command to the DCV command line tool. • Added a force logout option that uses the `--logout-user` option of the `close-session` command. You can use this option when closing a console session.

DCV 2020.2-9662— December 04, 2020

Build numbers	Changes and bug fixes
• nice-dcv-server: 9662 • nice-dcv-client (Windows): 7490 • nice-dcv-viewer (macOS): 2117	• Enhanced the security protocols used in the web browser client.

Build numbers	Changes and bug fixes
• nice-dcv-viewer (Linux): 3007 • nice-xdcv: 359 • nice-dcv-gl: 881 • nice-dcv-gltest: 259 • nice-dcv-simple-external-authenticator: 125	• Increased performance and robustness of Amazon EC2 G4ad instances used with the Windows client. • Fixed a problem with port selection in the connection settings dialog of the Windows client.

DCV 2020.2-9508— November 11, 2020

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 9508 • nice-dcv-client (Windows): 7459 • nice-dcv-viewer (macOS): 2078 • nice-dcv-viewer (Linux): 1737 • nice-xdcv: 359 • nice-dcv-gl: 881 • nice-dcv-gltest: 259 • nice-dcv-simple-external-authenticator: 125	• Added support for the QUIC (UDP-based) transport protocol. • Added support for SLES 15 and Ubuntu 20.4. • Added smart card support for Windows NICE DCV servers.	• Changed the default the NICE DCV frame rate limiter to 60 FPS for console sessions that are hosted on servers and EC2 instances with an NVIDIA GPU. • Optimized the GPU and CPU resources used on Windows NICE DCV servers that are hosted on EC2 instances with an NVIDIA GPU. • Added the `list-endpoints` NICE DCV CLI command. This lists the current active endpoints. • The `version` NICE DCV CLI command supports the `--json` option. • On Linux servers, the `create-session` NICE DCV CLI command now supports the `--disable-login-monitor` option. • Improved compatibility with different display managers on Linux NICE DCV servers. • Fixed several issues in the handling of keyboard input. • The USB devices allow list file is now dynamically reloaded.

DCV 2020.1-9012— September 30, 2020

Build numbers	Changes and bug fixes
• nice-dcv-server: 9012 • nice-dcv-client (Windows): 7342 • nice-dcv-viewer (macOS): 1986	• Added missing macOS client icons.

Build numbers	Changes and bug fixes
• nice-dcv-viewer (Linux): 1545 • nice-xdcv: 338 • nice-dcv-gl: 840 • nice-dcv-gltest: 246 • nice-dcv-simple-external-authenticator: 111	

DCV 2020.1-9012— August 24, 2020

Build numbers	Changes and bug fixes
• nice-dcv-server: 9012 • nice-dcv-client (Windows): 7342 • nice-dcv-viewer (macOS): 1910 • nice-dcv-viewer (Linux): 1545 • nice-xdcv: 338 • nice-dcv-gl: 840 • nice-dcv-gltest: 246 • nice-dcv-simple-external-authenticator: 111	• Fixed Amazon S3 access in AWS GovCloud Region • Web-based client improvements

DCV 2020.1-8942— August 03, 2020

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 8942 • nice-dcv-client (Windows): 7342 • nice-dcv-viewer (macOS): 1910 • nice-dcv-viewer (Linux): 1545 • nice-xdcv: 338 • nice-dcv-gl: 840 • nice-dcv-gltest: 246 • nice-dcv-simple-external-authenticator: 111	• The Linux NICE DCV server now supports AWS Graviton2-based Arm instances, such as M6g, C6g, and R6g. For more information, see AWS Graviton Processor. • Added support for RHEL 8.x and CentOS 8.x on Linux NICE DCV server. • Added support for printer redirection when using a Windows NICE DCV server and the Windows NICE DCV client. • Added stylus support with pressure sensitivity on macOS and Linux native NICE DCV client. • Added surround sound 5.1 support for Linux NICE DCV server and Linux NICE DCV client.	• Added support for the new NICE DCV Virtual Display driver on Amazon EC2 instances that don't have a GPU. • Resolved the issue that caused visual artifacts as a result of colorspace conversion when using the NVENC encoder. • The `dcv list-sessions` command now always includes the console session, if one is present • On newer Linux distributions, the agent for console sessions is now started as part of the desktop session to better support newer display managers, such as GDM3. • Native clients now automatically open when activating a URL with the `dcv://` scheme.

Build numbers	New features	Changes and bug fixes
	• Added touch screen support for Linux NICE DCV native client. • You can now associate a custom name to a NICE DCV session. • Support for hardware accelerated decoding and rendering on the macOS native NICE DCV client.	• Improved how the macOS native client and web client handle keyboard modifiers. • Improved visual and fbconfig selection in DCV-GL to improve support for some applications. • Reduced CPU usage during file transfer • Improved WebGL rendering in the web browser client to reduce resource usage.

DCV 2020.0-8428 — April 16, 2020

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 8428 • nice-dcv-client (Windows): 7238 • nice-dcv-viewer (macOS): 1716 • nice-dcv-viewer (Linux): 1358 • nice-xdcv: 296 • nice-dcv-gl: 759 • nice-dcv-gltest: 229 • nice-dcv-simple-external-authenticator: 87	• Added on-screen stylus and touch support on Linux server. • Added 7.1 surround sound playback support on Windows server to Windows native client. • Added hardware acceleration and stylus support on Linux native client. • Added a new API command to set display layout on the server side. • Added multi-monitor web client display support on the Microsoft Edge browser (version 79.0.309 or later).	• The toolbar grip on the Windows client can now be hidden while in full screen mode. • Added NTLM proxy support on Windows native client. • Improved support for Windows headless physical hosts using NVIDIA adapters. • Removed support for the legacy NVIDIA NvIFR library. • Added support for Windows Graphic Capture API on latest Windows 10. • Added support for Amazon EC2 Instance Metadata Service (IMDS) v2 on EC2 instances. • DCV CLI provides new `on-client-connected/disconnected` commands to detect when a client connects or disconnects from a session. • Added support for specifying the host name to bind certificates for the external authenticator. • DCV-GL now uses the GL Vendor-Neutral Dispatch library (GLvnd) on systems that support it.

DCV 2019.1-7644 — October 24, 2019

Build numbers	Changes and bug fixes
• nice-dcv-server: 7644 • nice-dcv-client (Windows): 7114 • nice-dcv-viewer (macOS): 1535 • nice-dcv-viewer (Linux): 1124 • nice-xdcv: 226 • nice-dcv-gl: 544 • nice-dcv-gltest: 220 • nice-dcv-simple-external-authenticator: 77	• Fixed an issue in the integration API used by NICE EnginFrame and other session managers. • Fixed an issue with the 32-bit version of the Windows native client.

DCV 2019.1-7423 — September 10, 2019

Build numbers	Changes and bug fixes
• nice-dcv-server: 7423 • nice-dcv-client (Windows): 7087 • nice-dcv-viewer (macOS): 1535 • nice-dcv-viewer (Linux): 1124 • nice-xdcv: 226 • nice-dcv-gl: 544 • nice-dcv-gltest: 220 • nice-dcv-simple-external-authenticator: 77	• Improved security for DCV server on Windows. • Fixed a rendering problem with Autodesk Maya on Linux. • Added improvements and bug fixes related to keyboard handling.

DCV 2019.0-7318 — August 5, 2019

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 7318 • nice-dcv-client (Windows): 7059 • nice-dcv-viewer (macOS): 1530 • nice-dcv-viewer (Linux): 968 • nice-xdcv: 224 • nice-dcv-gl: 529 • nice-dcv-gltest: 218 • nice-dcv-simple-external-authenticator: 72	• Multiple monitor support on Web client. • Stylus input support on Windows Server 2019. • Audio in/out on macOS and Linux native clients. • Enhanced clipboard capability on Linux server (middle-click paste).	• Added improved compatibility for pressure sensitivity for Windows touch input. • Improved behavior on systems that have heterogeneous graphic adapters on Windows. • Reduced time required to detect inactive connections (for example, in response to changes from wired to Wi-Fi networks on the client). • Reduced logging when the cursor icon can't be captured on Linux. • Support for disabling the Composite extension in

Build numbers	New features	Changes and bug fixes
		the virtual sessions Xdcv component. • Added the option to a limit on the number of concurrent virtual sessions. • Improved script compatibility for systems with Bash 5 installed. • Changed default for OpenGL and GLES to be detected and used automatically for rendering on the Linux client. • Updated the DCV-GL on-screen buffer when the visibility of a GL window changes. • Fixed the mouse wheel detection in the Windows client on Windows 7. • Fixed a problem that caused the Windows client to fail when loading libraries on some Windows 7 systems. • Improved printing on the Windows client when printing documents with landscape orientation.

DCV 2017.4-6898 — April 16, 2019

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 6898 • nice-dcv-client (Windows): 6969 • nice-dcv-viewer (macOS): 1376 • nice-dcv-viewer (Linux): 804 • nice-xdcv: 210 • nice-dcv-gl: 490 • nice-dcv-gltest: 216 • nice-dcv-simple-external-authenticator: 70	• New native client for macOS.	• The Windows native client now uses hardware acceleration for decoding and rendering, if available in the system. • The `dcv` command line tool now uses the same options and output format on both Windows and Linux. • The `dcv` command line tool now reports information about licenses. • Clients now show a warning to the user before disconnection due to inactivity. • Improved support for keyboard combinations that use multiple modifiers.

Build numbers	New features	Changes and bug fixes
		• Improved robustness of the interaction with the Reprise License Manager for communication failures. • The `dcvusers` command line tool now defaults to saving data to the `dcv` user home directory on Linux. • Followed the same ordering used by the `nvidia-smi` tool when using the NVENC hardware encoder with multiple GPUs on Linux. • The Linux client now receives and handles printed files from the Windows DCV printer.

DCV 2017.3-6698 — February 24, 2019

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 6698 • nice-dcv-client: 5946 • nice-dcv-viewer (Linux): 683 • nice-xdcv: 207 • nice-dcv-gl: 471 • nice-dcv-gltest: 210 • nice-dcv-simple-external-authenticator: 66	• Added support for Kerberos (GSSAPI) authentication. • Added support for touch events on Windows versions that support it. • Automatically unlock Windows sessions when using system authentication (Windows Credential Provider).	• Added an option to opt in to Y'UV444 encoding. • The EL6 RPM now includes the NVENC encoder module. • Windows system authentication now accepts the `name@domain` format. • Yubikey USB devices are now added to the allow list. • Improved Japanese keyboard support. • Input authorization permissions are more fine-grained. Added `pointer` permission to handle virtual cursors. Relative mouse mode depends on mouse (for motion injection) and pointer (for motion feedback). Added `keyboard-sas` permission to handle SAS on Windows (Control+Alt+ Del). `keyboardsas` depends on `keyboard` permission. • Fixed a problem with empty clipboard events in the Web client on browsers that support the async clipboard API.

Build numbers	New features	Changes and bug fixes
		• Fixed a race on the capture module that prevented clients from receiving the first frame. • Improvements to handling of concurrent file storage transfers. • Fixed NvIFR on Windows with newer NVIDIA drivers. New drivers changed behavior. The driver version is now detected automatically and memory handling is performed accordingly. • Never stop retrying re-acquiring an RLM license token. This allows you to recover from a `licensing error` state even after extended time periods. • Added an option to set full screen keyboard shortcut in the Windows client. • Improved auto-fit logic when dragging window across multiple monitors in the Windows client. • Fixed the prompt-reconnect option when disconnection is not triggered by UIin the Windows client. • Fixed DCV-GL incompatibility with NVIDIA driver 410.xx. • Fixed regressions in DCV-GL with the Matlab and Blender applications.

DCV 2017.2-6182 — October 8, 2018

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 6182 • nice-dcv-client: 5890 • nice-dcv-viewer (Linux): 503 • nice-xdcv: 180 • nice-dcv-gl: 427 • nice-dcv-gltest: 201 • nice-dcv-simple-external-authenticator: 58	• Added audio playback support on Linux virtual sessions. • Improved smart card performance. • Added file transfer support on the Linux client.	• Improvements and bug fixes related to keyboard handling. • Changing the log level in the configuration no longer requires a server restart. • The Windows server installer now skips installation of Microsoft C runtime redistributable if it's already installed.

Build numbers	New features	Changes and bug fixes
		• When running on EC2, if accessing S3 for the license fails, a notification is displayed in the user interface. • The Linux `dcv` command line tool now supports `list-connections` and `describe-session` sub-commands and includes an option to emit JSON output. • Added a `cuda-devices` setting in the `display` section. This configures the server to distribute NVENC encoding over different CUDA devices. • Improved robustness of session creation code when handling multiple concurrent commands. • Increased the default clipboard limit to 20 MB. • The Windows client now detects legacy `.dcv` files and launches the DCV 2016 Endstation (if installed). • The DCV simple external authenticator now always uses the system Python interpreter instead of the one set in the environment. • Improved the read-back strategy of DCV-GL for improved performance and robustness. • DCV-GL now checks if a window changed size after a front buffer readback. This fixes a rendering problem with the Coot application.

DCV 2017.1-5870 — August 6, 2018

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 5870 • nice-dcv-client: 5813 • nice-dcv-viewer (Linux): 450 • nice-xdcv: 170 • nice-dcv-gl: 366	Released package for Ubuntu 18.04. When working in console mode, the system must be configured to use LightDM or another display manager of your choice because GDM does not	• The license setting is now read when a session is created. This allows the administrator to change this setting without restarting the server.

Build numbers	New features	Changes and bug fixes
• nice-dcv-gltest: 198 • nice-dcv-simple-external-authenticator: 53	expose the required X11 display information. Virtual sessions are not affected by this limitation.	• Resolved stability problem in the Windows client that caused the program to exit unexpectedly on some systems. • Reduced logging in a possible error condition.

DCV 2017.1-5777 — June 29, 2018

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 5777 • nice-dcv-client: 5777 • nice-dcv-viewer (Linux): 438 • nice-xdcv: 166 • nice-dcv-gl: 366 • nice-dcv-gltest: 189 • nice-dcv-simple-external-authenticator: 51	• Added a Linux native client. • Added support for 3DConnexion mouses and USB storage devices. • Windows session locked automatically when the last client disconnects.	• Performance improvements in the Linux version. • Changed the default HW encoder on NVIDIA devices to NVENC to avoid problems with NvIFR in new NVIDIA drivers. • Improved smart card support on Linux. • Fixed file permissions for uploaded files when using Linux console sessions.

DCV 2017.0-5600 — June 4, 2018

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 5600 • nice-dcv-client: 5600 • nice-xdcv: 160 • nice-dcv-gl: 279 • nice-dcv-gltest: 184 • nice-dcv-simple-external-authenticator: 48	• Added support for multiple monitors on Linux. • Windows client performance improvements. • Used new Clipboard API on Chrome 66+. • Added NVENC encoder for Windows.	• Usage on EC2 now requires ability to reach S3 from the instance running DCV server. • Performance improvements to server frame processing and Windows client decoding. • Fixed keyboard issues related to NumPad and blocked modifiers. • Prevent file descriptor leak when using an external authenticator on Linux. • Fixed possible errors in smart card connection.

DCV 2017.0-5121 — March 18, 2018

Build numbers	New features	Changes and bug fixes
• nice-dcv-server: 5121 • nice-dcv-client: 5121 • nice-xdcv: 146 • nice-dcv-gl: 270 • nice-dcv-gltest: 184 • nice-dcv-simple-external-authenticator: 46	• Windows native client is now DPI aware. • Added support for relative mouse movement mode.	• Prevented hang on Ansys cfx5solve on Linux. • Fixed possible agent hang on Windows 10. • Improved the Web Client user interface. • Normalized Windows user name when a domain is specified. • Fixed the external authenticator on RHEL6.

DCV 2017.0-4334 — January 24, 2018

Build numbers	Changes and bug fixes
• nice-dcv-server: 4334 • nice-dcv-client: 4334 • nice-xdcv: 137 • nice-dcv-gl: 254 • nice-dcv-gltest: 184 • nice-dcv-simple-external-authenticator: 45	• Improved keyboard handling. • Fixed DBus problem on RHEL6 where closing of a session doesn't allow a new one to be created. • Improved support for SOCKS5 proxy on the native client. • Addressed the bug that cause crashes on Headwave when running on virtual sessions and on Chimera when running on virtual sessions. • Improved font support on virtual sessions.

DCV 2017.0-4100 — December 18, 2017

Build numbers
• nice-dcv-server: 4100 • nice-dcv-client: 4100 • nice-xdcv: 118 • nice-dcv-gl: 229 • nice-dcv-gltest: 158 • nice-dcv-simple-external-authenticator: 35

Document history

The following table describes the documentation for this release of NICE DCV.

Change	Description	Date
NICE DCV Version 2022.0	NICE DCV 2022.0 is now available. For more information, see DCV 2022.0-11954— February 23, 2022 (p. 43).	February 23, 2022
NICE DCV Version 2021.3	NICE DCV 2021.3 is now available. For more information, see DCV 2021.3-11591— December 20, 2021 (p. 44).	December 20, 2021
NICE DCV Version 2021.2	NICE DCV 2021.2 is now available. For more information, see DCV 2021.2-11048— September 01, 2021 (p. 45).	September 01, 2021
NICE DCV Version 2021.1	NICE DCV 2021.1 is now available. For more information, see DCV 2021.1-10557— May 31, 2021 (p. 47).	May 31, 2021
NICE DCV Version 2021.0	NICE DCV 2021.0 is now available. For more information, see DCV 2021.0-10242— April 12, 2021 (p. 48).	April 12, 2021
NICE DCV Web Client SDK	The NICE DCV Web Client SDK is now available. The NICE DCV Web Client SDK is a JavaScript library that you can use to develop your own NICE DCV web browser client applications that your end users can use to connect to and interact with a running NICE DCV session. For more information, see the NICE DCV Web Client SDK Developer Guide.	March 24, 2021
NICE DCV Version 2020.2	NICE DCV 2020.2 is now available. For more information, see DCV 2020.2-9508— November 11, 2020 (p. 49).	November 11, 2020
NICE DCV Version 2020.1	NICE DCV 2020.1 is now available. For more information, see DCV 2020.1-8942— August 03, 2020 (p. 50).	August 03, 2020
NICE DCV Version 2020.0	NICE DCV 2020.0 includes support for surround sound 7.1, touch and stylus, and multi-monitor using the new Microsoft Edge browser. For more information, see Installing the NICE DCV Server in the *NICE DCV Administrator Guide*.	April 16, 2020

Change	Description	Date
HTTP response headers	The NICE DCV server can be configured to send additional HTTP response headers.	August 26, 2019
macOS client	NICE DCV now offers a macOS client. For more information, see macOS Client in the *NICE DCV User Guide*.	April 18, 2019
Smart card caching	The NICE DCV server can now cache smart card data received from the client to help improve performance. For more information, see Configuring Smart Card Caching in the *NICE DCV Administrator Guide*.	October 08, 2018
Linux client	NICE DCV offers Linux clients for RHEL 7, CentOS 7, SLES 12, and Ubuntu 16.04/18.04. For more information, see Linux Client in the *NICE DCV User Guide*.	August 29, 2018
Updated Parameter Reference	The Parameter Reference was updated. For more information, see NICE DCV Server Parameter Reference in the *NICE DCV Administrator Guide*.	August 07, 2018
USB remotization	NICE DCV enables clients to use specialized USB devices, such as 3D pointing devices or graphic tablets. For more information, see Enabling USB Remotization in the *NICE DCV Administrator Guide*.	August 07, 2018
Initial release of NICE DCV	First publication of this content.	June 05, 2018