aws

NICE DCV Session Manager

NICE DCV Session Manager: CLI Guide

Table of Contents

What is NICE DCV Session Manager command line interface?

NICE DCV Session Manager is set of installable software packages (an Agent and a Broker) and an application programming interface (API) that makes it easy for developers and independent software vendors (ISVs) to build front-end applications that programmatically create and manage the lifecycle of NICE DCV sessions across a fleet of NICE DCV servers.

You can use NICE DCV Session Manager Command Line Interface (CLI) to interact with a DCV Session Manager broker using commands in your command-line shell.

- Linux shells — Use common shell programs such as `bash`, `zsh`, and `tcsh` to run commands in Linux or macOS.
- Windows command line — Run commands at the Windows command prompt or PowerShell.

You can use NICE DCV Session Manager CLI to create and manage the lifecycle of NICE DCV sessions. These are managed by a NICE DCV Session Manager.

Installing NICE DCV Session Manager Command Line Interface

This section describes how to install the NICE DCV Session Manager command line interface (CLI) on Linux, macOS, and Windows.

Prerequisites for installing NICE DCV Session Manager CLI

This section includes prerequisites for installing NICE DCV Session Manager CLI on Linux, macOS, and Windows.

Linux and macOS

- Python version 3.6 or later.

- Ability to extract or unzip a downloaded, compressed file. If your operating system doesn't have the built-in `unzip` command, use an equivalent.

Windows

- Python version 3.6 or later.

- Python must be properly set in the PATH environment variable to run NICE DCV Session Manager CLI using the startup script.

Installing NICE DCV Session Manager CLI

This section describes the process to install NICE DCV Session Manager CLI on Linux, macOS, and Windows.

Linux and macOS

To Install NICE DCV Session Manager CLI on Linux and macOS

1. Download the zipped package from the [NICE DCV website](#).

2. Unzip the downloaded package. If your Linux distribution doesn't have a built-in unzip command, use an equivalent to unzip it. The following example command unzips

the package and creates a directory that's named `nice-dcv-session-manager-cli-1.0.0-37` under the current directory.

```
$  unzip nice-dcv-session-manager-cli-1.0.0-37.zip
```

3. To validate the content of the zip file, run the following command inside the folder extracted in the previous step:

```
./dcvsm --version
```

Windows NICE DCV server

To install NICE DCV Session Manager CLI on Windows

1. Download the zipped package from the NICE DCV website.
2. Unzip the downloaded package.
3. To validate the content of the zip file, run the following command:

```
C:\>  dcvsm --version
```

Configuring the NICE DCV Session Manager command line interface

This section explains how to configure the settings that the NICE DCV Session Manager command line interface (CLI) uses to interact with the broker.

Credentials and configuration basics

To work with the CLI, you must register a Session Manager client with the broker and generate client credentials. Follow the process that's described in the Broker CLI reference.

The NICE DCV Session Manager CLI has a set of configuration parameters that the user can specify. The following parameters are required:

- Broker URL
- Client ID
- Client password

The configuration parameters can be defined in the following three ways:

- Command line
- Environment variables
- Configuration file

Hierarchy of configuration settings

The DCV SM CLI uses credentials and configuration settings that are located in multiple places. These inlude user environment variables, local DCV SM configuration file, or explicitly declared on the command line as a parameter. Certain locations take precedence over others.

The NICE DCV Session Manager CLI credentials and configuration settings take precedence in the following order:

- Command line options—Overrides settings in any other location.
- Environment variables—Some values can be stored in your system's environment variables.

- CLI configuration file—Specify options in the configuration file.

Command line options

In the DCV SM CLI, you can use the following command line options to override the default configuration settings, any corresponding profile setting, or environment variable setting for that single command. You can't use command line options to directly specify credentials. Each option that takes an argument requires a space or equals sign (=) separating the argument from the option name.

`--conf`

Specifies the path of the configuration file. If it's not defined the CLI will use the default configuration file. For more information about the configureation file, see [Configuration file](#).

Type: String

`--broker-url`

Specifies the URL to send the request to.

Type: String

`--debug`

A Boolean switch that enables debug logging. By default, the DCV SM CLI provides cleaned up information regarding any successes or failures regarding command outcomes in the command output. The --debug option provides the full logs. This includes additional stderr diagnostic information about the operation of the command that can be useful when troubleshooting why a command provides unexpected results.

Type: Boolean

Type: String

`--no-verify-ssl`

With this option, you can disable SSL certificate verification, but it's discouraged for security reasons. By default, the DCV SM CLI uses SSL when communicating with broker services.

Type: Boolean

`--ca-bundle`

Specifies the path to a custom certificate bundle (a file with a `.pem` extension) of CA to use when establishing SSL/TLS connections

`--auth-server-url`

Specifies the hostname of the authentication server that's used to request the OAuth 2.0 access token. If it's not specified, the broker url is used.

Type: String

`--version`

Displays the current version of the DCV SM CLI program that's running.

Type: String

`--help`

Display the list of options and commands of the DCV SM CLI.

Type: String

Environment variables

Environment variables provide another way to specify some configuration options and credentials.

The following examples describe how you can configure environment variables for the default user.

Set environment variables on Linux or macOS

Run the following commands to set environment variables on Linux or macOS:

```
$  export DCV_SM_AUTH_SERVER_URL=https://example.com
```

```
$  export DCV_SM_CLIENT_ID=ExampleClientID
```

Setting the environment variable changes the value that's used until the end of your shell session, or until you set the variable to a different value. You can make the variables persistent across future sessions by setting them in the startup script of your shell.

Set environment variables on Windows

Run the following commands to set environment variables on Windows Command Prompt:

```
$   setx DCV_SM_AUTH_SERVER_URL https://example.com
```

```
$   setx DCV_SM_CLIENT_ID ExampleClientID
```

Using set to set an environment variable changes the value that's used until the end of the current command prompt session. Or, it's until you set the variable to a different value. Using setx to set an environment variable changes the value that's used in both the current command prompt session and all command prompt sessions. These sessions you create after running the command. It doesn't affect other command shells that are already running at the time you run the command.

Set environment variables using PowerShell

Run the following commands to set environment variables using PowerShell:

```
$   $Env:DCV_SM_AUTH_SERVER_URL="https://example.com"
```

```
$   $Env:DCV_SM_CLIENT_ID="ExampleClientID"
```

If you set an environment variable at the PowerShell prompt as shown in the previous examples, it saves the value for only the duration of the current session. To make the environment variable setting persistent across all PowerShell and Command Prompt sessions, store it by using the *System* application in *Control Panel*. Alternatively, you can set the variable for all future PowerShell sessions by adding it to your PowerShell profile. For more information about storing environment variables or persisting them across session, see the PowerShell documentation.

NICE DCV Session Manager supported environment variables

NICE DCV Session Manager supports the following environment variables:

- DCV_SM_AUTH_SERVER_URL

  Specifies the hostname of the authentication server that's used to request the OAuth 2.0 access token. If defined, this environment variable overrides the value of the configuration file.

- DCV_SM_CLIENT_ID

  The unique client ID that's used by the Session Manager client to retrieve an OAuth 2.0 access token.

- DCV_SM_CLIENT_PASSWORD

  The password that's used by the Session Manager client to retrieve an OAuth 2.0 access token.

- DCV_SM_OAUTH2_TOKEN

  The access token that's used to make an API request.

Configuration file

The configuration file is the third way to specify some configuration options and credentials. By default, it's provided in the zip file at /conf/dcvsmcli.conf.

The user can specify the path of the configuration file at command line. If the user does not specify it explicitly the config file it will be read by default from a specific folder depending on the operating systems:

- Unix based systems: $XDG_CONFIG_HOME/dcvsmcli.conf

- Windows systems: %UserProfile%\dcvsmcli.conf

If the configuration file isn't found, the CLI reads the file from the path /conf/dcvsmcli.conf provided with the zip file. The configuration file is structured as follows:

```
[output]
# The formatting style for command output.
output-format = json

# Turn on debug logging
debug = true

[security]
# Disable SSL certificates verification.
no-verify-ssl = true

# CA certificate bundle to use when verifying SSL certificates.
```

```
ca-bundle = ca-bundle.pem

[authentication]
# hostname of the authentication server used to request the token
auth-server-url = https://broker-host:broker-port/oauth2/token?
grant_type=client_credentials

# The client ID
client-id = username

# The client password
client-password = password

[broker]
# hostname or IP of the broker
url = https://broker-host:broker-port
```

Working with the NICE DCV Session Manager CLI

This section introduces many of the common features and options available in the DCV SM Command Line Interface (DCV SM CLI).

Getting help

You can get help with any command when using the DCV SM Command Line Interface (DCV SM CLI). To do so, simply type `--help` at the end of a command name.

For example, the following command displays help for the general DCV SM CLI options and the available top-level commands.

```
dcvsm --help
```

The following example shows the output of the CLI help.

```
Usage: dcvsm [OPTIONS] COMMAND [ARGS]...

  NICE DCV Session Manager CLI

Options:
  --conf TEXT           Name of the configuration file to read the
                        configuration parameters
  --broker-url TEXT     The URL of the broker
  --debug               Specify to enable the debug mode. By default is
                        disabled
  --no-verify-ssl       Specify to disable the verification of SSL
                        certification. By default is enabled
  --output-format TEXT  Specify the format of the output
  --ca-bundle TEXT      Specifies the path to a custom certificate bundle
                        (a file with a .pem extension) of CA to use when
                        establishing SSL/TLS connections
  --auth-server TEXT    URL of the authentication server used to request
                        the token
  --version             Show the version and exit.
  -h, --help            Show this message and exit.

Commands:
  close-servers                 Closes one or more NICE DCV server
```

```
  create-session              Creates a new NICE DCV session
  delete-session              Deletes the specified NICE DCV session
  describe-servers            Describes one or more NICE DCV servers
  describe-sessions           Describes one or more NICE DCV sessions
  get-session-connection-data Gets connection data for a specific NICE DCV session.
  get-session-screenshots     Gets screenshots of one or more NICE DCV sessions.
  open-servers                Opens one or more NICE DCV servers
  update-session-permissions  Updates the user permissions for a specific NICE DCV
  session.
```

All commands accept --help for contextual help information. For example, the following command displays help for the create-session command.

```
dcvsm create-session --help
```

The following example shows the output for the `create-session --help` command.

```
Usage: dcvsm create-session [OPTIONS]

  Create sessions API

Options:
  --name TEXT                   The name for the session  [required]
  --owner TEXT                  The name of the session owner  [required]
  --type TEXT                   Session type: CONSOLE|VIRTUAL  [required]
  --init-file-path TEXT         Supported with virtual sessions on Linux
                                NICE DCV servers
  --autorun-file TEXT           The path to a file on the host server that
                                is to be run inside the session
  --autorun-file-arguments TEXT Command-line arguments passed to AutorunFile
                                upon its execution inside the session
  --max-concurrent-clients INTEGER RANGE
                                The maximum number of concurrent NICE DCV
                                clients, if provided must be between 1 and
                                100 [1<=x<=100]
  --dcv-gl-enabled              Indicates whether the virtual session is
                                configured to use hardware-based OpenGL.
                                Specify to enable it, by default is disabled
  --permissions-file TEXT       The Base64-encoded contents of the
                                permissions file
  --requirements TEXT           The requirements that the server must
                                satisfy in order to place the session
  --storage-root TEXT           Specifies the path to the folder used for
```

```
                                   session storage
   -h, --help                           Show this message and exit.
```

The NICE DCV SM CLI Command Reference also contains the help content for all NICE DCV SM CLI commands. All commands in the NICE DCV SM CLI correspond to requests made to the APIs of the broker. Each API has an API reference that can be found on the NICE DCV Session Manager Developer Guide section.

Command structure

This topic covers how DCV SM Command Line Interface (DCV SM CLI) command is structured. The DCV SM CLI uses a multipart structure on the command line that must be specified in this order:

1. The base call to the `./dcvsm` program.
2. The `options` which corresponds to common options that's used by all the commands and you can specify these in any order.
3. The `commands` that specifies which operation to perform.
4. The `args` corresponds to the arguments required by the command.

Syntax

```
dcvsm [options] commands args
```

Arguments can take various types of input values. These include numbers, strings, boolean. What is supported is dependent upon the command that you specify.

Parameter types

This section describes some of the common parameter types and the typical required format. If you're having trouble formatting an option or argument for a specific command, check the help by entering --help after the command name. The help for each command describes its function and options accepted.

String

String parameters can contain letters, numbers,and white space. Strings that contain white space must be surrounded by quotation marks. Strings can be surrounded by single quotation marks,

double quotation marks, or without them. We recommend that you don't use symbols because it can cause unexpected results.

String

One or more strings separated by comma and surrounded by quotation marks. The following example shows a list of `--session-ids`.

```
dcvsm describe-sessions --session-ids "session123,session456"
```

Boolean

Binary flag that turns an option on or off. For example, --debug describe-sessions has a Boolean --debug option that, when specified, calls the API enabling the debugging.

```
dcvsm --debug describe-sessions
```

If you don't specify explicitly a boolean option, it's set to FALSE by default.

Integer

An unsigned, whole number.

```
dcvsm describe-sessions --max-results 2
```

Return codes

The return code is usually but not always a hidden code sent after running a DCV SM Command Line Interface (DCV SM CLI) command, which describes the status of the command. You can use the echo command to display the code sent from the last DCV SM CLI command and use these codes to determine if a command was successful or if it failed, and why a command may have an error. In addition to the return codes, you can view more details about a failure by running your commands with the --debug switch. This switch produces a detailed report of the steps the DCV SM CLI uses to process the command, and what the result of each step was. To determine the return code of a DCV SM CLI command, run one of the following commands immediately after running the CLI command.

- Linux and macOS

```
$  echo $?
0
```

- Windows PowerShell

```
$  echo $lastexitcode
0
```

- Windows command prompt

```
$  echo %errorlevel%
0
```

The following return codes appear at the end of execution of a CLI command:

- 0 The command was successful. There were no errors thrown by either the CLI or by the service that the request was made to.

- 1 The configuration file parsed to the CLI was not found or might be corrupt.

- 2 The command entered on the command line failed to be parsed. Parsing failures can be caused by, but are not limited to, missing any required subcommands or arguments or using any unknown commands or arguments. Note that this return code meaning is applicable to all CLI commands.

- 130 The process received a SIGINT (Ctrl-C).

- 252 Command syntax was invalid, an unknown parameter was provided, or a parameter value was incorrect and prevented the command from running.

- 253 The system environment or configuration was invalid. While the command provided may be syntactically valid, missing configuration or credentials prevented the command from running.

- 254 The command was successfully parsed and a request was made to the specified service but the service returned an error. This will generally indicate incorrect API usage or other service specific issues.

- 255 General catch-all error. The command may have parsed correctly but an unspecified runtime error occurred when running the command. Because this is a general error code, an error may change from 255 to a more specific return code. A return code of 255 shouldn't be relied on to determine a specific error case.

NICE DCV Session Manager CLI reference

This section explains how to use the NICE DCV Session Manager command line interface (CLI) commands.

Use the following commands to interact with the NICE DCV server.

- open-servers
- describe-servers
- close-servers

Use the following commands to interact with the NICE DCV session.

- create-session
- describe-sessions
- get-session-connection-data
- get-session-screenshots
- update-session-permissions
- delete-session

close-servers

Closes one or more NICE DCV servers. When you close a NICE DCV server, you make it unavailable for NICE DCV session placement. You can't create NICE DCV sessions on closed servers. Closing a server ensures that no sessions are running on it and that users can't create new sessions on it.

Topics

- Synopsis
- Options
- Example

Synopsis

```
close-servers
```

```
--server-ids <value>
[--force]
```

Options

--server-ids

The comma-separated list of IDs of the NICE DCV servers to close.

Type: String

Required: Yes

--force

Operation that forces the server to close. By default, this is disabled.

Type: Boolean

Required: No

Example

```
dcvsm close-servers --server-ids "server123,server456"
```

create-session

Creates a new NICE DCV session with the specified details.

Topics

- Synopsis
- Options
- Example

Synopsis

```
create-session
--name <value>
--owner <value>
```

```
--type <value>
[--init-file-path <value>]
[--autorun-file <value>]
[--autorun-file-arguments <value>]
[--max-concurrents-clients <value>]
[--dcv-gl-enabled]
[--permissions-file <value>]
[--requirements <value>]
[--storage-root <value>]
[--enqueue-request <value>]
```

Options

--name

The name of the session.

Type: String

Required: Yes

--owner

The name of the session owner.

Type: String

Required: Yes

--type

The session type. This is either `Console` or `Virtual`.

Type: Choice

Required: Yes

--init-file-path

The path to the custom script on the NICE DCV server to run for initializing the session when it's created. The file path is relative to the `init` directory specified for the `agent.init-folder` Agent configuration parameter.

Type: String

Required: No

--autorun-file

The path to a file on the host server that's to be run inside the session. The file path is relative to the `autorun` directory specified for the `agent.autorun-folder` Agent configuration parameter.

Type: String

Required: No

--autorun-file-arguments

The command-line arguments passed to `AutorunFile` when it's run inside the session.

Type: String

Required: No

--max_concurrent-clients

The maximum number of concurrent NICE DCV clients. If provided, this must be between 1 and 100.

Type: Integer

Required: No

--dcv-gl-enabled

Configures the virtual session to use hardware-based OpenGL. It's disabled by default.

Type: Boolean

Required: No

--permissions-file

The path to the permissions file.

Type: String

Required: No

--requirements

The requirements that the server must meet to place the session.

Type: String

Required: No

--storage-root

The path to the folder used for session storage.

Type: String

Required: No

--enqueue-request

Indicates whether to queue the request if it cannot be immediately fulfilled. Requests are not enqueued by default.

Type: Boolean

Required: No

Example

```
dcvsm create-session --name session123 --owner sessionOwner
--type Console --requirements "server:Host.Os.Family = 'windows'"
```

delete-session

Deletes the specified NICE DCV session, and removes it from the cache of the broker.

Topics

- Synopsis
- Options
- Example

Synopsis

```
delete-session
--session-id <value>
--owner <value>
[--force]
```

Example   20

Options

--session-id

The ID of the session to delete.

Type: String

Required: Yes

--owner

The owner of the session to delete.

Type: String

Required: Yes

--force

The operation to ensure a session is removed from the cache of the broker. By default, this is disabled.

Type: Boolean

Required: No

Example

```
dcvsm delete-session --session-id session123 --owner sessionOwner
--force
```

describe-servers

Describe the specified NICE DCV server.

Topics

- [Synopsis](#)
- [Options](#)
- [Example](#)

Synopsis

```
describe-servers
[--server-ids <value>]
[--next-token <value>]
[--max-results <value>]
```

Options

--server-ids

The comma-separated list of IDs of the NICE DCV servers to describe.

Type: String

Required: No

--next-token

The token to use to retrieve the next page of results.

Type: String

Required: No

--max-results

The maximum number of results to be returned by the request in paginated output. If provided, this must be a number between 1 and 1000.

Type: Integer

Required: No

Example

```
dcvsm describe-servers --server-ids "server123,server456"
```

describe-sessions

Describes one or more NICE DCV servers.

Topics

- [Synopsis](#)
- [Options](#)
- [Example](#)

Synopsis

```
describe-sessions
[--session-ids <value>]
[--next-token <value>]
[--owner <value>]
[--max-results <value>]
```

Options

--session-ids

The comma-separated list of IDs of the NICE DCV sessions to describe.

Type: String

Required: No

--next-token

The token to retrieve the next page of results.

Type: String

Required: No

--owner

The owner of the session to describe.

Type: String

Required: No

--max-results

The number of results to show. If provided, must be between 1 and 1000.

Type: Integer

Required: No

Example

```
dcvsm describe-sessions --session-ids "session123,session456"
```

get-session-connection-data

Gets connection information for a specific user's connection to a specific NICE DCV session.

Topics

- [Synopsis](#)
- [Options](#)
- [Example](#)

Synopsis

```
get-session-connection-data
--session-id <value>
--user <value>
```

Options

`--session-id`

The ID of the NICE DCV sessions to get connection data from.

Type: String

Required: Yes

`--user`

The name of the user to view connection information for.

Type: Boolean

Example

24

Required: Yes

Example

```
./dcvsm get-session-connection-data --session-id session123
--user dcvuser
```

get-session-screenshots

Gets screenshots of one or more NICE DCV sessions.

Topics

- [Synopsis](#)
- [Options](#)
- [Example](#)

Synopsis

```
get-session-screenshots
--session-ids <value>
```

Options

--session-ids

The comma-separated list of IDs of the NICE DCV sessions to get the screenshots from.

Type: String

Required: Yes

Example

```
./dcvsm get-session-screenshots --session-id session123
```

Example

25

open-servers

Opens one or more NICE DCV servers. Before you can create sessions on a NICE DCV server, you must change the server's state to open. After the NICE DCV server is open, you can create NICE DCV sessions on the server.

Topics

- Synopsis
- Options
- Example

Synopsis

```
open-servers
--server-ids <value>
```

Options

--server-ids

    The comma-separated list of IDs of the NICE DCV servers to open.

    Type: String

    Required: Yes

Example

```
./dcvsm open-servers --server-ids "server123,server456"
```

update-session-permissions

Updates the user permissions for a specific NICE DCV session.

Topics

- Synopsis

- [Options](#)

- [Example](#)

Synopsis

```
update-session-permissions
--session-id <value>
--owner <value>
--permission-file <value>
```

Options

--session-id

The ID of the session to change the permissions for.

Type: String

Required: Yes

--owner

The owner of the session to change the permissions for.

Type: String

Required: Yes

--permission-file

The path to the permissions file.

Type: String

Required: Yes

Example

```
./dcvsm update-session-permissions --session-id session123
--owner Owner --permission-file path_to_file
```

Release notes and document history for NICE DCV Session Manager

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

Topics

- [NICE DCV Session Manager release notes](#)
- [Document history](#)

NICE DCV Session Manager release notes

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

Topics

- [2023.1— November 9, 2023](#)
- [2023.0-15065— May 4, 2023](#)
- [2023.0-14852— March 28, 2023](#)
- [2022.2-13907— November 11, 2022](#)
- [2022.1-13067— June 29, 2022](#)
- [2022.0-11952— February 23, 2022](#)
- [2021.3-11591— December 20, 2021](#)
- [2021.2-11445— November 18, 2021](#)
- [2021.2-11190— October 11, 2021](#)
- [2021.2-11042— September 01, 2021](#)
- [2021.1-10557— May 31, 2021](#)
- [2021.0-10242— April 12, 2021](#)
- [2020.2-9662— December 04, 2020](#)
- [2020.2-9508— November 11, 2020](#)

2023.1— November 9, 2023

Build numbers	Changes and bug fixes
• Broker: 410 • Agent: 732 • CLI: 140	• Bug fixes and performance improvements

2023.0-15065— May 4, 2023

Build numbers	Changes and bug fixes
• Broker: 392 • Agent: 675 • CLI: 132	• Added support for Red Hat Enterprise Linux 9, Rocky Linux 9, and CentOS Stream 9 on ARM platforms.

2023.0-14852— March 28, 2023

Build numbers	Changes and bug fixes
• Broker: 392 • Agent: 642 • CLI: 132	• Added support for Red Hat Enterprise Linux 9, Rocky Linux 9, and CentOS Stream 9.

2022.2-13907— November 11, 2022

Build numbers	Changes and bug fixes
• Broker: 382 • Agent: 612 • CLI: 123	• Added a `Substate` field in `DescribeSessions` response. • Fixed a problem that could cause the CLI to fail to connect to the broker depending on the URL in use.

2022.1-13067— June 29, 2022

Build numbers	Changes and bug fixes
• Broker: 355 • Agent: 592 • CLI: 114	• Added support to run the broker on AWS Graviton instances. • Added agent and broker support for Ubuntu 22.04.

2022.0-11952— February 23, 2022

Build numbers	Changes and bug fixes
• Broker: 341 • Agent: 520 • CLI: 112	• Added log rotation capability to the Agent. • Added configuration parameter to set Java home in the Broker. • Improved data flushing from cache to disk in the Broker. • Fixed URL validation in the CLI.

2021.3-11591— December 20, 2021

Build numbers	New features
• Broker: 307 • Agent: 453 • CLI: 92	• Added support for integrating with the NICE DCV Connection Gateway. • Added Broker support for Ubuntu 18.04 and Ubuntu 20.04.

2021.2-11445— November 18, 2021

Build numbers	Changes and bug fixes
• Broker: 288 • Agent: 413 • CLI: 54	• Fixed a problem with the validation of login names which include a Windows domain.

2021.2-11190— October 11, 2021

Build numbers	Changes and bug fixes
• Broker: 254 • Agent: 413 • CLI: 54	• Fixed a problem in the command line interface which prevented from launching Windows sessions.

2021.2-11042— September 01, 2021

Build numbers	New features	Changes and bug fixes
• Broker: 254 • Agent: 413 • CLI: 37	• NICE DCV Session Manager now offers command line interface (CLI) support. You can create and manage NICE DCV sessions in the CLI, instead of calling APIs. • NICE DCV Session Manager introduced Broker data persistence. For higher availability, brokers can persist server state information on an external data store and restore the data at startup.	• When registering an external authorization server, you can now specify the algorithm that the authorization server uses to sign JSON-formatted Web Tokens. With this change, you can use Azure AD as an external authorization server.

2021.1-10557— May 31, 2021

Build numbers	New features	Changes and bug fixes
• Broker: 214 • Agent: 365	• NICE DCV Session Manager added support for input parameters passed to the autorun file on Linux.	• We fixed a problem with the autorun file on Windows.

Build numbers	New features	Changes and bug fixes
	• Server properties can now be passed as requirements to the CreateSessions API.	

2021.0-10242— April 12, 2021

Build numbers	Changes and bug fixes

• Broker: 183
• Agent: 318
	• NICE DCV Session Manager introduced the following new APIs:
	  • OpenServers
	  • CloseServers
	  • DescribeServers
	  • GetSessionScreenshots
	• It also introduced the following new configuration parameters:
	  • Broker parameters: `session-screenshot-max-width` , `session-screenshot-max-height` , `session-screenshot-format` , `create-sessions-queue-max-size` , and `create-sessions-queue-max-time-seconds` .
	  • Agent parameters: `agent.autorun_folder` , `max_virtual_sessions` , and `max_concurrent_sessions_per_user` .
	
	  Agent parameters: `agent.autorun_folder` , `max_virtual_sessions` , and `max_concurrent_sessions_per_user` .
	
	  Agent parameters: `agent.autorun_folder` , `max_virtual_sessions` , and `max_concurrent_sessions_per_user` .

2020.2-9662— December 04, 2020

Build numbers	Changes and bug fixes
• Broker: 114 • Agent: 211	• We fixed a problem with the auto-generated TLS certificates that prevented the Broker from starting.

2020.2-9508— November 11, 2020

Build numbers	Changes and bug fixes
• Broker: 78 • Agent: 183	• The initial release of NICE DCV Session Manager.

Document history

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

Change	Description	Date
NICE DCV Version 2023.1	NICE DCV Session Manager has been updated for NICE DCV 2023.1. For more information, see 2023.1— November 9, 2023.	November 9, 2023
NICE DCV Version 2023.0	NICE DCV Session Manager has been updated for NICE DCV 2023.0. For more information, see 2023.0-14852— March 28, 2023.	March 28, 2023
NICE DCV Version 2022.2	NICE DCV Session Manager has been updated for NICE DCV 2022.2. For more information, see 2022.2-13907— November 11, 2022.	November 11, 2022

Change	Description	Date
NICE DCV Version 2022.1	NICE DCV Session Manager has been updated for NICE DCV 2022.1. For more information, see 2022.1-13067—June 29, 2022.	June 29, 2022
NICE DCV Version 2022.0	NICE DCV Session Manager has been updated for NICE DCV 2022.0. For more information, see 2022.0-11952—February 23, 2022.	February 23, 2022
NICE DCV Version 2021.3	NICE DCV Session Manager has been updated for NICE DCV 2021.3. For more information, see 2021.3-11591—December 20, 2021.	December 20, 2021
NICE DCV Version 2021.2	NICE DCV Session Manager has been updated for NICE DCV 2021.2. For more information, see 2021.2-11042—September 01, 2021.	September 01, 2021
NICE DCV Version 2021.1	NICE DCV Session Manager has been updated for NICE DCV 2021.1. For more information, see 2021.1-10557—May 31, 2021.	May 31, 2021
NICE DCV Version 2021.0	NICE DCV Session Manager has been updated for NICE DCV 2021.0. For more information, see 2021.0-10242—April 12, 2021.	April 12, 2021
Initial release of NICE DCV Session Manager	The first publication of this content.	November 11, 2020