
NICE DCV

管理员指南

亚马逊云科技



NICE DCV: 管理员指南

Table of Contents

什么是 NICE DCV ?	1
NICE DCV的工作原理	1
功能 :	1
定价	2
NICE DCV 服务器	3
Requirements	3
支持的功能	4
设置	6
安装	6
Windows	6
Linux	9
许可	28
NICE DCV 许可要求	28
安装延期评估许可证	30
安装生产许可证	31
升级	38
在 Windows 上升级 NICE DCV 服务器	38
在 Linux 上升级NICE DCV 服务器	39
管理服务器	40
启动服务器	40
在 Windows 上启动服务器	40
在 Linux 上启动服务器	41
停止服务器	41
在 Windows 上停止服务器	42
在 Linux 上停止服务器	42
更改 TCP 端口	43
在 Windows 上更改服务器的 TCP 端口	43
在 Linux 上更改服务器 TCP 端口	43
断开空闲客户端连接	44
在 Windows 上更改空闲超时期限	44
在 Linux 上更改空闲超时期限	45
在 Linux 上启用 GPU 共享	45
更改 TLS 证书	46
启用 USB 远程处理	47
在 Windows 上将设备添加到允许列表	47
在 Linux 上将设备添加到允许列表	48
配置智能卡缓存	48
启用会话存储	49
在 Windows 上启用会话存储	49
在 Linux 上启用会话存储	50
在 Linux 上配置剪贴板	50
将客户端剪贴板内容粘贴到主选区	50
将主选区内容复制到客户端剪贴板	51
启用触摸屏和触控笔支持	51
配置多通道音频	53
在 Windows NICE DCV 服务器上配置音频通道	54
在 Linux NICE DCV 服务器上配置音频通道	54
启用 QUIC UDP	55
配置 HTTP 标头	56
在 Windows 上配置 HTTP 标头NICE DCV 服务器	56
在 Linux 上配置 HTTP 标头NICE DCV 服务器	57
配置身份验证	57
在 Windows 上配置身份验证	58
在 Linux 上配置身份验证	59

配置授权	59
默认权限文件	59
自定义权限文件	59
使用权限文件	60
管理会话	64
NICE DCV 会议简介	64
控制台会话	64
虚拟会话	65
使用命令行工具管理会话	65
在 Windows NICE DCV 服务器上使用命令行工具	65
在 Linux NICE DCV 服务器上使用命令行工具	66
启动 会话	66
手动启动控制台和虚拟会话	66
启用自动控制台会话	68
停止会话	69
管理正在运行的 会话	69
管理会话存储	69
管理会话授权	70
管理会话显示布局	71
管理会话名称	72
查看会话	73
如何	75
使用外部身份验证	75
NICE DCV 服务器 配置	75
使用令牌	76
身份验证服务要求	76
查找并停止空闲会话	77
启用到 X 服务器的远程 X 连接	78
启用到 X 服务器的远程 X 连接	78
将 NICE DCV Web 浏览器客户端嵌入 iFrame 中	79
故障排除。	81
使用日志文件	81
在 Windows 上更改日志文件详细程度	81
在 Linux 上更改日志文件详细程度	82
常见问题	82
Windows NICE DCV 服务器 上的光标问题	82
参数参考	83
connectivity 参数	83
session-management 参数	85
session-management/defaults 参数	86
session-management/automatic-console-session 参数	86
security 参数	87
license 参数	91
input 参数	91
display 参数	92
display/linux 参数	94
audio 参数	95
log 参数	95
windows 参数	96
clipboard 参数	97
smartcard 参数	98
修改配置参数	98
Windows NICE DCV 服务器s	99
Linux NICE DCV 服务器s	99
NICE DCV 终止支持使用期限	101
EOL 时间轴	101
客户的 ESL 路径	101

ESLFAQs	101
发布说明和文档历史记录	103
发布说明	103
DCV 2020.2-9662	103
DCV 2020.2.958	104
DCV 2020.1-9012	104
DCV 2020.1-9012	105
DCV 2020.1-8942	105
DCV 2020.0-8428	106
DCV 2019.1-7644	106
DCV 2019.1-7423	107
DCV 2019.0-7318	107
DCV 2017.4-6898	108
DCV 2017.3-6698	108
DCV 2017.2-6182	110
DCV 2017.1-5870	110
DCV 2017.1-5777	111
DCV 2017.0-5600	111
DCV 2017.0-5121	111
DCV 2017.0-4334	112
DCV 2017.0-4100	112
文档历史记录	112
.....	CXIV

什么是 NICE DCV ?

NICE DCV 是一种高性能远程显示协议。它允许您在不同的网络条件下，将远程桌面和应用程序流从任何云或数据中心安全地传送到任何设备。通过将 NICE DCV 与 Amazon EC2 结合使用，您可以在 Amazon EC2 实例上远程运行图形密集型应用程序。然后，您可以将结果流式传输到更适中的客户端计算机，从而消除对昂贵的专用工作站的需求。

主题

- [NICE DCV 的工作原理 \(p. 1\)](#)
- [NICE DCV 功能 \(p. 1\)](#)
- [NICE DCV 定价 \(p. 2\)](#)

NICE DCV 的工作原理

要使用 NICE DCV，请在服务器上安装 NICE DCV 服务器软件。NICE DCV 服务器软件用于创建安全会话。在服务器上安装并运行应用程序。服务器使用其硬件执行安装的应用程序所需的高性能处理。您的用户通过使用 NICE DCV 客户端应用程序远程连接到会话来访问应用程序。建立连接后，NICE DCV 服务器软件会压缩应用程序的可视输出，并以加密的像素流形式将其流式传输回客户端应用程序。客户端应用程序接收压缩的像素流，对其进行解密，然后将其输出到本地显示。

NICE DCV 功能

NICE DCV 提供以下功能：

- 共享整个桌面 — 使用高性能 NICE DCV 协议共享整个远程桌面的完全控制权。
- 仅传输图像 — 将渲染的图像作为像素传输，而不传输几何体和场景信息。这就提供了另外一层安全性，因为此操作不会通过网络发送专有客户信息。
- 支持基于 H.264 的编码 — 使用基于 H.264 的视频压缩和编码，以降低带宽消耗。
- 支持无损质量的视频压缩 - 支持无损质量视频压缩（在网络和处理器条件允许的情况下）。
- 匹配显示布局 — 自动适应服务器的屏幕分辨率和显示布局，以匹配客户端窗口的大小。
- 支持多屏幕 — 允许您将会话桌面扩展至最多四个显示器。
- 适应压缩级别 — 自动根据网络的可用带宽和延迟适应视频压缩级别。
- 启用协作 — 提供支持多个协作客户端的动态会话。客户端可以在会话期间随时连接和断开连接。
- 支持每个服务器拥有多个会话（仅限 Linux NICE DCV 服务器）— 支持每个 Linux NICE DCV 服务器拥有多个虚拟会话，以最大限度地节省成本。
- 支持 GPU 共享（仅限 Linux NICE DCV 服务器）— 可让您在 Linux NICE DCV 服务器上运行的多个虚拟会话之间共享一个或多个物理 GPU。
- 支持 USB、智能卡和触控笔远程处理 — 可让您在 NICE DCV 会话中使用外围设备，就像在本地计算机上一样。
- 支持音频输入和输出、打印以及复制和粘贴 — 可让您在会话和本地计算机之间执行这些关键操作。
- 支持文件传输 — 可让您在会话和本地计算机之间传输文件。
- 提供 HTML5 客户端 - 提供一个 HTML5 客户端，它可在 Windows 和 Linux 上与任何现代 Web 浏览器结合使用。
- 支持现代 Linux 桌面环境 — 支持现代 Linux 桌面，例如 RHEL8 上的 Gnome3。

NICE DCV 定价

在 Amazon EC2 实例上使用 NICE DCV 服务器不收取任何额外费用。您需要为您使用的实例和其他 Amazon EC2 功能支付标准费用。

需要有许可证才能在本地服务器或其他基于云的服务器上安装 NICE DCV 服务器。有关更多信息，请参阅 [许可 NICE DCV 服务器 \(p. 28\)](#)。))

NICE DCV 服务器

NICE DCV 服务器适用于 Windows 和 Linux。这两个服务器提供了类似的功能，但有一些差异。选择最适合您的需要的 NICE DCV 服务器。下表比较了 Windows 和 Linux NICE DCV 服务器支持的功能。

主题

- [Requirements \(p. 3\)](#)
- [支持的功能 \(p. 4\)](#)

Requirements

为获得良好的 NICE DCV 用户体验，请确保服务器满足以下最低要求。请记住，您的用户体验在很大程度上取决于从 NICE DCV 服务器流式传输到 NICE DCV 客户端的像素数。

如果您正在 Amazon EC2 实例上安装 NICE DCV 服务器，我们建议您使用 Amazon EC2 G3 或 G4 实例类型。这些实例类型提供支持基于硬件的 NVIDIA 和 GPU 共享的 GPUs OpenGL。有关更多信息，请参阅 [Amazon EC2 G3 实例](#) 和 [Amazon EC2 G4 实例](#)。您可以在任何其他实例类型上安装 NICE DCV 服务器，但可能存在屏幕分辨率限制。要绕过此限制，请下载并安装 [适用于 EC2 的 NICE DCV 虚拟显示驱动程序](#)。

您的服务器必须满足下表中列出的最低要求。

	Windows 服务器	Linux 服务器
操作系统	<ul style="list-style-type: none"> • Windows 7 — 需要兼容 NvFBC 的 NVIDIA GPU • Windows Server 2008 R2 — 需要兼容 NvFBC 的 NVIDIA GPU • Windows 8.1 • Windows Server 2012 R2 • Windows 10 • Windows Server 2016 • Windows Server 2019 <p>Note</p> <p>所有支持的 Windows 操作系统都需要 .NET Framework 4.5，并且必须支持 x86-64 架构。</p>	<ul style="list-style-type: none"> • Amazon Linux 2 • CentOS 6.7 或更高版本 • CentOS 7.5 或更高版本 • CentOS 8.x • RHEL 6.7 或更高版本 • RHEL 7.5 或更高版本 • RHEL 8 .x • SUSE Linux Enterprise 12 (带 SP2) 或更高版本 • SUSE Linux Enterprise 15 - • Ubuntu 18.04 • Ubuntu 20.04
支持的架构	64 位 x86	<ul style="list-style-type: none"> • 64 位 x86 • 64 位 ARM (仅对于运行 Amazon EC2 的 Amazon Linux 2 实例、CentOS 7.x/8.x、RHEL 7.x/8.x 和 Ubuntu 18.04 支持)
GPU	(可选) 基于硬件的视频编码需要一个支持 NVENC 的 NVIDIA GPU。在 Ubuntu 20 上，需要具有计算功能 ≥ 3.5 的 GPU。如果您的服务器没有 NVIDIA GPU，或者其 NVIDIA GPU 不支持 NVENC 编码，或者没有所需的计算功能，则使用基于软件的视频编码。	

	Windows 服务器	Linux 服务器
		对于跨虚拟会话的 GPU 共享，需要一个 NVIDIA GPU。
Network	默认情况下，NICE DCV 服务器 通过端口 8443 进行通信。该端口是可配置的，但必须大于 1024。确保服务器允许通过所需的端口进行通信。	

有关 NICE DCV 客户端 要求的更多信息，请参阅 [用户指南NICE DCV 客户端](#) 中的 [要求NICE DCV](#)。

支持的功能

下表比较了 Windows 和 Linux NICE DCV 服务支持的功能。

功能	Windows NICE DCV 服务器 (p. 6)	Linux NICE DCV 服务器 (p. 9)
控制台会话 (p. 64)	✓	✓
虚拟会话 (p. 64)	✗	✓
自定义 TCP 端口 (p. 43)	✓	✓
空闲客户端断开连接 (p. 44)	✓	✓
GPU 共享 (p. 45)	✗	✓
自定义 TLS 证书 (p. 46)	✓	✓
USB 远程控制 (p. 47)	✓	✓
智能卡支持 (p. 48)	✓	✓
会话存储和文件传输 (p. 49)	✓	✓
复制和粘贴 (p. 50)	✓	✓
自定义 HTTP 标头 (p. 56)	✓	✓
从会话打印	✓	✓
立体声 2.0 音频播放	✓	✓ (不受 CentOS 6 支持)
环绕立体声音频播放	✓ (最高 7.1)	✓ (最高 5.1)
立体声 2.0 音频录制	✓	✗
触摸屏支持 (p. 51)	✓ (Windows 8 和 Windows Server 2012 及更高版本)	✓
触控笔支持 (p. 51)	✓ (Windows 10 和 Windows Server 2019)	✓

有关 NICE DCV 客户端 功能的更多信息，请参阅 [用户指南NICE DCV 客户端 中的 功能NICE DCV](#)。

设置 NICE DCV 服务器

要使用 NICE DCV，请在打算托管 NICE DCV 会话的服务器上安装 NICE DCV 服务器软件，并确保它已得到正确许可。

以下主题介绍如何安装和许可 NICE DCV 服务器。许可主题仅适用于本地服务器和其他基于云的服务器上的安装，因为您不需要许可证即可在 (p. 28) 实例上使用 NICE DCV 服务器。Amazon EC2

主题

- [安装 NICE DCV 服务器 \(p. 6\)](#)
- [许可 NICE DCV 服务器 \(p. 28\)](#)
- [升级 NICE DCV 服务器 \(p. 38\)](#)

安装 NICE DCV 服务器

以下主题说明如何在 Windows 和 Linux 上安装最新版本的 NICE DCV 服务器。如果您在 NICE DCV 实例上安装 Amazon EC2，或者在本地服务器或其他云端服务器上安装，请执行这些步骤。

Note

如果要从早期版本的 NICE DCV 服务器升级到最新版本，请参阅[升级 NICE DCV 服务器 \(p. 38\)](#)。

主题

- [在 Windows 上安装 NICE DCV 服务器 \(p. 6\)](#)
- [在 Linux 上安装 NICE DCV 服务器 \(p. 9\)](#)

在 Windows 上安装 NICE DCV 服务器

此部分介绍如何在 Windows 上安装 NICE DCV 服务器

主题

- [Amazon EC2 实例上的 Windows NICE DCV 服务器的先决条件 \(p. 6\)](#)
- [在 Windows 上安装 NICE DCV 服务器 \(p. 7\)](#)

Amazon EC2 实例上的 Windows NICE DCV 服务器的先决条件

本主题介绍如何在安装 Amazon EC2 之前配置 Windows NICE DCV 服务器实例。如果您没有在 NICE DCV 服务器 Windows 实例上安装 Amazon EC2，请跳过这些先决条件。

主题

- [加速计算实例的先决条件 \(p. 6\)](#)
- [其他实例系列的先决条件 \(p. 7\)](#)

加速计算实例的先决条件

GPU 图形实例的先决条件

如果您使用的是 GPU 图形实例（例如 G2、G3 或 G4），我们建议您安装和配置相应的 NVIDIA GPU 驱动程序。NVIDIA GPU 驱动程序支持：

- 适用于应用程序的 DirectX 和 OpenGL 硬件加速

- 适用于 H.264 视频流编码的硬件加速
- 可自定义的服务器监视器分辨率
- 提高了服务器监视器的最大分辨率 — 高达 4096x2160
- 增加了服务器监视器的数量

有关在 GPU 图形实例上安装 NVIDIA GPU 驱动程序的更多信息，请参阅 [中的在 Windows 上安装 NVIDIA 驱动程序](#)。Amazon EC2 用户指南（适用于 Windows 实例）

其他加速计算实例的先决条件

如果您使用的加速计算实例不是 GPU 图形实例（例如 P2、P3 或 P3dn），我们建议您安装和配置相应的 NVIDIA GPU 驱动程序。NVIDIA GPU 驱动程序支持 H.264 视频流编码的硬件加速。

有关在加速计算实例上安装 NVIDIA GPU 驱动程序的更多信息，请参阅 [中的公共 NVIDIA 驱动程序](#)。Amazon EC2 用户指南（适用于 Windows 实例）

与 GPU 图形实例不同，在加速计算实例上安装 NVIDIA GPU 驱动程序不会以任何方式提高服务器监视器限制或分辨率。要添加其他服务器监视器分辨率支持，您可以安装 NVIDIA GRID 驱动程序。有关更多信息，请参阅 [网站上的 NVIDIA vGPU 软件](#) NVIDIA。

其他实例系列的先决条件

对于加速计算实例之外的实例（例如通用型、计算优化实例、内存优化实例和存储优化实例系列中的实例），我们建议您安装 NICE DCV Virtual Display 驱动程序。

安装 NICE DCV 虚拟显示驱动程序将执行以下操作：

- 将最大服务器监视器数提高到 4。
- 启用自定义服务器监视器分辨率。
- 将支持的最大服务器监视器分辨率提高到 4K。

无法使用 Windows 控制面板管理由 NICE DCV 服务器连接的服务器监视器。

Note

Windows Server 2012 R2 及更高版本上支持 NICE DCV 虚拟显示驱动程序。

Important

使用任何其他 GPU 驱动程序（如 NICE DCV GPU 驱动程序）安装 NVIDIA Virtual Display 驱动程序可能会导致冲突。为避免冲突，我们建议您不要将 NICE DCV Virtual Display 驱动程序与任何其他 GPU 驱动程序一起安装。

在实例上安装 NICE DCV Virtual Display 驱动程序

1. 从 NICE DCV [网站](#) [NICE DCV 下载](#) Virtual Display 驱动程序安装程序。
2. 要通过运行向导安装驱动程序，请打开或双击安装文件。或者，使用以下命令运行自动安装。

```
C:\> nice-dcv-virtual-display-x64-Release-34.msi /quiet /norestart
```

3. 重启实例，然后重新连接到它。

在 Windows 上安装 NICE DCV 服务器

可以使用安装向导在 Windows 主机服务器上安装 NICE DCV 服务器。此向导将指导您完成一系列步骤，使您可以自定义 NICE DCV 服务器安装。也可以使用命令行来执行自动安装，这种方式使用默认设置来自动完成安装过程。

Note

NICE DCV 服务器 安装程序由 SHA-256 证书签名。如果您使用的是运行 Windows 7 或 Windows Server 2008 R2 的服务器，则必须安装 Microsoft 安全更新以支持此证书类型。有关更多信息，请参阅 [Microsoft 安全公告 3033929](#) 和 [Microsoft Windows 支持](#)。

目录

- [使用向导 \(p. 8\)](#)
- [自动安装 \(p. 8\)](#)

使用向导

使用 NICE DCV 服务器 安装向导执行指导式安装。

使用向导在 Windows 上安装 NICE DCV 服务器

1. 启动并连接到要安装 的服务器。NICE DCV 服务器。
2. 从 NICE DCV 服务器[NICE 网站下载](#) 安装程序。

Note

NICE DCV 服务器仅提供 64 位版本，并且在 64 位 Windows 操作系统上受支持。

3. 运行 `nice-dcv-server-x64-Release-2020.2-version_number.msi`。
4. 在欢迎屏幕上，选择 Next。
5. 在“End-User License Agreement (最终用户许可协议)”屏幕上，阅读许可协议。如果您接受这些条款，请选中 I accept the terms in the License Agreement (我接受许可协议中的条款) 复选框，然后选择 Next (下一步)。
6. (可选) 在驱动程序选择屏幕上，选择 USB 设备远程处理，然后依次选择将安装在本地硬盘驱动器上和下一步。这将安装支持某些专用 USB 设备所需的驱动程序，例如 3D 指点设备和绘图板。
7. 在“DCV Service Configuration (DCV 服务配置)”屏幕上：
 - a. (可选) 要手动配置您服务器的防火墙以允许通过所需端口进行通信，请选择 No, I will manually configure my firewall later (不，我将稍后手动配置我的防火墙)。
 - b. (可选) 要在安装之后手动启动 NICE DCV 服务器，请选择 No, I want to start a DCV Service manually (不，我想手动启动 DCV 服务)。如果您选择此选项，则无法在安装完成后自动启动控制台会话。如果选择此选项，则跳过步骤 9。
8. 选择 Next (下一步)。
9. 在“DCV Session Management Configuration (DCV 会话管理配置)”屏幕上，指定自动控制台会话的所有者。或者，要防止安装完成后启动自动控制台会话，请选择不，我将手动创建会话。

Note

仅当您以前选择了允许服务器自动启动的情况下，才完成此步骤。

10. 选择安装。

自动安装

默认情况下，自动安装执行以下操作：

- 添加防火墙规则以允许通过端口 8443 进行通信。
- 允许 NICE DCV 服务器自动启动。
- 创建自动控制台会话。
- 将控制台会话所有者设置为执行安装的用户。

您可以通过将以下选项附加到安装命令来覆盖默认操作：

- `DISABLE_FIREWALL=1` — 阻止安装程序添加防火墙规则。
- `DISABLE_SERVER_AUTOSTART=1` — 阻止 NICE DCV 服务器 在安装之后自动启动。
- `DISABLE_AUTOMATIC_SESSION_CREATION=1` — 阻止安装程序启动自动控制台会话。
- `AUTOMATIC_SESSION_OWNER=owner_name` — 为自动控制台会话指定不同的所有者。
- `ADDLOCAL=ALL` — 安装 USB 远程处理所需的 DCV 驱动程序。

使用自动安装功能在 Windows 上安装 NICE DCV 服务器

1. 启动并连接到要安装 的服务器。NICE DCV 服务器。
2. 从 NICE DCV 服务器[NICE 网站下载](#) 安装程序。

Note

NICE DCV 服务器仅提供 64 位版本，并且在 64 位 Windows 操作系统上受支持。

3. 打开命令提示符窗口，导航到下载安装程序的文件夹。
4. 运行自动安装程序：

```
C:\> msixexec.exe /i nice-dcv-server-x64-Release-2020.2-version_number.msi  
ADDLOCAL=ALL /quiet /norestart /!*v dcv_install_msi.log
```

在 Linux 上安装 NICE DCV 服务器

此部分介绍如何在 Linux 上安装 NICE DCV 服务器

主题

- [Linux NICE DCV 服务器的先决条件](#) (p. 9)
- [在 Linux 上安装 NICE DCV 服务器](#) (p. 17)
- [安装后检查](#) (p. 26)

Linux NICE DCV 服务器的先决条件

NICE DCV 允许客户端访问 Linux 服务器（提供对相应 Linux 桌面的访问）上的远程图形 X 会话。NICE DCV 支持两种 Linux 桌面流式处理：控制台会话和虚拟会话。有关控制台和虚拟会话的更多信息，请参阅[管理 NICE DCV 会话](#) (p. 64)。

本主题介绍如何准备好必需的先决条件以在 Linux 服务器上使用 NICE DCV

目录

- [安装桌面环境和桌面管理器](#) (p. 9)
- [禁用地貌协议（仅限 GDM3）](#) (p. 12)
- [配置 X 服务器](#) (p. 13)
- [安装 glxinfo 实用程序。](#) (p. 14)
- [验证 OpenGL 软件渲染](#) (p. 15)
- [安装和配置 NVIDIA 驱动程序](#) (p. 15)

安装桌面环境和桌面管理器

为帮助提高使用 Linux 服务器上 NICE DCV 的体验，您可以安装一个桌面环境和桌面管理器。

桌面环境是一个图形用户界面 (GUI)，有助于您与 Linux 操作系统交互。桌面环境有很多，NICE DCV 可与其中的大部分协作。桌面管理器是管理用户登录屏幕、启动和停止桌面环境会话和 X 服务器的项目。

以下选项卡式内容显示了在支持的操作系统上安装默认桌面环境和桌面管理器的步骤。

RHEL 7.x/8.x and CentOS 7.x/8.x

RHEL 7.x/8.x 和 CentOS 7.x/8.x 的默认桌面环境是 Gnome3，默认桌面管理器是 GDM。

在 RHEL 7.x/8.x 和 CentOS 7.x/8.x 上安装和配置桌面环境和桌面管理器

1. 安装桌面环境和桌面管理器程序包

- RHEL 7.x/8.x 和 CentOS 8.x

```
$ sudo yum groupinstall 'Server with GUI'
```

- CentOS 7.x

```
$ sudo yum groupinstall "GNOME Desktop"
```

2. 更新软件包来确保 Linux 服务器是最新的。

```
$ sudo yum upgrade
```

3. 重新启动 Linux 服务器。

```
$ sudo reboot
```

RHEL 6.x and CentOS 6.x

RHEL 6.x 和 CentOS 6.x 的默认桌面环境是 Gnome，默认桌面管理器是 GDM。

在 RHEL 6.x 和 CentOS 6.x 上安装和配置桌面环境和桌面管理器

1. 安装桌面环境和桌面管理器程序包

```
$ sudo yum groupinstall\ "X Window System" "Desktop" "General Purpose Desktop"
```

2. 更新软件包来确保 Linux 服务器是最新的。

```
$ sudo yum upgrade
```

3. 重新启动 Linux 服务器。

```
$ sudo reboot
```

Amazon Linux 2

Amazon Linux 2 的默认桌面环境是 Gnome3，默认桌面管理器是 GDM。

在 Amazon Linux 2 上安装和配置桌面环境和桌面管理器

1. 安装桌面环境和桌面管理器程序包

```
$ sudo yum install gdm gnome-session gnome-classic-session gnome-session-xsession
```

```
$ sudo yum install xorg-x11-server-Xorg xorg-x11-fonts-Type1 xorg-x11-drivers
```

```
$ sudo yum install gnome-terminal gnu-free-fonts-common gnu-free-mono-fonts gnu-free-sans-fonts gnu-free-serif-fonts
```

- 更新软件包来确保 Linux 服务器是最新的。

```
$ sudo yum upgrade
```

- 重新启动 Linux 服务器。

```
$ sudo reboot
```

Ubuntu 18.x/20.x

对于 Ubuntu 18.x，默认桌面环境为 Gnome3，默认桌面管理器为 GDM3。对于 Ubuntu 18.x，NICE DCV 控制台会话当前不支持 GDM3。因此，如果您计划使用 LightDM 控制台会话，我们建议您使用 NICE DCV 桌面管理器。

对于 Ubuntu 20.x，默认桌面环境为 Gnome3，默认桌面管理器为 GDM3。对于 Ubuntu 20.x，LightDM 控制台会话当前不支持 NICE DCV。因此，如果您计划使用 NICE DCV 控制台会话，我们建议您使用 GDM3 桌面管理器。

在 Ubuntu 18.x/20.x 上安装和配置桌面环境和桌面管理器

- 安装桌面环境和桌面管理器程序包

```
$ sudo apt update
```

```
$ sudo apt install ubuntu-desktop
```

安装 LightDM。

- Ubuntu 18.x

```
$ sudo apt install lightdm
```

- Ubuntu 20.x

```
$ sudo apt install gdm3
```

- (仅限 Ubuntu 20.x) 验证 GDM3 是否已设置为默认桌面管理器。

```
$ cat /etc/X11/default-display-manager
```

预期输出

```
/usr/sbin/gdm3
```

如果 GDM3 未设置为默认桌面管理器，请使用以下命令将其设置为默认桌面管理器。

```
$ sudo dpkg-reconfigure gdm3
```


- 更新软件包来确保 Linux 服务器是最新的。

```
$ sudo apt upgrade
```

- 重新启动 Linux 服务器。

```
$ sudo reboot
```

SUSE Linux Enterprise 12.x/15.x

SUSE Linux Enterprise 12.x 的默认桌面环境是 SLE Classic，默认桌面管理器是 GDM。

SUSE Linux Enterprise 15.x 的默认桌面环境是 SLE Classic，默认桌面管理器是 GDM3。NICE DCV 控制台会话当前不支持其他显示管理器，例如 LightDM。如果您计划使用 NICE DCV 控制台会话，我们建议您为 SUSE Linux Enterprise 15.x 使用 GDM3 桌面管理器。

在 SUSE Linux Enterprise 12.x 上安装和配置桌面环境和桌面管理器

- 安装桌面环境和桌面管理器程序包

- SUSE Linux Enterprise 12.x

```
$ sudo zypper install -t pattern gnome-basic
```

```
$ sudo sed -i "s/DISPLAYMANAGER=\"\"/DISPLAYMANAGER=\"gdm\"/" /etc/sysconfig/  
displaymanager
```

```
$ sudo sed -i "s/DEFAULT_WM=\"\"/DEFAULT_WM=\"sle-classic\"/" /etc/sysconfig/  
windowmanager
```

- SUSE Linux Enterprise 15.x

```
$ sudo zypper install -t pattern gnome_basic
```

```
$ sudo update-alternatives --set default-displaymanager /usr/lib/X11/  
displaymanagers/gdm
```

```
$ sudo sed -i "s/DEFAULT_WM=\"\"/DEFAULT_WM=\"gnome\"/" /etc/sysconfig/  
windowmanager
```

- 更新软件包来确保 Linux 服务器是最新的。

```
$ sudo zypper update
```

- 重新启动 Linux 服务器。

```
$ sudo reboot
```

禁用地貌协议 (仅限 GDM3)

NICE DCV 不支持路径协议。如果您使用的是 GDM3 桌面管理器，则必须禁用路径协议。如果您未使用 GDM3，请跳过此步骤。

禁用路径协议

1. 使用首选文本编辑器打开以下文件。

- RHEL 8.x、CentOS 8.x 和 SUSE Linux Enterprise 15.x

```
/etc/gdm/custom.conf
```

- Ubuntu 20.x

```
/etc/gdm3/custom.conf
```

2. 在 [daemon] 部分中，将 WaylandEnable 设置为 false。

```
[daemon]  
WaylandEnable=false
```

3. 重新启动 GDM 服务。

- RHEL 8.x 和 CentOS 8.x

```
$ sudo systemctl restart gdm
```

- Ubuntu 20.x

```
$ sudo systemctl restart gdm3
```

- SUSE Linux Enterprise 15.x

```
$ sudo systemctl restart xdm
```

配置 X 服务器

如果要使用控制台会话或 GPU 共享，则必须确保 Linux 服务器具有一个配置正确且正在运行的 X 服务器。

Note

如果要使用不带 GPU 共享的虚拟会话，则不需要 X 服务器。

X 服务器程序包通常作为桌面环境和桌面管理器的依赖项安装。我们建议您将 X 服务器配置为当 Linux 服务器启动时自动启动。

以下选项卡式内容说明了如何在支持的操作系统上配置和启动 X 服务器。

RHEL 6.x and CentOS 6.x

在 RHEL 6.x 和 CentOS 6.x 上配置和启动 X 服务器

1. 将 X 服务器配置为当 Linux 服务器启动时自动启动。

```
$ cat /etc/inittab | grep "id.*initdefault"
```

如果命令返回 id:5:initdefault，则表明 X 服务器已配置为自动启动。继续下一步。

如果命令返回 id:3:initdefault，则表明 X 服务器未配置为自动启动。运行以下命令。

```
$ sudo sed -i "s/id:3:initdefault:/id:5:initdefault:/" /etc/inittab
```

2. 启动 X 服务器。

```
$ sudo init 5
```

3. 验证 X 服务器是否正在运行。

```
$ ps aux | grep X | grep -v grep
```

下面显示 X 服务器在运行时的示例输出。

```
root      1891  0.0  0.7 277528 30448 tty7      Ssl+ 10:59   0:00 /usr/bin/Xorg :0  
-background none -verbose -auth /run/gdm/auth-for-gdm-wltseN/database -seat seat0  
vt7
```

RHEL 7.x/8.x, CentOS 7.x/8.x, Amazon Linux 2, Ubuntu 18.x/20.x, and SUSE Linux Enterprise 12.x/15.x

在 RHEL 7.x/8.x、CentOS 7.x/8.x、Amazon Linux 2、Ubuntu 18.x/20.x 和 SUSE Linux Enterprise 12.x/15.x 上配置和启动 X 服务器

1. 将 X 服务器配置为当 Linux 服务器启动时自动启动。

```
$ sudo systemctl get-default
```

如果命令返回 `graphical.target`，则表明 X 服务器已配置为自动启动。继续下一步。

如果命令返回 `multi-user.target`，则表明 X 服务器未配置为自动启动。运行以下命令：

```
$ sudo systemctl set-default graphical.target
```

2. 启动 X 服务器。

```
$ sudo systemctl isolate graphical.target
```

3. 验证 X 服务器是否正在运行。

```
$ ps aux | grep X | grep -v grep
```

下面显示 X 服务器在运行时的示例输出。

```
root      1891  0.0  0.7 277528 30448 tty7      Ssl+ 10:59   0:00 /usr/bin/Xorg :0  
-background none -verbose -auth /run/gdm/auth-for-gdm-wltseN/database -seat seat0  
vt7
```

安装 glxinfo 实用程序。

glxinfo 实用程序提供有关 Linux 服务器的 OpenGL 配置的信息。它可用于确定您的 Linux 服务器是否已配置为支持 OpenGL 硬件或软件渲染，并提供有关驱动程序和支持的扩展的信息。

glxinfo 实用程序作为 DCV GL 的程序包依赖项安装。因此，如果您安装了 DCV GL，则 glxinfo 实用程序已随之安装在您的 Linux 服务器上。

安装 glxinfo 实用程序

运行以下命令：

- RHEL 6.x/7.x/8.x、CentOs 6.x/7.x/8.x 和 Amazon Linux 2

```
$ sudo yum install glx-utils
```

- Ubuntu 18.x/20.x

```
$ sudo apt install mesa-utils
```

- SUSE Linux Enterprise 12.x/15.x

```
$ sudo zypper in Mesa-demo-x
```

验证 OpenGL 软件渲染

在非 GPU Linux 服务器上，仅在软件渲染模式下使用 Mesa 驱动程序支持 OpenGL。如果您使用的是非 GPU Linux 服务器，并且打算使用 OpenGL，请确保 Mesa 驱动程序已在 Linux 服务器上安装并正确配置。

Note

这仅适用于非 GPU 的 Linux 服务器。

验证 OpenGL 软件渲染是否可用

确保 X 服务器正在运行，并使用以下命令。

```
$ sudo DISPLAY=:0 XAUTHORITY=$(ps aux | grep "X.*\|-auth" | grep -v grep | sed -n 's/.*-auth\n\([^ ]+\)\.*/\1/p') glxinfo | grep -i "opengl.*version"
```

下面显示了 OpenGL 软件渲染可用时的示例输出：

```
OpenGL core profile version string: 3.3 (Core Profile) Mesa 17.0.5
OpenGL core profile shading language version string: 3.30
OpenGL version string: 3.0 Mesa 17.0.5
OpenGL shading language version string: 1.30
OpenGL ES profile version string: OpenGL ES 3.0 Mesa 17.0.5
OpenGL ES profile shading language version string: OpenGL ES GLSL ES 3.00
```

安装和配置 NVIDIA 驱动程序

对于拥有专用 NVIDIA GPU 的 Linux 服务器，必须确保相应的 NVIDIA 驱动程序已安装且正确配置。有关在 NVIDIA Linux 实例上安装 Amazon EC2 驱动程序的更多信息，请参阅 [中的](#)在 Linux 服务器上安装 NVIDIA 驱动程序。Amazon EC2 用户指南（适用于 Linux 实例）

Note

- 这仅适用于具有 NVIDIA GPUs 的 Linux 服务器。
- GRID 驱动程序支持每个 GPU 最多四个 4K 显示器，而游戏驱动程序支持每个 GPU 一个 4K 显示器。

在 Linux 服务器上安装了 NVIDIA 驱动程序后，必须更新 `xorg.conf`。

生成更新的 `xorg.conf`

1. 运行以下命令。

```
sudo nvidia-xconfig --preserve-busid --enable-all-gpus
```

如果您使用的是 G3 或 G4 Amazon EC2 实例，并且要使用多显示器控制台会话，请包含 `--connected-monitor=DFP-0,DFP-1,DFP-2,DFP-3` 参数，如下所示。

```
sudo nvidia-xconfig --preserve-busid --enable-all-gpus --connected-monitor=DFP-0,DFP-1,DFP-2,DFP-3
```

Note

确保您的服务器没有旧 `/etc/X11/XF86Config` 文件。如果服务器具有该文件，则 `nvidia-xconfig` 将更新该配置文件，而不是生成所需的 `/etc/X11/xorg.conf` 文件。运行以下命令可删除旧 `XF86Config` 文件：

```
sudo rm -rf /etc/X11/XF86Config*
```

2. 重新启动 X 服务器以使更改生效。

- RHEL 7.x、CentOs 7.x、Amazon Linux 2、Ubuntu 18.x 和 SUSE Linux Enterprise 12.x

```
$ sudo systemctl isolate multi-user.target
```

```
$ sudo systemctl isolate graphical.target
```

- RHEL 6.x 和 CentOs 6.x

```
$ sudo init 3
```

```
$ sudo init 5
```

验证 NVIDIA GPU 是否支持基于硬件的视频编码

您必须确保它支持 NVENC 编码并且其计算功能 ≥ 3.0 ，或对于 Ubuntu 20 为 ≥ 3.5 。

要验证 NVENC 支持，请参阅 [NVIDIA 视频编码和解码 GPU 支持矩阵](#)。要检查计算功能，请参阅 [NVIDIA 计算容量表](#)。

如果您的 NVIDIA GPU 不支持 NVENC 编码，或者它没有所需的计算功能，则使用基于软件的视频编码。

验证 OpenGL 硬件渲染是否可用

使用以下命令可确保 X 服务器正在运行。

```
$ sudo DISPLAY=:0 XAUTHORITY=$(ps aux | grep "X.*\s-auth" | grep -v grep | sed -n 's/.*-auth \([^ ]*\).*/\1/p') glxinfo | grep -i "opengl.*version"
```

下面显示了 OpenGL 硬件渲染可用时的示例输出。

```
OpenGL core profile version string: 4.4.0 NVIDIA 390.75
OpenGL core profile shading language version string: 4.40 NVIDIA via Cg compiler
OpenGL version string: 4.6.0 NVIDIA 390.75
OpenGL shading language version string: 4.60 NVIDIA
OpenGL ES profile version string: OpenGL ES 3.2 NVIDIA 390.75
```

```
OpenGL ES profile shading language version string: OpenGL ES GLSL ES 3.20
```

在 Linux 上安装 NICE DCV 服务器

NICE DCV 服务器的安装使用了一系列 RPM 或 .deb 程序包，具体取决于您的主机服务器的操作系统。这些程序包将安装所有必需的程序包及其依赖项，并执行必需的服务器配置。

Note

您必须以根用户身份登录才能安装 NICE DCV 服务器。

安装 NICE DCV 服务器

RHEL/CentOS 6.x

服务器仅适用于基于 64 位 x86 架构的 RHEL 和 NICE DCV 6.x 服务器。CentOS

在 RHEL 6.x 和 NICE DCV 服务器 6.x 上安装 CentOS

1. 启动并连接到要安装的服务器。NICE DCV 服务器。
2. NICE DCV 服务器程序包使用安全 GPG 签名进行数字签名。要允许程序包管理器验证程序包签名，您必须导入 NICE GPG 密钥。为此，打开一个终端窗口并导入 NICE GPG 密钥。

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

3. 从 [NICE 网站](#) 下载程序包。RPM 和 deb 程序包打包到一个 .tgz 存档中。确保为您的操作系统下载了正确的存档。

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2020.2/Servers/nice-dcv-2020.2-9662-e16-x86_64.tgz
```

4. 提取 .tgz 存档的内容。

```
$ tar -xvzf nice-dcv-2020.2-9662-e16-x86_64.tgz
```

5. 导航到解压的文件夹。

```
$ cd nice-dcv-2020.2-9662-e16-x86_64
```

6. 安装 NICE DCV 服务器。

```
$ sudo yum install nice-dcv-server-2020.2.9662-1.e16.x86_64.rpm
```

7. (可选) 如果您计划使用虚拟会话，请安装 nice-xdcv 程序包。

```
$ sudo yum install nice-xdcv-2020.2.359-1.e16.x86_64.rpm
```

8. (可选) 如果您计划使用 GPU 共享，请安装 nice-dcv-gl 程序包。

```
$ sudo yum install nice-dcv-gl-2020.2.881-1.e16.x86_64.rpm
```

Note

您可以有选择性地安装 nice-dcv-gltest 程序包。此程序包中包含一个简单的 OpenGL 应用程序，可用于确定您的虚拟会话是否已正确配置为使用基于硬件的 OpenGL。

9. (可选) 如果您计划将 NICE DCV 与 NICE EnginFrame 结合使用，请安装 nice-dcv-simple-external-authenticator 软件包。

```
$ sudo yum install nice-dcv-simple-external-  
authenticator-2020.2.125-1.el6.x86_64.rpm
```

10. (可选) 如果您计划使用 USB 远程控制支持专用 USB 设备, 请安装 DCV USB 驱动程序。

要安装 DCV USB 驱动程序, 必须在服务器上安装动态内核模块支持 (DKMS)。使用以下命令安装 DKMS。

可从 Extra Packages for Enterprise Linux (EPEL) 存储库安装 DKMS。运行以下命令以启用 EPEL 存储库:

```
$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-  
latest-6.noarch.rpm
```

启用 EPEL 存储库后, 运行以下命令以安装 DKMS:

```
$ sudo yum install dkms
```

安装 DKMS 后, 运行以下命令安装 DCV USB 驱动程序:

```
$ sudo dcvusbdriverinstaller
```

Amazon Linux 2 and RHEL/CentOS 7.x

服务器适用于基于 64 位 x86 和 64 位 ARM 架构的 NICE DCV、RHEL 和 Amazon Linux 2 7.x 服务器。CentOS

Important

和 nice-dcv-gl 包不适用于基于 64 位 ARM 架构的服务器。nice-dcv-gltest

在 NICE DCV 服务器、RHEL 7.x 和 Amazon Linux 2 7.x 上安装CentOS

1. 启动并连接到要安装 的服务器。NICE DCV 服务器。
2. NICE DCV 服务器程序包使用安全 GPG 签名进行数字签名。要允许程序包管理器验证程序包签名, 您必须导入 NICE GPG 密钥。为此, 打开一个终端窗口并导入 NICE GPG 密钥。

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

3. 从 [NICE 网站](#) 下载程序包。RPM 和 deb 程序包打包到一个 .tgz 存档中。确保为您的操作系统下载了正确的存档。

- 64 位 x86

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2020.2/Servers/nice-dcv-2020.2-9662-  
e17-x86_64.tgz
```

- 64 位 ARM

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2020.2/Servers/nice-dcv-2020.2-9662-  
e17-aarch64.tgz
```

4. 提取 .tgz 存档的内容并导航到提取的目录。

- 64 位 x86

```
$ tar -xvzf nice-dcv-2020.2-9662-el7-x86_64.tgz && cd nice-dcv-2020.2-9662-el7-x86_64
```

- 64 位 ARM

```
$ tar -xvzf nice-dcv-2020.2-9662-el7-aarch64.tgz && cd nice-dcv-2020.2-9662-el7-aarch64
```

5. 安装NICE DCV 服务器。

- 64 位 x86

```
$ sudo yum install nice-dcv-server-2020.2.9662-1.el7.x86_64.rpm
```

- 64 位 ARM

```
$ sudo yum install nice-dcv-server-2020.2.9662-1.el7.aarch64.rpm
```

6. (可选) 如果您计划使用虚拟会话 , 请安装 nice-xdcv 程序包。

- 64 位 x86

```
$ sudo yum install nice-xdcv-2020.2.359-1.el7.x86_64.rpm
```

- 64 位 ARM

```
$ sudo yum install nice-xdcv-2020.2.359-1.el7.aarch64.rpm
```

7. (可选) 如果您计划使用 GPU 共享 , 请安装 nice-dcv-gl 程序包。

- 64 位 x86

```
$ sudo yum install nice-dcv-gl-2020.2.881-1.el7.x86_64.rpm
```

Note

您可以有选择性地安装 nice-dcv-gltest 程序包。此程序包中包括一个简单的 OpenGL 应用程序 , 可用于确定您的虚拟会话是否已正确配置为使用基于硬件的 OpenGL。

8. (可选) 如果您计划将 NICE DCV 与 NICE EnginFrame 结合使用 , 请安装 nice-dcv-simple-external-authenticator 软件包。

- 64 位 x86

```
$ sudo yum install nice-dcv-simple-external-authenticator-2020.2.125-1.el7.x86_64.rpm
```

- 64 位 ARM

```
$ sudo yum install nice-dcv-simple-external-authenticator-2020.2.125-1.el7.aarch64.rpm
```

9. (可选) 如果您计划使用 USB 远程控制支持专用 USB 设备 , 请安装 DCV USB 驱动程序。

要安装 DCV USB 驱动程序 , 必须在服务器上安装动态内核模块支持 (DKMS)。使用以下命令安装 DKMS。

可从 Extra Packages for Enterprise Linux (EPEL) 存储库安装 DKMS。运行以下命令以启用 EPEL 存储库：

```
$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

启用 EPEL 存储库后，运行以下命令以安装 DKMS：

```
$ sudo yum install dkms
```

安装 DKMS 后，运行以下命令安装 DCV USB 驱动程序：

```
$ sudo dcvusbdriverinstaller
```

RHEL/CentOS 8.x

服务器适用于基于 64 位 x86 和 64 位 ARM 架构的 RHEL 和 NICE DCV 8.x 服务器。CentOS

Important

和 nice-dcv-gl 程序包不适用于基于 64 位 ARM 架构的服务器。nice-dcv-gltest

在 RHEL 8.x 或 NICE DCV 服务器 8.x 上安装CentOS

1. 启动并连接到要安装 的服务器。NICE DCV 服务器。
2. NICE DCV 服务器程序包使用安全 GPG 签名进行数字签名。要允许程序包管理器验证程序包签名，您必须导入 NICE GPG 密钥。为此，打开一个终端窗口并导入 NICE GPG 密钥。

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

3. 从 [NICE 网站](#) 下载程序包。RPM 和 deb 程序包打包到一个 .tgz 存档中。确保为您的操作系统下载了正确的存档。

- 64 位 x86

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2020.2/Servers/nice-dcv-2020.2-9662-e18-x86_64.tgz
```

- 64 位 ARM

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2020.2/Servers/nice-dcv-2020.2-9662-e18-aarch64.tgz
```

4. 提取 .tgz 存档的内容并导航到提取的目录。

- 64 位 x86

```
$ tar -xvzf nice-dcv-2020.2-9662-e18-x86_64.tgz && cd nice-dcv-2020.2-9662-e18-x86_64
```

- 64 位 ARM

```
$ tar -xvzf nice-dcv-2020.2-9662-e18-aarch64.tgz && cd nice-dcv-2020.2-9662-e18-aarch64
```

5. 安装NICE DCV 服务器.

- 64 位 x86

```
$ sudo yum install nice-dcv-server-2020.2.9662-1.el8.x86_64.rpm
```

- 64 位 ARM

```
$ sudo yum install nice-dcv-server-2020.2.9662-1.el8.aarch64.rpm
```

6. (可选) 如果您计划使用虚拟会话, 请安装 nice-xdcv 程序包。

- 64 位 x86

```
$ sudo yum install nice-xdcv-2020.2.359-1.el8.x86_64.rpm
```

- 64 位 ARM

```
$ sudo yum install nice-xdcv-2020.2.359-1.el8.aarch64.rpm
```

7. (可选) 如果您计划使用 GPU 共享, 请安装 nice-dcv-gl 程序包。

- 64 位 x86

```
$ sudo yum install nice-dcv-gl-2020.2.881-1.el8.x86_64.rpm
```

Note

您可以有选择性地安装 nice-dcv-gltest 程序包。此程序包中包含一个简单的 OpenGL 应用程序, 可用于确定您的虚拟会话是否已正确配置为使用基于硬件的 OpenGL。

8. (可选) 如果您计划将 NICE DCV 与 NICE EnginFrame 结合使用, 请安装 nice-dcv-simple-external-authenticator 软件包。

- 64 位 x86

```
$ sudo yum install nice-dcv-simple-external-  
authenticator-2020.2.125-1.el8.x86_64.rpm
```

- 64 位 ARM

```
$ sudo yum install nice-dcv-simple-external-  
authenticator-2020.2.125-1.el8.aarch64.rpm
```

9. (可选) 如果您计划使用 USB 远程控制支持专用 USB 设备, 请安装 DCV USB 驱动程序。

要安装 DCV USB 驱动程序, 必须在服务器上安装动态内核模块支持 (DKMS)。使用以下命令安装 DKMS。

可从 Extra Packages for Enterprise Linux (EPEL) 存储库安装 DKMS。运行以下命令以启用 EPEL 存储库:

```
$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-  
latest-8.noarch.rpm
```

启用 EPEL 存储库后, 运行以下命令以安装 DKMS:

```
$ sudo yum install dkms
```

安装 DKMS 后，运行以下命令安装 DCV USB 驱动程序：

```
$ sudo dcvusbdriverinstaller
```

SLES 12.x/15.x

服务器仅适用于基于 64 位 x86 架构的 SUSE Linux Enterprise Server (SLES) 12.x/15.x 服务器。NICE DCV

在 SLES 12.x/15.x 上安装 NICE DCV 服务器

1. 启动并连接到要安装的服务器。NICE DCV 服务器。
2. NICE DCV 服务器程序包使用安全 GPG 签名进行数字签名。要允许程序包管理器验证程序包签名，您必须导入 NICE GPG 密钥。为此，打开一个终端窗口并导入 NICE GPG 密钥。

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

3. 从 [NICE 网站](#) 下载程序包。RPM 和 deb 程序包打包到一个 .tgz 存档中。确保为您的操作系统下载了正确的存档。

- SLES 12.x

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2020.2/Servers/nice-dcv-2020.2-9662-sles12-x86_64.tgz
```

- SLES 15.x

```
$ curl -O https://d1uj6qtbmh3dt5.cloudfront.net/2020.2/Servers/nice-dcv-2020.2-9662-sles15-x86_64.tgz
```

4. 提取 .tgz 存档的内容并导航到提取的目录。

- SLES 12.x

```
$ tar -xvzf nice-dcv-2020.2-9662-sles12-x86_64.tgz && cd nice-dcv-2020.2-9662-sles12-x86_64
```

- SLES 15.x

```
$ tar -xvzf nice-dcv-2020.2-9662-sles15-x86_64.tgz && cd nice-dcv-2020.2-9662-sles15-x86_64
```

5. 安装 NICE DCV 服务器。

- SLES 12.x

```
$ sudo zypper install nice-dcv-server-2020.2.9662-1.sles12.x86_64.rpm
```

- SLES 15.x

```
$ sudo zypper install nice-dcv-server-2020.2.9662-1.sles15.x86_64.rpm
```

6. (可选) 如果您计划使用虚拟会话，请安装 nice-xdcv 程序包。

- SLES 12.x

```
$ sudo zypper install nice-xdcv-2020.2.359-1.sles12.x86_64.rpm
```

- SLES 15.x

```
$ sudo zypper install nice-xdcv-2020.2.359-1.sles15.x86_64.rpm
```

7. (可选) 如果您计划使用 GPU 共享, 请安装 nice-dcv-g1 程序包。

- SLES 12.x

```
$ sudo zypper install nice-dcv-g1-2020.2.881-1.sles12.x86_64.rpm
```

- SLES 15.x

```
$ sudo zypper install nice-dcv-g1-2020.2.881-1.sles15.x86_64.rpm
```

Note

您可以有选择性地安装 nice-dcv-g1test 程序包。此程序包中包含一个简单的 OpenGL 应用程序, 可用于确定您的虚拟会话是否已正确配置为使用基于硬件的 OpenGL。

8. (可选) 如果您计划将 NICE DCV 与 NICE EnginFrame 结合使用, 请安装 nice-dcv-simple-external-authenticator 软件包。

- SLES 12.x

```
$ sudo zypper install nice-dcv-simple-external-  
authenticator-2020.2.125-1.sles12.x86_64.rpm
```

- SLES 15.x

```
$ sudo zypper install nice-dcv-simple-external-  
authenticator-2020.2.125-1.sles15.x86_64.rpm
```

9. (可选) 如果您计划使用 USB 远程控制支持专用 USB 设备, 请安装 DCV USB 驱动程序。

要安装 DCV USB 驱动程序, 必须在服务器上安装动态内核模块支持 (DKMS)。使用以下命令安装 DKMS。

运行以下命令以安装 DKMS :

- SLES 12.x

```
$ sudo zypper install http://download.opensuse.org/repositories/home:/Ximi1970:/  
Dkms:/Staging/SLE_12_SP4/noarch/dkms-2.5-11.1.noarch.rpm
```

- SLES 15

启用 PackageHub 存储库。

```
$ sudo SUSEConnect -p PackageHub/15/x86_64
```

Note

如果您使用的是 SLES 15 SP1 或 SP2，请替换 **15**（在上述命令中使用 15.1 或 15.2）。

安装 DKMS。

```
$ sudo zypper refresh
```

```
$ sudo zypper install dkms
```

安装内核源。

```
$ sudo zypper install -y kernel-source
```

重启实例。

```
$ sudo reboot
```

安装 DKMS 后，运行以下命令安装 DCV USB 驱动程序：

```
$ sudo dcvusbdriverinstaller
```

Ubuntu 18.04/20.04

服务器适用于基于 64 位 x86 和 64 位 ARM 架构的 Ubuntu 服务器。NICE DCV

Important

和 nice-dcv-gl 包不适用于基于 64 位 ARM 架构的服务器。nice-dcv-gltest

在 Ubuntu 18.04/20.04 上安装 NICE DCV 服务器

1. 启动并连接到要安装的服务器。NICE DCV 服务器。
2. NICE DCV 服务器程序包使用安全 GPG 签名进行数字签名。要允许程序包管理器验证程序包签名，您必须导入 NICE GPG 密钥。为此，打开一个终端窗口并导入 NICE GPG 密钥。

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

```
$ gpg --import NICE-GPG-KEY
```

3. 从 [NICE 网站](#) 下载程序包。RPM 和 deb 程序包打包到一个 .tgz 存档中。确保为您的操作系统下载了正确的存档。

- Ubuntu 18.04 (64 位 x86)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2020.2/Servers/nice-dcv-2020.2-9662-ubuntu1804-x86_64.tgz
```

- Ubuntu 18.04 (64 位 ARM)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2020.2/Servers/nice-dcv-2020.2-9662-ubuntu1804-aarch64.tgz
```

- Ubuntu 20.04 (64 位 x86)

```
$ wget https://d1uj6qtbmh3dt5.cloudfront.net/2020.2/Servers/nice-dcv-2020.2-9662-ubuntu2004-x86_64.tgz
```

4. 提取 .tgz 存档的内容并导航到提取的目录。

- Ubuntu 18.04 (64 位 x86)

```
$ tar -xvzf nice-dcv-2020.2-9662-ubuntu1804-x86_64.tgz && cd nice-dcv-2020.2-9662-ubuntu1804-x86_64
```

- Ubuntu 18.04 (64 位 ARM)

```
$ tar -xvzf nice-dcv-2020.2-9662-ubuntu1804-arm64.tgz && cd nice-dcv-2020.2-9662-ubuntu1804-arm64
```

- Ubuntu 20.04 (64 位 x86)

```
$ tar -xvzf nice-dcv-2020.2-9662-ubuntu2004-x86_64.tgz && cd nice-dcv-2020.2-9662-ubuntu2004-x86_64
```

5. 安装NICE DCV 服务器。

- Ubuntu 18.04 (64 位 x86)

```
$ sudo apt install ./nice-dcv-server_2020.2.9662-1_amd64.ubuntu1804.deb
```

- Ubuntu 18.04 (64 位 ARM)

```
$ sudo apt install ./nice-dcv-server_2020.2.9662-1_arm64.ubuntu1804.deb
```

- Ubuntu 20.04 (64 位 x86)

```
$ sudo apt install ./nice-dcv-server_2020.2.9662-1_amd64.ubuntu2004.deb
```

6. (可选) 如果您计划使用虚拟会话 , 请安装 nice-xdcv 程序包。

- Ubuntu 18.04 (64 位 x86)

```
$ sudo apt install ./nice-xdcv_2020.2.359-1_amd64.ubuntu1804.deb
```

- Ubuntu 18.04 (64 位 ARM)

```
$ sudo apt install ./nice-xdcv_2020.2.359-1_arm64.ubuntu1804.deb
```

- Ubuntu 20.04 (64 位 x86)

```
$ sudo apt install ./nice-xdcv_2020.2.359-1_amd64.ubuntu2004.deb
```

7. (可选) 如果您计划使用 GPU 共享 , 请安装 nice-dcv-gl 程序包。

- Ubuntu 18.04 (64 位 x86)

```
$ sudo apt install ./nice-dcv-gl_2020.2.881-1_amd64.ubuntu1804.deb
```

- Ubuntu 20.04 (64 位 x86)

```
$ sudo apt install ./nice-dcv-gl_2020.2.881-1_amd64.ubuntu2004.deb
```

Note

您可以有选择性地安装 `nice-dcv-gltest` 程序包。此程序包中包含一个简单的 OpenGL 应用程序，可用于确定您的虚拟会话是否已正确配置为使用基于硬件的 OpenGL。

8. (可选) 如果您计划将 NICE DCV 与 NICE EnginFrame 结合使用，请安装 `nice-dcv-simple-external-authenticator` 软件包。

- Ubuntu 18.04 (64 位 x86)

```
$ sudo apt install ./nice-dcv-simple-external-authenticator_2020.2.125-1_amd64.ubuntu1804.deb
```

- Ubuntu 18.04 (64 位 ARM)

```
$ sudo apt install ./nice-dcv-simple-external-authenticator_2020.2.125-1_arm64.ubuntu1804.deb
```

- Ubuntu 20.04 (64 位 x86)

```
$ sudo apt install ./nice-dcv-simple-external-authenticator_2020.2.125-1_amd64.ubuntu2004.deb
```

9. (可选) 如果您计划使用 USB 远程控制支持专用 USB 设备，请安装 DCV USB 驱动程序。

要安装 DCV USB 驱动程序，必须在服务器上安装动态内核模块支持 (DKMS)。使用以下命令安装 DKMS。

可以在官方 Ubuntu 存储库中找到 DKMS。运行以下命令以安装 DKMS：

```
$ sudo apt install dkms
```

安装 DKMS 后，运行以下命令安装 DCV USB 驱动程序：

```
$ sudo dcvusbdriverinstaller
```

安装后检查

本主题提供了在安装 NICE DCV 后应执行的一些安装后检查任务，目的是确保您的 NICE DCV 服务器配置正确。

目录

- [确保 NICE DCV 服务器可访问 \(p. 27\)](#)
- [确保 X 服务器可访问 \(p. 27\)](#)
- [验证 DCV GL 是否正确安装 \(p. 28\)](#)

确保 NICE DCV 服务器可访问

默认情况下，NICE DCV 服务器配置为通过端口 8443 进行通信。确保可通过此端口访问该服务器。如果您的防火墙阻止通过端口 8443 访问，则必须更改 NICE DCV 服务器的通信端口。有关更多信息，请参阅 [更改 NICE DCV 服务器 TCP 端口 \(p. 43\)](#)。

此外，如果您正在 EC2 实例上设置 NICE DCV，请创建一个安全组以允许访问 NICE DCV 服务器用来通信的端口。有关更多信息，请参阅 [如何在 EC2 上配置安全组](#)。

确保 X 服务器可访问

必须确保 NICE DCV 控制台和虚拟会话可以访问 X 服务器。

控制台会话

安装 NICE DCV 服务器时，将创建一个 dcv 用户。必须确保该用户可以访问 X 服务器。

验证 **dcv** 用户能否访问 X 服务器

运行以下命令：

```
$ sudo DISPLAY=:0 XAUTHORITY=$(ps aux | grep "X.*\|-auth" | grep -v grep | sed -n 's/.*-auth\n\([^ ]+\)\.*/\1/p') xhost | grep "SI:localuser:dcv$"
```

如果命令返回 SI:localuser:dcv，则表明该 dcv 用户可以访问 X 服务器。

如果命令未返回 SI:localuser:dcv，则表明该 dcv 用户无权访问 X 服务器。运行以下命令重新启动 X 服务器：

- RHEL 7.x/8.x、CentOs 7.x/8.x、Amazon Linux 2、Ubuntu 18.x 和 SUSE Linux Enterprise 12.x

```
$ sudo systemctl isolate multi-user.target
```

```
$ sudo systemctl isolate graphical.target
```

- RHEL 6.x 和 CentOs 6.x

```
$ sudo init 3
```

```
$ sudo init 5
```

虚拟会话

如果您安装了 DCV GL 程序包，则必须确保本地用户可以访问 X 服务器。这可确保 OpenGL 硬件加速正常用于虚拟会话。

验证本地用户能否访问 X 服务器

运行以下命令：

```
$ sudo DISPLAY=:0 XAUTHORITY=$(ps aux | grep "X.*\|-auth" | grep -v grep | sed -n 's/.*-auth\n\([^ ]+\)\.*/\1/p') xhost | grep "LOCAL:$"
```

如果命令返回 LOCAL:，则表明本地用户可以访问 X 服务器。

如果命令未返回 LOCAL:，则表明本地用户无权访问 X 服务器。运行以下命令来重新启动 X 服务器，并禁用和重新启用 DCV GL：

- RHEL 7.x/8.x、CentOs 7.x/8x、Amazon Linux 2、Ubuntu 18.x 和 SUSE Linux Enterprise 12.x

```
$ sudo systemctl isolate multi-user.target
```

```
$ sudo dcvgladmin disable
```

```
$ sudo dcvgladmin enable
```

```
$ sudo systemctl isolate graphical.target
```

- RHEL 6.x 和 CentOs 6.x

```
$ sudo init 3
```

```
$ sudo dcvgladmin disable
```

```
$ sudo dcvgladmin enable
```

```
$ sudo init 5
```

验证 DCV GL 是否正确安装

当您安装 DCV GL 程序包时，会自动安装 dcvgldiag 实用程序。可以使用该实用程序检查 Linux 服务器配置是否符合 DCV GL 要求。

运行 dcvgldiag 实用程序

使用以下命令：

```
$ sudo dcvgldiag
```

该实用程序会返回警告和错误列表并给出可能的解决方案。

许可 NICE DCV 服务器

NICE DCV 许可要求因安装和使用 NICE DCV 服务器的位置而异。

Important

以下许可要求仅适用于 NICE DCV 版本 2017.0 及更高版本。

NICE DCV 许可要求

主题

- [NICE DCV](#)，发布时间：Amazon EC2 (p. 29)
- [NICE DCV在本地服务器和其他基于云的服务器上的](#) (p. 29)
- [远程访问 Windows Server 的 Microsoft 许可要求](#) (p. 30)

NICE DCV，发布时间：Amazon EC2

您不需要许可证服务器，即可在 EC2 实例上安装和使用 NICE DCV 服务器。NICE DCV 服务器自动检测它是否在 Amazon EC2 实例上运行，并定期连接到 S3 存储桶，以确定是否提供了有效的许可证。

确保您的实例：

- 可以访问 Amazon S3 终端节点。如果它能够访问 Internet，则使用 Amazon S3 公有终端节点进行连接。如果您的实例无法访问 Internet，请使用出站安全组规则或访问控制列表 (ACL) 策略为您的 VPC 配置网关终端节点，该策略允许您通过 HTTPS 访问 Amazon S3。有关更多信息，请参阅 <https://docs.amazonaws.cn/AmazonVPC/latest/UserGuide/vpce-gateway.html> 中的网关 VPC 终端节点 Amazon VPC 用户指南。如果您在连接到 S3 存储桶时遇到任何问题，请参阅 [AWS 知识中心](#) 中的为什么我无法使用网关 VPC 终端节点连接到 S3 存储桶？。
- 有权访问所需的 Amazon S3 对象。将以下 Amazon S3 访问策略添加到实例的 IAM 角色并替换 `region` 占位符替换为您的 AWS 区域（例如，`us-east-1`）。有关更多信息，请参阅 [创建 IAM 角色](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::dcv-license.region/*"
    }
  ]
}
```

- 如果您使用的是 Windows 实例，请确保该实例可以访问实例元数据服务。需要对此服务的访问权限以确保 NICE DCV 服务器可以获得正确许可。有关实例元数据服务的更多信息，请参阅 <https://docs.amazonaws.cn/AWSEC2/latest/WindowsGuide/ec2-instance-metadata.html> 中的实例元数据和用户数据 Amazon EC2 用户指南（适用于 Windows 实例）。

如果您使用的是自定义 Windows AMI，则必须安装 EC2Config 服务（Windows Server 2012 R2 及更早版本）或 EC2Launch（Windows Server 2016 及更高版本）。这将确保您的实例能够访问实例元数据服务。有关更多信息，请参阅 [中的 EC2Config 使用](#) 服务配置 Windows 实例或 [EC2Launch 使用](#) 配置 Windows 实例。Amazon EC2 用户指南（适用于 Windows 实例）

如果您正在 NICE DCV 服务器实例上安装和使用 Amazon EC2，可以跳过本章的其余部分。本章的其余部分仅适用于在本地服务器或其他基于云的服务器上安装和使用 NICE DCV 服务器的情况。

NICE DCV 在本地服务器和其他基于云的服务器上的

需要许可证才能在本地服务器或其他基于云的服务器上安装和使用 NICE DCV 服务器。提供了以下许可选项：

- 自动评估许可证 — 在您安装 NICE DCV 服务器时自动安装。这些许可证的有效期为从安装之日起的 30 天。在许可证过期后，您不再能够在服务器上创建和托管 NICE DCV 会话。这些许可证非常适合短期测试和评估。要测试一段较长的时间，可请求延期评估许可证。

Note

如果没有配置任何其他许可证，NICE DCV 服务器默认为使用自动评估许可证。

- 延期评估许可证 — 延期评估许可证是延长自动评估许可证提供的初始 30 天评估期的评估许可证。有效期由 NICE 根据每个案例确定。在到达到期日期后，延期评估许可证将变得无效，您此时不再能够在服务器上创建和托管 NICE DCV 会话。必须向 NICE 网站的 [How to Buy](#) 页面上列出的 NICE 分销商或经销商请求延期评估许可证。这些许可证以许可证文件的形式提供，必须安装在上。NICE DCV 服务器。
- 生产许可证 — 生产许可证是您从 NICE 购买的完整许可证。生产许可证是由许可证服务器管理的浮动许可证。浮动许可证可让您在网络中运行多个 NICE DCV 服务器，并限制您可以跨所有服务器创建的并发

NICE DCV 会话的数量。每个并发 NICE DCV 会话都需要一个许可证。生产许可证作为许可证文件分发，您必须将它安装在 Reprise License Manager (RLM) 服务器上。提供了两种类型的生产许可证：

- 永久许可证 — 永久许可证没有到期日期，可以不限时间地使用。
- 订阅 — 订阅的有效期有限，通常为一年。许可证文件中指明了许可证的到期日期。在许可证过期后，您不再能够在 NICE DCV 服务器上创建和托管 NICE DCV 会话。

有关如何购买 NICE DCV 永久许可证或订阅的信息，请参阅 [网站上的](#) 如何购买NICE并查找您所在区域的 NICE 分销商或经销商。

Note

NICE DCV 客户端 不需要许可证。

Note

- NICE DCV 服务器版本 2020 与 NICE DCV 服务器版本 2019 及更早版本中的生产许可证和延期评估文件不兼容。如果升级到 NICE DCV 服务器版本 2020，则必须请求兼容的许可证文件。有关更多信息，请联系您的 NICE DCV 分销商或经销商。
- NICE DCV 服务器版本 2020 许可证文件与 NICE DCV 服务器版本 2017 和 2019 向后兼容。
- NICE DCV 客户端不需要许可证。

远程访问 Windows Server 的 Microsoft 许可要求

除了 Windows Server 客户端访问许可证（CAL），Microsoft 要求对于远程访问服务器的图形用户界面（GUI）的每个用户，您的 Windows Server 版本都必须有 Windows Server 远程桌面服务（RDS）CAL，无论您使用的远程显示协议如何。如果您使用 NICE DCV 访问远程 Windows Server 主机的 GUI，则还需要此许可证。

如果您在 NICE DCV 服务器实例上运行 Amazon EC2，并且您使用 [Windows Server AMI](#)，则 Amazon 会处理 Windows Server CAL 的许可成本，并提供两个仅用于管理目的（即测试、维护和管理）的 Windows Server RDS CALs。

有关更多信息，请参阅 [Microsoft 产品条款网站](#)。如果您对 Microsoft 软件的许可或权利有任何疑问，请咨询您的法律团队、Microsoft 或 Microsoft 经销商。

安装延期评估许可证

当您从 NICE 请求延期评估许可证时，您会收到一个 `license.lic` 文件，该文件定义了许可证。

安装延期评估许可证

将以下文件夹中的 `license.lic` 文件放在您的服务器上：

- Windows 服务器

```
C:\Program Files\NICE\DCV\Server\license\license.lic
```

- Linux 服务器

```
/usr/share/dcv/license/license.lic
```

或者，要将 `license.lic` 放置在服务器上不同的文件夹中，必须更新 `license-file` 配置参数，以便其指定许可证文件的完整路径。

主题

- [更改 Windows Server 上的许可证路径 \(p. 31\)](#)
- [更改 Linux Server 上的许可证路径 \(p. 31\)](#)

更改 Windows Server 上的许可证路径

更新 Windows 服务器上的 `license-file` 配置参数

1. 打开 Windows 注册表编辑器。
2. 导航到 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/license/` 注册表项，并选择 `license-file` 参数。

如果该注册表项中没有 `license-file` 参数，请创建一个：

- a. 在左侧面板中打开 `license` 注册表项的上下文（右键单击）菜单，然后依次选择新建和字符串值。
 - b. 对于名称，输入 `license-file` 并按 Enter。
3. 打开 `license-file` 参数。对于 Value data（值数据），输入 `license.lic` 文件的完整路径。
 4. 选择 OK，然后关闭 Windows 注册表编辑器。

更改 Linux Server 上的许可证路径

更新 Linux 服务器上的 `license-file` 配置参数

1. 导航到 `/etc/dcv/`，并使用您常用的文本编辑器打开 `dcv.conf`
2. 找到 `license-file` 部分中的 `[license]` 参数，并使用 `license.lic` 文件的新完整路径替换现有路径。

如果 `license-file` 部分中没有 `[license]` 参数，则使用以下格式手动添加此参数：

```
license-file = "/custom-path/license.lic"
```

3. 保存并关闭文件。

安装生产许可证

本主题中的以下部分说明了如何购买和使用生产许可证（永久许可证或订阅）。

主题

- [步骤 1：安装 RLM 服务器 \(p. 31\)](#)
- [步骤 2：获取 RLM 服务器的主机 ID \(p. 32\)](#)
- [步骤 3：购买永久许可证或订阅 \(p. 33\)](#)
- [步骤 4：修改许可证文件 \(p. 33\)](#)
- [步骤 5：配置 RLM 服务器 \(p. 34\)](#)
- [步骤 6：配置 NICE DCV 服务器 \(p. 37\)](#)

步骤 1：安装 RLM 服务器

当您购买永久许可证或订阅时，您将获得一个定义许可证条款的许可证文件。您必须在 Reprise License Manager (RLM) 服务器上安装该许可证文件。

有关 RLM 的更多信息，请参阅 [Reprise Software](#) 网站。

主题

- 在 [Windows 上安装 RLM 服务器](#) (p. 32)
- 在 [Linux 上安装 RLM 服务器](#) (p. 32)

在 Windows 上安装 RLM 服务器

在 Windows 上安装 RLM 服务器

1. 从 [Reprise Software 网站](#) 下载 RLM 许可证管理包。
2. 将 RLM 许可证管理包安装到 C:\RLM。

在 Linux 上安装 RLM 服务器

在 Linux 上安装 RLM 服务器

1. 从 [Reprise Software 网站](#) 下载 RLM 许可证管理包。
2. 创建用户组和 rlm 用户。这可以是任何有效的用户或服务账户。我们强烈建议您不要将根账户用于此值。

```
$ groupadd -r rlm
```

```
$ useradd -r -g rlm -d "/opt/nice/rlm" -s /sbin/nologin -c "RLM License Server" rlm
```

3. 创建 RLM 服务器所需的 /opt/nice/rlm 和 /opt/nice/rlm/license 目录。

```
$ mkdir -p /opt/nice/rlm/license
```

4. 将 RLM 许可证管理包的内容提取到 /opt/nice/rlm/，并确保文件由 rlm 用户拥有。

```
$ tar xvf x64_l1.admin.tar.gz -C /opt/nice/rlm/ --strip-components 1
```

```
$ chown -R rlm:rlm /opt/nice/rlm
```

步骤 2：获取 RLM 服务器的主机 ID

安装 RLM 服务器后，您必须获取 RLM 服务器的主机 ID。在购买永久许可证或订阅时，您需要提供此主机 ID。

在 Windows 上获取 RLM 服务器主机 ID

要获取服务器的主机 ID，请打开命令提示符，

导航到 C:\RLM\，然后运行以下命令。

```
C:\> rlmutil.exe rlmhostid ether
```

该命令返回 RLM 服务器的主机 ID，如下所示。

```
Hostid of this machine: 06814example
```

记录主机 ID。您在下一个步骤中需要使用它。

在 Linux 上获取 RLM 服务器主机 ID

获取服务器的主机 ID

导航到 `/opt/nice/rlm/` 并运行以下命令。

```
$ ./rlmutil rlmhostid ether
```

该命令返回 RLM 服务器的主机 ID，如下所示。

```
Hostid of this machine: 06814example
```

记录主机 ID。您在下一个步骤中需要使用它。

步骤 3：购买永久许可证或订阅

有关如何购买 NICE DCV 永久许可证或订阅的信息，请参阅 [网站上的如何购买NICE](#) 并查找您所在区域的 NICE 分销商或经销商。

您必须提供 RLM 服务器的主机 ID。主机 ID 将嵌入到 NICE 提供的许可证文件中。

步骤 4：修改许可证文件

当您购买 NICE DCV 永久许可证或订阅时，您将收到一个定义许可证的 `license.lic` 文件。`license.lic` 文件包含以下信息：

- RLM 服务器的主机名。
- 您在购买许可证时提供的 RLM 服务器的主机 ID。
- RLM 服务器的 TCP 端口号。默认值为 5053。
- ISV 端口号。这是 RLM 服务器用来侦听 NICE DCV 许可证请求的可选端口。
- 许可证授权的 NICE DCV 产品，以及每种产品的以下详细信息：
 - 许可证所涵盖的主要版本（例如，2017 表示 2017 版 NICE DCV 产品）。
 - 到期日期。Permanent 指示许可证不过期。
 - 最大并发会话数（例如，10 表示服务器上的 10 个并发会话）。
 - 许可证校验和。
 - 许可证签名。

下面的代码块显示了 `license.lic` 文件的格式：

```
HOST RLM_server_hostname RLM_server_host_id RLM_server_port  
ISV nice port=port_number  
LICENSE product_1 major_version expiration_date concurrent_sessions share=hi _ck=checksum  
sig="signature"  
LICENSE product_2 major_version expiration_date concurrent_sessions share=hi _ck=checksum  
sig="signature"
```

以下代码块显示了已省略 ISV 端口的 `license.lic` 文件的示例。许可证文件包括两个 NICE 产品（DCV 和 `dcv-gl`）的许可证。

```
HOST My-RLM-server abcdef123456 5053  
ISV nice  
LICENSE nice dcv 2017 permanent 10 share=hi _ck=456789098a  
sig="abcdefghijklmnopqrstuvwxy1234567890abcdefghijklmnopqrstuvwxy1234567890ab"  
LICENSE nice dcv-gl 2017 permanent 10 share=hi _ck=123454323x  
sig="1234567890abcdefghijklmnopqrstuvwxy1234567890abcdefghijklmnopqrstuvwxy12"
```

编辑 `license.lic` 文件

1. 使用首选文本编辑器打开该文件。
2. 在该文件中的第一行（该行以 `HOST` 开头）中添加您的 RLM 服务器的主机名和 TCP 端口号。

Warning

这些区域有：`RLM_server_host_id` 是您在购买许可证时提供的主机 ID。您无法编辑 `RLM_server_host_id`。

3. （可选）通过添加 ISV，在文件中的第二行（该行以 `port=port_number` 开头）中添加 ISV 端口号。

如果您不想指定 ISV 端口，请省略 `port=port_number`。如果未指定端口，则使用随机端口。使用随机端口可能会导致与防火墙配置发生冲突。

4. 保存并关闭文件。

Warning

编辑许可证文件的任何其他部分会破坏文件的签名并使许可证无效。

步骤 5：配置 RLM 服务器

在修改许可证文件后，您必须将该文件放置到 RLM 服务器上，然后启动 RLM 服务。

主题

- 在 [Windows 上配置 RLM 服务器](#) (p. 34)
- 在 [Linux 上配置 RLM 服务器](#) (p. 35)

在 Windows 上配置 RLM 服务器

在 Windows 上配置 RLM 服务器

1. 连接到您的 RLM 服务器。
2. 将编辑后的 `license.lic` 文件复制到 `C:\RLM\license\`。
3. 从 `C:\Program Files\NICE\DCV\Server\license\nice.set` 复制 NICE DCV 服务器文件，然后将该文件放在您的 RLM 服务器上的 `C:\RLM\` 文件夹中。
4. 将 RLM 服务器安装为 Windows 服务。

```
C:\> rlm.exe -nows -dlog C:\RLM\rlm.log -c C:\RLM\license -install_service -  
service_name dcv-rlm
```

有关 RLM 启动选项的更多信息，请参阅 [RLM 许可证管理手册](#)。

5. 启动 RLM 服务器。

```
C:\> net start dcv-rlm
```

6. 确认 RLM 服务器正在运行。
 - a. 使用您的首选文本编辑器打开 `C:\RLM\nice.dlog`，并确认显示以下行。

```
date_time (nice) Server started on license1 (hostid: host_id) for: dcv dcv-gl
```

Note

rlm.log 文件的内容可能会随 RLM 服务器版本而异。

- b. 运行以下命令。

```
C:\RLM\rlmutil rlmstat -a -c rlm_server_hostname@5053
```

该命令应返回有关 RLM 服务器的信息。

在 Linux 上配置 RLM 服务器

在 Linux 上配置 RLM 服务器

1. 将编辑后的 license.lic 文件复制到 /opt/nice/rlm/license/.
2. 从 /usr/share/dcv/license/nice.set 复制 NICE DCV 服务器文件，然后将该文件放在您的 RLM 服务器上的 /opt/nice/rlm 中。
3. 创建 RLM 服务器服务，并确保该服务在启动时自动启动。
 - a. 在 dcv-rlm 文件夹中创建名为 /opt/nice/rlm/ 的文件：

```
$ touch /opt/nice/rlm/dcv-rlm
```

- b. 使用您常用的文本编辑器打开文件，并添加以下脚本。保存并关闭文件。

```
#!/bin/sh
# chkconfig: 35 99 01
# description: The Reprise License Manager daemon.
# processname: dcv-rlm

### BEGIN INIT INFO
# Provides: dcv-rlm
# Required-Start: $local_fs $remote_fs $syslog
# Required-Stop: $local_fs $remote_fs $syslog
# Default-Start: 3 4 5
# Default-Stop: 0 1 2 6
# Short-Description: The Reprise License Manager daemon.
# Description: A service that runs the Reprise License Manager daemon.
### END INIT INFO

# user used to run the daemon
RLM_USER="rlm"

# root of rlm installation
RLM_ROOT="/opt/nice/rlm"

# license directory (license files should have .lic extension)
RLM_LICENSE_DIR="/opt/nice/rlm/license"

# log file
RLM_LOG_FILE="/var/log/rlm.log"

_getpid() {
    pidof -o $$ -o $PPID -o %PPID -x "$1"
}

start() {
    echo -n "Starting rlm: "
    touch ${RLM_LOG_FILE}
    chown "${RLM_USER}" ${RLM_LOG_FILE}
}
```



```
su -p -s /bin/sh "${RLM_USER}" -c "${RLM_ROOT}/rlm -c ${RLM_LICENSE_DIR} \  
-nows -dlog +${RLM_LOG_FILE} &"  
if [ $? -ne 0 ]; then  
    echo "FAILED"  
    return 1  
fi  
echo "OK"  
}  
  
stop() {  
    echo -n "Stopping rlm: "  
    pid=`_getpid ${RLM_ROOT}/rlm`  
    if [ -n "$pid" ]; then  
        kill $pid >/dev/null 2>&1  
        sleep 3  
        if [ -d "/proc/$pid" ] ; then  
            echo "FAILED"  
            return 1  
        fi  
    fi  
    echo "OK"  
}  
  
status() {  
    pid=`_getpid ${RLM_ROOT}/rlm`  
    if [ -z "$pid" ]; then  
        echo "rlm is stopped"  
        return 3  
    fi  
    echo "rlm (pid $pid) is running..."  
    return 0  
}  
  
restart() {  
    stop  
    start  
}  
  
case "$1" in  
    start)  
        start  
        ;;  
    stop)  
        stop  
        ;;  
    status)  
        status  
        ;;  
    restart)  
        restart  
        ;;  
    *)  
        echo $"Usage: $0 {start|stop|status|restart}"  
        exit 1  
esac  
  
exit $?  
  
# ex:ts=4:et:
```

- c. 将脚本编写为可执行文件，将其复制到 /etc/init.d/，然后将其添加到 chkconfig 实用程序：

```
chmod +x /opt/nice/rlm/dcv-rlm
```

```
cp -a /opt/nice/rlm/dcv-rlm /etc/init.d/
```

```
chkconfig --add dcv-rlm
```

4. 启动 RLM 服务器：

```
$ service dcv-rlm start
```

5. 确保 RLM 服务器正在按预期运行和工作。使用您常用的文本编辑器打开 `var/log/rlm.log`，并确认显示以下行：

```
date_time (nice) Server started on license1 (hostid: host_id) for: dcv dcv-gl
```

Note

`rlm.log` 文件的内容可能会随 RLM 服务器版本而异。

步骤 6：配置 NICE DCV 服务器

配置 NICE DCV 服务器以使用 RLM 服务器。要执行此操作，您必须配置 `license-file` 上的 NICE DCV 服务器。配置参数。

主题

- [Windows NICE DCV 服务器配置 \(p. 37\)](#)
- [Linux NICE DCV 服务器配置 \(p. 37\)](#)

Windows NICE DCV 服务器配置

配置 Windows 服务器上的 `license-file` 配置参数

1. 打开 Windows 注册表编辑器。
2. 导航到 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/license/` 注册表项，并选择 `license-file` 参数。

如果该注册表项中没有 `license-file` 参数，您必须创建它：

- a. 在左侧面板中打开 `license` 注册表项的上下文（右键单击）菜单，然后依次选择新建和字符串值。
 - b. 对于名称，输入 `license-file` 并按 Enter。
3. 打开 `license-file` 参数。对于 Value data（值数据），输入 `5053@RLM_server_hostname` 格式的 RLM 服务器的端口号和主机名。

Note

您可以使用 RLM 服务器 IP 地址而不是主机名。

4. 选择 OK，然后关闭 Windows 注册表编辑器。

Linux NICE DCV 服务器配置

配置 Linux 服务器上的 `license-file` 配置参数

1. 导航到 `/etc/dcv/`，并使用您常用的文本编辑器打开 `dcv.conf`

2. 找到 `license-file` 部分中的 `[license]` 参数，并使用 `5053@RLM_server_hostname` 格式的 RLM 服务器端口和主机名替换现有路径。

如果 `license-file` 部分中没有 `[license]` 参数，则使用以下格式手动添加此参数：

```
license-file = "5053@RLM_server_hostname"
```

Note

您可以使用 RLM 服务器 IP 地址而不是主机名。

3. 保存并关闭 文件。

升级 NICE DCV 服务器

以下主题说明了如何升级 NICE DCV 服务器。

注意事项

- NICE DCV 服务器版本 2017 及更高版本与 NICE DCV 客户端版本 2017 及更高版本兼容。
- NICE DCV 服务器版本 2020 与 NICE DCV 服务器版本 2019 及更早版本中的生产许可证和延期评估文件不兼容。如果升级到 NICE DCV 服务器版本 2020，则必须请求兼容的许可证文件。有关更多信息，请联系您的 NICE DCV 分销商或经销商。
- NICE DCV 服务器版本 2020 许可证文件与 NICE DCV 服务器版本 2017 和 2019 向后兼容。

目录

- [在 Windows 上升级 NICE DCV 服务器 \(p. 38\)](#)
- [在 Linux 上升级 NICE DCV 服务器 \(p. 39\)](#)

在 Windows 上升级 NICE DCV 服务器

在 Windows 上升级 NICE DCV 服务器

1. 使用 RDP 客户端以管理员身份连接到 NICE DCV 服务器
2. 确保不存在正在运行的 NICE DCV 会话。使用 `dcv list-sessions NICE DCV` 命令检查所有正在运行的会话。如果存在正在运行的会话，请使用 `dcv close session NICE DCV` 命令将其停止。
3. 在确认没有正在运行的会话后，停止 NICE DCV 服务器。有关更多信息，请参阅 [在 Windows 上停止 NICE DCV 服务器 \(p. 42\)](#)。
4. 对 NICE DCV 服务器配置进行备份。打开注册表编辑器，导航到 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv`，右键单击 `dcv` 注册表项，然后选择导出。
5. 从 NICE DCV 服务器 [NICE 网站下载最新版本的](#)。
6. 执行 [使用向导 \(p. 8\)](#) 中所述的步骤，从步骤 3 开始。
7. 安装完成后，确认 NICE DCV 服务器配置仍然正确。打开注册表编辑器，导航到 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv`，并将参数与您在步骤 4 中导出的配置进行比较。
8. 通过启动新的 NICE DCV 服务器测试 NICE DCV 会话。有关更多信息，请参阅 [启动 NICE DCV 会话 \(p. 66\)](#)。

在 Linux 上升级 NICE DCV 服务器

在 Linux 上升级 NICE DCV 服务器

1. 使用 SSH 通过 root 用户登录到服务器。
2. 确保不存在正在运行的 NICE DCV 会话。使用 `dcv list-sessions` NICE DCV 命令检查所有正在运行的会话。如果存在正在运行的会话，请使用 `dcv close session` NICE DCV 命令将其停止。
3. 在确认没有正在运行的会话后，停止 NICE DCV 服务器。有关更多信息，请参阅 [在 Linux 上停止 NICE DCV 服务器 \(p. 42\)](#)。
4. 对 NICE DCV 服务器配置进行备份。将 `/etc/dcv/dcv.conf` 文件复制到安全位置。
5. 执行中所述的步骤。 [安装 NICE DCV 服务器 \(p. 17\)](#)。
6. 安装完成后，确认 NICE DCV 服务器配置仍然正确。打开您在步骤 4 中复制的文件，并将该文件与 `/etc/dcv/dcv.conf` 文件进行比较。
7. 通过启动新的 NICE DCV 服务器测试 NICE DCV 会话。有关更多信息，请参阅 [启动 NICE DCV 会话 \(p. 66\)](#)。

管理 NICE DCV 服务器

NICE DCV 服务器 作为操作系统服务运行。您必须以管理员 (Windows) 或根用户 (Linux) 身份登录才能启动、停止或配置 NICE DCV 服务器。

主题

- [启动 NICE DCV 服务器 \(p. 40\)](#)
- [停止 NICE DCV 服务器 \(p. 41\)](#)
- [更改 NICE DCV 服务器 TCP 端口 \(p. 43\)](#)
- [断开空闲客户端连接 \(p. 44\)](#)
- [在 Linux 上启用 GPU 共享 NICE DCV 服务器 \(p. 45\)](#)
- [更改 TLS 证书 \(p. 46\)](#)
- [启用 USB 远程处理 \(p. 47\)](#)
- [配置智能卡缓存 \(p. 48\)](#)
- [启用会话存储 \(p. 49\)](#)
- [在 Linux 上配置剪贴板 NICE DCV 服务器 \(p. 50\)](#)
- [启用触摸屏和触控笔支持 \(p. 51\)](#)
- [配置多通道音频 \(p. 53\)](#)
- [启用 QUIC UDP 传输协议 \(p. 55\)](#)
- [配置 HTTP 标头 \(p. 56\)](#)
- [配置 NICE DCV 身份验证 \(p. 57\)](#)
- [配置 NICE DCV 授权 \(p. 59\)](#)

启动 NICE DCV 服务器

NICE DCV 服务器 必须运行才能托管会话。

默认情况下，NICE DCV 服务器 配置为在所托管于的服务器启动时自动启动。如果您选择禁止服务器在安装 NICE DCV 服务器时自动启动，则必须使用以下过程手动启动服务器。

主题

- [在 Windows 上启动 NICE DCV 服务器 \(p. 40\)](#)
- [在 Linux 上启动 NICE DCV 服务器 \(p. 41\)](#)

在 Windows 上启动 NICE DCV 服务器

使用以下过程可通过 Microsoft 管理控制台的“服务”管理单元手动启动 NICE DCV 服务器

在 Windows 上启动 NICE DCV 服务器

1. 打开 Microsoft 管理控制台的“服务”管理单元。
2. 在右侧窗格中，打开 DCV 服务器。

3. 选择 Start (开始).

Note

如果服务器已在运行中，则开始按钮处于禁用状态。

使用以下过程可通过 Microsoft 管理控制台的“服务”管理单元将 NICE DCV 服务器配置为自动启动。

在 Windows 上将 NICE DCV 服务器 配置为自动启动

1. 打开 Microsoft 管理控制台的“服务”管理单元。
2. 在右侧窗格中，打开 DCV 服务器。
3. 对于启动服务，选择自动。

在 Linux 上启动 NICE DCV 服务器

使用以下过程可通过命令行手动启动 NICE DCV 服务器

在 Linux 上启动 NICE DCV 服务器

使用以下命令：

- RHEL 6.x 和 CentOS 6.x

```
$ sudo service dcvserver start
```

- RHEL 7.x/8.x、CentOS 7.x/8.x、SUSE Linux Enterprise 12 和 Ubuntu 18.x

```
$ sudo systemctl start dcvserver
```

使用以下过程可通过命令行将 NICE DCV 服务器 配置为自动启动。

在 Linux 上将 NICE DCV 服务器 配置为自动启动

使用以下命令：

- RHEL 6.x 和 CentOS 6.x

```
$ sudo chkconfig --add dcvserver
```

- RHEL 7.x/8.x、CentOS 7.x/8.x、SUSE Linux Enterprise 12 和 Ubuntu 18.x

```
$ sudo systemctl enable dcvserver
```

停止 NICE DCV 服务器

您可以随时停止 NICE DCV 服务器 停止服务器将终止所有活动的 NICE DCV 会话。在重新启动服务器之前，您无法启动新会话。

主题

- [在 Windows 上停止 NICE DCV 服务器 \(p. 42\)](#)

- [在 Linux 上停止 NICE DCV 服务器 \(p. 42\)](#)

在 Windows 上停止 NICE DCV 服务器

使用以下过程可通过 Microsoft 管理控制台的“服务”管理单元停止 NICE DCV 服务器

在 Windows 上停止 NICE DCV 服务器

1. 打开 Microsoft 管理控制台的“服务”管理单元。
2. 在右侧窗格中，打开 DCV 服务器。
3. 选择 Stop. (停止)。

Note

如果服务器已停止，则 Stop (停止) 按钮处于禁用状态。

使用以下过程可通过 Microsoft 管理控制台的“服务”管理单元禁止自动启动 NICE DCV 服务器

在 Windows 上阻止 NICE DCV 服务器 自动启动

1. 打开 Microsoft 管理控制台的“服务”管理单元。
2. 在右侧窗格中，打开 DCV 服务器。
3. 对于启动服务，选择手动。

在 Linux 上停止 NICE DCV 服务器

使用以下过程可通过命令行停止 NICE DCV 服务器

在 Linux 上停止 NICE DCV 服务器

使用以下命令：

- RHEL 6.x 和 CentOS 6.x

```
$ sudo service dcvserver stop
```

- RHEL 7.x/8.x、CentOS 7.x/8.x 和 SUSE Linux Enterprise 12

```
$ sudo systemctl stop dcvserver
```

使用以下过程可通过命令行禁用自动 NICE DCV 服务器 启动。

在 Linux 上阻止 NICE DCV 服务器 自动启动

使用以下命令：

- RHEL 6.x 和 CentOS 6.x

```
$ sudo chkconfig --del dcvserver
```

- RHEL 7.x/8.x、CentOS 7.x/8.x 和 SUSE Linux Enterprise 12

```
$ sudo systemctl disable dcvserver
```

更改 NICE DCV 服务器 TCP 端口

默认情况下，NICE DCV 服务器配置为通过端口 8443 进行通信。您可以在安装 NICE DCV 服务器后指定自定义 TCP 端口。该端口必须高于 1024。

要允许 NICE DCV 客户端通过标准 HTTPS 端口 (443) 访问 NICE DCV 服务器，建议您使用 Web 代理或负载均衡器作为前端网关来将客户端连接重定向到服务器。

确保将所有端口更改都传达给客户端。它们需要端口号才能连接到会话。

主题

- 在 [Windows 上更改 NICE DCV 服务器的 TCP 端口](#) (p. 43)
- 在 [Linux 上更改 NICE DCV 服务器 TCP 端口](#) (p. 43)

在 Windows 上更改 NICE DCV 服务器的 TCP 端口

要更改 NICE DCV 服务器使用的端口，您必须使用 Windows 注册表编辑器配置 `web-port` 参数。

在 Windows 上更改服务器的 TCP 端口

1. 打开 Windows 注册表编辑器。
2. 导航到 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/` 注册表项，并选择 `web-port` 参数。

如果该注册表项中没有 `web-port` 参数，请创建一个：

- a. 在左侧窗格中，打开 `connectivity` 注册表项的上下文（右键单击）菜单，并依次选择新建和 DWORD (32 位)值。
 - b. 对于名称，输入 `web-port` 并按 Enter。
3. 打开 `web-port` 参数。对于 Value data（值数据），输入新的 TCP 端口号。

Note

TCP 端口号必须高于 1024。

4. 选择 OK，然后关闭 Windows 注册表编辑器。
5. [停止](#) (p. 41)，然后 [\(p. 40\)](#) 重新启动 NICE DCV 服务器。它。

在 Linux 上更改 NICE DCV 服务器 TCP 端口

要更改 NICE DCV 服务器使用的端口，您必须在 `web-port` 文件中配置 `dcv.conf` 参数。

在 Linux 上更改服务器的 TCP 端口

1. 导航到 `/etc/dcv/`，并使用您常用的文本编辑器打开 `dcv.conf`
2. 在 `web-port` 部分中找到 `[connectivity]` 参数，并将现有 TCP 端口号替换为新的 TCP 端口号。

如果 `web-port` 部分中没有 `[connectivity]` 参数，则使用以下格式手动添加此参数：

```
[connectivity]
web-port=port_number
```

Note

TCP 端口号必须高于 1024。

3. 保存并关闭 文件。
4. [停止 \(p. 41\)](#) , 然后 [\(p. 40\)](#)重新启动NICE DCV 服务器.它。

断开空闲客户端连接

可将 NICE DCV 配置为断开在指定时段内未向 NICE DCV 服务器发送任何键盘或指针输入的空闲客户端的连接。默认情况下，NICE DCV 服务器会断开已处于空闲状态 60 分钟的 NICE DCV 客户端的连接。

您还可以将 NICE DCV 服务器配置为向空闲客户端发送通知，以通知其会话即将断开连接。仅 NICE DCV 服务器和客户端版本 2017.4 及更高版本支持超时通知。

您可以使用以下过程来指定自定义空闲超时期限。

主题

- [在 Windows 上更改空闲超时期限 \(p. 44\)](#)
- [在 Linux 上更改空闲超时期限 \(p. 45\)](#)

在 Windows 上更改空闲超时期限

要更改 NICE DCV 服务器的空闲超时期限，您必须使用 Windows 注册表编辑器配置 `idle-timeout` 参数。

在 Windows 上更改空闲超时期限

1. 打开 Windows 注册表编辑器。
2. 导航到 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/` 注册表项，并选择 `idle-timeout` 参数。

如果此参数不存在，请使用以下步骤进行创建：

- a. 在左侧窗格中，打开 `connectivity` 项的上下文（右键单击）菜单，并依次选择新建和 DWORD (32 位)值。
 - b. 对于名称，输入 `idle-timeout` 并按 Enter。
3. 打开 `idle-timeout` 参数。对于 Value data（值数据），输入空闲超时期限的值（以分钟为单位）。要避免断开空闲客户端的连接，请输入 0。
 4. 选择 OK 并关闭 Windows 注册表编辑器。

（可选）将 NICE DCV 服务器配置为向空闲客户端发送超时通知

1. 导航到 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/` 注册表项，并选择 `idle-timeout-warning` 参数。

如果此参数不存在，请使用以下步骤进行创建：

- a. 在左侧窗格中，打开 `connectivity` 注册表项的上下文（右键单击）菜单，并依次选择新建和 DWORD (32 位)值。
 - b. 对于名称，输入 `idle-timeout-warning` 并按 Enter。
2. 打开 `idle-timeout-warning` 参数。对于 Value data（值数据），输入在空闲超时断开连接之前应发送通知的时间长度（以秒为单位）。例如，如果您希望在达到空闲超时前两分钟发送通知，请输入 120。
 3. 选择 OK 并关闭 Windows 注册表编辑器。

在 Linux 上更改空闲超时期限

要更改 NICE DCV 服务器的空闲超时期限，您必须在 `idle-timeout` 文件中配置 `dcv.conf` 参数。

在 Linux 上更改空闲超时期限

1. 使用首选文本编辑器打开 `/etc/dcv/dcv.conf`
2. 在 `idle-timeout` 部分中找到 `[connectivity]` 参数，并将现有超时期限替换为新的超时期限（以分钟为单位）。

如果 `idle-timeout` 部分中没有 `[connectivity]` 参数，则使用以下格式手动添加此参数：

```
[connectivity]
idle-timeout=timeout_in_minutes
```

要避免断开空闲客户端的连接，请输入 0。

3. （可选）要将 NICE DCV 服务器配置为向空闲客户端发送超时通知，请将 `idle-timeout-warning` 参数添加到 `[connectivity]` 部分，并指定应先于空闲超时断开连接多长时间（以秒为单位）发送通知。

```
idle-timeout-warning=seconds_before_idle_timeout
```

例如，如果您希望在达到空闲超时前的两分钟发送通知，请指定 120。

4. 保存并关闭文件。

在 Linux 上启用 GPU 共享 NICE DCV 服务器

利用 GPU 共享，您可以在多个 GPUs 虚拟会话之间共享一个或多个物理 NICE DCV。有关会话的更多信息，请参阅[管理 NICE DCV 会话 \(p. 64\)](#)。GPU 共享使您能够使用一个 NICE DCV 服务器并托管多个共享服务器的物理 GPU 资源的虚拟会话。

Note

GPU 共享仅在 Linux NICE DCV 服务器上受支持。

Prerequisites

在开始之前，请满足以下先决条件：

- 在 Linux 服务器上安装 NICE DCV 服务器
- 在服务器上安装 NICE DCV `dcv-gl` 和 `nice-xdcv` 程序包。
- 确保服务器至少有一个支持的 NVIDIA GPU。
- 在服务器上安装 NVIDIA GPU 驱动程序。需要正式 NVIDIA 驱动程序。不支持开源 NVIDIA 驱动程序。
- 确保 NVIDIA GPU 驱动程序支持硬件加速的 OpenGL。
- 安装 X 服务器，并配置 `Device` 文件中的 `Screen` 和 `xorg.conf` 部分。

Note

您可以使用 `nvidia-xconfig` NVIDIA 实用工具自动创建 `xorg.conf` 文件并为所有可用的 NVIDIA GPUs 配置该文件。

- 确保 X 服务器正在运行。
- （可选）通过运行 NICE DCV 服务器工具验证 `dcvglldiag` 配置。有关更多信息，请参阅[安装后检查 \(p. 26\)](#)。

您也可以安装 `nice-dcv-glttest` 程序包并运行 `dcv-glttest` 测试应用程序以检查是否已为 GPU 共享正确配置服务器。

要启用 GPU 共享，您必须指定虚拟会话要使用的 GPUs 列表。如果未指定 GPUs，则仅使用标准 X 服务器使用的 GPU，显示名称为 `:0.0`。

在完成本主题前面所述的先决条件后，您必须在 GPUs 文件的 `gl-displays` 参数中指定 `dcv.conf`。

在 Linux 上启用 GPU 共享 NICE DCV 服务器

1. 导航到 `/etc/dcv/`，并使用您的首选文本编辑器打开 `dcv.conf` 文件。
2. 添加 `[display/linux]` 部分和 `gl-displays` 参数，然后采用以下格式指定可用的 GPUs：

```
[display/linux]
gl-displays = [':xserver_port.screen_number_1',':xserver_port.screen_number_2', ...]
```

其中 `xserver_port` 是服务器和 `screen_number` 是与 GPU 关联的屏幕所关联的编号。`screen_number` 从 0 开始。

以下示例显示了在默认 X 服务器会话上运行的两个 `gl-displays` 的 GPUs 参数：

```
[display/linux]
gl-displays = [':0.0',':0.1']
```

3. 保存并关闭文件。
4. [停止 \(p. 41\)](#)，然后 [\(p. 40\)](#) 重新启动 NICE DCV 服务器。它。

更改 TLS 证书

NICE DCV 自动生成自签名证书，该证书用于保护 NICE DCV 客户端和 NICE DCV 服务器之间的流量。如果未在上安装其他证书，则默认使用此证书。NICE DCV 服务器默认证书包含两个文件，即证书本身 (`dcv.pem`) 和密钥 (`dcv.key`)。

您可以将默认 NICE DCV 证书及其密钥替换为您自己的证书和密钥。

Important

如果您使用自己的证书和密钥，则必须将证书命名为 `dcv.pem`，并且必须将密钥命名为 `dcv.key`。

更改 NICE DCV 服务器的 TLS 证书

- Windows NICE DCV 服务器

将您的证书及其密钥置于 Windows 上的以下位置：NICE DCV 服务器：

```
C:\Windows\System32\config\systemprofile\AppData\Local\NICE\dcv\
```

- Linux NICE DCV 服务器

将您的证书及其密钥置于 Linux 上的以下位置：NICE DCV 服务器：

```
/etc/dcv/
```

将这两个文件的所有权授予 dcv 用户，并将其权限更改为 600（只有所有者可以读取或写入它们）。

```
$ sudo chown dcv dcv.pem dcv.key
```

```
$ sudo chmod 600 dcv.pem dcv.key
```

启用 USB 远程处理

NICE DCV 使客户端能够使用一些专用 USB 设备，如 3D 指点设备或绘图板。设备物理连接到其计算机，以便与上运行的应用程序交互。NICE DCV 服务器。

Important

NICE DCV 提供一种重定向 USB 设备的通用机制。一些对网络延迟敏感的设备可能会遇到问题。另外，由于驱动程序兼容性问题，某些设备可能无法按预期正常运行。在部署到生产环境之前，请确保您的设备按预期工作。

Note

仅 Windows 客户端支持 USB 远程处理。便携式 Windows 客户端或 Web 浏览器客户端不支持 USB 远程处理。上可能需要其他配置。NICE DCV 客户端。有关更多信息，请参阅 [用户指南](#) 中的使用 USB 远程处理 NICE DCV。

使用 NICE DCV 服务器允许列表确定客户端可以使用哪些 USB 设备。默认情况下，一些常用的 USB 设备会添加到允许列表中。这使客户可以将这些 USB 设备连接到他们的计算机，并在服务器上使用它们而无需任何其他配置。

但默认情况下，某些专用设备可能不会添加到允许列表中。必须先将这些设备手动添加到 NICE DCV 服务器上的允许列表单中，之后它们才可供客户端使用。添加后，它们会显示在 Windows 客户端的 Settings（设置）菜单中。

主题

- [在 Windows 上将 USB 设备添加到允许列表 NICE DCV 服务器 \(p. 47\)](#)
- [在 Linux 上将 USB 设备添加到允许列表 NICE DCV 服务器 \(p. 48\)](#)

在 Windows 上将 USB 设备添加到允许列表 NICE DCV 服务器

要将 USB 设备添加到允许列表，您必须从客户端获取 USB 设备的筛选条件字符串并将其添加到 `usb-devices.conf` 文件中。

在 Windows 上将 USB 设备添加到允许列表 NICE DCV 服务器

1. 确保您已安装最新版本的 NICE DCV 服务器，并且已选择安装 USB 远程处理驱动程序。有关更多信息，请参阅 [在 Windows 上安装 NICE DCV 服务器 \(p. 6\)](#)。
2. 在上安装 USB 设备的硬件驱动程序。NICE DCV 服务器。
3. 从客户端请求筛选器字符串。有关更多信息，请参阅 [用户指南](#) 中的使用 USB 远程处理 NICE DCV。
4. 使用首选文本编辑器打开 `C:\Program Files\NICE\DCV\Server\conf\usb-devices.conf`，并将筛选器字符串添加到文件底部的新行。
5. 保存并关闭文件。
6. **停止**，然后重新启动 NICE DCV 服务器。它。

在 Linux 上将 USB 设备添加到允许列表 NICE DCV 服务器

要将 USB 设备添加到允许列表，您必须从客户端获取 USB 设备的筛选条件字符串并将其添加到 `usb-devices.conf` 文件中。

在 Linux 上将 USB 设备添加到允许列表 NICE DCV 服务器

1. 确保您已安装 NICE DCV 服务器和 DCV USB 驱动程序的最新版本。有关更多信息，请参阅 [在 Linux 上安装 NICE DCV 服务器 \(p. 9\)](#)。
2. 在上安装 USB 设备的硬件驱动程序。NICE DCV 服务器。
3. 从客户端请求筛选器字符串。有关更多信息，请参阅 [用户指南](#) 中的使用 USB 远程处理 NICE DCV。
4. 使用首选文本编辑器打开 `/etc/dcv/usb-devices.conf`，并将筛选器字符串添加到文件底部的新行。
5. 保存并关闭文件。
6. **停止**，然后重新启动 NICE DCV 服务器。它。

配置智能卡缓存

智能卡缓存功能使 NICE DCV 服务器能够缓存智能卡值。启用此功能时，NICE DCV 服务器缓存最近调用客户端智能卡的结果。未来的调用直接从服务器的缓存中检索，而不是从客户端检索。这有助于减少在客户端和服务器之间传输的流量，并提高性能。如果客户端的互联网连接速度较慢，则这特别有用。

默认情况下禁用智能卡缓存。客户可以通过设置 `DCV_PCSC_ENABLE_CACHE` 环境变量为他们运行的每个应用程序手动启用智能卡缓存。有关更多信息，请参阅 <https://docs.amazonaws.cn/dcv/latest/userguide/using-smartcard.html> 用户指南 中的 NICE DCV 使用智能卡。或者，您可以将 NICE DCV 服务器配置为永久启用或禁用智能卡缓存，而不考虑为 `DCV_PCSC_ENABLE_CACHE` 环境变量指定的值。

Linux NICE DCV 服务器

在 Linux 上永久启用或禁用智能卡缓存 NICE DCV 服务器

1. 导航到 `/etc/dcv/`，并使用您常用的文本编辑器打开 `dcv.conf`
2. 在 `enable-cache` 部分中找到 `[smartcard]` 参数。要永久启用智能卡缓存，请输入 `'always-on'`。要永久禁用智能卡缓存，请输入 `'always-off'`。

如果 `enable-cache` 部分中没有 `[smartcard]` 参数，则使用以下格式手动添加此参数：

```
[smartcard]
enable-cache='always-on'|'always-off'
```

3. 保存并关闭文件。
4. **停止 (p. 41)**，然后 **(p. 40)** 重新启动 NICE DCV 服务器。它。

Windows NICE DCV 服务器

在 Windows NICE DCV 服务器上永久启用或禁用智能卡缓存

1. 打开 Windows 注册表编辑器。
2. 导航到 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/smartcard/` 注册表项，并选择 `enable-cache` 参数。

如果此参数不存在，请使用以下步骤进行创建：

- a. 在左侧窗格中，打开 smartcard 键的上下文（右键单击）菜单，然后依次选择 New 和 String Value。
- b. 对于名称，输入 enable-cache 并按 Enter。
3. 打开 enable-cache 参数。对于 Value data（值数据），输入 always-on 以永久启用智能卡缓存，或输入 always-off 以永久禁用智能卡缓存。
4. 选择 OK 并关闭 Windows 注册表编辑器。

启用会话存储

会话存储是 NICE DCV 服务器上的一个文件夹，客户端在连接到特定的 NICE DCV 会话时可访问该文件夹。当您启用会话的会话存储时，客户端既可从指定文件夹下载文件，又可将文件上传到指定文件夹。利用此功能，客户端可在连接到会话时共享文件。

主题

- 在 [Windows 上启用会话存储NICE DCV 服务器 \(p. 49\)](#)
- 在 [Linux 上启用会话存储NICE DCV 服务器 \(p. 50\)](#)

在 Windows 上启用会话存储NICE DCV 服务器

要启用会话存储，请先创建要用于会话存储的文件夹。然后，使用 Windows 注册表编辑器配置 storage-root 参数。

在 Windows 上启用会话存储

1. 创建要用于会话存储的文件夹（例如 c:\session-storage）。
2. 配置 storage-root 参数。
 - a. 打开 Windows 注册表编辑器。
 - b. 导航到 HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/session-management/automatic-console-session 注册表项，并选择 storage-root 参数。

如果该注册表项中没有 storage-root 参数，请创建一个此参数，如下所示：

- i. 在左侧窗格中，打开 session-management/automatic-console-session 注册表项的上下文（右键单击）菜单，然后依次选择新建和字符串。
- ii. 对于名称，输入 storage-root 并按 Enter。
- c. 打开 storage-root 参数。对于 Value data（值数据），输入步骤 1 中创建的文件夹的完整路径。

您还可以在路径中使用 %home% 来指定当前已登录的用户的主目录。例如，以下路径使用 c:\Users\username\storage\ 作为会话存储目录。

```
%home%/storage/
```

Note

如果指定的子目录不存在，则将禁用会话存储。

- d. 选择 OK，然后关闭 Windows 注册表编辑器。
- e. [停止 \(p. 41\)](#)，然后 [\(p. 40\)](#) 重新启动 NICE DCV 服务器。它。
3. 启动会话并指定 --storage-root 选项。有关更多信息，请参阅 [启动 NICE DCV 会话 \(p. 66\)](#)。

在 Linux 上启用会话存储NICE DCV 服务器

要启用会话存储，您必须创建用于会话存储的文件夹，然后配置 `storage-root` 文件中的 `dcv.conf` 参数。

在 Linux 上启用会话存储

1. 创建要用于会话存储的文件夹（例如 `/opt/session-storage/`）。
2. 配置 `storage-root` 参数。
 - a. 导航到 `/etc/dcv/`，并使用您常用的文本编辑器打开 `dcv.conf`
 - b. 在 `storage-root` 部分中找到 `[session-management/automatic-console-session]` 参数。将现有路径替换为步骤 1 中创建的文件夹的完整路径。

如果 `storage-root` 部分中没有 `[session-management/automatic-console-session]` 参数，则使用以下格式手动添加此参数。

```
[session-management/automatic-console-session]
storage-root="/opt/session-storage/"
```

您还可以在路径中使用 `%home%` 来指定当前已登录的用户的主目录。例如，以下参数使用会话存储的 `$HOME/storage/` 目录。

```
[session-management/automatic-console-session]
storage-root="%home%/storage/"
```

Note

如果指定的子目录不存在，则将禁用会话存储。

3. 保存并关闭文件。
4. [停止 \(p. 41\)](#)，然后 [\(p. 40\)](#)重新启动NICE DCV 服务器。它。
5. 启动会话并指定 `--storage-root` 选项。有关更多信息，请参阅 [启动 NICE DCV 会话 \(p. 66\)](#)。

在 Linux 上配置剪贴板NICE DCV 服务器

Linux 操作系统提供了两个可在复制和粘贴内容时使用的缓冲区：主选区和剪贴板。要将内容复制到主选区，请使用鼠标光标突出显示内容。要从主选区粘贴内容，请使用鼠标的中间按钮或 `Shift+Insert` 键盘快捷键。要将内容复制到剪贴板，请突出显示内容并从上下文（右键单击）菜单中选择 `Copy`（复制）。要从剪贴板粘贴它，请从上下文（右键单击）菜单中选择 `Paste`（粘贴）。

在 Linux NICE DCV 服务器上，您可以将服务器配置为在客户端和服务器之间执行复制和粘贴操作时使用主选区或剪贴板。

主题

- [将客户端剪贴板内容粘贴到主选区 \(p. 50\)](#)
- [将主选区内容复制到客户端剪贴板 \(p. 51\)](#)

将客户端剪贴板内容粘贴到主选区

默认情况下，客户端中复制的内容将置于剪贴板中。要将此内容粘贴到服务器上，必须使用上下文（右键单击）菜单从剪贴板粘贴内容。

可以将服务器配置为将剪贴板内容放入主选区中。这使用户能够使用鼠标的中间按钮或 Shift+Insert 键盘快捷键，使用上下文（右键单击）菜单和主选区从剪贴板粘贴复制的内容。

将服务器配置为将剪贴板内容放入主选区中

1. 导航到 `/etc/dcv/`，并使用您常用的文本编辑器打开 `dcv.conf`
2. 在 `primary-selection-paste` 部分中找到 `[clipboard]` 参数，并将值设置为 `true`。

如果 `primary-selection-paste` 部分中没有 `[clipboard]` 参数，则使用以下格式手动添加此参数：

```
[clipboard]
primary-selection-paste=true
```

3. 保存并关闭 文件。
4. [停止 \(p. 69\)](#)，然后 [\(p. 66\)](#)重新启动NICE DCV 会话.它。

将主选区内容复制到客户端剪贴板

默认情况下，用户只能使用剪贴板将内容从服务器复制到客户端。这意味着无法将复制到主选区的内容粘贴到客户端上。

可以将服务器配置为将主选区内容放入剪贴板中。这意味着当用户将内容复制到服务器上的主选区时，内容也会复制到剪贴板中。这使用户能够将剪贴板中的内容粘贴到客户端中。

将服务器配置为将主选区内容放入剪贴板中

1. 导航到 `/etc/dcv/`，并使用您常用的文本编辑器打开 `dcv.conf`
2. 在 `primary-selection-copy` 部分中找到 `[clipboard]` 参数，并将值设置为 `true`。

如果 `primary-selection-copy` 部分中没有 `[clipboard]` 参数，则使用以下格式手动添加此参数：

```
[clipboard]
primary-selection-copy=true
```

3. 保存并关闭 文件。
4. [停止 \(p. 69\)](#)，然后 [\(p. 66\)](#)重新启动NICE DCV 会话.它。

启用触摸屏和触控笔支持

所有受支持的 Windows 操作系统都支持触摸屏。仅 Windows 10 和 Windows Server 2019 支持触控笔。默认情况下，Windows NICE DCV 服务器上已启用这些功能。无需其他配置。

所有受支持的 Linux 操作系统都支持触摸屏和触控笔。默认情况下，已在 Linux NICE DCV 服务器上托管的虚拟会话上启用这些功能。但是，需要一些额外配置才能在 Linux NICE DCV 服务器上托管的控制台会话上启用这些功能。

为 Linux 上托管的控制台会话启用触摸屏和触控笔支持NICE DCV 服务器

1. 使用所需的文本编辑器打开 `/etc/X11/xorg.conf`
2. 将以下部分添加到此文件中。


```
Section "InputDevice"
  Identifier "DCV Stylus Pen"
  Driver "dcvinput"
EndSection

Section "InputDevice"
  Identifier "DCV Stylus Eraser"
  Driver "dcvinput"
EndSection

Section "InputDevice"
  Identifier "DCV Touchscreen"
  Driver "dcvinput"
EndSection
```

3. 将以下内容添加到 `ServerLayout` 部分的末尾。

```
InputDevice "DCV Stylus Pen"
InputDevice "DCV Stylus Eraser"
InputDevice "DCV Touchscreen"
```

例如：

```
Section "ServerLayout"
  ...existing content...
  InputDevice "DCV Stylus Pen"
  InputDevice "DCV Stylus Eraser"
  InputDevice "DCV Touchscreen"
EndSection
```

4. 保存更改并关闭该文件。
5. 重新启动 X 服务器。
 - RHEL 7.x/8.x、CentOs 7.x/8.x、Amazon Linux 2、Ubuntu 18.x 和 SUSE Linux Enterprise 12.x

```
$ sudo systemctl isolate multi-user.target
```

```
$ sudo systemctl isolate graphical.target
```

- RHEL 6.x 和 CentOs 6.x

```
$ sudo init 3
```

```
$ sudo init 5
```

6. 要确保正确配置输入设备，请运行以下命令。

```
$ sudo DISPLAY=:0 xinput
```

DCV 触控笔、DCV 触控笔擦除器和 DCV 触摸屏应显示在命令输出中。下面是示例输出。

```
| Virtual core pointer                id=2    [master pointer (3)]
| | Virtual core XTEST pointer        id=4    [slave pointer (2)]
| | dummy_mouse                       id=6    [slave pointer (2)]
| | dummy_keyboard                    id=7    [slave pointer (2)]
| | DCV Stylus Pen                   id=8    [slave pointer (2)]
```

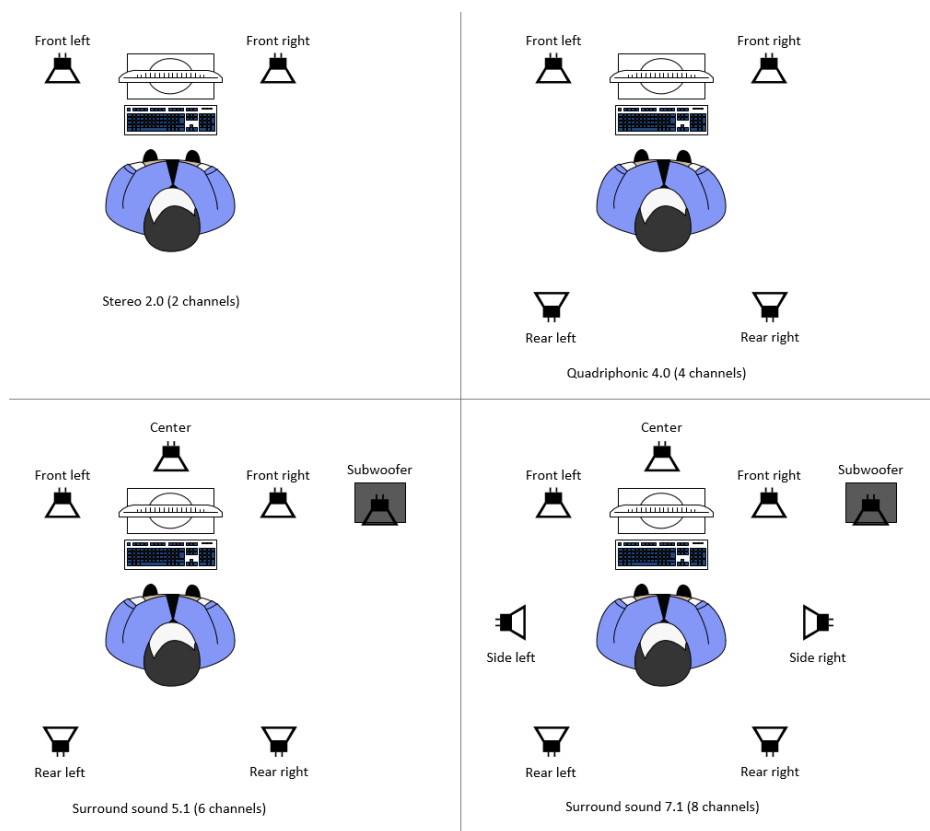
	DCV Stylus Eraser	id=9	[slave pointer (2)]
	DCV Touchscreen	id=10	[slave pointer (2)]
	Virtual core keyboard	id=3	[master keyboard (2)]
	Virtual core XTEST keyboard	id=5	[slave keyboard (3)]

配置多通道音频

使用 NICE DCV 本机客户端时，NICE DCV 最多支持 7.1 个音频通道。Web 浏览器客户端仅支持立体声 2.0 音频通道。

NICE DCV 支持以下多通道音频配置：

- 立体声 2.0 (2 个通道)
- Quadriphonic 4.0 (4 个通道)
- 环绕 5.1 (6 个通道)
- Curround 7.1 (8 通道) — 仅限 Windows NICE DCV 服务器



如果客户端请求数量的低于服务器提供的通道数的音频通道，则服务器将减少通道数以匹配客户端请求的通道。例如，如果客户端请求将声音 5.1 环绕，而服务器支持将声音 7.1 环绕，则服务器会自动将音频混音到 5.1。

服务器不会自动降低混音以匹配源应用程序的音频输出。例如，如果源应用程序提供环绕声音 7.1，而客户端仅支持立体声 2.0，则只有前左和前右音频通道会流式传输到客户端。其余通道将丢失。在这种情况下，您可能需要配置 NICE DCV 服务器 来降低音频通道，以防止音频通道丢失。

主题

- 在 [Windows NICE DCV 服务器上配置音频通道 \(p. 54\)](#)
- 在 [Linux NICE DCV 服务器上配置音频通道 \(p. 54\)](#)

在 Windows NICE DCV 服务器上配置音频通道

默认情况下，Windows 服务器支持环绕立体声 7.1 (8 个音频通道)。但是，您可以将服务器配置为使用不同的配置。

在 Windows 服务器上配置音频通道

1. 打开 Windows 注册表编辑器。
2. 导航到 HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/audio/ 注册表项，并选择 source-channels 参数。

如果该注册表项中没有 source-channels 参数，请创建一个：

- a. 在左侧窗格中，打开 audio 注册表项的上下文 (右键单击) 菜单，然后依次选择新建和 DWORD (32 位) 值。
 - b. 对于名称，输入 source-channels 并按 Enter。
3. 打开 source-channels 参数。对于 Value data (值数据)，输入以下内容之一：2 (对于 2.0)、4 (对于 4.0)、6 (对于 5.1) 或 8 (对于 7.1)。
 4. 选择 OK 并关闭 Windows 注册表编辑器。

在 Linux NICE DCV 服务器上配置音频通道

Linux 服务器默认支持立体声 2.0 (2 个音频通道)，并且需要一些额外配置才能支持多通道音频。

您需要执行以下操作：

1. 配置 PulseAudio 声音服务器。
2. 配置 NICE DCV 服务器 以使用 PulseAudio 设备。
3. 配置要使用的通道数。

配置 PulseAudio 声音服务器

1. 使用首选文本编辑器打开 /etc/pulse/default.pa
2. 将以下行添加到文件的末尾。

```
load-module module-null-sink sink_name=dcv format=s16be channels=6
channel_map=front-left,front-right,rear-left,rear-right,front-center,lfe rate=48000
sink_properties="device.description='DCV Audio Speakers'"
```

3. 保存并关闭 文件。

在配置 PulseAudio 声音服务器后，您必须配置 NICE DCV 服务器 以从 PulseAudio 声音服务器捕获音频。

配置 NICE DCV 服务器 以使用 PulseAudio 设备

1. 使用以下命令检索 PulseAudio 设备的名称。

```
$ C:\> pacmd list-sources
```

设备名称列在 `device.description` 字段中。

2. 使用首选文本编辑器打开 `/etc/dcv/dcv.conf`
3. 找到 `grab-device` 部分中的 `[audio]` 参数，并将现有值替换为您在上一步中检索到的设备名称。

如果 `grab-device` 部分中没有 `[audio]` 参数，则使用以下格式手动添加此参数：

```
[audio]
grab-device="device_name"
```

4. 保存并关闭 文件。

在配置 NICE DCV 服务器 以从 PulseAudio 声音服务器捕获音频后，您可以指定要使用的通道数。

配置要使用的通道数

1. 使用首选文本编辑器打开 `/etc/dcv/dcv.conf`
2. 找到 `source-channels` 部分中的 `[audio]` 参数，并将现有通道数替换为以下值之一：2（对于 2.0）、4（对于 4.0）或 6（对于 5.1）。

如果 `source-channels` 部分中没有 `[audio]` 参数，则使用以下格式手动添加此参数：

```
[audio]
source-channels=channels
```

3. 保存并关闭 文件。
4. [停止 \(p. 41\)](#)，然后 [\(p. 40\)](#)重新启动NICE DCV 服务器.它。

启用 QUIC UDP 传输协议

默认情况下，NICE DCV 使用基于 TCP 的 WebSocket 协议进行数据传输。

您可以将 NICE DCV 配置为使用基于 UDP 的 QUIC 传输协议进行数据传输。如果您的网络遇到高延迟和数据包丢失，则使用 QUIC 可以提高性能。如果您启用 QUIC，NICE DCV 服务器 将使用 QUIC 协议进行数据传输，但它会继续使用 WebSocket 进行身份验证流量。

Note

仅当您的网络和安全配置允许 UDP 流量时，才能使用 QUIC。

如果您启用 QUIC，客户端在连接到 NICE DCV 服务器 上托管的会话时，可以选择使用 QUIC 协议进行数据传输。如果客户端在连接时未明确选择 QUIC 协议，则默认情况下将使用 WebSocket。有关在连接到会话时选择 QUIC 协议的更多信息，请参阅 [用户指南NICE DCV 中的连接到 会话NICE DCV](#)。

Windows NICE DCV 服务器

将 NICE DCV 配置为使用 QUIC（UDP）进行数据传输

1. 打开 Windows 注册表编辑器并导航到 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/` 注册表项。
2. 打开 `enable-quick-frontend` 参数。对于 Value data（值数据），输入 1。

Note

如果该参数不存在，请创建新的 DWORD（32 位）参数并将其命名为 `enable-quick-frontend`。

3. (可选) 打开 quic-port 参数。对于 Value data (值数据), 输入要用于 QUIC 流量的端口。如果您不配置此参数, 则 NICE DCV 服务器 默认使用端口 8443。

Note

如果该参数不存在, 请创建新的 DWORD (32 位) 参数并将其命名为 quic-port。

4. (可选) 打开 web-port 参数。对于 Value data (值数据), 输入要用于 WebSocket (TCP) 流量的端口。如果您未配置此参数, NICE DCV 服务器 默认使用端口 8443。

Note

如果该参数不存在, 请创建新的 DWORD (32 位) 参数并将其命名为 web-port。

5. 关闭 Windows 注册表编辑器。
6. [停止 \(p. 41\)](#), 然后 [\(p. 40\)](#)重新启动NICE DCV 服务器.它。

Linux NICE DCV 服务器

将 NICE DCV 配置为使用 QUIC (UDP) 进行数据传输

1. 使用首选文本编辑器打开 /etc/dcv/dcv.conf
2. 请在 [connectivity] 部分执行以下操作：
 - 对于 enable-quic-frontend, 指定 true。
 - (可选) 对于 quic-port, 输入要用于 QUIC 流量的端口。如果您不配置此参数, 则 NICE DCV 服务器 默认使用端口 8443。
 - (可选) 对于 web-port, 输入要用于 WebSocket (TCP) 流量的端口。如果您未配置此参数, NICE DCV 服务器 默认使用端口 8443。

例如：

```
[connectivity]
enable-quic-frontend=true
quic-port=port_number
web-port=port_number
```

3. 保存并关闭 文件。
4. [停止 \(p. 41\)](#), 然后 [\(p. 40\)](#)重新启动NICE DCV 服务器.它。

配置 HTTP 标头

您可以将 NICE DCV 服务器配置为在用户使用 Web 浏览器客户端连接到会话时, 向 NICE DCV 客户端发送其他 HTTP 响应标头。响应标头可以提供有关用户正在连接到的 NICE DCV 服务器的其他信息。

主题

- [在 Windows 上配置 HTTP 标头NICE DCV 服务器 \(p. 56\)](#)
- [在 Linux 上配置 HTTP 标头NICE DCV 服务器 \(p. 57\)](#)

在 Windows 上配置 HTTP 标头NICE DCV 服务器

要在 Windows 上配置 HTTP 标头, 请使用 Windows 注册表编辑器配置 web-extra-http-headers 参数。

在 Windows 上配置 HTTP 标头

1. 打开 Windows 注册表编辑器。
2. 导航到 HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/ 注册表项。
3. 在左侧窗格中，打开 connectivity 注册表项的上下文（右键单击）菜单，并依次选择新建和字符串。
4. 对于名称，输入 web-extra-http-headers 并按 Enter。
5. 打开 web-extra-http-headers 参数。对于 Value data（值数据），采用以下格式输入 HTTP 标头名称和值：

```
[("header-name", "header-value")]
```

要指定多个标头，请将它们添加到以逗号分隔的列表中。例如：

```
[("header1-name", "header1-value"), ("header2-name", "header2-value")]
```

6. 选择 OK 并关闭 Windows 注册表编辑器。
7. [停止 \(p. 41\)](#)，然后 [\(p. 40\)](#)重新启动NICE DCV 服务器。它。

在 Linux 上配置 HTTP 标头NICE DCV 服务器

要在 Linux 上配置 HTTP 标头，请配置 web-extra-http-headers 文件中的 dcv.conf 参数。

在 Linux 上配置 HTTP 标头

1. 使用首选文本编辑器打开 /etc/dcv/dcv.conf
2. 找到 [connectivity] 部分。按以下格式指定 HTTP 标头名称和值：

```
[connectivity]
web-extra-http-headers=[("header-name", "header-value")]
```

要指定多个标头，请将它们添加到以逗号分隔的列表中。例如：

```
[connectivity]
web-extra-http-headers=[("header1-name", "header1-value"), ("header2-name", "header2-value")]
```

3. 保存并关闭 文件。
4. [停止 \(p. 41\)](#)，然后 [\(p. 40\)](#)重新启动NICE DCV 服务器。它。

配置 NICE DCV 身份验证

默认情况下，客户端在连接到 NICE DCV之前需要向托管 NICE DCV 会话. 的服务器进行身份验证。如果客户端未能进行身份验证，则将阻止它连接到会话。可禁用客户端身份验证要求以允许客户端在不针对服务器进行身份验证的情况下连接到会话。

NICE DCV 支持以下身份验证方法：

- system — 这是默认身份验证方法。客户端身份验证工作委派给基础操作系统。对于 Windows NICE DCV 服务器，身份验证工作委派给 WinLogon。对于 Linux NICE DCV 服务器，身份验证工作委派给 PAM。客户端在连接到时提供其系统凭证。NICE DCV 会话. 确保您的客户端拥有 上的相应用户账户的凭证。NICE DCV 服务器。

- none — 在连接到 NICE DCV 会话时不需要客户端身份验证。NICE DCV 服务器 将自动向所有尝试连接到会话的客户端授予访问权。

确保您的客户端了解 NICE DCV 服务器使用的身份验证方法，并且它们拥有连接到会话所需的信息。

主题

- 在 [Windows 上配置身份验证 \(p. 58\)](#)
- 在 [Linux 上配置身份验证 \(p. 59\)](#)

在 Windows 上配置身份验证

要更改 NICE DCV 服务器的身份验证方法，您必须使用 Windows 注册表编辑器配置 authentication 参数。

在 Windows 上更改身份验证方法

1. 打开 Windows 注册表编辑器。
2. 导航到 HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/security/ 注册表项，并选择 authentication 参数。

如果该注册表项中没有 authentication 参数，请创建一个：

- a. 在左侧窗格中，打开 authentication 注册表项的上下文（右键单击）菜单，然后依次选择新建和字符串值。
 - b. 对于名称，输入 authentication 并按 Enter。
3. 打开 authentication 参数。对于数值数据，输入 system 或 none。
 4. 选择 OK 并关闭 Windows 注册表编辑器。

Windows 凭证提供程序

Windows 凭证提供程序使用户能够绕过 Windows 登录（如果他们已成功针对 DCV 服务器进行身份验证）。

仅当 DCV authentication 参数设置为 system 时，才支持 Windows 凭证提供程序。如果 DCV authentication 参数设置为 none，则在针对 DCV 服务器自动进行身份验证后，用户必须手动登录到 Windows。

默认情况下，在安装时，将启用 Windows 凭证提供程序。NICE DCV 服务器。

禁用 Windows 凭证提供程序

1. 打开 Windows 注册表编辑器。
2. 导航到 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\Credential Providers\{8A2C93D0-D55F-4045-99D7-B27F5E263407} 密钥。
3. 依次选择编辑、新建、DWORD 值。
4. 对于名称，输入 Disabled。
5. 打开该值。对于值数据，输入 1 并选择确定。
6. 关闭 Windows 注册表编辑器。

重新启用 Windows 凭证提供程序

1. 打开 Windows 注册表编辑器。

2. 导航到 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\Credential Providers\{8A2C93D0-D55F-4045-99D7-B27F5E263407} 密钥。
3. 打开 Disabled 值。对于值数据，输入 0 并选择确定。
4. 关闭 Windows 注册表编辑器。

在 Linux 上配置身份验证

要更改 NICE DCV 服务器的身份验证方法，您必须配置 authentication 文件中的 dcv.conf 参数。

在 Linux 上更改身份验证方法

1. 导航到 /etc/dcv/，并使用您常用的文本编辑器打开 dcv.conf
2. 找到 authentication 部分中的 [security] 参数，并将现有值替换为 system 或 none。

如果 authentication 部分中没有 [security] 参数，则使用以下格式手动添加此参数：

```
[security]
authentication=method
```

3. 保存并关闭 文件。

配置 NICE DCV 授权

授权用于授予或拒绝 NICE DCV 客户端对特定 NICE DCV 功能的权限。在 NICE DCV 中，使用权限文件配置授权。权限文件定义了用户在连接到会话时有权访问的 NICE DCV 功能。

NICE DCV 支持两种类型的权限文件：

主题

- [默认权限文件 \(p. 59\)](#)
- [自定义权限文件 \(p. 59\)](#)
- [使用权限文件 \(p. 60\)](#)

默认权限文件

如果在创建会话时未指定自定义权限文件，则为所有会话使用默认权限文件。默认权限文件仅向会话拥有者授予对所有功能的完全访问权限。

可以自定义默认权限文件以包含自定义授权。默认权限文件位于 n c:\Program Files\NICE\DCV\Server\conf\default.perm (在 Windows NICE DCV 服务器上) 和 /etc/dcv/default.perm (在 Linux NICE DCV 服务器上)。

有关自定义默认权限文件的信息，请参阅[使用权限文件 \(p. 60\)](#)。

自定义权限文件

自定义权限文件可让您显式定义特定用户或组在连接到 时有权访问的功能。NICE DCV 会话. 在使用自定义权限文件时，将覆盖默认权限文件。

要使用自定义权限文件，您必须先创建该权限文件。接下来，在将 --permissions-file 选项与 dcv create-session 命令结合使用来启动会话时指定它。有关启动会话的更多信息，请参阅[启动 NICE DCV 会话 \(p. 66\)](#)。

有关创建自定义权限文件的信息，请参阅[使用权限文件 \(p. 60\)](#)。

使用权限文件

可以使用首选文本编辑器创建自定义权限文件或更新现有权限文件。权限文件通常采用以下格式：

```
#import file_to_import

[groups]
group_definitions

[aliases]
alias_definitions

[permissions]
user_permissions
```

以下部分说明如何在更新或创建权限文件时填充这些部分。

目录

- [导入权限文件 \(p. 60\)](#)
- [创建组 \(p. 60\)](#)
- [创建别名 \(p. 61\)](#)
- [添加权限 \(p. 62\)](#)

导入权限文件

`imports` 部分通常是权限文件的第一个部分。此部分可让您引用和包含现有权限文件。这使您能够将之前定义的 NICE DCV 权限并入您的权限文件中。

权限文件可包含多个导入。导入的权限文件可能导入其他权限文件。

将一个权限文件导入您的权限文件中

- 使用 `#import` 语句并使用绝对路径或相对路径指定文件的位置。
 - Windows NICE DCV 服务器:

```
#import ..\file_path\file
```

- Linux NICE DCV 服务器:

```
#import ../file_path/file
```

Example

以下语句使用绝对路径导入一个名为 `dcv-permissions.file` 的权限文件（此文件位于 Windows NICE DCV 上的 NICE DCV 服务器安装文件夹中）：

```
#import c:\Program Files\NICE\DCV\dcv-permissions.file
```

创建组

权限文件的 `[groups]` 部分可让您定义具有类似使用案例或权限要求的用户的用户组。可以为组分配特定的权限。分配给组的权限适用于组中包含的所有用户。

要在权限文件中创建组，您必须先将组部分标题添加到文件中。

```
[groups]
```

然后，您可以在部分标题的下方创建您的组。要创建组，请提供组名称，然后在逗号分隔的列表中指定组成员。组成员可以是单个用户、其他组和操作系统用户组。

```
group_name=member_1, member_2, member_3
```

如何为群组添加用户

指定用户名。

Note

您可以使用 `user:` 作为用户名的前缀。Windows 域用户名可以包含域名。

```
group_name=user_1, user:user_2, domain_name\user_3
```

将现有组添加到一个组

指定前缀为 `group:` 的组名 `group:`

```
group_name=group:group_1, group:group_2
```

将操作系统用户组添加到一个组中（仅限 Linux NICE DCV 服务器）

指定前缀为 `osgroup:` 的组名 `osgroup:`

```
group_name=osgroup:os_group_1, osgroup:os_group2
```

Example

以下示例添加组部分标题并创建一个名为 `my-group` 的组（该组包含名为 `john` 和 `jane` 的用户）、一个名为 `observers` 的现有组和一个名为 `guests:` 的操作系统用户组：

```
[groups]
my-group=john, user:jane, group:observers, osgroup:guests
```

创建别名

权限文件的 `[aliases]` 部分可让您创建 NICE DCV 功能集。在定义别名后，您可以授予或拒绝组或单个用户对该别名的使用权限。授予或拒绝对别名的权限将授予或拒绝对其中包含的所有功能的权限。

要在权限文件中创建别名，您必须先将别名部分标题添加到文件中。

```
[aliases]
```

然后，您可以在部分标题的下方创建您的别名。要创建别名，请提供别名名称，然后在逗号分隔的列表中指定别名成员。别名成员可以是单个 NICE DCV 功能或其他别名。

```
alias_name=member_1, member_2, member_3
```

Example

以下示例添加别名部分标题并创建一个名为 `file-management` 的别名 (包括 `file-upload` 和 `file-download` 供) 和一个名为 `clipboard-management` 的现有别名。

```
[aliases]
file-management=file-upload, file-download, clipboard-management
```

添加权限

权限文件的 `[permissions]` 部分可让您控制用户和组对特定功能或别名的访问权。

要将权限添加到权限文件, 请先将权限部分标题添加到文件中。

```
[permissions]
```

然后, 您可以在部分标题的下方添加您的权限。要添加一个权限, 请指定该权限控制的角色、要应用的规则 and 该权限应用于的功能。

```
actor rule features
```

角色可以是用户、组或操作系统组。组的前缀必须为 `group:`。操作系统组的前缀必须为 `osgroup:`。NICE DCV 包括一个内置 `%owner%` 引用, 该引用可用于引用会话所有者, 还包括一个内置 `%any%` 引用, 该引用可用于引用任何用户。

以下规则可用于权限语句:

- `allow` — 授予对此功能的访问权限。
- `disallow` — 拒绝对功能的访问, 但可由后续权限覆盖。
- `deny` — 拒绝对功能的访问, 且无法被后续权限覆盖。

功能可以包含单个 NICE DCV 功能和/或别名。功能列表必须用空格分隔。NICE DCV 包括一个内置 `builtin` 别名, 其中包括所有 NICE DCV 功能。

可在权限文件中引用以下功能:

- `display` — 从 NICE DCV 服务器接收视觉数据。
- `clipboard-copy` — 将数据从 NICE DCV 服务器复制到客户端剪贴板。
- `clipboard-paste` — 将数据从客户端剪贴板粘贴到 NICE DCV 服务器。
- `file-download` — 从会话存储下载文件。
- `file-upload` — 将文件上传到会话存储。
- `mouse` — 从客户端指针到 NICE DCV 服务器的输入。
- `keyboard` — 从客户端键盘到 NICE DCV 服务器的输入。
- `keyboard-sas` — 使用安全注意序列 (CTRL+Alt+Del)。需要 `keyboard` 功能。在 DCV 2017.3 版及更高版本上受支持。
- `touch` — 使用本机触摸事件。在 DCV 2017.3 版及更高版本上受支持。在 Linux NICE DCV 服务器上不受支持。
- `stylus` — 从专门的 USB 设备 (如 3D 指点设备或绘图板) 输入。
- `usb` — 从客户端使用 USB 设备。
- `pointer` — 查看 NICE DCV 服务器鼠标位置事件和指针形状。在 DCV 2017.3 版及更高版本上受支持。
- `audio-out` — 在客户端上播放 NICE DCV 服务器音频。
- `audio-in` — 将音频从客户端插入 NICE DCV 服务器。
- `printer` — 将 PDFs 或 XPS 文件从 NICE DCV 服务器打印到客户端。

- smartcard — 从客户端读取智能卡。
- builtin — 所有功能。

Example

以下示例添加权限部分标头和 4 项权限。第一项权限向名为 john 的用户授予对 display、file-upload 和 file-download 功能的访问权。第二项权限拒绝 observers 组对 audio-in 和 audio-out 功能以及别名为 clipboard-management 的功能的访问权。第三项权限向 guests 操作系统组授予对 clipboard-management 和 file-management 别名的访问权。第四项权限向会话所有者授予对所有功能的访问权。

```
[permissions]
john allow display file-upload file-download
group:observers deny audio-in audio-out clipboard-management
osgroup:guests allow clipboard-management file-management
%owner% allow builtin
```

管理 NICE DCV 会话

您必须在客户端可以连接到的 NICE DCV 会话上创建 NICE DCV 服务器。如果有活动会话，客户端只能连接到 NICE DCV 服务器。

每个 NICE DCV 会话都具有以下属性：

- ID — 用于唯一标识上的会话 NICE DCV 服务器。
- Owner — (所有者) 创建会话 NICE DCV 的用户。默认情况下，只有所有者可以连接到会话。

NICE DCV 客户端需要此信息来连接到会话。

主题

- [NICE DCV 会议简介 \(p. 64\)](#)
- [使用命令行工具管理 NICE DCV 会话 \(p. 65\)](#)
- [启动 NICE DCV 会话 \(p. 66\)](#)
- [停止 NICE DCV 会话 \(p. 69\)](#)
- [管理正在运行的 NICE DCV 会话 \(p. 69\)](#)
- [查看 NICE DCV 会话 \(p. 73\)](#)

NICE DCV 会议简介

NICE DCV 提供了两种类型的会话 — 控制台会话和虚拟会话。下表总结了两种会话类型之间的差异。

会话类型	支持	多个会话	所需权限	直接屏幕捕获	GPU 加速的 OpenGL 支持
控制台	Linux 和 Windows NICE DCV 服务器	否，每个服务器仅一个控制台会话	只有管理员用户可以启动和关闭会话	是	是，不带其他软件
虚拟	仅限 Linux NICE DCV 服务器	是，每个服务器多个虚拟会话	任何用户都可以启动和关闭会话	否，专用 X 服务器 (Xdcv) 为每个虚拟会话运行。从 X 服务器捕获屏幕。	是，但需要 DCV-GL 程序包

控制台会话

Windows 和 Linux NICE DCV 服务器上支持控制台会话。如果您使用的是 Windows NICE DCV 服务器，则只能使用控制台会话。

在 NICE DCV 服务器上，一次只能托管一个控制台会话。控制台会话由管理员（在 Windows NICE DCV 服务器上）和根用户（在 Linux NICE DCV 服务器上）创建和管理。

借助控制台会话，NICE DCV 直接捕获桌面屏幕的内容。如果为服务器配置了 GPU，则 NICE DCV 控制台会话可以直接访问 GPU。

Note

您不能同时在一个 NICE DCV 服务器上运行控制台和虚拟会话。

虚拟会话

仅 Linux NICE DCV 服务器上支持虚拟会话。

可以在同一 NICE DCV 服务器上同时托管多个虚拟会话。虚拟会话由 NICE DCV 用户创建和管理。NICE DCV 用户只能管理他们自己创建的会话。根用户可以管理当前在 NICE DCV 服务器上运行的所有虚拟会话。

使用虚拟会话，NICE DCV 启动 X 服务器实例 `xdcv`，并在 X 服务器内运行桌面环境。为每个虚拟会话 NICE DCV 启动一个新的专用 X 服务器实例。每个虚拟会话均使用其 X 服务器实例提供的显示内容。

如果 `dcv-g1` 包已安装并获得许可，则 NICE DCV 虚拟会话将共享对服务器 GPU 的访问权限。要在多个虚拟会话之间共享基于硬件的 OpenGL，必须通过配置 `dcv-g1.conf` 文件将虚拟 X 服务器实例连接到 GPU。

Note

您不能同时在一个 NICE DCV 服务器上运行控制台和虚拟会话。

使用命令行工具管理 NICE DCV 会话

NICE DCV 服务器包括一个命令行工具，可用于启动、停止和查看 NICE DCV 会话。

在 Windows NICE DCV 服务器上使用命令行工具

要在 Windows NICE DCV 服务器上使用命令行工具，您必须从 NICE DCV 安装目录运行命令，或者必须将 NICE DCV 目录添加到 PATH 环境变量中。如果将 NICE DCV 目录添加到 PATH 环境变量中，则可以从任何目录使用命令。

从 NICE DCV 安装目录使用命令行工具

导航到 `dcv.exe` 文件所在的文件夹（默认情况下为 `C:\Program Files\NICE\DCV\Server\bin\`），并打开命令提示窗口。

或者，您可以在从其他目录运行命令时指定完整路径。例如：

```
C:\> "C:\Program Files\NICE\DCV\Server\bin\dcv.exe" list-sessions
```

将 NICE DCV 目录添加到 PATH 环境变量中

1. 在文件资源管理器中，右键单击此 PC，然后选择属性。
2. 选择高级系统设置。
3. 在高级选项卡上，选择环境变量。
4. 在系统变量部分中，选择 Path 变量，然后选择编辑。
5. 选择新建并指定 `bin` 安装目录中的 NICE DCV 文件夹的完整路径（例如 `C:\Program Files\NICE\DCV\Server\bin\`）。
6. 选择确定并关闭“环境变量”窗口。

在 Linux NICE DCV 服务器上使用命令行工具

在 Linux NICE DCV 服务器上，在 `$PATH` 环境变量中自动配置命令行工具。这样，您可以从任何文件夹使用命令行工具。打开终端窗口并输入要运行的命令。

启动 NICE DCV 会话

默认情况下，在安装后，会在 Windows NICE DCV 服务器上自动创建一个控制台会话。默认控制台会话由 Administrator 拥有，其默认会话 ID 为 `console`。如果您在安装 NICE DCV 服务器时已选择阻止自动控制台会话，则必须手动创建一个。安装 NICE DCV 服务器后，您可以随时启用或禁用自动控制台会话。

Note

Linux NICE DCV 服务器在安装后不会获得默认控制台。

如果您正在本地或其他基于云的服务器上使用浮动许可证，并且您超出了许可证支持的最大并发会话数，您可能会收到 `no licenses` 错误。如果您收到此错误，请停止一个未使用的会话以释放许可证并重试。

NICE DCV 服务器 必须正在运行以启动会话。有关更多信息，请参阅 [启动 NICE DCV 服务器 \(p. 40\)](#)。

目录

- [手动启动控制台和虚拟会话 \(p. 66\)](#)
 - [Examples \(p. 66\)](#)
- [启用自动控制台会话 \(p. 68\)](#)

手动启动控制台和虚拟会话

您可以随时启动 NICE DCV 会话。一次只能运行一个控制台会话。如果您使用的是 Linux NICE DCV 服务器，您可以同时运行多个虚拟会话。

在 Windows 或 Linux NICE DCV 服务器上创建控制台或虚拟会话

使用 `dcv create-session` 命令并指定会话类型和唯一的会话 ID。

以下选项可以与 `dcv create-session` 命令一起使用：

`--type=console|virtual`

仅 Linux NICE DCV 服务器上支持此选项。它指定要创建的会话的类型，可以是 `console` 或 `virtual`。

`--name`

指定会话的名称。会话名称可以是最多包含 256 个字符的任何字符串。如果字符串超过 256 个字符，则命令将失败。会话名称不需要在整个正在运行的会话中是唯一的。

您可以随时使用 `dcv set-name` 命令更改会话的名称。有关更多信息，请参阅 [管理会话名称 \(p. 72\)](#)。

`--user`

仅在 Linux NICE DCV 会话上支持将此选项与虚拟会话一起使用。此值是要用于创建会话的用户。只有根用户可以模拟其他用户。

`--owner`

指定会话所有者。如果省略，则默认为当前已登录的用户。

--permissions-file

指定自定义权限文件的路径。如果省略，则默认为服务器默认值。

--storage-root

指定用于会话存储的文件夹的路径。

您可以使用 `%home%` 指定当前已登录的用户的主目录。例如，以下内容将会话存储的目录设置为 `c:\Users\username\storage\`（对于 Windows 服务器）或 `$HOME/storage/`（对于 Linux 服务器）。

```
--storage-root %home%/storage/
```

Note

如果指定的子目录不存在，则将禁用会话存储。

--gl

仅在 Linux NICE DCV 会话上支持将此选项与虚拟会话一起使用。它覆盖默认的 `dcv-gl` 状态，并且可以是 `on` 或 `off`。

--max-concurrent-clients

指定允许连接到会话的 NICE DCV 客户端的最大数量。如果省略，则默认为无限连接。

--init

仅在 Linux NICE DCV 服务器上支持将此选项与虚拟会话一起使用。它指定自定义 `init` 脚本的路径。该脚本可用于启动特定桌面环境以及在会话开始时自动启动特定应用程序。该脚本必须是可执行的。如果省略，则默认为启动默认桌面环境的脚本。

Examples

示例 1 – 控制台会话

以下命令创建一个由 `dcv-user` 拥有的 `console` 会话 `my-session`，该会话具有唯一的会话 ID 和会话名称 `my graphics session`。它还指定一个名为 `perm-file.txt` 的权限文件。

- Windows NICE DCV 服务器

```
C:\> dcv create-session --owner dcv-user --name "my graphics session" --permissions-file perm-file.txt my-session
```

- Linux NICE DCV 服务器

```
$ sudo dcv create-session --type=console --owner dcv-user --name "my graphics session" --permissions-file perm-file.txt my-session
```

示例 2 - 虚拟会话 (仅限 Linux NICE DCV 服务器)

以下命令将创建一个 `virtual` 会话，使用 `root` 用户模拟预期的会话所有者 `dcv-user`。会话由 `dcv-user` 拥有，即使它是由根用户创建。

```
$ sudo dcv create-session --owner dcv-user --user dcv-user my-session
```

示例 3 - 虚拟会话 (仅限 Linux NICE DCV 服务器)

以下命令创建一个 `virtual` 会话，该会话由创建它的用户拥有：

```
$ dcv create-session my-session
```

启用自动控制台会话

启用自动控制台会话可确保每次 NICE DCV 服务器 启动时都自动创建一个控制台会话。自动控制台会话由 NICE DCV 配置参数所指定的 `owner` 用户拥有。它的会话 ID 始终为 `console`。

影响自动控制台会话的其他参数是 `max-concurrent-clients`、`permissions-file` 和 `storage-root`。有关这些参数的更多信息，请参阅 [session-management/automatic-console-session 参数 \(p. 86\)](#)。

Note

NICE DCV 不支持自动创建虚拟会话。

在 Windows NICE DCV 服务器 上启用自动控制台会话

1. 打开 Windows 注册表编辑器。
2. 导航到 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/session-management` 注册表项。
3. 创建 `create-session` 参数：
 - a. 在左侧面板中，打开 `session-management` 项的上下文（右键单击）菜单，然后依次选择新建和 `DWORD (32 位)` 值。
 - b. 对于名称，输入 `create-session` 并按 `Enter`。
 - c. 打开 `create-session` 参数。对于值数据，输入 `1` 并选择确定。
4. 导航到 `HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/session-management/automatic-console-session` 注册表项。
5. 创建 `owner` 参数：
 - a. 在左侧面板中，打开 `automatic-console-session` 项的上下文（右键单击）菜单，然后依次选择新建和字符串值。
 - b. 对于名称，输入 `owner` 并按 `Enter`。
 - c. 打开 `owner` 参数。对于值数据，输入会话拥有者的名称并选择确定。
6. 选择确定，并关闭 Windows 注册表编辑器。
7. [停止 \(p. 41\)](#)，然后 [\(p. 40\)](#) 重新启动 NICE DCV 服务器它。

在 Linux NICE DCV 服务器 上启用自动控制台会话

1. 导航到 `/etc/dcv/`，并使用您常用的文本编辑器打开 `dcv.conf`。
2. 使用以下格式将 `create-session` 和 `owner` 参数添加到 `[session-management/automatic-console-session]` 部分：

```
[session-management]
create-session = true

[session-management/automatic-console-session]
owner="session-owner"
```

3. 保存并关闭文件。

4. [停止 \(p. 41\)](#) , 然后 [\(p. 40\)](#)重新启动NICE DCV 服务器它。

停止 NICE DCV 会话

控制台会话只能由管理员（在 Windows NICE DCV 服务器上）和根用户（在 Linux NICE DCV 服务器上）停止。Linux NICE DCV 服务器上的虚拟会话只能由根用户或创建它的 NICE DCV 用户停止。

Note

停止会话将关闭会话中正在运行的所有应用程序。

在 Windows 或 Linux NICE DCV 服务器上停止控制台或虚拟会话

使用 `dcv close-session` 命令并指定唯一的会话 ID：

```
dcv close-session session-id
```

例如，以下命令可停止具有唯一 ID `my-session` 的会话：

```
dcv close-session my-session
```

管理正在运行的 NICE DCV 会话

以下部分提供了有关如何管理正在运行的 NICE DCV 会话的信息。

主题

- [管理 NICE DCV 会话 存储 \(p. 69\)](#)
- [管理 NICE DCV 会话授权 \(p. 70\)](#)
- [管理 NICE DCV 会话显示布局 \(p. 71\)](#)
- [管理会话名称 \(p. 72\)](#)

管理 NICE DCV 会话 存储

会话存储是 NICE DCV 服务器上的一个目录，客户端在连接到 NICE DCV 会话时可访问该目录。

如果已在 NICE DCV 服务器上启用会话存储，则可以使用 `dcv set-storage-root` 命令指定要用于会话存储的目录。有关在 NICE DCV 服务器上启用会话存储的更多信息，请参阅[启用会话存储 \(p. 49\)](#)。

设置会话存储路径

使用 `dcv set-storage-root` 命令并指定会话 ID 和要使用的目录的路径。

```
dcv set-storage-root --session session_id path/to/directory
```

您可以使用 `%home%` 指定当前已登录的用户的主目录。例如，在 Windows 服务器上，`%home%/storage/` 路径解析为 `c:\Users\username\storage\`。在 Linux 服务器上，该路径解析为 `$HOME/storage/`。

Example

Windows NICE DCV 服务器示例

以下示例将会话 ID 为 `c:\session-storage` 的会话的存储路径设置为 `my-session`。

```
C:\> dcv set-storage-root --session my-session c:\session-storage
```

Linux NICE DCV 服务器示例

以下示例设置为当前用户的主目录中名为 `session-storage` 的目录的存储路径 (对于会话 ID 为 `my-session` 的会话)。

```
$ dcv set-storage-root --session my-session %home%/session-storage/
```

管理 NICE DCV 会话授权

授权用于授予或拒绝 NICE DCV 客户端对特定 NICE DCV 功能的权限。通常，在启动 NICE DCV 会话时配置授权。但是，可以编辑正在运行的会话的权限。有关 NICE DCV 授权的更多信息，请参阅[配置 NICE DCV 授权 \(p. 59\)](#)。

修改正在运行的会话的权限

使用 `dcv set-permissions` 命令。以下选项可以与 `dcv set-permissions` 命令一起使用。

`--session`

指定要为其设置权限的会话的 ID。

`--reset-builtin`

将会话的权限重置为默认会话权限。默认权限仅向会话拥有者授予对所有功能的完全访问权限。

`--none`

撤消对会话的所有权限。

`--file`

指定自定义权限文件的路径。如果指定的文件为空，则撤消所有权限。有关创建自定义权限文件的更多信息，请参阅[使用权限文件 \(p. 60\)](#)。

Examples

示例 1—撤消所有权限

以下示例撤消 ID 为 `my-session` 的会话的所有客户端权限。

```
C:\> dcv set-permissions --session my-session --none
```

示例 2—指定自定义权限

以下示例为 ID 为 `perm-file.txt` 的会话指定一个名为 `c:\dvc\` 的自定义权限文件，该文件位于 `my-session` 目录中。

```
C:\> dcv set-permissions --session my-session --file c:\dvc\perm-file.txt
```

示例 3—重置权限

以下示例将 ID 为 `my-session` 的会话的权限重置为默认值。

```
C:\> dcv set-permissions --session my-session --reset-builtin
```

管理 NICE DCV 会话显示布局

您可以设置正在运行的 NICE DCV 会话的显示布局。显示布局指定在客户端连接到会话时使用的默认配置。但是，客户端可以使用 NICE DCV 客户端设置或本机操作系统显示设置来手动覆盖布局。

如果主机服务器的硬件和软件配置不支持指定的分辨率或屏幕数，则 NICE DCV 服务器不会应用指定的显示布局。

主题

- [限制显示布局 \(p. 71\)](#)
- [指定显示布局 \(p. 71\)](#)
- [查看显示布局 \(p. 72\)](#)

限制显示布局

您可以配置 NICE DCV 服务器以防止客户端请求超出指定范围的显示布局。要限制显示布局更改，请配置以下 NICE DCV 服务器参数。

- [enable-client-resize \(p. 92\)](#)—要防止客户端更改显示布局，请将此参数设置为 `false`。
- [min-head-resolution \(p. 93\)](#) 和 [max-head-resolution \(p. 93\)](#)—分别指定允许的最小分辨率和最大分辨率。
- [web-client-max-head-resolution \(p. 93\)](#)—指定 Web 浏览器客户端允许的最大分辨率。`max-head-resolution` 限制基于 `web-client-max-head-resolution` 限制应用。默认情况下，Web 浏览器客户端的最大分辨率为 1920x1080。指定更高的分辨率可能会导致性能问题，具体取决于 Web 浏览器和客户端计算机规格。
- [max-num-heads \(p. 94\)](#)—指定最大显示数。
- [max-layout-area \(p. 93\)](#)—指定屏幕区域允许的最大像素数。屏幕区域总大小（以像素为单位表示）超过指定值的请求将被忽略。

有关这些参数的更多信息，请参阅“参数参考”中的[display 参数 \(p. 92\)](#)。

指定显示布局

为正在运行的 NICE DCV 会话配置显示布局

使用 `dcv set-display-layout` 命令并指定要为其设置显示布局和显示布局描述符的会话。

```
dcv set-display-layout --session session-id display-layout-descriptor
```

显示布局描述符指定显示数以及每个显示的分辨率和位置偏移。必须按以下格式指定描述：

```
widthxheight+|-x-position-offset+|-y-position-offset
```

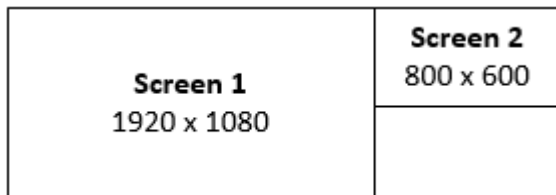
如果指定多个屏幕，请用逗号将屏幕描述符分隔开来。屏幕位置偏移指定屏幕左上角相对于屏幕 1 的位置。如果没有为屏幕指定位置偏移，则它默认为 `x=0` 和 `y=0`。

Important

如果要指定多个屏幕，请确保正确设置每个屏幕的位置偏移，以避免屏幕重叠。

例如，以下显示布局描述符指定两个屏幕：

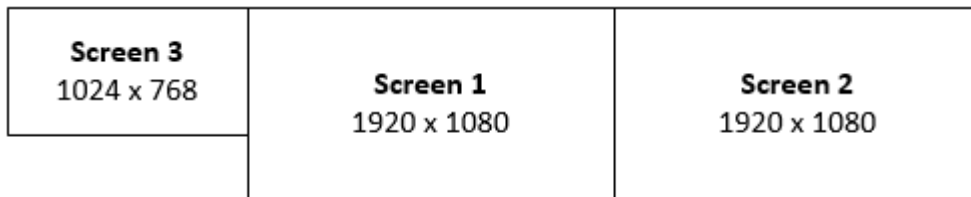
- 屏幕 1：1920x1080 分辨率，偏移为 $x=0, y=0$
- 屏幕 2：800x600 分辨率，偏移为 $x=1920, y=0$ ，以便它显示在屏幕 1 的右侧。



```
1920x1080+0+0,800x600+1920+0
```

以下显示布局描述符指定三个屏幕。

- 屏幕 1：1920x1080 分辨率，偏移为 $x=0, y=0$
- 屏幕 2：1920x1080 分辨率，偏移为 $x=1920, y=0$ ，以便它显示在屏幕 1 的右侧。
- 屏幕 3：1024x768 分辨率，偏移为 $x=-1024, y=0$ ，以便它显示在屏幕 1 的左侧。



```
1920x1080+0+0,1920x1080+1920+0,1024x768-1024+0
```

查看显示布局

查看会话的显示布局

使用 `dcv describe-session` 命令并查看输出中的 `display layout` 元素。有关更多信息，请参阅[查看 NICE DCV 会话 \(p. 73\)](#)。

管理会话名称

您可以随时更改正在运行的会话的名称。通过会话名称，您可以根据分配给特定会话的名称快速识别该会话。会话名称不需要在整个正在运行的会话中是唯一的。

更改正在运行的会话的名称

使用 `dcv set-name` 命令。对于 `session`，指定要更改其名称的会话的 ID，对于 `name`，指定会话的名称。会话名称可以是最多包含 256 个字符的任何字符串。如果指定的字符串超过 256 个字符，则命令将失败。

```
$ dcv set-name --session session-id --name "session-name"
```

从正在运行的会话中删除名称

使用 `dcv set-name` 命令。对于 `session`，指定要从中删除名称的会话的 ID，并包含 `--none` 参数。

```
$ dcv set-name --session session-id --none
```

Examples

示例 —1更改会话的名称

以下示例将 ID 为 `my-session` 的会话的名称设置为 `my graphics session`。

```
$ dcv set-name --session my-session --name "my graphics sessions"
```

示例 —2删除会话的名称

以下示例删除 ID 为 `my-session` 的会话的名称。

```
$ dcv set-name --session my-session --none
```

查看 NICE DCV 会话

Windows NICE DCV 服务器上的管理员或 Linux NICE DCV 服务器上的根用户可以查看服务器运行上的所有活动会话。NICE DCV 用户只能查看他们创建的会话。

在 Windows 或 Linux NICE DCV 服务器上列出活动的控制台会话或虚拟会话

使用 `dcv list-sessions` 命令：

```
dcv list-sessions
```

该命令使用以下格式返回活动会话的列表：

```
Session: session-id (owner:session-owner type:virtual/console name:'my session')
```

查看有关会话的信息

使用 `dcv describe-session` 命令并指定唯一的会话 ID。

```
$ dcv describe-session session-id
```

在以下示例输出中，`display-layout` 元素指示会话的显示布局已设置为使用两个 800x600 屏幕，第二个屏幕相对于第一个屏幕的偏移为 `x=800`（向右）。

```
Session: test
  owner: session-id
  name: session-name
  x display: :1
  x authority: /run/user/1009/dcv/test.xauth
  display layout: 800x600+0+0,800x600+800+0
```

您还可以包含 `--json`（或 `-j`）选项，以强制命令以 JSON 格式返回输出。JSON 输出提供了有关会话的其他详细信息。

```
$ dcv describe-session session-id --json
```

以下是示例 JSON 输出。

```
{
  "id" : "session-id",
  "owner" : "dcvuser",
  "name" : "session-name",
  "num-of-connections" : 0,
  "creation-time" : "2020-03-02T16:08:50Z",
  "last-disconnection-time" : "",
  "licenses" : [
    {
      "product" : "dcv",
      "status" : "licensed",
      "check-timestamp" : "2020-03-02T16:08:50Z",
      "expiration-date" : "2020-03-29T00:00:00Z"
    },
    {
      "product" : "dcv-gl",
      "status" : "licensed",
      "check-timestamp" : "2020-03-02T16:08:50Z",
      "expiration-date" : "2020-03-29T00:00:00Z"
    }
  ],
  "storage-root" : "",
  "type" : "virtual",
  "x11-display" : ":2",
  "x11-authority" : "/run/user/1009/dcv/vsession.xauth",
  "display-layout" : [
    {
      "width" : 800,
      "height" : 600,
      "x" : 0,
      "y" : 0
    },
    {
      "width" : 800,
      "height" : 600,
      "x" : 800,
      "y" : 0
    }
  ]
}
```

如何...

主题

- [使用外部身份验证 \(p. 75\)](#)
- [查找并停止空闲会话 \(p. 77\)](#)
- [启用到 X 服务器的远程 X 连接 \(p. 78\)](#)
- [将 NICE DCV Web 浏览器客户端嵌入 iFrame 中 \(p. 79\)](#)

使用外部身份验证

默认情况下，NICE DCV 客户端身份验证工作委派给底层操作系统。对于 Windows NICE DCV 服务器，身份验证工作委派给 WinLogon。对于 Linux NICE DCV 服务器，身份验证工作委派给 Linux PAM。

您可以将 NICE DCV 配置为使用外部身份验证服务器对客户端进行身份验证。这使您能够使用现有的身份验证系统。借助外部身份验证，NICE DCV 利用您的现有登录机制并将身份验证委派给外部身份验证服务器。

要使用外部身份验证服务器，您必须做好以下准备：

- **登录机制** —这是您的用户用于登录的前端机制。它应该能够使用现有凭证验证系统验证您的用户，并且应该能够生成令牌并将其提供给 NICE DCV 服务器。有关更多信息，请参阅[使用令牌 \(p. 76\)](#)。
- **身份验证服务器** —这是对登录机制生成的令牌进行身份验证的服务器。此服务器应该能够从包含令牌 NICE DCV 服务器的接收 HTTP (S) POST 请求，执行必要的身份验证，然后将响应发送回 NICE DCV 服务器。有关实施身份验证服务器的更多信息，请参阅[身份验证服务要求 \(p. 76\)](#)。
- **NICE DCV 服务器配置**—NICE DCV 服务器必须配置为使用外部身份验证服务器。有关更多信息，请参阅[NICE DCV 服务器配置 \(p. 75\)](#)。

主题

- [NICE DCV 服务器配置 \(p. 75\)](#)
- [使用令牌 \(p. 76\)](#)
- [身份验证服务要求 \(p. 76\)](#)

NICE DCV 服务器配置

您必须将配置为 NICE DCV 服务器使用外部身份验证服务。

Linux NICE DCV 服务器

在 Linux 上指定外部身份验证服务器

1. 导航到 `/etc/dcv/`，并使用您常用的文本编辑器打开 `dcv.conf`。
2. 找到 `auth-token-verifier` 部分中的 `[security]` 参数，并以以下格式将现有值替换为外部身份验证服务器的 URL 和要与之进行通信的端口 `url:port`。例如，如果您使用的是 `DcvSimpleExternalAuthenticator` 请指定以下内容 `http://127.0.0.1:8444`。

如果 `auth-token-verifier` 部分中没有 `[security]` 参数，则使用以下格式手动添加此参数：

```
[security]
auth-token-verifier=url:port
```


3. 保存并关闭文件。

Windows NICE DCV 服务器

在 Windows 上指定外部身份验证服务器

1. 打开 Windows 注册表编辑器并导航到 HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/security/ 注册表项。
2. 打开 auth-token-verifier 参数。对于值数据，采用以下格式输入外部身份验证服务器的 URL 和要通信的端口 `url:port` :。例如，如果您使用的是 DcvSimpleExternalAuthenticator 请指定以下内容 `http://127.0.0.1:8444` :。

Note

如果该参数不存在，请创建一个新的 String 参数并将其命名为 auth-token-verifier。

3. 关闭 Windows 注册表编辑器。
4. 停止 (p. 41) ，然后 (p. 40) 重新启动 NICE DCV 服务器它。

使用令牌

生成令牌后，您必须能够将其发送到 NICE DCV 服务器。使用 Web 浏览器客户端，将令牌附加到连接 URL，如下所示：

```
https://server_hostname_or_IP:port?authToken=token#session_id
```

例如：

```
https://my-dcv-server.com:8443/?authToken=1234567890abcdef#my-session
```

身份验证服务要求

您的自定义身份验证服务可以在同一主机上运行 NICE DCV 服务器，也可以运行在单独的主机上。身份验证服务必须侦听来自的 HTTP (S) POST 请求 NICE DCV 服务器。

下面显示了使用的 POST 请求格式 NICE DCV 服务器。

```
POST / HTTP/1.1  
Content-Type: application/x-www-form-urlencoded  
sessionId=session_id&authenticationToken=token&clientAddress=client_address
```

您的身份验证服务负责确定提供的令牌是否有效。

验证令牌后，身份验证服务器必须将响应返回到 NICE DCV 服务器。响应正文必须包含以下内容之一，具体取决于身份验证过程的结果：

- 如果身份验证成功，身份验证服务将返回的结果 yes 和用户标识符。例如：

```
<auth result="yes"><username>username</username></auth>
```

- 如果身份验证不成功，身份验证服务将返回结果 no。例如：

```
<auth result="no"/>
```

DcvSimpleExternalAuthenticator

NICE DCV 附带了一个名为 `dcvsimpleextauth` 的引用外部身份验证服务器 `DcvSimpleExternalAuthenticator`。 `DcvSimpleExternalAuthenticator` 是一个 Python 脚本，您可以使用该脚本创建自己的自定义身份验证服务器的起点。

`DcvSimpleExternalAuthenticator` 服务器支持 HTTP 和 HTTPS，并且必须在 NICE DCV 服务器安装了 `dcvsimpleextauth` 的同一服务器上运行。默认情况下 `DcvSimpleExternalAuthenticator` 侦听端口上的请求 8444。如果需要，您可以更改端口。为此 `/etc/dcv/simpleextauth.conf`，请使用首选文本编辑器打开，找到 `EXTAUTH_PORT` 参数，并将现有值替换为所需的端口号。

要使用 `DcvSimpleExternalAuthenticator` 您必须安装 `nice-dcv-simple-external-authenticator` 程序包。有关更多信息，请参阅 [安装 NICE DCV 服务器 \(p. 17\)](#)。

要在安装 `DcvSimpleExternalAuthenticator` 后启动它，请运行以下命令：

```
$ dcvsimpleextauth start-server
```

查找并停止空闲会话

您可以使用带 NICE DCV 会话 命令选项的 `dcv describe-sessions` CLI 命令标识空闲-j。指定 `-j` 选项会将此命令配置为返回 JSON 格式的输出，其中包含有关会话的其他详细信息。

例如，以下命令返回有关名为 `my-session` 的会话的信息。

```
$ dcv describe-session my-session -j
```

输出：

```
{
  "id" : "my-session",
  "owner" : "dcvuser",
  "x11-display" : ":1",
  "x11-authority" : "/run/user/1009/dcv/test3.xauth",
  "num-of-connections" : 1,
  "creation-time" : "2019-05-13T13:21:19.262883Z",
  "last-disconnection-time" : "2019-05-14T12:32:14.357567Z",
  "licensing-mode" : "DEMO",
  "licenses" : [
    {
      "product" : "dcv",
      "status" : "LICENSED",
      "check-timestamp" : "2019-05-14T12:35:40Z",
      "expiration-date" : "2019-05-29T00:00:00Z"
    },
    {
      "product" : "dcv-gl",
      "status" : "LICENSED",
      "check-timestamp" : "2019-05-14T12:35:40Z",
      "expiration-date" : "2019-05-29T00:00:00Z"
    }
  ]
}
```

在命令输出中，`num-of-connections` 参数表示活动客户端连接的数量。值 0 表示没有活动的客户端连接，并且会话当前处于空闲状态。您还可以使用 `last-disconnection-time` 参数确定会话上次具有活动客户端连接的时间。

您可以创建脚本或 cron 作业来使用此信息标识空闲会话。然后，您可以使用 `dcv stop-session` (p. 69) 命令停止使用这些会话。

Note

停止会话将关闭会话中正在运行的所有应用程序。

启用到 X 服务器的远程 X 连接

默认情况下，由于存在内在安全风险，NICE DCV 2017 会阻止使用 X 转发。NICE DCV 从较新的 Xorg 服务器版本继承了这一行为。NICE DCV 服务器实施了以下默认继承的缓解措施来最大限度降低安全风险：

- NICE DCV 服务器会阻止来自网络的 X 连接。NICE DCV 服务器配置为通过 `-nolisten tcp` 命令行选项来启动。但是，可以更改默认行为，以启用到 X 服务器的远程 X 连接。有关变通方法的更多信息，请参阅 [启用到 X 服务器的远程 X 连接](#) (p. 78)。
- X 服务器会禁用 GLX 间接上下文。由于与 DCV-GL 存在冲突，因此，目前没有变通办法来启用 GLX 间接上下文。

有关安全风险以及缓解措施的更多信息，请参阅 [X.Org 安全公告](#)。

启用到 X 服务器的远程 X 连接

默认情况下，NICE DCV 配置为通过 `-nolisten tcp` 命令行选项来启动以降低安全风险。但是，可以更改默认行为，以启用 X 转发。

启用 X 转发

使用所需的文本编辑器打开 `/etc/dcv/dcv.conf`。将以下内容添加到该文件的末尾：

- 通过 IPv4 和 IPv6 启用 X 转发

```
[session-management]
virtual-session-xdcv-args="-listen tcp"
```

- 仅通过 IPv4 启用 X 转发

```
[session-management]
virtual-session-xdcv-args="-listen tcp -nolisten inet6"
```

Note

启用 X 转发不会影响现有会话，而只会影响在启用它后启动的新会话。

测试 X 转发

1. 连接 NICE DCV 会话。
2. 确认 NICE DCV 服务器正在 6000-6063 范围内的一个端口上侦听。

```
$ netstat -punta | grep 600
```

3. 检索 NICE DCV 会话显示号码。

```
$ dcv describe-session session_name | grep display
```

4. 通过 SSH 连接到托管该应用程序的远程服务器。

```
$ ssh user@remote_server_ip
```

5. 从该远程服务器上，导出 display 环境变量，以指向 NICE DCV 会话的 X 服务器。

```
$ export DISPLAY=dcv_server_ip:display_number
```

6. 从该远程服务器上，运行一个应用程序来测试 X 转发功能。例如：

```
xterm
```

测试应用程序（这里为 xterm）应出现在 NICE DCV 服务器的桌面环境中。

将 NICE DCV Web 浏览器客户端嵌入 iFrame 中

默认情况下，为了防止点击劫持攻击，NICE DCV 不允许 Web 浏览器客户端嵌入在 iFrame 中。但是，您可以覆盖此默认行为以允许 Web 浏览器客户端在 iFrame 内运行。

有关防止点击劫持攻击的更多信息，请参阅[内容安全策略备忘单](#)。

要允许 Web 浏览器在 iFrame 内运行，您必须配置 NICE DCV 服务器以将以下其他 HTTP 响应标头发送到 Web 浏览器客户端：

- web-x-frame-options
- web-extra-http-headers

我们建议您添加这两个标头，以确保跨 Web 浏览器的最佳兼容性。

Windows server

1. 打开 Windows 注册表编辑器并导航到 HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/connectivity/ 注册表项。
2. 打开 web-x-frame-options 参数。对于值数据，输入 "ALLOW-FROM https://*server_hostname*"。

Note

如果该参数不存在，请创建一个新的 String 参数并将其命名为 web-x-frame-options。

3. 打开 web-extra-http-headers 参数。对于值数据，输入 [("Content-Security-Policy", "frame-ancestors https://*server_hostname*")]。

Note

如果该参数不存在，请创建一个新的 String 参数并将其命名为 web-extra-http-headers。

4. 关闭 Windows 注册表编辑器。
5. [停止 \(p. 41\)](#)，然后 [\(p. 40\)](#) 重新启动 NICE DCV 服务器它。

Linux server

1. 使用首选文本编辑器打开 /etc/dcv/dcv.conf。
2. 请在 [connectivity] 部分执行以下操作：

- 对于 web-x-frame-options , 输入 "ALLOW-FROM https://*server_hostname*"。
- 对于 web-extra-http-headers , 输入 [("Content-Security-Policy", "frame-ancestors https://*server_hostname*")]

例如 :

```
[connectivity]
web-x-frame-options="ALLOW-FROM https://my-dcv-server.com"
web-extra-http-headers=[("Content-Security-Policy", "frame-ancestors https://my-dcv-server.com")]
```

3. 保存并关闭文件。
4. **停止 (p. 41) , 然后 (p. 40)**重新启动NICE DCV 服务器它。

默认情况下, 大多数浏览器阻止访问某些功能, 如麦克风访问和全屏访问。要允许访问这些功能, 请修改网页上的 iFrame 元素。例如, 要允许访问麦克风和全屏模式, 请修改 iFrame 元素, 如下所示 :

```
<iframe src="..." allow="microphone; fullscreen">/iframe>
```

NICE DCV 问题排查

本章说明如何标识和解决您可能遇到的与 NICE DCV 相关的问题。

主题

- [使用日志文件 \(p. 81\)](#)
- [常见问题 \(p. 82\)](#)

如果您需要其他帮助，则可在 [NICE 支持门户](#) 上记录支持服务单。

使用日志文件

NICE DCV 日志文件可用于识别和解决与 NICE DCV 服务器相关的问题。可以在您的 NICE DCV 服务器上的以下位置找到 NICE DCV 日志文件：

- Windows 服务器

```
C:\ProgramData\NICE\dcv\log\server.log
```

Note

默认情况下，ProgramData 文件夹可能会隐藏。如果未看到 ProgramData 文件夹，请将文件浏览器设置为显示隐藏项目。或者，在地址栏中输入 %programdata%，然后按 Enter 键。

- Linux 服务器

```
/var/log/dcv/server.log
```

NICE DCV 服务器 使您能够配置详细程度等级的日志文件。提供了以下详细程度等级：

- error – 提供最少的细节。仅包括错误。
- warning – 包括错误和警告。
- info – 默认详细程度级别。包括错误、警告和信息消息。
- debug – 提供最多的细节。提供有助于调试问题的详细信息。

在 Windows 上更改日志文件详细程度

要配置日志文件详细程度，您必须使用 Windows 注册表编辑器配置 level 参数。

在 Windows 上更改日志文件详细程度

1. 打开 Windows 注册表编辑器。
2. 导航到 HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/log/ 注册表项。
3. 通过双击打开 level 参数。对于数值数据，键入 error、warning、info 或 debug，具体取决于所需的详细程度等级。
4. 选择确定，并关闭 Windows 注册表编辑器。

在 Linux 上更改日志文件详细程度

要配置日志文件详细程度，您必须配置 level 文件中的 dcv.conf 参数。

在 Linux 上更改日志文件详细程度

1. 导航到 /etc/dcv/，并使用您常用的文本编辑器打开 dcv.conf。
2. 找到 level 部分中的 [log] 参数，并将现有详细程度等级替换为 error、warning、info 或 debug。

```
[log]
level="verbosity_level"
```

3. 保存并关闭 文件。

常见问题

本主题列出一些常见问题。

主题

- [Windows NICE DCV 服务器 上的光标问题 \(p. 82\)](#)

Windows NICE DCV 服务器 上的光标问题

对于在 Windows Server 2012 或 Windows 8 及更高版本上运行的 NICE DCV 服务器，鼠标光标始终显示为箭头。甚至在文本输入字段或单击导航项上暂停时，也会出现此情况。如果没有连接到服务器的物理鼠标，或者设备管理器中未列出鼠标设备，则可能会发生这种情况。

解决问题

1. 打开“控制面板”，然后选择轻松访问中心。
2. 选择使鼠标更易于使用。
3. 选择打开鼠标键。
4. 选择应用、确定。

NICE DCV 服务器 参数参考

下表列出了可配置用于自定义 NICE DCV 服务器的参数。

Note

每个表中的重新加载上下文列指示重新加载参数的时间。可能的上下文包括：

- `server`—在服务器启动时将加载一次该参数。如果更新参数值，则在重新启动服务器时将加载新值。
- `session`—在创建会话时将加载该参数。如果更新参数值，则将为后续会话加载新值。
- `connection`—在建立新的客户端连接时将加载该参数。如果更新参数值，则新值将用于后续客户端连接。
- `custom`—参数的加载条件对于此参数是唯一的。有关更多信息，请参阅参数说明。

主题

- [connectivity 参数 \(p. 83\)](#)
- [session-management 参数 \(p. 85\)](#)
- [session-management/defaults 参数 \(p. 86\)](#)
- [session-management/automatic-console-session 参数 \(p. 86\)](#)
- [security 参数 \(p. 87\)](#)
- [license 参数 \(p. 91\)](#)
- [input 参数 \(p. 91\)](#)
- [display 参数 \(p. 92\)](#)
- [display/linux 参数 \(p. 94\)](#)
- [audio 参数 \(p. 95\)](#)
- [log 参数 \(p. 95\)](#)
- [windows 参数 \(p. 96\)](#)
- [clipboard 参数 \(p. 97\)](#)
- [smartcard 参数 \(p. 98\)](#)
- [修改配置参数 \(p. 98\)](#)

connectivity 参数

下表描述了 [connectivity] 文件的 `/etc/dcv/dcv.conf` 部分中（对于 Linux NICE DCV 服务器）和 `connectivity` 注册表项中（对于 Windows NICE DCV 服务器）的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
<code>web-port</code>	整数 - 32-bit)	<code>server</code>	8443	客户端的 TCP 端口 — 指定 DCV 服务器侦听客户端连接的 TCP 端口。端口号必须介于 1024 和 65535 之间。—从版本 0-4100 开始 (p. 112) 可用。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
web-url-path	字符串	server	'/'	嵌入式 Web 服务器的 URL 路径 — 指定嵌入式 Web 服务器的 URL 路径，必须以 "/" 开头。例如，将其设置为 /test/foo 意味着 Web 服务器可通过 https://host:port/test/foo 访问。—自版本 0-4100 起 (p. 112) 可用。
web-root	字符串	server	"	嵌入式 Web 服务器的文档根目录 — 指定嵌入式 Web 服务器的文档根目录。—从版本 0-4100 开始 (p. 112) 可用。
web-use-hsts	true 或 false - 32-bit)	server	Linux : true - Windows : 1	是否使用 HSTS — 启用此项可强制浏览器阻止将通过 HTTP 发送的任何通信。改为使用 HTTPS 完成到网页（以及所有子域）的所有传输。—从版本 0-4100 开始 (p. 112) 可用。
ws-keepalive-interval	整数 - 32-bit)	server	10	Websocket keepalive 间隔 — 指定发送 keepalive 消息的时间间隔（秒）。如果设置为 0，则禁用 keepalive 消息。—自版本 0-4100 起 (p. 112) 可用。
idle-timeout	整数 - 32-bit)	自定义	60	空闲超时 — 指定在断开空闲客户端之前等待的分钟数。指定 0 表示从不断开空闲客户端。此参数值每 5 秒读取一次。—自版本 0-4100 起 (p. 112) 可用。
idle-timeout-warning	整数 - 32-bit)	自定义	350	空闲超时警告 — 指定在向空闲客户端发出空闲超时断开连接警告之前等待的相对于空闲超时的秒数。指定 0 表示从不警告空闲客户端。—从版本 4-6898 开始 (p. 108) 可用。
enable-quic-frontend	true 或 false - 32-bit)	server	Linux : false - Windows : 0	是否启用 QUIC 前端 — 指定是否应启用 QUIC 前端。—发布自版本 2-9508。
quic-port	整数 - 32-bit)	server	8443	QUIC 前端的 UDP 端口 — 指定 DCV 服务器侦听 UDP 连接的 UDP 端口。端口号必须介于 1024 到 65535 之间。—从版本 2-9508 开始可用。

session-management 参数

下表描述了 [session-management] 文件的 /etc/dcv/dcv.conf 部分中 (对于 Linux NICE DCV 服务器) 和 session-management 注册表中 (对于 Windows NICE DCV 服务器) 的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
max-concurrent-sessions	整数 - 32-bit)	server	0	最大并发会话数 — 指定允许的最大并发会话数。此限制当前仅适用于虚拟会话，因为控制台会话本质限于一个。指定 0 不强制实施任何限制。—自版本 0-7318 起 (p. 107) 可用。
create-session	true 或 false - 32-bit)	server	Linux : false - Windows : 0	在服务器启动时创建控制台会话 — 指定是否在服务器启动时自动创建控制台会话 (使用 ID“控制台”)。—自版本 0-4100 起 (p. 112) 可用。
max-concurrent-clients	整数 - 32-bit)	session	-1	每个会话最多并发客户端数量 — 指定每个会话并发客户端的最多数量。如果设置为 -1，则表示无限制。要仅为自动会话设置限制，请使用“session-management/automatic-console-session”部分的“max-concurrent-clients”。—从版本 0-4100 开始 (p. 112) 可用。
enable-gl-in-virtual-sessions	字符串	session	“default-on”	是否使用 dcv-gl 功能 — 指定是否使用 dcv-gl 功能 (需要许可证)。允许的值：“always-on”、“always-off”、“default-on”、“default-off”。—从版本 0-4100 开始 (p. 112) 可用。
virtual-session-font-path	字符串	session	“”	是否添加特殊字体路径 — 指定特殊字体的路径。某些应用程序需要将特殊字体传递到 X 服务器。—自版本 0-4100 起 (p. 112) 可用。
virtual-session-default-layout	字符串	session	[]	虚拟会话的默认布局 — 如果设置此项，则将 Xdcv 配置为在启动时创建指定的布局。可以为每个监视器配置分辨率 (w,h) 和位置 (x,y)。启用所有指定的监视器。默认布局示例值：[{‘w’:<800>, ‘h’:<600>, ‘x’:<0>, ‘y’: <0>}, {‘w’:<1024>, ‘h’:<768>, ‘x’:<800>, ‘y’:<0>}] 对于此设置，最大监视器数 (在 virtual-session-monitors 设置中指定) 的优先级高于数

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
				组中的元素数。例如，如果设置了五台监视器，而最大监视器数为四，则只创建前四台监视器。如果设置此键，则忽略已启用的监视器数（在 virtual-session-monitors 设置中指定）。—自版本 0-5600 起 (p. 111) 可用。
virtual-session-xdcv-args	字符串	session	"	要传递给 Xdcv 的其他参数 — 指定要传递给 Xdcv 的任何其他参数。—自版本 0-4334 起 (p. 112) 可用。

session-management/defaults 参数

下表描述了 [session-management/defaults] 文件的 /etc/dcv/dcv.conf 部分中（对于 Linux NICE DCV 服务器）和 session-management/defaults 注册表中（对于 Windows NICE DCV 服务器）的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
permissions-file	字符串	session	"	所有会话中包括的默认权限 — 指定要与用户为每个会话选择的权限自动合并的权限文件的路径。如果为空，请使用“default.perm”文件，该文件位于 /etc/dcv/（对于 Linux）或 DCV 安装文件夹（例如，“C : \Program Files\NICE \DCV\Server\conf”）（对于 Windows）。—从版本 0-5600 开始 (p. 111) 可用。

session-management/automatic-console-session 参数

下表描述了 [session-management/automatic-console-session] 文件的 /etc/dcv/dcv.conf 部分中（对于 Linux NICE DCV 服务器）和 session-management/automatic-console-session 注册表中（对于 Windows NICE DCV 服务器）的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
max-concurrent-clients	整数 - 32-bit)	server	-1	每个会话最多并发客户端数量 — 指定每个会话所允许的并发客户端的最多数量。如果设置为 -1，则不施加任何限制。 — 自版本 0-5600 起 (p. 111) 可用。
permissions-file	字符串	server	"	自动“控制台”会话的权限文件 — 指定用于检查用户的 DCV 功能访问权限的权限文件的路径。如果为空，则只有所有者才具有会话的完全访问权限。 — 自版本 0-5600 起 (p. 111) 可用。
owner	字符串	server	"	自动创建的“控制台”会话的拥有者 — 指定“控制台”会话拥有者的用户名。如果为空，则所有者为启动了 DCV 服务器的用户。此设置仅应用于在 create-session 设置设置为 true 时在服务器启动时自动创建的“控制台”会话。 — 自版本 0-5600 起 (p. 111) 可用。
storage-root	字符串	server	"	文件存储根文件夹的路径 — 指定用于控制台会话存储的文件夹的完整路径。如果 storage-root 为空或文件夹不存在，则禁用文件存储。 — 自版本 0-5600 起 (p. 111) 可用。

security 参数

下表描述了 [security] 文件的 /etc/dcv/dcv.conf 部分中 (对于 Linux NICE DCV 服务器) 和 security 注册表项中 (对于 Windows NICE DCV 服务器) 的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
身份验证	字符串	server	"system"	身份验证方法 — 指定 DCV 服务器使用的客户端身份验证方法。使用“system”将客户端身份验证工作委派给底层操作系统。使用“none”禁用客户端身份验证并授予对所有客户端的访问权限。 — 从版本 0-4100 开始 (p. 112) 可用。
authentication-threshold	整数 - 32-bit)	server	3	身份验证阈值 — 指定在服务器关闭连接之前，每个客户端的

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
				身份验证可以失败的次数。要允许无限制的身份验证尝试，请使用 0。—从版本 0-4100 开始 (p. 112) 可用。
passwd-file	字符串	server	"	密码文件 — 指定用于检查用户证书的密码文件 (仅限于 dcv 身份验证模式)。如果为空，请在 \${XDG_CONFIG_HOME}/NICE/dcv/passwd (对于 Linux) 或 %CSIDL_LOCAL_APPDATA%\NICE\dcv\passwd (对于 Windows) 中使用默认文件。—从版本 0-4100 开始 (p. 112) 可用。
pam-service-name	字符串	server	"dcv"	PAM 服务名称 — 指定 DCV 使用的 PAM 配置文件的名称。默认 PAM 服务名称为 "dcv"，对应 /etc/pam.d/dcv 配置文件。此参数仅在使用 "system" 身份验证方法时使用。—从版本 0-4100 开始 (p. 112) 可用。
enable-gssapi	true 或 false - 32-bit)	server	Linux : false - Windows : 0	启用 GSSAPI SASL 机制 — 启用或禁用 GSSAPI SASL 机制，该机制允许使用 kerberos 进行 DCV 身份验证。—从版本 3-6698 开始 (p. 108) 可用。
service-name	字符串	server	"dcv"	Service Name (服务名称) — 服务的注册名称 (通常为协议名称)。—从版本 0-8428 开始 (p. 106) 可用。
server-fqdn	字符串	server	"	Server FQDN — 指定服务器完全限定域名。空表示 gethostname ()。—从版本 3-6698 开始 (p. 108) 可用。
user-realm	字符串	server	"	服务器用户领域 — 指定服务器的用户领域。—自版本 3-6698 起 (p. 108) 可用。
ca-file	字符串	server	"	CA 文件 — 指定包含 DCV 服务器信任的证书颁发机构 (CA) 的文件。如果为空，请使用系统提供的默认信任存储。—从版本 0-4100 开始 (p. 112) 可用。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
auth-token-verifier	字符串	server	"	身份验证令牌验证程序的终端节点 — 指定 DCV 服务器使用的身份验证令牌验证程序的终端节点 (URL)。如果为空，则使用内部身份验证令牌验证程序。 — 从版本 0-4100 开始 (p. 112) 可用。
no-tls-strict	true 或 false - 32-bit)	server	Linux : false - Windows : 0	启用或禁用严格的证书验证 在连接到外部身份验证令牌验证程序时 — 启用或禁用严格的证书验证。如果身份验证令牌验证程序使用自签名证书，则必须禁用严格的证书验证。 — 自版本 0-4100 起 (p. 112) 可用。
allowed-http-host-regex	字符串	server	'^.+\$\$'	允许的主机正则表达式 — 指定表示此 DCV 服务器可以服务的主机名的正则表达式模式。如果传入 HTTP 请求的主机标头与此模式不匹配，则请求本身会失败，并伴随出现“403 (禁止访问)”的状态码。这是防止 HTTP 主机标头攻击的一种安全手段。该模式必须是有效的类似于 Javascript 的正则表达式。模式中的字母大小写同时匹配。示例：'^ (www\ \.) ?example\\.com\$\$'。 — 自版本 0-4100 起 (p. 112) 可用。
allowed-ws-origin-regex	字符串	server	"^https://.+\$\$"	允许的源 — 指定表示此 DCV 服务器接受的源的正则表达式模式。建立 WebSocket 连接时，客户端的握手中的源标头字段指示建立连接的脚本的源。如果传入 HTTP 请求的源标头与此模式不匹配，则请求本身会失败，并伴随出现“403 (禁止访问)”的状态码。这是防止跨站点 WebSocket 劫持 (CSWSH) 攻击的一种安全手段。该模式必须是有效的类似于 Javascript 的正则表达式。模式中的字母大小写同时匹配。源标头的形式如下：<scheme> "://" <host> [":" <port>]。示例：'^https :// (www\ \.) ?example\\.com (: 443) ?\$\$'。 — 自版本 0-4100 起 (p. 112) 可用。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
max-connections-per-user	整数 - 32-bit)	server	10	用户连接的最多数量 — 指定每个用户所允许的并发连接的最多数量。超出的连接将被拒绝。 — 从版本 0-4100 开始 (p. 112) 可用。
connection-estab-timeout	整数 - 32-bit)	server	5	建立连接超时 — 指定在超时之前允许完成连接过程所用的时间 (以秒为单位)。如果连接过程超时, 则连接被关闭。如果设置为 0, 则建立连接不会超时。 — 从版本 0-4100 开始 (p. 112) 可用。
connection-setup-timeout	整数 - 32-bit)	server	5	通道连接设置超时 — 指定在超时之前允许完成通道连接设置过程所用的时间 (以秒为单位)。如果设置过程超时, 则通道被关闭。如果设置为 0, 则通道连接设置不会超时。 — 从版本 0-4100 开始 (p. 112) 可用。
auth-connection-setup-timeout	整数 - 32-bit)	server	120	身份验证通道连接设置超时 — 指定在超时之前允许完成身份验证通道连接设置过程所用的时间 (以秒为单位)。如果设置过程超时, 则通道被关闭。如果设置为 0, 则禁用身份验证通道连接设置超时。 — 从版本 0-4100 开始 (p. 112) 可用。
ciphers	字符串	server	“ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-SHA256:ECDHE-RSA-AES256-SHA384”	TLS 连接上使用的密码列表 — 指定 TLS 连接上使用的密码列表。密码列表必须使用字符“:”分隔, 并且必须受 openssl 和客户端支持。 — 从版本 0-4100 开始 (p. 112) 可用。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
os-auto-lock	true 或 false - 32-bit)	自定义	Linux : true - Windows : 1	最后一个客户端连接结束时是否锁定操作系统会话 — 如果启用此项, 则在关闭最后一个客户端连接时, 操作系统会话将被锁定。每次要关闭会话时, 都会读取此设置。 — 自版本 1-5777 起 (p. 111) 可用。

license 参数

下表描述了 [license] 文件的 /etc/dcv/dcv.conf 部分中 (对于 Linux NICE DCV 服务器) 和 license 注册表中 (对于 Windows NICE DCV 服务器) 的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
license-file	字符串	session	"	许可证 — 指定试用许可证文件或 RLM 服务器端口和主机名。如果在 RLM 服务器上使用的是浮动许可证, 请使用此参数以 port@hostname 格式指定 RLM 服务器的端口和主机名。如果您使用的是延期试用许可证, 并且没有在默认位置放置 license.lic 文件, 请使用此参数指定 license.lic 许可证文件的完整路径。如果该默认文件不存在, 则使用试用许可证。此值将从配置中读取, 并在每次创建新会话时更新。 — 从版本 0-4100 开始 (p. 112) 可用。

input 参数

下表描述了 [input] 文件的 /etc/dcv/dcv.conf 部分中 (对于 Linux NICE DCV 服务器) 和 input 注册表中 (对于 Windows NICE DCV 服务器) 的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
enable-relative-mouse	true 或 false - 32-bit)	session	Linux : true - Windows : 1	是否允许相对鼠标移动 — 指定是否允许相对鼠标移动。 — 自版本 0-5121 起 (p. 111) 可用。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
enable-autorepeat	true 或 false - 32-bit)	session	Linux : true - Windows : 1	是否允许在 Linux 上自动重复 — 指定是否允许自动重复单个键。排除键组合和键修饰符。—自版本 2-6182起 (p. 110)可用。
enable-touch	true 或 false - 32-bit)	session	Linux : true - Windows : 1	是否允许触摸输入 — 指定是否启用触摸。—从版本 3-6698 开始 (p. 108)可用。
enable-stylus	true 或 false - 32-bit)	session	Linux : true - Windows : 1	是否允许触控笔输入 — 指定是否启用触控笔。—自版本 0-7318 起 (p. 107)可用。

display 参数

下表描述了 [display] 文件的 /etc/dcv/dcv.conf 部分中 (对于 Linux NICE DCV 服务器) 和 display 注册表项中 (对于 Windows NICE DCV 服务器) 的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
max-compressor-threads	整数 - 32-bit)	session	4	最大压缩程序线程数 — 指定压缩程序线程的最大数量。—从版本 0-4100 开始 (p. 112)可用。
target-fps	整数 - 32-bit)	session	-1	每秒目标帧数 — 指定允许的每秒最多帧数。值 0 表示无限制。值 -1 表示将根据服务器特征和会话类型确定 target-fps 值。当版本 < 2020.2 时, 无法识别 -1 值, 默认值为 25。—自版本 0-4100 起 (p. 112)可用。
grabber-target-fps	整数 - 32-bit)	session	0	帧抓取器每秒的目标帧数 — 设置每秒抓取帧的上限。值 0 是 target-fps 的默认值。并非所有帧捕获后端都遵循此设置。—自版本 1-5870起 (p. 110)可用。
enable-qu	true 或 false - 32-bit)	session	Linux : true - Windows : 1	是否发送质量更新 — 指定是否发送质量更新。—从版本 0-4100 开始 (p. 112)可用。
enable-client-resize	true 或 false - 32-bit)	session	Linux : true - Windows : 1	是否允许客户端设置显示布局 — 指定是否允许客户端设置显示布局。—从版本 0-4100 开始 (p. 112)可用。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
max-layout-area	整数 - 32-bit)	自定义	0	最大布局区域 (以像素为单位) — 设置客户端可请求的显示布局的最大区域 (以像素为单位)。超出此限制的布局将被忽略。此最大值旨在提供必须发送的显示数据量的上限, 而不提供显示布局几何体的约束。如果设置为 0, 则不对布局区域施加任何限制。将在每个客户端布局请求时重新加载该设置。—从版本 1-7423 开始 (p. 107) 可用。
max-head-resolution	字符串	自定义	(4096, 2160)	最大头分辨率 — 设置客户端可请求的显示头的最大分辨率。显示头等效于主显示器。在发出每个客户端布局请求时将重新加载该设置。当客户端请求更大的头分辨率时, 服务器会调整分辨率, 以确保它与此选项设置的最大宽度值和最大高度值匹配。—从版本 0-4100 开始 (p. 112) 可用。
web-client-max-head-resolution	字符串	自定义	(1920, 1080)	Web 客户端的最大头分辨率 — 设置 Web 客户端可请求的显示头的最大分辨率。显示头等效于主显示器。在发出每个客户端布局请求时将重新加载该设置。如果 Web 客户端明确设置了最大分辨率, 则忽略此设置。基于此选项设置的最大宽度值和最大高度值应用 max-head-resolution 限制选项。如果值设置为 (0, 0), 则忽略该值。—发布自版本 0-8428 (p. 106)。
min-head-resolution	字符串	自定义	(640, 480)	Min head resolution (最小头分辨率) — 设置客户端可请求的显示头的最小分辨率。显示头等效于主显示器。在发出每个客户端布局请求时将重新加载该设置。当客户端请求较小的分辨率时, 服务器会调整分辨率, 以确保它与此选项设置的最小宽度值和最小高度值匹配。—自版本 0-4100 起 (p. 112) 可用。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
max-num-heads	整数 - 32-bit)	自定义	4	最大头数 — 指定客户端可请求的显示头的最大数量。显示头等效于主显示器。在发出每个客户端布局请求时将重新加载该设置。当客户端请求更多数量的头时，服务器会调整头数，以使值不超过此选项设置的值。 — 从版本 0-4100 开始 (p. 112) 可用。
console-session-default-layout	字符串	session	[]	控制台会话的默认屏幕分辨率和位置 — 指定控制台会话的默认屏幕分辨率和位置。如果设置了该项，则 DCV 在启动时设置请求的布局。可以为每个监视器配置分辨率 (w,h) 和位置 (x,y)。启用所有指定的监视器。默认布局示例值：[{ 'w' : <800>, 'h' : <600>, 'x' : <0>, 'y' : <0>}, { 'w' : <1024>, 'h' : <768>, 'x' : <800>, 'y' : <0>}] 自 — 0-5600 版本以来 (p. 111) 可用。
use-grabber-dirty-region	true 或 false - 32-bit)	session	Linux : true - Windows : 1	是否使用脏区域 — 指定是否使用脏屏幕区域。如果启用，则抓取器尝试从屏幕的脏区域计算新帧。 — 自版本 0-4100 起 (p. 112) 可用。
cuda-devices	字符串	连接	[]	用于流编码的 CUDA 设备 — 指定 DCV 用于分发编码和 CUDA 工作负载的本地 CUDA 设备的列表。每个设备均由一个可从 nvidia-smi 命令中检索的数字标识。例如，cuda-devices=['0', '2'] 表示 DCV 使用两个 ID 分别为 0 和 2 的 GPU。此设置与 CUDA_VISIBLE_DEVICES 环境变量类似，但它仅适用于 DCV。如果未设置 选项，DCV 将使用从 0 开始的增量会话索引来选择要使用的下一个设备。 — 自版本 2-6182 起 (p. 110) 可用。

display/linux 参数

下表描述了 [display/linux] 文件的 /etc/dcv/dcv.conf 部分中 (对于 Linux NICE DCV 服务器) 和 display/linux 注册表中 (对于 Windows NICE DCV 服务器) 的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
gl-displays	字符串	session	[:0.0]	3D 加速 X 显示器 — 指定 DCV 用于虚拟会话中 OpenGL 渲染的本地 3D 加速 X 显示器和屏幕列表。如果缺少此值，则无法在虚拟会话中运行 OpenGL 应用程序。对于控制台会话，忽略此设置。—从版本 0-4100 开始 (p. 112) 可用。
h264-encoder-displays	字符串	连接	[]	H.264 编码器 X 显示器 — 指定支持加速 H.264 编码的本地 X 显示器和屏幕的列表。如果为空，则 DCV 使用为 OpenGL 渲染选定的同一显示器。仅当系统上安装的某些 GPUs 未使用支持的技术之一为 H.264 编码提供加速时，此设置才有用。—从版本 0-4100 开始 (p. 112) 可用。

audio 参数

下表描述了 [audio] 文件的 /etc/dcv/dcv.conf 部分中 (对于 Linux NICE DCV 服务器) 和 audio 注册表项中 (对于 Windows NICE DCV 服务器) 的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
source-channels	整数 - 32-bit)	session	2	源设备上播放的通道数。—设置从源设备获取音频时应考虑的通道数。驱动程序提供的设备可以支持多个通道，例如，DCV 驱动程序支持 8 个通道，但 DCV 需要知道需有效使用多少个通道才能正确混音。集值必须小于或等于设备支持的通道数。允许的值包括：0 (使用所有通道)、2 (立体声)、4 (4.0 四声道声道)、6 (5.1 环绕)、8 (7.1 环绕声道)。默认值为 2 (立体声)。—发布自版本 0-8428 (p. 106)。

log 参数

下表描述了 [log] 文件的 /etc/dcv/dcv.conf 部分中 (对于 Linux NICE DCV 服务器) 和 log 注册表项中 (对于 Windows NICE DCV 服务器) 的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
directory	字符串	server	"	日志输出目录 — 指定保存日志的目标位置。如果未指定，则默认为 Windows 上的 ProgramData 和 Linux 上的“/var/log/dcv/”。—自版本 0-4100 起 (p. 112) 可用。
level	字符串	自定义	“info”	日志级别 — 指定日志文件的详细程度。详细程度有 (按提供的信息量顺序排列)：“错误”、“警告”、“信息”和“调试”。新值在配置上进行更改并传播到 DCV 代理进程后立即生效。当版本 <= 2019.1 时，DCV 代理进程上的日志级别仅在启动时设置。—自版本 0-4100 起 (p. 112) 可用。
rotate	整数 - 32-bit)	server	10	日志文件分割数量 — 指定日志文件在被删除前分割的次数。如果值为 0，则会删除旧版本而不是轮换旧版本。—从版本 0-4100 开始 (p. 112) 可用。
transfer-audit	字符串	server	“none”	要审核的传输方向 — 指定要审核的传输方向。如果启用此参数，则在服务器和客户端之间传输新的 CSV 文件日志。允许的值：none、server-to-client、client-to-server 和 all。如果此值缺失或等于“none”，则将禁用传输审核并且不会创建任何文件。—从版本 0-4100 开始 (p. 112) 可用。

windows 参数

下表描述了 [windows] 文件的 /etc/dcv/dcv.conf 部分中 (对于 Linux NICE DCV 服务器) 和 windows 注册表项中 (对于 Windows NICE DCV 服务器) 的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
disable-display-sleep	true 或 false - 32-bit)	session	Linux : true - Windows : 1	防止显示器进入节能模式 — 指定是否禁止显示器进入节能模式。—从版本 0-4100 开始 (p. 112) 可用。
printer	字符串	session	“DCV printer”	要设置为默认值的打印机 — 指定虚拟 DCV 打印机的名称。默认为“DCV printer”。该名称用于更改

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
				系统上的默认打印机。如果设置为空字符串，则 DCV 不会更改当前默认打印机。—自版本 0-4100 起 (p. 112) 可用。

clipboard 参数

下表描述了 [clipboard] 文件的 /etc/dcv/dcv.conf 部分中 (对于 Linux NICE DCV 服务器) 和 clipboard 注册表项中 (对于 Windows NICE DCV 服务器) 的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
已启用	true 或 false - 32-bit)	session	Linux : true - Windows : 1	是否应启用剪贴板功能 — 指定是否启用剪贴板功能。如果禁用剪贴板功能，用户将无法使用剪贴板远程处理。剪贴板监控也将被禁用。—发布自版本 0-8428 (p. 106)。
max-payload-size	整数 - 32-bit)	session	20971520	剪贴板数据的最大大小 — 指定可从服务器传输到客户端的剪贴板数据的最大大小 (以字节为单位)。如果缺少此值，则默认限制为 20 MB。—从版本 0-4334 开始 (p. 112) 可用。
max-text-len	整数 - 32-bit)	session	-1	剪贴板文本的最多字符数 — 指定可从服务器传输到客户端的剪贴板文本的最多字符数。如果此值缺失或设置为 -1，则不施加任何限制。—自版本 0-4334 起 (p. 112) 可用。
max-image-area	整数 - 32-bit)	session	-1	剪贴板图像的最大面积 — 指定可从服务器传输到客户端的剪贴板图像的最大面积 (像素数)。如果此值缺失或设置为 -1，则不施加任何限制。—自版本 0-4334 起 (p. 112) 可用。
primary-selection-paste	true 或 false - 32-bit)	session	Linux : false - Windows : 0	在 Linux — 桌面上启用主选区域支持多个剪贴板：通用剪贴板和主选区域。选择内容后，将更新或复制主选区域。然后，可以使用鼠标的中间按钮或 Shift +Insert 组合键粘贴它。启用后，客户端的剪贴板内容也将插入

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
				到主选区中。 —自版本 0-7318 起 (p. 107) 可用。
primary-selection-copy	true 或 false - 32-bit)	session	Linux : false - Windows : 0	从 Linux — 桌面启用主选区副本支持多个剪贴板：通用剪贴板和主选区。选择内容后，将更新或复制主选区域。然后，可以使用鼠标的中间按钮或 Shift+Insert 组合键粘贴它。启用后，将监控主要选择并将更新传播到客户端。 —自版本 0-7318 起 (p. 107) 可用。
update-timeout	整数 - 32-bit)	session	200	Update event notification timeout (更新事件通知超时) — 指定从上次向客户端发送通知的更新事件等待的时间 (以毫秒为单位)。默认值 200 毫秒。 —发布自版本 1-8942 (p. 105)。

smartcard 参数

下表描述了 [smartcard] 文件的 /etc/dcv/dcv.conf 部分中 (对于 Linux NICE DCV 服务器) 和 smartcard 注册表项中 (对于 Windows NICE DCV 服务器) 的配置参数。

参数	类型 - Windows 注册表类型	重新加载上下文	默认值	描述
enable-cache	字符串	自定义	“default-off”	是否启用智能卡缓存消息。 — 启用或禁用智能卡缓存。启用此项后，NICE DCV 服务器将缓存从客户端的智能卡收到的最后一个值。未来的调用直接从服务器的缓存中检索，而不是从客户端检索。这有助于减少在客户端和服务器之间传输的流量，并提高性能。允许的值包括“always-on”、“always-off”、“default-on”和“default-off”。每次启动客户端智能卡应用程序时，都会从配置中读取此值。 —自版本 2-6182起 (p. 110) 可用。

修改配置参数

本节介绍如何修改 NICE DCV 服务器的配置参数。有关 Windows 服务器的注册表项、Linux 服务器的部分、参数名称、类型和有效值的详细信息，请参阅 [NICE DCV 服务器 参数参考 \(p. 83\)](#)。

主题

- [Windows NICE DCV 服务器s \(p. 99\)](#)
- [Linux NICE DCV 服务器s \(p. 99\)](#)

Windows NICE DCV 服务器s

对于 Windows NICE DCV 服务器，请使用 Windows 注册表编辑器、PowerShell 或命令行修改配置参数。

使用 Windows 注册表编辑器修改配置参数

1. 打开 Windows 注册表编辑器。
2. 导航到以下注册表路径：

```
HKEY_USERS\S-1-5-18\Software\GSettings\com\nicesoftware\dcv/
```

3. 选择参数所在的注册表项。如果该注册表项不存在，则使用 [NICE DCV 服务器 参数参考 \(p. 83\)](#)中所述的确切的注册表项名称创建该注册表项。
4. 打开（双击）该参数。如果该参数不存在，则使用 [NICE DCV 服务器 参数参考 \(p. 83\)](#)中所述的类型和名称添加该参数。

使用 PowerShell 修改配置参数

1. 以管理员身份运行 PowerShell。
2. 使用 [NICE DCV 服务器 参数参考 \(p. 83\)](#)中所述的注册表项名称添加该注册表项。

```
PS C:\> New-Item -Path "Microsoft.PowerShell.Core\Registry::\HKEY_USERS  
\S-1-5-18\Software\GSettings\com\nicesoftware\dcv\" -Name registry_key -Force
```

3. 使用 [NICE DCV 服务器 参数参考 \(p. 83\)](#)中所述的类型和名称在注册表项中创建参数。

```
PS C:\> New-ItemProperty -Path "Microsoft.PowerShell.Core\Registry::\HKEY_USERS  
\S-1-5-18\Software\GSettings\com\nicesoftware\dcv\registry_key" -Name parameter_name -  
PropertyType parameter_type -Value parameter_value -Force
```

使用命令行修改配置

1. 以管理员身份运行命令行。
2. 使用 [NICE DCV 服务器 参数参考 \(p. 83\)](#)中所述的注册表项名称、参数类型和名称创建注册表项并添加参数。

```
C:\> reg.exe ADD "HKEY_USERS\S-1-5-18\Software\GSettings\com\nicesoftware\dcv  
\registry_key" /v parameter_name /t parameter_type /d parameter_value /f
```

Linux NICE DCV 服务器s

对于 Linux NICE DCV 服务器，可使用文本编辑器或命令行工具如 `crudini` 修改配置参数。

使用文本编辑器修改配置参数

1. 使用所需的文本编辑器打开 `/etc/dcv/dcv.conf`。

2. 找到文件中的相应部分。如果该部分不存在，则使用 [NICE DCV 服务器 参数参考 \(p. 83\)](#)中所述的部分名称添加该部分。

```
[section]
```

3. 找到部分中的相应参数并修改其值。如果该参数不存在，则使用 [NICE DCV 服务器 参数参考 \(p. 83\)](#)中所述的参数名称添加该参数。

```
parameter_name="parameter_value"
```

4. 保存并关闭文件。

使用 `crudini` 修改配置参数

使用 [NICE DCV 服务器 参数参考 \(p. 83\)](#)中所述的部分和参数名称创建该部分并添加参数。

```
$ sudo crudini --set /etc/dcv/dcv.conf section_name parameter_name 'parameter_value'
```

NICE DCV 终止支持使用期限

结束支持生命周期（EOSL）定义 NICE DCV 的特定主要版本（及其所有次要版本）不再收到支持的时间点，并且不再针对与更新版本的兼容性进行测试。NICE DCV

在 ESL 日期之前，NICE DCV 支持团队继续提供对配置问题的全面支持。缺陷解析和功能请求仅针对最新版本的 NICE DCV 服务器和 NICE DCV 客户端实施。它们不会针对旧版本实施。

ESL 日期之后，不再提供进一步的支持或维护。我们还将停止测试兼容性问题。要继续支持，您必须升级到最新的 NICE DCV 版本。

主题

- [EOL 时间轴 \(p. 101\)](#)
- [客户的 ESL 路径 \(p. 101\)](#)
- [ESLFAQs \(p. 101\)](#)

EOL 时间轴

下表显示了 NICE DCV 主要版本的 EOSL 时间轴。

NICE DCV 主要版本	首次发布日期	ESL 日期
NICE DCV 2016.x	2015 年 12 月 31 日	2021 年 3 月 31 日
NICE DCV 2017.x	2017 年 12 月 18 日	2021 年 12 月 31 日
NICE DCV 2019.x	2019 年 8 月 5 日	2020 年 12 月 31 日
NICE DCV 2020.x	2020 年 4 月 16 日	2023 年 12 月 31 日

客户的 ESL 路径

如果您在 NICE DCV 上运行 AWS，则不需要 NICE DCV 的许可证。您只需为用于工作负载的基础 AWS 资源付费。如果您当前使用的 NICE DCV 版本超过其 EOSL 日期，请使用 [NICE DCV 下载页面](#) [NICE 升级到最新的版本](#)，或者使用 [NICE DCV AMI](#) 从 AWS Marketplace 继续接收支持。

如果您在本地运行 NICE DCV 或使用第三方云服务提供商，并且您当前使用的 NICE DCV 的版本早于其 EOSL 日期，请联系您的经销商或分销商以评估可用的升级路径。如果您有有效的支持合同，则可以免费升级到最新版本的 NICE DCV。有关 NICE DCV 分销商和经销商的信息，请参阅 [NICE 网站](#)。

ESLFAQs

1. 我使用的是已在本地或通过第三方云服务提供商到达其 EOSL 的 NICE DCV 版本，但我已拥有支持合同。我是否会受 EOSL 的影响？

如果您具有有效的支持合同，则 NICE DCV 支持合同的条款使您能够将 NICE DCV 许可证升级到最新版本，而无需额外付费。在这种情况下，影响最小。如果您的支持合同已过期，您可以使用以下方法之一继续获得完全支持：

1. 使用新的付费许可证升级到最新版本的 NICE DCV 版本。
2. 在 EOL 时间线之前将支持合同续订到 ，这为您提供了到最新版本的 NICE DCV 的升级路径。
3. 通过支付恢复费用来恢复旧支持合同，该费用相当于您的支持合同到期后的时间段内对支持服务的当前费用的 70%。

2. 我使用的是已达到 NICE DCV 上的 EOSL 的 Amazon EC2 版本，我应该做什么才能升级到支持的版本？

升级到完全支持的 NICE DCV 版本以在 Amazon EC2 上使用时，客户始终可以免费使用。

3. 我能否使用已到达其 EOSL 的 NICE DCV 客户端 版本及支持的 NICE DCV 服务器，或相反？

是，但我们强烈建议您将客户端和服务器软件升级到最新版本，因为错误修复不再应用于已到达其 EOSL 的版本。

NICE DCV 的发布说明和文档历史记录

本页面提供了 NICE DCV 的发布说明和文档历史记录。

主题

- [NICE DCV 发布说明 \(p. 103\)](#)
- [文档历史记录 \(p. 112\)](#)

NICE DCV 发布说明

此部分按发布日期描述了 NICE DCV 功能、改进和错误修复。

主题

- [DCV 2020.2-9662— 2020 年 12 月 4 日 \(p. 103\)](#)
- [DCV 2020.2-958— 2020 年 11 月 11 日 \(p. 104\)](#)
- [DCV 2020.1-9012— 2020 年 9 月 30 日 \(p. 104\)](#)
- [DCV 2020.1-9012— 2020 年 8 月 24 日 \(p. 105\)](#)
- [DCV 2020.1-8942— 2020 年 8 月 3 日 \(p. 105\)](#)
- [DCV 2020.0-8428 — 2020 年 4 月 16 日 \(p. 106\)](#)
- [DCV 2019.1-7644 — 2019 年 10 月 24 日 \(p. 106\)](#)
- [DCV 2019.1-7423 — 2019 年 9 月 10 日 \(p. 107\)](#)
- [DCV 2019.0-7318 — 2019 年 8 月 5 日 \(p. 107\)](#)
- [DCV 2017.4-6898 — 2019 年 4 月 16 日 \(p. 108\)](#)
- [DCV 2017.3-6698 — 2019 年 2 月 24 日 \(p. 108\)](#)
- [DCV 2017.2-6182 — 2018 年 10 月 8 日 \(p. 110\)](#)
- [DCV 2017.1-5870 — 2018 年 8 月 6 日 \(p. 110\)](#)
- [DCV 2017.1-5777 — 2018 年 6 月 29 日 \(p. 111\)](#)
- [DCV 2017.0-5600 — 2018 年 6 月 4 日 \(p. 111\)](#)
- [DCV 2017.0-5121 — 2018 年 3 月 18 日 \(p. 111\)](#)
- [DCV 2017.0-4334 — 2018 年 1 月 24 日 \(p. 112\)](#)
- [DCV 2017.0-4100 — 2017 年 12 月 18 日 \(p. 112\)](#)

DCV 2020.2-9662— 2020 年 12 月 4 日

内部版本号	更改和错误修复
<ul style="list-style-type: none">• nice-dcv-server : 9662• nice-dcv-client (Windows) : 7490	<ul style="list-style-type: none">• Web 浏览器客户端中的安全改进。• 提高了将 Amazon EC2 G4ad 实例与 Windows 结合使用时的性能和可靠性。

内部版本号	更改和错误修复
<ul style="list-style-type: none"> nice-dcv-viewer (MacOS) : 2117 nice-dcv-viewer (Linux) : 3007 nice-xdcv : 359 nice-dcv-gl : 881 nice-dcv-gltest : 259 nice-dcv-ext-authenticator : 125 	<ul style="list-style-type: none"> 修复了 Windows 客户端的连接设置对话框中的端口选择问题。

DCV 2020.2-958— 2020 年 11 月 11 日

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> nice-dcv-server : 9608 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-ext-authenticator : 125 	<ul style="list-style-type: none"> 添加了对 QUIC (基于 UDP) 传输协议的支持。 增加了对 SLES 15 和 Ubuntu 20.4 的支持。 增加了对 Windows NICE DCV 服务器的智能卡支持。 	<ul style="list-style-type: none"> 对于服务器上托管的控制台会话和具有 NVIDIA GPU 的 EC2 实例, NICE DCV 帧率限制器现在设置为 60 FPS。 使用 NVIDIA GPU 优化了 EC2 实例上托管的 Windows NICE DCV 服务器上的 GPU 和 CPU 资源的使用。 添加了 list-endpoints NICE DCV CLI 命令, 其中列出了当前活动的终端节点。 CLI 命令支持 version 选项。NICE DCV--json 在 Linux 服务器上, create-session NICE DCV CLI 命令支持 --disable-login-monitor 选项。 改进了与 Linux NICE DCV 服务器上的不同显示管理器的兼容性。 解决了键盘输入处理方面的各种限制。 USB 设备允许列表文件现在动态重新加载。

DCV 2020.1-9012— 2020 年 9 月 30 日

内部版本号	更改和错误修复
<ul style="list-style-type: none"> nice-dcv-server : 9012 nice-dcv-client (Windows) : 7342 nice-dcv-viewer (MacOS) : 1986 nice-dcv-viewer (Linux) : 1545 nice-xdcv : 338 nice-dcv-gl : 840 	<ul style="list-style-type: none"> 添加了缺少的 macOS 客户端图标。

内部版本号	更改和错误修复
<ul style="list-style-type: none"> nice-dcv-glttest : 246 nice-dcv-ext-authenticator : 111 	

DCV 2020.1-9012— 2020 年 8 月 24 日

内部版本号	更改和错误修复
<ul style="list-style-type: none"> 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-glttest : 246 nice-dcv-ext-authenticator : 111 	<ul style="list-style-type: none"> 修复了 AWS Amazon S3区域中的 GovCloud 访问 基于 Web 的客户端改进

DCV 2020.1-8942— 2020 年 8 月 3 日

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> 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-glttest : 246 nice-dcv-ext-authenticator : 111 	<ul style="list-style-type: none"> Linux NICE DCV 服务器 现在支持基于 AWS Graviton2 的 Arm 实例，如 M6g、C6g 和 R6g。有关更多信息，请参阅 AWS Graviton 处理器。 在 Linux CentOS 上支持 RHEL 8.x 和 NICE DCV 服务器 8.x。 增加了在使用 Windows NICE DCV 服务器 和 Windows NICE DCV 客户端 时对打印机重定向的支持。 增加了对 macOS 和 Linux 本机 NICE DCV 客户端 上具有压力敏感性的触控笔支持。 添加了 Linux NICE DCV 服务器 和 Linux NICE DCV 客户端 的环绕声 5.1 支持。 增加了对 Linux NICE DCV 本机客户端 的触摸屏支持。 您现在可以将自定义名称与 NICE DCV 会话关联。 支持在 macOS 本机 NICE DCV 客户端 上进行硬件加速解码和渲染。 	<ul style="list-style-type: none"> 在没有 GPU 的 NICE DCV 实例上增加了对新的 Amazon EC2 虚拟显示驱动程序的支持。 解决了在使用 NVENC 编码器时，因颜色空间转换导致视觉伪影的问题。 命令现在始终包含控制台会话 (如果存在) <code>dcv list-sessions</code> 在较新的 Linux 发行版上，控制台会话的代理现在作为桌面会话的一部分启动，以更好地支持较新的显示管理器，如 GDM3。 使用 <code>dcv://</code> 模式激活 URL 时，本机客户端现在自动打开。 改进了 macOS 本机客户端 和 Web 客户端 处理键盘修饰符的方式。 改进了 DCV-GL 中的视觉对象和 <code>fbconfig</code> 选择以改进对某些应用程序的支持。 减少了文件传输期间的 CPU 使用率

内部版本号	新功能	更改和错误修复
		<ul style="list-style-type: none"> 改进了 Web 浏览器客户端中的 WebGL 渲染以减少资源使用。

DCV 2020.0-8428 — 2020 年 4 月 16 日

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> 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-ext-authenticator : 87 	<ul style="list-style-type: none"> Linux 服务器上的触控笔和触摸支持。 Windows 服务器到 Windows 本机客户端上的 7.1 环绕立体声播放。 Linux 本机客户端上的硬件加速和触控笔支持。 用于在服务器端设置显示布局的新 API 命令。 Microsoft Edge 浏览器 (版本 79.0.309 或更高版本) 上的多显示器 Web 客户端支持。 	<ul style="list-style-type: none"> 在全屏模式下, 现在可以隐藏 Windows 客户端上的工具栏控制。 Windows 本机客户端上的 NTLM 代理支持。 改进了对使用 NVIDIA 适配器的 Windows 无头物理主机的支持。 删除了对传统 NVIDIA NvIFR 库的支持。 增加了对最新的 Windows 10 上的 Windows 图形捕获 API 的支持。 增加了对 Amazon EC2 的支持 EC2 实例上的实例元数据服务 (IMDS) v2。 DCV CLI 提供了新的 on-client-connected/disconnected 命令来检测客户端与会话连接或断开连接的时间。 增加了对指定主机名以绑定外部身份验证器证书的支持。 DCV-GL 现在在支持 GL 厂商中立的分派库 (GLvnd) 的系统上使用它。

DCV 2019.1-7644 — 2019 年 10 月 24 日

内部版本号	更改和错误修复
<ul style="list-style-type: none"> 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 	<ul style="list-style-type: none"> 修复了 NICE EnginFrame 和其他会话管理器使用的集成 API 中的问题。 修复了 32 位版本的 Windows 本机客户端的问题。

内部版本号	更改和错误修复
<ul style="list-style-type: none"> nice-dcv-ext-authenticator : 77 	

DCV 2019.1-7423 — 2019 年 9 月 10 日

内部版本号	更改和错误修复
<ul style="list-style-type: none"> 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-ext-authenticator : 77 	<ul style="list-style-type: none"> 改进了 Windows 上的 DCV 服务器的安全性。 修复了 Linux 上的 Autodesk Maya 的渲染问题。 增加了与键盘操作相关的改进和错误修复。

DCV 2019.0-7318 — 2019 年 8 月 5 日

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> 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-ext-authenticator : 72 	<ul style="list-style-type: none"> Web 客户端支持多显示器。 Windows Server 2019 上支持触控笔输入。 和 Linux 本机客户端上的音频输入/输出。macOS Linux 服务器上的增强剪贴板功能 (中键单击粘贴)。 	<ul style="list-style-type: none"> 现在, Windows 上的触摸输入会遵循压力值 (如果可用)。 改进了 Windows 上具有异构图形适配器的系统的行为。 减少了检测非活动连接所需的时间 (例如, 响应客户端上从有线网络到 Wi-Fi 网络的更改)。 减少了无法在 Linux 上捕获光标图标时的日志记录。 支持在虚拟会话 Xdcv 组件中禁用复合扩展。 能够设置并发虚拟会话数限制。 改进了脚本与已安装 Bash 5 的系统的兼容性。 自动检测并使用 OpenGL 和 GLES 在 Linux 客户端上进行渲染。 更新了当 GL 窗口的可见性发生变化时的 DCV-GL 屏幕缓冲区。 修复了 Windows 7 上的 Windows 客户端中的鼠标滚轮检测。

内部版本号	新功能	更改和错误修复
		<ul style="list-style-type: none"> 修复了导致 Windows 客户端在某些 Windows 7 系统上加载库时出现故障的问题。 改进了横向打印文档时 Windows 客户端上的打印。

DCV 2017.4-6898 — 2019 年 4 月 16 日

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> 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-ext-authenticator : 70 	<ul style="list-style-type: none"> 适用于 macOS 的新本机客户端。 	<ul style="list-style-type: none"> Windows 本机客户端现在使用硬件加速进行解码和渲染 (如果它在系统中可用) 。 现在, dcv 命令行工具在 Windows 和 Linux 上使用相同的选项和输出格式。 dcv 命令行工具现在报告有关许可证的信息。 客户端现在会在因不活动导致断开连接之前向用户显示警告。 改进了对使用多个修饰符的键盘组合的支持。 针对通信失败改进了与 Reprise License Manager 的交互的稳健性。 在 Linux 上, dcvusers 命令行工具现在默认为将数据保存到 dcv 用户主目录。 在 Linux 上将 NVENC 硬件编码器与多个 nvidia-smi 结合使用时, 遵循 GPUs 工具使用的相同顺序。 Linux 客户端现在接收和处理来自 Windows DCV 打印机的打印文件。

DCV 2017.3-6698 — 2019 年 2 月 24 日

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> nice-dcv-server : 6698 nice-dcv-client : 5946 nice-dcv-viewer (Linux) : 683 nice-xdcv : 207 nice-dcv-gl : 471 	<ul style="list-style-type: none"> 增加了对 Kerberos (GSSAPI) 身份验证的支持。 增加了对支持触摸的 Windows 版本上的触摸事件的支持。 在使用系统身份验证 (Windows 凭证提供程序) 时自动解锁 Windows 会话。 	<ul style="list-style-type: none"> 增加了选择使用 Y'UV444 编码的选项。 EL6 RPM 现在包括 NVENC 编码器模块。 Windows 系统身份验证现在接受 name@domain 格式。

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> • nice-dcv-gltest : 210 • nice-dcv-ext-authenticator : 66 		<ul style="list-style-type: none"> • Yubikey USB 设备现已添加到允许列表中。 • 改进了日语键盘支持。 • 输入授权权限更为精细。增加了操作虚拟光标的 pointer 权限。相对鼠标模式取决于鼠标（针对运动注入）和指针（针对运动反馈）。增加了在 Windows 上处理 SAS 的 keyboard-sas 权限（Control+Alt+Del）。keyboardsas 取决于 keyboard 权限。 • 修复了支持异步剪贴板 API 的浏览器上 Web 客户端中的空剪贴板事件的问题。 • 修复了导致客户端无法接收到第一帧的捕获模块上的竞用问题。 • 改进了对并发文件存储传输的处理。 • 在 Windows 上使用较新的 NVIDIA 驱动程序修复了 NvIFR。新的驱动程序改变了行为。现在将自动检测驱动程序版本，并相应地执行内存处理。 • 永不停止重试重新获取 RLM 许可证令牌。这使您能够从 licensing error 状态恢复，甚至在延长时段后也是如此。 • 在 Windows 客户端中，增加了一个选项来设置全屏键盘快捷键。 • 在 Windows 客户端中，改进了在多个显示器之间拖动窗口时的自适应逻辑。 • 在 Windows 客户端中，修复了 UI 未触发断开连接时的 prompt-reconnect 选项。 • 修复了 DCV-GL 与 NVIDIA 驱动程序 410.xx 不兼容的问题。 • 使用 Matlab 和 Blender 应用程序修复了 DCV-GL 中的性能下降问题。

DCV 2017.2-6182 — 2018 年 10 月 8 日

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> 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-ext-authenticator : 58 	<ul style="list-style-type: none"> 增加了对 Linux 虚拟会话的音频播放支持。 改进了智能卡性能。 增加了 Linux 客户端的文件传输支持。 	<ul style="list-style-type: none"> 与键盘操作相关的改进和错误修复。 更改配置中的日志级别不再需要重新启动服务器。 Windows 服务器安装程序现在跳过 Microsoft C 运行时可再分发组件的安装 (如果它已安装)。 在 EC2 上运行时, 如果访问许可证的 S3 失败, 则用户界面中将显示一条通知。 Linux dcv 命令行工具现在支持 list-connections 和 describe-session 子命令, 并包含一个用于发出 JSON 输出的选项。 在 display 部分中添加了一个 cuda-devices 设置, 该设置可将服务器配置为在不同的 CUDA 设备上分发 NVENC 编码。 改进了处理多个并发命令时的会话创建代码的可靠性。 将默认剪贴板限制增加到了 20 MB。 Windows 客户端现在检测旧 .dcv 文件并启动 DCV 2016 Endstation (如果已安装)。 DCV 简单外部身份验证器现在始终使用系统 Python 解释器, 而不是环境中设置的解释器。 改进了 DCV-GL 的回读策略, 以提高性能和可靠性。 DCV-GL 现在检查窗口大小是否在前缓冲区回读后发生了更改。这修复了 Coot 应用程序的渲染问题。

DCV 2017.1-5870 — 2018 年 8 月 6 日

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> nice-dcv-server : 5870 nice-dcv-client : 5813 nice-dcv-viewer (Linux) : 450 	<p>发布了 Ubuntu 18.04 程序包。在控制台模式下工作时, 必须将系统配置为使用 LightDM 或您选择的其他显示管理器, 因为 GDM 不</p>	<ul style="list-style-type: none"> 现在可在创建会话时读取许可证设置, 这使管理员能够更改此设置, 而无需重新启动服务器。

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> nice-xdcv : 170 nice-dcv-gl : 366 nice-dcv-gltest : 198 nice-dcv-ext-authenticator : 53 	<p>会公开所需的 X11 显示信息。虚拟会话不受此限制的影响。</p>	<ul style="list-style-type: none"> 解决了 Windows 客户端存在的导致程序在某些系统上意外退出的稳定性问题。 减少了可能出现的错误情况下的日志记录。

DCV 2017.1-5777 — 2018 年 6 月 29 日

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> 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-ext-authenticator : 51 	<ul style="list-style-type: none"> 增加了 Linux 本机客户端。 增加了对 3DConnexion 鼠标和 USB 存储设备的支持。 Windows 会话在最后一个客户端断开连接时自动锁定。 	<ul style="list-style-type: none"> Linux 版本中的性能改进。 将 NVIDIA 设备上的默认硬件编码器更改为 NVENC，以避免新 NVIDIA 驱动程序中的 NvIFR 出现问题。 改进了 Linux 上的智能卡支持。 修复了使用 Linux 控制台会话时上传文件的文件权限。

DCV 2017.0-5600 — 2018 年 6 月 4 日

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> nice-dcv-server : 5600 nice-dcv-client : 5600 nice-xdcv : 160 nice-dcv-gl : 279 nice-dcv-gltest : 184 nice-dcv-ext-authenticator : 48 	<ul style="list-style-type: none"> 增加了对 Linux 上的多个显示器的支持。 Windows 客户端性能改进。 在 Chrome 66+ 上使用了新的剪贴板 API。 增加了适用于 Windows 的 NVENC 编码器。 	<ul style="list-style-type: none"> EC2 上的使用现在需要能够从运行 DCV 服务器的实例访问 S3。 针对服务器帧处理和 Windows 客户端解码的性能改进。 修复了与 NumPad 和受阻修饰符相关的键盘问题。 防止在 Linux 上使用外部身份验证器时发生文件描述符泄露。 修复了可能出现的智能卡连接错误。

DCV 2017.0-5121 — 2018 年 3 月 18 日

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> nice-dcv-server : 5121 nice-dcv-client : 5121 nice-xdcv : 146 nice-dcv-gl : 270 nice-dcv-gltest : 184 	<ul style="list-style-type: none"> Windows 本机客户端现在可以识别 DPI。 增加了对相对鼠标移动模式的支持。 	<ul style="list-style-type: none"> 防止 Linux 上的 Ansys cfx5solve 挂起。 修复了 Windows 10 上可能出现的代理挂起情况。

内部版本号	新功能	更改和错误修复
<ul style="list-style-type: none"> nice-dcv-ext-authenticator : 46 		<ul style="list-style-type: none"> 对 Web 客户端用户界面的改进。 指定域时规范化的 Windows 用户名。 修复了 RHEL6 上的外部验证器。

DCV 2017.0-4334 — 2018 年 1 月 24 日

内部版本号	更改和错误修复
<ul style="list-style-type: none"> nice-dcv-server : 4334 nice-dcv-client : 4334 nice-xdcv : 137 nice-dcv-gi : 254 nice-dcv-glttest : 184 nice-dcv-ext-authenticator : 45 	<ul style="list-style-type: none"> 改进了键盘操作。 修复了 RHEL6 上的 DBus 问题，其中关闭会话不允许创建新会话。 改进了对本机客户端上的 SOCKS5 代理的支持。 防止在虚拟会话上运行时 Headwave 发生崩溃。 防止在虚拟会话上运行时 Chimera 发生崩溃。 改进了对虚拟会话的字体支持。

DCV 2017.0-4100 — 2017 年 12 月 18 日

内部版本号
<ul style="list-style-type: none"> nice-dcv-server : 4100 nice-dcv-client : 4100 nice-xdcv : 118 nice-dcv-gi : 229 nice-dcv-glttest : 158 nice-dcv-ext-authenticator : 35

文档历史记录

下表介绍了此版本的 NICE DCV 的文档。

更改	描述	日期
NICE DCV 版本 2020.2	NICE DCV 2020.2 现已推出。有关更多信息，请参阅 DCV 2020.2-958— 2020 年 11 月 11 日 (p. 104) 。	2020 年 11 月 11 日
NICE DCV 版本 2020.1	NICE DCV 2020.1 现已推出。有关更多信息，请参阅 DCV 2020.1-8942— 2020 年 8 月 3 日 (p. 105) 。	2020 年 8 月 03 日

更改	描述	日期
NICE DCV 版本 2020.0	NICE DCV 2020.0 包括对环绕立体声 7.1、触摸和触控笔以及使用新 Microsoft Edge 浏览器的多显示器的支持。有关更多信息，请参阅 管理员指南NICE DCV 服务器 中的安装 。NICE DCV	2020 年 4 月 16 日
HTTP 响应标头	可将 NICE DCV 服务器配置为发送其他 HTTP 响应标头。	2019 年 8 月 26 日
macOS 客户端	NICE DCV 现在提供 macOS 客户端。有关更多信息，请参阅 https://docs.amazonaws.cn/dcv/latest/userguide/client-mac.html 用户指南 中的 NICE DCV macOS 客户端。	2019 年 4 月 18 日
智能卡缓存	NICE DCV 服务器现在可以缓存从客户端收到的智能卡数据，以帮助提高性能。有关更多信息，请参阅 https://docs.amazonaws.cn/dcv/latest/adminguide/manage-smart-card.html 管理员指南 中的 NICE DCV 配置智能卡缓存。	2018 年 10 月 8 日
Linux 客户端	NICE DCV 为 RHEL 7、CentOS 7、SLES 12 和 Ubuntu 16.04/18.04 提供了 Linux 客户端。有关更多信息，请参阅 https://docs.amazonaws.cn/dcv/latest/userguide/client-linux.html 用户指南 中的 NICE DCV Linux 客户端。	2018 年 8 月 29 日
更新了“参数参考”	“参数参考”已更新。有关更多信息，请参阅 NICE DCV 服务器 管理员指南 中的 参数参考 NICE DCV。	2018 年 8 月 7 日
USB 远程控制	NICE DCV 使客户端可以使用专门的 USB 设备，如 3D 指点设备或绘图板。有关更多信息，请参阅 管理员指南 中的 启用 USB 远程处理 NICE DCV。	2018 年 8 月 7 日
NICE DCV 的初始版本。	此内容的第一版。	2018 年 6 月 5 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。