



leostream

Remote Desktop Access Platform

Leostream and Third-Party Display Protocols

Connecting Leostream users to desktops with client-based display protocols

Version 202x
May 2023

Contacting Leostream

Leostream Corporation
77 Sleeper Street
PMB 02-123
Boston, MA 02210
USA

<http://www.leostream.com>
Telephone: +1 781 890 2019

To submit an enhancement request, email features@leostream.com.

To request product information or inquire about our future direction, email sales@leostream.com.

Copyright

© Copyright 2002-2023 by Leostream Corporation

This software program and documentation are copyrighted by Leostream. The software described in this document is provided under a license agreement and may be used or copied only under the terms of this agreement. No part of this manual may be copied or reproduced in any form without prior written consent from Leostream.

Trademarks

The following are trademarks of Leostream Corporation.

Leostream™

The Leostream graphical logo™

The absence of a product name or logo from this list does not constitute a waiver of the trademark or other intellectual property rights concerning that product, name, or logo by Leostream.

HP is a registered trademark that belong to Hewlett-Packard Development Company, L.P. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. UNIX is a registered trademark of The Open Group. OpenLDAP is a trademark of The OpenLDAP Foundation. Microsoft, Active Directory, SQL Server, Excel, Hyper-V, Windows, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Other brand and product names are trademarks or registered trademarks of their respective holders. Leostream claims no right to use of these marks.

Patents

Leostream software is protected by U.S. Patent 8,417,796.

Contents

| | |
|---|-----------|
| CONTENTS | 3 |
| SUPPORTED DISPLAY PROTOCOLS | 5 |
| CHOOSING A DISPLAY PROTOCOL | 5 |
| CONFIGURING DISPLAY PROTOCOLS IN LEOSTREAM | 7 |
| USING PROTOCOL PLANS | 7 |
| <i>How Protocol Plans Work</i> | 7 |
| <i>Building Protocol Plans</i> | 9 |
| SPECIFYING CONFIGURATION FILES AND COMMAND LINE ARGUMENTS | 10 |
| <i>Using Dynamic Tags in Configuration Files</i> | 10 |
| <i>Example: Using Different Login Names for User Connections</i> | 14 |
| <i>Example: Specifying Subnet for Desktop Connections</i> | 14 |
| <i>Dynamic Remapping of Desktop IP Address</i> | 15 |
| <i>Setting Configuration File Parameters Based on Client IP</i> | 16 |
| HP® ZCENTRAL REMOTE BOOST (RGS) | 17 |
| HP ZCENTRAL REMOTE BOOST (RGS) PROTOCOL PLAN OPTIONS | 17 |
| <i>Launching HP ZCentral Remote Boost Connections through the Leostream Gateway</i> | 18 |
| <i>Multi-Monitor Support with HP ZCentral Remote Boost</i> | 18 |
| <i>Activating HP Velocity and Advanced Video Compression Features</i> | 19 |
| <i>Setting User Configurable HP ZCentral Remote Boost Parameters</i> | 20 |
| SINGLE SIGN-ON WITH HP ZCENTRAL REMOTE BOOST | 22 |
| USB PASSTHROUGH WITH HP ZCENTRAL REMOTE BOOST | 22 |
| SESSION SHADOWING AND COLLABORATION | 23 |
| USING THE HP ZCENTRAL REMOTE BOOST RECEIVER FOR MACOS | 23 |
| MICROSOFT® RDP AND REMOTEFX | 24 |
| <i>Options for Encoding Desktop Login Credentials into RDP Configuration Files</i> | 24 |
| <i>Launching RDP Connections from the Leostream Web client</i> | 24 |
| <i>Configuring RDP for Low Bandwidth Connections</i> | 25 |
| <i>Microsoft RDP Viewer Command Line Parameters</i> | 26 |
| <i>Microsoft RDP Viewer Configuration File Variables</i> | 26 |
| <i>Connecting to RemoteApp Servers</i> | 32 |
| <i>Integrating with a Microsoft Remote Desktop Gateway</i> | 33 |
| MECHDYNE TGX | 34 |
| LAUNCHING MECHDYNE TGX CONNECTIONS | 34 |
| SETTING USER-CONFIGURABLE TGX PARAMETERS | 35 |
| NICE DCV | 37 |
| SPECIFYING SESSION IDS IN NICE DCV CONFIGURATION FILES | 37 |
| LAUNCHING NICE DCV CONSOLE CONNECTIONS | 38 |
| LAUNCHING NICE DCV VIRTUAL SESSIONS | 39 |
| USING THE NICE DCV HTML5 VIEWER | 40 |
| USING THE DCV EXTERNAL AUTHENTICATOR | 41 |
| <i>Configuring DCV Servers to use the External Authenticator</i> | 41 |
| <i>Configuring Protocol Plans when Using the External Authenticator</i> | 42 |

| | |
|--|-----------|
| NOMACHINE | 43 |
| LAUNCHING THE NOMACHINE CLIENT | 43 |
| LAUNCHING NOMACHINE HTML5 CONNECTIONS FROM THE WEB CLIENT | 43 |
| NOMACHINE CONFIGURATION FILE | 44 |
| SESSION SHADOWING AND COLLABORATION | 45 |
| SETTING USER-CONFIGURABLE NOMACHINE PARAMETERS | 45 |
| PCOIP® TECHNOLOGY | 48 |
| USING PCOIP CLIENTS WITH LEOSTREAM | 48 |
| ENABLING PCOIP CONNECTION MANAGEMENT IN LEOSTREAM | 50 |
| PCOIP CONNECTIONS TO VMWARE VIRTUAL MACHINES WITH A VIEW DIRECT-CONNECTION PLUG-IN | 50 |
| <i>Establishing Connections using Leostream Connect</i> | 50 |
| <i>Establishing Connections using the Leostream Web Client</i> | 51 |
| <i>Establishing Connections using a PCoIP Zero Client</i> | 52 |
| PENGUIN COMPUTING® SCYLD CLOUD WORKSTATION™ | 53 |
| LAUNCHING SCYLD CLOUD WORKSTATION CLIENTS FROM LEOSTREAM CONNECT | 53 |
| LAUNCHING SCYLD CLOUD WORKSTATION CLIENTS FROM THE WEB CLIENT | 54 |
| LAUNCHING THE SCYLD CLOUD WORKSTATION HTML5 VIEWER | 55 |
| RDESKTOP RDP REMOTE VIEWER | 56 |
| SCALE LOGIC REMOTE ACCESS PORTAL - VDI | 57 |
| LAUNCHING RAP - VDI CLIENTS FROM LEOSTREAM CONNECT | 57 |
| LAUNCHING RAP - VDI CLIENTS FROM THE WEB CLIENT | 58 |
| LAUNCHING THE RAP - VDI HTML5 VIEWER | 59 |
| VNC REMOTE VIEWER | 60 |
| <i>Setting up the Connection Broker to Use VNC</i> | 60 |
| <i>VNC Command Line Parameters</i> | 61 |
| SESSION SHADOWING AND COLLABORATION | 64 |
| CONFIGURING COLLABORATION IN THE CONNECTION BROKER | 64 |
| <i>Limiting Collaboration to Groups of Users</i> | 65 |
| <i>Sending Email Notifications Related to Collaboration Invitations</i> | 65 |
| WORKING WITH INVITATIONS IN THE LEOSTREAM WEB CLIENT | 66 |
| <i>Sending a Collaboration Invitation</i> | 66 |
| <i>Cancelling an Invitation</i> | 68 |
| <i>Accepting a Collaboration Invitation</i> | 68 |
| WORKING WITH INVITATIONS USING LEOSTREAM CONNECT | 69 |
| <i>Sending a Collaboration Invitation</i> | 69 |
| <i>Viewing and Cancelling Invitations</i> | 70 |
| <i>Accepting a Collaboration Invitation</i> | 70 |
| MANAGING INVITATIONS IN THE CONNECTION BROKER | 70 |
| USER CONFIGURABLE PROTOCOL PLAN PARAMETERS | 72 |
| DEFINING SCOPE OF THE CONFIGURED PARAMETER | 72 |
| END-USER INTERFACE FOR CONFIGURING PARAMETERS | 73 |
| <i>Leostream Web Client</i> | 73 |
| <i>Leostream Connect</i> | 74 |
| SETTING GLOBAL USER-CONFIGURABLE PARAMETERS | 74 |

Supported Display Protocols

The Leostream Connection Broker supports a wide range of display protocols that allow you to provide the required end-user experience throughout your entire organization. The Leostream Connection Broker can launch desktop connections using the following display protocols.

- HP ZCentral Remote Boost (RGS)
- Leostream HTML5-based RDP, VNC, and SSH (covered in the [Leostream Gateway guide](#))
- Microsoft RDP and RemoteFX (including FreeRDP, xrdp, and rdesktop clients)
- Mechdyne TGX
- NICE DCV
- NoMachine
- Teradici PCoIP (HP Anyware and Remote Workstation Cards)
- Scale Logic RAP VDI
- Penguin Computing Scyld Cloud Workstation
- VNC (RealVNC, TigerVNC, TightVNC, and UltraVNC)

Connection Broker protocol plans define which display protocols are used and how the remote session is launched. Defining protocol plans is covered in [Configuring Display Protocols in Leostream](#).

Before you build your protocol plans, you must choose the display protocols you will use in your environment. The next chapter provides general guidelines when considering different display protocols.

Choosing a Display Protocol

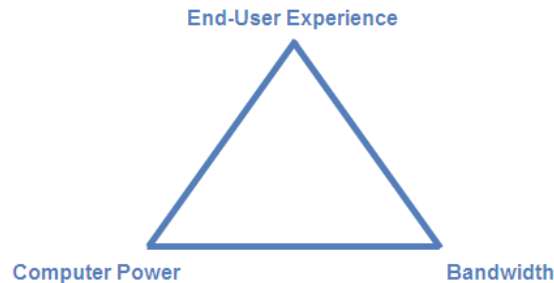
Leostream establishes a connection to a remote desktop using a variety of supported display protocols. After the connection is established, the Connection Broker is no longer in the data path of the user's desktop connection. For remote users, however, the Leostream Gateway may be in the data path of the user's connection.

The performance and requirements of the remote session are determined by the display protocol you select. This chapter provides some food-for-thought when investigating and choosing from the available display protocols.

Choosing the right protocol requires a balance between good end-user experience, the bandwidth available on the network, and the compute power supplied by the hardware. Every display protocol balances these requirements, with the ultimate goal being:

- Low bandwidth requirement
- Low computational requirements
- High-quality end-user experience

These three factors make up the *protocol triangle*, depicted in the following figure. As with any triangle, changing the angle for one corner has repercussions for the other angles.



You can achieve any two of your display-protocol goals, but will likely have to compromise on the third. For example, if your users accept a less performant viewing experience, you can choose a protocol that requires lower bandwidth and requires lower computing power. However, if you must provide a high-performance viewing experience, you need either higher bandwidth, higher computing power, or ideally both.

Each available display protocol handles the corners of the protocol triangle differently; each has its benefits and its drawbacks. When picking one or more display protocols, determine which protocol characteristics you need, and which trade-offs you can accept.

The following questions may help you define your display protocol requirements. Outline as many requirements as possible, and test different display protocols in your environment before committing to a particular protocol. Leostream can leverage multiple protocols in a single environment, so do not feel like you must commit to a single display protocol.

- What are your end-user requirements for multi-media, USB device redirection, response time, etc.?
- Do you have different types of users, for example task workers that run word processing applications and power users running graphic-intensive applications?
- What operating systems are you planning to deliver on your remote desktops or use on your client devices? Are you planning to support BYOD? If so, make sure your chosen display protocol handles all possible client device types.
- Do you want to use Zero or thin clients? If you are using Zero clients, which display protocol does it natively support? If using thin clients, what operating system does it use and can you install additional client software?
- Are your users accessing an entire desktop or only an application?
- Is single sign-on a requirement, or just nice-to-have?
- Do you need a display protocol that supports collaboration, where two users are simultaneously logged into the same session?
- How large will your deployment grow? (High computing power requirements affect scalability.)
- Do users connect to workstation with a GPU or are you using a virtual environment that supports GPU passthrough or vGPU?

Configuring Display Protocols in Leostream

Using Protocol Plans


Connection Broker protocol plans define which display protocol the Connection Broker uses when connecting a user to their desktop. Protocol plans define the order in which the Connection Broker tries to use the available protocols when connecting to a desktop and the configuration file or command line parameters used for the connection.

The Connection Broker provides one default protocol plan, which is shown on the **> Configuration > Protocol Plans** page, shown in the following figure.

| Actions | Name ▲ | In Use | Leostream API Protocols | Web Browser Protocols |
|---------|-------------------------------|--------|-------------------------|-----------------------|
| Edit | Default | Yes | RDP | RDP |
| Edit | HTML5 RDP / Leostream Gateway | Yes | RDP | Leostream HTML5 |
| Edit | Teradici via Gateway | Yes | | |

3 rows

Each protocol plan defines the display protocol used when the user logs in using different client types, such as Leostream Connect and thin clients, the Leostream Web client, and PCoIP clients. You configure the display protocol for each of these client types separately, using the appropriate section in the protocol plan.

 Your Leostream license determines which display protocols are included in your Connection Broker. Please, contact sales@leostream.com if you need access to a display protocol that is not currently listed in your Connection Broker.

How Protocol Plans Work

Protocol plans give you the flexibility to configure which display protocol to use for each pool used in a policy. A protocol plan tells the Connection Broker:

- Which display protocols are allowed for this pool
- What priority each protocol has, i.e., which protocol should the Connection Broker try first, second, etc.
- What, if any, command line parameters and configuration file should the Connection Broker use when establishing the connection

The following figure shows a portion of the **Leostream Connect and Thin Clients Writing to Leostream API** section of a protocol plan.

The screenshot shows the 'Create Protocol Plan' interface. At the top, there is a 'Plan name' input field. Below it, a section titled 'Leostream Connect and Thin Clients Writing to Leostream API' is expanded. This section includes an 'RDP' protocol with a 'Priority' dropdown set to '1'. It also has a 'Command line parameters' input field and a 'Configuration file' text area containing the following text:


```
screen mode id:i:2
desktopwidth:i:1024
desktopheight:i:768
desktopbpp:i:32
```


 Below this, another section titled 'HP ZCentral Remote Boost (RGS)' is visible, with a 'Priority' dropdown set to '2'. This section includes 'Send user login name as' (with '{USER}' in the input field), 'Send user password as' (with '{PLAIN_PASSWORD}' in the input field), and a 'Configuration file' text area containing:


```
Rgreceiver.IsBordersEnabled=0
Rgreceiver.IsBordersEnabled.IsMutable=0
Rgreceiver.IsMatchReceiverResolutionEnabled=1
Rgreceiver.IsMatchReceiverResolutionEnabled.IsMutable=0
```

The selection in the **Priority** drop-down menu indicates the order in which the Connection Broker checks if the remote desktop supports a particular display protocol. The Connection Broker performs a port check on the remote desktop to determine if it supports particular display protocol. For example, by default, Microsoft RDP communicates over port 3389. For the above example, if port 3389 is open on the remote desktop, the Connection Broker connects to the desktop using RDP. If port 3389 is not open, the Connection Broker checks the default HP ZCentral Remote Boost Sender port 42966.

For this example, if the HP ZCentral Remote Boost port is also closed, the Connection Broker looks for a protocol with a **Priority** of 3. If the **Priority** drop-down menu for all other display protocols is set to **Do not use**, the Connection Broker returns a warning that it cannot establish a connection to the remote desktop.

Where applicable, select a Leostream Gateway from the **Gateway** drop-down menu to indicate if the display protocol traffic is tunneled through a Leostream Gateway.

 The Connection Broker cannot distinguish between sections of the protocol plan that use the same port, for example Microsoft RDP and rdesktop. Therefore, if a protocol plan sets the priority for Microsoft

RDP to 1 and the priority of rdesktop to 2, the Connection Broker always uses the Microsoft RDP section of the Protocol Plan if port 3389 is open on the remote desktop, even if you are connecting from a Linux client that supports only rdesktop. For this example, you need a second protocol plan that assigns a priority of 1 to rdesktop, to support users logging in from a Linux client.

The Connection Broker does not consider the client device's capabilities when choosing the display protocol. For example, if the protocol plan sets the priority of HP ZCentral Remote Boost to 1 and the port check passes on the remote desktop, the Connection Broker instructs the client device to launch a Remote Boost connection, even if the client does not have an installed Remote Boost Receiver.

After the Connection Broker has selected the display protocol, it uses the **Configuration file** and **Command line parameters** to define how to launch the software client associated with that protocol. For example, for Microsoft RDP the `mstsc.exe` client is launched using the RDP-file parameters entered in the **Configuration file**. If the file or parameters contain any dynamic tags, the Connection Broker replaces those tags with the appropriate information before handing the file or command line parameters to the client device.

Building Protocol Plans

To determine how many protocol plans you need and how they should be configured, think about all the different ways your end users will connect to their desktops. Things to consider include:

- Do all users access their desktops using the same display protocol? If not, which protocols will they use? If these protocols communicate over the same port, you will need a protocol plan for each protocol.
- For each display protocol that you use, will the command line parameters and configuration file be the same for all users? If not, you will need a protocol plan for each configuration.
- Do your remote desktops support multiple protocols, such as RDP, TGX, and VNC? If so, and different users will launch connections with different protocols, you need a protocol plan that defines the appropriate priorities for each remote viewer.

After you define the different protocols you want to use and how you want to launch them, you create protocol plans as follows.

1. Go to the > **Configuration > Protocol Plans** page.
2. Click the **Create Protocol Plan** at the top of the page. The **Create Protocol Plan** form opens.
3. In the **Plan name** edit field, enter the name to use when referring to this protocol plan.
4. In the **Leostream Connect and Thin Clients Writing to Leostream API** section, shown in the previous figure, configure the protocols to use when a user logs in using one of the following client devices:
 - The Windows or Java version of Leostream Connect, installed on a Windows, Linux, or

- macOS device.
 - A thin client with an installed Leostream Connect client
 - A thin client with a customized Leostream client
5. In the **Web Browser** section, configure the protocol to use when a user logs in through the Leostream Web client. For in-browser RDP, VNC, or SSH connections using the Leostream Gateway, set the Priority for the **Leostream HTML5 Viewer** to **1** and then indicate which display protocol to use.
 6. Configure the **PCoIP Client Configuration** section of the protocol plan if your end users log in using a PCoIP software, mobile, or zero client.
 7. Use the **Notes** field to store any additional information with your protocol plan.
 8. Click **Save** to store any changes to the plan.

Specifying Configuration Files and Command Line Arguments

Configuration files and command line parameters allow you to customize the remote session. The format and contents of these fields differs for each display protocol. The following chapters discuss each display protocol and provide example syntax. The remainder of this chapter discusses Connection Broker concepts pertaining to using dynamic tags in a configuration file or command line parameter

Using Dynamic Tags in Configuration Files

Configuration files and command line parameters allow you to customize how the display protocol's software client establishes the desktop connection, for example, if the connection opens in full screen or windowed mode. The Connection Broker supports dynamic tags in the **Command line parameters** and **Configuration file** fields for any of the protocol. Before passing the command line parameters or configuration file to the client device, the Connection Broker replaces dynamic tags with the appropriate information, allowing you to reuse protocol plans for multiple users.

The following table contains a complete list of the supported dynamic tags. If the configuration file contains text enclosed in braces that is not included in the list of supported dynamic tags, the Connection Broker ignores the tag and it remains in the configuration file.

| Dynamic Tags | Purpose |
|----------------|---|
| { IP } | The IP address of the Leostream Agent on the desktop. If no Leostream Agent is installed on the desktop, { IP } is replaced with the hostname of the desktop or, if the hostname is not available, the IP address of the desktop. |
| { IP_ADDRESS } | The IP address of the desktop. |

| Dynamic Tags | Purpose |
|--|---|
| {IP_PRIVATE} | For cloud-hosted desktops, the internal IP address seen by the operating system. |
| {IP_PUBLIC} | For cloud-hosted desktops, the external IP address, if allocated, that is accessible from the outside network. |
| {IP_PRIVATE-or-IP_PUBLIC} | The private IP address of a cloud-hosted desktop or, if no private IP address exists, the public IP address of the desktop. |
| {IP_PUBLIC-or-IP_PRIVATE} | The public IP address of a cloud-hosted desktop or, if no public IP address exists, the private IP address of the desktop. |
| {IP_AGENT} | The Leostream Agent hostname or IP address. (If not available, {IP_ADDRESS} is returned.) |
| {HOSTNAME} | The hostname of the desktop. |
| {HOSTNAME_PRIVATE} | For desktops hosted in AWS, the instance's local hostname, as returned by the Leostream Agent |
| {HOSTNAME_PUBLIC} | For desktops hosted in AWS, the instance's public hostname, as returned by the Leostream Agent |
| {IP_ADDRESS-or-HOSTNAME} | The IP address of the desktop or, if the IP address is not available, the hostname of the desktop. |
| {HOSTNAME-or-IP_ADDRESS} | The hostname of the desktop or, if the hostname is not available, the IP address of the desktop. |
| {SHORT_HOSTNAME} | The short hostname of the desktop, or the hostname cut at the first dot. For example, if the hostname is <code>desktop.example.com</code> , the {SHORT_HOSTNAME} tag returns <code>desktop</code> . |
| {MACHINE_NAME} | The internal host name of the desktop, as returned by the Leostream Agent. Empty if no Leostream Agent is installed on the desktop. |
| {DCV_PORT}, {VNC_PORT} | For DCV and VNC connections, the port for the VNC session, as returned by the Leostream Agent. |
| {SESSION_ID_NAME} | For DCV connections, a unique session ID to pass to the Leostream Agent for starting the DCV session. Enter this dynamic tag for the <code>sessionid</code> parameter in the DCV configuration file. |
| {USER}, {USER:USER}, {USER:LOGIN_NAME}, or {LOGIN) NAME} | The user's login name. This value corresponds to the value shown in the Login name column on the > Resources > Users page. To force the login name on the remote desktop to upper or lower case, include the <code>:lowercase</code> or <code>:uppercase</code> modifier, for example <code>{USER:lowercase}</code> or <code>{USER:LOGIN_NAME:uppercase}</code> . |

| Dynamic Tags | Purpose |
|-----------------------------------|--|
| {AD:USER: <i>attribute_name</i> } | The value found in the user's Active Directory attribute given by <i>attribute_name</i> . Use this dynamic tag if you need to replace the user's login name for their remote session with a value different from the login name used for their Leostream session. See "Using Dynamic Tags" in the Connection Broker Administrator's Guide . |
| {NAME} or {USER:NAME} | The user's display name, corresponding to the value shown in the Name column on the > Resources > Users page. |
| {AD_DN} or {USER:AD_DN} | The user's Active Directory Distinguished Name. This value corresponds to the value shown in the AD Distinguished Name column on the > Resources > Users page. |
| {EMAIL} or {USER:EMAIL} | The user's email address. This value corresponds to the value shown in the Email column on the > Resources > Users page. |
| {PRE_EMAIL} or {USER:PRE_EMAIL} | The portion of the user's email address before the @ symbol. |
| {POST_EMAIL} or {USER:POST_EMAIL} | The portion of the user's email address after the @ symbol. |
| {DOMAIN} | The name entered into the Domain field for the authentication server that authenticated a user. If the Domain field is empty, the Connection Broker replaces this dynamic tag with the value entered or selected in the Domain field of the user's client. |
| {AUTH_DOMAIN} | The name entered in the Authentication server name field of the authentication server that authenticated the current user. |
| {PLAIN_PASSWORD} | The user's password, in plain text |
| {RDP_PASSWORD} | For Leostream Connect, the user's password encrypted for RDP usage |
| {SCRAMBLED_PASSWORD} | For NoMachine, only, the user's password scrambled to prevent casual eavesdropping. |
| {CREDENTIALS_MECHDYNE} | Encrypted user credentials to pass to the TGX Sender to provide single sign-on. |
| {STANDARD_RDP_PASSWORD:xxxx} | For Leostream Connect, a specific password encrypted for RDP usage |
| {PCOIP_HOST1} or {PCOIP_HOST2} | The last know IP address of the Teradici PCoIP Remote Workstation Card associated with the desktop for the connection. If the Connection Broker does not have an IP address for the card, then the dynamic tag is replaced with the card's hostname. |
| {CLIENT:IP} | The IP address of the user's client device used to log into the Connection Broker. This value corresponds to the value shown in the IP Address column on the > Resources > Clients page. |

| Dynamic Tags | Purpose |
|---------------------------|---|
| {CLIENT:MAC} | The MAC address of the user's client device used to log into the Connection Broker. This value corresponds to the value shown in the MAC Address column on the > Resources > Clients page. |
| {CLIENT:TYPE} | The type of client used to log into the Connection Broker. This value corresponds to the value shown in the Type column on the > Resources > Clients page. |
| {CLIENT:MANUFACTURER} | The manufacturer of client used to log into the Connection Broker. This value corresponds to the value shown in the Manufacturer column on the > Resources > Clients page. |
| {CLIENT:UUID} | The UUID of the client used to log into the Connection Broker. This value corresponds to the value shown for the Client UUID on the Edit Client page. |
| {POOL:NAME} | The name of the pool that contains the desktop that the user is connecting to |
| {VM:NAME} | The name of the desktop the user is connecting to, as shown in the Name field on the > Resources > Desktops page. |
| {WINDOWS_NAME} | The guest host name of a desktop, as returned by the Leostream Agent |
| {FQDN} | If the user authenticated against an authentication server, the fully qualified name, e.g., cn=Fred,ou=Users,o=Company |
| {DRIVE:CD} | For the RDP configuration file, use <code>drivestoredirect:s:{DRIVE:CD}</code> to redirect all CD drives found on system. No other drives are directed. |
| {DRIVE:DVD} | For the RDP configuration file, use <code>drivestoredirect:s:{DRIVE:DVD}</code> to redirect all DVD drives found on system. No other drives are directed. |
| {LOGOUT_URL} | The URL to log the user out of the session. |
| {LIST_URL} | The URL to view the list of desktops. |
| {ENV:*} | The value of the client side variable specified in *. So {ENV: HTTP_COOKIE} might return uid=25157202. |
| {REMAPPED_IP:X.X.X.X} | Re-maps IP addresses by replacing the non-X portion of the IP address with the specified tag. |
| {REMAPPED_IP:subnet_mask} | Re-maps IP addresses on different subnets. |
| {SESSION} | For use with the Java version of Leostream Connect. The session ID associated with session-based HP ZCentral Remote Boost Receiver configuration file parameters. |
| {USB_SESSION} | Indicates that the Java version of Leostream Connect should manage which remote HP ZCentral Remote Boost session has access to USB devices. |

Example: Using Different Login Names for User Connections

In some cases, you may need to use a login name for the user's remote desktop that is different from the login name used for the Leostream session. One example is the case where the user logs into Leostream with their Windows Active Directory credential, but needs to use their Linux username to connect to their Linux desktop. For these cases, you can use custom Active Directory attributes and dynamic tags to change the default user login.

First, you must populate an Active Directory attributes in the user's account with the value of the user's alternate login name. The Active Directory attribute can be a standard attribute, or you can create a custom attribute. For example, create a custom attribute named `linuxLogin`.

Second, in the protocol plan, replace the `{USER}` dynamic tag with the `{AD:USER:attribute_name}` dynamic tag. For example, when using the custom attribute named `linuxLogin` the dynamic tag is `{AD:USER:linuxLogin}`.

If the username varies only by case, you can use the `lowercase` and `uppercase` dynamic tag modifiers, instead of specifying a new Active Directory attribute. For example, if the user's Windows login is `JSmith`, but their Linux login is `jsmith`, use the `{USER:lowercase}` dynamic tag.

Example: Specifying Subnet for Desktop Connections

When a remote desktop has multiple network interfaces, the Leostream Agent and Connection Broker negotiate which IP address to use for remote connections. You can alternatively use the `{MATCHED_IP}` dynamic tag to specify a preferred IP address for the Connection Broker to use when establishing the remote connection. For example, you can modify the default line in the RDP configuration file to the following:

```
full address:s:{MATCHED_IP:partial_IP_address}
```

Where `partial_IP_address` indicates the beginning of the IP address that the Connection Broker should favor for the connection. When specifying `partial_IP_address`, trailing zeros are optional, for example, `{MATCHED_IP:172.29.0.0}` is equivalent to `{MATCHED_IP:172.29}`.

The `MATCHED_IP` dynamic tag instructs the Connection Broker to favor a specific IP address. For example, if the desktop returns two IP addresses of `172.29.229.151` and `10.110.1.14` and the tag is `{MATCHED_IP:10.110.1}` the IP address used for the connection is `10.110.1.14`.

If the desktop does not have an IP address beginning with the values to match, the Connection Broker will not establish a remote connection to the desktop. To allow the Connection Broker to fail over to any available IP address, use the following syntax:

```
{MATCHED_IP:partial_IP_address-or-IP}
```

For example, if the tag is `{MATCHED_IP:10.110.1-or-IP}` and the desktop returned a single IP address of `172.29.229.151` the Connection Broker uses the `172.29.229.151` for the connection even though it does not match the preferred IP address.

Dynamic Remapping of Desktop IP Address

You can enable display protocol traffic to traverse one or more NAT firewalls by dynamically changing the IP address provided to the display protocol's client to reflect the address of the desktop seen from the client's perspective as opposed to that seen by the desktop.

To do this, use the `{REMAPPED_IP}` dynamic tag in place of the `{IP}` dynamic tag. The Connection Broker applies the IP address mask specified in the dynamic tag to the IP address of the desktop, so that the address is modified.

As an example, imagine an offshore development center that runs on a 192.168.1.xxx network. One of its customers has a series of desktops running on a 172.29.229.xxx network. A NAT firewall makes the transition between the two networks. Therefore, a desktop at 172.29.229.131 appears to the offshore development center as a desktop at 192.168.1.131.

To accomplish this transition, in the configuration file, change instances of the `{IP}` tag to `{REMAPPED_IP:192.168.1.X}`.

To remap IP addresses on multiple subnets, use the advanced form of the `{REMAPPED_IP}` dynamic tag. This version of the `{REMAPPED_IP}` dynamic tag specifies a network mask length and a target range for the source and destination.

Use the `{REMAPPED_IP:X.X.X.X}` syntax to perform DNS resolution without remapping the IP address.

Use the wildcard (*) to map all subnets. For example:

- `{REMAPPED_IP:*/24->192.168.1.0}` replaces the first 24 bits of the IP address on all subnets with 192.168.1. Therefore, the IP address 10.153.172.5 maps to 192.168.1.5.
- `{REMAPPED_IP:*/8->194.0.0.0}` replaces the first 8 bits of the IP address on all subnets with 194. Therefore, the IP address 10.153.174.9 maps to 194.153.174.9.

To map different subnets to different IP address ranges, use the syntax in the following example.

```
{REMAPPED_IP:10.153.174.0/24 -> 192.168.204.0, 10.153.172.0/24 ->
192.168.201.0}
```

Each subnet map is separated by a comma. A subnet map can be defined using a wildcard, as described in the earlier `{REMAPPED_IP}` examples.

In this example, the first 24 bits of IP addresses in the subnet 10.153.174 are mapped to 192.168.204, while the first 24 bits of the IP addresses in the subnet 10.153.172 are mapped to 192.168.201. Therefore:

```
10.153.174.9 maps to 192.168.204.9
10.153.172.5 maps to 192.168.201.5
10.153.173.7 remains 10.153.173.7
```

In cases where multiple subnet maps are included, the order of the maps is irrelevant. The more specific map takes precedence over the less specific map. When a wildcard is provided, any IP addresses that are not mapped by one of the other rules will be mapped by the wildcard. The Connection Broker always performs wildcard mappings last.



Do not specify multiple wildcard mappings. If multiple wildcards are specified, the Connection Broker uses one of the mappings and ignores all other maps.

Setting Configuration File Parameters Based on Client IP

You can use the client's IP address to determine if a particular client configuration variable is sent to the client. You can also differentiate between users connecting locally and the same user connecting remotely.

To do this, wrap the configuration file parameter in a logic statement. These logic statements contain three parts. They can all be together, or on different lines. For example:

```
{NETWORK:10.0.0.0/255.255.255.0}
  compression:1
{END_NETWORK}
```

- `{NETWORK:10.0.0.0/255.255.255.0}` defines a network IP address range. The client IP address must fall within this range for the logic statement to be true.
- `compression:1` defines the configuration setting to be used if the logic statement is true. You can include multiple parameters spread across multiple lines.
- `{END_NETWORK}` closes the logic statement and must be present.

If you use multiple logic statements, the Connection Broker evaluates the statements in order and includes the parameter for the first statement that evaluates to true. For example:

```
{NETWORK:10.0.0.0/255.255.255.0} compression:0{END_NETWORK}
{NETWORK:192.168.10.0/255.255.255.128} compression:1{END_NETWORK}
{NETWORK:0.0.0.0/0.0.0.0}
  compression:0
{END_NETWORK}
```

In this example, if the client's IP address is 10.0.0.* they have data compression turned off. If the address is between 192.168.10.1 and 192.168.10.127 (VPN connected users), compression is turned on. If the address is anything else, compression is turned off.

HP® ZCentral Remote Boost (RGS)

HP® ZCentral Remote Boost (RGS) is a high-performance remote graphics system that renders the graphics on the desktop and sends the resulting screen image to the remote client.

HP ZCentral Remote Boost (RGS) Protocol Plan Options

The HP ZCentral Remote Boost (RGS) section of protocol plans, shown in the following figure, allows you to specify the user login name, password, and HP ZCentral Remote Boost Receiver parameters for the user's desktop connection. To ensure that the Connection Broker establishes an HP ZCentral Remote Boost connection, switch the **Priority** for HP ZCentral Remote Boost to 1, as shown for Leostream Connect logins in the following figure.

The screenshot shows the configuration interface for HP ZCentral Remote Boost (RGS). It includes the following elements:

- HP ZCentral Remote Boost (RGS)** section header.
- Priority:** A dropdown menu currently set to **1**.
- Send user login name as:** A text input field containing the dynamic tag `{USER}`.
- Send user password as:** A text input field containing the dynamic tag `{PLAIN_PASSWORD}`.
- Configuration file:** A text area containing configuration parameters:


```
Rgreceiver.IsBordersEnabled=0
Rgreceiver.IsBordersEnabled.IsMutable=0
Rgreceiver.IsMatchReceiverResolutionEnabled=1
Rgreceiver.IsMatchReceiverResolutionEnabled.IsMutable=0
```
- Gateway:** A dropdown menu with the text `Select ...`.
- Optional:** A checkbox labeled `Allow user to modify HP ZCentral Remote Boost (RGS) configuration file parameters`, which is currently unchecked.

The HP ZCentral Remote Boost Receiver requires a username and password to authorize the session, unless you enable Easy Login on the HP ZCentral Remote Boost Sender. Your Connection Broker does not know if it is connecting your user to a HP ZCentral Remote Boost Sender with Easy Login enabled, however, therefore the Connection Broker always sends a username and password to the client. By default, the Connection Broker sends the username and password used to log into Leostream. In some cases, you may need to send different credentials, for example:

- If the user logs into Leostream with their Active Directory credentials, but logs into a Linux desktop using a different username.
- If the user logs into their remote desktop using a generic account, not their account.

To satisfy these and other use cases, edit the **Send user login name as** edit field to launch the HP ZCentral Remote Boost session using a different login name than used for the Leostream session. You can use any of the dynamic tags associated with the user's account described in [Using Dynamic Tags in Configuration Files](#), including `{USER}`, `{EMAIL}`, `{AD:USER:attribute_name}`, etc.

If the password for the user specified in the **Send user login name as** edit field is different from the Leostream user's password, enter the new password in the **Send user password as** edit field.

Use the **Configuration file** field associated with HP ZCentral Remote Boost to specify the Remote Boost Receiver properties. The text you enter into the **Configuration file** field is analogous to the `rgreceiverconfig` file that sets Remote Boost Receiver parameters on the client computer when making native Remote Boost connections to a remote desktop.

The default configuration file is:

```
Rgreceiver.IsBordersEnabled=0
Rgreceiver.IsBordersEnabled.IsMutable=0
Rgreceiver.IsMatchReceiverResolutionEnabled=1
Rgreceiver.IsMatchReceiverResolutionEnabled.IsMutable=0
Rgreceiver.IsMatchReceiverPhysicalDisplaysEnabled=1
Rgreceiver.IsMatchReceiverPhysicalDisplaysEnabled.IsMutable=0
```

See the [HP ZCentral Remote Boost User Guide](#) for a complete description of the available Remote Boost Receiver properties. Every HP ZCentral Remote Boost Receiver installation provides a documented example `rgreceiverconfig` file in the installation directory, for example:

```
C:\Program Files\HP\Remote Graphics Receiver\rgreceiverconfig
```

The Connection Broker does not provide separate command line parameters for the Remote Boost connection. All command line parameters must be set using their configuration file equivalents.

Launching HP ZCentral Remote Boost Connections through the Leostream Gateway

The Leostream Gateway can proxy HP ZCentral Remote Boost connections, allowing you to establish Remote Boost connections from clients on different networks than the Remote Boost Sender. For a complete description of using the Leostream Gateway for HP ZCentral Remote Boost connections, consult the [Leostream Gateway Guide](#).

Multi-Monitor Support with HP ZCentral Remote Boost

The HP ZCentral Remote Boost Sender can automatically change the display settings on the remote desktop to match the monitor layout and resolution used on the client device running the Remote Boost Receiver.


The default configuration file in new protocol plans enables the following parameters:

```
IsMatchReceiverResolutionEnabled
IsMatchReceiverPhysicalDisplaysEnabled
```

to tell the HP ZCentral Remote Boost Sender to match the resolution and display layout of the client device. The configuration file also sets the `IsMutable` value for these parameters to 0, so the user's local Remote Boost Receiver does not override the protocol plan.

When the user establishes a connection, the HP ZCentral Remote Boost Sender attempts to match the

resolution and display layout. If the Remote Boost Sender cannot perform the match, it reverts to its previous resolution.

 When launching the connection using Leostream Connect, the user receives no warning that the HP ZCentral Remote Boost Sender is unable to match the desired resolution. Instead, the session opens using its last resolution without warning the user. Users who log into Leostream using the Leostream Web client receive a warning that their desired resolution cannot be matched.

Activating HP Velocity and Advanced Video Compression Features

The HP Velocity and Advanced Video Compression features improve HP ZCentral Remote Boost performance over WAN connections. You can utilize these new features when establishing Remote Boost connections using Leostream. The HP Velocity feature does not require additional configuration. To configure advanced video compression, include the following parameters in the HP ZCentral Remote Boost configuration file in your protocol plan.

- `Rgreceiver.ImageCodec.IsH264Enable`: Set to 1 to enable advanced video compression.
- `Rgreceiver.ImageCodec.IsCPUEncode`: Set to 1 to cause the Remote Boost Sender to use CPU encoding for h.264. If this parameter is set to zero, the Remote Boost Sender uses the GPU for encoding, if available.

The advanced video compression and HP Velocity functionality available in HP ZCentral Remote Boost require activation the first time the Remote Boost Receiver connects to the Remote Boost Sender. When connecting natively from the HP ZCentral Remote Boost Receiver to the Remote Boost Sender, activation dialogs open, indicating if the activation succeeded or failed. Leostream Connect suppresses the activation dialogs, however the activation continues to take place.

If you configured a proxy within Remote Boost to perform the activation, include the following three parameters in the HP ZCentral Remote Boost configuration file in your protocol plan.

- `Rgreceiver.Network.ProxyEnabled`: Set to 1 to enable the proxy, if required, in the environment
- `Rgreceiver.Network.ProxyPort`: Specify the proxy port
- `Rgreceiver.Network.ProxyAddress`: Specify the proxy hostname or IP address

HP ZCentral Remote Boost uses the system proxy settings, but only when manual proxy configuration is enabled. HP ZCentral Remote Boost does not support the use of use PAC, WPAD, or proxy authentication. If there is no internet access and no proxy possible, the Remote Boost session fails to activate and disables the HP Velocity and Advanced Video Compression features.

If the activation fails, use the following `Rgreceiver` parameters to configure the resultant behavior.

- `Rgreceiver.Activation.AutomationMode`: Specifies the path to take if the activation fails, either:

- 0 – Continue without activation: in this mode, the Remote Boost Receiver silently disables features requiring activation (HP Velocity and Advanced Video Compression) for the current session and continues with the connection. The next Remote Boost connection triggers activation again.
 - 1 – Retry the activation: in this mode, the Remote Boost Receiver retries activation before falling back. The number of retries is controlled by the `Rgreceiver.Activation.RetryAttempts` parameter.
 - 2 – (default) Do not activate: in this mode, the Remote Boost Receiver disables the features that require activation. On the next connection if the user has not re-enabled those features, no activation attempt will occur.
- `Rgreceiver.Activation.RetryAttempts`: (default = 5) The number of reactivation attempts before disabling features that require activation

Setting User Configurable HP ZCentral Remote Boost Parameters

The configuration file in the HP ZCentral Remote Boost section of the protocol plan defines the characteristics of the user's Remote Boost session. These settings may override any parameter settings made on the user's Remote Boost Receiver.


In some cases, you may want the user to customize certain Remote Boost connection parameters, including:

- Borders
- Resolution, including resolution and display layout
- Image quality
- Setup mode sequence

To allow users to set values for these parameters, select the **Allow users to modify configuration file parameters** option in the protocol plan.

To configure which parameters the user is allowed to modify:

1. Select the checkbox before each parameter that the user can customize.
2. After selecting the parameters the user can control, modify the text in the protocol plan's **Configuration file** field to use pre-defined dynamic tags, described in the following table, which the Connection Broker replaces at connection time with the values specified by the user.

 If you do not place the dynamic tags in the **Configuration file**, the user-specified settings will not be applied. Consult the HP ZCentral Remote Boost user's guide for more information on the proper syntax for configuring RGreceiver parameters.

| Parameter | RReceiver Parameter in the Protocol Plan | Leostream Dynamic Tag |
|---------------------|---|---|
| Setup mode sequence | .Hotkeys.SetupModeSequence | {SETUP_MODE_SEQUENCE} |
| Show borders | .IsBordersEnabled | {BORDERS} |
| Match resolution | .IsMatchReceiverResolutionEnabled | {MATCH_RESOLUTION} |
| Match layout | .IsMatchReceiverPhysicalDisplaysEnabled | {MATCH_DISPLAYS} |
| Resolution | .Session.{SESSION}.VirtualDisplay.IsPreferredResolutionEnabled .Session.{SESSION}.VirtualDisplay.PreferredResolutionHeight .Session.{SESSION}.VirtualDisplay.PreferredResolutionWidth | {RESOLUTION_ENABLED} {RESOLUTION_HEIGHT} {RESOLUTION_WIDTH} |
| Image quality | .ImageCodec.Quality | {IMAGE_QUALITY} |

For example, to allow the user to configure borders, image quality, and the setup mode sequence, you must add the following lines to your configuration file.

```
Rgreceiver.IsBordersEnabled={BORDERS}
Rgreceiver.ImageCodec.Quality={IMAGE_QUALITY}
Rgreceiver.Hotkeys.SetupModeSequence={SETUP_MODE_SEQUENCE}
```

The Connection Broker does not replace the {SESSION} dynamic tag. Instead, Leostream Connect automatically adjusts the value for the {SESSION} dynamic tag when the user connects to desktops using HP Remote Boost.

3. From the **Default value** drop-down menus, indicate the value to use if the user has not customized the parameter.
4. If you select **Custom** for the default value for resolution, enter the custom value into the **Default custom value** edit field. Enter the value as *heightxwidth* where *height* and *width* are in pixels and there is no space between the numbers and the x.
5. The drop-down menus in the end-user dialog display the values shown in the **Default value** drop-down menu on the Administrator interface. You can choose to show user-friendly descriptions of these items by defining display values. To define display values:
 - a. Click the **Edit** link in the **Display value** column
 - b. In the **Edit Display Values** form that opens, enter user-friendly names into the **Display value** edit field for each possible internal value.
 - c. Click **Save** on the **Edit Display Values** form. The new display values are shown in the **Default value** drop-down menu, as they will be displayed to users.

If the user never opts to customize values for the configurable protocol plan parameters, their connections

open using the default values specified in the protocol plan. If the user does specify a customized value for a parameter, the scope of that parameter is determined by the user's policy. See [User Configurable Protocol Plan Parameters](#) for information on how to define the scope of user-configurable parameters, as well as for instructions on using the end-user interfaces to define parameter values.

Single Sign-On with HP ZCentral Remote Boost

The Leostream Agent is not responsible for signing the user onto the remote desktop. Instead, to achieve single sign-on with HP ZCentral Remote Boost:

- When installing the Remote Boost Sender on a Windows desktop, ensure that the HP ZCentral Remote Boost Single Sign-on option is configured.
- For Linux remote desktop, single sign-on is not currently supported.

USB Passthrough with HP ZCentral Remote Boost

To connect USB devices to the remote Windows desktop, use either the HP USB redirector or Leostream Connect USB management. For predictable behavior, do not use these two features, simultaneously. If you use Leostream Connect USB management, you cannot use the **Assign to active desktop** USB option in the **When Device is Plugged In** section (see [USB Device Management](#)).

When using the Java version of Leostream Connect and the HP USB redirector, you can use the Leostream Connect sidebar to select which active remote session has access to all USB devices.


To turn on the sidebar for USB access:

1. Enable the sidebar by adding the following line to the `lc.conf` file on the client device.

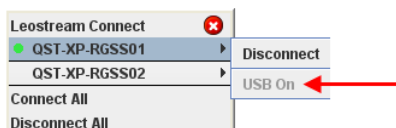
```
sidebar_enabled = true
```

2. In the protocol plan assigned to users that connect to desktops using HP ZCentral Remote Boost, add the following line to the **Configuration file** field for HP ZCentral Remote Boost.

```
Rgreceiver.Usb.ActiveSession={USB_SESSION}
```

 The Windows version of Leostream Connect does not support the `{USB_SESSION}` tag.

When a user logs in through Leostream Connect, by default, the first desktop they connect to using Remote Boost has access to all USB devices. The sidebar menu for this desktop, shown in the following figure, displays a **USB On** menu item.



When you attach a USB device to your client device, the USB device appears in the remote desktop that indicates **USB On**. You can switch all USB devices to another desktop by selecting the **Turn USB On** menu associated with that desktop, as shown in the following figure.



You must be connected to the desktop using Remote Boost before you can connect USB devices. Leostream Connect prompts you to confirm that all USB devices should be switched to the new desktop. Click **Yes** in the confirmation dialog, shown in the following figure to move USB devices to the new desktop. Click **No** to keep the USB devices attached to the current desktop.



HP ZCentral Remote Boost simultaneously allows access to USB devices from a single desktop.



If you disconnect from the Remote Boost session that has access to USB devices, Leostream Connect automatically switches all USB devices to the next active Remote Boost session.

Session Shadowing and Collaboration

The Connection Broker allows users that connect to desktops using the HP ZCentral Remote Boost protocol to collaborate by inviting a second *shadow* user to connect to their session. You must configure the HP ZCentral Remote Boost Sender to accept collaborators before using Leostream to send invitations

For information on enabling collaboration in the Connection Broker and sending invitations, see [Session Shadowing and Collaboration](#).

Using the HP ZCentral Remote Boost Receiver for macOS

You can install the Java version of Leostream Connect on an Apple macOS device and use the HP ZCentral Remote Boost Receiver for Mac to establish Remote Boost connections. See the [Leostream Installation Guide](#) for information on installing Leostream Connect on a Mac.

Leostream Connect does not automatically discover the location of the HP ZCentral Remote Boost Receiver app. Specify the full path to the HP Remote Boost Receiver executable in the **Viewers** tab of the Leostream Connect **Options** dialog. Do not enter the path to the `.app` directory, for example:

```
/Applications/HP RGS Receiver.app/Contents/MacOS/HP RGS Receiver
```

Microsoft® RDP and RemoteFX

The **RDP and RemoteFX** section of the protocol plan allows you to enter command line parameters and/or a configuration file to use when launching a Microsoft remote desktop connection. The Connection Broker uses the standard Microsoft RDP configuration file format for RDP sessions controlled by Leostream Connect. You can verify the configuration file you enter in the **Configuration file** edit field of the protocol plan using a standard Microsoft RDP client.

Options for Encoding Desktop Login Credentials into RDP Configuration Files

RDP requires an encrypted password in order to perform single sign-on. Typically, the Configuration file for RDP contains the following line:

```
password 51:b:{RDP_PASSWORD}
```

Using the `{RDP_PASSWORD}` dynamic tag in the protocol plan encodes the user's desktop login credentials into the RDP configuration file. The Connection Broker replaces the `{RDP_PASSWORD}` dynamic tag with the user's password encrypted for RDP connections before passing the RDP configuration file to the client.



Do not use the `password 51:b` parameter in the configuration file. The Web client cannot encrypt the user's password.

If the user's desktop requires a different password than what the user provided to Leostream Connect, you can use the `{STANDARD_RDP_PASSWORD:password}` dynamic tag to pass the desktop password down to the Leostream Connect client in order to enable single sign-on. In your configuration file, replace `password` with the password to log into the desktop. Leostream Connect then encrypts the password and places the encrypted password in the configuration file before launching the RDP connection.

Launching RDP Connections from the Leostream Web client

When launching an RDP connection from the Leostream Web client, the Connection Broker downloads an RDP configuration file in the same way as any other server-initiated file download. If your Web browser blocks the download, modify the browser's security settings to allow downloads from your Connection Broker. After the browser downloads the file, it prompts the user to open or save the file. Opening the file launches the RDP session, where the user must enter their password.



Single sign-on is not available when using the native RDP client from a Web browser.

Some Web browsers prompt users to download the RDP file used to launch the Connection to the desktop. To avoid this prompt, the first time the Connection Broker tries to download an RDP file, right-click on the download tab associated with that file and select **Always open this kind of file**. When the user subsequently launches additional desktops, the Web browser automatically launches the connection without prompting the user.

Ensure that you do not save credentials from the client device's native RDP client. If a user previously saved credential by checking the **Allow me to save credentials** option on the Remote Desktop Connection login page, ensure that you delete the credential before trying to log into that remote desktop from the Leostream Web client.

Configuring RDP for Low Bandwidth Connections

When connecting to a remote desktop using Microsoft RDP, the settings on the **Experience** tab of the Remote Desktop Connection software client influence the quality of the user's experience based on the bandwidth of their connection. The default RDP Configuration file in Leostream Protocol Plans is configured to maximize the user experience, which may require a higher bandwidth connection. You can modify the default parameters or create new plans that are better suited to groups of users with lower bandwidth connections

For example, the default value for the `connection type` parameter is set to 6, which is ideal for LAN connections with high bandwidth. To build a Protocol Plan for users with lower bandwidth, set the `connection type` parameter to 1 or 2, where:

1. A connection type of 1 [Modem (56 Kbps)] enables persistent bitmap caching and disables all other Experience features.
2. A connection type of 2 [Low Speed Broadband (256 Kbps - 2 Mbps)] enables persistent bitmap caching and visual styles (themes), but disables all other Experience features.

Additionally, you can manually disable the individual Experience features by setting their specific parameters in the Configuration File, for example:

```
connection type:i:1
disable wallpaper:i:1
allow font smoothing:i:0
allow desktop composition:i:0
disable full window drag:i:1
disable menu anims:i:1
disable themes:i:1
disable cursor setting:i:0
bitmapcachepersistenable:i:1
```

In addition to the Experience parameters, you can try disabling audio and reducing the color bit depth to improve the desktop connection over low bandwidth, for example:

```
audiomode:i:1
session bpp:i:16
```

For more information on these settings and tips on how to configure remote desktops for low bandwidth networks, see:

<https://learn.microsoft.com/en-us/windows-server/administration/performance-tuning/role/remote-desktop/session-hosts>

Microsoft RDP Viewer Command Line Parameters

The following is a list of some useful RDP command line parameters. For an online description all the RDP command line parameters, go to the following Microsoft Windows support page.

<http://windowshelp.microsoft.com/Windows/en-US/help/142d58b8-43f0-432f-93bb-7653333905911033.mspx>

/f

Start the RDP connection in full-screen mode.

/span

Use this parameter to span across multiple monitors with the same height and width.

/w:<width>

Specify the width of the RDP connection windows.

/h:<height>

Specify the height of the RDP connection window.

Microsoft RDP Viewer Configuration File Variables

The following is a list of the RDP file parameters contained in the default configuration file for new protocol plans. Where Connection Broker dynamic tags are included in the parameter name, ensure that you include the dynamic tag when using that parameter in the configuration file contained in the protocol plan.

use multimon:i: (RDP 7, only)

Indicates if the remote session should span across all monitors attached to the client device. When using this option, the monitors do not have to have the same resolution and orientation. If using RDP 6, use `span monitors`, instead.

| Value | Setting |
|-------|----------------------|
| 0 | Use a single monitor |
| 1 | Use all monitors |

span monitors:i

Indicates if the remote session should be spanned across multiple monitors.

| Value | Setting |
|-------|-----------------|
| 0 | Spanning is off |
| 1 | Spanning is on |

screen mode id:i

Determines if the remote session is opened in a window or in full screen.

| Value | Setting |
|-------|---------|
|-------|---------|

- | | |
|---|---------------------|
| 2 | Open in full screen |
| 1 | Open in a window |

desktopwidth:i

Corresponds to the desktop width (in pixels) on the **Display** tab in Remote Desktop Connection **Options** dialog.

desktopheight:i

Corresponds to the desktop height (in pixels) on the **Display** tab in Remote Desktop Connection **Options** dialog.

connection type:i

Corresponds to the selection in the **Choose your connection speed to optimize performance** drop-down on the **Experience** tab in Remote Desktop Connection **Options** dialog. To invoke RemoteFX, set this value to 6, and set the `session bpp` parameter to 32.

session bpp:i

Corresponds to the color depth you select in the **Colors** drop-down on the **Display** tab in Remote Desktop Connection **Options** dialog. To invoke RemoteFX, set this value to 32, and set the `connection type` parameter to 6.

winposstr:s

Corresponds to the window position on the **Display** tab in Remote Desktop Connection **Options** dialog.

On desktop computers, this setting determines the Remote Desktop Connection dialog box position on the screen. The six numbers represent a string form of the `WINDOWPOS` structure. For more information about the `WINDOWPOS` function, visit the following Microsoft Web page:

<http://msdn.microsoft.com/en-us/library/ms632612.aspx>

full address:s {IP}

Determines the IP address of desktop. The setting corresponds to the entry in the **Computer** field on the **General** tab of Remote Desktop Connection **Options** dialog. The Connection Broker can dynamically set this property using the `{IP}` or the `{Windows_Name}` dynamic tag.

password 51:b: {RDP_PASSWORD}

Controls password settings.

RDP requires an encrypted password to perform single sign-on. The Connection Broker passes as unencrypted password to the client device; the client device is then responsible for password encryption. Using the `{RDP_PASSWORD}` tag, the Windows version of Leostream Connection encrypts the password and places the encrypted version into the configuration file, resulting in single sign-on to the desktop.



The Java version of Leostream Connect and the Connection Broker Web client cannot encrypt the RDP password. Use the plain password option when using RDP to connect to a desktop from the Java version of Leostream Connect or the Web client.

password:s: {PLAIN_PASSWORD}

Controls password settings.

In this case, the Connection Broker sends a plain-text password to the client device. Use this option if launching Microsoft RDP connections from the Java version of Leostream Connect, the Leostream Web client, or thin clients from vendors such as HP that write to the Leostream API.

compression:i

Determines if data is compressed when it is transmitted to the client computer, according to the following values

| Value | Setting |
|-------|--------------------|
| 0 | Compression is off |
| 1 | Compression is on |

keyboardhook:i

Determines where Windows key combinations are applied. This setting corresponds to the selection in the **Keyboard** section on the **Local Resources** tab of Remote Desktop Connection **Options** dialog, according to the following values.

| Value | Setting |
|-------|--------------------------|
| 0 | On the local computer |
| 1 | On the remote computer |
| 2 | In full-screen mode only |

audiomode:i

Determines where sounds are played. This setting corresponds to the selection in the **Remote computer sound** section on the **Local Resources** tab of Remote Desktop Connection **Options** dialog, according to the following values.

| Value | Setting |
|-------|--------------------------|
| 0 | Bring to this computer |
| 1 | Leave at remote computer |
| 2 | Do not play |

redirectclipboard:i:l

Determines if the clipboard is enabled in the remote session. This setting corresponds to the selection of the **Clipboard** option in the **Local devices and resources** section on the **Local Resources** tab of Remote Desktop Connection **Options** dialog, according to the following rules.

| Value | Setting |
|-------|--------------------------|
| 0 | Clipboard is not enabled |
| 1 | Clipboard is enabled |

redirectdrives:i:0

Determines if disk drives are automatically connected when you log on to the remote desktop. This setting corresponds to the selection of the **Drives** option in the **More Local devices and resources** dialog, accessed via the **More** button in the **Local Resources** tab of Remote Desktop Connection **Options** dialog. The following values apply.

| Value | Setting |
|-------|--|
| 0 | Drives are not automatically reconnected |
| 1 | All drives are automatically reconnected |

drivestoredirect:s

Determines which drives are automatically connected when you log on to the remote desktop. Use `drivestoredirect:s:*` to redirect all existing drives and any subsequently connected drives. To redirect a specific drive, enter the drive name followed by a colon, for example, `drivestoredirect:s:C:`. To redirect multiple drives, use a semi-colon to separate the drive names. Use the `DynamicDrives` tag to redirect drives that are connected to the client after the remote session is established. For example, the following parameter redirects the C drive and dynamic drives:
`drivestoredirect:s:C;DynamicDrives`

The Windows version of Leostream Connect supports the following two dynamic tags when connecting using RDP 6. These two dynamic tags are supported for RDP 7 *only* when the RDP 7 client is installed on a Windows XP operating system and the drive is referenced as the volume label followed by the drive letter. Leostream Connect cannot redirect drives using RDP 7 if the drives are referenced by the drive label followed by the drive letter, or by a combination of drive label, drive letter, and volume label.

| Value | Setting |
|-------------|--|
| {DRIVE:DVD} | All DVD drives are automatically connected. No other drives are connected. |
| {DRIVE:CD} | All CD drives are automatically connected. No other drives are connected |

devicestoredirect:s

Determines which supported Plug and Play devices on the client computer are automatically redirected when you log on to the remote desktop. Use `devicestoredirect:s:*` to redirect all supported Plug and Play devices. To redirect any supported Plug and Play devices that are connected later, use the `DynamicDevices` tag, for example, `devicestoredirect:s:DynamicDevices`.

redirectposdevices:i:0

Determines whether media players based on the Media Transfer Protocol (MTP) and digital cameras based on the Picture Transfer Protocol (PTP) are redirected. This setting corresponds to the **Supported Plug and Play devices** option in the **More Local devices and resources** dialog, accessed via the **More** button in the **Local Resources** tab of Remote Desktop Connection **Options** dialog. The following values apply.

| Value | Setting |
|-------|---|
| 0 | Microsoft Point of Service for .NET (POS for .NET) device redirection is disabled |
| 1 | Microsoft Point of Service for .NET (POS for .NET) device redirection is enabled |

redirectprinters:i

Determines whether printers are automatically connected when you log on to the remote computer. This setting corresponds to the selection in the **Printers** check box in the **Local devices and resources** section on the **Local Resources** tab of Remote Desktop Connection **Options** dialog. The following values apply.

| Value | Setting |
|--------------|--|
| 0 | Printers are not automatically reconnected |
| 1 | Printers are automatically reconnected |

redirectcomports:i

Determines if COM ports are automatically connected when you log on to the remote computer. This setting corresponds to the **Serial Ports** option in the **More Local devices and resources** dialog, accessed via the **More** button in the **Local Resources** tab of Remote Desktop Connection **Options** dialog. The following values apply.

| Value | Setting |
|--------------|---|
| 0 | COM ports are not automatically reconnected |
| 1 | COM ports are automatically reconnected |

redirectsmartcards:l

Determines if smart cards are automatically connected when you log on to the remote computer. This setting corresponds to the **Smart cards** box in the **More Local devices and resources** dialog, accessed via the **More** button on the **Local Resources** tab of Remote Desktop Connection **Options** dialog. The following values apply.

| Value | Setting |
|--------------|---|
| 0 | Smart cards are not automatically reconnected |
| 1 | Smart cards are automatically reconnected |

displayconnectionbar:i

Determines whether the connection bar is displayed when you log on to the remote computer in full-screen mode. This setting corresponds to the selection in the **Display the connection bar when in full screen mode** option on the **Display** tab of Remote Desktop Connection **Options** dialog. The following values apply.

| Value | Setting |
|--------------|--------------------------------|
| 0 | Connection bar does not appear |
| 1 | Connection bar appear |

username:s; {USER}

Determines the user account used to log into the desktop. This setting corresponds to the entry in the **User name** edit field on the **General** tab of Remote Desktop Connection **Options** dialog. The Connection Broker can dynamically set this property using the {USER} field.

domain:s {DOMAIN}

Determines the domain used to authenticate the user. This setting corresponds to the entry in the **Domain** edit field on the **General** tab of Remote Desktop Connection **Options** dialog. The Connection Broker can dynamically set this setting using the {DOMAIN} field.

alternate shell:s

Determines if a program is started automatically when you connect with RDP. The setting corresponds to the entry in the **Program path and file name** edit field on the **Programs** tab of Remote Desktop Connection **Options** dialog.

shell working directory:s

Indicates the starting folder for the application that is automatically started when you connect with RDP. The setting corresponds to the entry in the **Start in the following folder** edit field on the **Programs** tab of Remote Desktop Connection **Options** dialog.

disable wallpaper:i

Determines if the desktop background appears when you log on to the remote computer. This setting corresponds to the selection in the **Desktop background** check box on the **Experience** tab of Remote Desktop Connection **Options** dialog, according to the following values.

| Value | Setting |
|--------------|---------------------------|
| 0 | Wallpaper appears |
| 1 | Wallpaper does not appear |

disable full window drag:i

Determines if folder contents appear when you drag the folder to a new location. This setting corresponds to the selection in the **Show contents of window while dragging** check box on the **Experience** tab of Remote Desktop Connection **Options** dialog, according to the following values.

| Value | Setting |
|--------------|--|
| 0 | Folder contents appear while dragging |
| 1 | Folder contents do not appear while dragging |

disable menu anims:i

Determines how menus and windows appear when you log on to the remote computer. This setting corresponds to the selection in the **Menu and window animation** check box on the **Experience** tab of Remote Desktop Connection **Options** dialog, according to the following values.

| Value | Setting |
|--------------|--|
| 0 | Menu and window animations are permitted |
| 1 | Menu and window animations are not permitted |

disable themes:i

Determines if themes are permitted when you log on to the remote computer. This setting corresponds to the selection in the **Themes** check box on the **Experience** tab of Remote Desktop Connection **Options** dialog, according to the following values.

| Value | Setting |
|--------------|--------------------------|
| 0 | Themes are permitted |
| 1 | Themes are not permitted |

bitmapcachepersistenable:i

Determines if bitmaps are cached on the local computer. This setting corresponds to the selection in the **Bitmap caching** check box on the **Experience** tab of Remote Desktop Connection **Options** dialog, according to the following values.

| Value | Setting |
|-------|------------------------|
| 0 | Caching is not enabled |
| 1 | Caching is enabled |

Connecting to RemoteApp Servers

To offer the user RemoteApp sessions, first create a **Remote Desktop Services/Multi-User** center for the Windows Server that publishes the applications (see “Remote Desktop Services / Multi-User Centers” in the **Leostream Connection Broker Administrator’s Guide**). When creating the center, indicate the maximum number of users that can simultaneously log into the server.

Next, group the RemoteApp sessions created by the center into pools on the **> Configuration > Pools** page. You can publish all the applications from a single pool or separate the sessions into individual pools for each application. If you plan to offer multiple applications to a particular user, separate the sessions into multiple pools, by application.

Finally, if you are connecting directly to the RemoteApp server, use the following parameters in the RDP Configuration File to launch a particular application on the RemoteApp server. Each protocol plan publishes a single application. Create as many protocol plans as needed, based on the number of applications published on the server.



If you will connect to the RemoteApp server through the Leostream Gateway, see the Leostream Gateway Guide for information on how to build an appropriate protocol plan.

remoteapplicationmode:i

(Required) Determines whether a RemoteApp should be launched when connecting to the remote computer.

| Value | Setting |
|-------|--|
| 0 | Use a normal session and do not start a RemoteApp. |
| 1 | Connect and launch a RemoteApp. |

remoteapplicationname:s

(Required) Specifies the name of the RemoteApp in the Remote Desktop interface while starting the RemoteApp.

remoteapplicationprogram:s

(Required) Specifies the alias or executable name of the RemoteApp.

remoteapplicationcmdline:s

Optional command line parameters for the RemoteApp.

remoteapplicationfile:s

Specifies a file to be opened on the remote computer by the RemoteApp.

When building policies to offer RemoteApp sessions, associate each protocol plan with the appropriate pool of sessions. A Leostream policy can offer sessions from multiple pools, allowing you to publish multiple applications in one policy. However, a particular pool can only be referenced once, per policy.



By default, RemoteApp servers keep the user logged into the session even after the disconnect from their application. If you do not want to modify the group policy on your RemoteApp server, you can use Leostream release plans to log the users out of their disconnected sessions. See the “Release Plans” section in the [Leostream Connection Broker Administrator’s Guide](#) for more information.

Integrating with a Microsoft Remote Desktop Gateway

Use the following parameters in the RDP Configuration File to support connections to desktops via a Microsoft Remote Desktop Gateway.

gatewaycredentialssource:i

Specifies the credentials that should be used to validate the connection with the RD Gateway.

| Value | Setting |
|-------|----------------------------|
| 0 | Ask for password (NTLM) |
| 1 | Use smart card |
| 4 | Allow user to select later |

gatewayhostname:s

Specifies the hostname of the RD Gateway.

gatewayprofileusagemethod:i

Determines the RD Gateway authentication method to be used.

| Value | Setting |
|-------|--|
| 0 | Use the default profile mode, as specified by the administrator. |
| 1 | Use explicit settings. |

gatewayusagemethod:i

Specifies if and how to use a Remote Desktop Gateway (RD Gateway) server.

| Value | Setting |
|-------|---|
| 0 | Do not use an RD Gateway server |
| 1 | Always use an RD Gateway, even for local connections. |
| 2 | Use the RD Gateway if a direct connection cannot be made to the remote computer (i.e. bypass for local addresses) |
| 3 | Use the default RD Gateway settings |
| 4 | Do not use an RD Gateway server (default) |

Mechdyne TGX

Mechdyne TGX delivers high resolution without sacrificing image quality or impacting performance. End users can launch TGX connections from either the Leostream Connect client or using the Leostream Web client.

Launching Mechdyne TGX Connections

You must install the Mechdyne TGX client on any device that will connect to remote desktops using TGX. Leostream launches the TGX client using the TGX-file configured by the user's Leostream protocol plan. To configure a protocol plan to use TGX:

1. Scroll down to the **Mechdyne TGX** section of the protocol plan in the **Leostream Connect and Thin Clients Writing to Leostream API** or **Web Browser** sections.
2. Change the **Priority** of TGX to 1.
3. If any other protocol in the associated section of the protocol plan has a priority of 1, modify that protocol's priority to a lower number or to **Do not use**.
4. If your users do not have network access to the TGX Sender, you can use the Leostream Gateway to forward the TGX traffic. In this case, use the **Gateway** drop-down menu to select the Leostream Gateway that handles the TGX traffic for desktop connections associated with this protocol plan.

The Leostream Gateway forwards TCP and UDP ports 40001 for TGX 202x versions.

Consult the [Leostream Gateway guide](#) for more information on the ports used for forwarding traffic, as you may need to make adjustments to your load balancers or firewalls to accommodate the traffic.

5. Use the **Configuration file** field associated with TGX to customize the remote session. The configuration file specifies parameters in the `.tgx` file used to launch the Mechdyne client.

The default configuration file uses the `{CREDENTIALS_MECHDYNE}` dynamic tag. The Connection Broker uses a proprietary Mechdyne algorithm to encrypt the user's credentials to pass to the TGX Sender and provide single sign-on.

Setting User-Configurable TGX Parameters

The configuration file in the **Mechdyne TGX** section of the protocol plan defines the characteristics of the user's TGX session. The values in the protocol plan configuration file override any parameter settings made on the user's TGX client.

You can allow the user to configure the following TGX connection parameters:

- Image quality
- Client resolution
- Local mouse redirection
- Update rate


To allow users to set values for these parameters, select the **Allow users to modify configuration file parameters** option in the protocol plan. The form expands to include the additional fields shown in the following figure.

The screenshot shows the configuration interface for Mechdyne TGX. It includes a 'Priority' dropdown set to 1, a 'Configuration file' text area containing parameters like `removeFile=false`, `scaled=false`, `hostname={IP}`, and `credentials={CREDENTIALS_MECHDYNE}`. Below this is a 'Gateway' dropdown set to 'Select ...'. A checkbox labeled 'Allow user to modify TGX configuration file parameters' is checked. Below this checkbox is a table with columns for 'Parameter', 'Default value', 'Default custom value', and 'Display values'.

| Parameter | Default value | Default custom value | Display values |
|--|---------------|----------------------|----------------|
| <input type="checkbox"/> Image quality | 52 | | Edit |
| <input type="checkbox"/> Match my desktop | No | | Edit |
| <input type="checkbox"/> Mouse redirection | Local | | Edit |
| <input type="checkbox"/> Update rate | 48 | | Edit |

To configure which parameters the user is allowed to modify:

1. Select the checkbox before each parameter that the user can customize.
2. Modify the text in the protocol plan's **Configuration file** field to use the dynamic tags associated with the selected parameters, as described in the following table. The Connection Broker replaces the dynamic tags at connection time with the values specified by the user.

 If you do not place the dynamic tags in the **Configuration file**, the user-specified settings are not applied.

| Parameter | TGX Parameter in the Protocol Plan | Leostream Dynamic Tag |
|-------------------|------------------------------------|-----------------------|
| Image quality | imageQuality | {IMAGE_QUALITY} |
| Match my desktop | matchDesktop | {MATCH_DESKTOP} |
| Mouse redirection | showLocalMouse | {MOUSE_REDIRECTION} |
| Update rate | updateRate | {UPDATE_RATE} |

For example, if the user can configure all four parameters, the configuration file looks as follows.

```
removeFile=false
hostname={IP}
domain={DOMAIN}
credentials={CREDENTIALS_MECHDYNE}
matchDesktop={MATCH_DESKTOP}
showLocalMouse={MOUSE_REDIRECTION}
imageQuality={IMAGE_QUALITY}
updateRate={UPDATE_RATE}
```

3. Use the **Default value** drop-down menus to indicate the default value to use if the user does not customize the parameter.
4. The drop-down menus in the end-user dialog display the values shown in the **Default value** drop-down menu on the Administrator interface. You can choose to show user-friendly descriptions of these items by defining display values. To define display values:
 - a. Click the **Edit** link in the **Display value** column.
 - b. In the **Edit Display Values** form that opens, enter user-friendly names into the **Display value** edit field for each possible internal value.
 - c. Click **Save** on the **Edit Display Values** form. The new display values are shown in the **Default value** drop-down menu, as they will be displayed to users.

If the user does not customize values for the configurable protocol plan parameters, their connections open using the default values specified in the protocol plan. If the user does specify a customized value for a parameter, the scope of that parameter is determined by the user's policy. See **User Configurable Protocol Plan Parameters** for information on how to define the scope of user-configurable parameters, as well as for instructions on using the end-user interfaces to define parameter values.

NICE DCV

NICE DCV is an advanced technology that enables users to access 2-D and 3-D interactive applications over a standard network. End users can launch DCV connections when logging into Leostream using either the Leostream Connect client or the Leostream Web client.



To manage DCV connections in Leostream, configure your DCV server so they do *not* automatically launch DCV sessions on boot time. On Microsoft Windows DCV servers, ensure the create-session DWORD value is set to **0**. This registry key value is described in the “Enabling Automatic Console Sessions” section of the **NICE DCV Administrators Guide**.

The Connection Broker leverages the Leostream Agent on the NICE DCV desktop or server to launch a DCV session for the Leostream user at the time they request their connection. Leostream supports launching console connections to Microsoft Windows and Linux operating systems, as well as virtual DCV sessions for Linux NICE DCV servers.

Specifying Session IDs in NICE DCV Configuration Files

Leostream launches NICE DCV sessions by downloading the DCV configuration file associated with the user’s protocol plan. Before the file is downloaded, the Leostream Agent launches the NICE DCV session for the user and returns information to the Connection Broker about the user’s session, which the Connection Broker requires to replace the dynamic tags found in the DCV configuration file.

Starting with Leostream Agents 5.2.6 for Linux and 7.4.8 for Windows operating systems, Leostream changed the session ID used to launch DCV sessions.

- Prior to these agent versions, all DCV sessions were launched with a `leo- $\{USER\}$` session ID.
- To support special characters in usernames, newer Leostream Agent versions launch the DCV session as `{SESSION_ID_NAME}` where `{SESSION_ID_NAME}` a unique session ID to pass to the Leostream Agent for starting the DCV session. When upgrading your Leostream Agents, ensure that you enter the `{SESSION_ID_NAME}` dynamic tag for the `sessionid` parameter in the DCV configuration file.

The default DCV configuration file in protocol plans includes the new `sessionid` format used by the Leostream Agents.


Depending on the version of the Leostream Agent and Connection Broker you are using, your Protocol Plan configuration files must be configured differently to launch the DCV sessions, as described below.

1. When using Leostream Agent 5.2.6 for Linux operating systems or Leostream Agent 7.4.8 for Windows operating systems, set the `sessionid` line in the configuration file, as follows:

```
sessionid={SESSION_ID_NAME}
```

2. When using Leostream Agent 5.1.24 for Linux operating systems or Leostream Agent 7.3.13 for Windows operating systems, set the `sessionid` line in the configuration file, as follows.

```
sessionid=leo-{USER}
```

 Improperly defining the `sessionid` parameter in the protocol plan can result in running, but orphaned, DCV sessions and Leostream Gateway forwarding rules. Leostream recommends using Connection Broker 9.1 and upgrading all Leostream Agents to the latest version, when using DCV.

Launching NICE DCV Console Connections

To configure a protocol plan to launch NICE DCV console sessions to Microsoft Windows or Linux DCV servers:

1. Go to the **> Configuration > Protocol Plans** page and edit or create a new protocol plan.
2. Scroll down to the **DCV** section of the protocol plan in the **Leostream Connect and Thin Clients Writing to Leostream API** or **Web Browser** sections.
3. Change the **Priority** for NICE DCV to 1.
4. If any other protocol in the associated section of the protocol plan has a priority of 1, modify that protocol's priority to a lower number or to **Do not use**.
5. Use the **Configuration file** field associated with DCV to customize the remote session. The configuration file specifies parameters in the `.dcv` file used to launch the DCV client. (See [Specifying Session IDs in NICE DCV Configuration Files](#).)
6. Leave the **Launch virtual session for connection** option *unchecked* to instruct the Leostream Agent to launch a console session.
7. Optionally select the **Use DCV External authenticator with token** option to use the Connection Broker as an external authenticator for the DCV session, instead of passing a password in the DCV file (see [Using the DCV External Authenticator](#)).
8. If your users do not have network access to the DCV server, you can use the Leostream Gateway to forward the DCV traffic. In this case, use the **Gateway** drop-down menu to select the Leostream Gateway that handles the DCV traffic for the desktop connections associated with this protocol plan.

Launching NICE DCV Virtual Sessions

To configure Leostream to launch NICE DCV virtual sessions to Linux DCV servers, you must create a Multi-User Center for your Linux DCV server. You can create a Multi-User Center for any Linux DCV server with a running Leostream Agent that is inventoried in your Connection Broker. If the Leostream Agent on your DCV server has not registered with your Connection Broker, you cannot convert it to a Multi-User Center.

After your Linux DCV server and its Leostream Agent are listed as running on the **> Resources > Desktops** page, you can convert it to a Multi-User Center, as follows.

1. Go to the **> Setup > Centers** page.
2. Click **Add Center**.
3. From the **Type** drop-down menu, select **Remote Desktop Services/Multi-User**.
4. In the Name edit field, enter a descriptive name to use for the DCV sessions that will be displayed on the **> Resources > Desktops** page.
5. Select your DCV server from the **Select server to convert to a Remote Desktop Services/Multi-User Center** drop-down menu.
6. In the **Maximum concurrent connections/sessions**, enter the maximum number of simultaneous DCV sessions you want running on this server.
7. Click **Save**.

After you save the center, the Connection Broker creates placeholder DCV sessions on the **> Resources > Desktops** page. You can group these sessions into a pool to offer to users via a Leostream policies. When referencing this pool in your policies, assign a protocol plan that instructs the Connection Broker to launch virtual DCV session. You build a protocol plan that launches virtual DCV sessions, as follows.

1. Go to the **> Configuration > Protocol Plans** page and edit or create a new protocol plan.
2. Scroll down to the **DCV** section of the protocol plan in the **Leostream Connect and Thin Clients Writing to Leostream API** or **Web Browser** sections.
3. Change the **Priority** of NICE DCV to 1.
4. If any other protocol in the associated section of the protocol plan has a priority of 1, modify that protocol's priority to a lower number or to **Do not use**.
5. Use the **Configuration file** field associated with DCV to customize the remote session. The configuration file specifies parameters in the `.dcv` file used to launch the DCV client. The configuration file must contain the following line:

```
sessionid={SESSION_ID_NAME}
```

6. Select the **Launch virtual session for connection** option to instruct the Leostream Agent to start a virtual session on the DCV server.
7. Optionally select the **Use DCV External authenticator with token** option to use the Connection Broker as an external authenticator for the DCV session, instead of passing a password in the DCV file (see [Using the DCV External Authenticator](#)).
8. If your users do not have network access to the DCV server, you can use the Leostream Gateway to tunnel the DCV connection. In this case, use the **Gateway** drop-down menu to select the Leostream Gateway that manages the DCV traffic for desktop connections using this protocol plan.



When using the Leostream Gateway to handle DCV connections for virtual sessions, configure your Leostream Gateway to route desktop traffic using one of the options that filters based on the client source IP. Do *not* select the **From random gateway port to protocol-specific desktop port** option for the routing method in the Leostream Gateway, as this results in all DCV sessions routing through the same port on the Leostream Gateway and, therefore, closing one DCV connection causes all DCV connections to that server to close.

Using the NICE DCV HTML5 Viewer

NICE DCV includes a built-in HTML5 viewer for clientless connections. DCV HTML5 connections are available for any user logging in using the Leostream Web client. To configure a protocol plan to launch the DCV HTML5 Viewer:

1. Go to the > **Configuration > Protocol Plans** page and edit or create a new protocol plan.
2. Scroll down to the **DCV HTML5 Viewer** section of the protocol plan in **Web Browser** sections.
3. Change the **Priority** of the NICE DCV HTML5 Viewer to 1.
4. If any other protocol in this section of the protocol plan has a priority of 1, modify that protocol's priority to a lower number or to **Do not use**.
5. Select the **Launch virtual session for connection** option if you want the Leostream Agent to launch a virtual session instead of a console session.
6. If your users do not have network access to the DCV server, you can use the Leostream Gateway to tunnel the DCV connection. In this case, use the **Gateway** drop-down menu to select the Leostream Gateway that tunnels the DCV traffic for desktop connections using this protocol plan.

When using the Leostream Gateway, the DCV HTML5 traffic is forwarded through the gateway using the routing method selected in the **Method for routing display protocol traffic through this Leostream gateway** option of the selected Leostream Gateway. DCV traffic is not routed along port 443. Consult the [Leostream Gateway guide](#) for more information on the ports used for forwarding traffic, as you may need to make adjustments to your load balancers or firewalls to accommodate the traffic.

Using the DCV External Authenticator

The default configuration file for DCV connections includes the user's password as plain text, to use for authenticating the DCV connection. If users log into Leostream using the Leostream Connect client, Leostream Connect removes the DCV-file from the filesystem after the DCV session is established, minimizing the risk of password exposure.

For users logging into Leostream using the Leostream Web client, the DCV-file persists in the user's Downloads directory after the DCV connection is established. On shared client devices, this could expose users' passwords.

To avoid exposing passwords, you can leverage your Leostream Connection Broker as a DCV external authenticator. In this case, the DCV session is authenticated with a unique token and the password does not need to be included in the configuration file.

To use the Connection Broker as a DCV external authenticator, the RESTful API must be enabled on the Connection Broker and your DCV servers must to point to your Connection Broker for authentication.

Configuring DCV Servers to use the External Authenticator

After the RESTful API is enabled in your Leostream environment, the DCV external authenticator is located at:

```
https://ConnectionBrokerAddress/rest/dcv_auth
```

Where *ConnectionBrokerAddress* is your Leostream address. Note the URL does not contain any port information.

You must configure your DCV servers to use that URL for authentication.

- On Linux DCV servers, add the `auth-token-verifier` parameter in the `[security]` section of the `/etc/dcv/dcv.conf` file and set the value to your Connection Broker external authenticator URL.
- On Windows DCV servers, open the Windows Registry Editor and navigate to the **HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/security/** key. Add a String parameter named `auth-token-verifier` and set its value to your Connection Broker external authenticator URL.

Restart your DCV servers after making the change. For complete instructions, consult the [NICE DCV Administrator Guide](#).



If you use HTTPS in your Connection Broker external authenticator URL, the authenticator service must use a valid certificate trusted by DCV.

On Linux DCV servers, you can use the following configuration parameter in the `/etc/dcv/dcv.conf` file to specify the path to the certificate:

```
[security]
ca-file="/path/to/ca/file"
```

Alternatively, you can disable strict TLS communication by adding the following line to the `/etc/dcv/dcv.conf` file.

```
[security]
no-tls-strict=true
```

On Windows DCV servers, open the Windows Registry Editor and navigate to the **HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/security/** key. Add a String parameter named `ca-file` and set its value to the path to your CA-file.

Alternatively, for Windows DCV Servers, you can disable strict TLS communication in the Windows Registry Editor by navigating to the **HKEY_USERS/S-1-5-18/Software/GSettings/com/nicesoftware/dcv/security/** key, adding a Dword parameter named `no-tls-strict`, and setting its value to 1

Configuring Protocol Plans when Using the External Authenticator

Your Connection Broker uses the session ID in the Configuration File of the user's Protocol Plan to validate the authentication token returned by the DCV Agent. The session ID format differs depending on which version of the Leostream Agent you are running on the DCV Server. Ensure that you enter the correct value for the `sessionid` parameter in the Configuration File, as described below.



If the `sessionid` parameter is omitted from the Configuration File, the Leostream Agent starts the DCV session however user authorization fails and the DCV client is unable to connect to the session.

- Leostream Agent 7.4.8 for Windows operating systems
- Leostream Agent 5.2.6 for Linux operating systems

```
sessionid={SESSION_ID_NAME}
```

- Leostream Agent 7.3.13 for Windows operating systems
- Leostream Agent 5.1.24 for Linux operating systems

```
sessionid=leo-{USER}
```

NoMachine

The NoMachine section of the protocol plan allows you to specify a default configuration file for the NoMachine connection, as well as indicate any protocol plan parameters that are user configurable. End users can launch NoMachine connections from either the Leostream Connect client or using the Leostream Web client.

Launching the NoMachine Client

To configure a protocol plan to launch the NoMachine software client:

1. Scroll down to the **NoMachine** section of the protocol plan in the **Leostream Connect and Thin Clients Writing to Leostream API** or **Web Browser** sections.
2. Change the **Priority** of NoMachine to 1.
3. If any other protocol in the associated section of the protocol plan has a priority of 1, modify that protocol's priority to a lower number or to **Do not use**.
4. If your users do not have network access to the NoMachine server, you can use the Leostream Gateway to forward the NoMachine traffic. In this case, use the **Gateway** drop-down menu to select the Leostream Gateway that handles the NoMachine traffic for desktop connections associated with this protocol plan.

Consult the [Leostream Gateway guide](#) for more information on the ports used for forwarding traffic, as you may need to make adjustments to your load balancers or firewalls to accommodate the traffic.

5. Use the **Configuration file** field associated with NoMachine to customize the remote session. The configuration file specifies parameters in the `.nxs` file used to launch the NoMachine client.
6. If users are able to override the value for certain NoMachine parameters, select the **Allow user to modify configuration file parameters** option. See [Setting User-Configurable NoMachine Parameters](#) for a list of configurable parameters and instructions on setting up this feature.

Launching NoMachine HTML5 Connections from the Web Client

To setup a protocol plan to launch the NoMachine HTML5 client when a user logs into Leostream using the Leostream Web client:

1. Go to the **> Configuration > Protocol Plans** page.
2. Create a new protocol plan or edit an existing plan.

3. In the **Create Protocol Plan** or **Edit Protocol Plan** form, scroll down to the **NoMachine HTML5 Viewer** item in the **Web Browser** section.
4. Change the **Priority** of the NoMachine HTML5 Viewer to 1.
5. If any other protocol in the Web browser section of the protocol plan has a priority of 1, modify that protocol's priority to a lower number or to **Do not use**.
6. If your users do not have network access to the NoMachine server, you can use the Leostream Gateway to forward the NoMachine traffic. In this case, use the **Gateway** drop-down menu to select the Leostream Gateway that handles the NoMachine traffic for desktop connections associated with this protocol plan.

The Leostream Gateway forwards NoMachine HTML5 traffic along port 4443. Consult the [Leostream Gateway guide](#) for more information on the ports used for forwarding traffic, as you may need to make adjustments to your load balancers or firewalls to accommodate the traffic.

NoMachine Configuration File

The configuration file for NoMachine is an XML representation of the fields in the NoMachine Client GUI. The following table describes the important option keys when integrating with Leostream.

| Option Key | Purpose |
|-------------|--|
| User | (Default = {USER}) The user's login name. The Connection Broker replaces the dynamic tag {USER} with the name the user entered when logging in to the Connection Broker. |
| Auth | (Default = {SCRAMBLED_PASSWORD}) The user's password. By default, the configuration file is configured to pass a scrambled password. If your NX server is configured to expect a plain password, replace the {SCRAMBLED_PASSWORD} dynamic tag in the <code>configuration file</code> field with the {PLAIN_PASSWORD} dynamic tag. When the {SCRAMBLED_PASSWORD} is used, the Connection Broker uses the NoMachine method for scrambling passwords. This scrambled password is then passed in the configuration file. |
| Server host | (Default = {IP}) The Connection Broker replaces the dynamic tag {IP} with the hostname or IP address of the NoMachine server. |
| Public Key | Enter the key into the configuration file by placing an option key <code>Public Key</code> after the <code>Login Method</code> option, as follows. <pre> ... <option key="Login Method" value="nx" /> <option key="Public Key" value'" -----BEGIN DSA PRIVATE KEY----- < Insert DSA Key here> -----END DSA PRIVATE KEY----- />" ... </pre> |

Session Shadowing and Collaboration

The Connection Broker allows users that connect to desktops using the NoMachine protocol to collaborate by inviting a second *shadow* user to connect to their session. You must configure the NoMachine server to accept collaborators before using Leostream to send invitations

For information on enabling collaboration in the Connection Broker and sending invitations, see [Session Shadowing and Collaboration](#).

Setting User-Configurable NoMachine Parameters

The configuration file in the NoMachine section of the protocol plan defines how the user's NoMachine session is launched. These settings override any parameter settings made on the user's NoMachine software client.

In some cases, you may want the user to be able to customize certain NoMachine connection parameters. Currently, Leostream allows end-users to customize the following parameters.

- Connection type
- Desktop option
- Disable backingstore
- Resolution
- Span monitors
- Window manager

To allow users to set values for these parameters, select the **Allow users to modify NoMachine configuration file parameters** option in the protocol plan. The form expands to include the additional fields shown in the following figure.

NoMachine Priority:

Configuration file

```
<!DOCTYPE NXClientSettings>
<NXClientSettings version="1.5" application="nxclient" >
<group name="Login" >
<option key="User" value="{USER}" />
</group>
</NXClientSettings>
```

Gateway

Select ...

Optional

Allow user to modify NoMachine configuration file parameters

| Parameter | Default value | Default custom value | Display values |
|---|-----------------|----------------------|----------------|
| <input type="checkbox"/> Connection type | LAN | | Edit |
| <input type="checkbox"/> Desktop option | Virtual desktop | | Edit |
| <input type="checkbox"/> Disable backingstore | True | | Edit |
| <input type="checkbox"/> Resolution | Available area | | Edit |
| <input type="checkbox"/> Span monitors | No | | Edit |
| <input type="checkbox"/> Window manager | KDE | | Edit |

To configure which parameters the user is allowed to modify:

1. Select the checkbox before each parameter that the user can customize.
2. After selecting the parameters that the user can control, modify the text in the protocol plan's **Configuration file** field to use pre-defined dynamic tags, described in the following table, which the Connection Broker replaces at connection time with the values specified by the user.



You cannot save the Protocol Plan form if you do not include all relevant dynamic tags.

| Parameter | Option Key | Original Text | Replace with |
|----------------------|---|---|---|
| Connection type | Link speed | value="lan" | value="{CONNECTION_TYPE}" |
| Desktop option | Virtual desktop | value="false" | value="{VIRTUAL_DESKTOP}" |
| Disable backingstore | Disable backingstore | value="false" | value="{BACKINGSTORE}" |
| Resolution | Resolution Resolution height Resolution width | value="available" value="600" value="800" | value="{RESOLUTION}" value="{RESOLUTION_HEIGHT}" value="{RESOLUTION_WIDTH}" |
| Span Monitors | Spread over monitors | value="false" | value="{SPAN_MONITORS}" |
| Window manager | Desktop | value="kde" | value="{WM_TYPE}" |

3. From the **Default value** drop-down menus, indicate the value to use before the user customizes the parameter.
4. If you select **Custom** for the default value for window manager or resolution, enter the custom value into the **Default custom value** edit field. For resolution, enter the value as *heightxwidth* where *height* and *width* are in pixels and there is no space between the numbers and the *x*.
5. The drop-down menus in the end-user dialog include all values shown in the **Default value** drop-down menu on the Administrator interface. You can choose to show user-friendly descriptions of these items by defining display values. To define display values:
 - a. Click the **Edit** link in the **Display value** column
 - b. In the **Edit Display Values** form that opens, enter user-friendly names into the **Display value** edit field for each possible internal value.
 - c. Click **Save** on the **Edit Display Values** form. The new display values are shown in the **Default value** drop-down menu, as they will be displayed to users.
6. You must repeat steps 1 through 4 for the **Web browser** and **Leostream Connect** sections of the protocol plan, if users log in from both clients. The display values in step 5 can be specified only once. The Leostream Connect and Leostream Web clients use the same set of display values.

If the user never opts to customize values for the configurable protocol plan parameters, their connections open using the default values specified in the protocol plan. If the user does specify a customized value for a parameter, the scope of that parameter is determined by the user's policy. See [User Configurable Protocol Plan Parameters](#) for information on how to define the scope of user-configurable parameters, as well as for instructions on using the end-user interfaces to define parameter values.

PCoIP® Technology

Leostream can initiate PCoIP connections to the following types of remote desktops.

- Microsoft Windows, Linux, and macOS physical machines with installed or attached PCoIP Remote Workstation cards. See the [Quick Start with Teradici PC-over-IP](#) for details on setting up the Connection Broker to manage PCoIP connections to PCoIP Remote Workstation Cards.
- Microsoft Windows and Linux virtual machines with installed PCoIP Agent. See the [Quick Start Guide Using Leostream with HP Anyware](#) for complete details.
- VMware virtual machines with an installed VMware Horizon Direct-Connection Plug-In (see [PCoIP Connections to VMware Virtual Machines with a View Direct-Connection Plug-In.](#))

Users can connect to virtual machines running the VMware Horizon Direct-Connection Plug-In using the Leostream Web client, Leostream Connect, or a PCoIP Zero Client. In this workflow, a Leostream policy assigns the desktop to the user and, therefore, no VMware Horizon View Connection Server configuration is required.

Using PCoIP Clients with Leostream

You can use any supported PCoIP software, mobile, or zero client to log into Leostream. The type of client you use and whether the client communicates with Leostream or the PCoIP Connection Manager determines what types of PCoIP resources can be connected. The following table describes the types of resources users can connect to from different PCoIP client.

| Client Type | Client Points To | The client can connect to: Virtual Machines | The client can connect to: Physical Machines |
|-----------------------|-------------------------------------|--|---|
| PCoIP Software Client | PCoIP Connection Manager | Running the Cloud Access Software PCoIP Standard or Graphics Agent | With installed PCoIP Remote Workstation cards if the operating system has an installed PCoIP Agent for Remote Workstation Cards |
| PCoIP Mobile Client | Security Gateway <i>Disabled</i> | | And |
| PCoIP Zero Client | | | Running the Cloud Access Software PCoIP Standard or Graphics Agent. |

| Client Type | Client Points To | The client can connect to: Virtual Machines | The client can connect to: Physical Machines |
|--|---|--|---|
| PCoIP Software Client (macOS, Windows, ChromeOS) PCoIP Mobile Client PCoIP Zero Client | PCoIP Connection Manager Security Gateway <i>Enabled</i> | Running the Cloud Access Software PCoIP Standard or Graphics Agent | Running the Cloud Access Software PCoIP Standard or Graphics Agent |
| PCoIP Zero Client | Leostream Connection Broker | Running the VMware Horizon View Direct Connection Plug-In | With an installed PCoIP Remote Workstation Cards (no PCoIP RWC Agent installed) |
| PCoIP Zero Client PCoIP Software Client - Windows | Leostream Gateway, forwarding to the Connection Broker | Not currently supported | With an installed PCoIP Remote Workstation Cards (no PCoIP RWC Agent installed) |
| Leostream Connect and PCoIP Software Client (Windows only) | Leostream Connection Broker Or Leostream Gateway, forwarding to the Connection Broker | Not currently supported | With an installed PCoIP Remote Workstation Cards (no PCoIP RWC Agent installed) |

Enabling PCoIP Connection Management in Leostream


Your Leostream license determines if you can manage PCoIP Connections. Contact sales@leostream.com if you need to enable PCoIP for your environment.

After you apply a license that enables the PCoIP feature, reboot your Connection Broker.

- The > **Resources** page contains a new **PCoIP Host Devices** section. The > **Resources > PCoIP Host Devices** page lists the PCoIP Remote Workstation cards registered with your Connection Broker.
- The > **Setup > Centers** page contains a new **PCoIP Devices** center. This center instructs the Connection Broker on how often to refresh the information associated with your PCoIP devices.

PCoIP Connections to VMware Virtual Machines with a View Direct-Connection Plug-In

The Connection Broker can establish PCoIP connections to VMware virtual machines running the VMware Horizon View Direct-Connection Plug-in. The virtual machine must have an installed Leostream Agent.

 When installing the Leostream Agent, ensure that you *do not* select the **Credential Provider** task when performing the installation. The Leostream Agent credential provider may conflict with the Direct-Connection Plug-in.

Ensure that the PCoIP connection can be established from the VMware Horizon View Client to the virtual machine, before attempting to use with Leostream. You must configure the **View Agent Direct-Connection Users** on the virtual machine before Leostream can establish the PCoIP connection.

Users can connect to the desktop using the Leostream Web client, Leostream Connect, or a PCoIP zero client.

Establishing Connections using Leostream Connect

When using Leostream Connect or the Leostream Web client, the user's client device must have an installed VMware Horizon View client. You can then use Leostream protocol plans to launch the VMware client and establish a PCoIP connection to a Windows virtual machine running the VMware View Direct-Connection Plugin.

To configure the protocol plan for software-based PCoIP connections:

1. Go to the > **Configuration > Protocol Plans** page.
2. Create a new protocol plan or edit an existing plan.

3. In the **Leostream Connection and Thin Clients Writing to Leostream API** section, select **1** from the **Priority** menu associated with **VMware View**.
4. Also in the **Leostream Connection and Thin Clients Writing to Leostream API** section, select **Do not use** or set lower priority to all other protocols.
5. In the **Command line parameters** edit field, enter the command line parameters needed to connect the user with single sign-on.

The default parameters, shown below, launch the Windows version of the VMware View client.

```
-nonInteractive -serverURL {IP} -userName {USER} -password
{PLAIN_PASSWORD} -domainName {DOMAIN} -desktopName {VM:NAME} -
desktopProtocol PCOIP
```

The Linux version of the VMware View client requires different parameter. If your users are logging in from a Linux client device, modify the command line parameters, as follows;

```
--nonInteractive --serverURL {IP} --userName {USER} --password
{PLAIN_PASSWORD} --domainName {DOMAIN} --desktopName {VM:NAME} --protocol
PCOIP
```



If you have users logging in from Windows and Linux devices, create two protocol plans and assign the appropriate plan based on the user's location. See "Assigning Plans to Locations" in the [Connection Broker Administrator's Guide](#) for more information.

6. In the **Port for remote viewer check** specify the port number that the Connection Broker pings to determine if the desktop is available for PCoIP connections.
7. Click **Save**.

After you create your protocol plans, build a pool that contains only these virtual machines running the VMware View Direct-Connection Plugin. When creating a policy that uses this pool, ensure that you select the protocol plan that uses the VMware View client.

Establishing Connections using the Leostream Web Client

The Leostream Web client uses the VMware Horizon View client URI to launch a PCoIP connection to the desktop. To configure the Connection Broker to support PCoIP connections to virtual machines:

1. Create a pool of virtual machines with a running VMware Horizon View Agent Direct-Connection Plug-In.
2. Create a protocol plan to assign to these virtual machines. In the **Web Browser** section of the protocol plan:
 - a. Set the **Priority** of the **External viewer** to **1**.

- b. Set the **Priority** of all other protocols to **Do not use**.
- c. In the **Configuration file** for the external viewer, enter:

```
vmware-view://{HOSTNAME}/{VM:NAME}?desktopProtocol=PCoIP
```

3. Build a policy that assigns the protocol plan from step 2 to the pool of virtual machines created in step 1.
4. Assign the policy to the user.

When a user who is assigned this policy logs into the Connection Broker, the broker offers the user a virtual machine from the pool. When the user requests a connection to the virtual machine, the Connection Broker launches the VMware Horizon View client, which establishes the PCoIP connection to the desktop.



The VMware Horizon View client URI does not support single sign-on.

Establishing Connections using a PCoIP Zero Client

When using a PCoIP zero client to connect to virtual machines, the Connection Broker ignores the protocol plan selected in the user's policy and, instead, always establishes a PCoIP connection when the virtual machine has a running VMware Horizon View Direct-Connection Plugin.

Penguin Computing© Scyld Cloud Workstation™

Scyld Cloud Workstation is a high-performance remote workstation solution designed to deliver real-time interactive enterprise-class visualization through a standard browser or from an installed client. Leostream can connect users to Scyld Cloud Workstations using either the Scyld Cloud Workstation client or using the Scyld Cloud Workstation HTML5 viewer.

The Scyld Cloud Workstation client is available for Microsoft Windows, Apple macOS, and Linux operating systems. End users can launch the Scyld Cloud Workstation client when logging into Leostream from either the Leostream Web client or one of the Leostream Connect clients.

Launching Scyld Cloud Workstation Clients from Leostream Connect

To configure a protocol plan to launch the Scyld Cloud Workstation software client from a Leostream Connect login:

1. Scroll down to the **Scyld Cloud Workstation** section of the protocol plan in the **Leostream Connect and Thin Clients Writing to Leostream API** section.
2. Change the **Priority** of Scyld Cloud Workstation to 1.
3. If any other protocol in the associated section of the protocol plan has a priority of 1, modify that protocol's priority to a lower number or to **Do not use**.
4. In the **Command line parameters**, enter the default command line parameters to use when launching the Scyld Cloud Workstation client. The default parameters are defined using the Windows format (/). If this protocol plan will be assigned to users logging in from a Linux or macOS client device, replace the forward slashes with a double dash (--).

For a full list of supported command line parameters, install the Scyld Cloud Workstation client and run the client from the command line with the `/help (--help)` parameter.



If you plan to enable the Leostream Gateway in this protocol plan, ensure that you place the string `https://` before the `{IP}` tag in the Command line parameters, for example:

```
/server=https://{IP} /user={USER} /password={PLAIN_PASSWORD}
```

5. If your users do not have network access to the Scyld Cloud Workstation server, you can use the Leostream Gateway to forward the display protocol traffic. In this case, use the **Gateway** drop-down menu to select the Leostream Gateway that handles the Scyld Cloud Workstation traffic for desktop connections associated with this protocol plan.



Scyld Cloud Workstation traffic is always along port 443. If you are using your Leostream Gateways to forward login traffic to your Connection Broker, ensure that your Leostream Gateways are configured in your Connection Broker to forward traffic along a random port, preferably filtered

by Client IP for additional security, for example:

Edit Gateway "Leostream Gateway" ?

Name

Add this Leostream Gateway to a Gateway Cluster
[None available]

Public IP address or FQDN for use in Protocol Plans

If this Gateway is located behind a load balancer or external firewall, enter its public IP address or FQDN. This address must be accessible from the user's client device.

IP address or FQDN used for Connection Broker communications to this Gateway

Unique IP address or FQDN of this Leostream Gateway. Not required if same as Public IP address above.

Method for routing display protocol traffic through this Leostream gateway

If you are required to use the protocol-specific port 443 for the Scyld Cloud Workstation traffic and you use your Leostream Gateways to forward Connection Broker login traffic, you must configure your Leostream Gateway to use a port other than 443. For instructions on changing the default Leostream Gateway port, see the [Leostream Gateway Guide](#).

Launching Scyld Cloud Workstation Clients from the Web Client

Users logging in using the Leostream Web client can connect to their Scyld Cloud Workstation using either the Scyld Cloud Workstation client or the Scyld Client Workstation HTML5 viewer (see [Launching the Scyld Cloud Workstation HTML5 Viewer](#)). The Leostream Web client launches the Scyld Cloud Workstation client using an URI, which is supported by Scyld Cloud Workstation version 11.2 and higher.

To configure a protocol plan to launch the Scyld Cloud Workstation software client from a Leostream Web client login:

1. Scroll down to the **Scyld Cloud Workstation** section of the protocol plan in the **Web Browser** section.
2. Change the **Priority** of Scyld Cloud Workstation to 1.
3. If any other protocol in the associated section of the protocol plan has a priority of 1, modify that protocol's priority to a lower number or to **Do not use**.
4. In the **URI to launch client** edit field, enter URI that launches the client. If you have not installed valid SSL certificates on the Scyld Cloud Workstation sever, launching the default URI displays a security warning.

For testing purposes, you can replace the default URI with the following URI to suppress the security warnings until you obtain a valid certificate.

```
scw://{IP}?ignoreSSLerror&user={USER}
```



Leostream does not recommend suppressing the security warnings as an alternative to obtaining a valid certification.

5. If your users do not have network access to the Scyld Cloud Workstation server, you can use the Leostream Gateway to forward the display protocol traffic. In this case, use the **Gateway** drop-down menu to select the Leostream Gateway that handles the Scyld Cloud Workstation traffic for desktop connections associated with this protocol plan.

Scyld Cloud Workstation Traffic is always along port 443, so ensure that your Leostream Gateways are configured in your Connection Broker to forward traffic along a random port, as depicted in step 5 in [Launching Scyld Cloud Workstation Clients from Leostream Connect](#).

Launching the Scyld Cloud Workstation HTML5 Viewer

Users logging into Leostream using the Leostream Web client can launch in-browser remote workstation connections using the Scyld Cloud Workstation HTML5 viewer.

To configure a protocol plan to launch the Scyld Cloud Workstation HTML5 viewer from a Leostream Web client login:

1. Scroll down to the **Scyld HTML5 Viewer** section of the protocol plan in the **Web Browser** section.
2. Change the **Priority** of Scyld HTML5 Viewer to 1.
3. If any other protocol in the associated section of the protocol plan has a priority of 1, modify that protocol's priority to a lower number or to **Do not use**.
4. If your users do not have network access to the Scyld Cloud Workstation server, you can use the Leostream Gateway to forward the display protocol traffic. In this case, use the **Gateway** drop-down menu to select the Leostream Gateway that handles the Scyld Cloud Workstation traffic for desktop connections associated with this protocol plan.

Scyld Cloud Workstation Traffic is always along port 443, so ensure that your Leostream Gateways are configured in your Connection Broker to forward traffic along a random port, as depicted in step 5 in [Launching Scyld Cloud Workstation Clients from Leostream Connect](#).

rdesktop RDP Remote Viewer

You can use the rdesktop open-source RDP remote viewer to connect to Windows desktops from a Linux client. To configure a protocol plan to use rdesktop:

1. In the protocol plan, scroll down to the **rdesktop** section.
2. Change the rdesktop **Priority** to 1 to make rdesktop the primary protocol for the Connection Broker to use or select a lower priority to use rdesktop as a backup protocol.

If your protocol plan assigns priorities to multiple protocols, you must ensure that rdesktop has a higher priority than RDP. All three of these protocols use the same port. Therefore, the Connection Broker uses whichever protocol has the highest priority without trying the other two protocols.

3. Use the **Command line parameters** field to customize the remote viewer. The default command line parameters are:

```
-u {USER} -p {PLAIN_PASSWORD} -d {DOMAIN} {IP} -f
```



Remove the `-f` option for users that need access to the Leostream Connect Sidebar menu. When in fullscreen mode, rdesktop forces the remote desktop window to the top, hiding the Sidebar.

You can use any rdesktop command line option, such as `-f` for full screen mode. See the [rdesktop documentation](#) for a description of supported command line parameters.

To use rdesktop in conjunction with the Java version of Leostream Connect running Apple macOS, you must recompile rdesktop. Consult the FAQ in the Leostream Web site for more information.

Scale Logic Remote Access Portal - VDI

The **Scale Logic Remote Access Portal (RAP) – VDI** solution provides a proven software-enabled VDI solution for remote editors and post-production professionals to seamlessly access systems and files from virtually anywhere, through any OS.

Leostream allows you to provide VPN-less remote access that leverages RAP – VDI to enable high-end 3D accelerated desktops, deploy GPU-enabled workstations for AI/ML, enable remote access to software suites for content creators, or build a platform to enable thousands of users with intuitive secure access to a familiar remote desktop.

Launching RAP - VDI Clients from Leostream Connect

To configure a protocol plan to launch the RAP - VDI software client from a Leostream Connect login:

1. Scroll down to the **Scale Logic RAP - VDI** section of the protocol plan in the **Leostream Connect and Thin Clients Writing to Leostream API** section.
2. Change the **Priority** of RAP - VDI to 1.
3. If any other protocol in the associated section of the protocol plan has a priority of 1, modify that protocol's priority to a lower number or to **Do not use**.
4. In the **Command line parameters**, enter the default command line parameters to use when launching the RAP - VDI client. The default parameters are defined using the Windows format (/). If this protocol plan will be assigned to users logging in from a Linux or macOS client device, replace the forward slashes with a double dash (--).

For a full list of supported command line parameters, install the RAP - VDI client and run the client from the command line with the `/help` (`--help`) parameter.



If you plan to enable the Leostream Gateway in this protocol plan, ensure that you place the string `https://` before the `{IP}` tag in the Command line parameters, for example:

```
/server=https://{IP} /user={USER} /password={PLAIN_PASSWORD}
```

5. If your users do not have network access to the virtual or physical machine running the RAP - VDI server software, you can use the Leostream Gateway to forward the display protocol traffic. In this case, use the **Gateway** drop-down menu to select the Leostream Gateway that handles the RAP - VDI traffic for desktop connections associated with this protocol plan.



RAP - VDI traffic is always along port 443. If you are using your Leostream Gateways to forward login traffic to your Connection Broker, ensure that your Leostream Gateways are configured in your Connection Broker to forward traffic along a random port, preferably filtered by Client IP for additional security, for example:

Edit Gateway "Leostream Gateway" ?

Name

Add this Leostream Gateway to a Gateway Cluster
[None available]

Public IP address or FQDN for use in Protocol Plans

If this Gateway is located behind a load balancer or external firewall, enter its public IP address or FQDN. This address must be accessible from the user's client device.

IP address or FQDN used for Connection Broker communications to this Gateway

Unique IP address or FQDN of this Leostream Gateway. Not required if same as Public IP address above.

Method for routing display protocol traffic through this Leostream gateway

If you are required to use the protocol-specific port 443 for the RAP – VDI traffic and you use your Leostream Gateways to forward Connection Broker login traffic, you must configure your Leostream Gateway to use a port other than 443. For instructions on changing the default Leostream Gateway port, see the [Leostream Gateway Guide](#).

Launching RAP - VDI Clients from the Web Client

Users logging in using the Leostream Web client can connect to remote desktops using either the RAP – VDI software client or the RAP – VDI HTML5 viewer. The Leostream Web client launches the RAP – VDI software client using a URI.

To configure a protocol plan to launch the RAP - VDI software client from a Leostream Web client login:

1. Scroll down to the **Scale Logic RAP - VDI** section of the protocol plan in the **Web Browser** section.
2. Change the **Priority** of Scyld Cloud Workstation to 1.
3. If any other protocol in the associated section of the protocol plan has a priority of 1, modify that protocol's priority to a lower number or to **Do not use**.
4. In the **URI to launch client** edit field, enter URI that launches the client. If you have not installed valid SSL certificate to use with the RAP – VDI sever running on the remote machine, launching the default URI displays a security warning.

For testing purposes, you can replace the default URI with the following URI to suppress the security warnings until you obtain a valid certificate.

```
rap://{IP}?ignoreSSLerror&user={USER}
```



Leostream does not recommend suppressing the security warnings as an alternative to obtaining a valid certification.

5. If your users do not have network access to the remote desktop, you can use the Leostream Gateway to forward the display protocol traffic. In this case, use the **Gateway** drop-down menu to select the Leostream Gateway that handles the RAP – VDI traffic for desktop connections associated with this protocol plan.

RAP – VDI uses port 443 so ensure that your Leostream Gateways are configured in your Connection Broker to forward traffic along a random port, as depicted in step 5 in [Launching RAP - VDI Clients from Leostream Connect](#).

Launching the RAP - VDI HTML5 Viewer

Users logging into Leostream using the Leostream Web client can launch in-browser remote workstation connections using the RAP - VDI HTML5 viewer.

To configure a protocol plan to launch the RAP – VDI HTML5 viewer from a Leostream Web client login:

1. Scroll down to the **Scale Logic RAP - VDI HTML5 Viewer** section of the protocol plan in the **Web Browser** section.
2. Change the **Priority** of the RAP – VDI HTML5 Viewer to **1**.
3. If any other protocol in the associated section of the protocol plan has a priority of 1, modify that protocol's priority to a lower number or to **Do not use**.
4. If your users do not have network access to the remote desktop running the RAP - VDI server, you can use the Leostream Gateway to forward the display protocol traffic. In this case, use the **Gateway** drop-down menu to select the Leostream Gateway that handles the RAP – VDI traffic for desktop connections associated with this protocol plan.

RAP – VDI uses port 443 so ensure that your Leostream Gateways are configured in your Connection Broker to forward traffic along a random port, as depicted in step 5 in [Launching RAP - VDI Clients from Leostream Connect](#).

VNC Remote Viewer

Leostream Connect supports the following versions of VNC; RealVNC®, RealVNC Enterprise, TigerVNC, TightVNC, and UltraVNC. UltraVNC allows the Windows username and password to be sent, enabling single sign-on.



To support single sign-on to a remote Windows desktop connected via VNC, you must select the single sign on task when installing the Leostream Agent on the remote desktops. Also, these users must have the **Enable single-sign-on to desktop console** option selected in the **When User is Assigned to Desktop** section of their policy.

Setting up the Connection Broker to Use VNC

To configure a protocol plan to use VNC

1. Scroll down to the **VNC** section of the protocol plan.
2. Select **1** from the **Priority** drop-down menu.
3. Use the **Command line parameters** and **Configuration file** fields to customize the remote viewer session.
4. Use the **Alternate port for remote viewer port check** to specify which port the Connection Broker uses to determine if the server is healthy. When using Leostream to start the VNC sessions, you must enter a port other than a standard VNC port, such as the Leostream Agent port.
5. If users are connecting to Linux VNC servers, use the **Command line parameters for VNC servers** to specify any VNC server parameters to pass to the Leostream Agent, to be used when launch the VNC session.
6. The **Check if VNC password is set (Linux, only)** checkbox indicates if the Connection Broker requires the VNS server to require a VNC password.


When checked, the VNC server must have an existing `$HOME/.vnc/passwd` file. When unchecked, the VNC server may have VNC password, but the Connection Broker does not check for the existence of the `passwd` file, allowing VNC sessions without VNC passwords.

In either case, if the VNC server includes a `passwd` file, the user is prompted for their VNC password when the Connection Broker attempts to establish the VNC connection.

When a user requests a VNC connection to a desktop, the Connection Broker contacts the Leostream Agent running on the desktop to start the underlying VNC session for the user. After starting the session, the Leostream Agent returns the port associated with the session to the Connection Broker.

New protocol plans include the `{VNC_PORT}` dynamic tag. The Connection Broker replaces this dynamic

tag the correct port for the user's VNC connection before sending the configuration file to the client device.

 The default configuration file includes a dynamic tag for the VNC port, which the Connection Broker automatically replaces with the port returned by the Leostream Agent for the VNC session. If users connect to desktops with existing VNC sessions, you can hard-code the port for the VNC session. By default, the VNC port is 5900 when connecting to a Windows desktop. If using VNC to connect to a Linux desktop, the port is 5901.

VNC Command Line Parameters

You can customize the VNC session using command line settings entered in the Command line parameters field. The command line parameters have the following format:

```
{IP}:nnnn [other_options]
```

Where:

- {IP}: The IP address completed by the Connection Broker.
- :nnnn: The port.

`-listen [port]`

Make the viewer listen on the given port for reverse connections from a VNC server. If no port is supplied, the command defaults to port 5500. WinVNC supports reverse connections using the **Add New Client** menu option, or the `-connect` command line option. Xvnc requires the use of the helper program `vncconfig`.

`-via gateway`

Automatically create encrypted TCP tunnel to the *gateway* machine before connection, connect to the *host* through that tunnel (TightVNC-specific). By default, this option invokes SSH local port forwarding, assuming that SSH client binary can be accessed as `/usr/bin/ssh`. Note that when using the `-via` option, the host machine name should be specified as known to the gateway machine, e.g. `localhost` denotes the *gateway*, not the machine where `vncviewer` was launched. See the ENVIRONMENT section below for the information on configuring the `-via` option.

`-shared`

When connecting, specify that a shared connection is requested. If this option is not set, when you make a connection, all other existing connections are closed. In TightVNC, this option is on, by default, allowing you to share the desktop with other clients already using it.

`-noshared`

When connecting, specify that the session may not be shared. This would either disconnect other connected clients or refuse your connection, depending on the server configuration.

`-viewonly`

Disable transfer of mouse and keyboard events from the client to the server. Often used in conjunction with `-shared`.

`-fullscreen`

Start in full-screen mode. Operating in full-screen mode may confuse X window managers. Typically, such conflicts cause incorrect handling of input focus or make the viewer window disappear mysteriously. See

the `grabKeyboard` setting in the RESOURCES section below for a method to solve input focus problem.

`-noraiseonbeep`

By default, the viewer shows and raises its window on remote beep (bell) event. This option disables such behavior (TightVNC-specific).

`-user username`

User name for UNIX[®] login authentication. Default is to use current UNIX user name. If this option is given, the viewer prefers UNIX login authentication over the standard VNC authentication.

`-passwd passwd-file`

File from which to get the password (as generated by the `vncpasswd(1)` program). The file is typically stored in `~/ .vnc/passwd`. This option affects only the standard VNC authentication and does not log the user in to Microsoft Windows.

`-encodings encoding-list`

TightVNC supports several different compression methods to encode screen updates. This option specifies a set of compression methods to use in order of preference. Specify encodings separated with spaces and enclosed in quotes, if more than one is specified. Available encodings, in default order for a remote connection, are `copyrect tight hextile zlib corre rre raw`. For a local connection (to the same machine), the default order to try is `raw copyrect tight hextile zlib corre rre`. Raw encoding is always assumed as a last option if no other encoding can be used for some reason. For more information on encodings, see the section ENCODINGS below.

`-bgr233`

Always use the BGR233 format to encode pixel data. This reduces network traffic, but colors may be represented inaccurately. The bgr233 format is an 8-bit true color format, with 2 bits blue, 3 bits green, and 3 bits red.

`-owncmap`

Try to use a PseudoColor visual and a private colormap. This allows the VNC server to control the colormap.

`-truecolour, -truecolor`

Try to use a TrueColor visual.

`-depth depth`

On an X server which supports multiple TrueColor visuals of different depths, attempt to use the specified one (in bits per pixel). If successful, this depth is requested from the VNC server.

`-compresslevel level`

Use specified compression *level* (0 to 9) for "tight" and "zlib" encodings (TightVNC-specific). Level 1 uses minimum of CPU time and achieves weak compression ratios, while level 9 offers best compression but is slow in terms of CPU time consumption on the server side. Use high levels with very slow network connections, and low levels when working over high-speed LANs. Do not use compression level 0; start with the level 1.

`-quality level`

Use the specified JPEG quality level (0 to 9) for the "tight" encoding (TightVNC-specific). Level 0 denotes bad image quality but impressive compression ratios, while level 9 offers good image quality at lower

compression ratios. The "tight" encoder uses JPEG to encode only those screen areas that are suitable for compression that experiences loss, so quality level 0 does not always mean unacceptable image quality.

`-nojpeg`

Disable JPEG compression that experiences loss in tight encoding (TightVNC-specific). Disabling JPEG compression is not a good idea in typical cases, as the tight encoder becomes less efficient. Use this option if it is absolutely necessary to achieve perfect image quality (see also the `-quality` option).

`-nocursorshape`

Disable cursor shape updates, protocol extensions used to handle remote cursor movements locally on the client side (TightVNC-specific). Using cursor shape updates decreases delays with remote cursor movements, and can improve bandwidth usage dramatically.

`-x11cursor`

Use a real X11 cursor with X-style cursor shape updates, instead of drawing the remote cursor on the framebuffer. This option also disables the dot cursor, and disables cursor position updates in non-fullscreen mode.

`-autopass`

Read a plain-text password from stdin. This option affects only the standard VNC authentication.

Session Shadowing and Collaboration

The Connection Broker allows users that connect to desktops using the Mechdyne TGX, NoMachine, and HP ZCentral Remote Boost protocols to collaborate by inviting a shadow user to connect to their session.

Configuring Collaboration in the Connection Broker

In order to collaborate, a user must have the appropriate role and policy setting, as described in the following procedure. The role determines which users have permission to share or shadow sessions. The policy determines which specific desktops support collaboration.

1. **Build pools:** To simplify creating policies, construct pools that contain only desktops that support collaboration.
2. **Configure roles:** You must explicitly give permission to each user that either shares their remote session or shadows another user's session. To provide the necessary permission, select the **Allow user to collaborate with other users** option in the user's role, as shown in the following figure.

3. **Create policies:** In addition to giving the user permission to collaborate with other users, you must indicate which remote sessions support collaboration. Use settings in the user's policies to indicate which desktops support collaboration,
 - a. Go to the **> Configuration > Policies** page.
 - b. Edit the user's policy or create a new policy.
 - c. On the **Pool Assignments** tab, add or edit the pool assignment that should allow collaboration.
 - d. In the **Edit Pool Assignment** form, select a pool that contains desktops that support collaboration.
 - e. In the **When User Connects to Desktop** section of the form, select the **Enable collaboration and**

session shadowing option, as shown in the following figure.

When User Connects to Desktop

Log user into remote desktop as: <Determined by user's role>

Log out any rogue users (also applies when reconnecting to assigned desktop)

Enable single sign-on to desktop console (DCV, DCV HTML5, VNC, NoMachine, NoMachine HTML5, Scyld HTML5 only)

Enable collaboration and session shadowing (NoMachine, RGS, TGX, and DCV only)

Allow collaboration: With all users

Allow users to generate email notifications for collaboration requests (SMTP Server not configured)

Send email to session owner if collaboration request is declined (SMTP Server not configured)

By default, a user assigned to this policy can collaborate with any other Leostream user with a Role that enables collaboration. The following section describes how to limit the user to inviting specific groups of users.

4. **Define assignments:** After configuring a role and policy that support collaboration, you must configure the > **Configuration > Assignments** pages to assign that role and policy to the appropriate users.

Limiting Collaboration to Groups of Users

If, for example, you have users working on different projects and want to enable collaboration only between users on the same project, you can switch the **Allow collaboration** drop-down menu to **Only with users matching specific attribute rules**, as shown in the following figure.

Enable collaboration and session shadowing (NoMachine, RGS, TGX, and DCV only)

Allow collaboration: Only with users matching specific attribute rules

| User LDAP/SAML attribute | Conditional | Attribute value |
|--------------------------|-------------|-----------------|
| memberOf | contains | Collaborators |
| | | |
| | | |

Add row

The user must match any of the attribute rules (OR)

The user must match all of the attribute rules (AND)

In this mode, you can use the table below the drop-down menu to specify who a user of this policy can invite to collaborate. For example, in the previous figure, only users with an AD `memberOf` attribute that contains `Collaborators` can be invited. The Collaborate users must still have an appropriate Leostream Role that allows them to participate in collaboration.

Sending Email Notifications Related to Collaboration Invitations



In order to send email notifications related to invitations, you must define an SMTP server in your Connection Broker. See “Configuring an SMTP Server for Alerts” in the [Connection Broker Administrators](#)

Guide for more information.

Users see their sent and received invitations anytime they log into the Leostream Web client or Leostream Connect. You can use the policy options shown in the following figure to send email notifications that an invitation is available or has been declined.

The screenshot shows a configuration panel with the following options:

- Enable collaboration and session shadowing (NoMachine, RGS, TGX, and DCV only)
- Allow collaboration:
- Allow users to generate email notifications for collaboration requests (SMTP Server not configured)
- Send email to session owner if collaboration request is declined (SMTP Server not configured)

- **Allow users to generate email notifications for collaboration request** – Gives the user sending the invitation an option to send an email to the user they are inviting to collaborate
- **Send email to session owner if collaboration request is declined** – Automatically sends an email to the session owner if the invited user declines their invitation.

The content of both emails is determined by two template text files that live within your Connection Broker in the `/var/lib/leo/app/templates` directory.

- `collaboration_invite_request.mail.txt` – Edit this file to change the default email that is sent when a user invites another user to collaborate.
- `collaboration_invite_declined.mail.txt` – Edit this file to change the default email that is sent when the invited user declines the invitation.



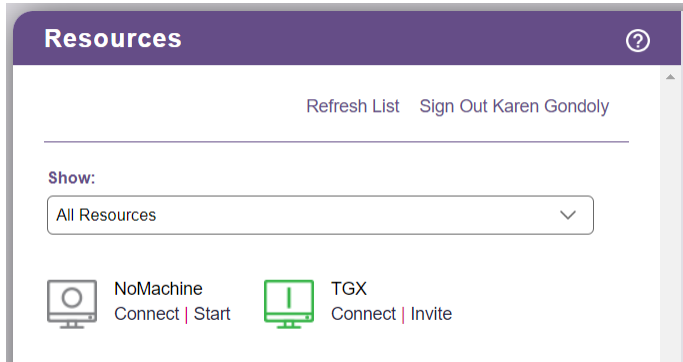
If you have a Connection Broker cluster, ensure that you modify the text files on every Connection Broker in your cluster.

Working with Invitations in the Leostream Web Client

Sending a Collaboration Invitation

A user with permission to invite other users to shadow their remote session sends the invitation, as follows.

1. Click the **Connect** link to establish the desktop connection.
2. After the sessions starts, click the **Refresh List** link in the Leostream Web client. All desktops with an assigned desktop that allows collaboration includes a new **Invite** link.
3. Click the **Invite** link, shown in the following figure.



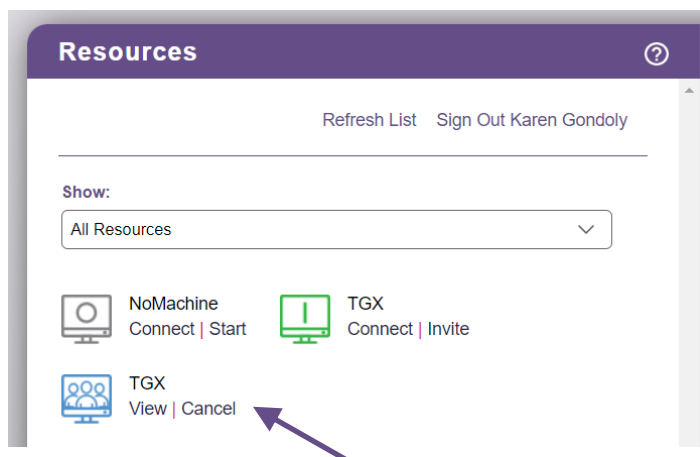
1. In the **Create Collaboration Request** form that opens, select the user to invite to the session from the **Send invitation to** drop-down menu.



Only Leostream users who are already listed on the Connection Broker > **Resources > Users** page can be entered as a shadow user. The selected user *must* have a role that gives them permission to collaborate with other users. Otherwise, the user will not see the invitation.

4. From the **Cancel invitation if not accepted within** drop-down menu, indicate when the invitation should expire. Leave this drop-down menu blank if the invitation should be valid for the lifetime of the remote session.
5. In the **Notes** field, provide an optional message to display to the invited user.
6. Check the **Send email notification to invited user** option if the invited user should be notified that the invitation is waiting for them.
7. Click **Send**.

The Leostream Web client now displays the new invitation, for example:



Click the **View** link associated with an invitation to see the details of that information, including any entered notes, for example:

View Invitation

| Invitation ID | Type | Invitation Status | Invitation Sent |
|---------------|------------|-------------------|-----------------------|
| 1 | Invitation | Sent | 2023-05-11 - 12:53:29 |

Invitation Expiration
2023-05-11 - 12:58:29

Session owner
kgondoly (Karen Gondoly)

Collaborator
maybel (Maybel Bable)

Desktop
TGX

Notes
no value

Close

Canceling an Invitation


To cancel an invitation, click the **Cancel** link associated with that invitation, as shown in the following figure.

Resources ?


Refresh List Sign Out Karen Gondoly

Show:


All Resources v



NoMachine
Connect | Start



TGX
Connect | Invite



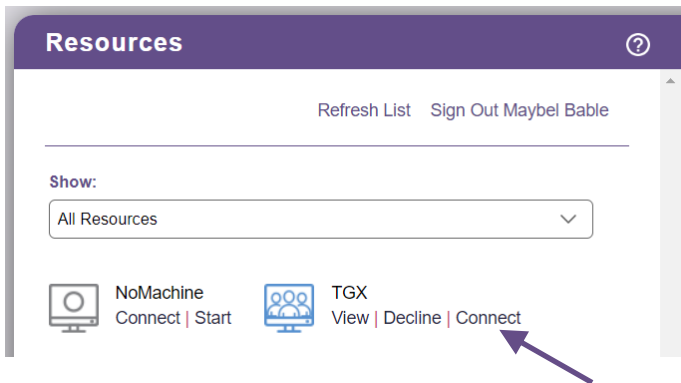
TGX
View | **Cancel**



Invitations are automatically cancelled when the desktop is released back to its pool, for example, when you log out of the remote session and have a release plan that releases the desktop on logout.


Accepting a Collaboration Invitation

A user's pending invitations appear in the Resource list with other offered desktops, as shown in the following figure.



- Click the **View** link to see the invitation's details, including any message sent with the invitation.
- Click the **Decline** link to remove the invitation from the resource list without connecting to the shadowed session. Clicking Decline will send an email notification to the session owner if their policy was configured to do so.
- Click the **Connect** link to connect to the session.

Working with Invitations using Leostream Connect

 Leostream Connect currently supports collaboration only for users with a policy that allows all users to collaborate. To limit the user to inviting certain group of users for collaboration, the users must log in using the Leostream Web client.

Sending a Collaboration Invitation

After logging into the Windows version of Leostream Connect and launching their remote session, a user with permission to invite other users to shadow their session can invite the user, as follows.

2. After the remote sessions starts, right-click on the Leostream Connect System Tray menu and select the **Refresh List** option. All desktops with an assigned desktop that allows collaboration includes a new **Invite** link.
3. Again, right-click on the Leostream Connect System Tray menu and highlight the desktop that you want to invite the user to shadow. When you highlight the desktop name, a submenu opens.
4. Select **Invite...** from the submenu. The **Send Invitation** form opens.
5. In the **Shadow user** field, begin typing the name of the user to invite.



Only Leostream users who are already listed on the Connection Broker > **Users** page can be entered as a shadow user. The selected user *must* have a role that gives them permission to

collaborate with other users. Otherwise, the user will not see the invitation.

6. From the **Expire after** drop-down menu, indicate when the invitation should expire. Leave this drop-down menu blank if the invitation should be valid for the lifetime of the remote session.
7. In the **Notes** field, provide an optional message to display to the invited user.
8. Click **Send** to send the invitation or **Cancel** to close the form without sending the invitation.

Viewing and Cancelling Invitations

After you send an invitation, right-click on the Leostream Connect System Tray menu and select the **Refresh List** option. A new **View Invitations** menu item is added to the list.

Selecting the **View Invitations** menu opens up a dialog displaying the list of invitations that have been sent by you.

Click the **Cancel** link to cancel any invitations you sent to other users.

Accepting a Collaboration Invitation

To access an invitation sent to you from another user, log into Leostream Connect and then close the **Connect** dialog. Right click on the Leostream Connect System Tray menu and select **View Invitations**. The Invitations list dialog that opens shows all the pending invitations for the logged in user.

- Click the **Decline** link to remove the invitation from the resource list without connecting to the shadowed session.
- Click the **Connect** link to shadow the session.

Managing Invitations in the Connection Broker

All past and present invitations appear on the > **Resources** > **Collaboration** page, shown in the following figure.



| Signed in as Administrator | | | | | | |
|----------------------------|------|-------------------|---------------|---------------|--------------|--|
| Dashboards | | | | | | |
| Setup | | | | | | |
| Configuration | | | | | | |
| Resources | | | | | | |
| Desktops | | | | | | |
| Images | | | | | | |
| Users | | | | | | |
| Collaborations | | | | | | |
| Actions | Type | Invitation Status | Invitation ID | Session Owner | Collaborator | |
| View Dismiss | All | All | 1 | Karen Gondoly | Maybel Bable | |

The **Actions** links allow you to do the following:

- Click **View** to see details about the invitation.
- Click **Dismiss** to cancel the invitation

The **Status** column provides information about what actions have been taken on the initiation.

- **Cancelled** indicates that the invitation was cancelled. The **Viewed** column indicates if the invited user connected to the shadowed session before the invitation was cancelled.
- **Declined** indicates that the invitee declined the invitation.
- **Expired** indicates that the invitation has expired and the invitee can no longer join the session.
- **Sent** indicates an invitation has been sent, but has not been declined or viewed by the invitee.
- **Viewed** indicates that the invitee has connected to the shadowed session.

User Configurable Protocol Plan Parameters

User-configurable protocol plan parameters give users control over the look-and-feel of their desktop connection when using NoMachine NX and HP ZCentral Remote Boost display protocols.

Defining Scope of the Configured Parameter

User-configured values are stored either globally and used for all connections made by that user, or individually for a desktop/client pair.

The scope of the user-configured values is determined by the setting of the **Store user-configured protocol parameters** drop-down menu in the user's policy, shown in the following figure.

The screenshot shows the 'Edit Policy "Collaborator"' interface. The 'General' tab is selected. The 'Policy name' field contains 'Collaborator'. Below this, there are several checkboxes for various settings. At the bottom, the 'Store user-configured protocol parameters' dropdown menu is open, showing three options: 'Individually for each connection/client pair' (which is highlighted in blue), 'Individually for each connection/client pair', and 'Globally for all connections and from every client'.

- **Globally for all connections and from every client** allows the user to set the value once and use it everywhere. In this case, the setting applies to all policies, pools, and desktops assigned to the user.
- **Individually for each connection/client pair** allows the user to configure different values for each of their connections. In this case, the user must reset their desired value when they log into Leostream at a different client device.

The Connection Broker considers Leostream Connect and the Leostream Web client accessed from the same physical device as two different clients.

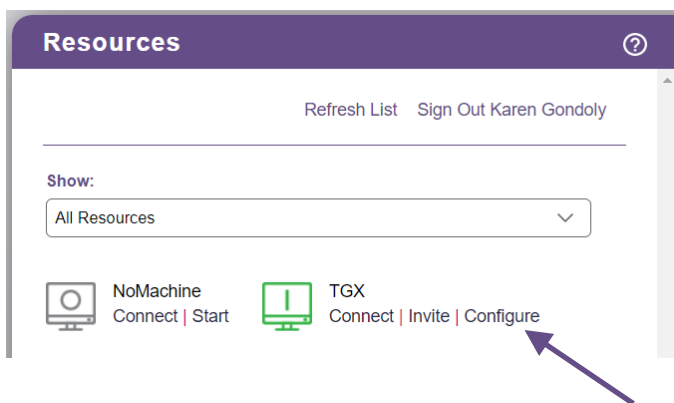
End-User Interface for Configuring Parameters

End-users can configure protocol plan parameters when logging in through the Leostream Web client and Leostream Connect.

Leostream Web Client

Users logging in from the Leostream Web client set user-configurable parameters, as follows.

1. After logging into the Web client, any connections with configurable parameters include a **Configure** link, as shown in the following figure.



2. After the user clicks the **Configure** link, a new form opens, displaying the parameters they are allowed to set.
3. Click **Save** to store new values for all subsequent applicable connections. Click **Apply and Close** to use the new values only for the current connection.

Leostream Connect

Users logging in from Leostream Connect set protocol parameters, as follows.

1. After logging into the Leostream Connect client, highlighting any connection with configurable parameters enables a **Configure** button at the bottom-right of the dialog.

The Java version of Leostream Connect uses a JavaFX based browser component to display the dialog. You must install a JRE that includes JavaFX to configure parameters using the Java version of Leostream Connect.

2. After the user clicks the **Configure** link, a new form opens, displaying the parameters they are allowed to set.
3. Clicks **Save and Close** to store new values for all subsequent applicable connections. Click **Apply and Close** to use the new values only for the current connection.

Setting Global User-Configurable Parameters

When the user's policy is set to store user-configure parameters globally, the same parameter values are used regardless of which policy, pool, or desktop the user is currently offered. For example, even if the user's policy changes based on their location, the same configured values are used at each location.

In addition, the Connection Broker stores global parameters based on the protocol. Therefore, all connections that use the HP ZCentral Remote Boost display protocol, for example, use the same global parameter set.

If the policy specifies that user-configured parameters are stored globally, the user is prompted to set the parameters for all connections offered by a policy when they click the **Configure** link for a particular connection.

If users need more flexibility when setting protocol plan parameters, modify their policy to store parameters individual for each desktop/client pair.