



Preview Server Administrator Guide

Version 5.0



Preview Server





Copyright ©2026 Vizrt. All rights reserved.

No part of this software, documentation or publication may be reproduced, transcribed, stored in a retrieval system, translated into any language, computer language, or transmitted in any form or by any means, electronically, mechanically, magnetically, optically, chemically, photocopied, manually, or otherwise, without prior written permission from Vizrt. Vizrt specifically retains title to all Vizrt software. This software is supplied under a license agreement and may only be installed, used or copied in accordance to that agreement.

Disclaimer

Vizrt provides this publication “as is” without warranty of any kind, either expressed or implied. This publication may contain technical inaccuracies or typographical errors. While every precaution has been taken in the preparation of this document to ensure that it contains accurate and up-to-date information, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained in this document.

Vizrt’s policy is one of continual development, so the content of this document is periodically subject to be modified without notice. These changes will be incorporated in new editions of the publication. Vizrt may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time. Vizrt may have patents or pending patent applications covering subject matters in this document. The furnishing of this document does not give you any license to these patents.

Antivirus Considerations

Vizrt advises customers to use an AV solution that allows for custom exclusions and granular performance tuning to prevent unnecessary interference with our products. If interference is encountered:

- **Real-Time Scanning:** Keep it enabled, but exclude any performance-sensitive operations involving Vizrt-specific folders, files, and processes. For example:
 - C:\Program Files\[Product Name]
 - C:\ProgramData\[Product Name]
 - Any custom directory where [Product Name] stores data, and any specific process related to [Product Name].
- **Risk Acknowledgment:** Excluding certain folders/processes may improve performance, but also create an attack vector.
- **Scan Scheduling:** Run full system scans during off-peak hours.
- **False Positives:** If behavior-based detection flags a false positive, mark that executable as a trusted application.

Technical Support

For technical support and the latest news of upgrades, documentation, and related products, visit the Vizrt web site at www.vizrt.com.

Created on

2026/09/08

Contents

1	Introduction	5
1.1	Related Documents	5
1.2	Feedback and Suggestions	5
1.3	Support	5
2	About Preview Server.....	6
2.1	Basic Workflow	6
3	Requirements and Recommendations	8
3.1	Software Requirements	8
3.1.1	Antivirus.....	8
3.2	Hardware Requirements	8
3.3	Virtual Environments.....	8
3.4	Security Advisor	10
3.4.1	Recommendations	10
3.4.2	Ports Used by Preview Server	10
4	Installation and Configuration	11
4.1	Installation	12
4.1.1	Installing Preview Server	12
4.2	Configuration	14
4.2.1	Preview Server Service.....	14
4.2.2	Viz Engines	14
4.2.3	Service Configuration.....	14
4.2.4	Setting Up HTTPS	17
4.2.5	Viz Pilot Data Server	17
4.2.6	Configuring Preview Server Settings in Viz Trio	17
4.3	Settings	18
4.3.1	Using the Web Interface	18
4.3.2	Using the REST API	18
4.3.3	Settings that Require a Restart.....	20
5	Working with Preview Server.....	21
5.1	Web Interface	22
5.1.1	Home.....	22
5.1.2	Settings	23
5.1.3	Frame API.....	23

5.1.4	Connection Broker API	24
5.1.5	Licenses	25
5.1.6	About.....	26
5.2	Common Procedures.....	28
5.2.1	Startup and Shutdown.....	28
5.2.2	Viz Engine Configuration.....	29
6	Version API.....	32
7	Troubleshooting.....	33
7.1	Services Won't Start After Installing the Latest Preview Server	33
7.2	Known Limitations	33

1 Introduction

This Administrator Guide covers the configuration and operation of Preview Server.

1.1 Related Documents

See the following documents at the [Documentation Center](#):

- *Graphics Plugin Administrator and User Guides*
 - *Viz Engine Administrator Guide*
 - *Viz Multichannel User Guide*
 - *Viz Pilot User Guide*
 - *Viz Trio User Guide*
-

1.2 Feedback and Suggestions

We encourage suggestions and feedback about our products and documentation. To give feedback and/or suggestions, please contact your local Vizrt customer support team at www.vizrt.com.

1.3 Support

Support is available at the [Vizrt Support Portal](#).

For more information about all Vizrt products, visit:

- www.vizrt.com
- [Vizrt Documentation Center](#)
- [Vizrt Training Center](#)
- [Vizrt Forum](#)

2 About Preview Server

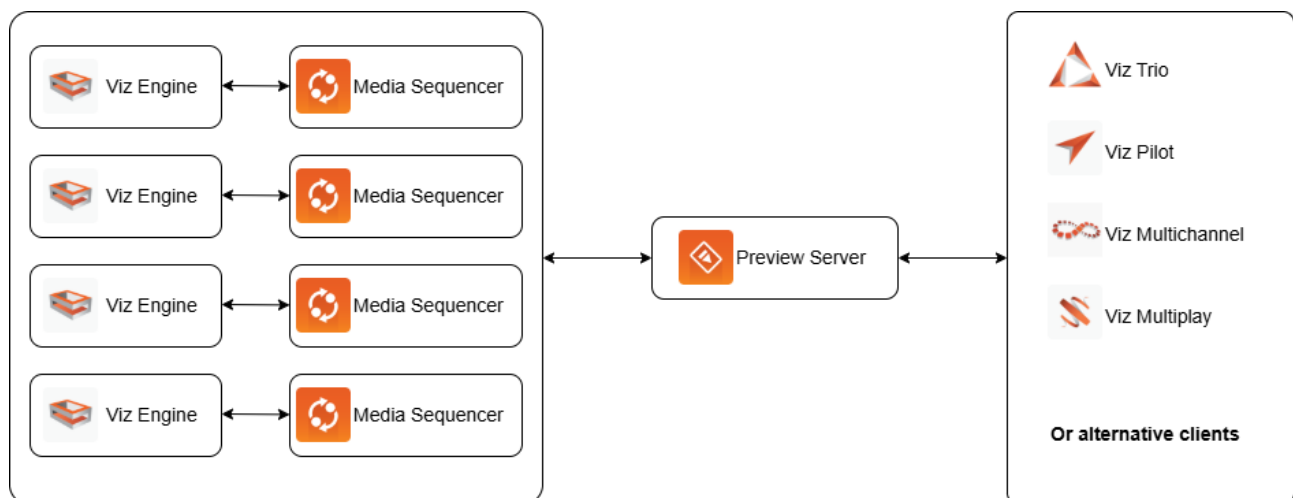
Preview Server is a Windows service that manages a pool of **Viz Engine** processes. Client software, such as **Viz Multichannel**, **Viz Pilot**, **Viz Pilot Edge**, **Viz Multiplay** and **Viz Trio**, send requests to Preview Server for snapshots of **Viz Engine** graphics. The content and animation in and out points of the graphics can be specified in the request.

Preview Server provides snapshots of scenes during the creative preview process. The preview snapshots provide an indication of the look of the graphics that are played out in high resolution on a Viz Engine. It also provides load balancing and can be used to create a redundant renderer pool.

Upon installation, the service is started and the startup type is set to *Automatic* by default. The service then starts every time the operating system is started, and continues to run in the background as long as Windows is running. For more details, see [See the Status of the Preview Server Service](#).

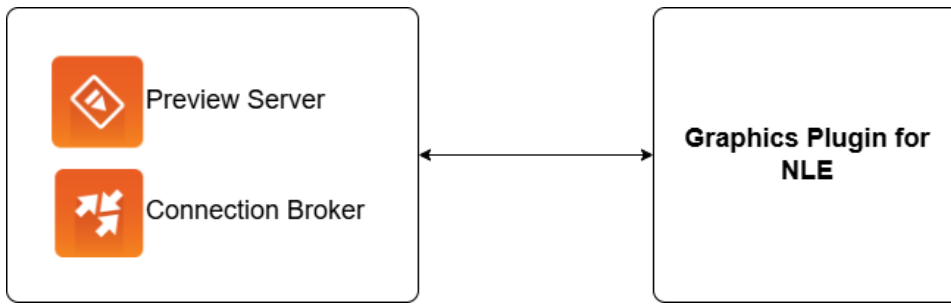
2.1 Basic Workflow

A typical use case is as follows: A Viz Trio user wants to add graphics to a video in Viz Trio's Timeline editor, a video clip and one or more data elements (graphics) are dragged to the timeline. In the Timeline editor, the graphics elements are added as placeholders on the video timeline. Since the user wants to display snapshots of the graphics on these placeholders, a Preview Server connection is set up (for details, see [Configuring Preview Server Settings in Viz Trio](#)). Viz Trio sends a request to Preview Server for an available Viz Engine to generate the necessary preview snapshots, and Preview Server passes the task to a suitable Viz Engine using Media Sequencer to which this Viz Engine has been attached. This Viz Engine in turn generates the snapshots that were requested by Viz Trio.



When used together with Vizrt control applications like Viz Multichannel, Viz Pilot, and Viz Trio, Preview Server is installed from a standalone installer.

Alternatively, when used in a workflow together with the Graphics Plugin for NLE products, Preview Server is installed from a joint installer package, containing both Preview Server and Connection Broker.



For more details on this joint Preview Server and Connection Broker setup, see the [Graphics Plugin Administrator Guide](#).

3 Requirements and Recommendations

3.1 Software Requirements

Microsoft .NET Framework 4.6.2 or later.

3.1.1 Antivirus

Vizrt advises customers to use an AV solution that allows for custom exclusions and granular performance tuning to prevent unnecessary interference with our products.

A list of services and folders which should be excluded from scanning can be found below:

Preview Server	Connection Broker
Services: • Preview Server Folder: • C:\Program Files\Vizrt • C:\ProgramData\vizrt Processes: • PreviewServer.exe	Services: • Connection Broker Folder: • C:\Program Files\Vizrt • C:\ProgramData\vizrt Processes: • ConnectionBroker.exe

3.2 Hardware Requirements

A licensed Viz Engine.

 **Note:** Preview Server can run on the same machine as a Viz Engine and Media Sequencer, but it is recommended to specify additional pairs of Viz Engine and Media Sequencer in order to spread the load.

3.3 Virtual Environments

Preview Server can run in a virtual environment. However, if you plan to run Viz Engine on the same virtual machine, make sure that the virtual machine is set up with the requirements to run Viz Engine, see chapter 4 and 5 in the [Viz Engine Administrator Guide](#).

Since Preview Server is configured via a [web browser](#) on localhost and the web browser and Preview Server must run on the same machine, you must use remote desktop to configure your system.

 **Tip:** When running in a virtual environment, it can be convenient to configure Preview Server for several Viz Engines.

See Also

- [Preview Server Release Notes](#)

3.4 Security Advisor

3.4.1 Recommendations

After installing the software, we recommend the following settings:

- Enable HTTPS communication port.
- If HTTPS is active, disable the unsecure HTTP port.
- Enable *AdministrationFromLocalhostOnly*.

See the [Configuration](#) page to enable these settings.

3.4.2 Ports Used by Preview Server

4443 (configureable)	TCP	Secure HTTPS Port
21098 (configureable)	TCP	Configuration HTTP Web Service
54000 (configureable)	TCP	REST Command Web Interface

4 Installation and Configuration

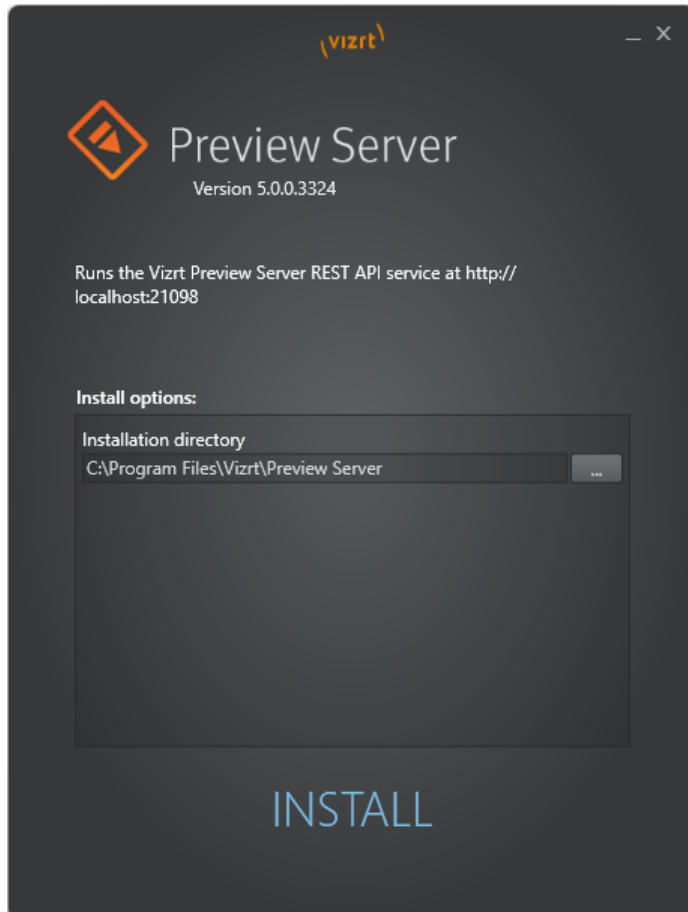
This section covers the following topics:

- [Installation](#)
- [Configuration](#)
- [Settings](#)

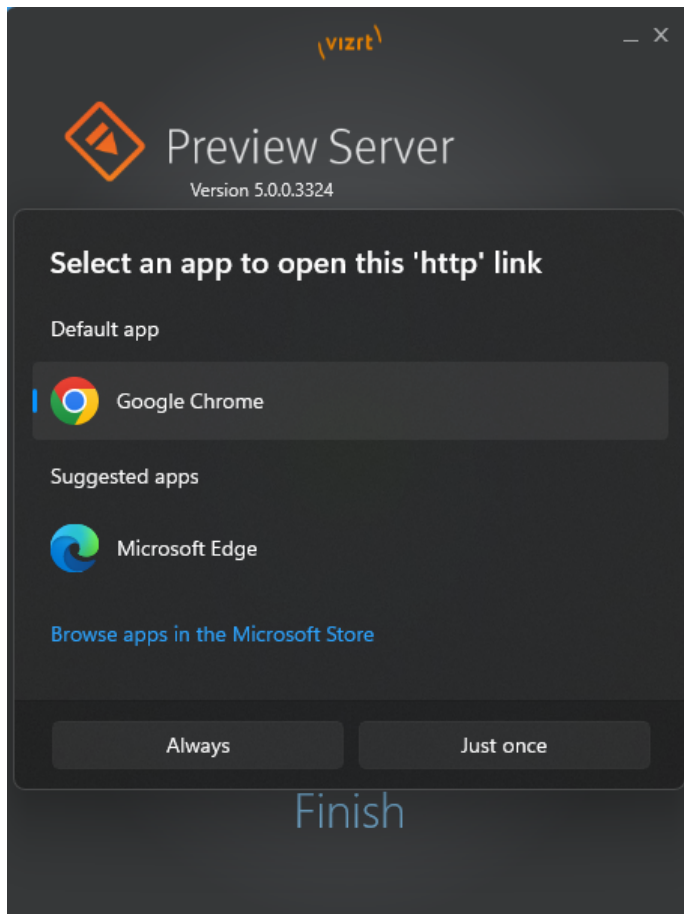
4.1 Installation

4.1.1 Installing Preview Server

1. Run the Preview Server installer file (*.exe) to start the *Preview Server Setup*.
2. Make sure the **Installation directory** is correctly defined.
3. Click **INSTALL**.



4. After the installation is complete, the installer attempts to open the [Web Interface](#) in the default browser. If multiple browsers are installed on your machine, a popup window may appear prompting you to choose which browser to use to open the configuration page.



5. Click **Finish** to close the installer.

Note: The Preview Server service is automatically started, and the startup type is set to *Automatic*, meaning that the service starts every time the operating system is started, and then runs in the background for as long as Windows is running. For more details, see [See the Status of the Preview Server Service](#).

4.2 Configuration

This section covers the following topics:

- [Preview Server Service](#)
- [Viz Engines](#)
- [Service Configuration](#)
- [Setting Up HTTPS](#)
- [Viz Pilot Data Server](#)
- [Configuring Preview Server Settings in Viz Trio](#)

4.2.1 Preview Server Service

Preview Server is a Windows Service that sets up an HTTP REST service on port 21098 on the host on which it is running. This service accepts requests for graphical snapshots of layout elements from a Viz Engine.

4.2.2 Viz Engines

Although Preview Server can run on the same machine as a Viz Engine (also used for the data element preview in Viz Pilot's newsroom client) and Media Sequencer to which this Viz Engine is attached, it is also possible to specify additional pairs of Media Sequencer and attached Viz Engine in order to spread the load. Frame requests are balanced across available Viz Engines in a way that minimize the memory load of each engine. The pool of Viz Engines is configured in the [Web Interface](#) Home page.

 **Note:** Scaling of the system should be monitored to avoid excessive client waiting time and any potential overload of the Viz Engines.

4.2.3 Service Configuration

You can modify Preview Server's service behavior using the settings in the `appsettings.json` file, available in `ProgramData` directory (`c:\ProgramData\vizrt\Preview Server\` by default).

Settings in `appsettings.json` file are divided into 2 groups:

- Actual service settings are coming under the `Settings` key:

Setting	Description
<code>CorsAllowedOrigin</code>	This setting can be used to restrict the allowed cross-origin resource sharing pattern utilized by Preview Server.
<code>ServicePort</code>	The service port setting lets you change the port number for regular HTTP requests for the Preview Server service. The service does not accept any HTTP requests if 0 is specified.

Setting	Description
<code>SecurePort</code>	The secure port setting lets you change the port number for secure HTTPS requests for the Preview Server service. The service does not accept any HTTPS requests if <code>0</code> is specified.
<code>Redraws</code>	This number specifies how many redraws a renderer must perform before taking a snapshot of the graphic.  Note: Although increasing the redraw number slows down the overall performance of snapshot requests, it might be required for some scenes, for example when using data pool plugins to render correctly.
<code>ExternalBaseUrl</code>	Changes the default URL base for snapshot requests, for example. This setting also overrides any <code>Host</code> , <code>X-Forwarded-Proto</code> and <code>X-Forwarded-Port</code> headers in requests. An empty string uses the default URL base and enables recognition of such headers in requests.
<code>AlwaysRunPreviewScript</code>	If set to <code>True</code> running <code>OnPreview</code> script hooks in the scene before taking a snapshot is enabled. Note that this might delay snapshot requests for scenes containing such hooks.
<code>OpenIdConnectServerUrl</code>	The URL to the OpenId Connect server that is trusted to handle user authentication. When this is set, all REST communication with the server must include an Authorization HTTP header value with a valid bearer token signed by this OpenId Connect server.
<code>ValidAudience</code>	When <code>OpenIdConnectServerUrl</code> is set, the bearer token must contain this value in its list of valid audiences. Leave empty to allow any authenticated audience.
<code>AdministrationFromLocalhostOnly</code>	If set to <code>True</code> this setting locks down the administration REST APIs (including documentation and web pages) so they can only be used from the machine on which this server is running. This is set to true by default.

Setting	Description
LogLevel	This setting can be used to specify the verbosity of logs produced by Preview Server. Possible values are: Debug : A debug log message is used for writing messages that are generally very uninteresting for the end user but may be useful for debug purposes. Info : An Info log message is used for writing general messages describing what Preview Server is doing at the moment. Warning : A Warning log message is used for irregular situations that may cause unexpected results. Error : An Error log message is used for serious errors that are unexpected, and may cause situations with unwanted results. Set to Warning by default.
ThumbPrint	Allows users to specify which certificate to use, to handle HTTPS connection. See Setting Up HTTPS for more details
PreviewServerLegacyPort	Specifies the legacy communication port used by Preview Server for backward compatibility with old clients. The default value is 54000. Modification of this setting won't change anything, and Preview Server still uses port 54000 for backward compatibility. Provided in options for clarity.

- List of Media Sequencer attached to this instance of Preview Server is coming under the **Sequencers** key. Even though this field can be modified manually in the configuration file, it is highly recommended to configure this part of Preview Server using the **Web Interface** Home page.

The following is an example of a configuration file (`appsettings.json`):

```
{
  "Settings": {
    "CorsAllowedOrigin": "*",
    "ServicePort": 21098,
    "SecurePort": 4443,
    "PreviewServerLegacyPort": 54000,
    "ExternalBaseUrl": "",
    "OpenIdConnectServerUrl": "",
    "ValidAudience": "api.vizrt.com",
    "Redraws": 0,
    "AlwaysRunPreviewScript": false,
    "AdministrationFromLocalhostOnly": true,
    "ThumbPrint": "",
    "LogLevel": "Warning"
  },
}
```

```
"Sequencers": {
  "List": [
    "http://localhost:8580/"
  ]
}
```

4.2.4 Setting Up HTTPS

- The certificate must be installed in the `LocalMachine\My` store with the private key.
- Once installed, the certificate thumbprint must be added to the Preview Server configuration file (`Thumbprint` option).
- If both `Thumbprint` and `SecurePort` are provided, Preview Server service automatically opens the HTTPS port upon restart using the options provided.

Certificate's thumbprint on Windows can be found by running the following command in PowerShell `Get-ChildItem -Path Cert:\LocalMachine\My`, or in cmd `certutil -store My`.

4.2.5 Viz Pilot Data Server

To grant applications connected to the Viz Pilot database access to Preview Server, see the [Pilot Data Server Administrator Guide](#).

4.2.6 Configuring Preview Server Settings in Viz Trio

If Preview Server is used together with Viz Trio, some Preview Server settings must be defined in Viz Trio:

1. In **Trio Configuration**, navigate to the **Connectivity > Viz One** panel.
2. Enter the hostname and port number in the **Preview Server Host** box (format `hostname:port`).
This enables snapshots of graphics to be displayed as video overlays in Viz Trio's Timeline editor.

To enable the Preview Server to load actual thumbnails in the playlist, rather than displaying the default template thumbnails throughout, perform the following steps:

1. In **Trio Configuration**, navigate to the **User Interface > Page List/Playlist** panel.
2. Enable the **Use Preview Server to load thumbnails** box.

For more details, see the [Viz Trio User Guide](#).

4.3 Settings

The settings of Preview Server can be changed in one of two ways, described below.

 **Note:** Both of these approaches are only available when accessing Preview Server as `localhost`.

4.3.1 Using the Web Interface

Navigate to `<hostname>:21098/settings`, or click the **Settings** link on the home page, and change the settings by putting the desired values in the form. You are not required to click a “submit” button, as the settings are automatically updated and saved when you change any settings on this page.

4.3.2 Using the REST API

Fetching Current Settings

You can send a `GET` request to `<hostname>:21098/api/settings` to receive a JSON object representing Preview Server’s current settings. Below is an example using PowerShell:

```
$settings = (Invoke-WebRequest http://localhost:21098/api/settings).Content |  
ConvertFrom-Json
```

This structure returns a PowerShell object:

```
corsAllowedOrigin      : *  
servicePort            : 21098  
securePort             : 4443  
previewServerLegacyPort : 54000  
externalBaseUrl        :  
openIdConnectServerUrl :  
validAudience          : api.vizrt.com  
redraws                : 0  
alwaysRunPreviewScript : False  
administrationFromLocalhostOnly : True  
thumbPrint             :  
logLevel               : Warning
```

You can then make any changes to the settings object, and send it back, for example changing the log level to *Info*:

```
$settings.logLevel = "Info"
```

Now the object looks like this:

```

corsAllowedOrigin      : *
servicePort            : 21098
securePort             : 4443
previewServerLegacyPort : 54000
externalBaseUrl        :
openIdConnectServerUrl :
validAudience         : api.vizrt.com
redraws                : 0
alwaysRunPreviewScript : False
administrationFromLocalhostOnly : True
thumbPrint             :
logLevel               : Info

```

Now, we can send the updated settings object back to Preview Server.

Updating Settings

To update the settings, send a **POST** to the same URL with a JSON object representing the desired settings. Below is an example using PowerShell:

```

Invoke-WebRequest -Method POST http://localhost:21098/api/settings -Body "$($settings
| ConvertTo-Json)" -ContentType "application/json"

```

Returns a response like this:

```

StatusCode      : 200
StatusDescription : OK
Content         : {}
RawContent      : HTTP/1.1 200 OK
                  Content-Length: 0
                  Date: Mon, 24 Mar 2025 12:45:16 GMT
                  Server: Kestrel

Headers         : {[Content-Length, 0], [Date, Mon, 24 Mar 2025 12:45:16 GMT],
                  [Server, Kestrel]}
RawContentLength : 0

```

You can verify the setting was changed by fetching the setting again, as described above.

You can also change the settings by specifying the settings you want to change, for example to change the log level back to *Warning*:

```

Invoke-WebRequest -Method POST http://localhost:21098/api/settings -Body
'{"logLevel": "Warning"}' -ContentType "application/json"

```

4.3.3 Settings that Require a Restart

The settings exposed in the [web interface](#) do not require a restart of Preview Server to take effect. Below is a list of settings that *do* require a restart before they take effect:

- `corsAllowedOrigin`
- `servicePort`
- `securePort`
- `previewServerLegacyPort`
- `externalBaseUrl`
- `openIdConnectServerUrl`
- `validAudience`
- `administrationFromLocalhostOnly`
- `thumbPrint`

5 Working with Preview Server

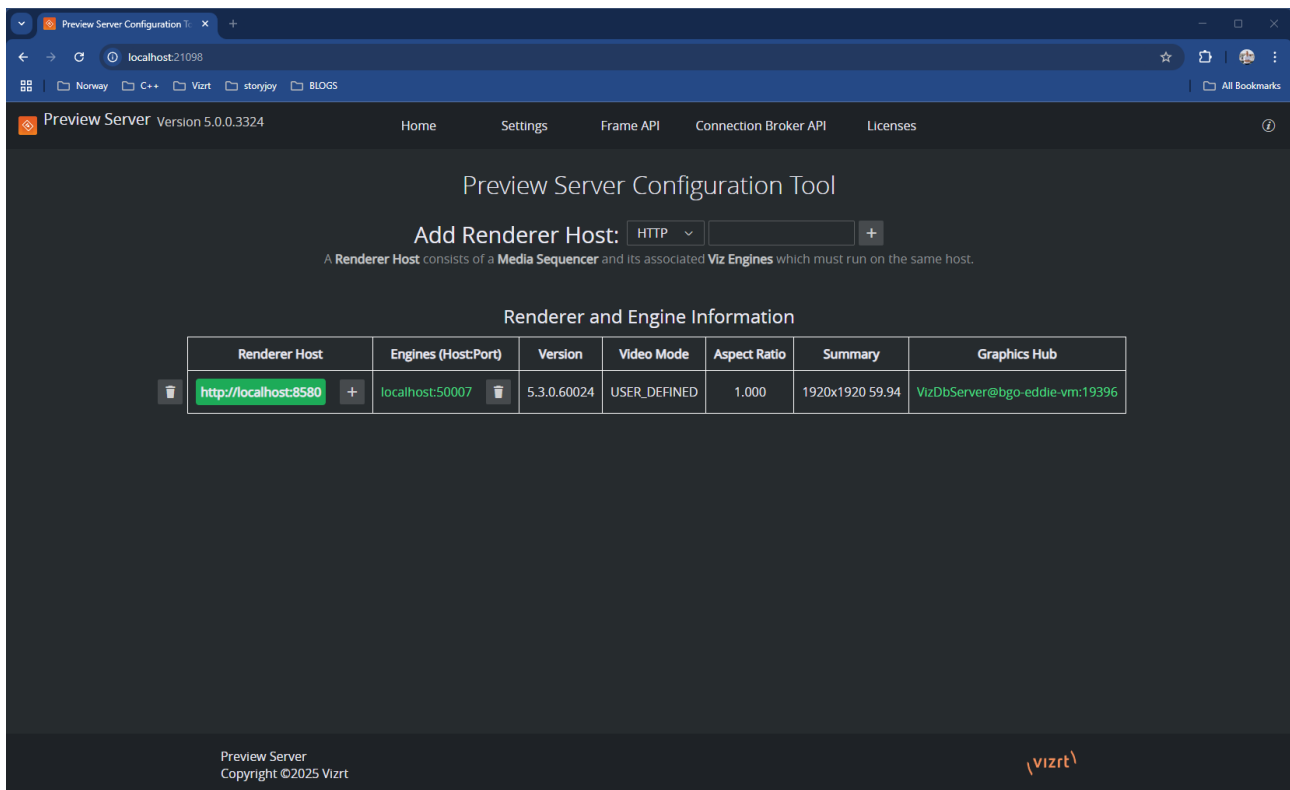
This section covers the following topics:

- [Web Interface](#)
- [Common Procedures](#)

5.1 Web Interface

Preview Server's web interface, allows users to access features.

For details on how to open this interface, see [Accessing the Preview Server Web Interface](#).



These are the menu options of the web interface:

- [Home](#)
- [Settings](#)
- [Frame API](#)
- [Connection Broker API](#)
- [Licenses](#)
- [About](#)

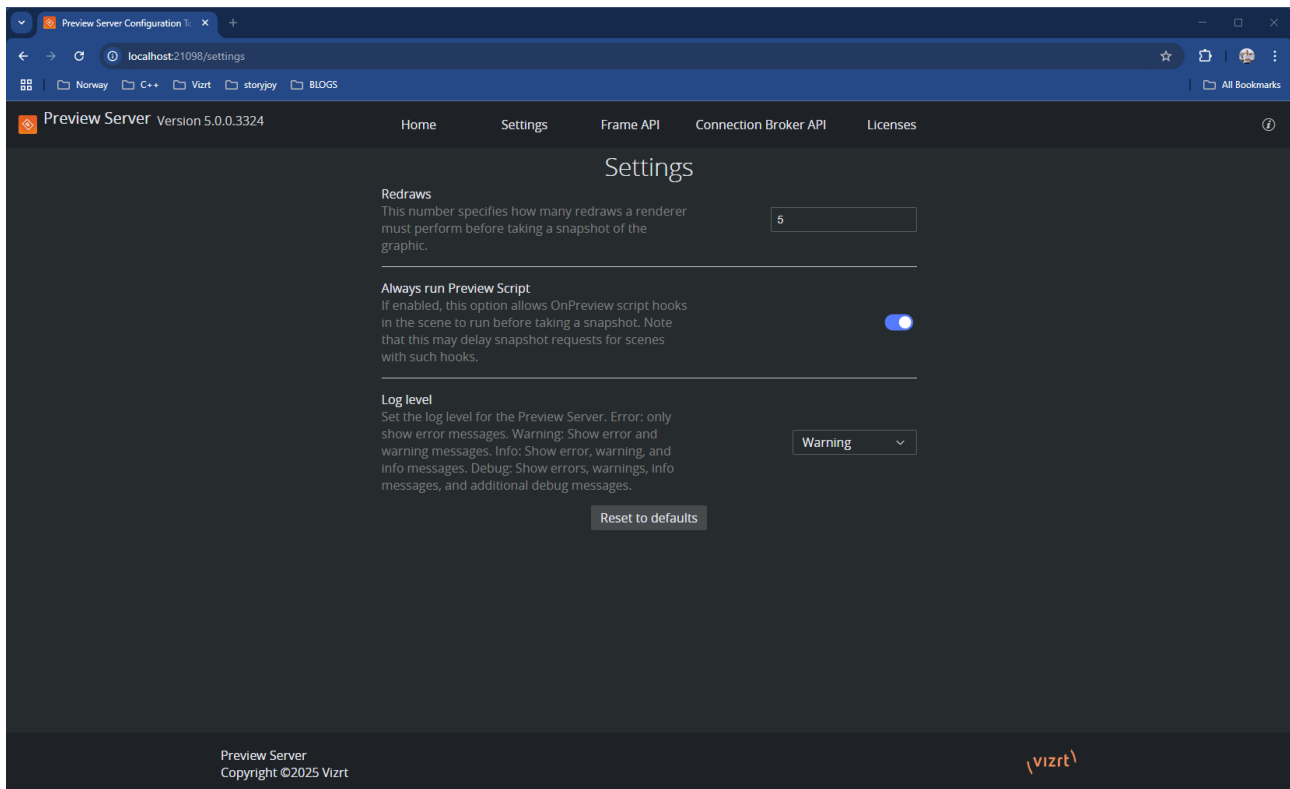
5.1.1 Home

You can add, monitor and remove Media Sequencers and Viz Engines from a pool of renderers on the **Home** page. Preview Server can interact with one or more Media Sequencers and Viz Engines.

Note: The term **Renderer Host** is widely used in the Web Interface, it refers to a physical or virtual machine that runs an instance of Media Sequencer along with a Viz Engine attached to it.

5.1.2 Settings

The **Settings** page allows users to modify certain Preview Server settings that are crucial for the rendering process. These changes do not affect the behavior of the Windows Server itself and therefore do not require a service restart.

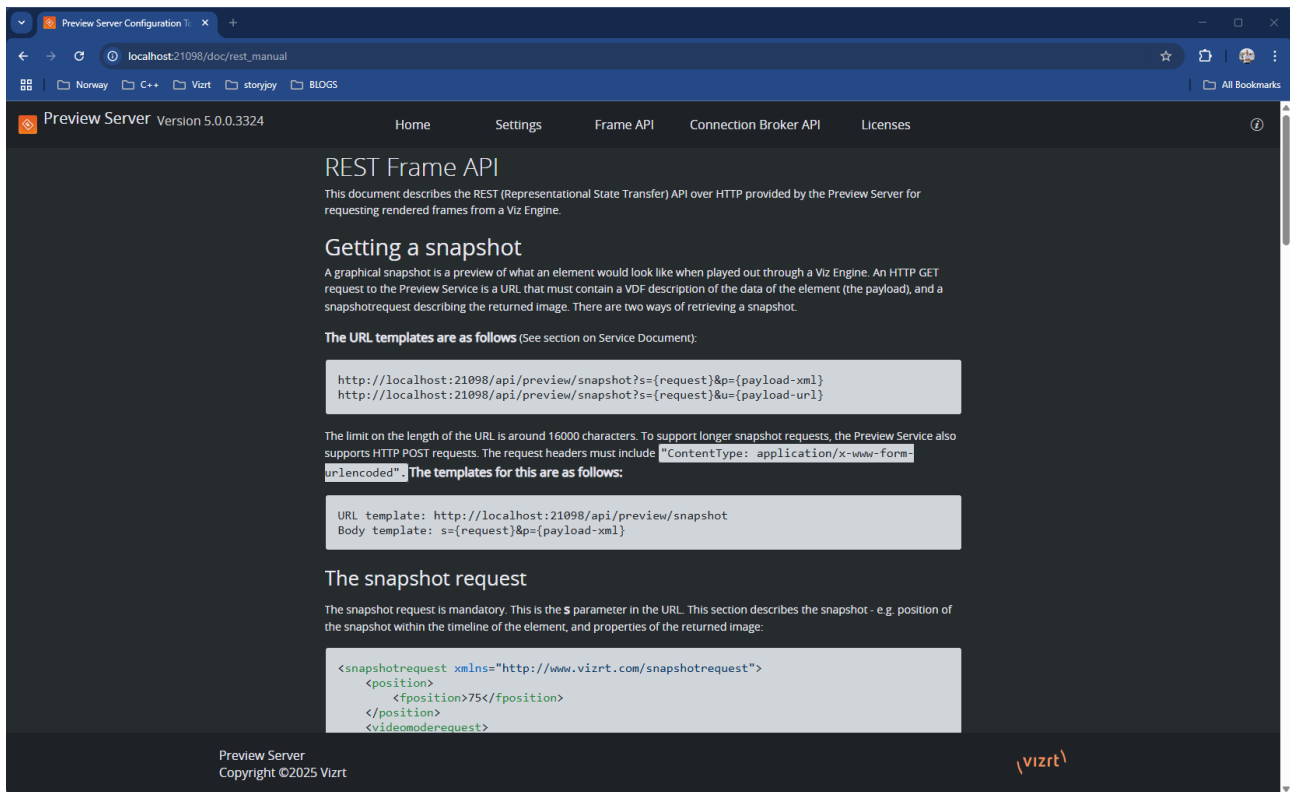


5.1.3 Frame API

Click the **Frame API** link in the Preview Server web interface to open the **REST Frame API** documentation, which describes the REST (Representational State Transfer) API over HTTP, provided by Preview Server, to request rendered frames from a Viz Engine.

The document also lists the *Resources* and *Data Types* that are used in the interface.

Example: `http://<hostname>:21098/doc/rest_manual`.



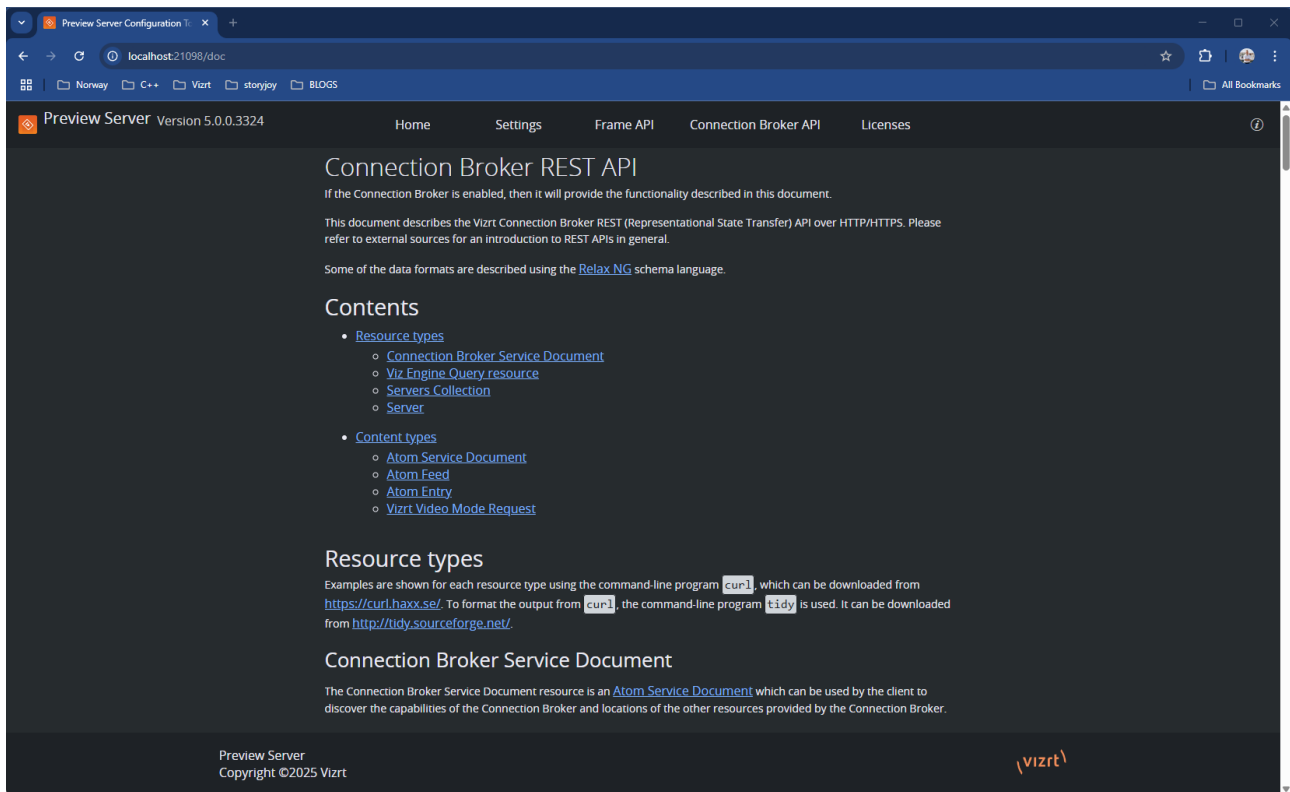
5.1.4 Connection Broker API

Click the **Connection Broker API** link in the Preview Server web interface to open the **Connection Broker REST API** documentation, which describes the REST (Representational State Transfer) API over HTTP, provided by the Connection Broker for requesting available Viz Engine instance for rendering.

The document also lists the *Resources* and *Content Types* that are used in the interface.

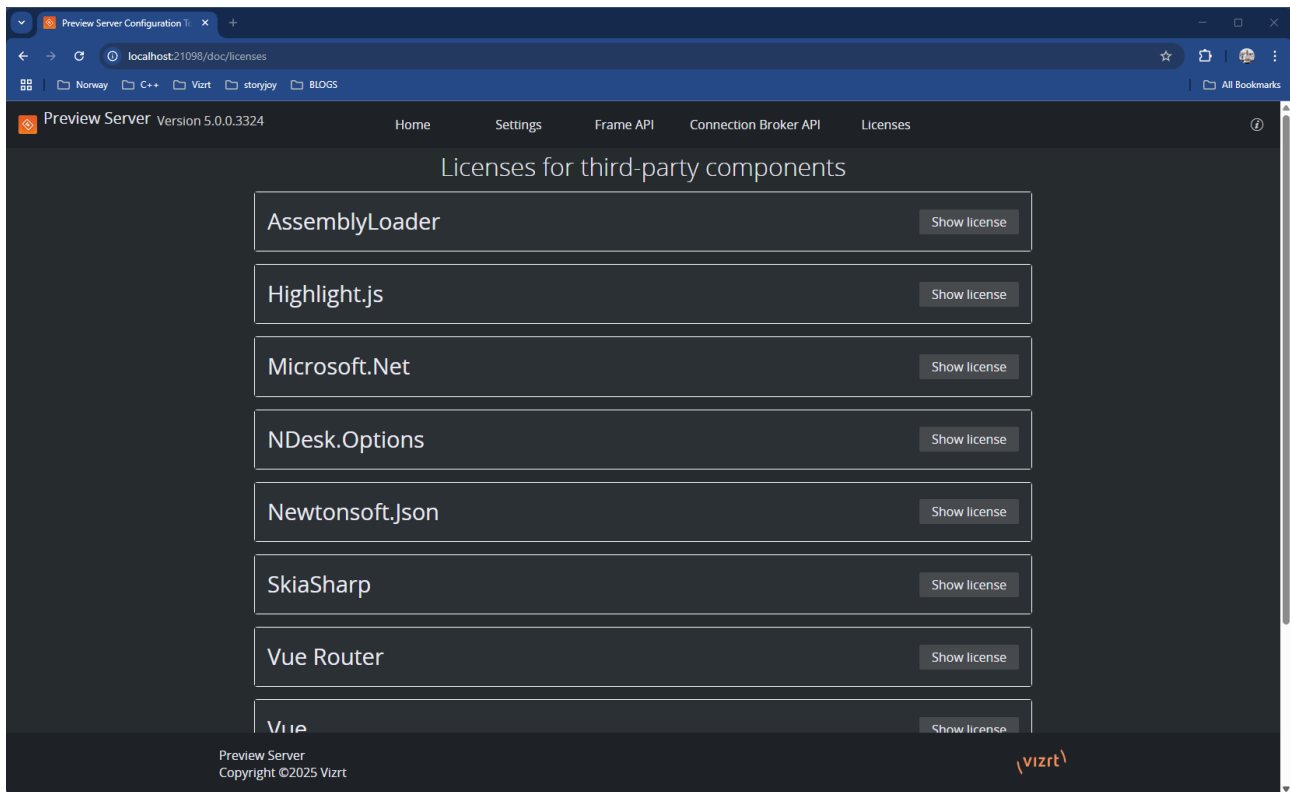


Example: `http://localhost:21098/doc .`



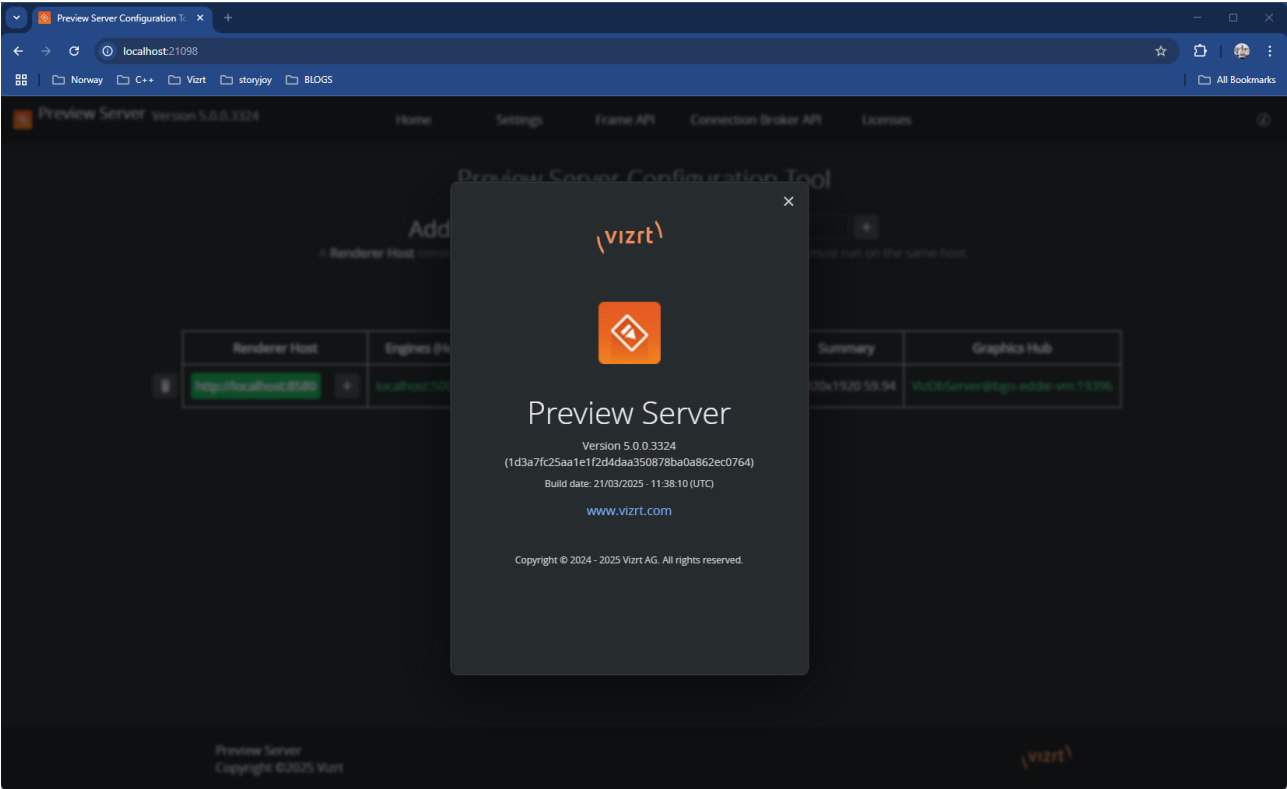
5.1.5 Licenses

The **Licenses** page displays the license texts of third-party components used in Preview Server.



5.1.6 About

Clicking the **Information** icon in the web interface opens the **About Preview Server** window, containing information on the installed version of Preview Server.



See Also

- [Viz Engine Configuration](#)

5.2 Common Procedures

This section covers the following topics:

- [Startup and Shutdown](#)
 - [Accessing the Preview Server Web Interface](#)
 - [See the Status of the Preview Server Service](#)
 - [Starting the Preview Server Service](#)
 - [Stopping the Preview Server Service](#)
- [Viz Engine Configuration](#)
 - [Adding a Renderer Host \(Media Sequencer\)](#)
 - [Removing a Renderer Host \(Media Sequencer\)](#)
 - [Adding a Viz Engine to Renderer Host \(Media Sequencer\)](#)
 - [Removing a Viz Engine](#)

5.2.1 Startup and Shutdown

Accessing the Preview Server Web Interface

- In the **Start** menu search for **Preview Server Config**, or
- Double-click the **Preview Server Config** desktop icon (if a shortcut is created), or
- Navigate to Preview Server's web interface in a browser, using the hostname of the machine running Preview Server, and port 21098.

 **Example:** `http://<hostname>:21098/` .

 **Note:** The default web browser is used when opening Preview Server's web interface from the Windows Start menu or desktop.

See the Status of the Preview Server Service

1. In the **Start** menu search for **Services**.
2. Scroll down to the **Vizrt Preview Server** service.
The status of the Preview Server service is listed in the **Status** column.

Starting the Preview Server Service

1. Perform the steps in [See the Status of the Preview Server Service](#).
2. Make sure the Preview Server status is empty.
3. Right-click the Preview Server entry in the list.
4. From the menu that appears, click **Start**.

Note: Upon installation, the Preview Server service starts and the startup type is set to *Automatic*. This means the service starts every time the operating system is started, and continues to run in the background as long as Windows is running. However, if the Preview Server service is manually stopped, the service can be restarted by following the steps in this procedure.

Stopping the Preview Server Service

1. Perform the steps in [See the Status of the Preview Server Service](#).
2. Make sure the Preview Server status is set to *Started*.
3. Right-click the Preview Server entry in the list.
4. From the menu that appears, click **Stop**.

5.2.2 Viz Engine Configuration

Adding a Renderer Host (Media Sequencer)

1. Open the Preview Server [Home](#) page.
2. Select the protocol (HTTP or HTTPS) to be used for communication between Preview Server and the Renderer Host (Media Sequencer). Then enter the **Renderer Host** hostname or IP address in the text box.

Add Renderer Host:

A **Renderer Host** consists of a **Media Sequencer** and its associated **Viz Engines** which must run on the same host.

Info: The default Media Sequencer port is used for communication between Preview Server and the Renderer Host (Media Sequencer), 8543 for HTTPS and 8580 for HTTP.

3. Click the “+” button. The newly added instance of the Renderer Host (Media Sequencer) should then appear in the main table.

Renderer and Engine Information						
	Renderer Host	Engines (Host:Port)	Version	Video Mode	Aspect Ratio	Summary
	http://localhost:8580 +	No engines configured for this renderer host				

Note: The background color around the Renderer Host name indicates its availability, specifically the availability of Media Sequencer running on that host.

- Green: Media Sequencer is available on the expected port, and Preview Server successfully established a connection.
- Red: Media Sequencer is not available on this host.

Removing a Renderer Host (Media Sequencer)

1. Open Preview Server’s [Home](#) page.

- Click the **trash can** next to a Renderer Host entry.

Renderer and Engine Information

Renderer Host	Engines (Host:Port)	Version	Video Mode	Aspect Ratio	Summary	Graphics Hub
 http://localhost:8580 	No engines configured for this renderer host					

Adding a Viz Engine to Renderer Host (Media Sequencer)

- Open Preview Server's [Home](#) page.
- Locate the Renderer Host in the main table to which you want to add a new Viz Engine. If no Renderer Host is available, add a new one first.
- Click the **“+”** button next to the Renderer Host entry.
- In the popup window, specify the hostname and port number where the Viz Engine is running.

Add new engine to renderer localhost

New engine

OK
Cancel

Note: If port is not specified, Viz Engine is added with the default port number 50007. When using Viz Engines in a so-called dual channel setup, it is possible to run multiple instances of Viz Engine on a single host. Each instance uses a unique port, which means that only two Viz Engines should be considered duplicates when both the hostnames and port numbers are identical.

- Click the **OK** button to confirm the addition.
- The new Viz Engine should be displayed in the main table.

Renderer and Engine Information							
Renderer Host	Engines (Host:Port)	Version	Video Mode	Aspect Ratio	Summary	Graphics Hub	
 http://localhost:8580 	localhost:50007 	5.3.0.60024	USER_DEFINED	1.000	1920x1920 59.94	VizDbServer@bgo-eddie-vm:19396	

Info: The font color of the Viz Engine name indicates its availability.

- Green: Viz Engine is available on the expected port, and Media Sequencer successfully established a connection with it.
- Red: The Viz Engine is not available on this host.

Removing a Viz Engine

- Open the Preview Server [Home](#) page.
- Click the **trash can** next to a Viz Engine entry.

Renderer and Engine Information						
Renderer Host	Engines (Host:Port)	Version	Video Mode	Aspect Ratio	Summary	Graphics Hub
 http://localhost:8580 	localhost:50007 	5.3.0.60024	USER_DEFINED	1.000	1920x1920 59.94	VizDbServer@bgo-eddie-vm:19396

6 Version API

You can retrieve the version of Preview Server, along with additional information (for example, build number, git commit ref, etc.), by sending a `GET` request to `<hostname>:<port>/version`.

An example is shown below using PowerShell:

```
(Invoke-WebRequest http://localhost:21098/version).Content
```

This returns a JSON object:

```
{
  "version": "5.0.0",
  "commit_ref": "1d3a7fc25aa1e1f2d4daa350878ba0a862ec0764",
  "build_date": "21/03/2025",
  "build_time": "11:38:10 (UTC)",
  "build_number": "3324"
}
```

7 Troubleshooting

If you do not find answers to your issue here, please contact Vizrt support at <https://www.vizrt.com/support>.

7.1 Services Won't Start After Installing the Latest Preview Server

While earlier versions of Preview Server had reportedly issues with Apple's Bonjour installed, we were unable to recreate them with the latest build. If problems occur, uninstalling Bonjour may help (TRIO-3541).

7.2 Known Limitations

- We are unable to detect duplicated Viz Engines, therefore, it is possible to add the same Viz Engine multiple times on the same Media Sequencer if you use different host specifications, for example `localhost` and `127.0.0.1`. In this case, the Viz Engine has double the weight in load distribution.
- Snapshots of scenes with no key defined, result in empty images (TRIO-4084).
- Adding video mode Viz Engines as renderers, could produce incorrect results in scenes using ControlVideo (TRIO-4084).

See Also

- [Preview Server Release Notes](#)