

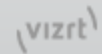


Graphics Plugin Administrator Guide

Version 2.3



Graphics Plugin





Copyright © 2020 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.

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

2020/06/30

Contents

1	Introduction.....	5
1.1	Feedback	5
1.2	Submitting a Support Request.....	5
1.2.1	Submitting a Support Request.....	5
2	Requirements and Recommendations	6
2.1	Supported Operating Systems	6
2.1.1	Windows-based Systems	6
2.1.2	Mac-based Systems.....	6
2.2	Supported NLE Systems	7
2.2.1	Adobe Premiere Pro	7
2.2.2	Avid Media Composer and Avid NewsCutter.....	7
2.2.3	Grass Valley EDIUS	7
2.2.4	Apple Final Cut Pro	8
2.2.5	Quantel Rio.....	8
2.3	Software Requirements	8
2.3.1	Client Software Requirements	8
2.3.2	Server Software Requirements.....	9
2.3.3	Software Limitations	10
2.4	Network Requirements.....	11
2.4.1	Port Numbers	11
3	Installation and Configuration.....	13
3.1	System Components	13
3.2	Graphics Rendering	14
3.2.1	Setting Up a Viz Engine for Graphics Rendering	14
3.3	Connection Broker	14
3.3.1	Adding a Viz Engine.....	15
3.3.2	Removing a Viz Engine.....	16
3.3.3	Modifying the Ping Interval from the Connection Broker to a Viz Engine.....	16
3.4	Windows-based NLE Systems.....	18
3.4.1	Typical Viz Trio Setup	18
3.4.2	Typical Viz Pilot Setup	19
3.4.3	Installation.....	19
3.4.4	Configuration	19
3.4.5	Advanced Configurations	20


3.5	Mac-based NLE Systems	20
3.5.1	Prerequisites.....	21
3.5.2	Installation.....	21
3.5.3	Configuration	22
3.5.4	Allow Mac NLE to use Editors on a Windows Remote Host	22
3.5.5	Uninstall	22
4	Tuning the System	24
4.1	Introduction to Tuning the NLE System	24
4.2	File Compression	25
4.3	Prefetching	26
4.4	Concurrent Graphics Plugin Users	27
4.5	System Processing	27
5	Troubleshooting and Logging	28
5.1	Log Files	28
5.2	Questions and Answers.....	30
5.2.1	General.....	30
5.2.2	Using the NLE System	31
5.2.3	Configuring the NLE System.....	32
5.2.4	Tuning the NLE System	32
5.2.5	Rendering with Viz Engine	33

1 Introduction

The Graphics Plugin integrates Vizrt's graphics workflow into the *Non-Linear Editing (NLE)* workflow.

Use your normal workflow using their preferred NLE system, while adding real-time rendered Vizrt graphics.

The Graphics Plugin integrates with a variety of [NLE systems](#): Adobe Premiere Pro, Apple Final Cut Pro, Avid Media Composer, Quantel Rio and Grass Valley Edius. Although the Graphics Plugin versions are tailored to the specific platforms, the overall setup and user interface is the same.

 **Note:** The Graphics Plugin provides plugins for Adobe Premiere Pro, Avid Media Composer and Edius.


Viz Trio, Viz Pilot Edge, Viz Pilot, and the Graphics Plugin Editor are all interfaces that let the NLE user interact with the Vizrt graphics workflow and edit graphics elements. The Graphics Plugin supports NLE systems on both Windows and Apple platforms. Viz Trio and Viz Pilot Edge and Viz Pilot are available for [Windows-based NLE Systems](#). Viz Pilot Edge and the Graphics Plugin Editor is available for the [Mac-based NLE Systems](#).

1.1 Feedback

We welcome your feedback and suggestions regarding Vizrt products and this documentation. Please contact your local Vizrt customer support team at <http://www.vizrt.com>.

1.2 Submitting A Support Request

The Graphics Plugin is (despite the simple interface) a complex integration of several products. The Graphics Plugin is hiding most of this complexity to the users and to a certain extent also to the people who set things up. When completing a support request, please add as much information as possible. See [Log Files](#) for information about how to reproduce the problem for the specific Graphics Plugin installed.

 **Note:** Any issues should be registered using Vizrt's Support Portal. If you do not have access to the portal, please send a report to a local Vizrt representative or directly to support@vizrt.com.

1.2.1 Submitting a Support Request

1. Go to www.vizrt.com.
2. Click **Menu > Support > Get Support**.
3. Click on **Login to the Vizrt Global Support portal** to submit your support request.

2 Requirements And Recommendations

This section covers the following topics:

- [Supported Operating Systems](#)
 - [Supported NLE Systems](#)
 - [Software Requirements](#)
 - [Network Requirements](#)
-

2.1 Supported Operating Systems

The Graphics Plugin supports two operating systems; Microsoft Windows and Apple Macs using OS X 10.6 Snow Leopard, or later. The basic setup is in principle the same; however, there are some differences in how they are used, and what software is used in addition to the plugin.

Both setups need a Viz Engine for graphics rendering, and a database that stores the data for each graphics template or data element. The database can either be a Media Sequencer or a Pilot database.

The main components used with the Graphics Plugin are:

- *Viz Engine* is a real-time graphics renderer, and at the core of the Vizrt graphics workflow.
- *Connection Broker* performs connection broking, providing all Graphics Plugin system users with a single connection point for graphics rendering.
- *Media Sequencer* is a framework for defining and executing media elements on a Viz Engine. The media elements are defined in a tree-based schedule that the Media Sequencer interprets, but for the Graphics Plugin, it is simply used as a storage solution for Viz Trio, which is used by the Graphics Plugin Editor.
- *Viz Pilot database* is an Oracle 10g or 11g database that stores template data and references to the template graphics. The Viz Pilot database is mainly used in conjunction with Viz Pilot.

2.1.1 Windows-based Systems

For Windows users, there are three editors available for adding graphics to a video clip: Viz Trio, Viz Pilot and Viz Pilot Edge. Viz Trio is bundled with the Graphics Plugin installer, whereas Viz Pilot must be installed separately, as it connects to the Viz Pilot database for template data and graphics. Viz Pilot Edge is a web application which is accessed through the Graphics Plugin by adding the URL in the Configuration Tool.

2.1.2 Mac-based Systems

For Mac users, there are two editors available: Viz Pilot Edge and the plugin's own built-in Graphics Plugin Editor. The editor is capable of reading Viz Trio shows stored on the Media Sequencer and remote previewing the Viz Engine graphics as well as editing the text and image properties of graphic scenes.

The Mac-based Graphics Plugins are unable to control or use Viz Trio and Viz Pilot remotely as a user interface, hence the Graphics Plugin Editor.

See the [Graphics Plugin User Guide](#) for more information on how to use the different editors.

2.2 Supported NLE Systems

The list below currently supported NLE systems within the scope of this document. Other versions are not officially supported unless specifically mentioned as part of this document.

The following systems are supported:

- [Adobe Premiere Pro](#)
- [Avid Media Composer and Avid NewsCutter](#)
- [Grass Valley EDIUS](#)
- [Apple Final Cut Pro](#)
- [Quantel Rio](#)

2.2.1 Adobe Premiere Pro

The **Graphics Plugin for Adobe** has been tested against, and supports the following versions:

- Premiere Pro CC 2019 (13)
- Premiere Pro CC 2018 (12)
- Premiere Pro CC 2017 (11)

This plugin works on both Windows and Mac machines.

2.2.2 Avid Media Composer and Avid NewsCutter

The **Graphics Plugin for Avid** supports the following versions:

- Avid Media Composer 8.9
- Avid Media Composer 2018

This plugin works on both Windows machines and Mac machines.

The following systems support the Avid Extensions (AVX) 2.0 System Development Kit (SDK):

- Avid Digital Studio (DS)
- Media Composer (Older Release and Post Adrenaline HD release that can run software only), Media Composer Adrenaline, and Media Composer Adrenaline HD
- NewsCutter, NewsCutter FX, NewsCutter XP, and NewsCutter Adrenaline (FX)
- Symphony and Symphony Nitris
- Xpress, Xpress Pro, and Xpress Pro HD

2.2.3 Grass Valley EDIUS

The **Graphics Plugin for EDIUS** supports the following versions:

- EDIUS 8
- EDIUS 9

This plugin works only on Windows machines.


2.2.4 Apple Final Cut Pro

The **Graphics Plugin for Final Cut** works together with Apple Final Cut Pro 6 and 7 (available in Final Cut Studio 2 and 3, respectively), and Final Cut Pro X. This plugin works on Mac-machines only.

 **Note:** There are separate installers for versions 6+7, and X.

2.2.5 Quantel Rio

The **Graphics Plugin for Quantel** works together with Quantel Rio 4.3, or later. This plugin works on Windows machines only.

 **Note:** Version 2.0 of the Graphics Plugin supports earlier versions of Quantel QEffects/Rio (version 4.2).

2.3 Software Requirements

This section covers the following topics:

- [Client Software Requirements](#)
- [Server Software Requirements](#)
- [Software Limitations](#)

2.3.1 Client Software Requirements

The Graphics Plugin has been tested with the following software:

Adobe 2.2

- Graphics Plugin for Adobe on Windows:
 - Microsoft Windows 7, 64-bit or later.
 - Adobe Premiere Pro CC 2017 (11) or later.
 - Viz Pilot 6.0.0 or later.
- Graphics Plugin for Adobe on Mac:
 - Apple macOS 10.14.1 or later.
 - Adobe Premier Pro CC 2017 (11) or later.

Avid 2.2

- Graphics Plugin for Avid on Windows:
 - Microsoft Windows 7, 64-bit or later.
 - Avid Media Composer 8.9 or later.
 - Viz Pilot 6.0.0 or later.
- Graphics Plugin for Avid on Mac:
 - Avid Media Composer 8.9 or later.
 - Apple macOS 10.14.1 or later.

EDIUS 2.2

- Microsoft Windows 7, 64-bit or later.
- EDIUS 8 or 9
- Viz Pilot 6.0.0 or later.

Final Cut 2.0

- Final Cut Pro 7.
- Apple macOS 10.10.3 on Intel Hardware.

2.3.2 Server Software Requirements

The server components used by Graphics Plugin have different system requirements. A summary is provided here, but refer to the individual release notes and manuals for details. Please note that it is not recommended to install all components on a single server.

- **Connection Broker 4.0**
 - Optional, it is still possible to specify a specific Viz Engine directly.
 - Requires Microsoft .NET 4.5.
 - It can be installed on the same server as a Media Sequencer or a Viz Engine, or on a separate computer.
 - Connection Broker 1.0.0 can be used, but some of the new features in Viz Engine 3.5 will not work. This may impact performance.
 - Connection Broker 1.1 can be used but some of the new features in Viz Engine 3.7 will not work.
- **Media Sequencer 2.0.3 and 3.0.0**
 - Requires Microsoft .NET 4.0.
 - Connecting multiple Graphics Plugin workstations to the same Media Sequencer makes it easier to share graphics elements between Graphics Plugin clients.
 - Elements saved to an MSE from Viz Trio on Windows can be used from the Graphics Plugin Editor for Mac OS X, and vice versa.
 - Installing it on a Viz Engine server is not recommended since high memory usage can interfere with Viz Engine performance.
- **Viz Pilot 6.0.0, Pilot Data Server 6.0.0**

- Used by Graphics Plugin Editor (the Mac OS X editor) as a source for images to be used in graphics.
- The components are optional. If they are not used, Graphics Plugin Editor can still use images from the Viz Engine.
- **Viz Engine 3 / Viz Artist 3**
 - Viz Engine 3.7.0 and 3.8.0.
 - If you need to use older versions the recommended versions are 3.2.8, 3.3.1 and 3.5.1
 - Viz Engine 3.7 or newer is required to take advantage of the new aspect independent frame fetching.
 - Viz Engine 3.5 or newer is required to take advantage of the new custom resolution frame fetching and anti-aliasing.
 - Viz Engine 3.5.0, 3.3.0, and 3.2.7 or earlier have some known issues that can cause (rare) rendering artifacts when multiple NLE clients are using the same engine. We recommend Viz Engine 3.2.8, 3.3.1 and 3.5.1 or newer.
 - It is highly recommended to have a fast (Gigabit) network connection between each NLE workstation and Viz Engine.
 - This version of Graphics Plugin only supports UTF-8 encoding for the Viz Engine.

2.3.3 Software Limitations

There are limitations to what a Graphics Plugin solution can do compared to a full SDI in-and-out Viz Engine renderer.

These are the general Graphics Plugin limitations on *both Windows and Mac*:

- Media Sequencer 1.19 and later, Viz Engine 3, and Connection Broker 3.0 and later, do not announce their presence on the computer network. The Graphics Plugin configuration tool will therefore not automatically discover these services.


These are the specific Graphics Plugin limitations on *Windows only*:

- For Avid installations, not all Graphics Plugin versions prior to 1.5 are automatically removed when upgrading.

These are the specific Graphics Plugin limitations on *Mac only*:

- It is only possible to input text, change numeric values, and select images/geometries/fonts/materials. There is no support for input of boolean values. There is no support for editing ControlList/Tables or Scrolls. However, the editor can pick up Viz Trio pages created using a regular Viz Trio, and render them, even though they may have these field types.
- The Graphics Plugin Editor cannot create shows and import scenes, so you must use Viz Trio 2.6 or later to create shows and import scenes for the Graphics Plugin Editor to use.
- It is not possible to select images from a local path in the Graphics Plugin Editor.

Viz Trio are considerably slower on Windows 10 systems when connecting to a Graphic Hub which contains many scenes.

 **Note:** For more details, see the Graphics Plugin Release Notes.

2.4 Network Requirements

A Viz Engine can be shared by multiple simultaneous NLE clients; however, there is no recommended number of users for SD and HD as this usually involves a performance issue and not a fixed limitation.

The Windows and Mac-based NLE system setups typically operate on a single Viz Engine. This requires transportation of frames (images) over the network, so a fast network connection is recommended, preferably a gigabit switched network connection; however, a megabit (100 Mbit/s) connection will also work as the plugin supports lossless data compression using Portable Network Graphics (PNG) or Run-length Encoding (RLE) instead of uncompressed raw images.

 **Note:** RLE is only supported by Viz Engine 3.2.2, or later.

2.4.1 Port Numbers

The table below describes all listeners and port numbers, and applications that use the port numbers to establish a connection:

Listener	Port	Description
Viz Engine	6100	Default port used for graphics rendering.
Viz Trio	6210	Used by the Graphics Plugin to establish a connection to Viz Trio.
Viz Pilot	6220	Used by the Graphics Plugin to establish a connection to Viz Pilot.
Graphics Plugin Editor	6230	Used by the Graphics Plugin to establish a connection to the Graphics Plugin Editor.
Viz Pilot Edge	6230	Used by the Graphics Plugin to establish a connection to Viz Pilot Edge.
Graphics Plugin Config	6240	Used by the Graphics Plugin to establish a connection to the Graphics Plugin configuration tool.
Media Sequencer	8594	Port used by all clients that connect to the Media Sequencer.
Viz Engine	14300	Alternative port used to avoid conflicts with port 6100. Port 6100 is normally used by renderers that are On Air, hence, it is (for example when running Viz Pilot version 4) recommended to use another port.

Listener	Port	Description
Viz Engine (multiplexed)	50007	Used by clients that connect to a multiplexed Viz Engine.
	50008	50007: Isolated port. All connections to this port get their own session. The port is used by the Graphics Plugin to deliver frames to the host NLE system. For example: rendering or scrubbing of video clips with graphics. Using this port will also suppress bounding box commands. Cannot be used by Viz Pilot.
	50009	50008: Shared port. All connections from one single host share one session. Used by Viz Trio, Viz Pilot and Viz Pilot Edge to display preview frames. 50009: Fixed port. Same as shared port, except that allocated resources are never cleared from memory. Used by Viz Pilot 4.
Viz Engine (Dual Channel)	50107	Second default port for a dual channel Viz Engine.
	50108	50107: Isolated port. All connections to this port get their own session. The port is used by the Graphics Plugin to deliver frames to the host NLE system. For example: rendering or scrubbing of video clips with graphics. Using this port will also suppress bounding box commands. Cannot be used by Viz Pilot.
	50109	50108: Shared port. All connections from one single host share one session. Used by Viz Trio, Viz Pilot and Viz Pilot Edge to display preview frames. 50109: Fixed port. Same as shared port, except that allocated resources are never cleared from memory. Used by Viz Pilot 4.
Connection Broker	21098	Connection to the Connection Broker configuration interface (for example http://localhost:21098/).

3 Installation And Configuration

The following sections describe the different setup procedures. The sections follow the order of a new installation.

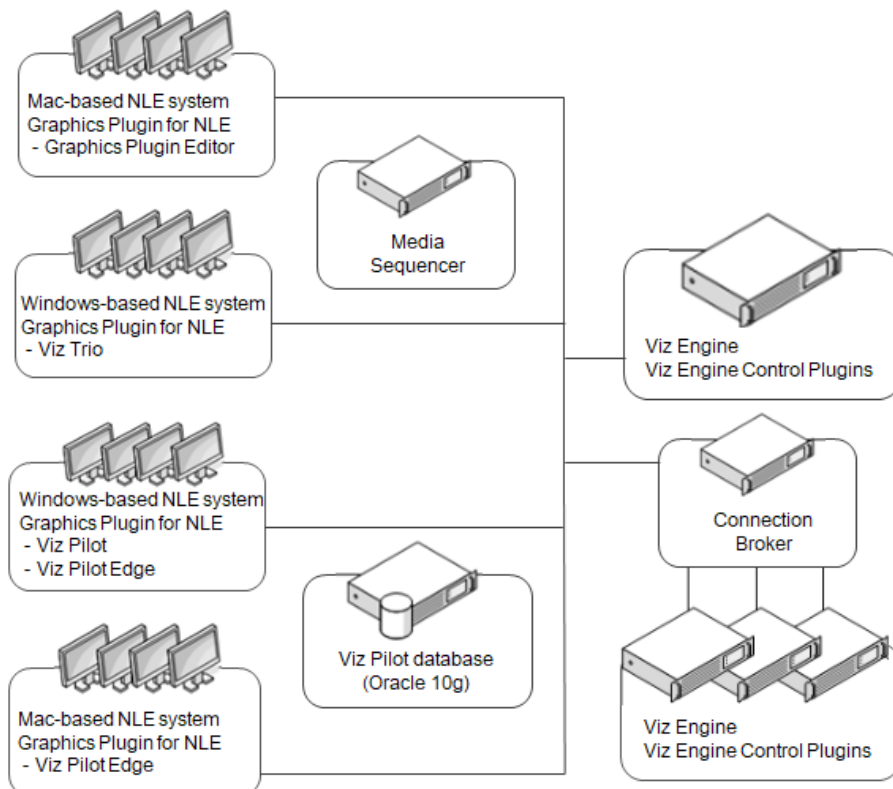
- [System Components](#)
- [Graphics Rendering](#)
- [Connection Broker](#)
- [Windows-based NLE Systems](#)
- [Mac-based NLE Systems](#)

3.1 System Components

All installations use a Viz Engine in combination with a database. The database is either a Media Sequencer, used by Viz Trio or Graphics Plugin Editor, or the Pilot Database used by Viz Pilot or Viz Pilot Edge.

A basic setup requires a Viz Engine, while a more advanced setup uses the Connection Broker to select engines from a pool of Viz Engines.

An overview of NLE systems using different editors and Viz Engine setups is shown below. See [Connection Broker](#) for more information about this setup.



Viz Engine must handle multiple simultaneous requests. In Viz Engine, the multiplexing functionality is a built-in feature.

The Graphics Plugin supports different NLE systems on both Windows and Mac, and the setup is basically the same for both platforms. The Graphics Plugin configuration tool is used to configure the plugin, Viz Engine, Media Sequencer and editors.

The Graphics Plugin Editor for Mac supports browsing and previewing of Viz Trio templates and pages stored on a Media Sequencer.

Viz Trio supports browsing and previewing of Viz Trio templates and pages stored on a Media Sequencer, while Viz Pilot Edge and Viz Pilot support browsing and previewing templates and data elements stored on the Viz Pilot database.

⚠ Note: Effects are bound to the editor that made them, even if you subsequently switch to a different editor. This binding happens the first time **Editor** is clicked for a specific effect, not when the effect is dragged down onto the timeline.

⚠ Note: NLE system-specific installations are outside the scope of this guide.

3.2 Graphics Rendering

In order for an NLE system to render graphics, a successful connection to a Viz Engine must be established. For the Viz Engine, off-screen rendering is the default rendering mode.

3.2.1 Setting Up a Viz Engine for Graphics Rendering

- Start the installer and select **VGA** mode.

3.3 Connection Broker

Connection Broker is a component that manages a pool of Viz Engines. Clients can query it for a Viz Engine that matches a specific video mode. Connection Broker provides load balancing, video mode matching and can be used to create a redundant renderer pool.

Open the interface using the address <http://localhost:21098/>. It's possible to open Connection Broker configuration from other machines. In order to do so, replace "localhost" with the actual hostname:

The screenshot shows the Connection Broker web interface. At the top, there are navigation tabs: Home (selected), Broker API, and Frame API. The version number 4.2.0.2625 is displayed in the top right corner. Below the navigation, there is a text input field labeled "VIZ ENGINE NAME OR IP ADDRESS" with an "Add New" button to its right. A green message below the input field reads "Renderer added: bgo-anjw:50007". Below this is a table with the following columns: NAME OR IP ADDRESS, VIDEO MODE, ASPECT, VERSION, GRAPHIC HUB, and STATUS. The table contains one row with the following data: bgo-anjw:50007, unknown, 1.600, v3.9.1.71532, DEMO-GH2, and Responsive. A trash icon is visible at the end of the row. At the bottom of the interface, there is a footer with the text "Connection Broker", "Copyright 2012–2018 · Vizrt AG · All rights reserved", and the Vizrt logo.

Note: Preview Server is bundled together with Connection Broker. Although both components have their own service document URL, they share the same pool of Viz Engines. If both services are used extensively, the load on the joint pool may be higher than recommended.

This section describes the following procedures:

- [Adding a Viz Engine](#)
- [Removing a Viz Engine](#)
- [Modifying the Ping Interval from the Connection Broker to a Viz Engine](#)
 - [Code Example](#)

3.3.1 Adding a Viz Engine

1. In the Connection Broker web interface, enter the Viz Engine hostname or IP address in the text box.

This is a close-up screenshot of the input field from the previous image. The text "VIZ ENGINE NAME OR IP ADDRESS" is displayed above the input box. The input box contains the text "preview". To the right of the input box is the "Add New" button.


2. Click **Add New**.

Note: If not specified, the Viz Engine is added with default port 50010. When using Viz Engines in a so-called dual channel setup, multiple instances of Viz Engine can run on a single host. Each instance uses a unique port, which means that two Viz Engines should only be considered duplicates when both the hostnames and port numbers are identical. When referring to Viz Engines in a dual channel setup, type the port number after the hostname (for example, *VizEngine1:51007*).

Viz Engine host port number restrictions when previewing templates on Editor: Previewing templates on Graphics Plugin Editor, Viz Trio, Viz Pilot or Viz Pilot Edge requires the Viz Engine port to be 50007 or 50107. This will match with preview ports 50008 and 50108. See [Network Requirements](#) for port numbering information.

3.3.2 Removing a Viz Engine

In the Connection Broker web interface, click the **trash can** next to the Viz Engine entity that is to be removed.

NAME OR IP ADDRESS	VIDEO MODE	ASPECT	VERSION	GRAPHIC HUB	STATUS	
bgo-anjw:50007	unknown	1.600	v3.9.1.71532	DEMO-GH2	Responsive	

3.3.3 Modifying the Ping Interval from the Connection Broker to a Viz Engine

1. Open the *Connection Broker.exe.config* file, typically located at *C:\Program Files (x86)\Vizrt\Connection Broker*.
2. In the configuration file, set a new value for the *VizEnginePollIntervalMilliseconds* option, which controls the interval for how often the Connection Broker polls each Viz Engine.
3. Set a new value for the *VizEngineTimeoutMilliseconds* option, which controls the timeout in milliseconds when trying to connect to a Viz Engine.

Code Example

```

<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>
    <sectionGroup name="applicationSettings" type="System.Configuration.Applicati
onSettingsGroup, System, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089">
      <section name="Vizrt.Frames.Properties.Settings" type="System.Configurati
on.ClientSettingsSection, System, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089" requirePermission="false" />
    </sectionGroup>
  </configSections>
  <applicationSettings>
    <Vizrt.Frames.Properties.Settings>
      <setting name="VizEnginePollIntervalMilliseconds" serializeAs="String">
        <value>10000</value>
      </setting>
      <setting name="VizEngineTimeoutMilliseconds" serializeAs="String">
        <value>10000</value>
      </setting>
      <setting name="CacheMaxAge" serializeAs="String">
        <value>86400</value>
      </setting>
      <setting name="VizEngineCleanupIntervalMilliseconds" serializeAs="String">
        <value>60000</value>
      </setting>
      <setting name="UseRunLengthEncoding" serializeAs="String">
        <value>False</value>
      </setting>
    </Vizrt.Frames.Properties.Settings>
  </applicationSettings>
  <system.serviceModel>
    <services>
      <service behaviorConfiguration="Vizrt.Frames.AtomBrokerService" name="Viz
rt.Frames.AtomBrokerService"></service>
    </services>
    <behaviors>
      <serviceBehaviors>
        <behavior name="Vizrt.Frames.AtomBrokerService">
          <serviceDebug includeExceptionDetailInFaults="true"
httpHelpPageEnabled="false" httpsHelpPageEnabled="false" />
        </behavior>
      </serviceBehaviors>
    </behaviors>
  </system.serviceModel>
  <uri>
    <schemeSettings>

```

```

    <add name="http" genericUriParserOptions="DontUnescapePathDotsAndSlashes"
  />
    <add name="https" genericUriParserOptions="DontUnescapePathDotsAndSlashes
" />
  </schemeSettings>
</uri>
<startup>
  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.2" />
</startup>
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
    <dependentAssembly>
      <assemblyIdentity name="System.Web.Http" publicKeyToken="31bf3856ad36
4e35" culture="neutral" />
      <bindingRedirect oldVersion="0.0.0.0-5.2.2.0" newVersion="5.2.2.0" />
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="Microsoft.Owin" publicKeyToken="31bf3856ad364
e35" culture="neutral" />
      <bindingRedirect oldVersion="0.0.0.0-3.0.0.0" newVersion="3.0.0.0" />
    </dependentAssembly>
    <dependentAssembly>
      <assemblyIdentity name="System.Net.Http.Formatting" publicKeyToken="3
1bf3856ad364e35" culture="neutral" />
      <bindingRedirect oldVersion="0.0.0.0-5.2.2.0" newVersion="5.2.2.0" />
    </dependentAssembly>
  </assemblyBinding>
</runtime>
</configuration>

```

3.4 Windows-Based NLE Systems

Installing the Graphics Plugin for Windows-based NLE systems, the Configuration Tool and an NLE version of Viz Trio are also installed.

Viz Trio is an advanced character generator that is used to produce content for use with a Windows-based NLE system. Viz Trio requires a Viz Engine and a Media Sequencer, see illustration in [System Components](#).


3.4.1 Typical Viz Trio Setup

- NLE machines
 - NLE system
 - Graphics Plugin (includes bundled Viz Trio)
- Media Sequencer
- Viz Engine
 - Graphic Hub with graphic scenes
 - *Optional*: Datapool plugins for Viz Engine

Viz Pilot and Viz Pilot Edge can also be used as the graphics editor for NLE systems, but is not bundled with the installation, and requires a Viz Engine and a Viz Pilot database. The Viz Pilot database (Oracle 10g) is where the templates, template data elements, and its associated data is stored.

3.4.2 Typical Viz Pilot Setup

- NLE machines
 - NLE system
 - Viz Pilot/Viz Pilot Edge
 - Graphics Plugin
- Viz Pilot database installation
 - Oracle 10g
- Viz Engine
 - Graphic Hub with graphic scenes
 - *Optional*: Datapool plugins for Viz Engine

 **Note:** Viz Engine and the Media Sequencer both run on Windows versions 7 or newer. Viz Engine 3.7, and later, do not support Windows XP. It is recommended to run the Media Sequencer on a separate machine. In smaller production environments, however, the Media Sequencer may run on the Viz Engine machine.

This section contains information on the following topics:

- [Installation](#)
- [Configuration](#)
- [Advanced Configurations](#)

3.4.3 Installation

Installing the Graphics Plugin for EDIUS, Adobe, Avid or Quantel on Windows, the Configuration Tool and the NLE version of Viz Trio are also installed.

Notes

- The installer requires a Windows user with administrator privileges.
- Adobe Premiere Pro (or Adobe AfterEffects) must be installed on the workstation before installing the Graphics Plugin for Adobe.
- Separate installers apply for 32- and 64-bit NLE systems. The 32-bit installer should be used together with Media Composer 5/NewsCutter 9 (or older), the 64-bit installer with Media Composer 6/NewsCutter 10.

3.4.4 Configuration

A Configuration Tool is installed with the Graphics Plugin.

Please see the Graphics Plugin User Guide on how to access the Configuration Tool in [Adobe](#), [Avid](#) and [Edius](#), and the [Configuration Tool](#) section for its settings.

3.4.5 Advanced Configurations

This section explains how to set up a default configuration for all users of a computer by adding a master settings file.

A master settings file is useful in large installations where a system administrator can collect common settings into a shared file, and have this picked up by all users on a computer or all users on a network (by putting the file on a Windows share). There is currently no facility in the Configuration Tool to edit such a shared settings file, but an advanced user should be able to get this to work by copying the `%APPDATA%\Vizrt\VizNLE\vizplugin.xml` file that the Configuration Tool produces into the desired shared location.

If a master settings file is loaded and a user settings file is loaded later, then the latter takes precedence in case of any conflicts. If for instance the loglevel defaults to "none", and the master settings file sets the loglevel to "report" and the user settings file sets the loglevel to "extreme", then the value of "extreme" is the one that is respected by the Graphics Plugin. But if the master settings file sets the Viz Engine hostname, and the user settings file is silent on this point, then the Viz Engine hostname in the master settings file is used.

Before loading the user settings, the master settings file is looked for and potentially loaded as follows:

- The Graphics Plugin asks the Windows registry for the *SettingsDir* string in the registry path under the listed registry locations.
- If the string is found, and its content turns out to be a path to a folder, then the filename *vizplugin.xml* is looked for at that location.
- If that file is present, it is loaded as a default configuration before loading the individual user settings file.

Registry paths:

Host system	Registry path
Adobe	<code>HKLM\SOFTWARE\[vizrt]\NLE\After Effects</code>
Avid	<code>HKLM\SOFTWARE\[vizrt]\NLE\AVX2</code>
Quantel	<code>HKLM\SOFTWARE\[vizrt]\NLE\Quantel</code>
Edius	<code>HKLM\SOFTWARE\[vizrt]\NLE\EDIUS</code>

3.5 Mac-Based NLE Systems

Installing the Graphics Plugin for Mac-based NLE systems, the Configuration Tool, and the Graphics Plugin Editor are also installed.

The Graphics Plugin Editor is a tool used to fetch graphics content for use with NLE systems on Mac. This editor enables Mac-users to incorporate graphics from Vizrt's Windows-based Viz Engine and Media Sequencer to the NLE workflow. A prerequisite is that a Viz Trio operator has created a show with graphics templates on the Media Sequencer. Once the show is available on the Media

Sequencer, the Graphics Plugin Editor is able to display the graphics such that the NLE user can select and edit the content and add it to the timeline.

Viz Pilot Edge is also available as an editor for Mac users, however, this setup requires a Pilot Database, see illustration in [System Components](#).

A typical setup is as follows:


- NLE machines
 - Mac-based NLE system (Adobe, Avid, or Final Cut)
 - Graphics Plugin (with Graphics Plugin Editor or Viz Pilot Edge)
- Viz Engine
 - Graphic Hub with graphic scenes
 - *Optional*: Datapool plugins for Viz Engine
- *Optional*: Connection Broker

This section covers the following topics regarding Mac-based NLE systems:

- [Prerequisites](#)
- [Installation](#)
- [Configuration](#)
- [Allow Mac NLE to use Editors on a Windows Remote Host](#)
- [Uninstall](#)

3.5.1 Prerequisites

Before installing the Graphics Plugin, a Media Sequencer and a Viz Engine must be running (on a Windows machine).

 **Note:** Viz Engine and the Media Sequencer must both run on Windows 7, or later. It is recommended to run the Media Sequencer separately on a Windows Server 2008 or Windows 7. In smaller production environments, however, the Media Sequencer may run on the Viz Engine machine.

The Media Sequencer should preferably contain a Viz Trio show to confirm that the connection is working properly; however, the Graphics Plugin Editor has status indicators indicating connections to Viz Engine, the host application (FCP or Premiere for Mac) and the Media Sequencer which should be sufficient information to confirm a correct setup.

3.5.2 Installation

The Configuration Tool and the Graphics Plugin Editor are also installed when installing the Graphics Plugin for Adobe, Avid, Final Cut or Final Cut X on Mac.

Notes

- The installer requires an OS X user with administrator privileges.
- Separate Graphics Plugin installers exist for Final Cut Pro 6+7 and Final Cut Pro X. Previous versions of the plugin cannot be used with Final Cut Pro X.

- The Graphics Plugin may not be shown in Final Cut Pro X if previous versions of the plugin are still installed. If this occurs, run the following commands in a terminal window:

```
sudo rm -rf /Library/Plug-Ins/FxPlug/viznle_plugin.fxplug
sudo rm -rf /Library/Application Support/Final Cut Pro System Support/Plugins/VizFcpNotifier.bundle
```

3.5.3 Configuration

A Configuration Tool is installed with the Graphics Plugin.


Please see the Graphics Plugin User Guide on how to access the Configuration Tool in [Adobe](#) and [Avid](#), and the [Configuration Tool](#) section for its settings.

3.5.4 Allow Mac NLE to use Editors on a Windows Remote Host

To allow a Mac-based NLE program to use other editors, such as Viz Pilot NLE, on a Windows remote host, do the following:

1. Start *C:\Program Files(x86)\vizrt\Pilot 6.1\VCPAxNle.exe*.
2. In Premiere/Avid on the Mac, create a new effect and click **Edit**.
3. NLE now starts communicating with Viz Pilot on the Windows host.
4. Open a data element, and click **Save**.
5. The effect may now be rendered in Premiere/Avid

If you have used Graphic Plugin Editor on your Mac before, you must delete the existing plugin. You may also need to exit and re-open Avid Media Composer in order for the setting to be applied.

 **Note:** Transition Logic scenes need to use Viz Pilot as the editor. They will not work if using Viz Trio/Graphic Plugin Editor.

3.5.5 Uninstall

The following table show which files to remove to uninstall the Graphics Plugin for the different Mac NLE applications.

Use finder, Terminal or some other tool to remove the following files:

Adobe	<ul style="list-style-type: none"> • <i>/Library/Plug-ins/MediaCore/viznle_plugin.plugin</i> • <i>/Library/Application Support/Adobe/Common/Plug-ins/[CS version]/MediaCore/viznle_plugin</i>. Plugin per installed CS version (for example CS5 and CS6)
Avid	<ul style="list-style-type: none"> • <i>/Library/Application Support/Avid/AVX2_Plug-ins/Vizrt/AvidGraphicsPlugin.avx/</i>

Final Cut	<ul style="list-style-type: none">· <i>/Library/Plug-ins/FxPlug/viznle_plugin.fxplug</i>· <i>/Library/Application Support/Final Cut Pro System Support/Plugins/VizFcpNotifier.bundle</i>· <i>/Library/Frameworks/VizCocoa.framework (used by older Graphics Plugin versions)</i>
Final Cut Pro X	<ul style="list-style-type: none">· <i>/Applications/FCPXGraphicsPlugin.app</i>

4 Tuning The System

The Graphics Plugin is a fairly complex product that involves several applications communicating over a network. These applications need to be individually robust and the communication between them need to be equally robust.

The first step of any installation is to see that everything works fine; hence, the plugin works out of the box in most cases. After everything is working, the next step is to configure it to work reliably in the work environment in which it will be used, which can be challenging.

This section covers the following topics:

- [Introduction to Tuning the NLE System](#)
- [File Compression](#)
- [Prefetching](#)
- [Concurrent Graphics Plugin Users](#)
- [System Processing](#)

See Also

- [Graphics Plugin Performance Tester](#)
-

4.1 Introduction To Tuning The NLE System

The Graphics Plugin is designed to give the NLE system direct access to preview, edit and apply graphics to the NLE project and consequently onto the video on the timeline. In addition, the Graphics Plugin is designed around the concept of one renderer being shared by several concurrent users on a fast network. The complexities of this design are hidden from the normal user and to a certain extent to the system administrator. Hence, there are multiple challenges with this approach when effects are rendered.

The following is a common workflow for most NLE systems:

- The NLE system demands a frame or field from the Graphics Plugin.
- The Graphics Plugin checks its configuration and tries to connect to the Viz Engine over the network.
- When connected, the Graphics Plugin queries Viz Engine for basic information, load the graphical scene in question, set the correct values in the scene, and stretch the graphics to fit the effect in the NLE project.
- The Graphics Plugin starts requesting fields and frames from Viz Engine depending on the NLE system requests.
- Viz Engine receives the requests and ensures that the right scene is loaded and move to the right frame or field and grab the graphics and send them back over the network.
- The Graphics Plugin receives the graphics and hands them back to the NLE system.
- The NLE system processes the data it received before requesting more graphics, thus repeating the steps above.

The procedure described above is simple to follow, it works fine and is easy to test. The problem with it is that it is not very fast. The most obvious problem is that NLE systems ask for a simple

frame or field at a time and blocks until it gets a reply. The Graphics Plugin sends the request to Viz Engine and has to wait for a reply before it replies back to the NLE system.

The first concern is the amount of data being sent, for example, a simple interlaced HD (1080i) field with full alpha is 4MB (1920*540*4 bytes) in size which takes some time to transmit. To render a five second interlaced HD clip at 50 frames a second means that more than a gigabyte of raw data has to be rendered and then processed by Viz Engine, sent over the network, received by the Graphics Plugin, and then moved into the buffers provided by the NLE system whereupon the NLE system needs to process the gigabyte of data.

Every network also has latency and is affected by other traffic on the network. Waiting for every request returning with 4MB of graphic will take time, and then waiting for the NLE system to process it further also means that there is a lot of inactivity. Therefore, the Graphics Plugin has a goal of whenever the NLE system asks for a frame or a field it will have it ready to cut the delay. The plugin tries to accomplish this by keeping the network and Viz Engine continuously busy. It attempts to predict what the NLE system will ask for next by asking Viz Engine in advance for graphics. Once Viz Engine has received the request, it renders the graphics, has it sent back to the Graphics Plugin which processes it to fit the format of the NLE system and have it ready.

To keep this workflow going there are a number of factors affecting it:

- What TV-standard to use, for example, PAL is a lot smaller than 1080i and is quicker to render, process and send over the network.
- The CPU and GPU of the Viz Engine define how fast it can deliver graphics. As always, a better graphics card, faster CPU and more memory helps.
- The network card on Viz Engine and the NLE client machine.
- The network capacity defines how much data can be sent over the network.
- The compression of the data decides both how much CPU is used on Viz Engine to compress and on the client to uncompress, as well as how much data is sent over the network.
- The amount of ask-in-advance requests keeping the renderer and the network busy.
- The processing of the NLE system needs to do with the data it receives (affected by CPU and/or GPU, memory, software and so on).

Some things can be improved by adding hardware to the renderer and the clients, and by using as fast a network as possible. Other things, such as [File Compression](#), [Prefetching](#) and [Concurrent Graphics Plugin Users](#), can be configured, either by the plugin or by convention. In addition, the processing done by the NLE system has a great impact on the performance as well.

See Also

- [System Processing](#)

4.2 File Compression

The Graphics Plugin currently supports two different types of compression and the sending of uncompressed data.

The following options are available:

- No compression (raw)
 - **Benefit:** No CPU spent on the renderer to compress and no CPU spent on the client to decompress. Usually very good for SD on good networks. Very good on clients with weak CPU and a good network card.

- **Problems:** For HD the amount of data is large, and it typically takes longer to transfer over the network than compressed alternatives and it uses a lot of network capacity.
- Portable Network Graphics compression (PNG)
 - **Benefit:** Compresses graphics very well. On very slow networks this compression has a very small footprint, which is useful for example over a Digital Subscriber Line (DSL) network, 10Mbit networks or district offices rendering on a renderer in a central office.
 - **Problems:** Can use a lot of CPU to compress and decompress, which is a problem on slower computers. Generally outperformed by the other two options and mostly useful for special cases.
- Run-length Encoding compression (RLE)
 - **Benefit:** Compresses graphics well. Is easy and fast to compress and decompress. Uses a small fraction of the space for a 1080i field compared to no compression so network footprint is quite good. For HD on fairly updated computers, it often performs best. Also does very well on 100Mbit networks. Has special handling that exploits the nature of TV-graphics and NLE graphics which contain bigger open areas, for example lower thirds, over the shoulders, and so on.
 - **Problems:** Uses CPU on the renderer and the client, so less powerful machines might get better performance with no compression. The smaller amounts of data with SD negates some of the compression benefits, but can often be best for SD too. Might not give much benefit if there are big horizontal gradients.

Lab testing has shown that the general advice above is not always sufficient - only on-site testing can determine what works best for a given network.

4.3 Prefetching

As shown in the [File Compression](#) section, the Graphics Plugin predicts what the NLE system will ask for next and requests those frames and fields from the renderer in advance. Although it is possible to disable this feature, safe mode would then be required, as described earlier, and safe mode is only for use if there are problems with pre-fetching.

When fine-tuning the performance, the question is how aggressive the pre-fetching should be. Current releases use more aggressive pre-fetching than earlier, so it's essential to configure how many fields to pre-fetch, that is, how many graphics requests should be on the network from one client at any given moment. It's also possible to ask the system to pre-fetch the whole clip, which would probably provide adequate performance; however, this blocks the Viz Engine for other users who want to edit their effects, scrub the timeline or render. In general, the default setting of pre-fetching 20 fields is a good compromise, although the setup might require a different value (between 5 and 30 are acceptable values).

See Also

- [File Compression](#)

4.4 Concurrent Graphics Plugin Users

The Graphics Plugin has been designed to enable several users to edit, scrub the timeline and render at the same time. Users should have a responsive system and each requires their own state on the Viz Engine. Viz Engine has its own built-in multiplexer functionality.

When fine-tuning performance, it's important to take into account that testing on a single machine does not give the full answer as to how things will be when all machines are working at the same time. In a video editing environment, one can also assume that big video files are also transferred over the network, taking up a certain amount of the network capacity. The *Graphics Plugin performance tester* only tests speed for one user, so the real test is to run a test that uses several users once everything works fine for a single user.

See Also

- [Graphics Plugin Performance Tester](#)
-

4.5 System Processing

Despite an optimally configured Graphics Plugin, users may still experience performance issues. One important factor is the size of the project. Working on a fully interlaced HD (1080i) project uses more resources than PAL or NTSC.

When there is a mismatch between video size and the timeline/project settings, the NLE system might need to re-scale graphics and video, which is a costly process. Rendering on a video that resides on a shared drive on the network also utilizes valuable network capacity.

For example, an interlaced HD project with a well-configured Graphics Plugin has an effect added to a layer above the video, and rendering is started. The following happens:

- The NLE system requests a frame,
- Receives it from the Graphics Plugin,
- Processes the graphics and possibly caches the data on disk.

These steps are then repeated for each frame. Steps 1 to 2 might only take 20 milliseconds (ms); however, the time from 2 to 3 can be as much as 600-1000 ms. Ensuring that the video size, format, project, and timeline settings correspond with each other, and that rendering is done locally on a fast disk is therefore very beneficial. Although the performance analysis reports that everything is fine, the system as a whole might be slow due to slow hardware on the NLE system.

5 Troubleshooting And Logging

The following subsections describe common questions and answers and how to log Graphics Plugin activities. It can be useful to provide log files when [submitting a support request](#).

This section contains information on the following topics:

- [Log Files](#)
 - [Questions and Answers](#)
-

5.1 Log Files

Logging is performed by the Graphics Plugin, and the different applications that the Graphics Plugin integrates with. The following paragraphs describe the different log files, their log levels, and where they are stored.


The Graphics Plugin generates log files that are located in the directory specified by the Graphics Plugin (see below).

The Windows-based log files are placed by default in the following directory:

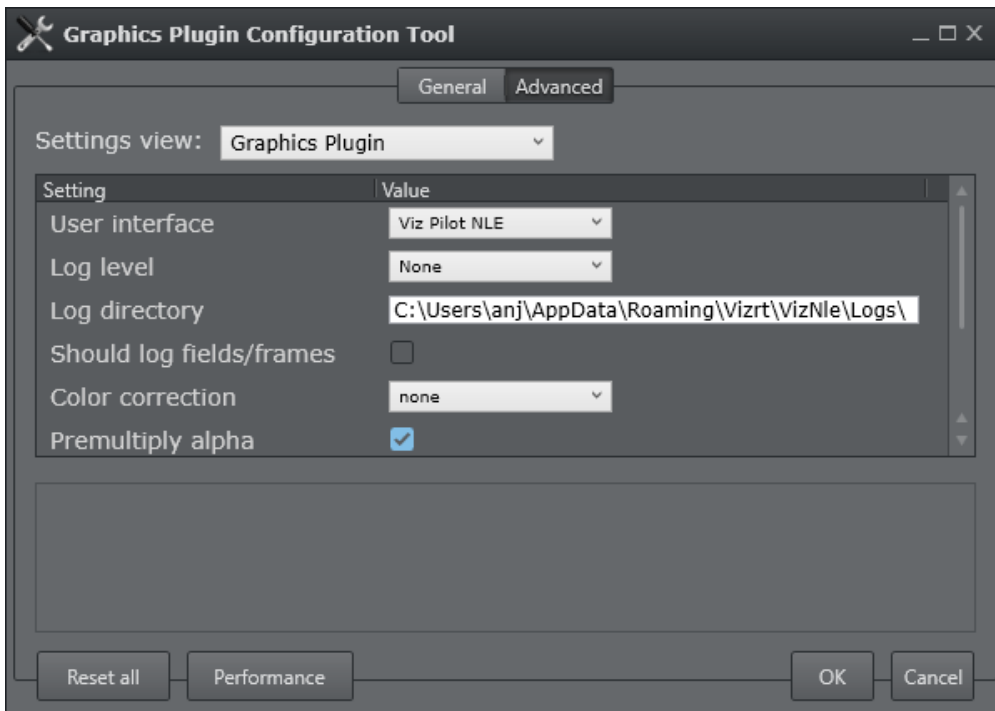
- *%APPDATA%/Roaming/Vizrt/VizNle/Logs*
 - *viznle-log-<date_time>.txt*
 - *nleconfig-<date_time>.txt*
 - *vcp_ax_nle.log*
 - *trio-<date_time>.log*

The Mac-based log files are placed in the following directory:

- *~/Library/Logs/VizNLE/*
 - *viznle_log--<date_time>.txt*
 - *VizNLEConfig--<date_time>.log*
 - *VizNLE_Editor--<date_time>.log*

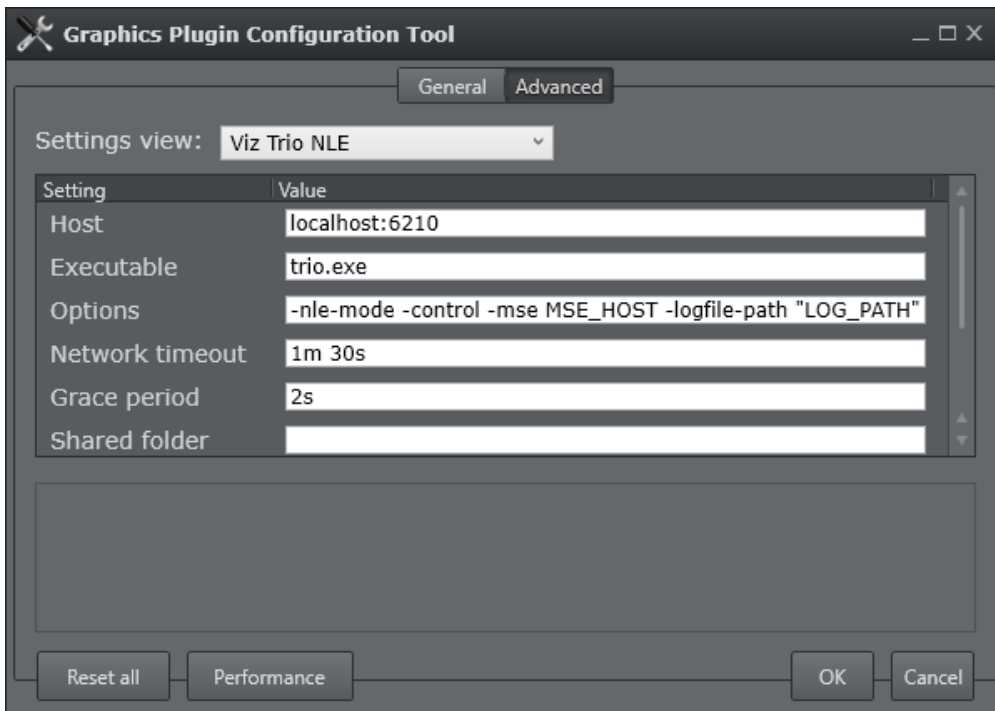
 **Note:** The **Log directory** does not apply for Graphics Plugin Editor.

The log directory can be modified from the Configuration Tool. Go to the Graphics Plugin Settings view in the Advanced tab and specify a new path in the **Log directory** setting.



The path for the Viz Trio and Viz Pilot log files can also be modified from the Configuration Tool, under the Viz Trio NLE and Viz Pilot NLE Settings view respectively. By default, these log files are defined to be placed in the same folder as the Graphics Plugin specific log files, by using the `-logfile-path LOG_PATH` command in the options text box (LOG_PATH refers to the Plugin log directory setting).

To place the Viz Trio and Viz Pilot log files in a separate folder from the Graphics Plugin log files, replace LOG_PATH in the Options setting with a folder name.



5.2 Questions And Answers

This section contains a mix of troubleshooting and frequently asked questions to help you with issues that may arise when using Graphics Plugin.

Note: Differences between the Mac and Windows-based versions are noted where relevant.

This section covers the following topics:

- [General](#)
- [Using the NLE System](#)
- [Configuring the NLE System](#)
- [Tuning the NLE System](#)
- [Rendering with Viz Engine](#)


5.2.1 General

- **How do I contact Vizrt with my Graphics Plugin questions?**
 - Before a question is forwarded, please read the questions and answers in the sections below. For problem reporting, see the Reporting Problems section.
- **Why are my start, stop and continue buttons in Viz Trio disabled?**

- When using an external Viz Engine, the start, stop and continue buttons in Viz Trio are disabled. This also applies the **Restart Viz Engine** option available from the context menu in the Render window.

5.2.2 Using the NLE System


- **Why does the Vizrt graphics stop showing in the preview window?**
 - This can occur when still image playback is turned on and a non-existing graphic has been added to the timeline. If this happens, remove the faulty graphic, then temporarily turn off still image playback. Turn still image playback back on after verifying that the graphics preview is correct. The bug does not affect rendered output. Still image playback is turned on/off in the **Configuration Tool**, see the *Graphics Plugin User Guide*.
- **Why am I seeing double-sized graphics?**
 - There is a difference between broadcast and consumer resolution. Several of the plugins are confused by for example consumer HD, producing graphics of the wrong sizes. If the camera says 1080i and has a resolution of 1440x1080 or 1280x1080, it's consumer HD as opposed to broadcast HD of 1920x1080.
 - Check that the software versions are correct. For details, see [Requirements and Recommendations](#).
- **I have edited and saved a scene but I cannot see the graphics on the video. Why is this?**
 - When editing a scene in Viz Trio, Viz Pilot or the Graphics Plugin Editor, select to see the scene with *key* turned on (available above the renderer window). If the scene goes black, there is a design flaw in the scene. Redesign the scene by adding key to it. For details, see the [Scene Design](#) section.
- **Why does the preview in Viz Pilot only display PreviewPanel?**
 - This might be a network bandwidth problem. In most cases this happens when the Graphics Plugin's compression is set to No Compression and there is a bandwidth mismatch between Viz Engine and the NLE machine. Usually, this happens when for example the NLE machine is connected to a 1Gbps network and the Viz Engine is connected to a 100Mbps network.
 - Try to set compression to *PNG Compression* or configure both machines to use the same network with the same bandwidth. Remember to check that the network cards are compatible with the network it is connected to.
 - Viz Engine should ideally be connected to a fast network. If the NLE machine is connected to a slower network it should work fine.
 - In order to preview templates in Graphics Plugin Editor, Viz Trio, Viz Pilot and Viz Pilot Edge, the Viz Engine port on the graphics plugin Configuration Tool must be 50007 (preview port 50008) or 50107 (preview port 50108). See [Network Requirements](#) for more information.

 **Note:** Recommended bandwidth for No Compression is 1Gbps.

- **How do I set Viz Pilot's preview aspect ratio?**
 - The preview aspect ratio is set in Config's **Output Format** section (Viz Engine's configuration tool).
- **Why is Avid locked while using the Graphics Plugin?**
 - Avid and most NLE systems are designed to block whenever a plugin is executing, including when the user has pressed **Edit**, the editor is open and the plugin is waiting for a response.
- **Why are the Graphics Plugin editors and configuration programs not closed when the NLE system is closed?**
 - The NLE system is unable to handle a shutdown of the plugin itself (see the previous question).
 - If the NLE system is accidentally terminated or closed this might cause the plugin process to still be running, and a manual and forced shutdown of the plugin is necessary. Force shutdown by restarting Avid or using the Task Manager.

5.2.3 Configuring the NLE System

- **Which ports are used in a typical installation?**
 - See [Port numbers](#).
- **On a Windows system, how do I configure the Viz Engine host and port?**
 - For details about installing Viz Engine, see [Graphics Rendering](#).
- **On a Windows system, how do I configure the output from the Viz Engine?**
 - Some settings related to the output from the Viz Engine are configurable. For example, the image compression protocol (*No, PNG or RLE*).
 - If some of the Viz Engine settings are incorrect, the preview is disabled and a message is displayed stating the cause. Incorrect settings may include if the host is not found, cannot connect to the port, and so on. In addition, the user is notified if the Viz Engine connection is lost during runtime.

 **Note:** Some GUI settings are only valid for Windows-based NLE integrations using Viz Trio and Viz Pilot.

- **On a Windows system, how do I configure the user interface that the Graphics Plugin should use?**
 - Open the Graphics Plugin configuration tool and set the **Plugin user interface** setting to **Viz Pilot** or **Viz Trio**.
 - The setting ensures that the NLE application sends the extra data that the user interface needs. This includes, but is not limited to, the scene length in milliseconds.
 - An error message is always displayed if the NLE system is not configured properly. The log files also show warnings for non-critical problems.

5.2.4 Tuning the NLE System

- **How can I find out how much time is used between each call from the NLE system to the plugin?**

- Empty the log directory. Start a project, make an effect, render it and exit the NLE system. Go to the log directory (see the [Log Files](#) section) and run the following command:

```
grep --binary-files=text PERFORMANCE viznle-log*.txt
```

- Disregard the first few numbers and you might find for example:

```
14:42:03 [T:2689318816] [/Users/<username>/fcp-1.0/plugins/fcp/VizNlePlugin.mm,876] QUEUE PERFORMANCE: Time between renders 397 msecs.
14:42:03 [T:2689318816] [/Users/<username>/fcp-1.0/plugins/fcp/VizNlePlugin.mm,876] QUEUE PERFORMANCE: Time between renders 170 msecs.
14:42:03 [T:2689318816] [/Users/<username>/fcp-1.0/plugins/fcp/VizNlePlugin.mm,876] QUEUE PERFORMANCE: Time between renders 394 msecs.
14:42:04 [T:2689318816] [/Users/<username>/fcp-1.0/plugins/fcp/VizNlePlugin.mm,876] QUEUE PERFORMANCE: Time between renders 161 msecs.
14:42:04 [T:2689318816] [/Users/<username>/fcp-1.0/plugins/fcp/VizNlePlugin.mm,876] QUEUE PERFORMANCE: Time between renders 157 msecs.
14:42:04 [T:2689318816] [/Users/<username>/fcp-1.0/plugins/fcp/VizNlePlugin.mm,876] QUEUE PERFORMANCE: Time between renders 411 msecs.
14:42:04 [T:2689318816] [/Users/<username>/fcp-1.0/plugins/fcp/VizNlePlugin.mm,876] QUEUE PERFORMANCE: Time between renders 161 msecs.
```

5.2.5 Rendering with Viz Engine

- **Which Viz Engine version should I run for HD?**
 - A standard-PC version for all HD output is recommended due to a number of issues in the INCA version.
- **Can I include Datapool in the Graphics Plugin workflow?**
 - When including the Datapool mechanism in the Graphics Plugin workflow, you are always working on a scene-copy. The built-in multiplexer functionality of Viz Engine rewrites references to RENDERER to the scene-copy you are using. This means that if your scene has DataActions that refer to RENDERER, they are not rewritten to the right scene, and they will fail fairly often. Referring to \$SCENE will likely also not work.

See Also

- [Graphics Rendering](#)
- [Tuning the NLE System](#)