



# Viz Trio User's Guide

Version 2.10

November 27, 2013







**Copyright © 2013 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](http://www.vizrt.com).

### **Last Updated**

27 November 2013



# Contents

- 1 Introduction ..... 1
  - 1.1 Viz Trio User Guide ..... 1
    - 1.1.1 Document Structure ..... 1
    - 1.1.2 Related Documents ..... 2
    - 1.1.3 Conventions ..... 2
  - 1.2 Words and Expressions ..... 2
  - 1.3 Contact Vizrt ..... 3
  
- 2 Viz Trio ..... 4
  - 2.1 New Features in Viz Trio 2.10 ..... 4
  
- 3 Requirements ..... 5
  - 3.1 General Requirements ..... 5
  - 3.2 Hardware Requirements ..... 5
  - 3.3 Software Requirements ..... 6
    - 3.3.1 Viz Anchor ..... 6
    - 3.3.2 Viz Artist ..... 6
    - 3.3.3 Viz Content Pilot ..... 7
    - 3.3.4 Viz Engine ..... 12
    - 3.3.5 Viz Gateway ..... 13
    - 3.3.6 Viz Graphics Hub ..... 14
    - 3.3.7 Media Sequencer ..... 14
    - 3.3.8 Viz Trio ..... 15
    - 3.3.9 Viz Video Hub ..... 16
    - 3.3.10 Vizrt Maps ..... 16
  - 3.4 Shared Data ..... 17
  - 3.5 Ports and Connections ..... 17
    - 3.5.1 Port Numbers ..... 18
    - 3.5.2 Multiplexing Ports ..... 21
  
- 4 Storage and Backup Recommendations ..... 24
  - 4.1 Graphics Data ..... 24
  - 4.2 Viz Content Pilot ..... 25
  - 4.3 Media Sequencer ..... 25
  - 4.4 Viz Trio ..... 26
  
- 5 Installation ..... 27
  - 5.1 System Overview ..... 27
  - 5.2 Prerequisites ..... 28
  - 5.3 Viz Engine ..... 28
  - 5.4 Media Sequencer ..... 29
  - 5.5 Viz Trio ..... 30
    - 5.5.1 Keyboard ..... 30

5.5.2	Viz Content Pilot .....	31
<b>6</b>	<b>Getting Started.....</b>	<b>33</b>
6.1	Basic Configuration and Startup .....	33
6.1.1	Connecting to Media Sequencer .....	33
6.1.2	Connecting to Viz Engine .....	34
6.1.3	Starting Viz Trio .....	34
6.2	Configuring the Graphics Database .....	34
6.3	Configuring the Media Sequencer .....	37
<b>7</b>	<b>User Interface.....</b>	<b>39</b>
7.1	Viz Trio Keyboard .....	39
7.2	Playout and Design Mode .....	43
7.3	Viz Artist Mode .....	43
7.4	On Air Mode .....	44
7.5	Slave Mode .....	44
7.6	Show Modes .....	44
7.6.1	Context-enabled Shows .....	44
7.6.2	User-defined Contexts .....	46
7.7	Show Control .....	46
7.7.1	Show Directories .....	47
7.7.2	Add Page List View .....	55
7.7.3	Create Playlist .....	56
7.7.4	Show Settings .....	57
7.7.5	Cleanup Channels .....	62
7.7.6	Initialize Channels .....	62
7.7.7	Show Concept .....	63
7.7.8	Callup Page .....	63
7.8	Template List .....	64
7.9	Page List .....	67
7.10	Playlist Modes .....	79
7.10.1	New Playlist .....	81
7.10.2	Old Playlist .....	84
7.11	Tab Fields Window .....	88
7.12	Status Bar .....	88
7.13	Page Editor .....	91
7.13.1	Controls .....	92
7.13.2	Text .....	95
7.13.3	Database Linking .....	99
7.13.4	Images .....	102
7.13.5	Transformation Properties .....	102
7.13.6	Tables .....	104
7.13.7	Clock .....	106
7.13.8	Maps .....	107
7.14	Create New Scroll .....	108

7.14.1	Create New Scroll Editor .....	109
7.14.2	Scroll Elements Editor .....	110
7.14.3	Scroll Configuration .....	111
7.14.4	Scroll Live Controls .....	112
7.14.5	Scroll Control .....	112
7.14.6	Element Spacing .....	112
7.14.7	Easepoint Editor .....	113
7.14.8	Working with Scrolls .....	114
7.15	Snapshot .....	116
7.16	Render Videoclip .....	117
7.17	Search Media .....	118
7.18	Import Scenes .....	120
7.19	Graphics Preview .....	122
7.19.1	Graphics Control Buttons .....	122
7.19.2	Connection Status .....	124
7.20	Video Preview .....	125
7.20.1	Video Control Buttons .....	125
7.21	TimeCode Monitor .....	126
<b>8</b>	<b>Configuration Interface .....</b>	<b>128</b>
8.1	User Interface .....	128
8.1.1	General .....	129
8.1.2	Keyboard Shortcuts and Macros .....	130
8.1.3	Text Editor .....	131
8.1.4	Page List .....	132
8.2	Output .....	135
8.2.1	Program and Preview Channels .....	136
8.2.2	Profile Setups .....	136
8.2.3	Forked Execution .....	137
8.2.4	Working With the Profile Configuration .....	140
8.2.5	Upgrading Old Profiles .....	145
8.3	Directories .....	146
8.4	Database (Viz 3.x) .....	146
8.5	Viz Content Pilot Database .....	147
8.5.1	Database Settings .....	147
8.5.2	Picture Database Settings .....	148
8.6	Connectivity .....	149
8.6.1	MOS .....	149
8.6.2	Intelligent Interface .....	152
8.6.3	General Purpose Input (GPI) .....	154
8.6.4	Video Disk Communication Protocol (VDCP) .....	158
8.6.5	MCU/AVS .....	159
8.6.6	Socket Object Settings .....	160
8.6.7	Viz Video Hub .....	162
8.7	Miscellaneous .....	163

8.7.1	General .....	163
8.7.2	Local Viz Renderer .....	165
8.7.3	Viz Proxy .....	166
8.7.4	User Restrictions .....	166
8.8	Import/Export Settings .....	168
8.9	About .....	169
9	Scripting .....	170
9.1	Viz Trio Scripting .....	170
9.1.1	Script Directory .....	171
9.1.2	Script Editor .....	171
9.1.3	Script Backup .....	175
9.2	Viz Template Wizard Scripting .....	175
10	Macro Language .....	177
10.1	Working with Macro Commands .....	178
10.1.1	Apropos and Help Commands .....	179
10.1.2	Show, Context and Tab-field Commands .....	179
10.2	Working with Macro Commands over a Socket Connection .....	182
10.2.1	Sending Macro Commands .....	182
10.2.2	Unicode Support .....	183
10.2.3	Escape Sequences .....	183
10.2.4	Error Messages .....	184
10.3	Working with Shortcut Keys .....	184
11	Macro Commands and Events .....	187
11.1	Commands .....	187
11.2	Events .....	219
12	Designer .....	221
12.1	About Viz Trio Designer .....	221
12.1.1	Scene .....	222
12.1.2	Resources .....	223
12.1.3	Scene Tree .....	225
12.1.4	Tab Fields .....	226
12.1.5	Page Editor .....	226
12.1.6	Properties .....	227
12.2	Creating Scene Elements in Viz Artist .....	232
12.2.1	Viz Artist User Interface .....	233
12.2.2	Creating Graphics .....	233
12.2.3	Adding Control Plug-ins .....	240
12.2.4	Creating Backgrounds .....	241
12.2.5	Creating Backplates .....	243
12.2.6	Creating Text Objects .....	246
12.2.7	Creating Image Objects .....	248
12.2.8	Creating Animations .....	250



13	Creating Standalone Scenes.....	254
14	Creating Transition Effects .....	266
15	Command Line Parameters.....	269
16	Logging.....	271
16.1	Viz Trio Log Files .....	271
16.2	Viz Engine Log Files .....	271
16.3	Media Sequencer Log Files .....	272
16.4	Log Level .....	273
17	Appendix .....	275
17.1	Cherry Keyboard .....	275
17.1.1	Editing Keys (green) .....	276
17.1.2	Navigation Keys (white) .....	276
17.1.3	Program Channel Keys (red) .....	277
17.1.4	Preview Channel Keys (blue) .....	277
17.1.5	Program and Preview Channel Keys (red and blue) .....	278

---

# 1 Introduction

This is the user's guide for Viz Trio version 2.10.

This section contains information on the following topics:

- [Viz Trio User Guide](#)
- [Words and Expressions](#)
- [Contact Vizrt](#)

---

## 1.1 Viz Trio User Guide

This user guide covers the information needed in order to configure and operate Viz Trio 2.10.

This section contains information on the following topics:

- [Document Structure](#)
- [Related Documents](#)
- [Conventions](#)

### 1.1.1 Document Structure

This user guide introduces and explains the different aspects of Viz Trio. Starting with an introduction to the user's guide and Viz Trio's latest and improved features in chapters 1 and 2.

Chapters 3 and 4 describe the basic requirements for Viz Trio and its integrated systems to run, and how to handle storage and backup.

Chapters 5, 6 and 7 cover the installation, configuration and how to start Viz Trio and its most important integration points; the Media Sequencer and Viz Engine.

Chapters 8 and 9 cover all aspects of the user interface; for example how to use the Viz Trio keyboard, create shows, add templates and create pages of templates, how to configure the client's output channels and so on.

Chapters 10, 11 and 12 cover Viz Trio's scripting possibilities, its macro language, macro commands and scripting events.

Chapters 13, 14 and 15 cover some of the design aspects of Viz Trio using Viz Trio's built-in Designer tool, how to create a basic lower third and how to use transition effects. For more advanced scene design refer to the Viz Artist User's Guide.

Chapters 16 and 17 cover the command line parameters and logging capabilities.

Chapter 18 covers information on the old Cherry keyboard.

## 1.1.2 Related Documents

For complementary information, see the following documents:

1. *Viz Artist User's Guide*: How to create standalone and transition logic scenes.
2. *Viz Content Pilot User's Guide*: How to create a playlist in VCP that can be monitored in Viz Trio, and use the Newsroom Component to create newsroom data elements for a newsroom system playlist.
3. *Viz Engine Administrator's Guide*: How to configure your Viz Engines.

## 1.1.3 Conventions

The following typographic conventions are used in this document:

- **Bold is used to indicate emphasized text.**
- *Italic is used to refer to related documents and captions in the user interface.*
- [The color Blue is used to indicate cross-references.](#)

---

## 1.2 Words and Expressions

The following is a number of common words and expressions used throughout the documentation:

- **Control Plug-ins** – A graphics scene can contain all sorts of objects that can be controlled from a template such as text, back plates, images, colors and more. The graphics designer uses control plug-ins to expose objects as tab fields.
- **Control Object** – Every scene with control plug-ins must have one instance of the control object plug-in at the root level of a scene tree. Control Object reads the information from all other control plug-ins. When a scene is imported to Viz Template Wizard, it reads the information about other lower level control plug-ins through Control Object.
- **Newsroom Component** – In a Newsroom Computer System (NCS), the Newsroom Component (NrC) is used to add data elements to a story. The user is typically a journalist working on a story. The NrC is an embedded application in the NCS that connects to a database of templates. The templates can be filled with text, images, video and maps.
- **Page** – A page is based on a template and saved as a data instance of the template. A page contains a set of data and references to where data (e.g. images and video clips) can be found. When a page is loaded, all static elements will be loaded from the template and the variable elements (items/tab fields) will be set to what they were when the page was saved. Therefore, a page is not saved as a complete scene. Just the values of the tab fields are saved together with a reference to the template-scene.

-----  
**Note:** Only pages can be taken on air.  
-----

- **Scene** – A scene is built in Viz Artist. It can be a single scene, or one part (layer) of a combination of scenes (transition logic).
- **Template** – A template is based on a Viz Artist scene, and is created by Viz Trio on-the-fly while importing it to Viz Trio. The template is used to create pages that are added to a show for playout. A template may be based on one or several (transition logic) scenes (known as a combination template).
- **Transition Logic Scene** – A set of scenes built in Viz Artist. It contains one scene that controls the state of or toggles a set of scenes (layers). The layered scenes are used by the controlling scene to toggle in and out the layered

scenes, using pre-configured or customized transitions effects, without the need to take scenes already on-air, off-air. For example; A lower third may be on-air at the same time as a bug, and the lower third may be taken off-air without taking the bug off-air or reanimating it.

- **Viz Artist** – The design tool where the graphics scenes and all animations are created.
- **Viz Engine** – The output engine used for playout of graphics and video.

---

## 1.3 Contact Vizrt

We encourage your suggestions and feedback regarding the products and this documentation. Our support e-mail address is [support@vizrt.com](mailto:support@vizrt.com). You are also welcome to use the web-based support at [www.vizrt.com](http://www.vizrt.com), or to contact your local Vizrt support person by telephone.

---

## 2 Viz Trio

Viz Trio is a graphics control system from Vizrt, which includes, but is not limited to, all the features of a typical character generator (CG) system. It is based on Viz Engine as the graphics system and the Media Sequencer as the control engine. It allows users to trigger graphical elements stored as pages in a directory structure, with each page utilizing a unique callup code. It also supports content entry in a WYSIWYG manner. In addition to the basic CG features, it has a set of advanced features that gives the operator powerful graphical and operational possibilities. Such possibilities are: Connect to multiple newsroom systems, do seamless context switches on graphics, use specialized editors that can change almost any property of a graphics, produce on-the-fly graphics with a built-in design tool and more.

As a client to the Media Sequencer, the Viz Trio client can run on any computer that has a network connection to the Media Sequencer. For the local preview to function properly the computer must have an above average graphics card with OpenGL support.

Viz Trio contains some JEDI Visual Component Library (JVCL) source code.

---

### 2.1 New Features in Viz Trio 2.10

- New playlist view with significantly improved performance for playlists within shows and external playlists (e.g. MOS). Switch to old play list view is supported for compatibility reasons.
- New profile configuration with support for forked execution on several Viz Engines.
- New media search view that supports search for video clips, audio and stills. Standalone stills can now be added to the page list or playlist.
- New script events are added. OnBeforeTake with possibility to omit the take operation, and OnMove, OnCopy and OnDelete of pages.

For more information, see the Viz Trio 2.10 release notes.

---

## 3 Requirements

This section describes the different requirements, such as hardware, software, access rights and general policies for a number of software components that can be integrated with Viz.

.....  
**IMPORTANT!** Always check release notes for information on supported versions.  
.....

For more in depth system information, see the respective user and administrator guides.

This section contains information on the following topics:

- [General Requirements](#)
- [Hardware Requirements](#)
- [Software Requirements](#)
- [Shared Data](#)
- [Ports and Connections](#)

---

### 3.1 General Requirements

There are some general requirements for any Vizrt system to run. These requirements apply when setting up a complete system with integration to other Vizrt and third party software products:

- All machines should be part of the same domain.
- Users of the Vizrt machines should ideally be separated in at least two groups, administrators and designers/operators.
- Most machines running desktop applications must be logged in with sufficient privileges in order to run Vizrt programs, while services by default do not require users to be logged in.
- Vizrt servers must have static IP addresses.

.....  
**CAUTION!** Third party systems that provide Vizrt systems with files must only use Microsoft Windows operating system compatible characters as the file name.  
.....

- Vizrt has license restrictions on all Viz Engine and Viz Artist instances. To have an output of Vizrt generated graphics (preview and program channels), either an USB or parallel port dongle on the renderer machine is required.

---

### 3.2 Hardware Requirements

Hardware requirements vary depending on the system purchased; however, every system delivered by Vizrt has an accompanying hardware specification sheet that, for a new system, matches the [Software Requirements](#).

For older hardware that are used with newer versions of Vizrt software (e.g. Viz Engine) it is always recommended to check against the current hardware

specifications for the new software version to make sure that the latest software can run on the old hardware specification.

Additional hardware must always be checked to see if it is compatible with existing hardware. For example, the GPI cards supported by Vizrt must fit in the Media Sequencer servers.

For more information on hardware specifications take contact with your local Vizrt representative or send an e-mail to [support@vizrt.com](mailto:support@vizrt.com).

---

## 3.3 Software Requirements

The following sections describe software requirements for the product described in this manual and a range of components that may be integrated with it. For more information see the following topics:

- [Viz Anchor](#)
- [Viz Artist](#)
- [Viz Content Pilot](#)
- [Viz Engine](#)
- [Viz Gateway](#)
- [Viz Graphics Hub](#)
- [Media Sequencer](#)
- [Viz Trio](#)
- [Viz Video Hub](#)
- [Vizrt Maps](#)

### 3.3.1 Viz Anchor

Viz Anchor is a Vizrt application targeted at the presenter in the studio. The application runs natively on Apple's iPad and allows users to control playlists with video and graphics directly from the handheld device.

Software	Viz Anchor 1.0 or later, Media Sequencer 1.21 or later
Operating system	Apple iOS 3.2 or later
Network access	Uses the Bonjour protocol to automatically discover the Media Sequencer and Preview Server if the wireless router/switch allows it.
Hardware	Apple iPad 1 or 2

### 3.3.2 Viz Artist

Viz Artist is an advanced real-time motion graphics authoring system for the creation of stunning real-time graphics. Built with an elegant and easy to use drag-and-drop user interface and sophisticated 3D animation and modeling tools, Viz Artist enables the digital artist to produce complex and engaging visual

content for broadcast, virtual sets, and visualization in less time and with greater creative freedom.

**Table 1:** Viz Artist specifications

Software	Viz Engine 3.2.8 or later, or Viz Engine 2.8 PL5 HotFix3  Viz World Client 10.0 or later (see <a href="#">Vizrt Maps</a> ) Viz DataPool 2.10 or later Extra Viz 2 plug-ins (for Viz 2.x)  <i>Optional:</i> Viz Content Pilot with Viz Template Wizard, Media Sequencer, Viz Multiplexer and Thumbnail Generator.
Executable(s)	viz.exe VizGui.exe vizSend.exe
<a href="#">Ports and Connections</a>	TCP: 6100 (preview and playout) 14300 (Viz Multiplexer) 50007–50009 (multiplexing).
Local drive access	Read and write access to C:\Program Files\vizrt\viz\
Network access	Mapped drive to VOS still store folder and Viz Engine data root (see <a href="#">Shared Data</a> )
Operating system	Windows XP SP2/SP3 (32-bit) and Windows 7 (64-bit)

The Viz Artist design machine should preferably have the same specifications as the [Viz Engine](#) playout renderers, especially if the designers need to test performance issues on demanding scenes.

If designers are creating templates for [Viz Content Pilot](#) (VCP), it is recommended that VCP is installed on a separate machine for more accurate playout emulation on [Viz Engine](#).

### 3.3.3 Viz Content Pilot

Viz Content Pilot (VCP) is built on a client-server software model where the VCP client connects to the [Viz Content Pilot Database](#) (Oracle database server) for templates and content, and the [Media Sequencer](#) for playout.

- VCP requires an Oracle database as the back-end server.
- VCP requires a Media Sequencer for communication with Viz Engine and other systems.
  - VCP 5.5 requires Media Sequencer version 1.21 or later.

On the server side, the database serves all clients storing and retrieving content data for the control, delivery and triggering of graphics rendered on the graphics renderer Viz Engine.

This section contains information on the following topics:

- [Viz Content Pilot Client](#)



- [Viz Content Pilot Database](#)
- [Viz Content Pilot Database Administrator](#)
- [Viz Content Pilot Newsroom Component](#)
- [Viz Content Pilot Thumbnail Generator](#)
- [Viz Template Wizard](#)
- [Pilot Data Server](#)

## Viz Content Pilot Client

The [VCP specifications](#) shown below are machine specifications for a typical newsroom system setup, where VCP is the control application receiving the playlist.

**Table 2:** VCP specifications

Software	Oracle 10g Runtime Client 32-bit Viz Content Pilot 5.5 Media Sequencer 1.21 or later <i>Optional:</i> Viz World Client, VCP TimeCode Monitor, Viz PreCut. <i>Optional:</i> If local preview is chosen, Viz Engine (Extra Viz 2.x Plugins and Viz DataPool) must be installed. <i>Optional:</i> Windows Media Player 11 for video clip preview in Viz Object Store.
Executable(s)	VizContentPilot.exe vizPreviewEngine.exe (local preview)
Local drive access	Access to Oracle client files and folders Read and write access to C:\Program Files\vizrt\
Network access	Mapped drive to Viz Object Store still store folder <i>Optional:</i> If local preview using Viz Engine 2 is chosen, a mapped drive to Viz Engine's data root is needed Also see <a href="#">Shared Data</a>
Operating system	Windows 7 Professional 64-bit (recommended) Windows 7 Enterprise 64-bit Windows 7 Ultimate 64-bit Windows XP Professional 32-bit

Vizrt recommends the use of remote preview that has no need for a local Viz Engine or graphics cards on the client machine.

### Viz Content Pilot Database

The VCP database is an Oracle database server. Usually two database servers are installed where one is used for manual failover and to hold the Viz Engine 2.x data root. Viz Engine 3.x graphics data is stored on the [Viz Graphics Hub](#).

**Table 3:** VCP database specifications

Software	Oracle Administrator Client 10g/11g (32-bit standard editions)
Executable(s)	Please refer to official Oracle documentation.
<a href="#">Ports and Connections</a>	1521 (queries)
Services	Oracle database service, TNS listener service
Local drive access	A shared folder on the second server (failover) has to be accessible for all machines showing the rendered graphics.
Operating System	Please refer to official Oracle documentation
Oracle Documentation	Oracle Database Documentation Library: 11g: <a href="http://www.oracle.com/pls/db112/homepage">http://www.oracle.com/pls/db112/homepage</a> 10g: <a href="http://www.oracle.com/pls/db102/homepage">http://www.oracle.com/pls/db102/homepage</a>

### Viz Content Pilot Database Administrator

The Viz Content Pilot Database Administrator tool (VCP DBA) is a small application used for installing, upgrading, exporting, importing, and setting various parameters for the [Viz Content Pilot Database](#). It is purely a DBA tool, and should therefore only be used by database administrators.

**Table 4:** VCP DBA specifications

Software	Oracle Administrator Client 10g/11g 32-bit VCP DBA
Executable(s)	VCPDBA.exe
Services	Oracle database service TNS listener service
Operating system	Windows Server 2003 32-bit Windows XP 32-bit Windows 7 32-bit or 64-bit

### Viz Content Pilot Newsroom Component

The newsroom client machine specification describes a basic setup for journalists and editors. For a more detailed view on available software options, see the Viz

Content Pilot and other administrator guides for descriptions on different types of setup.

**Table 5:** Newsroom Component specifications

Software	<p>Newsroom system client          Oracle 10g Instant Client 32-bit or later          VCP's Newsroom Component 5.5          Viz Object Store 5.5          Viz World Client 10.0 or later (see <a href="#">Vizrt Maps</a>)  <i>Optional:</i> Vizky 1.6 and later (older versions will not work)  <i>Optional:</i> Oracle 10g Runtime Client or later.  <i>Optional:</i> If local preview is chosen, Viz Engine (with Extra Viz 2.x Plugins and Viz DataPool) must be installed.  <i>Optional:</i> Viz EasyCut or Viz PreCut for video clip editing.</p>
Executable(s)	<p>VCPAxFiller.ocx          VCPAxFiller.exe (NLE)          viz.exe          VizObjectStore.exe</p>
Local drive access	<p>Read access to Oracle client files and folders          Read and write access to C:\Program Files\vizrt\</p>
Network access	<p>Mapped drive to VOS still store folder.  <i>Optional:</i> If local preview using Viz Engine 2 is chosen, a mapped drive to Viz Engine's data root is needed.          Also see <a href="#">Shared Data</a>.</p>
Other	<p>Registry settings for preview          Registry settings for Media Object Server (MOS) ID</p>
Operating system	<p>Windows 7 Professional 64-bit (recommended)          Windows 7 Enterprise 64-bit          Windows 7 Ultimate 64-bit          Windows XP Professional 32-bit</p>

Vizrt recommends the use of remote preview that has no need for a local Viz Engine on the client machine; hence, local preview is not recommended.

All users of machines installed with Viz Engine must have read and write access to the following folder: C:\Program files\vizrt\.

The Newsroom Component derives its Viz Video Hub parameters from the VCP database.

### Viz Content Pilot Thumbnail Generator

VCP's Thumbnail Generator is an optional addition to a VCP setup that generates data element snapshots used as thumbnails in order to visualize graphics and video elements in the VCP client's playlist.

It is recommended to install Thumbnail Generator on the [Viz Graphics Hub](#) or [Viz Content Pilot Database](#) machine.

It is also recommended to configure Viz Thumbnail Generator to fetch scene snapshots from one of the newsroom Viz Engine preview machines. If installed on a local Viz Engine, see also [Viz Engine](#).

---

**CAUTION!** Do not use an on-air Viz Engine to generate thumbnails.

---

**Table 6:** Viz Thumbnail Generator specifications

Software	Oracle 10g Instant Client 32-bit Viz Content Pilot's Thumbnail Generator 5.5 <i>Optional:</i> Oracle 10g Runtime Client 32-bit
Executable(s)	ThumbnailGenerator.exe
Operating system	Windows 7 Professional 64-bit (recommended) Windows 7 Enterprise 64-bit Windows 7 Ultimate 64-bit Windows XP Professional 32-bit Windows 2003 32-bit.

### Viz Template Wizard

Viz Template Wizard is a template design and scripting tool used to create templates for graphics and video. It is possible to install Viz Template Wizard on the same machine as [Viz Artist](#); however, it is recommended to install a separate design client for template designers.

**Table 7:** Viz Template Wizard specifications

Software	Oracle 10g Instant Client 32-bit Media Sequencer 1.20 Viz Template Wizard 5.5 <i>Optional:</i> Oracle 10g Runtime Client 32-bit <i>Optional:</i> Windows Media Player 11 for video clip preview in Viz Object Store.
Executable(s)	scheduler.exe (when running in Console mode) VizTemplateWizard.exe
Operating system	Windows 7 Professional 64-bit (recommended) Windows 7 Enterprise 64-bit Windows 7 Ultimate 64-bit Windows XP Professional 32-bit.

Viz Template Wizard connects to the [Media Sequencer](#) for testing and previewing of template graphics. The Media Sequencer connection defaults to *localhost*. To use a different host set the command line option *-mse <host>* in Viz Template Wizard's target path.

## Pilot Data Server

The Pilot Data Server is installed as an application layer on top of the Viz Content Pilot database. The Pilot Data Server may be asked to handle requests from scripts to provide information on data elements, or to provide frame servers the information needed in order to resolve which scene and data that is to be rendered by the frame server.

**Table 8:** Pilot Data Server specifications

Software	Oracle 10g Instant Client 32-bit Microsoft .NET Framework 4 Pilot Data Server 1.1 or later
Executable(s)	Vizrt Pilot Data Server (service)
Ports and Connections	8177
Operating system	Windows 7 Professional 64-bit (recommended) Windows 7 Enterprise 64-bit Windows 7 Ultimate 64-bit Windows XP Professional 32-bit Windows 2008 server 32-bit

### 3.3.4 Viz Engine

Viz Engine is an extremely powerful rendering engine and at the core of Vizrt's real-time graphics solutions. 2D and 3D animated scenes designed in [Viz Artist](#) are rendered in real-time as high-end animations, and the output can be SD or HD video. Viz Engine systems work with all other Vizrt products to provide users with the total solution for producing on-air graphics content.

To run Viz Engine as a program or preview (optional) machine, the following software and configuration is needed:

**Table 9:** Viz Engine specifications

Software	Extra Viz 2 or 3 Plug-ins (only for Viz 2.x) Viz Engine 2.8 PL5 HotFix3 or Viz Engine 3.2.8 or later Viz DataPool 2.10 or later Viz World Client 10.0 or later (see <a href="#">Vizrt Maps</a> )
Hardware	Licensed dongle
Executable(s)	viz.exe
Ports and Connections	6100 (preview and playout), 14300 (Viz Multiplexer), 50007–50010 (multiplexing).
Local drive access	<i>Local preview:</i> Read and write access to folder C:\Program Files\vizrt\
Network access	Mapped drive to VOS still store folder <i>Local preview:</i> Mapped drive to Viz Engine data root Also see, <a href="#">Shared Data</a>

**Table 9:** Viz Engine specifications

Operating system	Windows XP SP2/SP3 (32-bit) and Windows 7 (64-bit)
------------------	--

Machines setup for local preview need an OpenGL compatible graphics card and at least 512MB of memory (RAM) in addition to a reasonably new processor. Some graphics features on a preview machine will not be shown exactly as on a Viz Engine renderer. This is limitations in the OpenGL features on the graphics cards, and not related to Vizrt's software.

This section also contains information on the following topics:

- [Preview Server](#)

### Preview Server

The Preview Server option is used in situations where Viz Engine will be used to provide frames for snapshot/thumbnail generation. Note that the Preview Server must be installed on a separate Viz Engine machine with its own license.

Software	Preview Server 1.0 or later Microsoft .NET Framework 4 Viz Engine 2.8 or Viz 3.2 and later
Hardware	Licensed dongle
Executable(s)	PreviewServer.exe
<a href="#">Ports and Connections</a>	54000 is used when connecting over http using the REST interface.
Network access	Uses the Bonjour protocol to announce available services.
Operating system	Windows XP 32-bit Windows 7 32-bit or 64-bit

### 3.3.5 Viz Gateway

Viz Gateway is Vizrt's implementation of the MOS Protocol. It enables users of a Newsroom Computer System (NCS) to perform instant updates on playlists in Vizrt's control applications (Viz Trio and VCP).

The Viz Gateway is a framework built to help in newsroom integration tasks. Currently Viz Gateway support NCSs based on the MOS Protocol that is supported by the leading NCS vendors.

Viz Gateway is in principle an extension to the Media Sequencer; hence, any Viz Gateway supported version of the Media Sequencer may be configured to run as a Viz Gateway server using the Viz Gateway configuration tool.

**Table 10:** Viz Gateway specifications

Software	Oracle 10g Runtime Client 32-bit Viz Gateway 1.0.10 or 2.0 (beta)
Executable(s)	scheduler.exe (if running in Console mode)

**Table 10:** Viz Gateway specifications

Ports and Connections	10540 – 10541 (MOS lower and upper port) 10640 (DB event port) 10002 (Viz Gateway Controller Client)
Services	vizgwservice.exe
Local drive access	Access to Oracle client files and folders Read and write access to folder C:\Program Files\vizrt\
Operating system	Windows 2003 32-bit (recommended) Windows XP 32-bit

### 3.3.6 Viz Graphics Hub

Viz Graphics Hub is delivered as a pre-installed system with recommended hardware and software. Viz Graphics Hub must be installed as a separate server; hence, it is not recommended to install anything else on the server that will cause the system to lose performance.

.....  
**Note:** The database directory cannot reside on a remote machine.  
.....

**Table 11:** Viz Graphics Hub specifications

Software	Viz Graphics Hub 2 or later
Executable(s)	VizDbNamingService.exe VizDbTerminal.exe
Ports and Connections	19392–19396 (in most cases the ports are configurable).
Local drive access	Read and write access to database folders (configurable).
Operating system	Windows XP 32-bit

.....  
**IMPORTANT!** Do not run firewall or antivirus scanning software on the server.  
.....

### 3.3.7 Media Sequencer

The Media Sequencer is middleware software primarily used by control applications to connect to for example render engines, newsroom systems and media asset management systems.

More concrete the Media Sequencer is a framework for defining and executing media elements. The media elements are defined in a tree-based schedule that the sequencer interprets. The schedule is saved as an XML file (named **default.xml**). The XML file, in short, contains the configuration settings and the playlists saved to it by the control application.

The Media Sequencer's scheduler has a high-resolution timer that manages a Virtual Document Object Model (VDOM) that contains the schedule. The scheduler

also performs the actual execution of the schedule by interpreting elements describing actions from the VDOM. The communication between end user products and the Media Sequencer mostly goes through a protocol named TreeTalk.

**Table 12:** Media Sequencer specifications

Software	Oracle 10g Runtime Client 32-bit Media Sequencer 1.20
Executable(s)	scheduler.exe (if running in Console mode)
Ports and Connections	8580 (REST), 8594 (Media Sequencer TreeTalk)
Services	scheduler.exe
Local drive access	Access to Oracle client files and folders. Read and write access to the following folders: C:\Program Files\vizrt\ C:\Documents and Settings\All Users\Application Data\Vizrt\Media Sequencer
Network access	Uses the Bonjour protocol to announce available services.
Operating system	Windows 2003 32-bit (recommended) Windows XP 32-bit

### 3.3.8 Viz Trio

The Viz Trio client is the CG operator's user interface. It has a TCP connection to a [Media Sequencer](#). The client gives the user access to creation, editing and playout of pages with graphics. A Viz Engine for local preview renders the graphics within the Viz Trio client.

**Table 13:** Viz Trio specifications

Software	Extra Viz 2 or 3 Plug-ins (only for Viz 2.x) Microsoft .NET Framework 3.5 SP1 (Viz Trio 2.9 and later) Viz Engine 2.8 PL5 HotFix3 or Viz Engine 3.2.8 or later Media Sequencer 1.20 Viz Trio 2.10 client Viz World Client 10.0 or later (see <a href="#">Vizrt Maps</a> ) Optional: Local <a href="#">Media Sequencer</a> and Oracle 10g Runtime Client for connecting to the <a href="#">Viz Content Pilot Database</a> . <i>Optional:</i> Windows Media Player 11 for video clip preview in Viz Object Store.
Executable(s)	trio.exe trionle.exe viz.exe
Operating system	Windows XP SP3 32-bit Windows 7 32-bit or 64-bit



### 3.3.9 Viz Video Hub

Viz Video Hub is a Media Asset Management (MAM) system that allow users to ingest video clips for use in graphics and as full screen video, supporting both SD and HD output.

Viz Video Hub can be used for finding and adding video to graphics using a range of different control applications for both preview and playout on [Viz Engine](#).

.....  
**IMPORTANT!** Before Viz Video Hub is used the first time, it is important to decide upon the playout format. Ingesting and storing other formats on Viz Video Hub may lead to an undefined behavior if they are different from what is configured on [Viz Engine](#).  
.....

**Table 14:** Viz Video Hub specifications

Hardware	Viz Video Hub 2 or later
<a href="#">Ports and Connections</a>	22, 80, 445, 6555

### 3.3.10 Vizrt Maps

Vizrt provides a map solution that offers branded maps and geographic animations using [Viz Curious Maps](#) and [Viz World Client and Server](#).

#### Viz Curious Maps

Viz Curious Maps is ideal for designers, program researchers, and producers who need to create high-quality map animations for news, documentary, promotional videos, and online. It is designed to be simple and intuitive to use, so that users with no specific training in computer graphics, or video editing, can create professional maps on demand and at short notice.

**Table 15:** Viz Curious Maps specifications

Software	Viz Curious Maps 7.2 or later
Executable(s)	WorldMaps.exe
<a href="#">Ports and Connections</a>	80, 8080 (Microsoft Bing and Imagery on Demand) 1947 Sentinel HASP Run-time Environment
Operating system	Windows XP 32-bit

It is possible to run the server and design machine at the same time on the same machine, but it is likely that this will impact the server performance.

.....  
**Note:** Minimum screen resolution is 1280x1024, and it has to be a display size of 96 DPI.  
.....

## Viz World Client and Server

Viz World Client and Server integrates [Viz Curious Maps](#) mapping ability and database into Viz Artist and Viz Engine graphics. By utilizing a set of geographic referencing plug-ins and the maps produced by WoC, the creation of location based graphics using maps, 3D objects, texts, and so on, is seamless.

**Table 16:** Viz World Client and Server specifications

Software	Viz World Client 10.0 or later Viz World Server 10.0 or later
Executable(s)	AxMapsClient.ocx (embedded editor) MapBuilder.exe MapConfigClient.exe ServerAllocator.exe ServerLauncher.exe
<a href="#">Ports and Connections</a>	101, 102, 103 80, 8080 (Microsoft Bing and Imagery on Demand).
Operating system	Windows XP 32-bit (client and server) Windows Server 2003 32-bit (server)

### See Also

- [Viz World Client and Server documentation](#)

---

## 3.4 Shared Data

Vizrt recommends having mapped drive letters for all your shared data. Mapped drives are commonly used for your Viz Engine 2.x data root, Viz Object Store images, Viz Curious Maps cache.

Note that it is possible to configure most systems to use Universal Naming Convention (UNC) as an alternative to mapped drives.

.....  
**Example:** \\hostname\shared folder\resource  
.....

---

## 3.5 Ports and Connections

This section contains information on the following topics:

- [Port Numbers](#)
- [Multiplexing Ports](#)

### 3.5.1 Port Numbers

The table below lists all default server and listening port numbers that are used. It is, if possible, recommended to run the system on a network without a firewall.

**Table 17:** Listening port numbers

Listener	Port(s)	Descriptions and Comments
ArdFTP Viz MPS	21	Used for video transfers from <a href="#">Viz Video Hub</a> to <a href="#">Viz Engine</a> . Also used by Viz MPS (service: FTP).
Viz Video Hub	22	TCP and UDP for logging in to the <a href="#">Viz Video Hub</a> operating system (service: SSH).
Viz World Server	102– 103	102 (TCP) is a Viz World Server listener port for Viz World Client connections when Server Allocator is not in use or only has one Viz World Server running.  103 (TCP) is a Viz World Server listener port for configuration tool connections to the first Viz World Server instance (as configurations are controlled by the first server instance).  See also <a href="#">Vizrt Maps</a> .
Viz Video Hub	137 139	Used for SMB file sharing (service: Netbios)
Viz Video Hub, Microsoft Bing and Imagery on Demand	80, 8080	Web interface and client software. SOAP port for communication with <a href="#">Viz Video Hub</a> .  For download of Microsoft Bing and Imagery on Demand images.  (service: HTTP)
Viz Video Hub	443 445	(service: HTTPS)  TCP & UDP used for SMB file sharing (service: Microsoft-DS).
Viz Mobilize	554	Real Time Streaming Protocol (service: TCP).
Oracle 10g database	1521	For clients that connect to the <a href="#">Viz Content Pilot Database</a> .
Sentinel HASP Run-time Environment	1947	The Sentinel HASP Run-time Environment uses port 1947 to communicate with local and remote components. This relates to hardlock dongles used with Viz Curious Maps.
Viz Video Hub	3080	Low resolution video and index files (service: lighttpd).
Video servers	5250	MVCP and Xlator control port for video servers. Note: this port is only necessary in combination with the video server extension (service: AVCP).

**Table 17:** Listening port numbers

Viz Engine	6100 6700 6800	Ports are used by Media Sequencers that connect to a <a href="#">Viz Engine</a> program and/or preview channel. Viz Engine's default program and preview port is 6100. In a Viz Trio OneBox setup, the default preview port is set to 6800 in order to separate the program and preview channels. In a dual channel setup, the default program ports are 6100 and 6800 for channel 1 and channel 2, respectively. In a dual channel setup, when used for stereo production, the default program ports are 6700 and 6800 for channel 1 (left eye) and channel 2 (right eye), respectively.
Viz Trio	6200 6210	6200 is used for controlling the <a href="#">Viz Trio</a> client over a socket connection. 6210 is used by the Viz NLE plugin to establish a connection to Viz Trio.
Newsroom Component	6220	Used by the Viz NLE Plugin to establish a connection to Viz Content Pilot's Newsroom Component.
Viz NLE Editor	6230	Used by the Viz NLE Plugin to establish a connection to the Viz NLE Editor (on Mac).
Viz NLE Config	6240	Used by the Viz NLE plugin to establish a connection to the Viz NLE Configuration tool (on Mac).
Ticker Service	6300 6301	Ticker handler in Media Sequencer connects to port 6300 for feedback from Ticker Service. Ticker handler in Media Sequencer connect to port 6301 when controlling the ticker via a socket connection.
Viz Content Pilot	6484	Socket connection used for controlling Viz Content Pilot using macro commands.
Viz Video Hub	6555	Message bus port for communication with <a href="#">Viz Video Hub</a> (service: Message bus).
Viz Preview License server	7452	For <a href="#">Viz Engine</a> clients (unlicensed/no dongle) connecting to the Viz Preview License server.
Pilot Data Server	8177	Used when connecting over http using the REST interface.
Media Sequencer	8580 8594	For clients connecting to the <a href="#">Media Sequencer</a> . 8580 is specifically used when connecting over http using the REST interface.
Viz Video Hub	8080	Used for sending keyframes (service: ardok).

**Table 17:** Listening port numbers

Viz Gateway	10001	For DB notification events.
	10002	For <a href="#">Viz Gateway</a> controller clients.
	10540	For MOS object updates.
	10541	For MOS playlist updates.
Viz World Server	10100	10100 (TCP) is a Server Allocator listener port for Viz World Client connections, and is only used in order for clients to get connection details about Viz World Server(s). The first client connection will always be diverted to port 102. In case of multiple server instances port numbers are assigned according to a predefined schema (i.e. 10101, 10102 for server instance 2 and 3 and so on). In case there is no Server Allocator, Viz World Server will itself switch to port 102.  10100 (UDP) is a Viz World Server listener port for Server Allocator communication.  10200 (UDP) is a Server Allocator listener port for Viz World Server communication.  Both UDP ports are internal ports used between the servers.  For more information, please see the <a href="#">Viz World Client and Server 11.1 User's Guide</a> and later. See also <a href="#">Vizrt Maps</a> .
	10200	
Viz Content Pilot	10640	Used by Viz Gateway to establish a connection to Viz Content Pilot in order to send and receive updates on MOS messages (e.g. elements and playlists).
Viz Engine	14300	Alternative port used to avoid conflicts with port 6100 (e.g. when using Viz Multiplexer). Port 6100 is normally used by renderers that are on air, hence, it is (e.g. when running Viz Content Pilot version 4 or Viz NLE Plugin towards Viz 2.x) recommended to use another port.  Port 14300 is an optional port. The default 6100 may also be used if the renderer is not used on air.
Viz Graphics Hub	19392– 19396	Ports in use when connecting to different Viz Graphics Hub components.
Viz Connection Broker	21098	Connection to the Viz Connection Broker configuration interface (e.g. <a href="http://localhost:21098/">http://localhost:21098/</a> )

**Table 17:** Listening port numbers

Viz Engine	50007 50008 50009 50010	Ports 50007 – 50009 are all <a href="#">Multiplexing Ports</a> that enables Viz Engine to work on other scenes in sessions that are used for preview purposes. <b>Isolated port (50007):</b> All connections to this port get its own session. <b>Shared port (50008):</b> All connections from one single host shares one session. <b>Fixed port (50009):</b> Same as shared port except that allocated resources are never cleared from memory. <b>On Air Preview port (50010):</b> Enables a user to request a preview of the next scene to be put on air while another scene is on air.
Preview Server	54000	Used when connecting over http using the REST interface.

### 3.5.2 Multiplexing Ports

For Viz Engine 2 users an external application named Viz Multiplexer is used to allow multiple users connect to Viz for graphics preview. Viz Multiplexer acts as a special proxy between the client software and Viz, and stores the engine's state for each connected client and restores/updates that state each time a client sends a command to Viz.

For Viz Engine 3 users the multiplexer functionality is an integral part of Viz Engine; hence, there is no need for additional installation and configuration. When using Viz Engine a session management takes place internally, with one default session for the GUI and internal/external commands, and additional sessions created on-demand for the multiplexing ports or the preview port.

With multiplexing ports other than 50010 the Viz Engine state is only switched on when a command is received; hence, ten consecutive commands from a client will only result in one state switch on the first command.

With 50010 the state is switched when a command is received and immediately switched back to the main session such that on air rendering will not be hindered in any way.

The 50009 port is traditionally used by Viz Content Pilot 4's Newsroom Component, and is the same as the shared port (50008) except that allocated resources are never cleared from memory. To avoid memory overload, it is recommended to clean up the Viz Engine regularly when using this port.

The 50008 port is a shared port where all connections from one single host shares one session. It is most often used by Viz Trio and the Newsroom Component to display preview frames.

The 50007 port is an isolated port where all connections get their own session. It is used, for example in an NLE setup, to deliver frames to the host NLE-system when rendering or scrubbing video clips with graphics. Using this port will also

suppress bounding box commands. Note that 50007 cannot be used by the Newsroom Component.

---

**Note:** All multiplexing ports are supported by all Viz Engine versions

---





---

## 4 Storage and Backup Recommendations

This section gives some general recommendations on storage and usage of data which can be divided in two main parts;

- Graphics, image and video storage and settings
- Control room and newsroom configuration storage and settings

The following sub sections divide these parts into more user specific areas in order to give an overview of what is stored where and what is important to maintain for each installation.

This section contains information on the following topics:

- [Graphics Data](#)
- [Viz Content Pilot](#)
- [Media Sequencer](#)
- [Viz Trio](#)

---

### 4.1 Graphics Data

Graphics data is where the scenes, images, fonts, objects, materials and animations are stored.

It is *important* to regularly take backups of the data as data is basically stored in a normal Windows directory structure.

Data is organized by graphics designers using Viz Artist. Data can also be organized using the Viz Graphic Hub Manager tool. Graphics data is structured as file objects on Viz Graphic Hub.

Default location for data is:

- Windows 7: %LOCALAPPDATA%\VirtualStore\Program Files (x86)\vizrt\VizDb
- Windows XP: C:\Program Files\vizrt\

.....  
**IMPORTANT!** In Windows 7 and later, it is not possible for a user to edit or save files directly in the program folder (C:\Program Files), so data must be stored in the VirtualStore (%LOCALAPPDATA%\VirtualStore) and appropriate backup and restore operations must be done on those files.  
.....

The location of the data root is configured by Viz Engine's Viz Config tool, while the location of the database is configured by the Viz GH Terminal. Both data sets should ideally be placed on a dedicated machine.

.....  
**IMPORTANT!** Only offline backup operations are allowed as the server must be shut down during a backup process.  
.....

## See Also

- *Viz Graphic Hub User's Guide*

---

## 4.2 Viz Content Pilot

In addition to information stored on the Media Sequencer and VCP's database, VCP's INI files and playlists are important to backup when needed. INI files are used for local configurations for the application itself, third party integrations, and overrides of database settings. All VCP playlists, both local and newsroom playlists, can be exported and imported for backup and transfer to other VCP installations.

The INI file *ContentPilot.ini* is used by VCP, Viz Template Wizard, and Viz Object Store. VCP's Thumbnail Generator has its own INI file named *ThumbnailGenerator.ini*.

Original versions of the INI files are stored in the program folder during installation

- Windows 7: C:\Program Files (x86)\Vizrt\Viz Content Pilot 5.x
- Windows XP: C:\Program Files\Vizrt\Viz Content Pilot 5.x

However, in Windows 7 and later, it is not possible for a user to edit or save the file directly in the program folder, so subsequent edits are made to a copy stored in the VirtualStore:

- %LOCALAPPDATA%\VirtualStore\Program Files (x86)\Vizrt\Viz Content Pilot 5.x

-----  
**IMPORTANT!** If using Windows 7 or later, backup and restore operations must be done to the INI files in the VirtualStore.  
-----

---

## 4.3 Media Sequencer

The file *default.xml* is a Media Sequencer file.

Microsoft Windows XP:

- C:\Documents and Settings\All Users\Application Data\Vizrt\Media Sequencer

Microsoft Windows Vista and 7:

- C:\ProgramData\Vizrt\Media Sequencer

The file holds a stored copy – of the memory held information – over control application client settings, user data for graphics templates, references to scenes on the database, stills in the still store and video clips.

For a backup it is recommended to set up the system, then save a copy of the *default.xml* remotely to more easily set up a new build of a similar machine in the event of a system failure.

The *default.xml* file stores all persistent data, and at regular intervals the Media Sequencer creates a backup file named *default.xml.1*, which again replaces the

file *default.xml.2*. In a failure situation where *default.xml* may get corrupted, *default.xml.1* is used to create a new instance of *default.xml*. If *default.xml.1* is also corrupted it will try to use *default.xml.2*.

---

## 4.4 Viz Trio

Unlike Viz Content Pilot (VCP), Viz Trio does not use an INI file or a database (by default). Viz Trio is dependent on the Media Sequencer to keep local client specific configurations intact. The only exception is command line options used locally for each client. As with VCP, Viz Trio is able to export and import shows (similar to VCP's playlists).

Generally it is recommended to do regular show exports in order to preserve the show if something is corrupt. A show export is performed using Viz Trio to create an archive of the show, its graphics templates, pages, settings and so on. For more information, see Viz Trio's user guide on show export and import and the available options. For graphics export or archiving, see Viz Artist's user guide.

It is also recommended to backup, at regular intervals, the Media Sequencer's *default.xml* file as it stores a lot of the data used by Viz Trio and other control applications, except graphics data and external files, such as file scripts. This enables the system to be restored to a previous state. If the system must be reset entirely, a new *default.xml* file will be created. The latter requires that the Media Sequencer is shut down, the *default.xml* file deleted or removed, and the Media Sequencer restarted. This also requires that all configurations, shows with graphics and script references are setup.

If Viz Trio is used in combination with Viz Content Pilot, see the Viz Content Pilot User's Guide for further information on use of VCP's database and still store.

---

## 5 Installation

The following sections describe the different steps that are needed in order to have a multiple Viz Trio client setup with a remote Media Sequencer connected to one or several Viz Engine output renderers.

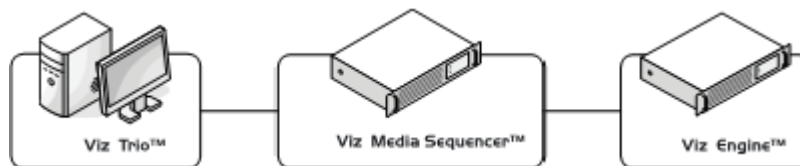
-----  
**Note:** It is recommended that the installation is performed in the same order as the sections below.  
-----

This section contains information on the following topics:

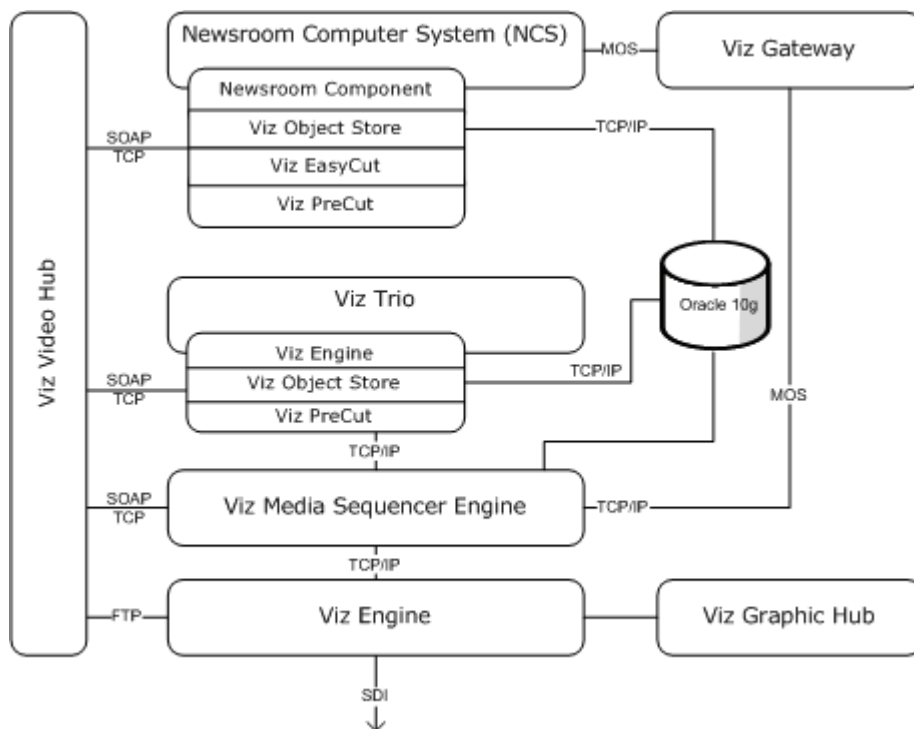
- [System Overview](#)
- [Prerequisites](#)
- [Viz Engine](#)
- [Media Sequencer](#)
- [Viz Trio](#)

---

### 5.1 System Overview



The system overview shown above depicts a basic setup of Viz Trio. Typically the Viz Trio machine also has a Viz Engine for local preview.



A more advanced setup is the newsroom and video integration which involves several other Vizrt products such as Viz Video Hub, Viz Content Pilot, Viz Gateway and third party systems such as a newsroom and database system.

As a replacement on the newsroom side it is also possible to integrate Viz Trio with Avid iNEWS ControlAir using Viz Device Manger instead of Viz Gateway.

**Note:** The control application can also be Viz Content Pilot; however, Viz Content Pilot will also have a TCP/IP connection to the Oracle database.

---

## 5.2 Prerequisites

Before installing a Viz Trio system make sure that the correct hardware and latest software is available. All needed software is accessible from Vizrt's FTP server, and contains the latest official releases of all Vizrt software.

### See Also

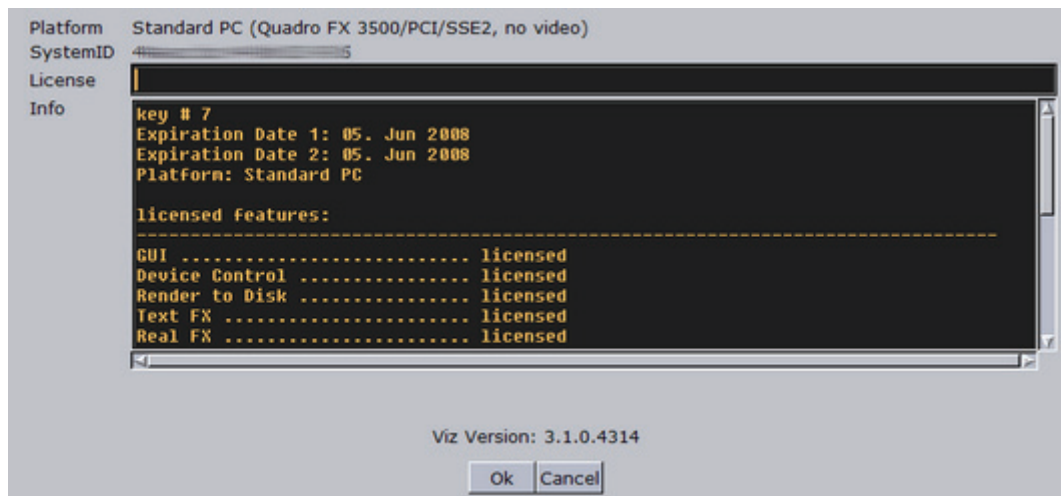
- [Hardware Requirements](#)
- [Viz Trio software specifications](#)
- [Ports and Connections](#)

---

## 5.3 Viz Engine

The Viz Engine is the output machine for Viz Trio, and is a separate installer. A renderer is needed in order for Viz Trio to preview graphics before payout.

**Figure 1:** Viz Engine's information window



A licensed Viz dongle must be attached to the USB or printer port on the machine. After installing Viz Engine, plug-ins such as Extra Viz plug-ins and so on, must be installed on the machine. Plug-ins needed will vary depending on the scene design and integrations needed.

#### To install Viz Engine

- Run the installer and follow the directions given by the installation wizard.

#### To activate the license key

1. Start Viz Engine and click the **information ( i )** button, positioned in the upper right corner of the interface, to open [Viz Engine's information window](#).
2. Add the key to the License field, and click **OK**.

#### See Also

- [Viz Engine requirements](#)
- [Viz Artist User's Guide](#)
- [Viz Engine Administrator's Guide](#)

---

## 5.4 Media Sequencer

The Media Sequencer stores Viz Trio references and cache information. It is in most cases installed on a server acting as a middle tier between Viz Trio and Viz Engine.

#### To install the Media Sequencer

1. Run the installer and follow the directions given by the installation wizard.
2. *Optional:* Install the Oracle 10g Runtime Client.
  - Restart the Media Sequencer after installing the database client.
  - A database client is needed when connected to Viz Gateway for accessing the Viz Content Pilot database.

## See Also

- [Media Sequencer](#) requirements

---

## 5.5 Viz Trio

Viz Trio is the control application used by the operator to manage Viz Trio shows and/or playlists from Viz Content Pilot and its integrated newsroom systems.

Before installing Viz Trio, install the Viz Engine, Extra Viz plug-ins, .NET Framework and the hardware dongle that is needed in order to run graphics preview within Viz Trio.

If running a single client setup, perform the Media Sequencer installation as well.

-----  
**Note:** Viz Trio versions 2.9 and later requires the Microsoft .NET Framework 3.5 Service Pack 1 or later.  
-----

### To install Viz Trio

- Run the installer and follow the directions given by the installation wizard.

This section contains information on the following topics:

- [Keyboard](#)
- [Viz Content Pilot](#)

## See Also

- [Viz Engine](#) installation
- [Media Sequencer](#) installation
- [Viz Trio](#) requirements

### 5.5.1 Keyboard

To install the Viz Trio keyboard, the keyboard's USB connector must be connected to the computer's USB port. The computer should be able to detect and automatically install the needed keyboard driver(s).

-----  
**Note:** To learn more about the keyboard's technical specifications, visit [DevlinGroup – Devlin Electronics Limited](#).  
-----

### To import the keyboard mapping file



1. Open Viz Trio and open the Show settings window and click the Keyboard button.
2. Click Import to open the Import Keyboard Shortcuts dialog.
3. Click the folder button to browse for and import the *keyboard* file (extension \*.kbd).

4. *Optional*: Select to merge the existing keyboard shortcuts with the new ones.
5. Click Import.

**See Also**

- [Viz Trio Keyboard](#)

## 5.5.2 Viz Content Pilot

In order to connect to Viz Content Pilot's Oracle database, a runtime installation of an Oracle database client is required. It is also recommended to use the same client version as the database. After the client is installed, Viz Trio's database connection can be configured.

**See Also**

- [Viz Content Pilot Database](#)





---

## 6 Getting Started

This section describes how to connect Viz Trio to the Media Sequencer and Viz Engine. It also describes how to setup Viz Trio for playout.

Before the Viz Trio client is started make sure that it has a Media Sequencer, Viz Engine and Viz Engine database or data root configured and running.

This section contains information on the following topics:

- [Basic Configuration and Startup](#)
- [Configuring the Graphics Database](#)
- [Configuring the Media Sequencer](#)

---

### 6.1 Basic Configuration and Startup

In order to start Viz Trio successfully, it needs a connection to the Media Sequencer and the local Viz Engine.

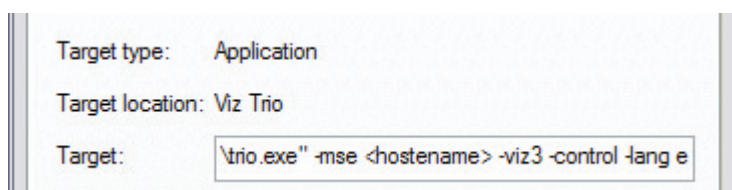
This section contains information on the following topics:

- [Connecting to Media Sequencer](#)
- [Connecting to Viz Engine](#)
- [Starting Viz Trio](#)

#### 6.1.1 Connecting to Media Sequencer

Which connection a Viz Trio client has to the Media Sequencer is decided by parameters in the program target path. If no Media Sequencer target path parameter is set, Viz Trio will default to a local Media Sequencer.

**To configure the Media Sequencer**



- Right-click the Viz Trio desktop shortcut and set a parameter for the `-mse` setting in the **Target** path.

-----  
**Example:** C:\Program Files\Vizrt\Viz Trio\trio.exe -mse <hostname>  
-----

**See Also**

- [Command Line Parameters](#)
- [Configuring the Media Sequencer](#)

## 6.1.2 Connecting to Viz Engine

Viz Trio needs a local Viz Engine to run its preview and to be able to import scenes. Viz Trio will detect and run Viz Engine itself; however, if the machine has both Viz Engine 2 and 3 versions installed a command line option must be used to separate the two.

-----  
**Example:** C:\Program Files\Vizrt\Viz Trio\trio.exe -viz3  
-----

As with the Media Sequencer, Viz Trio also has a command line option for selecting the Viz versions; however, Viz Engine 2.x does not need a target path parameter as it is used by default if no other is specified.

All machines should be installed with the same [Viz Engine](#) version, and also share the same data.

-----  
**IMPORTANT!** For local preview, Viz Trio must have a hardware dongle installed.  
-----

### See Also

- [Command Line Parameters](#)
- [Viz Engine](#) requirements
- [Configuring the Graphics Database](#)

## 6.1.3 Starting Viz Trio



1. Double-click the icon on the desktop, or
2. Select the program from the Start menu (All Programs > Vizrt > Viz Trio > Viz Trio Client), or
3. Select the program from the program directory.

-----  
**Note:** For Viz Trio 2.9 and later the path is C:\Program Files\Vizrt\Viz Trio. For older versions it is C:\Program Files\vizrt\Trio Client  
-----

### See Also

- [Command Line Parameters](#)
- [Configuring the Graphics Database](#)
- [Configuring the Media Sequencer](#)

---

## 6.2 Configuring the Graphics Database

Viz Trio must be configured to access the same graphics data as the program and preview Viz Engine channels. The configuration is therefore used for local preview in order to test and preview graphics before it is added to a show or playlist and played out on-air.

Viz Engine 2.x and 3.x have two different data setups, and are referred differently. A Viz Engine 2's data is commonly referred to as the *data root*. Viz Engine 3's data is stored on the Viz Graphic Hub.

As all Viz Artist and Viz Engine machines in a Viz Trio system must use the same data, it is also recommended that the data is located on a high performance machine.

*For Viz 2.x:* Configuring the data root for Viz Trio using Viz Engine's Viz Config tool is the only way to configure Viz Trio's access to graphics data using Viz 2.x.

*For Viz 3.x:* Configuring the Viz Graphics Hub database for Viz Trio using Viz Engine's Viz Config tool is the recommended way to configure Viz Trio's access to graphics data. An alternative is to configure it through Viz Trio.

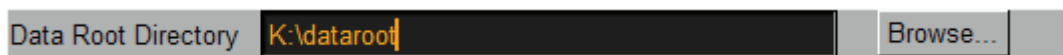
*For Viz 3.x:* Configuring the Viz Graphics Hub database using Viz Trio can be done in three ways; at startup using the Viz Engine login window, through Viz Trio's own configuration interface, or by setting command line options in Viz Trio's target path.

-----  
**Note:** If Auto Login is set for Viz 3.x it is not possible for Viz Trio to alter the settings; however, this does not apply if Auto Login is set in the Viz Trio interface.  
-----

This section contains information on the following topics:

- [To configure the Viz 2.x data root](#)
- [To configure the Viz 3.x database using Viz Config](#)
- [To configure the Viz 3.x database using Viz Engine's login window](#)
- [To configure the Viz 3.x database using Viz Trio's configuration window](#)
- [To configure the Viz 3.x database using Viz Trio's target path](#)

### To configure the Viz 2.x data root

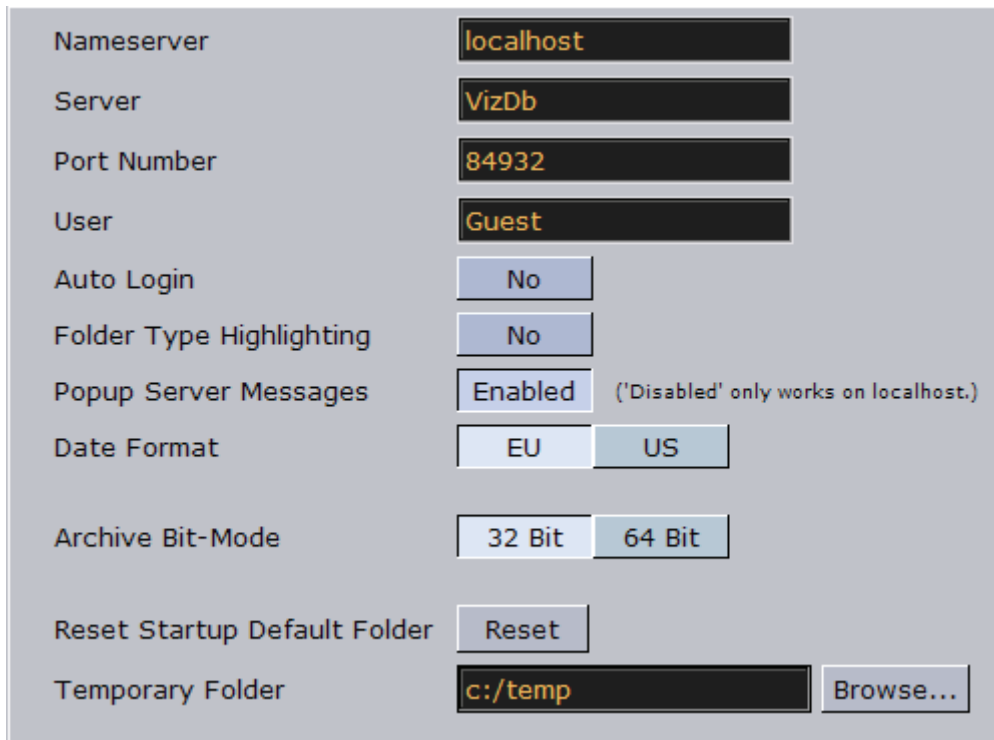


1. Start Viz Config and select the **Data Root Directory** section.
2. Click the **Browse...** button and select the data root directory, or enter it manually.
3. Click **Save**, and close the application.

-----  
**Note:** A shared data root directory can be mapped as a network drive (recommended) or entered as a Universal Naming Convention (UNC) path.  
-----

-----  
**Example:** K:\<dataroot folder> *or* \\<hostname>\<dataroot folder>  
-----

### To configure the Viz 3.x database using Viz Config



The screenshot shows the 'Viz Config' tool interface for database configuration. The settings are as follows:

Nameserver	localhost
Server	VizDb
Port Number	84932
User	Guest
Auto Login	No
Folder Type Highlighting	No
Popup Server Messages	Enabled ('Disabled' only works on localhost.)
Date Format	EU US
Archive Bit-Mode	32 Bit 64 Bit
Reset Startup Default Folder	Reset
Temporary Folder	c:/temp Browse...

1. Start the **Viz Config** tool and select the Database section.
2. Enter the following connection parameters:
  - **Nameserver** – Sets the database hostname.
  - **Server** – Sets the database name.
  - **Port number** – Sets the database port number. Default port is **84932**.
  - **User** – Sets the database user.
  - **Auto Login** – Sets login to automatic when starting Viz Engine. If Auto Login is enabled it will override Viz Trio's database login and disable its *Database (Viz 3)* configuration section.
3. Click **Save**, and exit the application.

### To configure the Viz 3.x database using Viz Engine's login window

1. Start **Viz Trio** with the following target parameters: **-viz3**
2. During startup a login window for the database will appear.
3. Enter the following connection parameters:
  - Name server host, database, username and password.
4. Check the **Auto Login** option.
5. Click **OK** to continue.

### To configure the Viz 3.x database using Viz Trio's configuration window

1. Start **Viz Trio** with the following target parameters: **-control -viz3**.
2. Click Viz Trio's **Config** button (upper left) to open the [Configuration Interface](#).
3. Select the **Database (Viz 3)** section, and enter the following parameters:
  - Name server host, database, username and password.

4. Check the **Auto Login** option.
5. Click **Apply** to save the settings, and exit the [Configuration Interface](#).

The change of database is immediate, however, to see the change when importing scenes it might be necessary to refresh the *Import Scenes* view.

**Note:** If Auto Login is set using the Viz Config tool or Viz Engine login window (not Viz Trio) the Viz Trio *Database (Viz 3)* configuration section will be disabled.

#### To configure the Viz 3.x database using Viz Trio's target path

- Add the following parameters to Viz Trio's target path: `-vizdb <host>:<server>:<username>:<password>`

**Note:** Command line options will not override the database configuration if Auto Login is set using the Viz Config tool or Viz Engine login window (not Viz Trio), however, it will override settings configured using Viz Trio.

**Tip:** Check that the correct database is available when selecting Import Scenes.

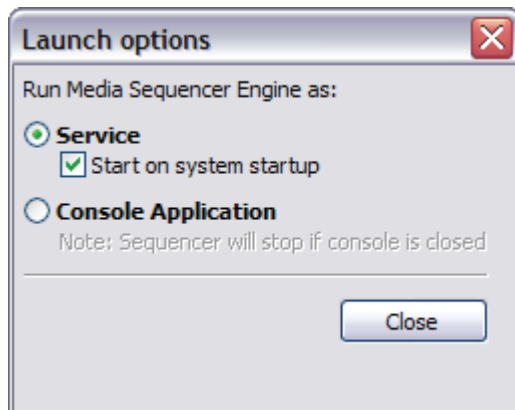
#### See Also

- [Requirements](#)
- [Graphics Data](#) storage and backup recommendations
- [Viz Link Administrator's Guide](#) for configuration of video playout using Viz Video Hub
- [Viz Engine Administrator's Guide](#)

---

## 6.3 Configuring the Media Sequencer

Figure 2: Media Sequencer Launch options

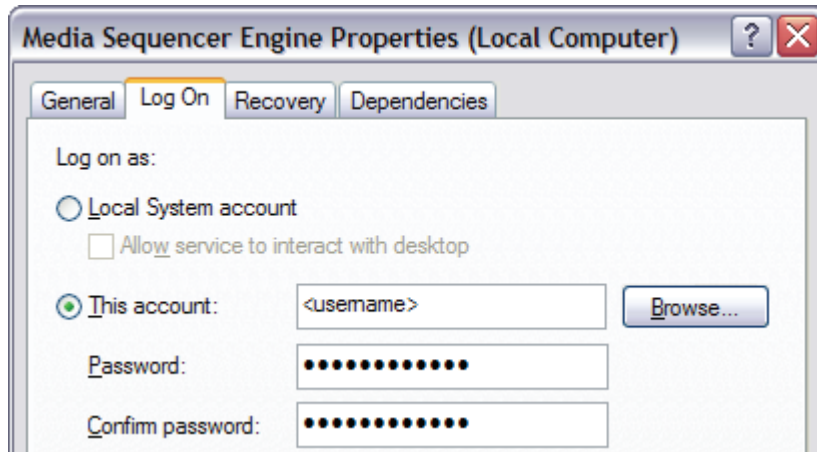


The Media Sequencer is a core component for managing shows and playlists.

The Media Sequencer can run in either service or console mode. It is generally recommended to run the Media Sequencer as a console application when installed on a server as this will enable a system administrator to monitor the application's

console. If the Media Sequencer is installed on the operator's control PC/ workstation it is recommended to run it as a service; however, mode selection also depends on the use. It is therefore recommended to run the Media Sequencer on a dedicated server for optimal performance.

**Figure 3:** Media Sequencer "Log on" service properties



When using Viz Content Pilot's Oracle database an Oracle database client is needed. If the Media Sequencer is running as a service, it is recommended that the service logs on with a user account, and not with the default Local System account (SYSTEM). This is caused by the fact that Oracle's OCI library is installed per user, hence the System user is not able to read environment variables set for a user.

#### **To allow the Media Sequencer service to log on with a user account**

1. Open the **Administrative Tools** found under Windows' Control Panel
2. Open the **Services** window.
3. Select and right-click the **Media Sequencer** service, and from the appearing context menu select **Properties**.
4. In the appearing dialog box, select the **Log On** tab.
5. Select the **This account** option and enter the **account information**.
6. When done, click **OK**.

---

# 7 User Interface

This section describes the Viz Trio client's user interface for playout. The Viz Trio client's [Designer](#) and [Configuration Interface](#) is not described in this section.

This section contains information on the following topics:

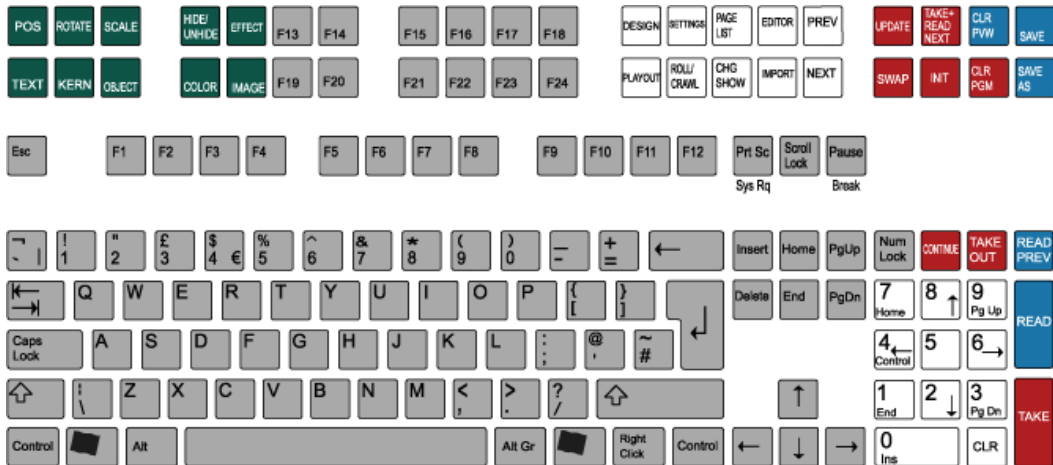
- [Create New Scroll](#)
- [Graphics Preview](#)
- [Import Scenes](#)
- [On Air Mode](#)
- [Page Editor](#)
- [Page List](#)
- [Playlist Modes](#)
- [Playout and Design Mode](#)
- [Render Videoclip](#)
- [Search Media](#)
- [Show Control](#)
- [Show Modes](#)
- [Slave Mode](#)
- [Snapshot](#)
- [Status Bar](#)
- [Tab Fields Window](#)
- [Template List](#)
- [TimeCode Monitor](#)
- [Video Preview](#)
- [Viz Artist Mode](#)
- [Viz Trio Keyboard](#)

---

## 7.1 Viz Trio Keyboard

Viz Trio is designed to be operated by mouse and keyboard or by keyboard only. In the explanation below, an overview of the Viz Trio keyboard is described. For users with the old Cherry keyboard, please refer to the [Cherry Keyboard](#) chapter.





The keyboard contains two rows with extra function keys which have been assigned to different Viz Trio actions. The keyboard has its own configuration software. A Viz Trio configuration file must be loaded to create the correct keyboard map. In the Viz Trio client a keyboard mapping file must be imported to assign the correct actions to the keys. This is pre-installed on all Viz Trio clients, so normally these settings will not have to be changed.

For information on how to import a keyboard mapping file and assign shortcuts, see the [Keyboard Shortcuts and Macros](#) section.

This section contains information on the following topics:

- [Editing Keys](#)
- [Navigation Keys](#)
- [Program and Preview Keys](#)

### Editing Keys

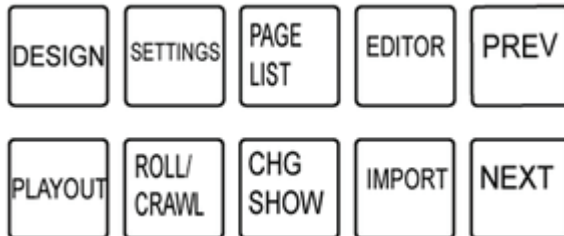


The green keys all perform editing operations. The current tab field must have the property of the key exposed for editing. If not, the key will have no effect and an error message will be written to the log file when the key is pressed.

- **POS:** Displays the position editor.
- **TEXT:** Displays the text editor.
- **ROTATE:** Displays the rotation editor.
- **KERN:** Displays the character kerning editor.
- **SCALE:** Displays the scaling editor.
- **OBJECT:** Displays the object pool where one can browse for 2D and 3D objects.
- **HIDE/UNHIDE:** Hides/shows the tab field.

- **COLOR:** Shows the pool of colors.
- **EFFECT:** Opens the Effect editor.
- **IMAGE:** Opens the Image pool.

### Navigation Keys



The white keys allow a switch between different views and editors in the program.

- **DESIGN:** Opens Viz Trio in design mode displaying the designer tool user interface.
- **PLAYOUT:** Opens Viz Trio in playout mode displaying the playout user interface.
- **SETTINGS:** Displays the Show Settings dialog.
- **ROLL/CRAWL:** Opens the Scroll editor.
- **PAGE LIST:** Displays the Page List.
- **CHG SHOW:** Opens the Open Show window.
- **EDITOR:** Displays the Page editor.
- **IMPORT:** Opens the Import Scenes dialog.
- **PREV:** If extra page views are defined, this key shows the view above the one currently active, see the [Add Page List View](#) section.
- **NEXT:** If extra page views are defined, this key shows the view below the one currently active.

### Program and Preview Keys



The red function keys all affect actions on the program channel, and the blue the preview channel. The blue keys all affect actions on the preview channel.

- **UPDATE:** When a change has been done to a page that is already on-air, hitting update will merge in the changes without running any animations. This is typically used for fixing typing errors. If a page is updated and Take is used instead of Update, all animation directors in the scene will be executed which normally creates an unwanted “hard cut” effect.
- **TAKE+ READ NEXT:** Takes the page currently read, and reads the next one in the list.

- **SWAP:** The swap key takes to air what is currently read and visible on the preview, and it takes of what is currently on-air and reads that page again.
- **INIT:** Initializes the current show on both the program and preview channel.
- **CLR PGM:** Clears all loaded content on the program channel.



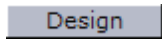
- **CLR PVW:** Clears the preview channel.
- **SAVE:** Saves the page currently shown on the preview channel.
- **SAVE AS:** Saves the page currently shown on the preview channel to the page number typed in.
- **CONTINUE:** When a page halts at a stop point, hitting the Continue key will make the animation continue.
- **TAKE OUT:** If transition logic is used, the Take Out key will take out any page loaded in the layer that is currently read. If transition logic is not used, the Take Out key will perform a “clear” which will be a “hard cut”. To obtain a smooth out animation, the scene must be designed with a stop point and an “out animation” and the Continue key must be used to take out the page.
- **READ PREV:** Reads the previous page in the page list.
- **READ:** Reads the page currently highlighted by the cursor.
- **TAKE:** Performs a Take on the page that is currently read, and plays it out.

### See Also

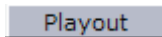
- [Cherry Keyboard](#)
- [Page Editor](#)

---

## 7.2 Playout and Design Mode



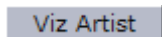
Selecting the Design button in the upper right corner, when in playout mode, will switch the user interface to design mode, displaying the [Designer](#) tool. Viz Trio provides a library of pre-defined design objects for easy creation of new scenes. The various design objects are grouped together in order to create more advanced graphics.



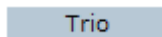
Click the Playout button (upper right corner of the [Designer](#) tool) to return to playout mode.

---

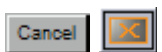
## 7.3 Viz Artist Mode



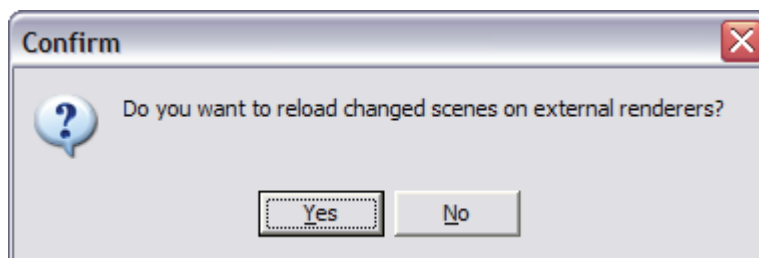
Viz Trio templates are based on scenes created in Viz Artist. By clicking the Viz Artist button, Viz Artist is started, opening the template's scene that is currently loaded in Viz Trio's preview window. In Viz Artist it is possible to edit the scene, and save the changes. The Viz Artist button is available in both [See Also](#).



When in Viz Artist mode, Viz Artist will have a button labelled Trio in the upper right corner. Click the Trio button to return to Viz Trio after making changes to the scene.



To return to playout mode without saving the changes, press the Cancel or X button in the upper right corner. The Cancel button belongs to Viz Artist version 2.x, and the X button to Viz Artist version 3.x.



When back in playout mode a confirmation dialog appears, asking if the page on the program and preview renderers should be reloaded. The template is reimported automatically.

---

## 7.4 On Air Mode



On the top bar to the right there is a button that takes Viz Trio on and off-air. When in on-air mode, the button is red.

---

## 7.5 Slave Mode



In the upper right corner of the Viz Trio client there is a *Slave* button. Click this button to enable slave mode. When in slave mode, the button is yellow. This means that any other Viz Trio client connected to the same Media Sequencer, which is not in slave mode, is in effect master.

A master will trigger the slaves to change the show folder when they change. This functionality typically make sense when the commands to change the folder are issued from a newsroom system, the newsroom system is then master.

To exit slave mode, press the button again.

---

## 7.6 Show Modes

Viz Trio supports two show modes; traditional shows where templates and pages are part of the same view, and [Context-enabled Shows](#) where templates and pages are split in two views.

This section contains informaiton on the following topics:

- [Context-enabled Shows](#)
- [User-defined Contexts](#)

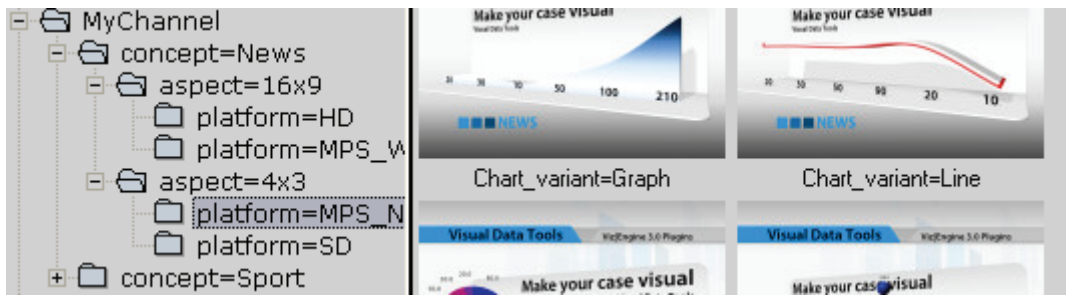
### 7.6.1 Context-enabled Shows

A context-enabled show has the possibility to switch concepts and variants within a show. For example a News concept could be switched to a Sports concept. Within that concept there could be variants of a specific scene (e.g. lower and top third).

Graphics for all these variations must be made, but once they are available and imported into Viz Trio, they can literally be switched and played out by the touch of a button.

All variants of a scene, within a specific concept, will be shown as one template in Viz Trio's [Template List](#).

**Figure 4:** Viz tree structure



For a show to successfully use context-enabled scenes in Viz Trio, the folders and scene names in Viz must follow a certain naming convention, giving them properties and values for Viz Trio to recognize and use.

The conventions are as follows:

**Table 18:**

Type	Property	Name	Example
Concept (folder)	concept	= <conceptname>	concept=<conceptname> concept=News concept=Sport concept=Weather
Context (folder) User-defined	<context>	= <contextname>	<context>=<contextname> aspect=16x9 aspect=4x3 platform=HD platform=SD
Variant (file)	_variant	= <variantname>	<scene>_variant=<variantname> banner_variant=Lower banner_variant=Top chart_variant=Graph chart_variant=Line adds_variant=Adds adds_variant=NoAdds weather_variant=3DayForecast weather_variant=1DayForecast

**IMPORTANT!** For the variants it is important to add an underscore “\_” to separate the scene name from the variant’s keyword.

## 7.6.2 User-defined Contexts

Viz Trio supports the creation of user-defined contexts. The number of user-defined contexts are unlimited. However, user-defined contexts are only controllable through the Viz Trio commands command-line editor. The examples below show Viz Trio commands used with the user-defined contexts *Aspect* and *Platform*:

```
show:set_context_variable Aspect 16x9
show:set_context_variable Platform HD
show:set_context_variable X Y
```

**Tip:** User-defined contexts can be controlled using Viz Trio commands, which in turn can be used to create use-defined macro commands for use with scripts and shortcut keys.

### See Also

- [Macro Commands and Events](#), and in particular the [show](#) commands.

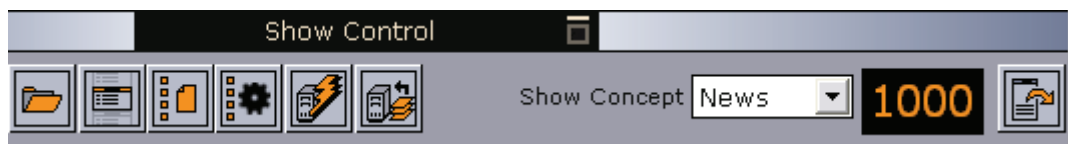
---

## 7.7 Show Control

The Show Control user interface is used to manage the show before, during and after a show is taken on-air.

The Show control gives access to previous shows, Viz Content Pilot playlists, newsroom playlists and the possibility to create new shows and show playlists.

Note that there are differences between a show, show playlist and a playlist, especially in the way they are managed and monitored. This section and the following sub-sections will describe some of these differences.



The Show Control view contains several buttons that in turn opens different options. In short they can be described as such:

- **Change Show:** Displays the [Show Directories](#) pane.
- **Add Page View:** Opens the [Add Page List View](#) dialog box.
- **Create Playlist:** Opens the [Create Playlist](#) dialog box.
- **Show Settings:** Opens the [Show Settings](#) window.
- **Cleanup Renderers:** Cleans up all data on the renderer(s) (see the [Cleanup Channels](#) section).
- **Initialize:** Initializes the show on the renderer(s) (see the [Initialize Channels](#) section).
- **Show Concept:** Displays the show's concept, for example Sport or News.
- **Callup Code:** Shows the callup code for the next page to read from the page list.
- **Read Page:** Reads the page with the callup code shown in the Page to read window.

This section contains information on the following topics:

- [Add Page List View](#)
- [Callup Page](#)
- [Cleanup Channels](#)
- [Create Playlist](#)
- [Initialize Channels](#)
- [Show Concept](#)
- [Show Directories](#)
- [Show Settings](#)

### 7.7.1 Show Directories



Show Directories by default displays two tabs. A [Show Types](#) and a [Playlists](#) tab. However, for compatibility with previous Viz Trio versions, a [Viz Directories](#) view is also available.

Clicking the **Change Show** button displays the Show Directories, and by default the Viz Trio [Show Types](#).

This section contains information on the following topics:

- [Viz Directories](#)
- [Show Types](#)
- [Playlists](#)
- [Import Show](#)
- [Export Show](#)
- [To create a show folder](#)
- [To create a new show](#)
- [To import an exported show](#)
- [To export the current show](#)

#### Viz Directories

The Show logic is the recommended way of organizing pages. However, it is also possible to set the path by choosing a Viz Artist folder for each show. To do this, choose the **Viz directories** tab below the Show directories heading.

---

**Note:** The Viz directories tab is by default disabled. Open the Configurations, and in the Miscellaneous/User Restrictions section enable "*Browse viz directories when changing shows*".

---





When the Viz Artist folder is set and pages are imported, a show folder with the same path and name will be created in the Shows view. To keep using the Viz Artist directory workflow, where each show is tied to a Viz Artist folder, simply ignore the Shows view.

It is possible to switch at anytime to using the Shows view logic. However, when first using the Shows view logic, and creating show folders with paths that differ from the Viz Artist scene data structure and there is more than one show per folder, these shows can only be accessed using the Shows view.

### Show Types

There are mainly two types of shows. The traditional show with scene- and transition-logic based templates, and the type of show that also is context-enabled which categorizes the same templates in concepts with variants of the same template or scene. Shows also has the possibility to create a [Create Playlist](#) in order to layout pages in a carousel like manner.

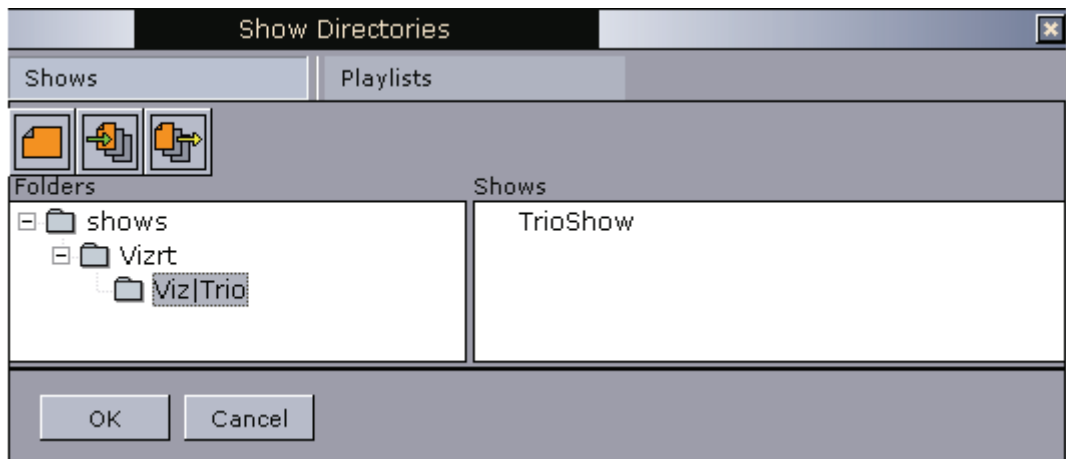
---

**Note:** A traditional show can be converted to a concept and variant enabled show using the Trio command: `show:enable_context`. However, it is not possible to convert back.

---

All scenes added to a show are added to the Templates list for context-enabled shows or the page list for traditional non context-enabled shows.

A show folder is used to organize shows that for instance belong to a certain show or production. Scenes from the whole Viz Artist scene tree can be imported into a show.



The two panes, Folders and Shows, have context menus that provide some options for show management.



The buttons are used for creating new shows, and to import and export shows. The Import and Export Show buttons allow a show to be exported or imported with all Viz Artist archives, Viz Trio pages, page views, local macros and key bindings, database setups, script units and so on.

### Context Menu

Folders pane:

- **Add folder:** Creates a new show folder at any level in the tree structure.
- **Rename folder:** Renames any folder at any level in the tree structure.
- **Delete folder:** Deletes any folder at any level in the tree structure.

-----  
**CAUTION!** Export or delete all shows and sub-folders before deleting a root folder.  
 -----

Shows pane:

- **Rename Show:** Renames an existing show.
- **Create Show:** Creates a new show.
- **Delete Show:** Deletes an existing show.
- **Export Show:** Exports an existing show. Selecting this option opens the [Export Show](#) window.

It is possible to **move a show** to a different Show folder by drag and drop. Select a show and drag and drop it into the new folder location.

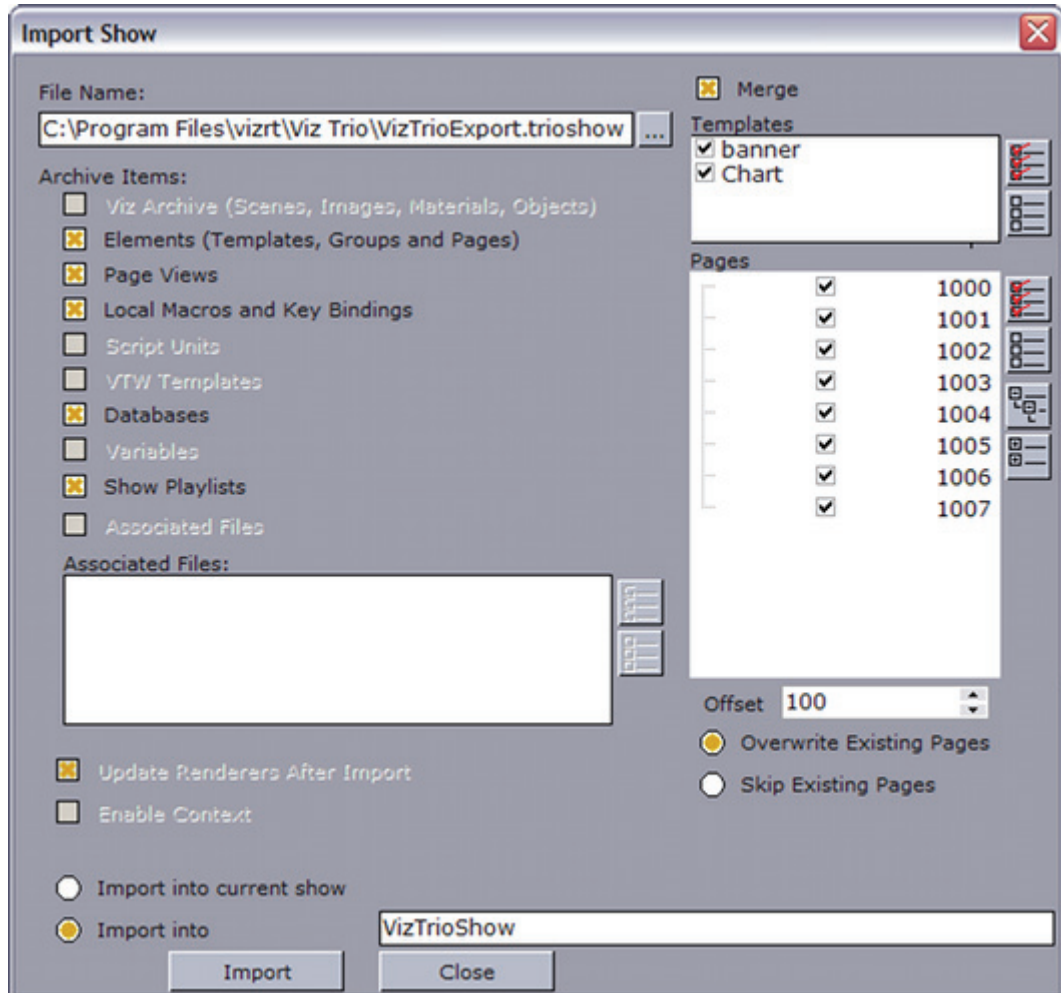
### Import Show



Clicking the Import show button opens the Import Show dialog. As Viz Trio supports two types of shows, context-enabled and non context-enabled shows

there are different options to note when importing a show into a context or non context-enabled show.

**Figure 5:** Import Show dialog box



A context-enabled show, that supports concepts and variants, can import non context-enabled shows; hence, non context-enabled shows can either be merged into the existing context-enabled show, thus converting non context-enabled shows to become context-enabled. Such an import will make sure that all templates belonging to a concept with several variants of the same scene will be displayed correctly in the template list. It is also possible when importing a non context-enabled show to convert a context-enabled show into a non context-enabled show. Such an import will convert all template variants into single templates.

A non context-enabled show, that does not support concepts and variants, can import context-enabled shows; however, this will force the show to become context enabled.

- **File name:** Sets the path to the show to be imported. When clicking the browse button, a browse dialog opens allowing a show file (\*.trioshow) to be selected.
- **Archive Items:** For details, see the [Export Show](#) section.
- **Update Renderers After Import:** Reloads all scenes on the renderers.

- **Enable Context:** Enables the context view, splitting templates and pages in separate views. Shows that are not context-enabled can be converted to use the context view.
- **Import into current show:** Imports templates and pages into the current show.
- **Import into:** Imports templates and pages into a specified, suggested, or new show.
- **Merge:** When checked, this option will merge templates and pages into the existing show.
- **Templates:** Lists all templates in the file.
  - **Select all (button):** Selects all templates.
  - **Deselect all (button):** Deselects all templates.
- **Pages:** Lists all in the file.
  - **Select all (button):** Selects all pages.
  - **Deselect all (button):** Deselects all pages.
  - **Expand all (button):** Expands all nodes.
  - **Collapse all (button):** Collapses all nodes.
- **Offset:** Sets the numeric value used for pages that are imported with offset values. If a page has callup code 1000, and the offset is 100, the new code will be 1100. Note that only pages can be merged with an offset.
- **Overwrite Existing Pages:** If the callup codes already exists in the show, the pages will be overwritten by the imported pages.
- **Skip Existing Pages:** If callup codes already exists in the show, the pages will not be overwritten by the imported pages.

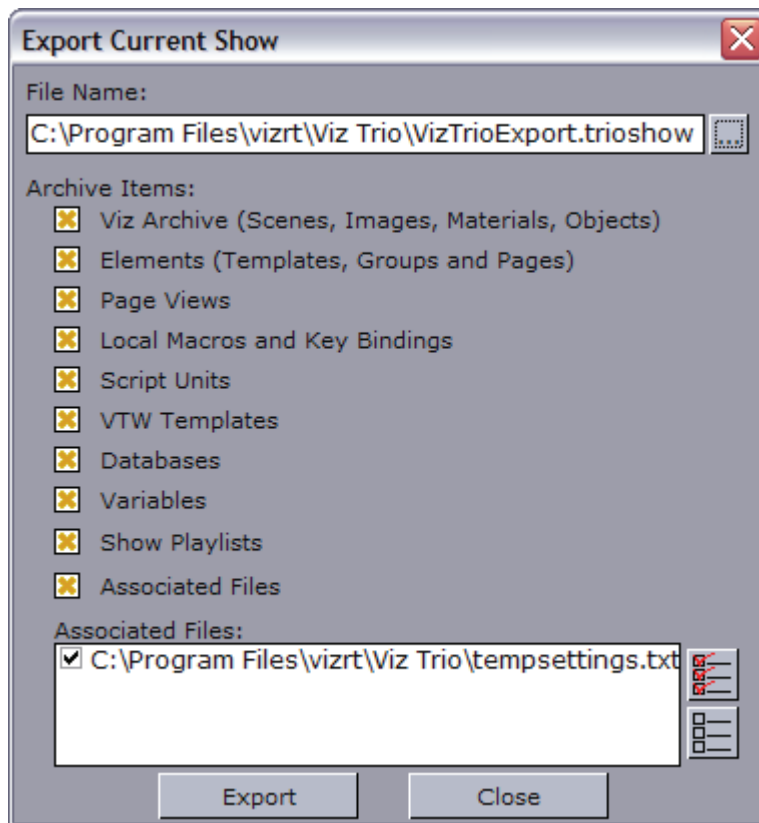
-----  
**Note:** The default import is all elements present in the show archive.  
 -----

## Export Show

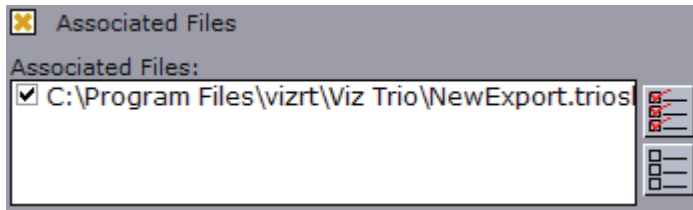


Clicking the Export show button opens the Export Current Show dialog.

Figure 6: Export Current Show dialog box



- **Viz archive (scenes, images, materials, objects):** Viz Engine archives should normally be included in an export. An exception is when the export is done to the same data root, or a data root with the same structure and content as the one exported from.
- **Elements (templates, groups and pages):** Includes all the show's templates, groups and pages.
- **Page views:** Maintains the show's page view organization.
- **Local macros and key bindings:** Includes the show's macro and key shortcut specifications (see the [Show Settings](#) sections).
- **Script units:** Includes the show's script units that are stored on the Media Sequencer (show scripts). File scripts must be added manually to *Associated files* (see the [Show Settings](#) section).
- **VTW Templates:** Includes the show's Viz Template Wizard templates.
- **Databases:** Includes the show's database setups.
- **Variables:** Includes the show's stored variables (for example a counter) and their intermediate information (see the [Show Variables](#) section).
- **Show Playlists:** Includes the show playlists that are created as part of the Viz Trio show, hence, this is not related to Viz Content Pilot and newsroom playlists.



- **Associated Files:** Includes the show's associated files (see the [Show Settings](#) section).
- **Select all (button):** Selects all associated files.
- **Deselect all (button):** Deselects all associated files.

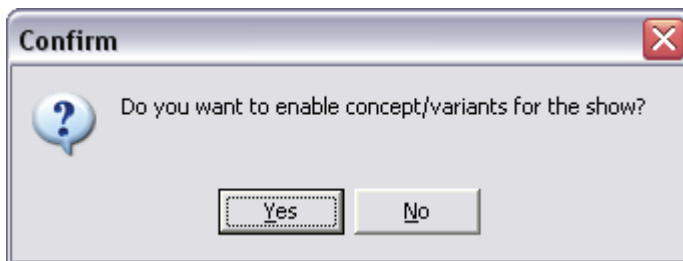
-----  
**Note:** The export file extension is *\*.trioshow*.  
 -----

### To create a show folder

1. Click the Change Show (folder) button to open the *Show Directories* view, and click on the *Shows* tab.
2. Right click the Folders pane, and from the appearing context menu select **Add folder**.

### To create a new show

1. Click the **Create show** button, or right-click the shows pane area and select **Create show** on the appearing context menu.



2. Confirm if the show should be context-enabled or not.
  - If a show is not context-enabled it can be converted later by issuing the Viz Trio command *show:enable\_context*. A show that is context-enabled cannot be converted back to a traditional show.
  - For details on how to use Viz Trio commands, see the [Macro Language](#) section.



3. Enter a name for the new show in the appearing text field.
4. Double-click the new entry in the show pane to open the show.

### To import an exported show

1. From the Viz Trio main user interface, click the Change Show button.
2. In the Show Directories view under the Shows tab, click the [Import Show](#) button to open the **Import Show** dialog.
3. Browse for the show which is saved as a *\*.trioshow* file.
  - When the file is selected the dialog will enable the archived elements.
4. Select the elements of the archive to import.
5. Select if the show should be merged into an existing show.
  - If the show is to be merged, select the templates and pages to import.
  - Set the page offset if pages are not to merged with pages using the same call-up code range.
  - Select whether to overwrite existing pages or to skip existing pages. This has only an effect on pages with the same callup code.
6. Select if the show should be imported into the current show or a new location.
7. Click the Import button to start the import.

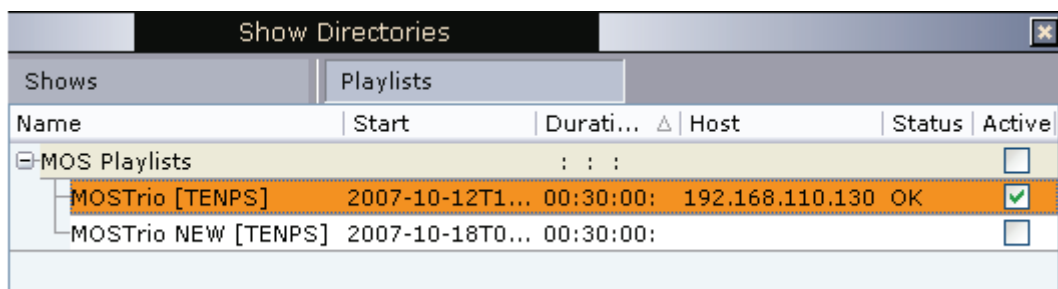
### To export the current show

1. From the Viz Trio main user interface, click the Change Show button.
2. In the Show Directories view under the Shows tab, click the [Export Show](#) button to open the **Export Current Show** dialog.
3. Enter a filename for the archive.
4. *Optional:* Select what to include in the exported archive. By default all options are selected.
5. *Optional:* Browse and select a folder where the archive will be stored. The folder last used is selected as default.
6. Click the Export button to start the export.

-----  
**IMPORTANT!** The show to export must be loaded in Viz Trio.  
-----

### Playlists

The Playlists tab allows the operator to open a Viz Content Pilot (VCP) and/or newsroom playlist. With a VCP or newsroom system integration, Viz Trio can be used to playback VCP and newsroom elements.



Name	Start	Durati... Δ	Host	Status	Active
⊖ MOS Playlists		: : :			<input type="checkbox"/>
MOSTrio [TENPS]	2007-10-12T1...	00:30:00:	192.168.110.130	OK	<input checked="" type="checkbox"/>
MOSTrio NEW [TENPS]	2007-10-18T0...	00:30:00:			<input type="checkbox"/>

- **Name:** Shows the names of the playlists provided by the newsroom control system.
- **Start:** Specifies the start time of the various playlists.
- **Duration:** Shows the duration of each playlist.

- **Host:** Indicates the host of a determined playlist.
- **Channel ID:** Shows the ID of the channel assigned to a playlist.
- **Status:** Indicates a playlist's current state.
- **Active:** Indicates the currently active playlist.

**Note:** Newsroom playlists are by default read only; however, there is a [User Restrictions](#) setting that can be disabled to allow editing of MOS playlists.

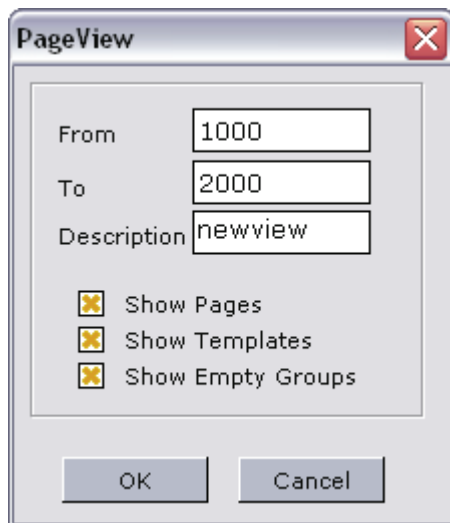
### See Also

- [Playlist Modes](#)
- [Viz Content Pilot Database](#)
- [MOS](#)

## 7.7.2 Add Page List View



Clicking the Page List view button adds a filtered Page List view. There are no limits to the number of Page List views.



When a new view is created, a callup code range must be specified (for example 1000 to 2000). All pages within that range will be displayed in the new view. The original views, such as the template and/or page list, remain unchanged. The additional views are filtered views of the main view.

**Note:** The traditional show view will also list templates in the filtered Page List view.



### 7.7.3 Create Playlist



Clicking the Create Playlist button creates a show playlist that can be filled with Viz Trio pages, Viz Content Pilot and newsroom data elements.

#### To create a show playlist



1. Click the Create Playlist button, and enter a playlist description.
2. Drag and drop pages from the current show to add them to the Show Playlist.

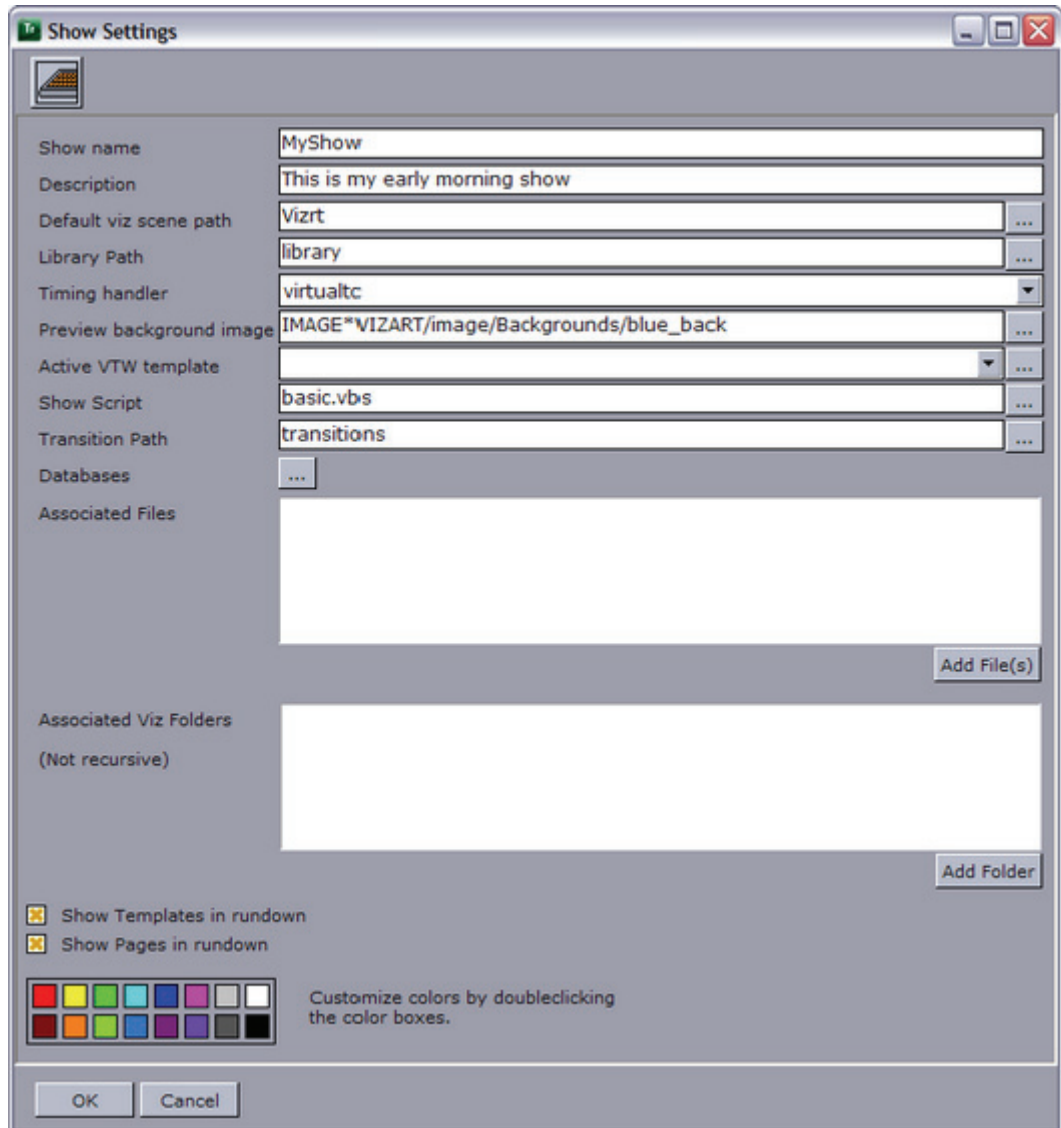
-----  
**CAUTION!** A traditional show that is converted to a context-enabled show cannot be converted back to a traditional show.  
-----

#### See Also

- [Playlist Modes](#)

## 7.7.4 Show Settings

Figure 7: Show Settings dialog box



- **Macros for current show (button):** Opens the Macros for Current Show editor that allows macro commands and scripts to be assigned to keyboard shortcuts for the current show only. Show specific macros or scripts should be saved here. For global commands, see the [Keyboard Shortcuts and Macros](#) section.
- **Show name:** Name of the show.
- **Description:** Description of the show.
- **Default Viz scene path:** Default Viz Engine path for imported scenes.
- **Library Path:** Sets the library path used by the Viz Trio Designer (see [Resources](#)).
- **Timing handler:** When a time code card is installed, a time code handler can be set (see the [Time Codes](#) section).
- **Preview background image:** Sets the image path for the preview background image.
- **Active VTW template:** Sets a Viz Template Wizard template to be used for the current show. Clicking the browse button opens the “VTW Templates: Current

Show” window. From this window templates can be imported, exported or removed. When a template is set for a show it is displayed as part of the Show Control.

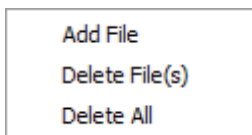
---

**Note:** For information on how to export Viz Content Pilot templates, see the *Viz Content Pilot User's Guide* or *Viz Template Wizard User's Guide*.

---

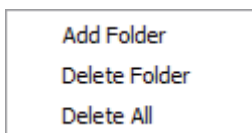
- **Show Script:** Sets a show script that acts as a global script for all templates and pages.
- **Transition Path:** Configures the show’s transition effects’ scene path. Effects are set per page.
- **Databases (ellipse button):** Opens the Database Config editor that allows you [To create a new database connection](#) for linking tab fields to values in a database or file (e.g. [Microsoft Excel and Unicode Support](#)).
  - In early versions of Viz Trio there was only support for database linking of text properties. It was possible to link a value of a single text property to one cell within a spreadsheet. This is called **scalar linking** as it is only possible to hold one value at a time.
  - In more recent versions of Viz Trio table properties were added, and introduced the concept of **table linking**, where the contents of the table property were linked to the contents within a Microsoft Excel spreadsheet. Thus it is possible to link several rows or columns at once.
  - It's important to understand the principle difference between the two as they manifest in different versions of the database link editor (see the [Database Linking](#) section).

**Figure 8:** Associated Files context menu



- **Associated Files:** Displays a list of associated show files, for example video and audio clips to the show. The files are saved with the show when it is exported.
  - **Add File:** Opens a Windows file browser to browse for and add a file to the current Show Settings.
  - **Delete File(s):** Removes the selected file from the current Show Settings.
  - **Delete All:** Removes all associated files from the current Show Settings.

**Figure 9:** Associated Viz Folders context menu



- **Associated Viz Folders (not recursive):** Displays a list of associated show folders. The files are saved with the show when it is exported. When imported it is imported as part of the *Viz archive* option.
  - **Add Folder:** Opens a Viz Engine browser to browse for and add scene, image, geom, material or font folders to the current Show Settings.
  - **Delete Folder:** Removes the selected folder from the current Show Settings.
  - **Delete All:** Removes all associated folders from the current Show Settings.

- **Show Templates in rundown:** Enable this option to display templates in a page list (not used for context-enabled shows).
- **Show Pages in rundown:** Enable this option to display pages in a page list (not used for context-enabled shows).
- **Show specific character colors:** Double-click on the color boxes to customize show specific character colors (see also how [To edit the text color](#)).

This section contains information on the following topics:

- [To open show settings](#)
- [To set the transition effects path](#)
- [To create a new database connection](#)
- [To edit a connection](#)
- [To remove a connection](#)

### To open show settings



- Click the [Show Settings](#) button.

### To set the transition effects path

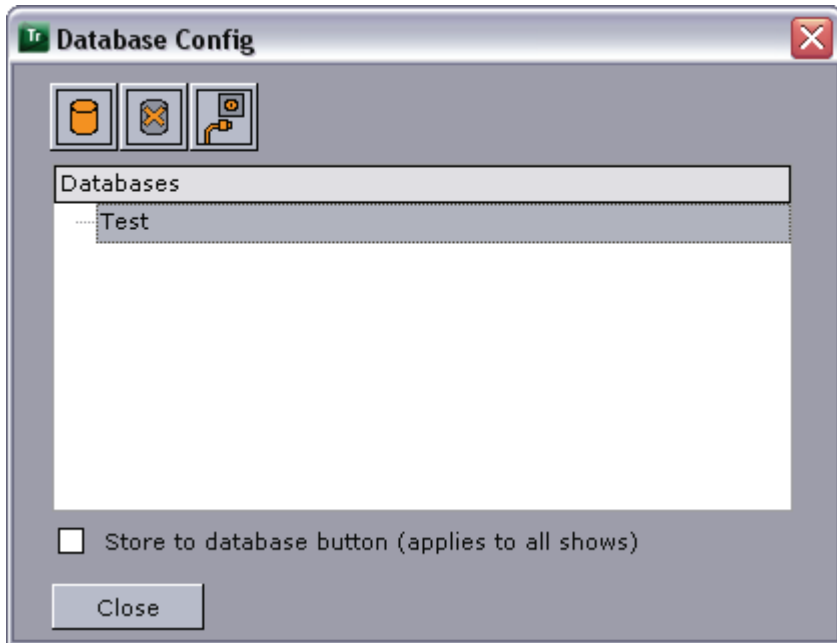
1. Click the **Show Settings** button to open the [Show Settings](#) pane.
2. Click the **ellipse (...)** button next to the **Transition Path**.
3. Search Viz for the **transitions** effects folder, and click **OK**.
4. Click **OK** to close the Show Settings window.

-----  
**Note:** The effects folder should ideally be placed according to the Viz Content Pilot preferences which is at the root of the Viz database.  
 -----

### To create a new database connection

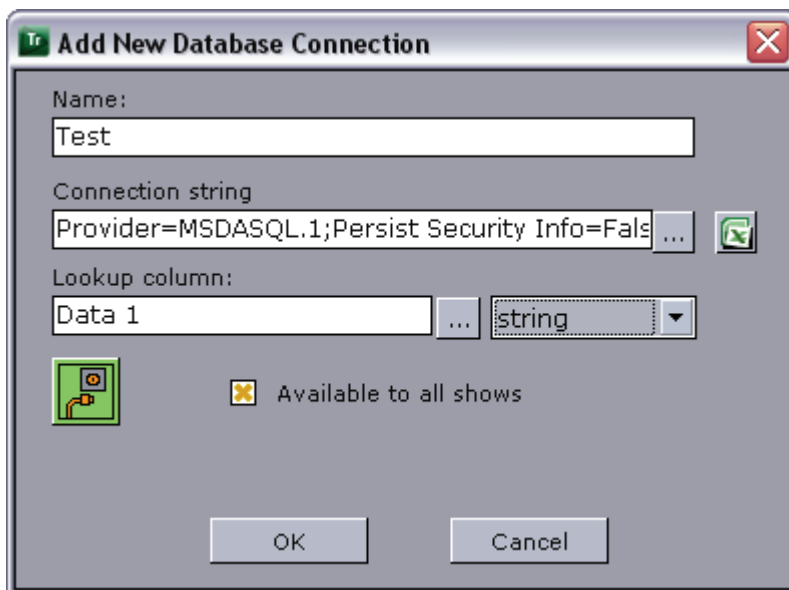
1. Click the Databases settings' ellipse (...) button in the **Show Settings** section to open the [Database Config](#).

Figure 10: Database Config



2. Click **Add database** to open the [Add New Database Connection window](#).

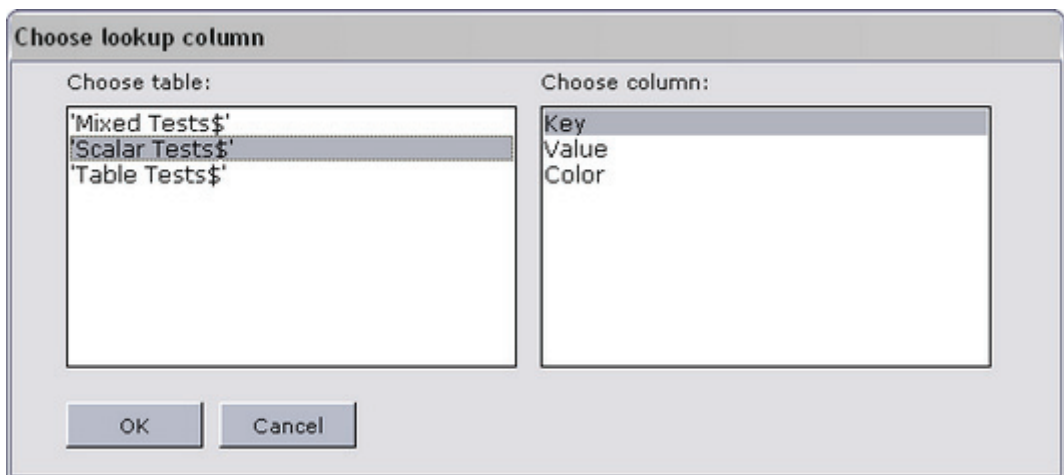
Figure 11: Add New Database Connection window



3. Enter a **Name** for the connection and select one of the following options:
4. *Option 1: OLE DB*
  - a. Click the **Connection string**'s ellipse (...) button to configure the Data Link Properties
  - b. Select an **OLE DB** provider, and click **Next**.
  - c. Add the connection parameters.
  - d. Click the **Test Connection** button.
5. *Option 2: Microsoft Excel*

- Click the **Microsoft Excel** button to use a spreadsheet. Note that configuring an Excel spreadsheet will automatically set the correct provider.
6. Click the **Test connection** button to check that the connection was successful.
  7. Click **OK** save the connection string.
  8. Click the **Lookup column's** ellipse (...) button to **choose lookup column**.

**Note:** When clicking the Lookup column's ellipse (...) button, and the connection parameters are correct, the connection button will turn green.



9. *Optional:* Select if the lookup column type should be string or integer.

**Note:** Specifying the lookup column is only necessary in scalar linking cases.

**Note:** A Microsoft Excel spreadsheet column is used to match a key value in order to select the scalar value for the text property. For table linking it can be left blank.

10. Specify if the database should be Available to all shows.
  - **Available to all shows:** Shows/pages that are exported and imported into other shows will also import the Excel file and database link settings.

11. Click **OK** to save the database connection.

#### To edit a connection

- Double-click the database connection to open the **Edit Database Connection <name>** window.

#### To remove a connection



- Click the **Delete Database** button in the configuration dialog.

### See Also

- [Page Editor section on Database Linking](#)
- [Keyboard Shortcuts and Macros](#)
- [Macro Language](#)
- [Macro Commands and Events](#)
- [Scripting](#)

## 7.7.5 Cleanup Channels



This function clears all loaded graphics from memory on the program and preview renderers for the output profile currently in use. It should be used before initializing a new show or in order to re-initialize the same show into the renderer's memory.

---

**Note:** Cleanup commands will affect all Viz Trio clients that are connected to the same Media Sequencer, and using the same output profile.

---

### See Also

- [Initialize Channels](#)
- [Macro Commands](#)
  - `trio:cleanup_channel`
  - `trio:cleanup_renderers`

## 7.7.6 Initialize Channels



The Initialize button loads the current show's graphics on the preview and program renderers.

---

**Note:** In case of transition logic scenes, the state of background scenes will be reset.

---

Initialize does not refresh everything (it performs a load, not a reload on the Viz Engine). If changes have been made to a scene that was already loaded, a Cleanup renderer command must be issued, and thereafter an Initialize command.

When initializing a show the Initialize button in the user interface changes color to yellow to indicate that the initialization process is running. When it is finished loading scenes to the renderer the button changes to a green color.

Several shows can be initialized if necessary. After having initialized the first, another show can be switched to and initialized. The graphics for the first one will still be there, ready for use.

When initializing shows that require a lot of memory, please take the memory use on the program and preview renderers into consideration when loading the

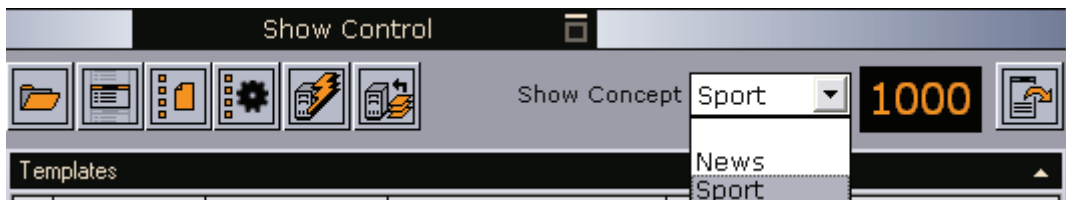
graphics. Too much graphics on the renderer(s) will use up all physical memory, causing the performance to drop below real-time, which again may cause the renderer(s) to become unstable.

**Note:** Initialize commands will affect all Viz Trio clients that are connected to the same Media Sequencer, and using the same output profile.

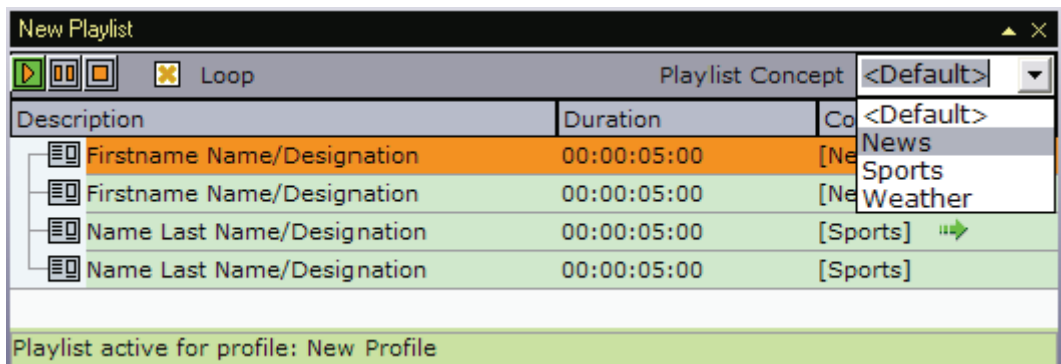
### See Also

- [Cleanup Channels](#)
- [Macro Commands](#)
  - [trio:initialize\\_show\\_or\\_playlist](#)

## 7.7.7 Show Concept



A concept can be switched directly from the [Show Modes](#). It is also possible to switch concepts using context variables in a macro or a script.

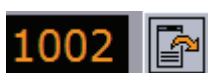


Switching concepts for a show's playlist is done in the same way as for a show; however, they can also be switched independently. Concept **<Default>** refers to the concept the page was saved with.

### See Also

- [Show, Context and Tab-field Commands](#)

## 7.7.8 Callup Page



Part of the show control is a field and a button used for selecting and reading specific pages. When the pages are read, they are also opened in the [Page Editor](#).




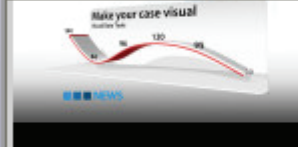

### To read a page

- Enter the page's **callup code** (number) using the keyboard's **numeric pad**, and press the **plus (+)** key to read the page.

---

## 7.8 Template List

Figure 12: Depicts the Template List, and the template variants.

Icon	Name	Variant	Description	Layer(s)
	banner	Lower	Myriad-Pro-Semibold/Designer	[MAIN]
	Line		Visual Data Tools/Make your c...	[MAIN]
	Pie		Bergen/GEOM*#54623/37°C/...	[MAIN]

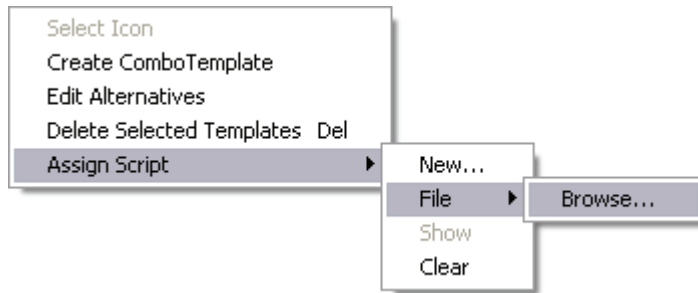
The Template List, unlike the traditional Page List, only contains templates. A separate Page List contains the pages derived from the templates.

Template variants can be switched directly from the [Template List](#) and [Page List](#), or from the [Page Editor](#). As with concepts, variants can also be switched using a macro command in a user-defined macro or a script.

This section contains information on the following topics:

- [Context Menu](#)
- [Columns](#)
- [Combination Template](#)
- [To select a variant from the template list](#)

## Context Menu



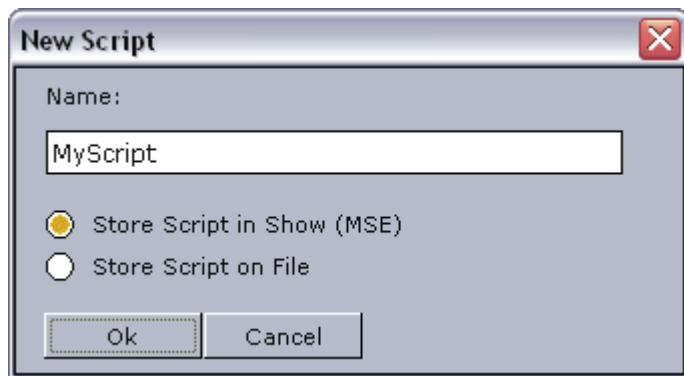
The Template List has its own context menu for creating combinations of templates, assigning scripts and so on.

- **Select Icon:** Use this option to select which icon to show for the template.
- **Create Combo Template:** Creates a combination template that contains all selected templates. This is only relevant for transition logic templates. All templates that are to be merged into a [Combination Template](#) must be in different transition logic layers

Layer	I...	vari...	concept	platform	Lower	Scene Path
[-] banner						
		Lower	News	HD	Top	Vizrt_Internal/TradeShows/VizMPS/MyChannel/con...
		Top	News	HD	Top	Vizrt_Internal/TradeShows/VizMPS/MyChannel/con...
		Lower	News	MPS_W...	Top	Vizrt_Internal/TradeShows/VizMPS/MyChannel/con...
		Lower	News	SD	Top	Vizrt_Internal/TradeShows/VizMPS/MyChannel/con...
		Top	News	SD	Top	Vizrt_Internal/TradeShows/VizMPS/MyChannel/con...
		Lower	Sport	HD	Top	Vizrt_Internal/TradeShows/VizMPS/MyChannel/con...
		Top	Sport	HD	Top	Vizrt_Internal/TradeShows/VizMPS/MyChannel/con...
		Lower	Sport	SD	Top	Vizrt_Internal/TradeShows/VizMPS/MyChannel/con...
		Top	Sport	SD	Top	Vizrt_Internal/TradeShows/VizMPS/MyChannel/con...

Ok Cancel

- **Edit Alternatives:** Opens a dialog for viewing scene information, such as concept, variant, user-defined contexts (platform) and scene path, associated with a template.
- **Delete Selected Templates:** Deletes a template, or templates, from the Template List.
- **Assign Script:** Assigns a script to a template. It is not possible to assign scripts to pages. A page references the template script.



- **New:** Creates a new blank script. The script can be saved to a local or shared repository, or to the Media Sequencer.
- **File:** Clicking the Browse ... button opens a file browser on the local computer.
- **Show:** Selects scripts placed on the Media Sequencer.
- **Clear:** Clears the assigned script.

---

**Note:** To edit the show script, see the [Scripting](#) section.

---

## Columns

Description
Icon
Layer(s)
Name
Variant
Auto Width
Clear Sorting

- **Description:** Shows the contents of the template tab fields.
- **Icon:** Shows the thumbnail image of the scene.
- **Layer(s):** Shows which layer, [FRONT], [MAIN], [BACK], the scene belongs to.
- **Name:** Shows the name of the template.
- **Variant:** Shows the variants for the template. For example lower, top and full screen.
- **Auto Width:** Adjusts the column width automatically.
- **Clear Sorting:** Clears any sorting performed by the user when clicking the column headers.

## Combination Template

A combination template can only be created using transition logic templates. In order to create a combination template the templates must be in different layers (see the template list's [Columns](#) or the page list's [Columns](#)).

**Figure 13:** Combination template with layers and variants



It is also possible to create combination templates of transition logic scenes with concepts and variants in context-enabled shows. However, there is a small difference in editing the two. A transition logic combo template with concepts and variants will expose a drop list named Layer. This will enable the operator to select the variant's correct layer, for example a scene with a *lower third* and an *over the shoulder* layer. After selecting the layer, the operator can select the correct variant, for example a lower third variant with *one* or *two* lines.

#### To select a variant from the template list

- Right-click the Icon column and select the variant from the appearing drop-list.

#### See Also

- Template list [Context Menu and Columns](#)
- Page list [Context Menu and Columns](#)

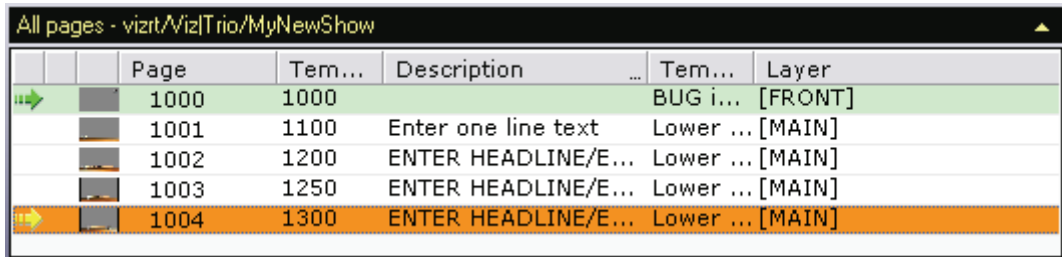
## 7.9 Page List

**Figure 14:** Non context-enabled page list

All pages - vizrt/VizTrio/MyShow						
		Page	Tem...	Description	...	Tem... Layer
		1000	1000			BUG i... [FRONT]
		1100	1100	Enter one line text		Lower ... [MAIN]
		1200	1200	ENTER HEADLINE/E...		Lower ... [MAIN]
		1250	1250	ENTER HEADLINE/E...		Lower ... [MAIN]
		1300	1300	ENTER HEADLINE/E...		Lower ... [MAIN]
→		1001	1000			BUG i... [FRONT]
		1002	1100	Enter one line text		Lower ... [MAIN]
		1003	1200	ENTER HEADLINE/E...		Lower ... [MAIN]
		1004	1250	ENTER HEADLINE/E...		Lower ... [MAIN]
→		1005	1300	ENTER HEADLINE/E...		Lower ... [MAIN]

The traditional Page List shows the list of accessible templates and pages. They are all identified by a number and a description. The description is by default the values of the template or page's items and tab fields, separated by a slash.

**Figure 15:** Context-enabled page list



	Page	Tem...	Description	...	Tem...	Layer
➡	1000	1000			BUG i...	[FRONT]
	1001	1100	Enter one line text		Lower ...	[MAIN]
	1002	1200	ENTER HEADLINE/E...		Lower ...	[MAIN]
	1003	1250	ENTER HEADLINE/E...		Lower ...	[MAIN]
➡	1004	1300	ENTER HEADLINE/E...		Lower ...	[MAIN]

A [Context-enabled page list](#) contains pages and no templates. However, all features are the same, except the ones that are specific to templates which are now placed in the [Template List](#)'s context menu.

Template variants can be switched directly from the [Template List](#) and [Page List](#), or from the [Page Editor](#). As with concepts, variants can also be switched using a macro command in a user-defined macro or a script.

A page can be played out from a page list or from a show playlist. The page list can play its pages one by one. A show playlist has the option of creating a carousel, playing the items like a scheduled playlist. It can also be looped.

Pages are edited using the [Page Editor Controls](#), and its available editors. Most page editors are made available to the operator by the scene designer through exposed scene properties within the scene. Others, like a database connection, are made available through the [Show Settings](#) window or by using more advanced features such as [Macro Commands and Events](#).

This section contains information on the following topics:

- [Context Menu](#)
- [Columns](#)
- [Change Template](#)
- [Create Groups](#)
- [Colors](#)
- [Channel Not Found](#)
- [Manually Sorting Pages](#)
- [Time Codes](#)
- [To create a page](#)
- [To read a page](#)
- [To edit a page](#)
- [To take pages on-air](#)
- [To delete a page](#)
- [To save to XML ...](#)
- [To load from XML...](#)
- [To change a template](#)
- [To change a template using a macro command](#)
- [To select a variant from the page list](#)

## Context Menu

A right-click on any element in the page list activates the context menu. Select a menu item to perform an action or define a setting for a template or a page. The traditional Page List contains both templates and pages. Thus, some settings in the traditional Page List can only be made to templates and others only to pages. This is explained in the list below.

---

**Note:** Clicking an image icon will, for context-enabled shows, open a drop-list of available template variants.

---

**Figure 16:** Traditional page list (left), new page list (right)

Change Filter...		Change Filter...	
Sort pages		Sort pages	
Edit	▶	Edit	▶
Read		Read	
Direct Take		Direct Take	
Direct Take Out		Direct Take Out	
Direct Continue		Direct Continue	
Initialize		Initialize	
Delete page	Del	Delete page	Del
DeleteAll		DeleteAll	
Print/Save snapshots		Print/Save snapshots	
Load from xml...		Load from xml...	
Save to xml...		Save to xml...	
Cut Pages	Ctrl+X	Cut Pages	Ctrl+X
Copy Pages	Ctrl+C	Copy Pages	Ctrl+C
Paste Pages	Ctrl+V	Paste Pages	Ctrl+V
Move to number		Move to number	
Move with offset		Move with offset	
Copy to number		Copy to number	
Copy with offset		Copy with offset	
Create Combo Page			
Reimport		Reimport	
Rename...		Rename...	
Change template...		Change template...	
Create Group		Create Group	
Find	Ctrl+F	Find	Ctrl+F
Set status	▶	Set status	▶
✓ Autogenerate description		✓ Autogenerate description	
Script	▶		
Rundown Settings	▶	Rundown Settings	▶

**Note:** The menu options Create Combo Page and Script are removed. This is now a part of the [Template List](#)'s context menu.

Template combinations can only be made using templates, and scripts can only be assigned to templates. The latter is due to the fact that pages reference scripts, just as a page reference a template. Hence, pages cannot be assigned a script or create combinations of pages. Template scripts can be edited by opening a page and clicking the Edit script button in the Page Editor. However, this will change the behavior of all pages that reference the same script.

- **Change Filter ...:** Changes the Page List filter. This menu option is enabled when a Page List filter is created, see the [Add Page List View](#) section.

- **Sort pages:** When enabled, the page list will be sorted by page numbers. When disabled, it is possible to drag and drop pages freely to create the other page orders.
- **Edit:** Opens a submenu for page (not template) description editing.
- **Read:** Performs a read on the selected template or page.
- **Direct Take:** Performs a direct take on the selected page.
- **Direct Take Out:** Takes out the selected page, or any that is loaded in the same transition logic layer as the selected page.
- **Direct Continue:** Continues the animation on the selected page, or any that is loaded in the same transition logic layer as the selected page.
- **Initialize:** Loads the selected page(s) on the program and preview renderer.
- **Delete page:** Deletes the selected page(s).
- **Delete All:** Deletes all pages in the page list.
- **Print/Save snapshots:** Opens a dialog box where snapshots can be taken and either being saved to file or sent to a printer.
- **Load from XML:** Imports a page list from an XML file (see [To save to XML ...](#)).
- **Save to XML:** Saves the selected pages to an XML file (see [To save to XML ...](#)).
- **Cut Pages (CTRL+X):** Cuts the given page(s) to the clipboard.
- **Copy Pages (CTRL+C):** Copies the given page(s) to the clipboard.
- **Paste Pages (CTRL+V):** Pastes the given page(s) from the clipboard to the current show.

---

**Note:** Pages in context-enabled shows cannot be copied to non context-enabled shows, and conversely.

---

- **Move to number:** Moves a page, or pages, to a given callup number. For example pages 1001 and 1002 that are moved to number 2000 will restart the numbering of the pages and give them the callup codes 2000 and 2001.
- **Move with offset:** Moves a page, or pages, with a given offset. For example pages 1001 and 1002 with offset 100, will move pages and give them the new callup codes 1101 and 1102.
- **Copy to number:** Copies a page, or pages, to a given callup number. For example pages 1001 and 1002 that are copied to number 4000 will restart the numbering and give the copied pages the callup codes 4000 and 4001.
- **Copy with offset:** Copies a page, or pages, with a given offset. For example pages 1001 and 1002 with offset 10, will create page copies with callup codes 1011 and 1012.
- **Create Combo Page:** Creates a combination template that contains all selected transition logic templates of foreground scenes. This menu option is only enabled for transition logic templates. All templates that are to be merged into a combination template must be in different transition logic layers (see the [Template List](#) section for the Create Combo Template menu option).
- **Reimport:** Re-imports the selected page.
- **Rename:** Sets a new name for the page.
- **Change template:** Opens the [Change Template](#) dialog box.
- **Set external cursor:** Sets the global cursor to the selected template or page. This setting is related to GPI or Automation setups.
- **Clear external cursor:** Clears the current global cursor.
- **Create Group:** Creates a group node in the page list.

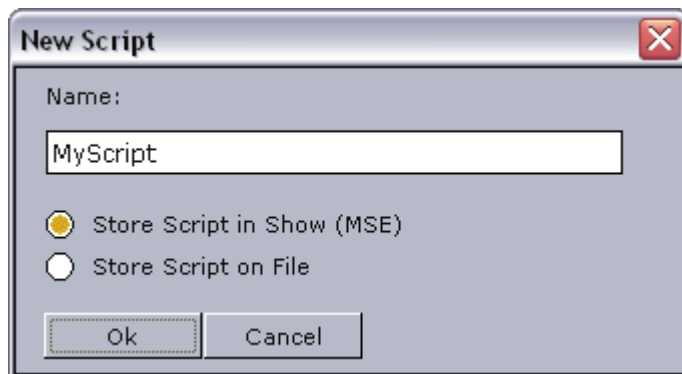


---

**Note:** The external cursor option must be enabled in the Configurations window, see the [Page List](#) section.

---

- **Collapse selected node:** Collapses the selected node of a group.
- **Expand selected node:** Expands the selected node of a group.
- **Collapse All:** Collapses all nodes of all groups.
- **Expand All:** Expands all nodes of all groups.
- **Move Pages To Group:** Allows one or more pages to be moved to groups within the show. Available groups are listed in the appearing submenu. Selecting the Root option will move the selected page(s) outside of the group.
- **Find (CTRL+F):** Opens a text search editor. Perform searches in the page list for a certain word or expression. The page ID, description, template description and layer name columns are searchable. Hidden columns are not searchable.
- **Set status:** Tags pages with a status icon to indicate that the page is finished and ready or the opposite if it is unfinished.
- **Autogenerate description:** For pages (not for templates) with this option disabled, it is possible to enter a custom description for these pages in the list. When enabled, the description for a page will consist of the contents of its tab fields separated by dashes.
- **Script:** Displays a submenu with two options, Assign script and Edit Show Script.
  - **Assign Script:** Assigns a script to a template. It is not possible to assign scripts to a page. A page references the template's script that the page derives from.



- **New:** Creates a new blank script. The script can be saved to a local or shared repository, or to the Media Sequencer.
- **File:** Clicking the Browse ... button opens a file browser on the local computer.
- **Show:** Selects scripts placed on the Media Sequencer.
- **Clear:** Clears an assigned script.
- **Edit Show Script:** This option allows an assigned show script to be edited. For further details on scripts, see the [Scripting](#) section.
- **Rundown settings:** Allows a user to lock/unlock the column widths and set the image size for the page thumbnails
  - **Show Pages:** Displays the pages in the page list.
  - **Show Templates:** Displays the scene templates in the page list.
  - **Lock column widths:** Locks the column widths.

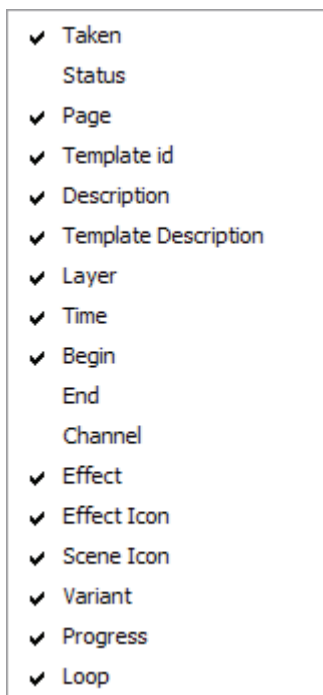
- **Image Size:** Sets the image icon size in the page list to Small (default), Medium, Large or Extra Large.

**Note:** It is not possible to browse for scripts on the Media Sequencer from the template or page list. Use the Show Script browse button in Show Settings, or the *Manage Show Scripts* button in the Page Editor.

## Columns

Select the columns that should be visible in the page list. Right-click one of the existing columns, and a context menu appears:

**Figure 17:** Page List columns menu.



Choose the columns to add or remove.

- **Taken:** Shows the arrow that depicts the status of the page. A yellow arrow (→) depicts the last previewed page, a green arrow (→) depicts the last taken page, and a red arrow (→) depicts a page taken on-air using an external cursor (GPI).
- **Status:** Shows the status of the selected page. Page status indicators are Finished (✓), Unfinished (□) and None. See the macro command `page:set_pagestatus` in the [Macro Commands and Events](#) section.
- **Page:** Shows the page ID that usually is a numeric callup code.
- **Template id:** A page reference a template, thus showing template ID. The ID is usually a numeric callup code.

**Note:** Template and Page IDs are usually numeric. However, Alphanumeric IDs can be allowed. See the [General](#) section and enable *Allow Alphanumeric Imports*.

- **Description:** Shows by default the text entered in the editable tab fields. Only a page's Description field can be changed.

- **Template Description:** Shows the template's description. Usually information about the kind of template it is. For example lower third, bug and full screen. Descriptions are predefined by the scene designer.
- **Layer:** Shows the layer information. This can be [FRONT], [MAIN] and [BACK].
- **Begin:** *Not supported by Viz Trio.*
- **End:** *Not supported by Viz Trio.*
- **Channel:** Selects the playout channel the graphics will be rendered on.
- **Effect:** Opens the Choose Effect dialog when the clicking the browse button.
- **Effect Icon:** Shows a thumbnail image of the transition effect.

---

**Note:** Transition effects cannot be used for transition logic scenes.

---

- **Scene Icon:** Shows a thumbnail image of the graphics. See the [Page List](#) section for more information on adjusting rundown and image size settings.
- **Variant:** Shows the template variant text. To view a list of variants with thumbnails, left-click the text field.
- **Progress:** Displays the progress of video transfer to video playout engines such as Viz Engine and Quantel.
- **Loop:** Loop enables a loop mode for a group of scenes or a full screen video clip on the video channel. Note that a full screen video clip will play once by default; however, video clips in graphics loop by default. A complete playlist (see control bar), group or video clip may be looped; however, it is of course not recommended to set more than one of these modes to loop at once.

## Change Template

For standalone scene templates it is possible [To change a template](#) a page is using without having to retype its content. This will work under the following conditions:

- The tab fields in the original, and the new template, must be of the same type. If tab field 1 is of type text in the original template, the changed template must have text in tab field 1 too. The same is the case for images.
- Be aware of restrictions or differences between the two templates such as character limits. A property exposed on the source template might not be exposed in the target. As a general rule the templates should be quite similar in their logical structure.
- The total number of tab-fields should be the same.
  - If the source template for instance has three bullet points (exposed as three tab-fields) and the target has four, the first three will be filled with existing values and the fourth will have a default value.
  - If the source template has four bullet points and the target has only three, the last bullet will not be shown. If no new text have been saved or added, it is possible to change back and the original template will be restored.

See also how [To change a template using a macro command](#).

## Create Groups

It is possible to create groups in a page list to organize pages. To create a group, right click in the page list and choose "Create group". A new group appears with the default name "new\_group". Click on the name once to edit, or right click and choose edit/name.

To add pages to the group, drag and drop elements onto the right side of the group element. A page added to a group is added as a child node of the group when it is dropped onto the group. To move pages out of a group, drag and drop it outside the group, or use the context menu option Move Pages To Group, and select Root.

**Note:** Pages cannot act as groups for other pages.

An alternative option is to use the Move Pages To Group option available on the Page List context menu. Select one or more pages in the page list, then press the right mouse button and select the Move Pages To Group option from the appearing menu.

**Note:** Stories imported from an Avid Newstar newsroom system will automatically appear as groups (one group per story). This is similar to a MOS playlist, and is built using the traditional show type. Hence, it has no support for context switching (concept and variants).

## Colors

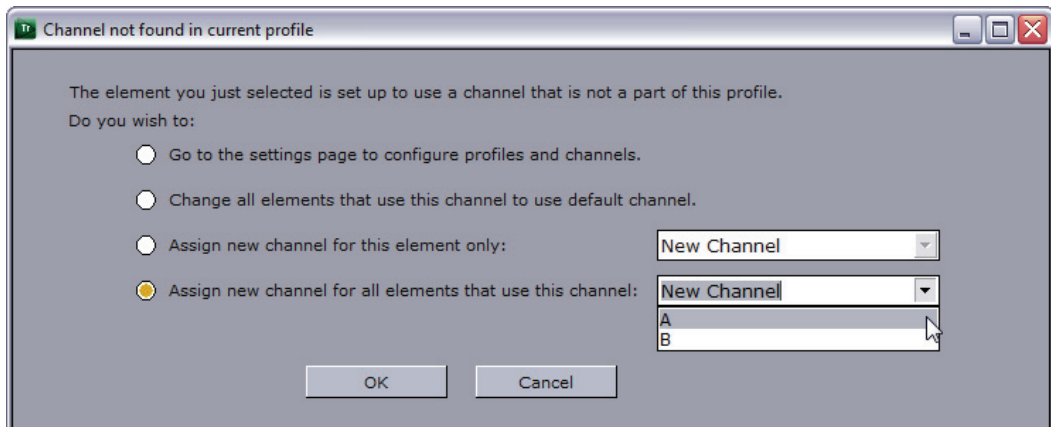
All pages - Vizrt/Tutorial/TrainingShow				
Page	Templat...	Description	Template...	▲
0006	0006	vizrt/all the time/IMAGE*00Generic/GB	Clock	
0007	0007	vizrt/The complete picture	Opener vizrt	
0017	0017	Styrefinner/GPS-enhet/Stridshode	JDAM Bom...	
1200	1200	ENTER HEADLINE/Enter subtext on line one ...	Lower 3rd...	
2101	2100	David Beckham/Football Idol/Age: 29/Club: ...	Fullscreen...	
0020	0019	183 km/183 km	Map of Scud	
0021	0019	183 km/183 km	Map of Scud	
Group				
0023	0019	183 km/183 km	Map of Scud	
0024	0019	183 km/183 km	Map of Scud	
9000	9000	Scroll created with the old scroll function		
5003	5002			

The items in the page list can have eight different colors:

- Brown indicates a template.
- Orange indicates the currently selected page.
- White indicates a normal page.
- Red indicates a page whose template is deleted.
- Blue indicates a scroller page created with the old scroll function.
- Beige indicates a group.
- Dark green indicates an initialized template.
- Light green indicates an initialized page.

The page list [Colors](#) can be customized as needed.

## Channel Not Found



The channel not found dialog box opens if a channel has been set for a page (or template) that does not exist in the current profile. This occurs if a page has been configured to use a specific channel (e.g. C) and that channel has for example been deleted.

The easiest way to avoid this scenario it to always use the default assignment [PROGRAM] as this refers to the status of the channel, and not the channel name.

## Manually Sorting Pages

Disable sorting in the page list to move pages up and down by drag and drop. This possibility can for example be used for creating a sequenced playlist.

## Time Codes

A time code reader is a handler that reads time-codes from some source, and prepares and executes elements by comparing the received time-codes with the begin time-codes specified on the elements. How the time-code reader receives the time-codes will vary between different time-code readers, but in most cases it will involve specific hardware for decoding time-code signals embedded in video signals.

Viz Trio supports the use of time code reader hardware. Currently a card from *Alpermann+Velte Electronic Engineering GmbH*, named "PCL PCI D" is supported.

Contact Vizrt for instructions on how to configure Viz Trio to work with the card. The card can be used as a timing handler for a show. The card enables start times on the pages to be set. The time code from the video system will trigger the pages on-air.

## To read a page

1. Select a page from the [Page List](#).
2. Double-click the page in the page list, or
3. Right-click the Page and from the appearing context menu select **Read**, or
4. Press the Read key on the [Cherry Keyboard](#).

---

**Note:** When “reading” the page it will be displayed in the preview window and opened for editing in the [Page Editor](#).

---

#### **To take pages on-air**

1. Select a page from the [Page List](#).
2. **Read** the page.
3. Click the **OnAir** button in the upper right corner to set Viz Trio in on-air mode.
4. Click the **Take** button or the Take key on the keyboard.
5. Click the **Continue** button to continue the animation if the scene has stop-points.
6. Click the **Take Out** button or Take Out key on the keyboard to take the page off air.

#### **To delete a page**

1. Select and right-click a page from the [Page List](#), and from the appearing context menu select **Delete page**, or
2. Press the **Delete** key on the keyboard.

---

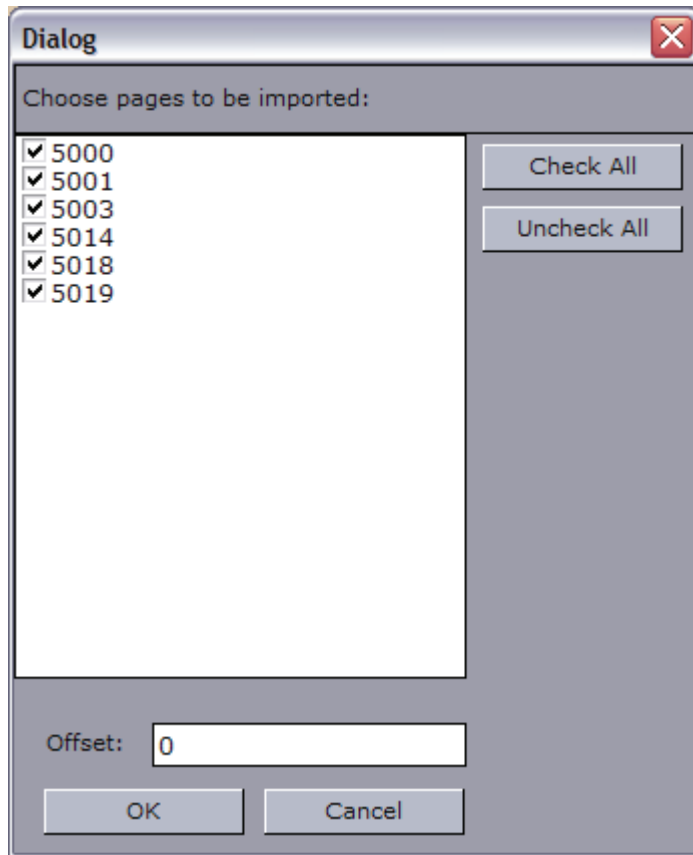
**Note:** It is also possible to multi-select pages by holding the SHIFT or CTRL button and selecting the range or individual pages to be deleted.

---

#### **To save to XML ...**

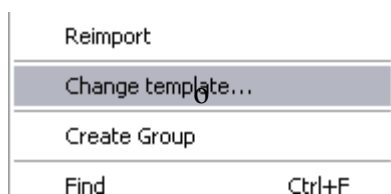
1. Right-click the page list, and from the appearing [Context Menu](#) select the **Save to XML...** option.
2. Specify the file name and location in the appearing file dialog.
3. Click **Save**.

### To load from XML...



1. Right-click the page list, and from the appearing **Context Menu** select the **Load from XML...** option.
2. Choose the pages to be imported from the list.
  - Click **Check All** to select all pages.
  - Click **Uncheck All** to deselect all pages.
3. *Optional:* Set an **Offset** for the imported page numbers such that the imports do not conflict with existing page numbers.

### To change a template



1. Select the pages to change a template for (use CTRL or SHIFT to multi-select).
2. Right-click the page(s), and from the appearing context menu select **Change template...**
3. Enter the template name/number for the template to change to.
4. Click **OK**.

-----  
**Note:** The name of the template is case sensitive.  
-----

#### To change a template using a macro command

1. Assign the macro command *page:change\_template(string templateName)* to a keyboard shortcut.
2. Press the keyboard shortcut assigned to the *change\_template* command to perform the change.

-----  
**Note:** The name of the template is case sensitive.  
-----

3. Click **OK**.

#### To select a variant from the page list

- Right-click the Scene Icon column and select the variant from the appearing drop-list.

#### To add transition effects using the page list

1. Right-click a show's page list column header, and from the appearing context menu select **Effect** and **Effect Icon** (last is only for visual reference)
  - This will display the Effect and Effect Icon columns
2. Click the **ellipse** button in the **Effect** column and select, from the appearing dialog box select a transition effect

#### See Also

- [Creating Transition Effects](#)
- [Keyboard Shortcuts and Macros](#)
- [Show Settings](#)
- [Transition Effects](#)

---

## 7.10 Playlist Modes

There are three playlist modes available in Viz Trio:

- Viz Trio show playlists are created using the [Create Playlist](#) button and exists only as part of the show.
- Viz Content Pilot [Playlists](#) are requested by the Viz Trio operator and modified when integrated with a Viz Content Pilot system. Note that a VCP playlist element cannot be edited, only the structure of the playlist. The VCP playlist elements can be added to a Viz Trio show playlist, but not a page list.
- Newsroom [Playlists](#) are requested by the Viz Trio operator, but not modified by default (see [General](#) user interface settings). Edits to any element in the newsroom playlist will by default be discarded as soon as it is updated by the newsroom system. Viz Trio is therefore in most cases, when monitoring newsroom playlists, only used to read and play out its elements.



For the playout of newsroom playlists to work, the Media Sequencer must establish a connection to Viz Gateway (see [MOS](#) integration).

All shows and playlists are stored on the Media Sequencer for playout.

---

**Note:** Viz Trio 2.10 comes with a [New Playlist](#); however, the [Old Playlist](#) can still be used by disabling the “Use the new playlist by default” option (see [General](#) settings).

---

All Show Playlists can be monitored, stopped and started from the [Active Tasks](#) window.

This section contains information on the following topics:

- [New Playlist](#)
- [Old Playlist](#)
- [To activate or deactivate a playlist](#)
- [To take pages on-air from a show playlist](#)
- [To show or hide columns in the playlist](#)

#### To activate or deactivate a playlist



1. [Create Playlist](#) or open [Playlists](#)
2. Right-click the bar at the bottom of the playlist, and from the appearing context menu select Activate or Deactivate to activate or deactivate the playlist.
  - Activating a playlist will initialize the elements and start the transfer of video clips to Viz Engine.

#### To take pages on-air from a show playlist

1. Add pages to the playlist.
  - *Optional:* Enable looping.
  - *Optional:* Set the concept for the playlist or for each element.
  - *Optional:* Set the variant for each element.
  - *Optional:* Set the duration for each element.
2. Click the **Play** button to start the carousel.
3. Click the **Pause** or **Stop** button to pause or stop the playlist.

---

**IMPORTANT!** Scenes with stop-points and out-animations are being cut as they are **not** automatically continued when used in a playlist.

---

#### To show or hide columns in the playlist

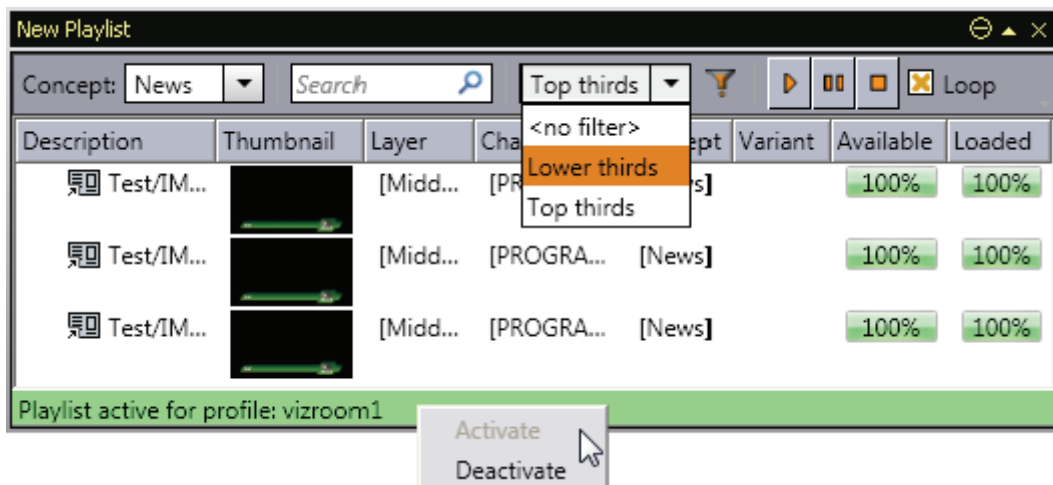
- Right-click the playlist’s header and from the appearing menu select the columns to display or hide.

- The *Auto width* option enables automatic resize of columns within the MOS Playlist area.
- Disable the *Auto width* option to manually resize the columns.

### See Also

- [Playlists](#)
- [Create Playlist](#)
- [Macro Commands and Events](#)
- [Macro Language](#)
- [Monitoring Playlists](#)
- [MOS configuration](#)
- [Viz Gateway Administrator's Guide](#)
- [Viz Content Pilot User's Guide](#)
- [Creating Transition Effects](#)

## 7.10.1 New Playlist



The control bar is used to set the concept, search, set filters, manage filters, run, pause, stop and loop the playlist. Available concepts are the same as for the show. However, a concept set for the show does not override the playlist's concept and conversely.

This section contains information on the following topics:

- [Context Menu](#)
- [Columns](#)

**Note:** The [New Playlist](#) is enabled by default. The [Old Playlist](#) is still available; however, in order to use it you need to disable the "Use the new playlist by default" option (see [General](#) settings).

## Context Menu

Direct Take	
Continue	
Out	
Read	
Remove	Select All
Rename	Select Inverse
Create Group	Select None
Set External Cursor	Expand All
Clear External Cursor	Collapse All
Activate Channel Allocator	Hide Empty Groups
Deactivate Channel Allocator	Wrap Text
Carousel	Font...
Tree	<input checked="" type="checkbox"/> Keep Selected Row In View

The Playlist's context menu provides options to manage pages and data elements in the playlist.

- **Direct Take:** Performs a direct take on the selected data element.
- **Continue:** Continues the playout of an element.
- **Out:** Takes the element out (hard cut).
- **Read:** Reads the element, opening it in the page editor and local preview.
- **Remove:** Removes the element from the playlist.
- **Rename:** Renames the Description property of the element.
- **Create Group:** Creates a group in the playlist. Groups can be nested.
- **Set External Cursor:** Sets the global cursor (normally related to GPI or Automation system setups) to the selected data element. This requires that the Show External Cursor (GPI) option has been enabled in the User Interface configuration dialog, see the [Page List](#) and [Cursor](#) sections for further details.
- **Clear External Cursor:** Clears the current global cursor.
- **Activate Channel Allocator:** Allows you to automatically associate channels to each video element in the playlist and perform clip playout on different channels (e.g. A and B). All channels that have "allocate" enabled in the profile configuration will be used for allocation.
- **Deactivate Channel Allocator:** See Activate Channel Allocator.
- **Carousel:** Opens a sub menu with start and stop options.
  - **Start:** Starts the carousel.
  - **Stop:** Stops the carousel after the last item has been played out.
- **Tree:** The Tree option contains submenus for selecting, expanding, collapsing, and moving data elements in the playlist.
  - **Select All:** Selects all data elements in the playlist.
  - **Select Inverse:** Inverts the selection.
  - **Select None:** Removes all selections.
  - **Expand All:** Expands all nodes in the tree, revealing the grouped data elements.
  - **Collapse All:** Collapses all nodes in the tree, hiding the grouped data elements.
  - **Hide Empty Groups:** Hides empty groups.

- **Wrap Text:** Wraps all text properties of the element, adjusting the row height accordingly.
- **Font...:** Opens the Font Chooser to set a different font for the playlist.
- **Keep Selected Row In View:** Keeps the selected row in view at all times.

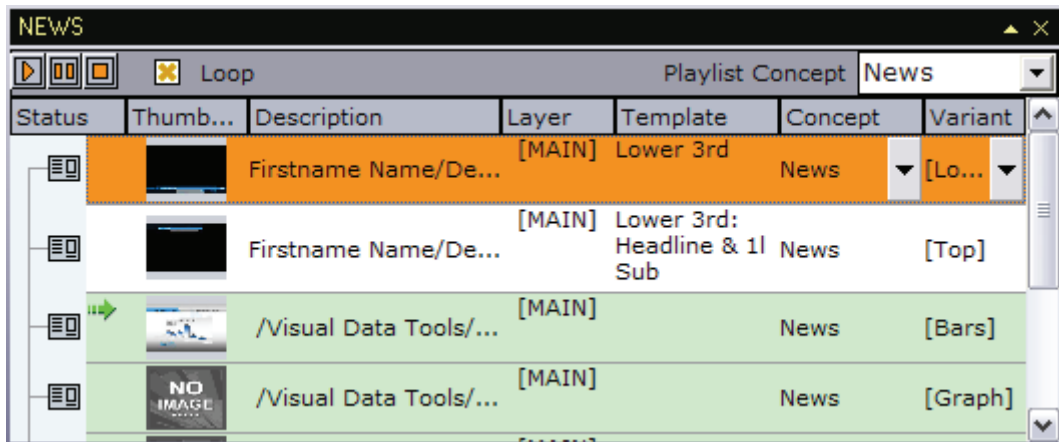
## Columns

- **Description:** Shows the description of the element. By default this will show the path of the scene. Viz Trio page descriptions may be edited in the [Page List](#).
- **Page:** Shows the page's callup code.
- **Thumbnail:** Shows thumbnails of the scenes.
- **Template Id:** Shows the template ID.
- **Template Description:** Shows the template description (e.g. name).
- **Layer:** Allows loading of graphics in separate layers on Viz Engine (front, middle, back). For example, a lower third can be shown in front of a virtual studio set or any other background, or a bug can be shown in the front layer while a lower third is shown in the middle layer. This column is presented by default.
- **Channel:** Shows which output channel an element should be sent to. Various elements can be sent to different output channels. The output channels can be set directly in the column. By default the main (program) output channel is selected, but this can be overruled by setting an alternative channel in a template (that all data elements made from it will have). This column is presented by default.
- **Concept:** Shows which concept(s) the data element is associated with.
- **Variant:** Select a concept's variant from the drop-list (see the Concept column).
- **Available:** Displays the status of external resources needed by the Viz Engine (e.g. transferred video, and if it is available on the video playout engines). Errors are shown as tooltips. This column was previously named *Progress*.
- **Loaded:** Shows the loaded status (in memory) of the scene and images used for a data element of that scene. Errors are shown as tooltips.
- **Mark In:** Sets mark in times for video clips.
- **Mark Out:** Sets mark out times for video clips.
- **Loop:** Displays a column that enables looping to be set per group instead of the whole playlist.
- **Begin:** Shows the start time of a group. The parameter is set by VCP or a newsroom system. This cannot be set using Viz Trio.
- **End:** Shows the end time for a group. The parameter is set by VCP or a newsroom system. This cannot be set using Viz Trio.
- **Duration:** Shows the length of the element. The parameter is set by VCP or a newsroom system. This cannot be set using Viz Trio.
- **Store as Default:** Stores the layout as the default.
- **Use Default:** Reverts to the default layout.
- **Auto Fit Columns:** Automatically fits all displayed columns to the given width of the playlist.

## See Also

- [Old Playlist](#)
- [General configuration](#)

## 7.10.2 Old Playlist



The control bar is used to set the concept, run, pause, stop and loop the playlist. Available concepts are the same as for the show. However, a concept set for the show does not override the playlist's concept and conversely.

When using the old playlist, and working with newsroom playlists, most of the control bar is not available with the exception of the concept drop-down.

---

**Note:** The [New Playlist](#) is enabled by default. The [Old Playlist](#) is still available; however, in order to use it you need to disable the "Use the new playlist by default" option (see [General](#) settings).

---

This section contains information on the following topics:

- [Context Menu](#)
- [Columns](#)

### Context Menu

Find...	Ctrl+F
Insert Item	▶
Delete item	Del
Set External Cursor	
Clear External Cursor	
Direct Take	
Direct Take Out	
View	▶
Tree	▶

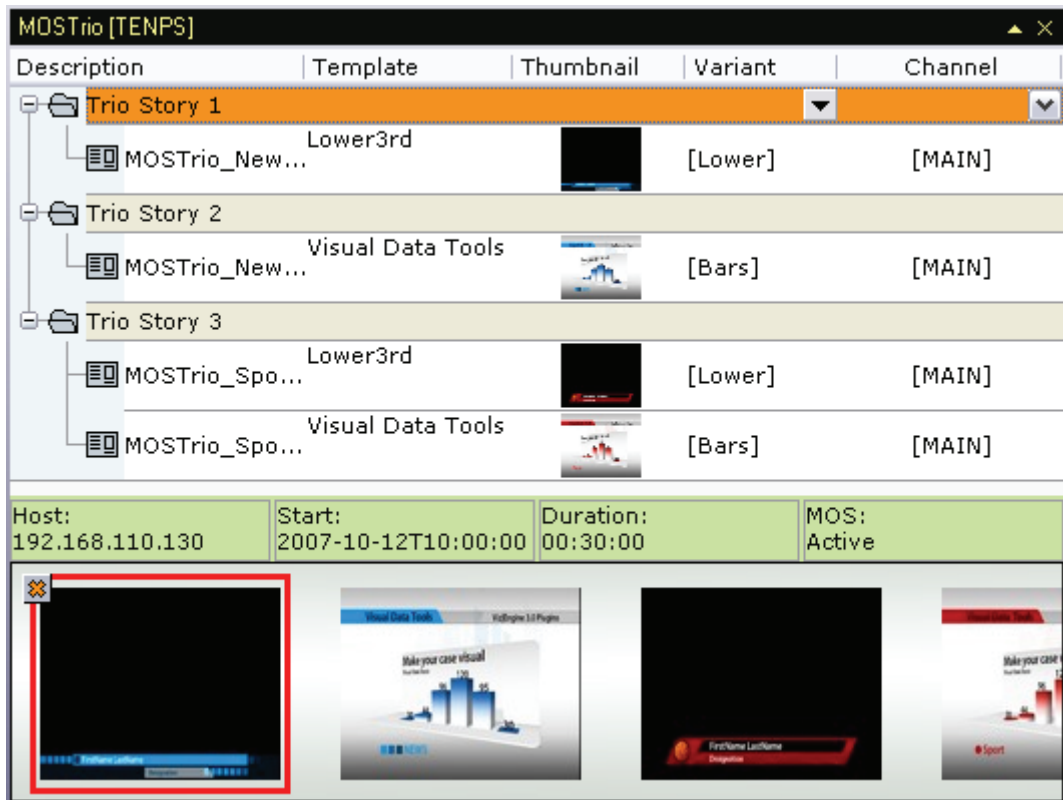
The Playlist's context menu provides options to manage pages and data elements in the playlist.



- **Find (CTRL+F):** Displays a Find dialog at the bottom of the playlist. Use this to search for elements in the playlist.
- **Insert Item: Group:** Adds a group to the playlist to organize the data elements.
- **Delete Item:** Removes a data element or a group from the playlist.
- **Set External Cursor:** Sets the global cursor (normally related to GPI or Automation system setups) to the selected data element. This requires that the Show External Cursor (GPI) option has been enabled in the User Interface configuration dialog, see the [Page List](#) and [Cursor](#) sections for further details.
- **Clear External Cursor:** Clears the current global cursor.
- **Direct Take:** Performs a direct take on the selected data element.
- **Direct Take Out:** Takes out the selected data element or any data element loaded in the same layer as the selected data element.
- **View:** The View option contains submenus that customizes the Playlist Control dialog.
  - **Show ImageBar:** Enables an image bar to be displayed with snapshots of the data elements in the lower part of the playlist.
  - **Filters:** Selects a filter. Filters for playlists cannot be defined in Viz Trio. Filters are defined using the Viz Template Manager, and are only defined for VCP templates.
  - **Font:** Displays a Font dialog. Used to define the playlist text's font, style, and size.
  - **Settings: Store as Default:** Stores the current settings as default.
  - **Image Size:** Modifies the size of the data elements to small, medium, or large.
- **Tree:** The Tree option contains submenus for selecting, expanding, collapsing, and moving data elements in the playlist.
  - **Select All:** Selects all data elements in the playlist.
  - **Expand All:** Expands all nodes in the tree, revealing the grouped data elements.
  - **Collapse All:** Collapses all nodes in the tree, hiding the grouped data elements.
  - **Allow Move:** Allows the playlist's data elements to be organized manually.

## Columns

Figure 18: Newsroom playlist



Begin, end, timecode and duration can only be set directly in the playlist by VCP or the newsroom system operators. The latter requires that the newsroom playlist propagates the necessary information.

**Note:** VCP is able to override timing information from newsroom playlists, however, this is not recommended.

- **Begin:** Shows the start time of a group. The parameter is set by VCP or a newsroom system. This cannot be set using Viz Trio.
- **Channel:** Shows which output channel an element should be sent to. Various elements can be sent to different output channels. The output channels can be set directly in the column-field. By default the program output channel is selected. This column is displayed by default.
- **Concept:** Shows the current concept, and which concept(s) the data element is associated with.
- **Description:** Shows the element name as it was saved in the database.
- **Duration:** Shows the length of the element. The parameter is set by VCP or a newsroom system. This cannot be set using Viz Trio.
- **Effect:** The Effect column contains the name of the data element's transition effect.
- **Effect Image:** The Effect Image column displays a snapshot of the Viz Artist scene which will be used to create a transition effect when the data element is taken to air.

---

**Note:** Transition effects cannot be used for transition logic scenes.

---

- **End:** Shows the end time for a group. The parameter is set by VCP or a newsroom system. This cannot be set using Viz Trio.
- **Layer:** Indicates in which Viz Artist layer the data element will be played out on-air.
  - **Loop:** Loops a full screen video clip on the video channel. Default behavior is to play once. Video clips in graphics loop by default.
  - **Available:** Displays the status of external resources needed by the Viz Engine (e.g. transferred video, and if it is available on the video playout engines). This column was previously named Progress.
- **Status:** Shows the status for graphics and video clips; cuing, cued, playing, paused, or aborted.
- **Taken:** Shows the last taken and last read data elements, as well as the external cursor (GPI).
- **Story #: Story #:** Shows the Story number(s), as they were set in the NCS.
- **Template:** Contains the name of the template used as basis for the data element.
- **Thumbnail:** Displays a snapshot of the data element.
- **Timecode:** Shows the time code for a data element. The parameter is set by VCP or a newsroom system. This cannot be set using Viz Trio.
- **Variant:** Shows the current variant selected, and which variant(s) the data element is associated with.
- **Auto Width:** When enabled, this option automatically adjusts the width of the playlist's columns. When disabled, it is possible to manually resize the columns.
- **Enable Sorting:** Enables the user to sort the playlist.
- **Clear Sorting:** Clears the list sorting and reverts the sort to default. Default sort is the sequence the playlist was created in.

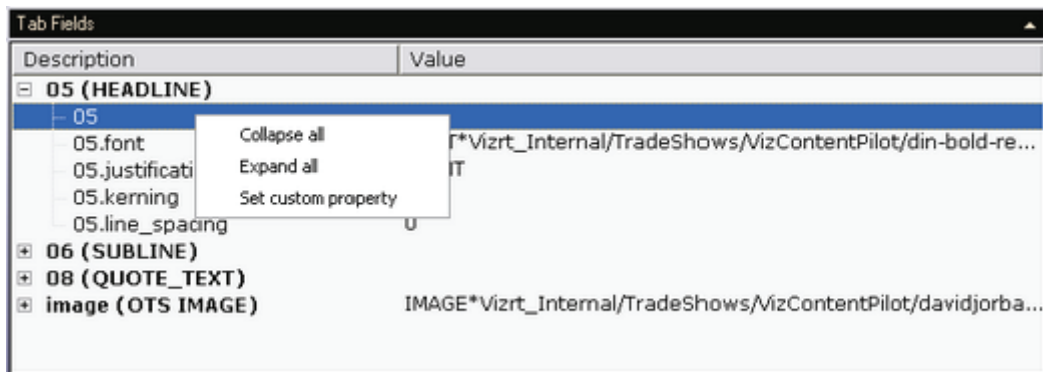
#### See Also

- [New Playlist](#)
- [General configuration](#)



---

## 7.11 Tab Fields Window



The Tab Fields window shows the tab fields for the page or template currently loaded. Click on a tab field to display the editor for an element.

When a tab field for graphics elements is selected, a bounding box highlights the object in the preview window.

When a tab field for video elements is selected, the [Search Media](#) editor is opened. A page containing video elements will automatically search for the video and preview it (see the [Video Preview](#) section).

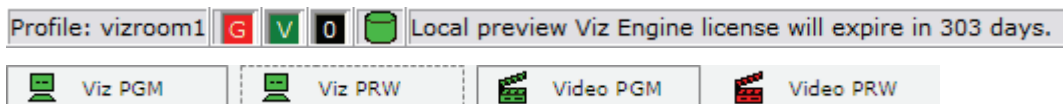
The tab-field window has a context menu for collapsing or expanding all nodes. The Set custom property is used to set a custom variable assigned to a tab field. For more information on how to use custom properties, see the [Tab-field Variables](#) section.

---

## 7.12 Status Bar

In the lower left corner of Viz Trio there is a status bar showing general status information and the Media Sequencer's connection status to devices such as Viz Engine and Viz Gateway.

**Figure 19:** Status Bar



The following icons can be displayed:

- **Profile:** Shows the currently active profile.
- **G:** Shows the Media Sequencer's connection status to Viz Gateway. The icon implies that a newsroom system, or several, is available. If no connection is established, the symbol is not visible. A Viz Gateway connection is used to retrieve and control playout of [Create Playlist](#). Double-click the icon to edit the Viz Gateway connection parameters (see the [MOS](#) section).
- **V:** Shows the Media Sequencer's connection status to the media asset management. The icon implies that a media asset management system (MAM) such as Viz Video Hub is connected. If no connection is established, the icon is not visible. A MAM connection is used to send shows and playlists, containing references to media such as video, to the MAM system. Double-click the icon

to edit the [Viz Video Hub](#) connection. The indicator icon is not shown if Viz Video Hub is not configured.

- **Active Tasks (0–n):** Indicates the number of playlist tasks that are active for the current profile(s). For more information, see the [Active Tasks](#) section.
- **Database:** The green cylinder shows the Media Sequencer’s connection status to the VCP database. If no connection is established, the symbol is not visible.
- **Messages:** Shows messages to and from the system. For example error messages, commands sent to the Media Sequencer and so on. Error messages are collected in the Error log. The log can be viewed directly from the status bar by clicking the red Error button (visible when errors occur).
- **Channel:** Displays the channel status for all channels configured for the currently selected profile.

This section contains information on the following topics:

- [Active Tasks](#)
- [To configure a profile](#)
- [To change profile](#)
- [To change channel assignment](#)
- [To check memory usage](#)

## Active Tasks

**Figure 20:** Active show playlists.

The screenshot shows a window titled "1 of 2 tasks running in active profile". It contains a table with the following data:

Description	Type	Status	Looping	Started by	Started at	Last changed...
- BGO422w1						
VizTrioShow_SPORT/SPORT	Trio Playlist	Running	True	jlr@BGO422w1	2008-09-17T10:14:23.555	
VizTrioShow_NEWS/NEWS	Trio Playlist	Stopped	True	jlr@BGO422w1	2008-09-17T10:14:38.007	jlr@BGO422w1
- vizroom1						
VizTrioShow_NEWS_2/WEATHER	Trio Playlist	Running	True	jlr@BGO422w1	2008-09-17T10:17:38.841	

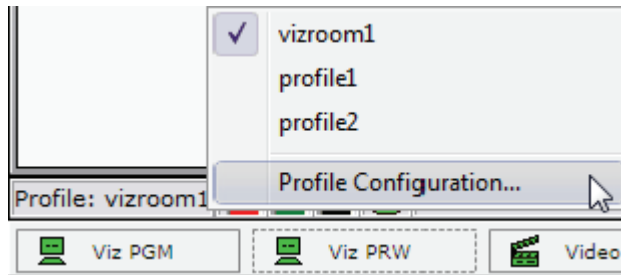
At the bottom of the window, there are several controls: checkboxes for "Show all profiles" and "Show stopped tasks", and buttons for "Cleanup", "Start", "Stop", and "Close".

The Active Tasks window displays show specific playlists that are running in active profiles. As Viz Trio is able to operate with more than one profile this status window will show if there are active tasks running in other profiles as well as different show playlists within one profile.

The Active Tasks window has the following options:

- **Show all profiles:** Shows all profiles with Show Playlists.
- **Show stopped tasks:** Shows all stopped tasks.
- **Cleanup:** Removes all information about stopped tasks.
- **Start:** Starts the selected Show Playlist.
- **Stop:** Stops the selected Show Playlist.
- **Close:** Closes the Active Tasks window.

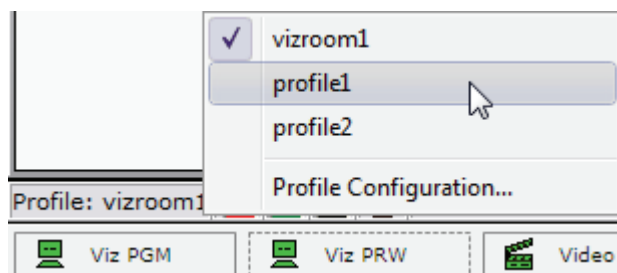
### To configure a profile



- Right-click the currently active profile (displayed on the statusbar) and select [Profile Setups](#).

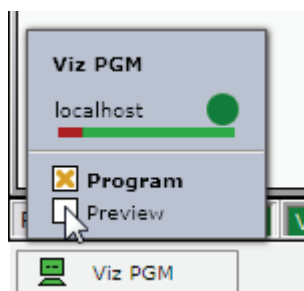
**Note:** You cannot change profiles unless you have the `-control` parameter set.

### To change profile



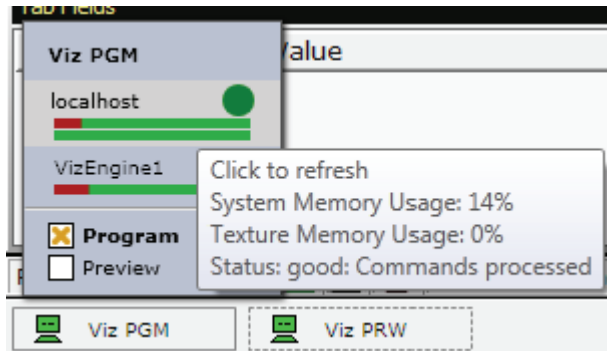
- Right-click the currently active profile (displayed on the statusbar) and select a new profile from the appearing pop-up menu.

### To change channel assignment



- Right-click a channel and select whether it should be assigned as a program or preview channel.

### To check memory usage



- Right-click a channel and hover your mouse pointer over one of the bars to see the total system and texture memory usage (in percent as depicted above).

### See Also

- [Configuration Interface](#)
- [Show Modes](#)
- [Playlist Modes](#)

---

## 7.13 Page Editor

Figure 21: Page editor



The page editor is used for editing pages, and depending on the template or page edited it can display a wide range of specialized editors. Page Editing is done using a wide range of editors; For example the text, database and scroll editors. All are covered in this section.

When pages have editable elements it is possible to jump from element to element by pressing the Tab key. It is also possible to select elements by clicking on the elements in the local preview window.

The page editor is placed in the upper right corner of the user interface and shown as the default view when starting Viz Trio.

### To edit a page

The following example uses a page to edit common text properties.

1. **Double-click** or **Read** a page from the [Page List](#) to open it in the page editor.
2. Click the **Kerning** button, or press the [Cherry Keyboard](#)'s Kern key.
3. Click the **Text** button to go back to text editing mode, or press the keyboard's Text key.
4. Click the **Take** button to play out the page on-air, or press the Take key on the keyboard.
5. Click the **Save** or **Save As** button, or press the Save or Save As key on the keyboard.

---

**Note:** It is not necessary to save a page before playing it out on air; however, a template cannot be played out before it is saved as a page.

---

This section contains information on the following topics:

- [Controls](#)
- [Text](#)
- [Database Linking](#)
- [Images](#)
- [Transformation Properties](#)
- [Tables](#)
- [Clock](#)
- [Maps](#)

## 7.13.1 Controls

The main functionality of the editor window consists of the playout, script, save and transition effect buttons. In addition to the main functionality a range of editors can be used to edit the different properties of a template or page.

This section contains information on the following topics:

- [Callup Code](#)
- [Playout Buttons](#)
- [Save Buttons](#)
- [Script Buttons](#)
- [Transition Effects](#)
- [Variants](#)
- [To create a page](#)
- [To add transition effects using the page editor](#)
- [To apply a transition effect](#)
- [To select a variant from the page editor](#)

## Callup Code

1000

The callup code field is used to save callup codes for the pages. The callup code can be set when saving a template as a page the first time, or when saving a page as a new page.

## Playout Buttons



Located above the page editor are the control buttons for playout of pages.

**Note:** Only pages (data instances of templates) can be taken on air.

The **Take** button runs on-air the page that is currently displayed in the local preview, assuming that the client is set in on-air mode.

The **Continue** button continues the animation in the page currently loaded if it contains any stop points.

The **Take Out** button takes out the page that is in the preview window. If a page is in the preview window, that is not on-air, and the Take Out button is clicked, Viz Trio will take out any element that is in the same layer as the page currently displayed in the preview.

The **Take+ Reads Next** button takes the page currently read, and reads the next one in the page list.

## Save Buttons



The **Save** button saves the currently open template in the [Page Editor](#) with a new callup code. The callup code is by default set to be the highest callup code (number) in the page list + one (1). If a page is open, clicking the **Save** button will only save the page without any changes to the callup code; however, clicking the **Save As** button will save the page with a new callup code. The page's callup code is shown in the callup code field. Press *Enter* to save the page.

## Script Buttons



The **Execute script** button is used to execute a template's script (see [execute\\_script](#) macro) and perform a syntax check. The show script is not executed unless the template script calls procedures or functions within the show

script. For the button to work the script that is to be executed must have an *OnUserClick* function wrapped around it.

```
Sub OnUserClick ()  
    ...  
End Sub
```

The **Edit script** button opens the [Script Directory](#). The editor by default opens the assigned script.

## Transition Effects



[Creating Transition Effects](#) is done using Viz Artist and can be applied to pages or playlist elements to create custom transition effects from one scene to the other. If an effect is applied, the effect will be shown when the scene is taken on-air. Effects are typically wipes, dissolves, alpha fades and so on.

The Transition Path to the transition effect scenes is set in the [Show Settings](#) window.

## Variants

Template variants can be switched directly from the [Template List](#) and [Page List](#), or from the [Page Editor](#). As with concepts, variants can also be switched using a macro command in a user-defined macro or a script.

### To create a page

1. Double-click a template from the [Page List](#) to open the template in the [Page Editor](#).
2. Edit and save it as a page with a new [Callup Code](#).

### To add transition effects using the page editor



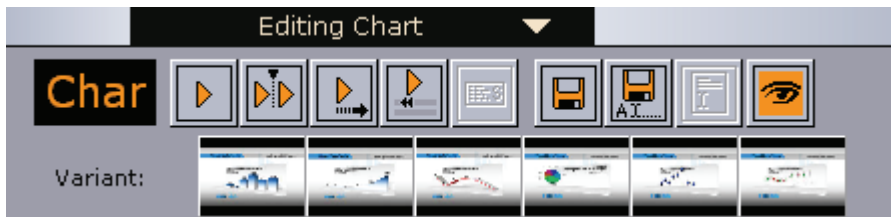
1. Open a page in the [Page Editor](#).
2. Click the [Transition Effects](#) button and select, from the appearing dialog box, a transition effect.

### To apply a transition effect

1. Right-click the column headers, and from the appearing context menu select **Effect** and **Effect Icon** (last is only for visual reference)
2. Click the **ellipse** button in the Effect column and select, from the appearing dialog box select a transition effect.

## To select a variant from the page editor

Figure 22: Variants



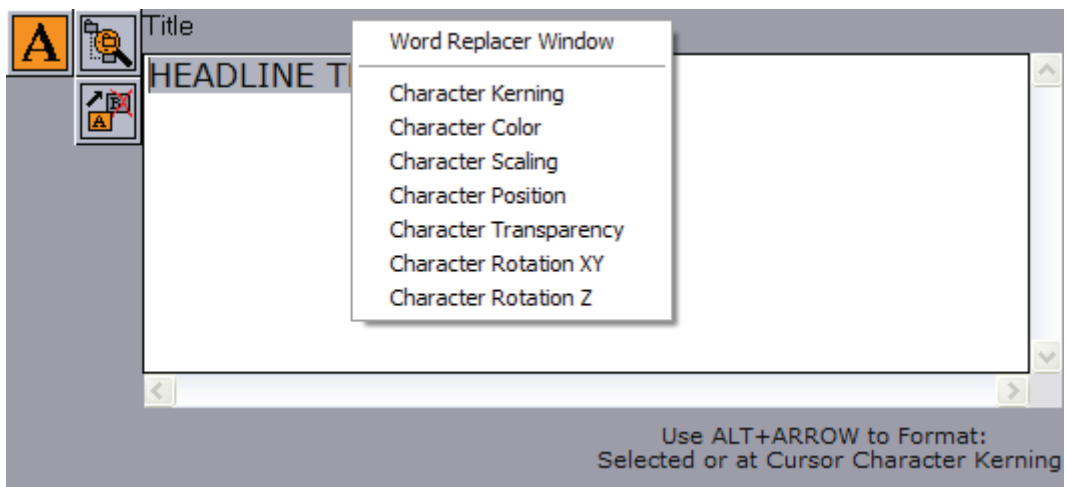
1. Open the template or page in the Page Editor.
2. Click the thumbnail images to select the variant.

## See Also

- [Scripting](#)
- [To set the transition effects path](#)
- [To add transition effects using the page list](#)
- [Creating Transition Effects in Viz Artist](#)

## 7.13.2 Text

Figure 23: Text editor's context menu



The text editor is the main editor for most users. Within the text editor it is possible to do a number of operations if the scene's text editing properties are exposed. Such operations as setting text color, adjusting the position and scale of the text.

In order to enable character formatting, the scene's Control Text plug-in must have its **Use formatted text** property enabled (*On*). This is done by the Viz Artist designer. The following can be formatted if enabled:

Table 19:

Mode	Command	Procedures
Alpha	text:set_alpha_mode	<a href="#">To edit the text transparency (alpha)</a>



**Table 19:**

Mode	Command	Procedures
Color	text:set_color_mode	<a href="#">To edit the text color</a>
Kerning	text:set_kerning_mode	<a href="#">To edit the text kerning</a>
Position	text:set_position_mode	<a href="#">To edit the text position</a>
Replace character	text:show:replace_list	<a href="#">To replace a word</a>
Rotation X, Y and Z	text:set_rotate_xy_mode text:set_rotate_z_mode	<a href="#">To edit the text rotation</a>
Scaling	text:set_scale_mode	<a href="#">To edit the text scaling</a>

The desired text manipulation mode can be assigned to a shortcut key or used in scripts. See the [User Interface](#) section for how to assign shortcut keys. The active mode is indicated in the lower right corner of the Page Editor's window. For more information on text commands, see the [Macro Commands and Events](#) section.

**Tip:** Text commands can be used in scripts or as assigned keyboard shortcuts to switch between text manipulation modes.

If text effects are used in graphics there might, in some situations, be conflicts with editing of parameters such as scaling, alpha and position. For example; If the text effect controls the scaling of text, it is not possible to change the scaling of the fonts from the text editor. The text effect's control over that parameter will override any changes made in Viz Trio. Text effects are Viz Artist plug-ins like TFX Color, Scale and Rotation that are used by scene designers.



The *Word Replacement* function allows the user to substitute a typed character with a word from a list of predefined substitution options.

The list of substitution options becomes available in the text editor when pressing an assigned shortcut key for the *text:show\_replace\_list* command, see the [Keyboard Shortcuts and Macros](#) section. The editor matches a typed character with the list of key characters that are configured.

This section contains information on the following topics:

- [To edit a text item](#)
- [To edit the text color](#)
- [To edit the text kerning](#)
- [To edit the text position](#)
- [To edit the text rotation](#)
- [To edit the text scaling](#)
- [To edit the text transparency \(alpha\)](#)
- [To replace a word](#)

### To edit a text item



1. Open the appropriate template or page, and select a text element in the Tab Fields or Preview window.
2. To edit the next item, press the Tab key or select the item from the Tab Fields list or by right-clicking the graphical elements in the Preview window.
3. To perform character specific changes, right-click to open the text editor's context menu, and select one of the options.

### To edit the text color

Figure 24: Text editor, color mode.



1. Open the appropriate template or page, and select a text element in the Tab Fields or Preview window.
2. Right-click the text editor, and from the [Text editor's context menu](#) select **Character Color**.
3. To change the color of a character or selected characters click a color in the color palette, or hold the **ALT** key down while using the right and left arrow keys to switch between the available colors.

---

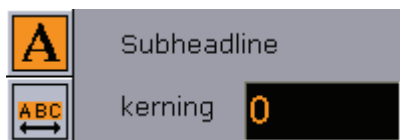
**Tip:** Color can also be set using the command `text:set_color_mode` when enabled by the designer.

---

See also [Show Settings](#) on how to customize the color palette.

### To edit the text kerning

Figure 25: Text Kerning



1. Right-click the text editor, and from the [Text editor's context menu](#) select **Character Kerning**.
2. Select a character or multiple characters.
3. Hold the **ALT** key down while using the right and left arrow keys to change the kerning, or
4. Click the Kerning button, and set the kerning for the whole text.

5. Click the Text button to switch back to the text editor.

-----  
**Tip:** Character kerning can also be set using the command *text:set\_kerning\_mode* when enabled by the designer.  
-----

-----  
**Note:** The Control Text plug-ins' Expose kerning property enables the Page Editor's kerning button.  
-----

#### To edit the text position

1. Right-click the text editor, and from the [Text editor's context menu](#) select **Character Position**.
2. Select a character or multiple characters.
3. Hold the **ALT** key down while using the arrow keys.

-----  
**Tip:** Position can also be set using the command *text:set\_position\_mode* when enabled by the designer.  
-----

#### To edit the text rotation

1. Right-click the text editor, and from the [Text editor's context menu](#) select **Character Rotation XY** or **Character Rotation Y**.
2. Select a character or multiple characters.
3. Hold the **ALT** key down while using the arrow keys.

-----  
**Tip:** Rotation can also be set using the commands *text:set\_rotate\_xy\_mode* or *text:set\_rotate\_z\_mode* when enabled by the designer.  
-----

#### To edit the text scaling

1. Right-click the text editor, and from the [Text editor's context menu](#) select **Character Scaling**.
2. Select a character or multiple characters.
3. Hold the **ALT** key down while using the arrow keys.

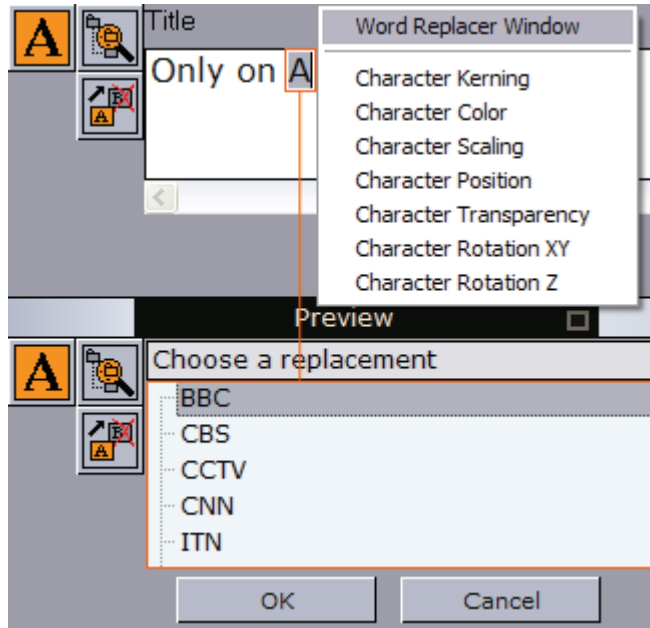
-----  
**Tip:** Scaling can be also set using the command *text:set\_scale\_mode* when enabled by the designer.  
-----

#### To edit the text transparency (alpha)

1. Right-click the text editor, and from the [Text editor's context menu](#) select **Character Transparency**.
2. Select a character or multiple characters.
3. Hold the **ALT** key down while using the arrow keys.

-----  
**Tip:** Transparency can be also set using the command *text:set\_alpha\_mode* when enabled by the designer.  
-----

### To replace a word



1. **Type** a character in the text editor. Make sure the cursor is placed after the character that will be replaced.
2. Click the word replacement button or right-click the text editor and select **Word Replacer Window**.
3. Select a replacement word from the list and click **OK**, or press the **Enter** key to return to the text editor.
4. To close the word replacement window without changes click **Cancel**, or press the **ESC** key to quit the operation and return to the text editor.

### 7.13.3 Database Linking

In order to automate the way a page is filled with content, an external data source, like a database or a Microsoft Excel spreadsheet, can be used to link a page's tab field to a value, or several values, in a database or Microsoft Excel spreadsheet (see [Show Settings](#) and how [To create a new database connection](#)).

There are two variants of the database link editor. The type of editor depends on the type of control plug-ins used in the scene. Table plug-ins such as Control List and Control Chart have their own table editor. Clicking the Database link button, shown below, opens one of the editors that can be used to link data to a page.



**Database link:** Opens one of the database link editors.



**Save to database:** Saves the updated data back to the data source.

This section contains information on the following topics:

- [Database Link Editor Common Properties](#)
- [Database Link Editor for Scalar Linking](#)
- [Database Link Editor for Table Linking](#)
- [Microsoft Excel and Unicode Support](#)

### Database Link Editor Common Properties

The Data Link editors common properties are:

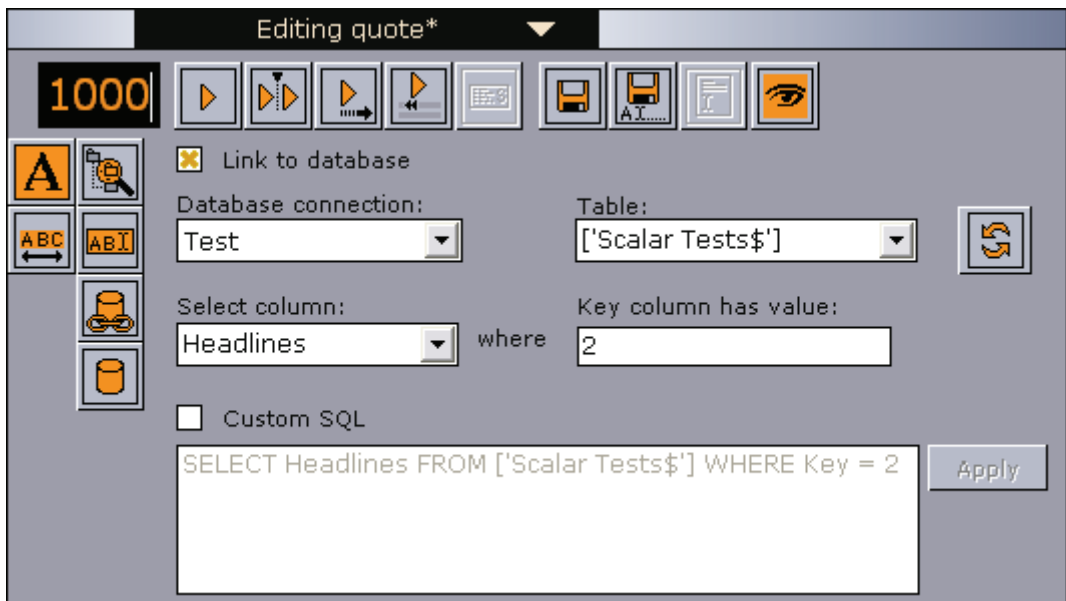
- **Link to database:** Enables the database link.
- **Database connection:** Sets the connection to a selected database link defined in the [Show Settings](#)' Databases settings.
- **Table:** Sets the database table for the lookup.



- **Refresh:** Refreshes the dataset.
- **Custom SQL:** Displays by default the resulting SQL query to get the value (scalar or table) from the database. The query can be overridden to perform enhanced queries. This is mostly useful for table properties, for example to restrict the number of rows or apply a different sorting for the rows.
- **Apply:** Applies the Custom SQL query.

### Database Link Editor for Scalar Linking

Figure 26: Database link editor for scalar linking



The database link editor used for text properties (scalar linking) has the following properties:

- **Select column:** Sets the Microsoft Excel spreadsheet column, from which the text value is selected.
- **Key column has value:** Specifies the key value to match with the lookup column, which is specified in the Databases settings (see [Show Settings](#)). This mechanism selects the row of the value to use.

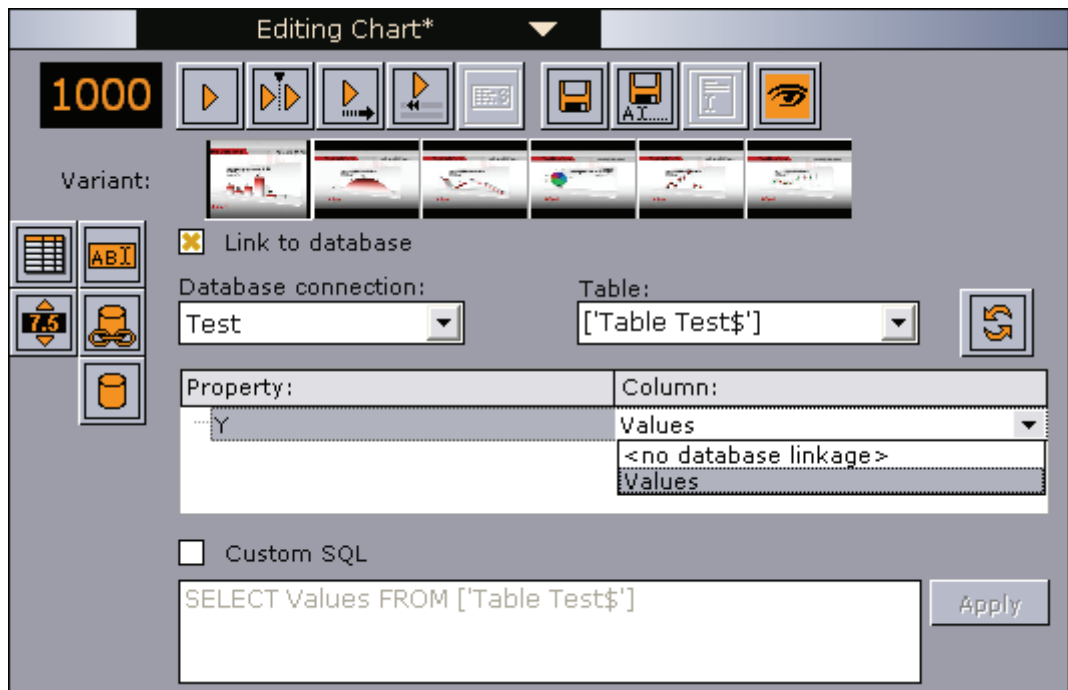
To restrict which rows from an Microsoft Excel spreadsheet to be inserted, a custom SQL query can be specified in the text box at the bottom of the database link editor (below is a Microsoft Excel spreadsheet example).

```
SELECT Value, Color FROM [Sheet1$] where Key > 2
```

**Note:** Setting both Column and Key determines both column and row of the scalar value for the text property.

## Database Link Editor for Table Linking

Figure 27: Database link editor for table linking



The database link editor used for table linking has the following properties:

- **Property:** The scenes table property that is linked with the column of the data source.
- **Column:** The data source column that is linked with the property of the scene.

**Note:** Mapping a tab-field property with a data source immediately updates the graphics.

## Microsoft Excel and Unicode Support

Viz Trio supports the use of Microsoft Excel spreadsheets to provide templates with data.

Note that Unicode characters can be used in Microsoft Excel table names and in cell values of a table, but not as column names. Supporting Unicode column names is not possible with the current implementation.

## See Also

- [To create a new database connection](#)
- The [dblink](#) macro commands for use with scripting.

### 7.13.4 Images



When the page's tab field or item is an image, the Page Editor will look like the image above. When the template contains an image you can, given that the properties are exposed, choose to load an image from file, Viz Graphic Hub, Viz Object Store or Viz Video Hub.



- **Browse File:** Opens a standard Windows file browser.



- **Browse Viz Artist:** Opens a browser in the Page Editor for browsing images on the Viz Engine. It is possible to sort by name and date. For convenience both sorts can be reversed and refreshed.



- **Browse:** Opens the [Search Media](#) pane for browsing images, video clips and person information.

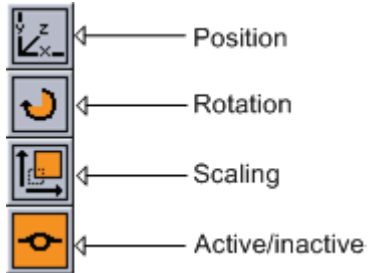
-----  
**Note:** Additional Browse buttons will appear if other image repositories are used (e.g. Viz Video HubProximity).  
-----

### 7.13.5 Transformation Properties

It is possible for scene designers to expose transformation properties for a tab field, as well as an active or inactive toggle. The following properties can be edited if enabled:

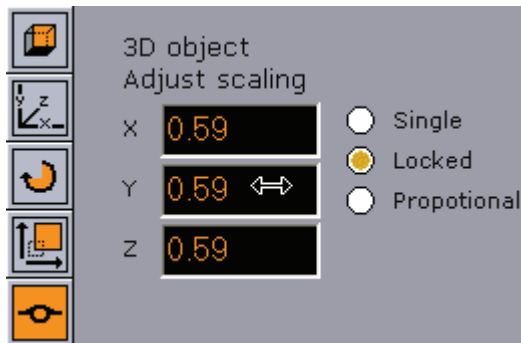
- Active
- Position
- Rotation
- Scaling

The properties are displayed as buttons on the left side of the editor:



Clicking the Position-, Rotation- or Scale button opens an editor which can be used [To set transformation properties](#).

### To set transformation properties



1. Enter the new values with the keyboard, or
2. Left-click the value field and drag in horizontal directions. Move between the fields using the CTRL + TAB keys.
3. To return to the previous selection mode, click the top button.

**Note:** The selection mode icon varies, depending on the type of object exposed. Others could be image, text and so on. The image above depicts a Geom.



## 7.13.6 Tables

#	logo (1)	product (2)	score (3)
1	Viz_2x/image/vizrt/triotutor	Viz   Trio	33
2	Viz_2x/image/vizrt/triotutorial/logos/trio		5
3			1
4	Viz_2x/image/vizrt/triotutorial/logos/weather		8
	Viz_2x/image/vizrt/triotutorial/logos/engine		
	Viz_2x/image/vizrt/triotutorial/logos/arena		

If a scene contains a Control List plug-in, a table editor will be shown for that tab field when the scene is used in Viz Trio.

The cells will have different editors depending on the data type they display. Changes made in the table editor will immediately be updated in the preview window. In addition to text fields, the table editor can show the following special editors in the table cells.



- **Image Cells:** The drop-list shows a list of all images currently available for a specific column. This provides a quick way of choosing images when many cells are to show the same image. The browse button to the right opens the image pool.

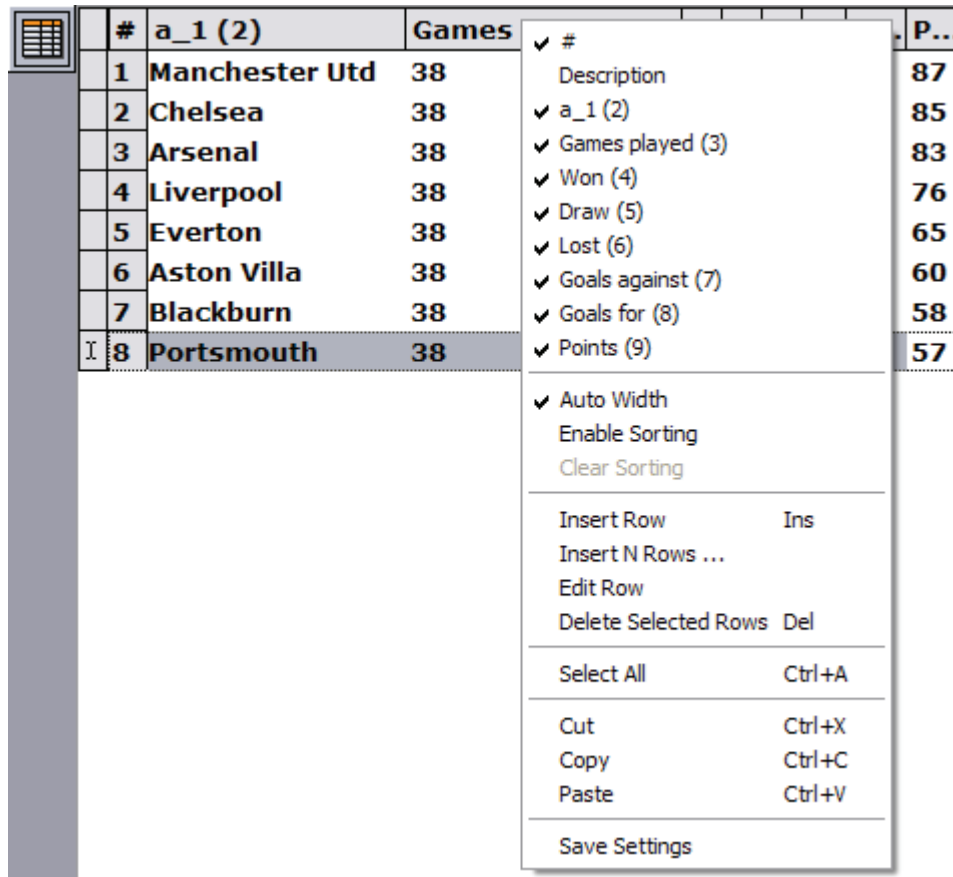


- **Number Cells:** Scenes with exposed numbers, with and without decimals, get a small text editor with a spin button functionality. Click the spin buttons, use the arrow keys, or enter them directly by keyboard to change the value.
- **Checkbox Cells:** For Boolean values, such as hide and show, a checkbox will be displayed.
- **Table Keyboard Navigation:** To move between the cells, use the arrow keys on the keyboard.

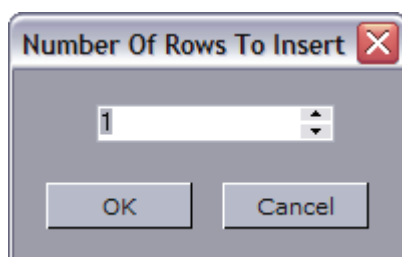
1	Viz_2x/image/vizrt/triotutorial/lk Viz   Weather	25
2	Viz_2x/image/vizrt/triotutorial/lk Viz   Engine	21
3	Viz_2x/image/vizrt/triotutorial/lk Viz   Trio	33

- **Sorting:** The table can be sorted manually by clicking on the header bar of the column to sort by. Rows can be rearranged manually by drag and drop. Click on the row once and hold the mouse button to drag and drop it to its new location.

Figure 28: Table editor context menu



- **Customizing a Table View:** By right-clicking on the table header bar, a context menu appears. Columns can be hidden or shown by selecting or unselecting the column. The table editor context menu contains the following columns:
  - #: Shows the row number (#)
  - **Description:** Shows the description field. Descriptions can only be edited in Viz Artist by the designer.
  - **Auto Width:** When disabled the column width can be set manually.
  - **Enable Sorting:** Enables manual sorting of the table.
  - **Clear Sorting:** Clears any sorting and take the table back to its initial state
  - **Insert Row (Ins):** Inserts a single row.



- **Insert N Rows ...:** Opens a small input dialog for multiple row insert. The number of rows allowed to add are limited by the scene design.
- **Edit Row:** Edit the row data.

- **Delete Selected Rows (DEL):** Deletes selected rows. For multiple select use CTRL + Left mouse key or SHIFT + Arrow keys up or down.
- **Select All (CTRL+A):** Selects all rows.
- **Cut (CTRL+X):** Cuts rows out of the table. Press CTRL+V to re-add the row(s).
- **Copy (CTRL+C):** Copies rows from the table. Press CTRL+V to add the new row(s).
- **Paste (CTRL+V):** Adds the cut or copied rows to the table.
- **Save Settings:** Saves the column setting for an active template.

---

**Note:** How many rows that can be inserted or deleted depends on the configuration of the plug-in. See the Chart and List plug-ins in the Viz Artist User’s Guide.

---

### Populating Tables Using Macro Commands

By using Viz Trio’s macro language, a table can be populated with data through a script, or through an external application that connects via a socket connection. To populate a table, the macro `table:set_cell_value` is used. The macro takes 4 arguments:

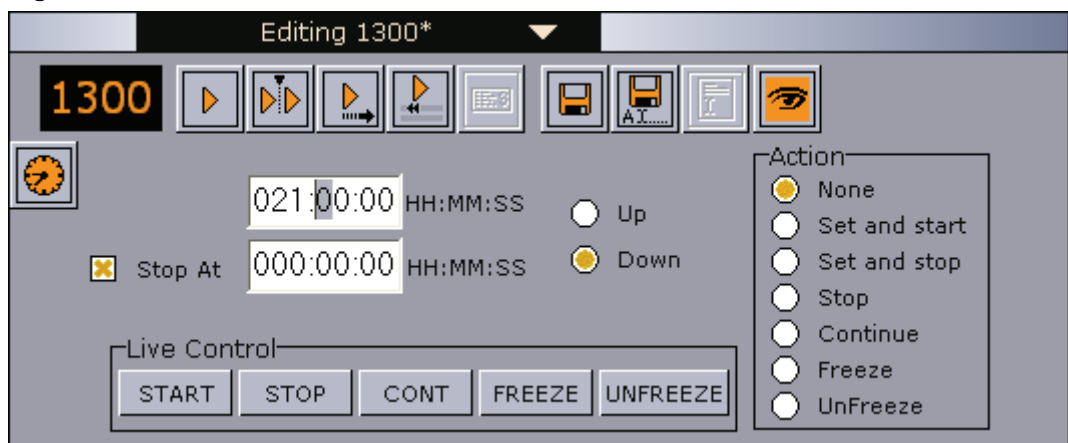
- The field identifiers for the **table** (string) and the **column** (string), the table’s **row** number (integer: starts at 0), and the **data** to be inserted.
- The example shown below will insert the text string “test”, in row 4, in the column with id 1, in the table with id 1.

```
table:set_cell_value 1 1 3 test
```

### 7.13.7 Clock

If the template contains a clock object that is made editable with the Control Clock plug-in, a clock editor opens when tabbing to the clock’s tab field.

**Figure 29:** Clock editor



The clock editor has two main functions:

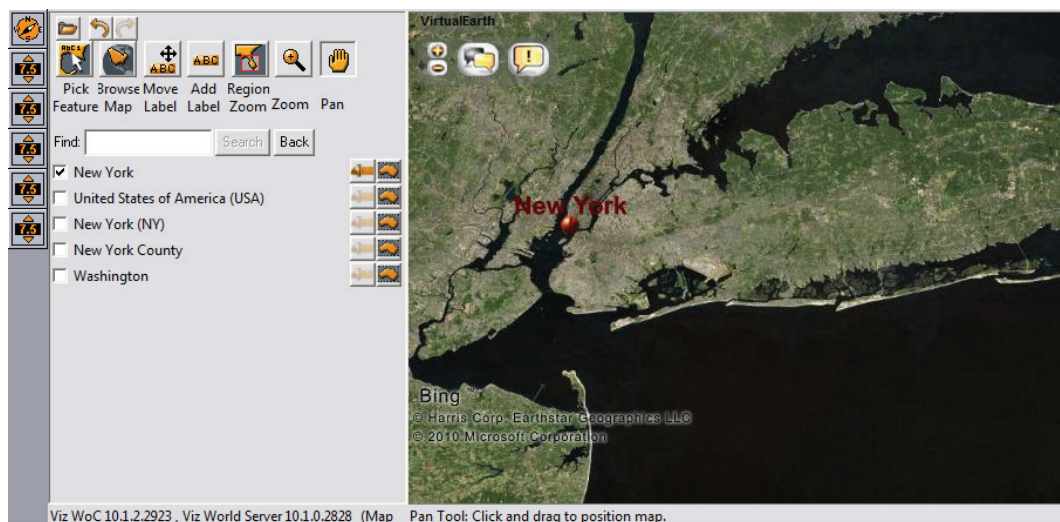
1. **Normal editor:** Specify what should happen when the page is taken on-air.
2. **Live Control:** With a page on-air, a clocks counting can be started, stopped, continued, “frozen”, or “unfrozen”.

The clock editor's properties are:

- **Start At:** Sets the starting time for the clock.
- **Stop At:** If enabled, a stop time for the clock can be set in the template.
- **Up/Down:** If enabled, the direction the clock should run can be chosen in the template.
- **Action:** Select what the clock is to do when the page is taken on-air. Either:
  - **None:** The time is just set and the clock must be started with the Live Control Start button.
  - **Set and start:** The time is set and the clock is started.
  - **Set and stop:** The time is set and the clock is stopped.
  - **Stop:** The clock is stopped when the page is taken on-air. This can be used to create a shortcut that stops the clock. Just create a page that stops the clock and link it to a key shortcut.
  - **Continue:** Similar use as with stop.
  - **Freeze:** The clock is "frozen" when the page is taken on-air.
  - **Unfreeze:** The clock is "unfrozen" when the page is taken on-air.

## 7.13.8 Maps

Figure 30: Viz World Client (small version)



There are two versions of the Viz World Client available to Viz Trio. A simplified version and a full version. The simplified version is enabled in the configuration window, see [General](#) settings under the [Miscellaneous](#) section.

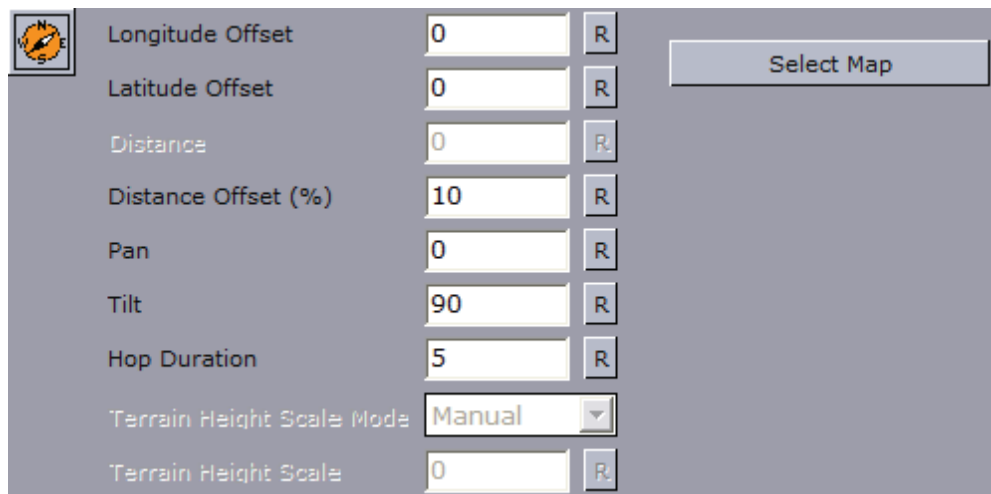


- **Edit Map:** If the map's tab-field property has more than one property exposed, clicking this button will open the corresponding editors (for example text and text kerning).



- **Browse Viz World:** Opens the full version of the Viz World Client. To use the full version, disable the small version.

**Figure 31:** Control World plugin interface



In addition to the Viz World Client editor, Viz World also provides an easy to use control interface when using the Control World plugin. The control is a replacement for the Control Map plugin with much more options and on-the-fly feedback from Viz.

**Note:** The two controls are not compatible. The new control will not work with Viz Content Pilot, and is only a part of Viz 3.3 and later.

The control can expose different fields based on the container it resides on. When tabbing to the control (in a navigator scene) the camera will jump to the map location and all feedback (exact camera position) will be immediate.

- **Longitude Offset:** Sets the longitude offset.
- **Latitude Offset:** Sets the latitude offset.
- **Distance:** Sets the distance.
- **Distance Offset (%):** Sets the distance offset in percentage.
- **Pan:** Sets the pan.
- **Tilt:** Sets the tilt.
- **Hop Duration:** Sets the hop duration.
- **Terrain Height Scale Mode:** Sets the scale mode of the terrain height.
- **Terrain Height Scale:** Sets the scale of the terrain height.
- **Select Map (button):** Opens the Viz World Editor. If the control is configured to use the simple Viz World Editor, Viz Trio must be configured to do the same.

#### See Also

- [General](#) settings for switching to simple Viz World Editor
- Viz Artist User's Guide for information on Control World
- Viz World Client and Server User's Guide

---

## 7.14 Create New Scroll

Creating a scroll allows the user in an easy way to create credit lists or any other scroll, either vertical, horizontal, with ease points and so on. When creating a

scroll from scratch Viz Trio automatically creates a scroll scene on Viz which the user can redesign.

---

**Note:** New scrolls must be created using the scroll function.

---

This section contains information on the following topics:

- [Create New Scroll Editor](#)
- [Scroll Elements Editor](#)
- [Scroll Configuration](#)
- [Scroll Live Controls](#)
- [Scroll Control](#)
- [Element Spacing](#)
- [Easepoint Editor](#)
- [Working with Scrolls](#)

### 7.14.1 Create New Scroll Editor

Create New Scroll

Create New Scroll Template:

Template Name: NewScroll

Scene Directory: 02\_GFX

SCENE\*02\_GFX/NewScroll

Template description: Viz Trio scroll

Scroll Area Width: 500

Scroll Area Height: 375

Show Scroll Area

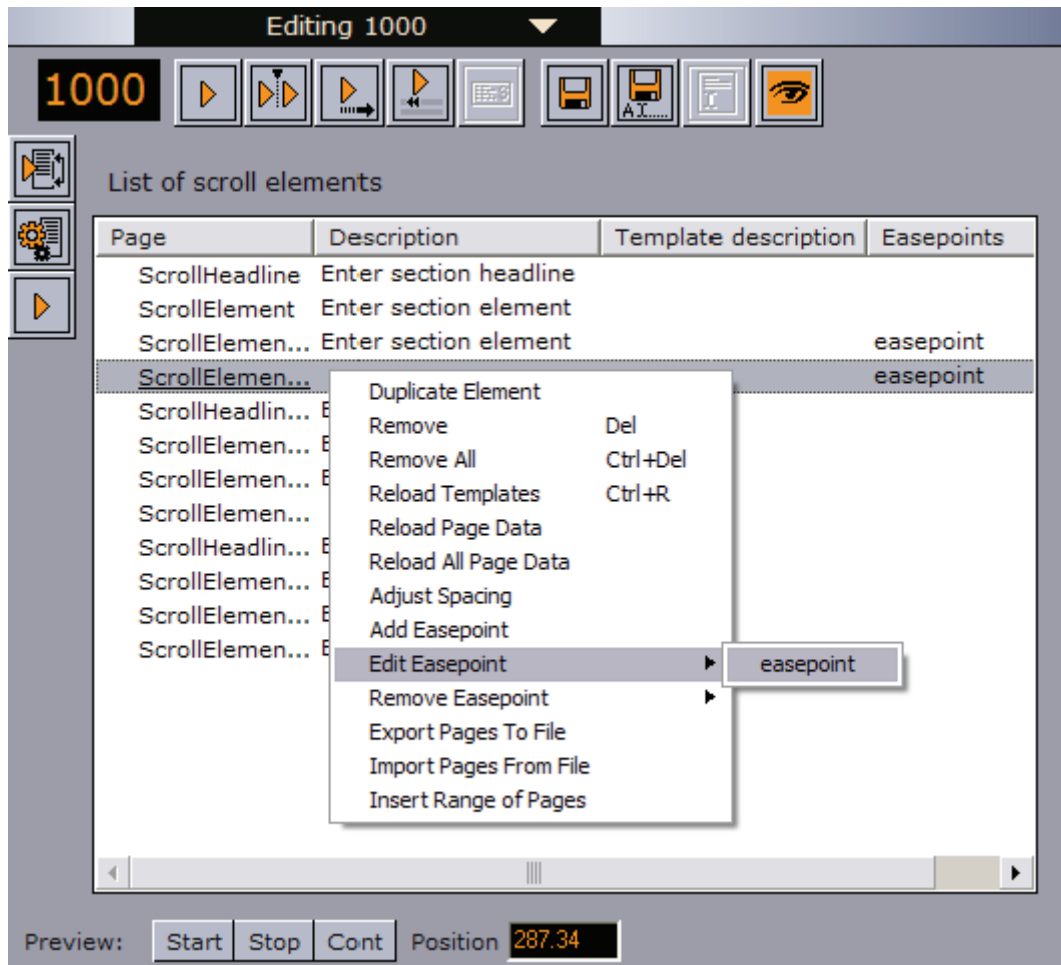
Save Template

- **Template Name:** Enter a name for the template.
- **Scene Directory:** Specify the path to a scene in which to save the template, or select a show by pressing the Browse button.
- **Template description:** Enter a description for the template.
- **Scroll Area Width:** Sets the width of the scroll area.
- **Scroll Area Height:** Sets the height of the scroll area.
- **Show Scroll Area:** Enable this option to show the scroll area in the preview window.
- **Save Template:** Saves the template in the specified scene directory.

The Scroll Elements editor opens automatically when saving or opening an already saved scroll template. A scroll is created by compiling a set of templates or pages.

## 7.14.2 Scroll Elements Editor

Figure 32: Scroll Elements Editor



Right-click on any element in the scroll list and a menu appears where actions can be performed and parameter settings defined.

The different options in the Scroll list context menu are:

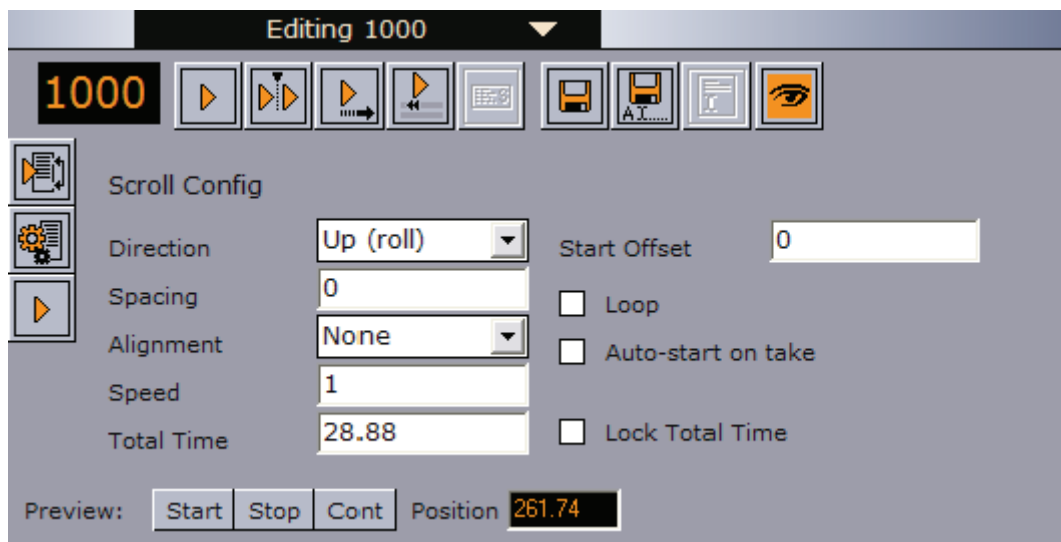
- **Duplicate Element:** Makes a copy of the selected scroll element and adds it to the scroll list.
- **Remove:** Removes the selected element from the scroll list.
- **Remove All:** Removes all elements from the scroll list.
- **Reload Templates:** Reloads the original templates.
- **Reload Page Data:** Reloads the original page data of the selected scroll element.
- **Reload All Page Data:** Reloads the original page data of all the elements in the scroll list.
- **Adjust Spacing:** Opens the [Element Spacing](#) editor which allows the user to adjust the distance before and/or after the selected scroll element.
- **Add Easepoint:** Opens the [Easepoint Editor](#) which allows the user to add an easepoint to the selected scroll element.
- **Edit Easepoint:** Opens the [Easepoint Editor](#) to adjust the easepoint settings of the selected scroll element.
- **Remove Easepoint:** Removes the ease point from the selected scroll element.

- **Export Pages To File:** Exports all scroll elements in the currently selected scroll to an XML-file.
- **Import Pages From File:** Imports a set of scroll elements from an XML-file to the scroll. Note that this option will erase the scroll elements currently in the scroll.
- **Insert Range of Pages:** This option allows the user to insert one or more pages in the scroll. Specify the range of pages to be inserted in the appearing dialog.

### 7.14.3 Scroll Configuration



Clicking the Scroll Config button opens the scroll main property page.



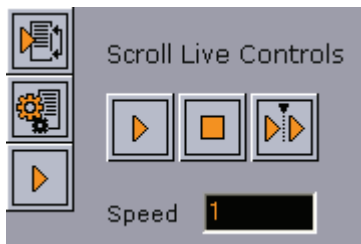
- **Direction:** Sets the direction of the scroll.
- **Spacing:** Adjusts the distance between the pages in the scroll.
- **Alignment:** Aligns the elements of the scroller to each other. The bounding box of the elements is used for aligning the objects.
- **Speed:** Sets the speed of the scroll.
- **Total Time:** Shows the total scroll time in seconds.
- **Start Offset:** Set an offset position from where the scroll should start.
- **Loop:** If enabled, the scroll elements will be looped.
- **Auto-start on take:** If enabled, the scroll will start automatically on take.
- **Lock Total Time:** Lock the total time the scroll is running. This option distributes the time evenly for all pages. Pages with more or less content are not taken into account. See the [Scroll Elements Editor](#) section for how to add and edit ease points.



#### 7.14.4 Scroll Live Controls

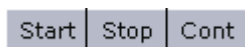


Clicking the Scroll Live Controls button, opens the Scroll Live Controls sub page, which controls the start, stop, and continue actions of a scroll on-air.



- **Start:** Takes and starts the scroll that is currently displayed in the local preview on-air, assuming that the client is set in on-air mode.
- **Stop:** Stops the scroll.
- **Continue:** Resumes the scroll.
- **Speed:** Adjusts the speed of the scroll while on-air.

#### 7.14.5 Scroll Control



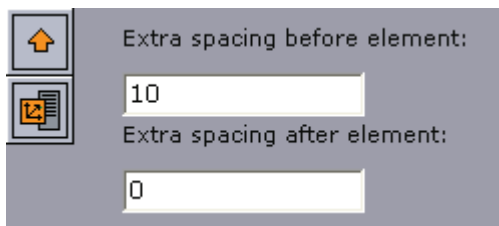
The **Start**, **Stop** and **Continue** buttons in the lower left corner of the Scroll Elements editor enables a preview of the scroll in the local preview renderer window.

- **Start:** Starts the current scroll in the local preview window.
- **Stop:** Stops the scroll.
- **Cont:** Resumes the scroll.
- **Position:** Use the Position slider to manually run a preview of the scroll. Click the value field, keep the button pressed and move the mouse horizontally. The cursor will change to an arrow to indicate that the position value can be changed.

The Scroll Elements editor has two sub-pages, [Scroll Configuration](#) and [Scroll Live Controls](#). Set parameters for the scroll using the [Scroll Configuration](#) editor, and control the scroll while on-air through the [Scroll Live Controls](#) editor.

#### 7.14.6 Element Spacing

Open the adjust element spacing editor to adjust the distance before and/or after a single scroll element:

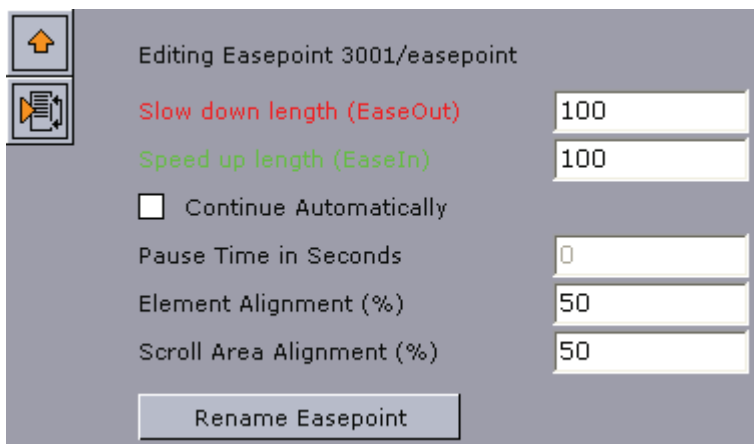


The Adjust Spacing editor's properties are:

- **Extra spacing before element:** Sets the extra distance before the element.
- **Extra spacing after element:** Adjusts the extra distance after the element.

### 7.14.7 Easepoint Editor

The Easepoint editor allows for ease point parameters to be set for a scroll element.



The Easepoint editor's properties are:

- **Slow down length (EaseOut):** Specifies the slow down length of an ease point.
- **Speed up length (EaseIn):** Sets the speed-up length of an ease point.
- **Continue Automatically:** When enabled, the scroll continues automatically after the specified pause-time.
- **Pause Time in Seconds:** Sets the pause-time in seconds.
- **Element Alignment (%):** Aligns the bounding box of an element relative to the position of an ease point.
  - The default element alignment is 50%, which aligns the center point of the element's bounding box with the ease point.
  - When creating rolls, selecting an element alignment of 0% aligns the bottom of the element's bounding box with the ease point, while 100% adjusts the bounding box's top with the ease point.
  - When creating crawls, an element of 0% aligns the left side of the bounding box with the ease point, while 100% adjusts the bounding box's right side to the ease point.
- **Scroll Area Alignment (%):** Adjusts the position of an ease point relative to the screen's center point. Scroll area alignment is defined in percentage of the scroll area.
  - The default scroll area alignment is 50%, which aligns the ease point with the center point of the screen.

- For rolls a scroll area alignment of 0%, aligns the ease point with the scroll area's lower side, while 100% adjusts it to the upper side.
- For crawls, a scroll area alignment of 0% corresponds with the scroll area's left side, while 100% aligns it at the right side.
- When in ease point edit mode, the current position of the ease point is indicated by a line in the local preview window.
- **Rename Easepoint:** Click the button to rename the ease point. Enter the new name in the appearing dialog.

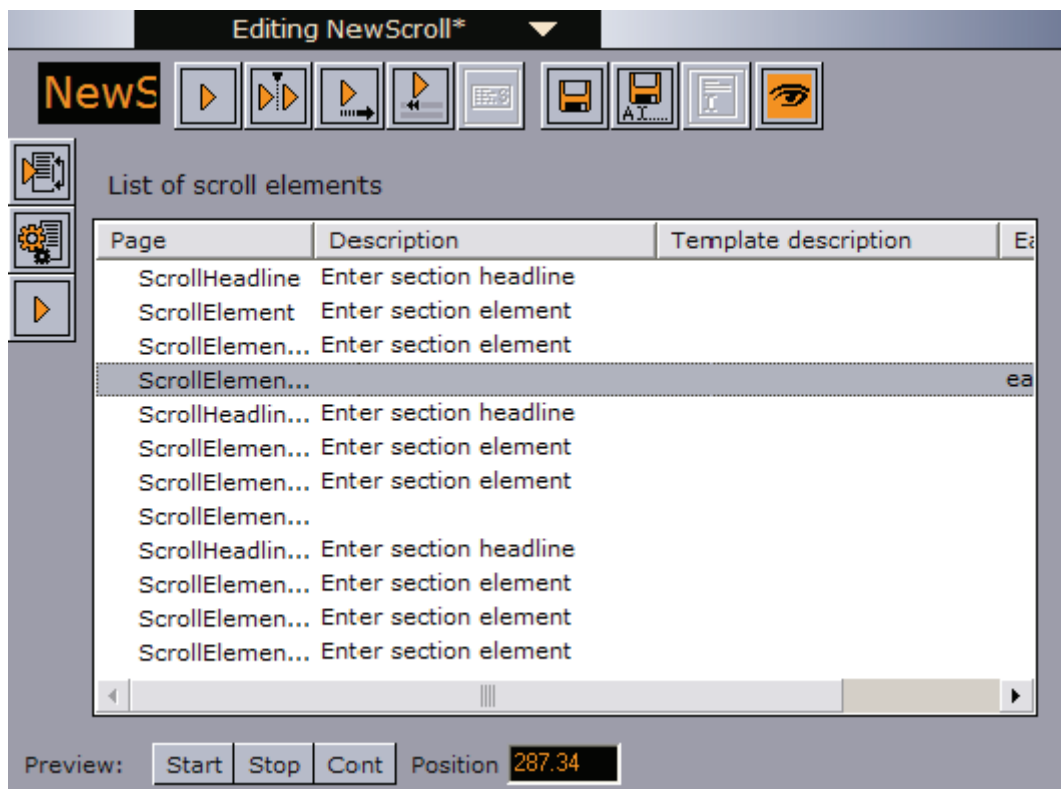
## 7.14.8 Working with Scrolls

This section contains information on the following topics:

- [To create a scroll](#)
- [To configure the scroll](#)
- [To edit a scroll element](#)
- [To copy an element in the scroll elements list](#)
- [To test a scroll](#)

### To create a scroll

1. Select **Create New Scroll** from the editor's drop-list menu.
2. Enter a **Template Name**.
3. Select a **Scene Directory**.
4. Enter a **Template description**.
5. Set the **Scroll Area Width** and **Height**.
6. Check the **Show Scroll Area** to show the scroll area in the preview window.



7. Drag and drop templates and/or pages into the [Scroll Elements Editor](#).

**Note:** Different templates for different scroll elements are needed, so the graphics designer should create templates to fill the need for scroll elements.

### To configure the scroll

1. Click the [Scroll Configuration](#) button to adjust the scroll's parameters for layout and animation speed.
2. Adjust spacing before and/or after a single scroll element using the [Element Spacing](#) editor.
3. Add and configure easepoints using the [Easepoint Editor](#) editor.
4. Click **Save** (CTRL+S) to save the scroll.

### To edit a scroll element

1. Double-click a scroll element in the [Scroll Elements Editor](#) list to open the element's editor.
2. Click the **Read Parent Page** (arrow) button to switch back to the [Scroll Elements Editor](#).

### To copy an element in the scroll elements list

1. Right-click the list of scroll elements, and select **Duplicate Element** from the appearing context menu, or
2. Drag and drop an element into the scroll list while pressing the **CTRL** button.

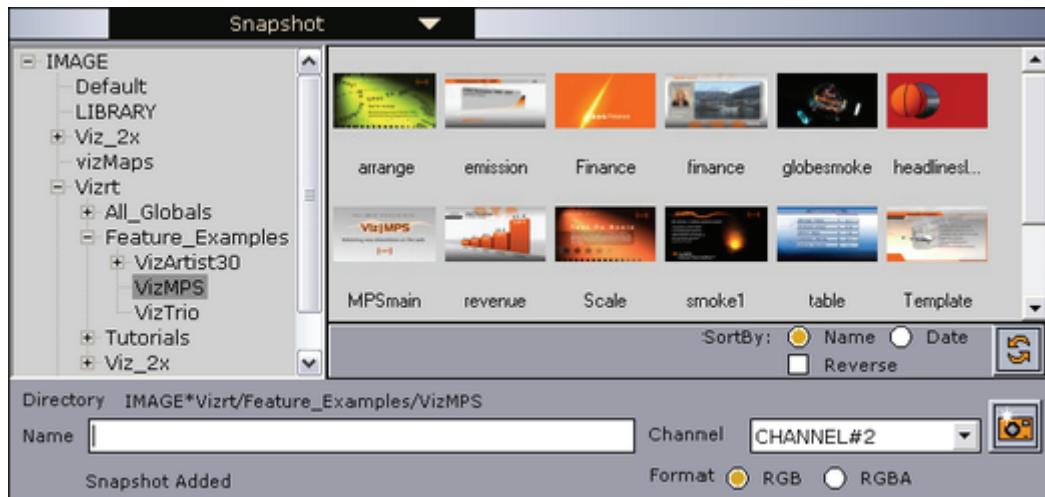
### To test a scroll

- Click the [Scroll Control](#) buttons in the lower left corner of the scroll, or scrub the **Position** value with the mouse or arrow keys on the keyboard.

---

## 7.15 Snapshot

Select *Snapshot* from the drop-down menu. This opens a window that enables a snapshot of the current view on the on-air renderer to be taken.



Specify the channel to take a snapshot from, and select the preferred format. Browse the Image database to select the folder where the snapshot is to be placed. The images can be sorted by Name or Date. Select the Reverse check-box to reverse the sorting. To refresh the content of the selected folder, click the Refresh button.

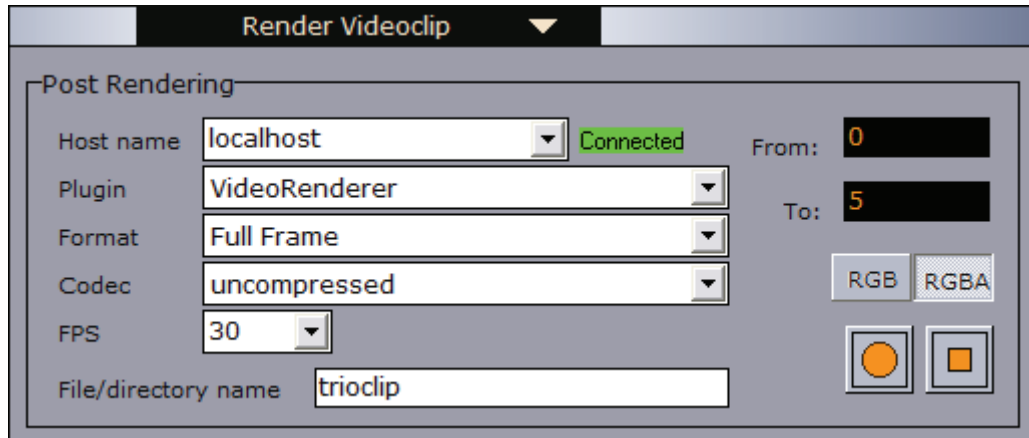


Then enter a name for the image and press the Take Snapshot button to the right. A thumbnail of the snapshot will then be visible, and the snapshot is shown in the preview window.

-----  
**Note:** Snapshots can only be deleted from Viz Artist.  
-----

## 7.16 Render Videoclip

Figure 33: Render video clip



The Render Videoclip option from the drop down menu exposes an editor for post rendering of pages to a device specified in Viz Artist, such as an AVI or a tape recorder.

Post rendering is used to create images or video files of graphical scenes. The files can be used for playout on Viz Engine. Selecting a video plug-in will create one file; however, selecting an image plug-in will render an image according to the configured frame rate. For example; Rendering a scene for five seconds will result in 250 images if the frame rate is 25 frames per second (fps).

Rendered data elements can be full screen graphics or graphics with Alpha values such as *lower thirds* and *over the shoulder* graphics.

The Viz 2.x render devices must be activated and set up in Viz Config. The render devices for Viz 3.x is automatically setup and does not need any configuration.

-----  
**Note:** Only scene-based pages can be post rendered.  
-----

- **Host name:** Sets the Viz machine that will post render the scene.
- **Plugin:** Sets the renderer device.
- **Format:** Sets the format. Available options are full frame, fields top, fields bottom, full frame/interlaced top, full frame/interlaced bottom and full frame skip. The options vary depending on the Plugin selected.
- **Codec:** Sets the codec to be used.
- **File/directory name:** Sets the filename. A full or relative path can be added. If no path is given, the file is stored in the Viz program folder on the rendering machine.
- **FPS:** Sets frame per second. Available options are 30, 60, 29.97 and 59.94.
- **From:** Sets the start time in seconds.
- **To:** Sets the end time in seconds.
- **RGB:** Sets the pixel format to RGB.
- **RGBA:** Sets the pixel format to RGBA which includes the alpha channel (blending/transparency).
- **Record button (circle):** Starts the rendering process.

- **Stop button (square):** Stops the rendering process before the configured stop time.

## 7.17 Search Media

From the drop-down menu select “Search Media”. This menu item opens a search interface that allows you to search the connected Viz Object Store database and MAM system for images and video clips. Viz Object Store traditionally stores still images and person information. Viz Video Hub traditionally stores video, audio and video stills. The Search Media option combines the two sources into one.

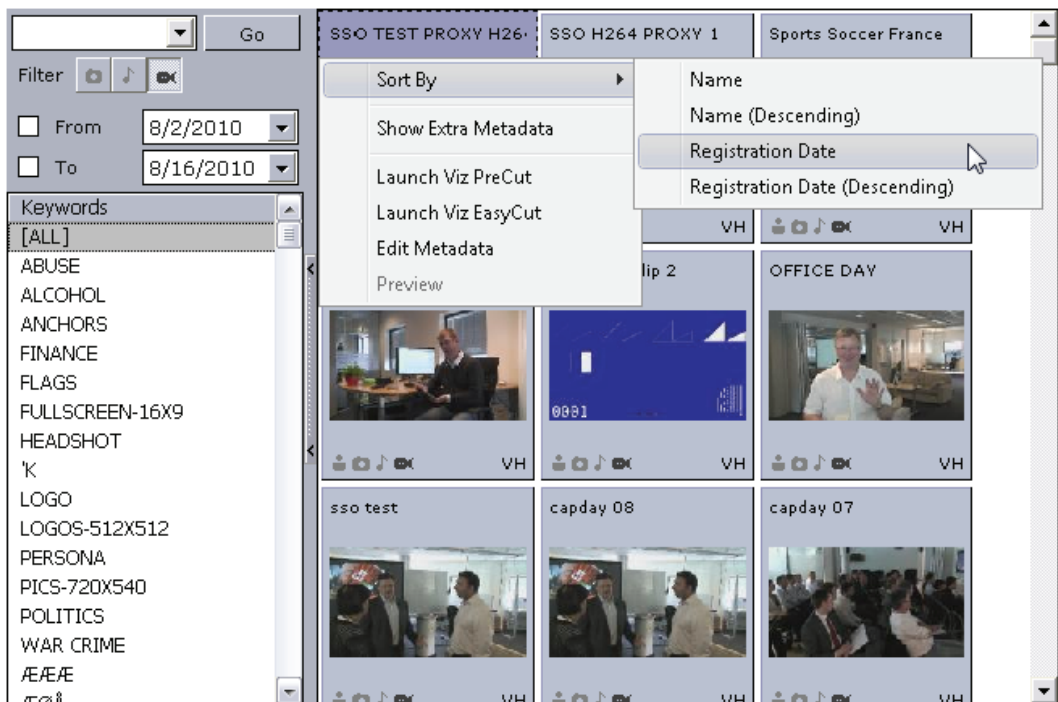
**Note:** Viz Trio does not support the use of audio files.



When editing templates and pages that have exposed image and video properties, you can search for media by clicking the **Browse Video Hub or Object Store** button.

Media items can be dragged and dropped into a show, show playlist, playlist, template or page (e.g. as full screen or part of the graphics). Double-clicking an image or video clip in any of the aforementioned scenarios will open the Search Media editor again.

**Figure 34:** Search Media



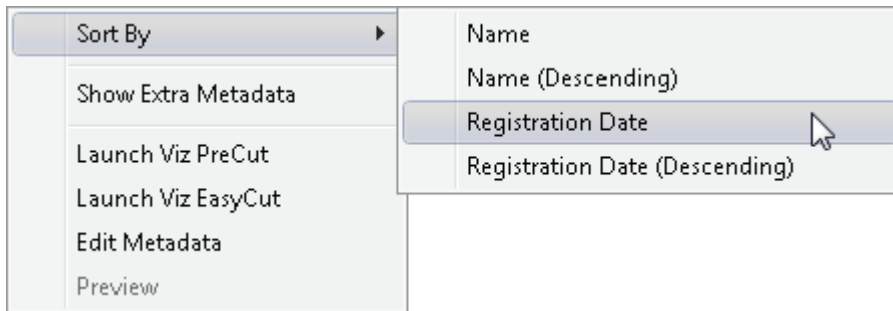
The left pane shows the [Search and Filter Options](#) and a list of categories, while the right pane displays the search result.

This section contains information on the following topics:

- Context Menu
- Search and Filter Options

## Context Menu

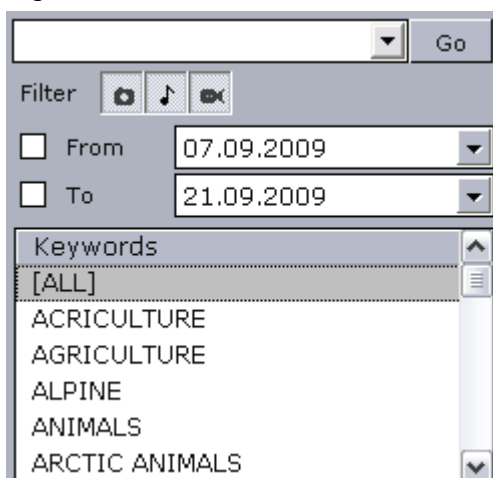
Figure 35: Media context menu



- **Sort By:** Displays a sub menu with sort options.
  - **Name/Name (Descending):** Sorts by name in ascending and descending order.
  - **Registration Date / Registration Date (Descending):** Sorts by registration date in ascending and descending order.
- **Show Extra Metadata:** Switches the media icons to display meta data such as complete filename, creation date, and clip length and so on.
- **Launch Viz PreCut:** Opens the selected video clip(s) in Viz PreCut for editing.
- **Launch Viz EasyCut:** Opens the selected video clip(s) in Viz EasyCut for editing.
- **Edit Metadata:** Enables the user to edit the meta data for the selected clip (in Viz Video Hub) for editing.
- **Preview:** Previews images using the *Windows Picture and Fax Viewer*. Is only available for Viz Object Store items.

## Search and Filter Options

Figure 36: Search and filter



- **Search field:** Combo box for entering a search criterion. Previously entered search criteria are remembered per session.
- **Filter:** Enables/disables the image, audio or video filters.



- **From/To:** Filters the search result based on From and To registration dates.
- **Keywords:** Filters the search result based on keywords.

-----  
**Note:** Viz Trio does not support the use of audio files.  
 -----

### See Also

- [Picture Database Settings](#)
- [Viz Video Hub](#)

---

## 7.18 Import Scenes

From the drop-down menu select “Import Scenes”. This menu item opens a browser window where Viz Artist scenes can be selected and enabled for Viz Trio.

**Figure 37:** Import of scenes from Viz DB.



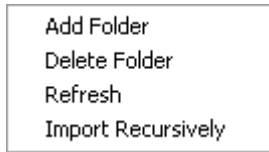
Choose the scene, or scenes, to import into the template list and click the Import button. It is possible to multi-select scenes and import scenes from any scene folder. To make the scenes ready for playout as fast as possible, press the Import and Initialize button. This will load all scenes on the program and preview renderer (Viz Engine).

The scenes can be sorted by Name or Date. Select the Reverse check-box to reverse the sorting. To refresh the content of the selected folder, click the Refresh button.

This section contains information on the following topics:

- [Import recursively](#)
- [Import scenes with Toggle or Scroller Plugin](#)

## Import recursively



Another import option is to import a whole folder, or folder tree structure. This option is a recursive import, and it does not initialize the show on the renderers.

To import recursively, right-click the folder in the scene tree and select Import Recursively on the context menu.

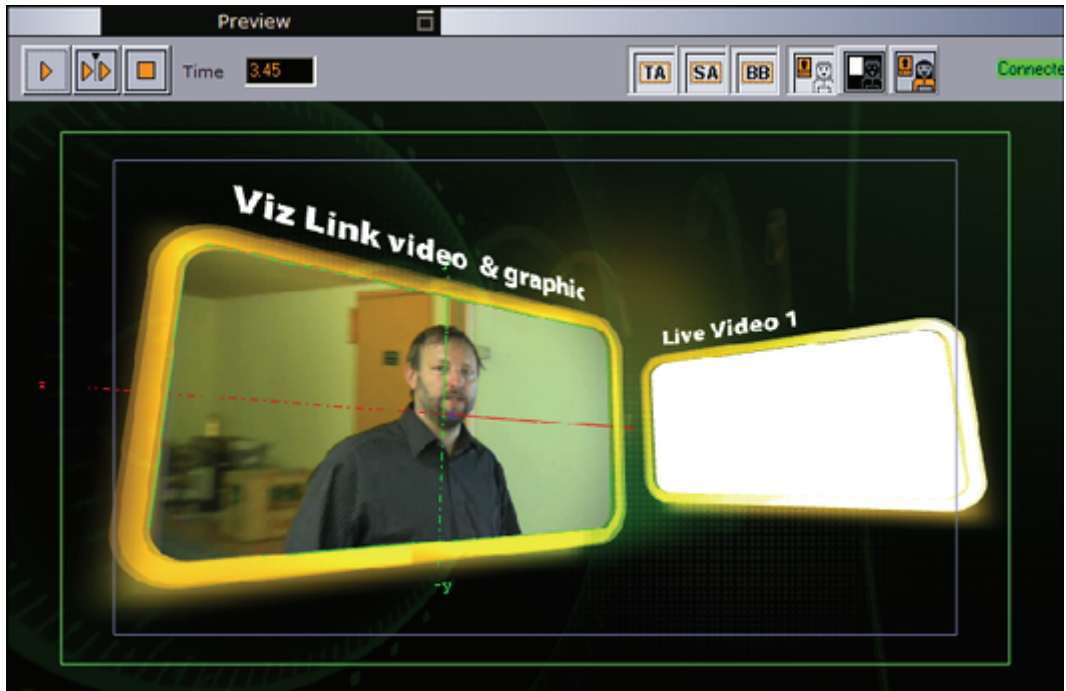
-----  
**CAUTION!** A scene with the same name as an existing template in the current show will be attempted to be merged with the existing template (e.g. scene 1000 with template 1000).  
-----

## Import scenes with Toggle or Scroller Plugin

When importing a scene with a Toggle or Scroller plugin, in combination with Control Object, you will be given a choice to import that scene or not. Under normal circumstances you will not import scenes with Toggle or Scroller plugin, but in those minute cases where you are using Toggle or Scroller plugin in scenes that are not Transition Logic or ticker related you will have the option of importing them as templates.

## 7.19 Graphics Preview

Figure 38: Real-time rendered graphics and video



The local Preview window shows a “What you see is what you get” representation of the graphics and video. In addition it will show the defined title and safe area and the bounding boxes. Bounding boxes are related to the graphics scene’s editable elements (see the [Tab Fields Window](#) section). For the full screen video preview there is also a timeline for scrubbing of full screen videos.

This section contains information on the following topics:

- [Graphics Control Buttons](#)
- [Connection Status](#)
- [Video Control Buttons](#)

### 7.19.1 Graphics Control Buttons

Above the Preview window is a button bar with some buttons and a timeline display:



The Play button starts the scene animation currently loaded in the local preview.



The Continue button continues the scene animation currently loaded if it contains any stop points.



The Stop button stops the scene animation.



With the Time slider it is possible to scrub the timeline manually by clicking in the value field, keeping the button pressed and moving the mouse horizontally back and forth. The cursor will change to an arrow to indicate that the time value can be changed.



Click this button to show or hide the Title Area in the Preview window.



Click this button to show or hide the Safe Area in the Preview window.



Click this button to show or hide the Bounding Box for the tab field currently selected in the Preview window.



**Figure 39:** Preview modes RGB, Key and RGB with background image.



The three buttons can be used to see the previewed graphics in three different modes. The first shows the graphics in an RGB mode with a default background color (black). The next shows the graphics in Key mode, showing how the key signal looks like. The last option provides an example of an RGB preview with a background image. This will illustrate how the graphics appear on top of a video.

### See Also

- [Show Settings](#) and how to set the Preview background image

## 7.19.2 Connection Status

Connected

In the upper right corner of the Preview window, the connection status of the local Viz renderer can be seen. Clicking the adjacent arrow button will display a context menu.

**Figure 40:** Local Viz Engine render menu



These functions are mainly for debug purposes:

- **Set OnAir:** Only relevant if the local preview function is on an external machine. Then this function will force the renderer in on-air mode.
- **Connect:** Only relevant if the local preview function is on an external machine. This function will reconnect to the external machine.
- **Show Commands:** Opens the command shell for the local Viz Engine renderer.
- **Show Status:** Shows a status box for the local Viz Engine renderer.
- **Restart Viz:** Restarts the local preview process if it has crashed.

## 7.20 Video Preview

Figure 41: Full screen video preview



The local preview window can also show, in addition to the [Graphics Preview](#), the compressed version of full screen video clips. Full screen video clips are added directly to the show or playlist.

This section contains information on the following topics:

- [Video Control Buttons](#)

### See Also

- [Graphics Control Buttons](#)

### 7.20.1 Video Control Buttons

Above the full screen video clip preview window is a button bar with some buttons and a timeline display:



The Play button starts the video clip currently loaded in local preview. Press the Play button to start the clip. Press it again and it will stop. If a Mark in point has been set it will jump back to the Mark in point. When loading the clip in Preview it will play the clip automatically.



The Pause button pause the video clip currently loaded in local preview. Press the Pause button to freeze the clip. Press it again to continue playing the clip.



The Stop button stops and rewinds the video clip currently loaded in local preview. If a Mark in point has been set it will jump back to the Mark in point.



The timeline has a marker and the possibility to set Mark in and out points. It also shows the actual length of the clip in hours, minutes, seconds and milliseconds.

If a Mark in and out point has been set, the length of the clip will show the time of the clip marked by the in and out points.

Use the Mark in and out points to create new time code references for Viz Engine. Save the video clip and Viz Engine will only play out the frames between the Mark in and out points. Use Save As to create new and edited versions of the original clip.

-----  
**Note:** Mark in and out only creates references for Viz Engine; hence, it does not create new clips that are saved back to Viz Video Hub. New clips are made using Viz EasyCut or Viz PreCut. For Viz Trio users this is limited to Viz PreCut.  
-----

#### See Also

- [Graphics Control Buttons](#)
- [Search Media](#)

---

## 7.21 TimeCode Monitor

Figure 42: TC Monitor

Next	Location	Unit	Overridden
Status	Remaining	Clip	
	00:00:04	VIDEO CHANNEL B	■
	00:00:05		

Selecting TimeCode (TC) monitor on the Tools menu opens the TC monitor window. TC monitor monitors the time code and the video server channels; hence,

it will only work if the system is configured to be integrated with a video server. The VCP client uses the Multiport Video Control Protocol (MVCP) to communicate with a translator (Xlator) which again communicates with the video server.

For more information on supported video server setups, please [Contact Vizrt](#).

- **Next:** Shows the next clip to be played.
- **Location:** Shows current location (time) in the clip (for example 00:10:01). This time code increases when playing the clip (see, Remaining).
- **Unit:** Shows the video channel name.
- **Overridden:** If the user checks this box, it is possible to cue a new clip on the same unit even though the tally is high (channel is currently in use/on air).
- **Status:** Shows the status of the unit, CUEING, CUED, LOADING, PLAYING and so on.
- **Remaining:** Shows the remaining time the clip has to play. The time code decreases when playing the clip (see, Location).
- **Clip:** Shows the name of the clip.

---

**Note:** The video workflow is not fully supported; hence, the TC monitor will not display information in the Next, Override and Status columns.

---

---

**IMPORTANT!** In order to monitor video clips embedded in graphics, the render engine must also be defined as a video channel (see the [Output](#) section).

---

#### See Also

- [Macro Language](#)
- [Macro Commands and Events](#)



---

## 8 Configuration Interface

A rectangular button with a light gray background and a thin black border. The word "Config" is centered on the button in a dark gray, sans-serif font.

The **Config** button opens the configuration window where the user interface, output channels, Viz Graphic Hub and other settings are applied.

The configuration window has four buttons that apply for all configuration settings made.

- **OK:** Applies any changes, and closes the window.
- **Cancel:** Closes the window
- **Apply:** Applies any changes without closing the window (ref. OK).
- **Reset:** Resets all changes made, unless they are applied locally.

This section contains information on the following topics:

- [About](#)
- [Connectivity](#)
- [Database \(Viz 3.x\)](#)
- [Directories](#)
- [Import/Export Settings](#)
- [Miscellaneous](#)
- [Output](#)
- [User Interface](#)
- [Viz Content Pilot Database](#)

---

### 8.1 User Interface


This section contains the user interface settings that are specific to the local Viz Trio client. It does not affect any other Viz Trio clients using the same Media Sequencer.

This section contains information on the following topics:

- [General](#)
- [Keyboard Shortcuts and Macros](#)
- [Text Editor](#)
- [Page List](#)

## 8.1.1 General

**Figure 43:** General settings, Configuration.



Allow Alphanumeric Imports

Enable Tabbing

Enable the numpad

Only allow numbers when doing callups

Show Tabfield list

Playlist AutoRead

Select next element in playlist after read

Callup digits

Remove MOS-playlists from show

Allow editing of MOS playlists

Use the new playlist by default

Automatic look-up of media elements when reading

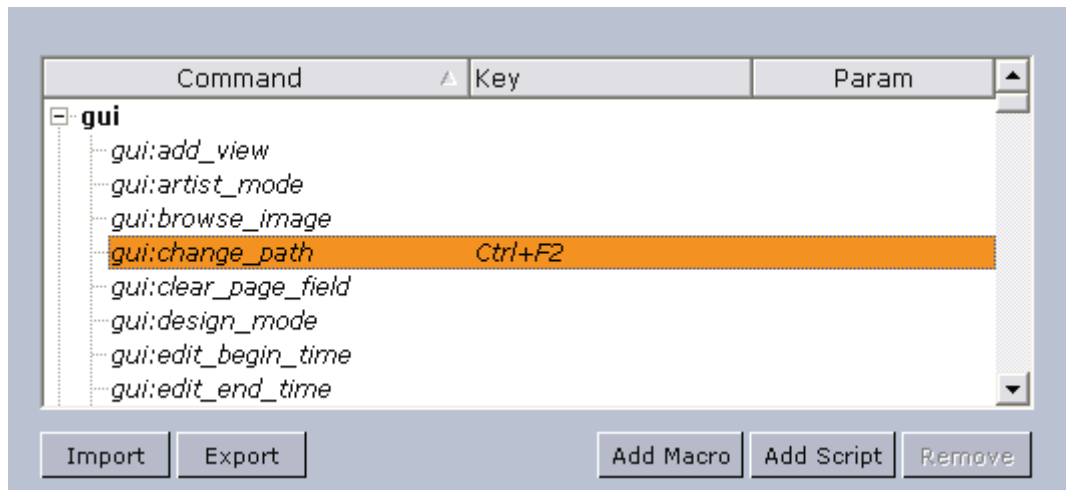
- **Allow importing scenes with alphanumeric names:** Enables import of scenes with alphanumeric names.
- **Enable Tabbing:** Use this setting to enable or disable the default behavior of the tabulator key. When enabled tabbing will iterate through the tab-fields of the element currently read.
- **Enable the numpad:** Use this setting to enable or disable the default behavior of the numeric keypad (numpad). When enabled, the numpad will type directly into the callup field. Holding down the CTRL button will disable this function
- **Only allow callup codes with numeric values:** When checked, this setting will only allow callup codes with numeric values when a page is read or saved. When unchecked, this setting will allow all values.
- **Show Tabfield List:** Select this option for the tab field list to be visible.
- **Playlist AutoRead:** When enabled, the page will automatically be read when selecting a page in a playlist.
- **Select next element in playlist after read:** When checked, enables Viz Trio to automatically select the next element in the playlist after a read.
- **Callup digits:** Select how many digits to use for the callup codes.
- **Remove MOS-playlists from show:** Removes newsroom playlists that are monitored using the MOS protocol when the show is closed. Setting this to Yes will also affect other clients connected to the same Media Sequencer. The available options are: Yes, No and Ask.
- **Allow editing of MOS Playlists:** When checked, allows the operator to edit a monitored MOS Playlist from a newsroom system. Affects only old style playlists (see [Old Playlist](#)). New style MOS playlists are always read-only (see [New Playlist](#)).
- **Use the new playlist by default:** When checked (default), Viz Trio will use a new playlist implementation that is faster than the old one. Revert back to the old

by unchecking this option. Note that changing the playlist implementation will not affect the playlist elements on the Media Sequencer.

- **Automatic look-up of video elements when reading:** When enabled, this option will automatically display the video in the video search area when a video element is read. Note that this will slow down the reading of the element.

## 8.1.2 Keyboard Shortcuts and Macros

**Figure 44:** Keyboard shortcuts and macros, Configuration.



In this section keyboard shortcuts can be defined for the different operations in the program. These shortcuts are global, which means that they are valid for all page folders. Local keyboard shortcuts can be specified per show by adding keyboard shortcuts and macros under the [Show Settings](#).

- **Import and Export:** The Import and Export function can be used to import and export shortcut settings to and from an XML-file for backup and reuse of typical configurations.

---

**Note:** The keyboard file has a *.kbd* file extension.

---

- **Add Macro:** The Add Macro button allows an operator to write a Viz Trio macro command, and link it to a shortcut key.
- **Add Script:** The Add Script button allows an operator to write a Visual Basic script, and link it to a shortcut key.

If a key combination is already in use, you are given the option of overriding it. This will leave the other command without an assigned keyboard shortcut.

Macros and scripts that are custom made can be removed using the Remove button. Select the custom made shortcut and click Remove. Note that no warning is issued when performing this operation.

---

**IMPORTANT!** Avoid using SHIFT as part of a command as this will disable the possibility to use the assigned key when writing characters (e.g. in upper case).

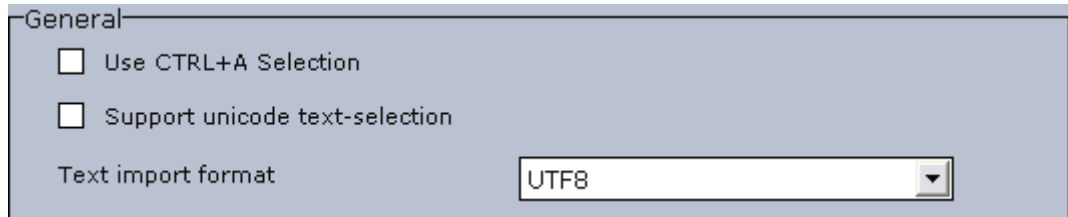
---

### See Also

- [Macro Language](#)
- [Macro Commands and Events](#)
- [Scripting](#)

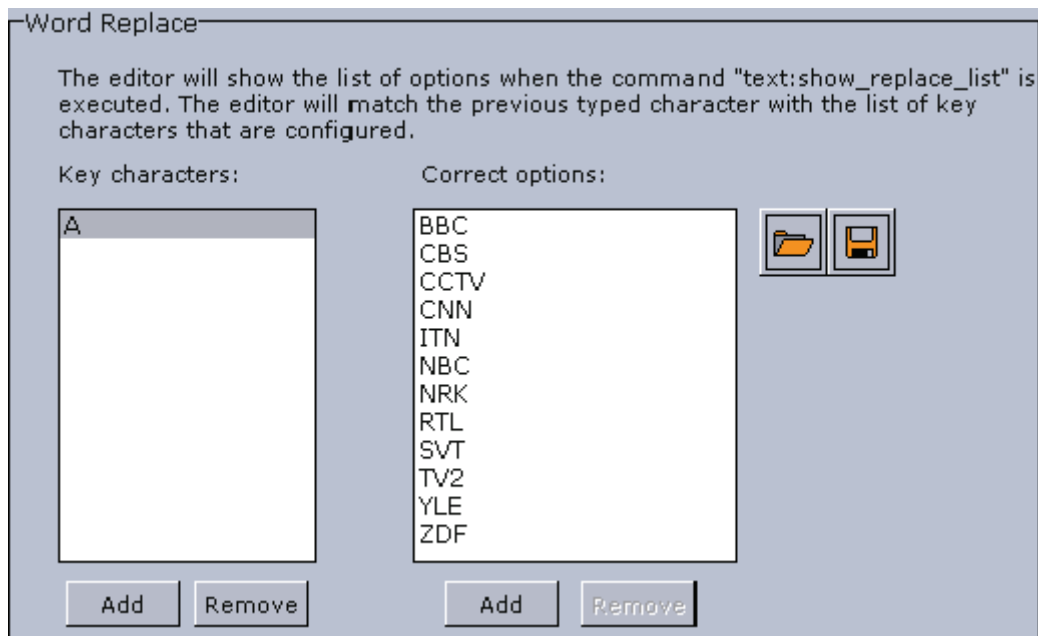
### 8.1.3 Text Editor

The text editor option has some general settings related to text formats and how to replace key characters.



The Text Editor properties are:

- **Use CTRL+A Selection:** When deselected, the normal key combination for selecting all text, CTRL+A will be disabled.
- **Support Unicode text-selection:** Use this setting to enable or disable support for Unicode text-selection. This option is used if the default language for non-Unicode programs in Windows is set to a Unicode language. For example: If copying text from a text editor that by default does not support Unicode language into Viz Trio. The representation of text would not be correct.
- **Text import format:** This option allows a character set to be specified for text imported and interpreted by Viz Trio. By default this is set to UTF-8.



Word replace allows key characters to have replace words defined. The list of replacement options becomes available in the text editor when running the "text:show\_replace\_list" command, see the [User Interface](#) section for how to define keyboard shortcuts.

The text editor matches the typed character with the list of key characters configured, and displays the corresponding replacement list. The word correction function is case sensitive, making it possible to define replacement options for both lower and upper case letters. The list of word correction options is local.

- **Key characters:** Shows a list of defined replacement key characters.
- **Correct options:** Shows a list of a key character's corresponding replacement options.
- **Add:** To add *Correct options*, a key character must be added. To add *Correct options*, select the key character, and click the associated Add button.
- **Remove:** Select a list item to remove, and click the associated Remove button.
- **Folder:** Imports a predefined list exported to an XML file.
- **Save:** Exports the predefined list to an XML file.

### Example

```
<entry name="customcharacters">
  <entry name="A">
    <entry name="CBS"></entry>
    <entry name="CNN"></entry>
    <entry name="BBC"></entry>
    <entry name="ITN"></entry>
    <entry name="NBC"></entry>
    <entry name="NRK"></entry>
    <entry name="YLE"></entry>
    <entry name="SVT"></entry>
    <entry name="CCTV"></entry>
    <entry name="TV2"></entry>
    <entry name="ZDF"></entry>
    <entry name="RTL"></entry>
  </entry>
</entry>
```

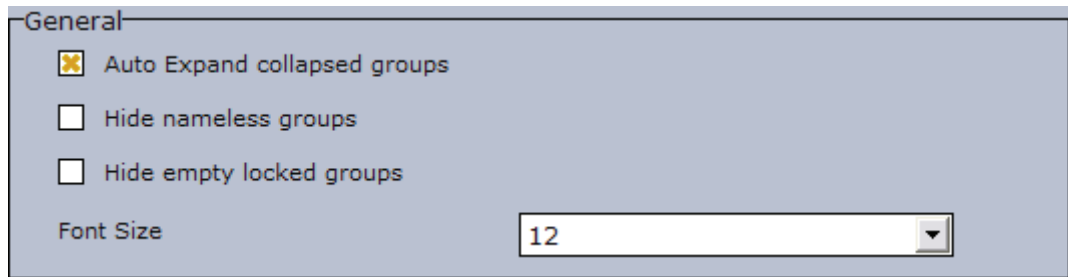
## 8.1.4 Page List

This section contains general settings as well as cursor and color settings for the Page List.

This section contains information on the following topics:

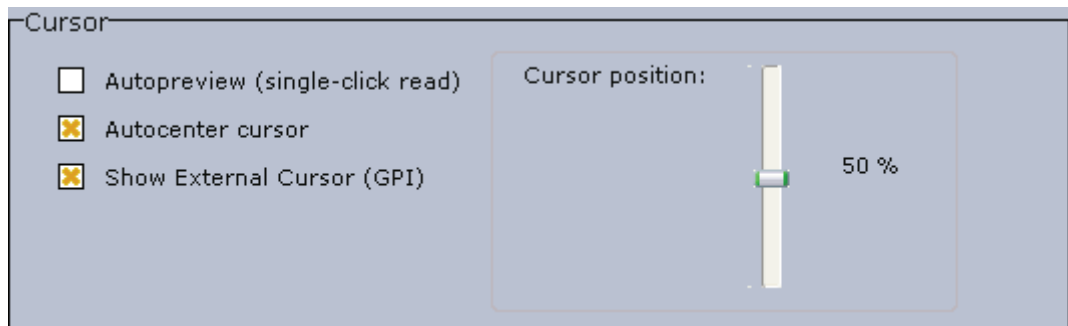
- [General](#)
- [Cursor](#)
- [Colors](#)

## General



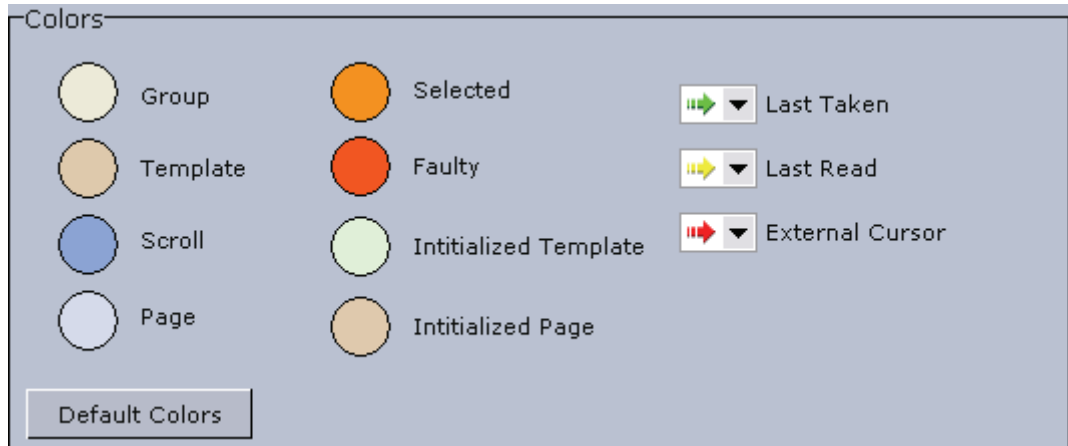
- **Auto Expand collapsed groups:** This setting will turn on or off automatic expansion of collapsed groups when doing a read.
- **Hide nameless groups:** This setting will prevent groups without a name from being displayed in the page list.
- **Hide empty locked groups:** This setting will prevent locked groups with no pages from being displayed in the page list.
- **Font Size:** Sets the font size for the Show and Page views. Show Playlist and newsroom playlists have their own font setting in the context menu (see [7.7.3 Create Playlist](#)).

## Cursor



- **Autopreview (single-click read):** When enabled, a page will be read as soon as it is selected.
- **Autocenter cursor:** With this function enabled, the cursor will have a static position when the page list is longer than the Page List window. In these situations the window is provided with a scrollbar on the right side. With Autocenter Cursor enabled, as long as some pages are hidden in the direction the list is browsed, the cursor will remain static and the list items will move instead. When the last item is revealed, the cursor will start to move again.
- **Show External Cursor (GPI):** This property is only relevant if Viz Trio is set up to be controlled by GPI. When enabled, a red arrow will show the global cursor for the GPI integration. When a “take and advance” command is issued through a GPI, the page where the cursor is, will be taken to air and the cursor will advance to the next page in the list.
- **Cursor position:** When Autocenter cursor is active, it decides where in the window the cursor stays in the situations described above. The position can be adjusted by moving the slider control.

## Colors



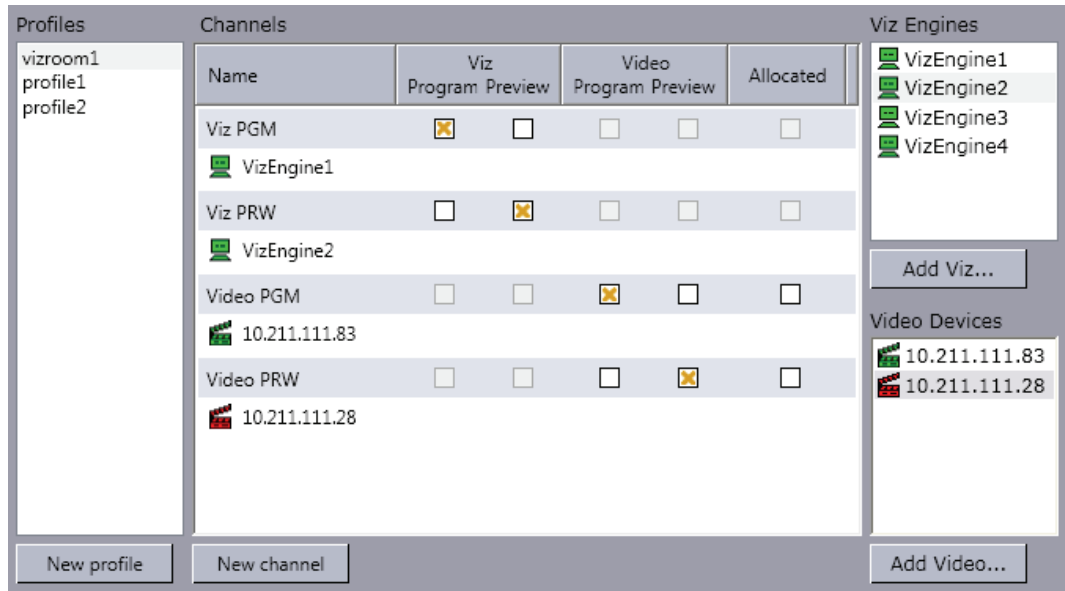
The Page List's color section provides an overview of the currently defined colors of the different elements in the page list. The element colors can be customized as needed. To change the color of an element, select the colored circle adjacent to the appropriate element label. This displays the Colors dialog, which allows the color of the selected element to be customized.

It is also possible to change the colors of the cursors (arrows) indicating the last taken and last read elements, as well as the external cursor (GPI).

Click the [Configuration Interface](#)'s global Apply button to apply the changes. Clicking the Default Colors button resets both the page list element colors to the default color settings.

## 8.2 Output

**Figure 45:** Profile Configuration



An important part of the output configuration is to create profiles for different purposes. For example; [Profile Setups](#) can be created where different channels have the program and preview function. This makes it fast and easy to switch between different output settings.

In the Output channels section profiles can be defined with different program and preview channels. Most common is to define a studio with belonging channels to renderer machines (Viz Engine hosts).

Viz Trio One Box machines running with more than one Graphical Processing Unit (GPU) may run both preview and program channels locally. The first time Viz Trio starts, and detects more than one GPU, it will automatically add a Viz host named **LocalProgramChannel** with host **localhost:6800**.

**Note:** The LocalProgramChannel host will only be set once unless the Media Sequencer is reset.

Currently Viz Trio supports four methods for resolving which channel is the local program channel. If an unsupported method is used, the [Connection Status](#) button will not show up in the [Status Bar](#). Consequently the operator will not be able to control the local program channel status.

**Table 20:** Valid methods for resolving the local program channel name

Method	Comment
localhost	For the program channel remember to add the port 6800.
127.*.*.*	Not only 127.0.0.1 refers to localhost.
Hostname	Name of computer without domain.



**Table 20:** Valid methods for resolving the local program channel name

Method	Comment
IP	The computer's IP. If the computer has multiple IPs (e.g. it has multiple network cards), it may still fail.

.....  
**IMPORTANT!** The computer may have several names (e.g. in /etc/hosts), but only the NetBIOS name can be used (i.e. the one obtained when running the "hostname" command). It is also not possible to add the domain (e.g. <hostname>.domain.internal).  
.....

This section contains information on the following topics:

- [Program and Preview Channels](#)
- [Profile Setups](#)
- [Forked Execution](#)
- [Working With the Profile Configuration](#)
- [Upgrading Old Profiles](#)

## 8.2.1 Program and Preview Channels

**Local preview** is Viz Trio's embedded render window. It is placed in the lower right corner and always shows the last read page. It displays the graphics in a true WYSIWYG manner (what you see is what you get) when editing a page. Text and images change immediately as text and images are added to the page. Select which elements to edit by pointing and clicking on them with the mouse directly in the render window.

**External preview** (Optional) is a separate Viz Engine renderer with its own reference monitor that displays a true preview. It shows how the program channel will look like before the page is taken to air.

**Program** is a separate Viz Engine renderer that renders the content that is taken on-air.

## 8.2.2 Profile Setups

**Table 21:** Profile setups

Profile	Program	Preview
Main	A (Viz 1)	B (Viz 2)
Backup	B (Viz 2)	

A typical profile setup is a two-channel setup, program and preview, with one render engine assigned to each channel. A more advanced setup is the use of forked execution. As the term suggests, forked execution is a way to configure a single output channel to contain multiple render engines.

Profiles are used to create different setups. An example for when it makes sense to use different setups would be in a backup configuration where a switch from a main renderer to a backup is needed. For example if two output renderers are named A and B, where A is program and B is preview, a profile named “Main” will then have channel A=program and channel B=preview.

A profile named “Backup” could, if the renderer that is acting as program (A) in the “Main” profile fails, have channel B=Program.

The playout can also be controlled through a General Purpose Input (GPI), for instance through hardware such as a vision mixer.

When GPI is enabled, the external cursor (the GPI system’s cursor) will be displayed/shown in any client that is using the same profile as the external system. A typical setup would be that one Viz Trio client is in the same profile as the GPI system, and functions as a “prepare station” for the producer sitting at the vision mixer desk. Data elements are then made ready and displayed on a preview visible to the producer, and then the elements are triggered from the vision mixer. This configuration needs a separate “GPI” profile that is not used by other control application clients. Other clients can be in other profiles and produce content to the same output channels. However, they need to be on other transition layers or on another Viz layer so that they will not interfere with the graphics controlled by the external system.

A channel can be designated as a Program or Preview channel by selecting the check box in the appropriate columns (Program or Preview). The program and preview channels are reciprocally exclusive – only one channel can be set to program, and only one channel can be set to preview. If for example A is set to program and B is set to preview, and then C is set to program, A will no longer be set as program.

When configured to use video in graphics from a video server (e.g. Viz Video Hub), it is currently recommended to use IP addresses for both the Viz and Video channels in order to visualize the transfer progress correctly; hence, a hostname on the Viz channel will work, but not on the Video channel. They must also match the video configuration for both program and preview.

The Video device configuration is used when configured to trigger video elements in a video server setup, hence, it is not used for other video server configurations. The Video device configuration is only used for playout of video clips from video servers.

-----  
**IMPORTANT!** Current limitations require both the graphics and video channel(s) to use an IP address.  
-----

#### See Also

- [Forked Execution](#)

### 8.2.3 Forked Execution

Forked execution can be used with standalone and transition logic scenes. It also replaces the execution of visible containers.

This section contains information on the following topics:

- [Standalone scenes](#)
- [Transition logic scenes](#)
- [Visible containers](#)

### Standalone scenes

Forked execution supports standalone scenes by executing the same graphics with different concepts on two or more render engines. Concepts are defined per channel when [Working With the Profile Configuration](#) tool.

As an option you can also use this setup to handle fail situations by having the same graphics concept rendered on both engines.

### Transition logic scenes

As with [Standalone scenes](#), forked execution supports setting concepts for Transition Logic scenes. In addition Transition Logic scenes, by defining channels with different render engine setups, can show different states of the same scene on a per engine basis. All states are synchronized for all engines (at all times) in order to achieve an artifact-free and smooth morphing of the graphics from one state to the other.

If for example you have three render engines you can set up a range of channels with different combinations of render engines per channel.

Channel	Viz Engines
A	1,2,3
B	1,3
C	1
D	2,3
E	2









If for example you have a scene with four layers each layer can be controlled separately from the other layers (see table below). By setting a state per layer you can achieve a varied output depending on the channel used and how that channel is configured in terms of render engines (see table above).


- Layer 1: Shows and hides a geometry (e.g. a cube)
- Layer 2: Shows and hides a geometry (e.g. a cube)
- Layer 3: Positions the geometries.
- Layer 4. Animates the layer 1 geometry by showing the next image or a logo.

States for layer 1	States for layer 2	States for layer 3	States for layer 4
Show cube	Show text	Position left	Next image
Hide cube	Hide text	Position right	Show logo

States for layer 1	States for layer 2	States for layer 3	States for layer 4
<ignore>	<ignore>	Position center	<ignore>
		<ignore>	

With the aforementioned scene layers and configured channels you can have the following output on each of the three engines.

Channel	Layer state	Output on Viz Engine 1, 2 and 3
Channel C (Viz 1)	Show cube Pos left Show text Show logo	
Channel D (Viz 2,3)	Show cube Pos right Show Text Show Logo	
Channel E (Viz 2)	Pos center Hide text	
Channel A (Viz 1,2,3)	Next image	
Channel A (Viz 1,2,3)	Next image	
Channel A (Viz 1,2,3)	Show text Next image	
Channel A (Viz 1,2,3)	Show logo	
Channel B (Viz 1,3)	Hide cube	

Channel	Layer state	Output on Viz Engine 1, 2 and 3
Channel E (Viz 2)	Hide cube	

### Visible containers

Viz Trio still support, though it is considered **deprecated**, a behavior similar to that of forked execution. By designing a standalone scene where each root container is a variant of the other, you can configure each render engine to render specific containers (by name). In effect a scene with two or more root containers can have one or several containers assigned and rendered visible by one render engine.

Due to potential performance issues when using large textures (e.g. HD) this option is no longer recommended. Concept and variant design of standalone and transition logic scenes is therefore the recommended design convention.

### See Also

- [Output](#)
- [Working With the Profile Configuration](#)
- Viz Artist's user guide on Transition Logic

## 8.2.4 Working With the Profile Configuration

This section contains information on the following topics


- Profile configuration:
  - [To open the profile configuration](#)
  - [To add a new profile](#)
  - [To rename a profile](#)
  - [To delete a profile](#)
- Channel configuration:
  - [To add an output channel to the Channels list](#)
  - [To rename an output channel in the Channels list](#)
  - [To remove an output channel from the Channels list](#)
  - [To add an output device to the Channels list](#)
  - [To remove an output device from the Channels list](#)
  - [To add a concept override for a channel's output device](#)
- Output device configuration:
  - [To add a Viz Engine](#)
  - [To add a video device](#)
  - [To edit a Viz Engine or video device](#)
  - [To delete a Viz Engine or video device](#)
  - [To enable scene transitions](#)
  - [To enable still preview](#)

- To render specific scene containers
- To set a different font encoding

### To open the profile configuration

1. Select **Tools** from the main menu, and then **Profile Configuration (CTRL+M)**, or
2. Right-click the profile on the status bar (lower-left) and select **Profile Configuration...** from the appearing context menu.

### To add a new profile



- In the Profile Configuration window click the **Add profile** button, and in the appearing text field enter a new unique profile name, and press **Enter**.

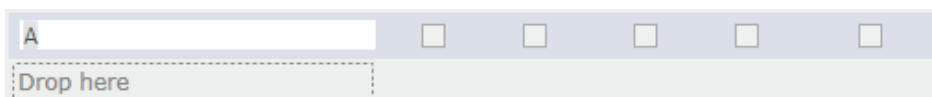
### To rename a profile

1. Right-click the profile and from the appearing context menu select **Edit Profile Name**, or simply double-click it and enter the new name.
2. When finished editing the name, press **Enter** or click the cursor outside the Profiles list.

### To delete a profile

- Right-click the profile and from the appearing context menu select **Delete Profile**, or simply select it and press the **Delete** button.

### To add an output channel to the Channels list

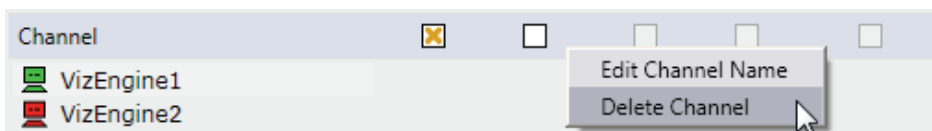


- Click the New Channel button, or drag and drop a Viz Engine or video device to the Channels list.

### To rename an output channel in the Channels list

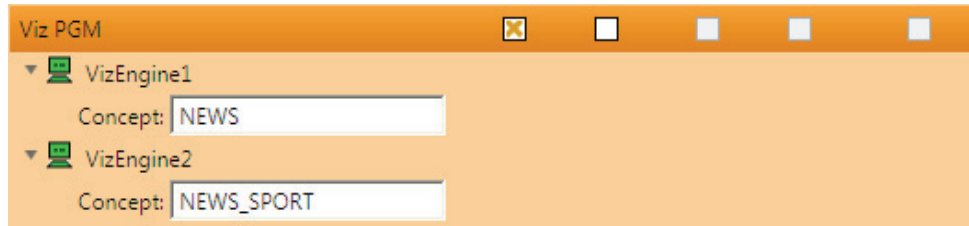
- Right-click the channel and select Edit Channel Name from the appearing context menu, or double-click the name.

### To remove an output channel from the Channels list



- Right-click the channel entry and select Delete channel from the appearing context menu, or select the channel and press the Delete button.

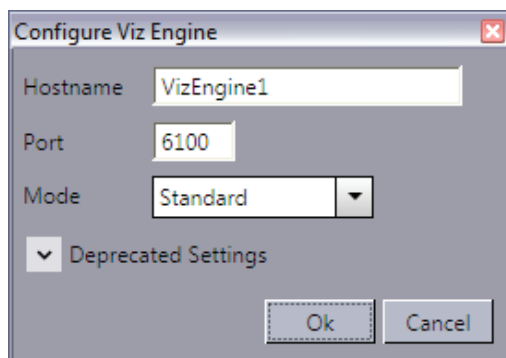
### To add a concept override for a channel's output device



1. **Expand** the channel's output device and append the concept name.
  - This will override any concepts set elsewhere
2. Click **OK**

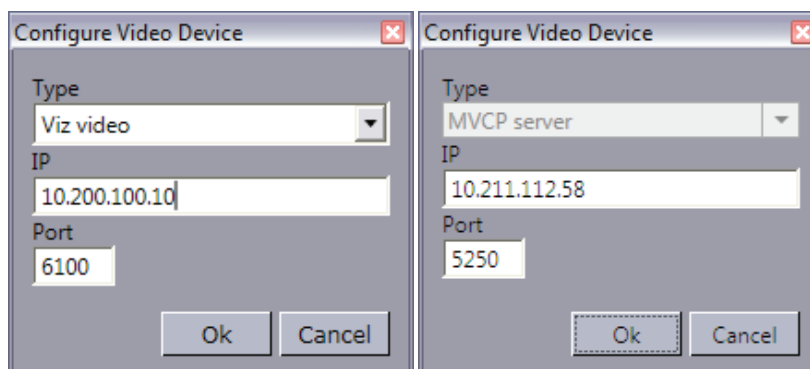
-----  
**CAUTION!** Note that concept names are case sensitive.  
-----

### To add a Viz Engine



1. Click the **Add Viz...** button in the [Profile Setups](#) window to open the Configure Viz Engine dialog box.
2. Enter the hostname and port (optional).
3. Click **OK**
  - A status indicator will show if the renderer is on-air.

### To add a video device



1. Click the **Add Video...** button in the [Profile Setups](#) window to open the Configure Video Device dialog box.

2. Select the video server type.
3. Enter the IP address and the port (optional).
4. Click **OK**.
  - A status indicator will show if the video device is online.

-----  
**Tip:** If you to transfer video from Vizrt's MAM systems you can use the default **vizroom1** profile (unless you configure a profile manually in both systems yourself).  
 -----

#### To edit a Viz Engine or video device

- Right-click the device and select **Edit** from the appearing context menu, or simply **double-click** it.

#### To delete a Viz Engine or video device

- Right-click the device and select **Delete** from the appearing context menu, or simply select it and press the **Delete** button.

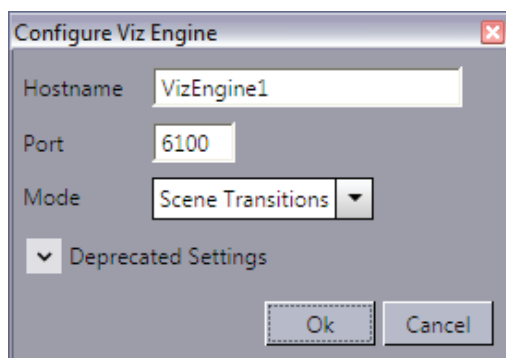
#### To add an output device to the Channels list

- Simply **drag and drop** a Viz Engine or video device onto the channel in the Channels list, or select it and from the appearing context menu select **Add to profile** (creating a new channel) or **Add to selected channel**.

#### To remove an output device from the Channels list

- Right-click the Viz Engine or video device and from the appearing context menu select **Delete Output**, or simply select it and press **Delete**.

#### To enable scene transitions

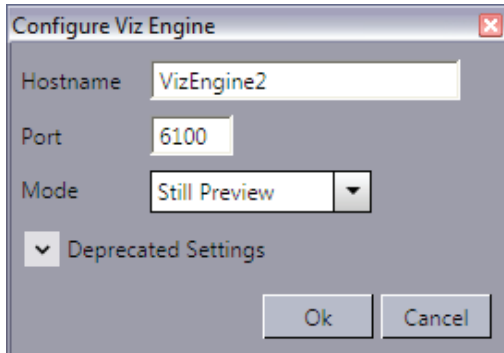


1. Configure the Viz Engine settings as seen in how [To add a Viz Engine](#)
2. Set the **Mode** to **Scene Transitions**
  - **Scene Transitions:** Allows the renderer to copy (or snapshot) the scenes to create a transition effect between them.
3. Open the [Show Settings](#), and set the Transition Path.
  - See also [Transition Effects](#)
4. Click **OK**
5. Add the program renderer to the program channel



-----  
**Note:** To see the effects the program channel must be configured and on-air.  
-----

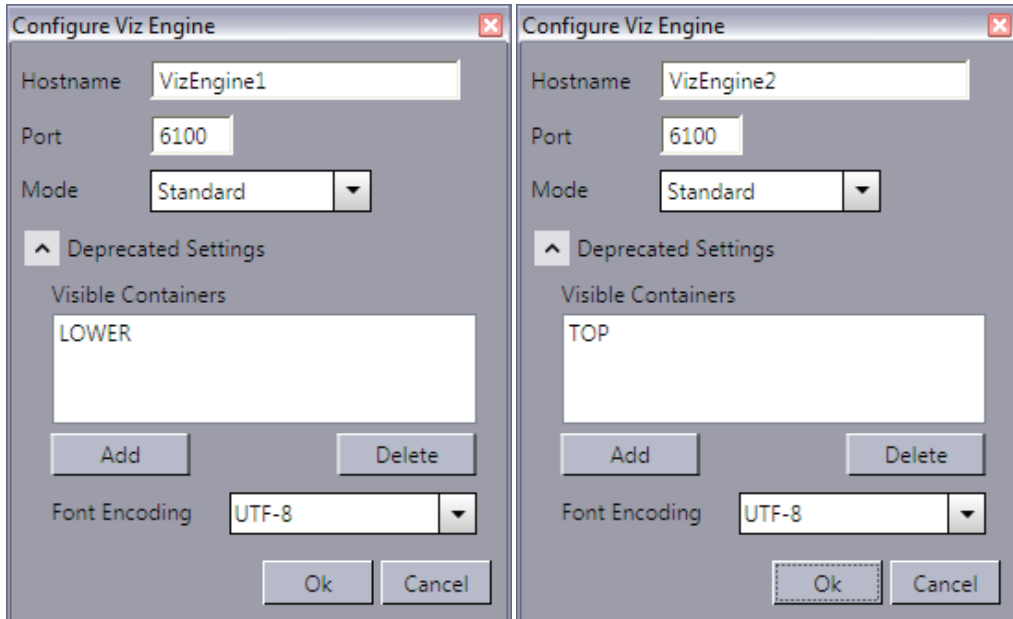
### To enable still preview



1. Create two channels, one for program and one for preview
2. Configure the same render engine twice as seen in how [To add a Viz Engine](#)
3. For the second render engine set the **Mode** to **Still Preview**
  - **Still Preview:** Allows you to use the program output to get a still preview. To achieve this the Media Sequencer creates a copy of the scene being read and sends commands to your program output renderer asking for a snapshot of the scene while the current scene on air is being rendered.
4. Click **Ok**
5. Add the program renderer to the program channel and the still preview renderer to the preview channel

-----  
**WARNING!** This setup requires sufficient ring buffer on the program renderer in order for the rendered still preview not to cause the scene on air to drop frames; hence, this setting is deprecated.  
-----

## To render specific scene containers



1. Configure the Viz Engine settings as seen in how [To add a Viz Engine](#)
2. Click the **Deprecated settings** button to expand the editor
3. Enter the scene's container name(s) that should be rendered visible to the [Visible containers](#) field.
4. *Optional:* Press **Enter** to add another container name
5. Click **Ok**

## To set a different font encoding

1. Configure the Viz Engine settings as seen in how [To add a Viz Engine](#)
2. Click the **Deprecated settings** button to expand the editor
3. Select the **Font encoding**
  - **Font encoding:** Sets the font encoding of the Viz Engine. The encoding can either be set to UTF-8 or ISO-8859-1. Default is UTF-8.
4. Click **Ok**.

## 8.2.5 Upgrading Old Profiles

If you are running Viz Trio versions prior to 2.10.1, and you want to reuse your current profile setup you need to be aware of the following facts:

When Viz Trio starts, it makes copies of the old profiles and give them a new name using the old name with a *tag* appended at the end:

```
<profile name> (Upgraded)
```

The **old profile** name will be kept as is, and can only be used by Viz Trio versions older than 2.10.1.

- Old profiles from older versions are disabled in the new profile configuration; however, you can delete them.
- Old profiles will not show up in the profile selector on the status bar, the intelligent interface configuration or the VDCP configuration.

- Old profiles have a tool tip with the name of the upgraded profile.

The **new profile** name (with the `Upgraded` tag) should be used by Viz Trio 2.10.1 and later.

- New profiles can be renamed without any problems.
- New profiles (both the upgraded ones and manually created ones) will be usable from old clients, but changes done will not have any effect.

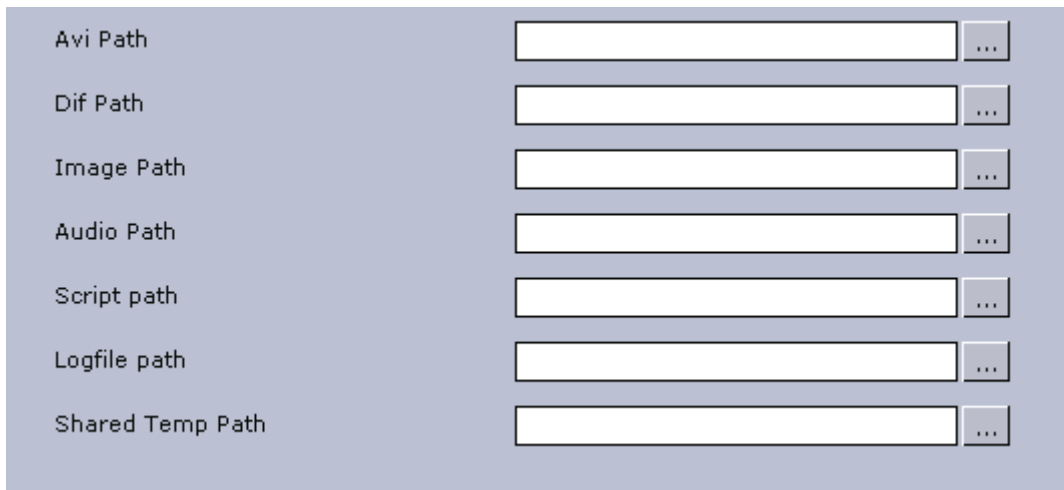
-----  
**Tip:** When you have upgraded all clients, the old profiles can be deleted.  
-----

---

## 8.3 Directories

The directories category contains path settings for several functions. All settings are local to the specific Viz Trio client.

**Figure 46:** Directories



Avi Path	<input type="text"/>	...
Dif Path	<input type="text"/>	...
Image Path	<input type="text"/>	...
Audio Path	<input type="text"/>	...
Script path	<input type="text"/>	...
Logfile path	<input type="text"/>	...
Shared Temp Path	<input type="text"/>	...

- **Avi Path:** Sets the default location when browsing for AVI files.
- **Dif Path:** Sets the default location when browsing for DIF files.
- **Image Path:** Sets the default location when browsing for Image files.
- **Audio Path:** Sets the default location when browsing for Audio files.
- **Script Path:** Sets the path to the directory Viz Trio will use as a temporary running directory for scripts. The scripts themselves are stored in the server's repository.
- **Logfile Path:** The log file path option can be set to specify the path to the directory where the Viz Trio client should store log files.
- **Shared Temp Folder:** Used for temporary files when importing or exporting Viz Trio shows on an external Viz Engine. This must be a network share that is both accessible for the client and the external Viz Engine.

---

## 8.4 Database (Viz 3.x)

The Database (Viz 3.x) category contains settings for logging onto the Viz Graphic Hub. All settings are local to the specific Viz Trio client.

**Figure 47:** Viz Graphic Hub connection

Nameserver Host	<hostname>
Database	VizDb
User Name	Guest
Password	*****
<input type="checkbox"/> Auto Login	

- **Nameserver Host:** Sets the hostname or IP address for the Viz DB server.
- **Database:** Sets the name of the database.
- **User Name:** Sets the user name used for logging on to the database.
- **Password:** Sets the password used for logging on to the database.
- **Auto Login:** Enables Viz Trio to log on automatically at startup.

**Note:** If the local Viz Engine's Auto Login is enabled, this section is disabled.

## 8.5 Viz Content Pilot Database

This section is used for Viz Content Pilot (VCP) specific integrations such as the database that holds all data elements and thumbnails used in playlists, and the Viz Object Store image database setup.

This section contains information on the following topics:

- [Database Settings](#)
- [Picture Database Settings](#)

### 8.5.1 Database Settings

The VCP Database pane is used to configure a VCP database connection.

**Note:** Before configuring a Viz Content Pilot (VCP) database connection, an Oracle 10g Runtime client must be installed.

**Figure 48:** Viz Content Pilot database settings

Viz Media Sequencer Connection String:	<user>/<password>@<hostname>/<SID>
Schema	PILOT

**Confirm**  
Do you want to reinitialize?

- **Media Sequencer Connection String:** The connection string field is used to configure a connection to the database. Specify the connection string using a TNS name alias or connection string.

- **TNS name alias:** <username>/<password>@<tns alias>
- **Connection string:** <username>/<password>@<hostname>/<SID>
- **Schema:** Enter the name of the Oracle schema.

#### To configure the Viz Content Pilot database connection

1. Click the **Config** button (upper left) to open the [Configuration Interface](#).
2. Select the **Database Settings** under the **VCP Database** category, and enter the database connection string and schema.
  - Example: pilot/pilot@10.10.10.1/VizrtDB.
  - Default schema: *PILOT*.
3. Click **Apply** to save the settings to the Media Sequencer.
4. Close the configuration window and check that Viz Trio's [Status Bar](#) shows a green status indicator for the database (cylinder).

-----  
**Note:** The database schema name should always be written in upper case.  
 -----

#### See Also

- [Picture Database Settings](#)
- [Viz Content Pilot User's Guide](#)

## 8.5.2 Picture Database Settings

The picture database is a connection to Viz Content Pilot's database that allows you to [Search Media](#) stored by Viz Object Store (VOS). In order to take advantage of VOS, two configuration steps are needed.

**Figure 49:** Picture database settings



- **Use Picture Database:** Enables or disables the VOS picture database.
- **Picture Database Connection String:** Sets the database connection string or TNS name alias.

#### To configure the picture database

1. Map the shared image drive used by Viz Object Store (VOS) on the Viz Trio and Viz Engine machines.
2. Enable the **Use Picture Database** option, and add the VCP database connection string.
3. Click **Apply**.

-----  
**Note:** If a TNS name alias is configured it can be used as a replacement for the hostname and SID in the connection string.  
 -----

## See Also

- [Database Settings](#)
- [Viz Content Pilot User's Guide](#)
- [Viz Object Store User's Guide](#)

---

## 8.6 Connectivity

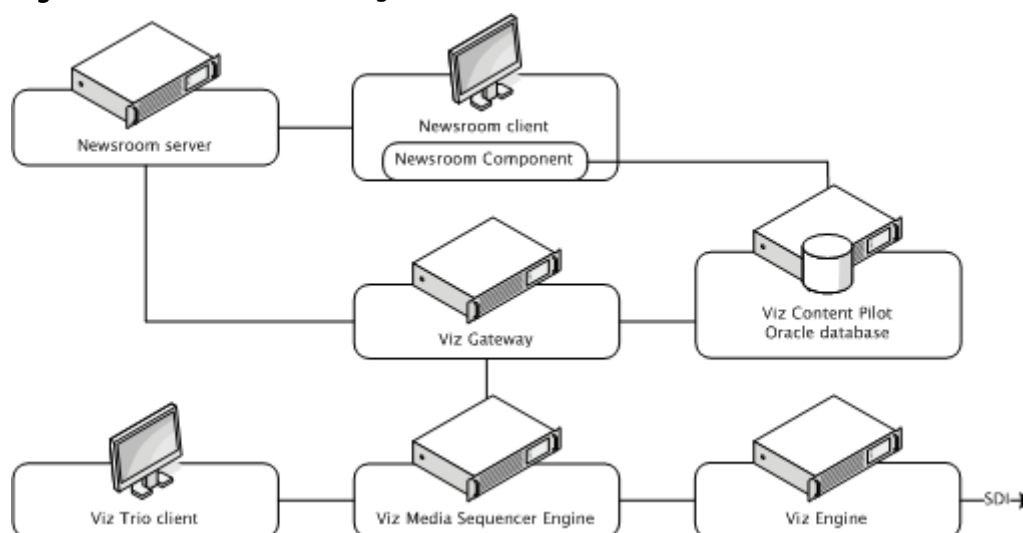
The connectivity section is used to configure different protocols that Viz Trio interfaces with. These are the Media Object Server (MOS) protocol, Intelligent Interface (MOS), General Purpose Input (GPI), Video Disk Communication Protocol (VDCP), Newstar MCU/AVS, and socket object settings that allows Viz Trio to act as client or a server in a socket connection.

This section contains information on the following topics:

- [MOS](#)
- [Intelligent Interface](#)
- [General Purpose Input \(GPI\)](#)
- [Video Disk Communication Protocol \(VDCP\)](#)
- [MCU/AVS](#)
- [Socket Object Settings](#)
- [Viz Video Hub](#)

### 8.6.1 MOS

**Figure 50:** MOS newsroom integration



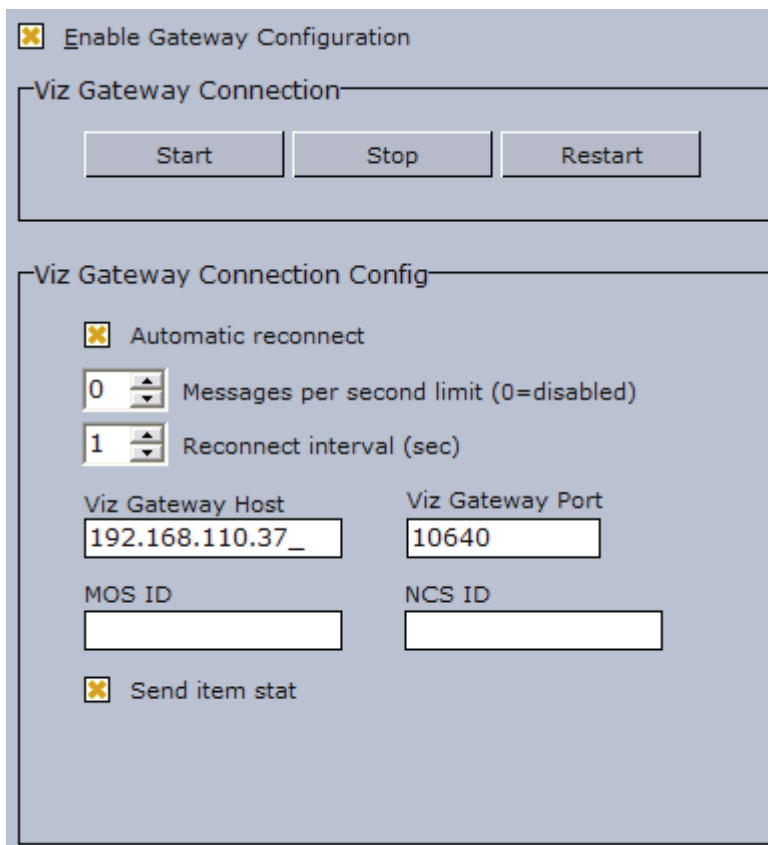
A Viz Gateway (i.e. MOS) connection allows Viz Trio to monitor playlists (see [Playlist Modes](#)) from any Newsroom Computer System (NCS) that supports the Media Object Server (MOS) protocol.

Newsroom Computer Systems (NCS) that support the MOS protocol can deliver newsroom playlists to Viz Trio through the Viz Gateway and Viz Content Pilot's database; hence, a MOS integration requires a Viz Gateway, an Oracle database and Viz Content Pilot's Newsroom Component.

To establish a MOS connection, the NCS and Viz Gateway must be pre-configured with an NCS and MOS ID. These ID's are set in the newsroom system.

**Note:** Most newsroom systems support the Media Object Server (MOS) protocol and/or the [Intelligent Interface](#) (IIF) protocols (i.e. Avid iNEWS ControlAir).

**Figure 51:** Viz Gateway, MOS.



- **Enable Gateway Configuration:** When selected, enables the user to configure the Viz Gateway connection. When enabled, an icon is also visible in the [Status Bar](#).
- **Viz Gateway Connection:** The controls in the Viz Gateway Connection section allows a start, stop, and restart action to be performed on a Viz Gateway connection.
  - **Start:** Starts the Viz Gateway connection.
  - **Stop:** Stops the Viz Gateway connection.
  - **Restart:** Restarts the Viz Gateway connection.
- **Viz Gateway Connection Configuration:** Define Viz Gateway connection settings.
  - **Automatic Reconnect:** When enabled, this option will automatically reconnect the Media Sequencer to Viz Gateway if one of them fails and/or is restarted.
  - **Message throttle: messages/sec:** Sets how many messages per second that should be sent from Viz Gateway to the NCS. This is done to prevent flooding of the NCS. 0 disables the message throttling.

- **Reconnect Interval (sec):** Specify the time in seconds for the Media Sequencer to wait before trying to reconnect to Viz Gateway when a connection is lost.
- **Viz Gateway Host:** IP address of the Viz Gateway host.
- **Viz Gateway Port:** Connection port for the Viz Gateway.
- **MOS ID:** Enter the ID of the connecting MOS device.
- **NCS ID:** Specify the ID of the newsroom control system.

The [Database Settings](#) for the [Viz Content Pilot Database](#) on the Media Sequencer must be established in order for the Viz Gateway integration to work.

---

**Note:** MOS and NCS ID are not needed for Viz Gateway versions 2 and newer.

---

### To configure a Viz Gateway connection

1. Click the **Config** button to open the [Configuration Interface](#).
2. Select the **MOS** section found under the **Connectivity** category, and enter Viz Gateway's IP address and port number (default port is *10640*).
3. Click Apply to save the settings to the Media Sequencer.
4. Close the [Configuration Interface](#) and check that Viz Trio's status bar shows a green status indicator for Viz Gateway (G).
5. Click the **Change Show** button, and select the **Playlists** tab to test that [Playlist Modes](#) are available.

---

**IMPORTANT!** It is required that Viz Trio sets the NCS and MOS IDs when connected to Viz Gateway versions prior to 2.0.

---

### See Also

- [Database Settings](#)
- [Ports and Connections](#)
- [Viz Gateway Administrator's Guide](#)
- [Viz Content Pilot User's Guide on Newsroom Component](#)



## 8.6.2 Intelligent Interface

**Figure 52:** Intelligent Interface Settings

Intelligent Interface Settings

Port: COM1

Baudrate: 9600

Verbose: no

Show path: Vizrt/Viz|Trio/TrioShow

Profile: BGO422W1

Encoding: ISO-8859-1

Loglevel:

II Status:  
uninitialized (no port)

The intelligent interface (IIF) configuration is used to set the parameters the automation system use to connect to the Media Sequencer over the intelligent interface protocol. Further it defines the Viz Content Pilot playlist or Viz Trio show the automation system can control and on which output profile.

With intelligent interface the Media Sequencer listens to a communications port and acts as the Character Generator (CG) device side of the intelligent interface for use with devices that use certain dialects of the protocol. The Media Sequencer expects the other side of the communications port to be connected to a newsroom system or similar system that knows how to talk one of the supported dialects of intelligent interface.

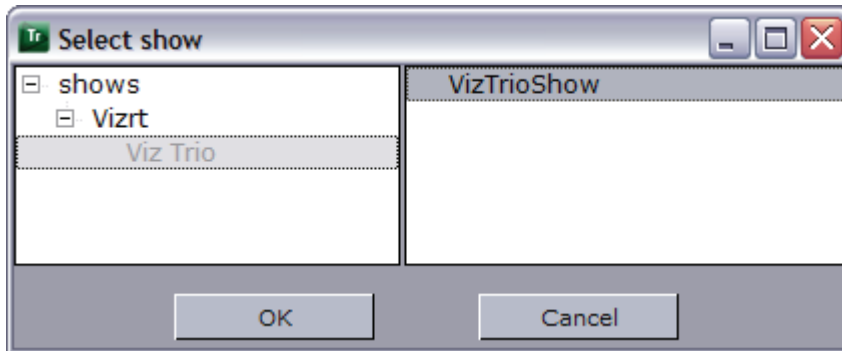
The Media Sequencer has two primary tasks:

- Receive callup data from a newsroom system and store it in the schedule.
- Trigger actions based on simulated keyboard commands it receives.

### Properties and Parameters

- **Port:** Sets the serial communications port to be used.
- **Baudrate:** Sets the appropriate baud rate for the connection.
- **Verbose:** Makes the messages from the Intelligent Interface driver being displayed in the Media Sequencer console. This feature is useful when setting up the system.

Figure 53: Select show



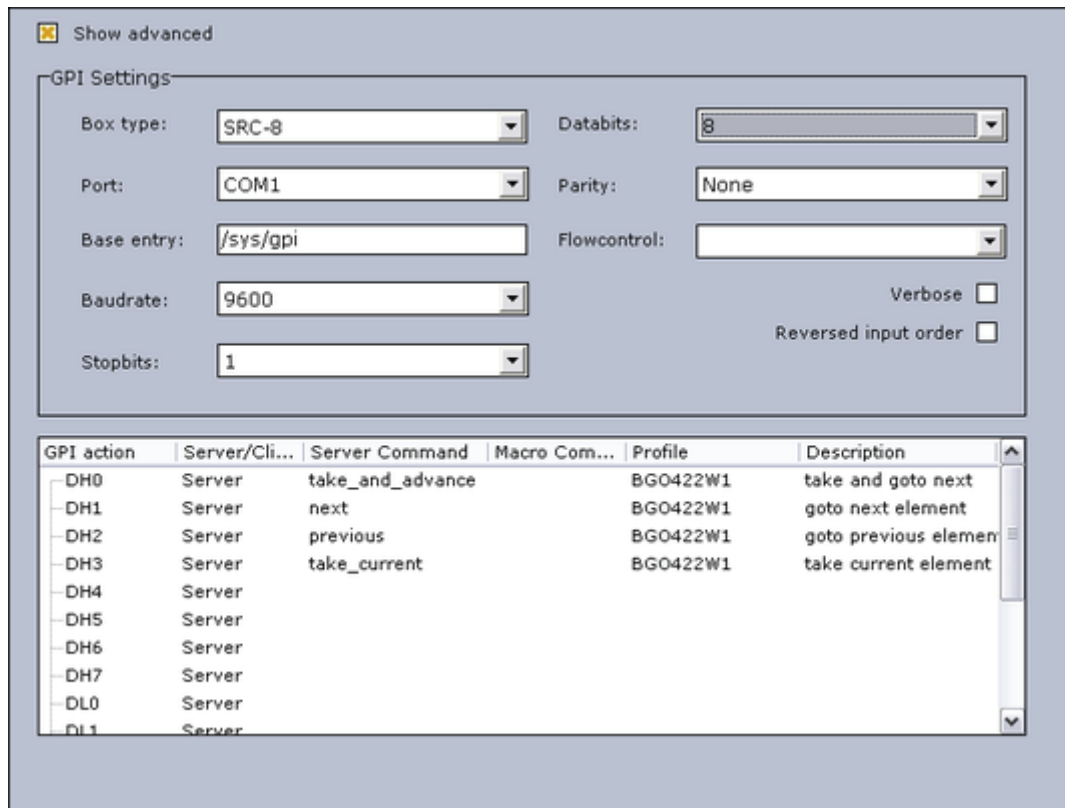
- **Show path:** Sets the path of the playlist or show.
- **Profile:** Sets the appropriate profile to be used.
- **Encoding:** Sets the appropriate font encoding for the connection.
- **Loglevel:** Sets the [Log Level](#) for the Media Sequencer.
- **Space means empty:** If set to **Yes**, the graphics template's default text (when creating an element) is replaced with a space. If set to **No**, the graphics element will show the default text unless it is manually changed and saved with no text.

#### See Also

- [Log Level](#)

### 8.6.3 General Purpose Input (GPI)

Figure 54: General Purpose Input



The GPI settings allows the Media Sequencer to be configured to handle GPI commands. The commands can be handled by the Media Sequencer itself (Server Command) or forwarded to the Viz Trio client (Macro Command).

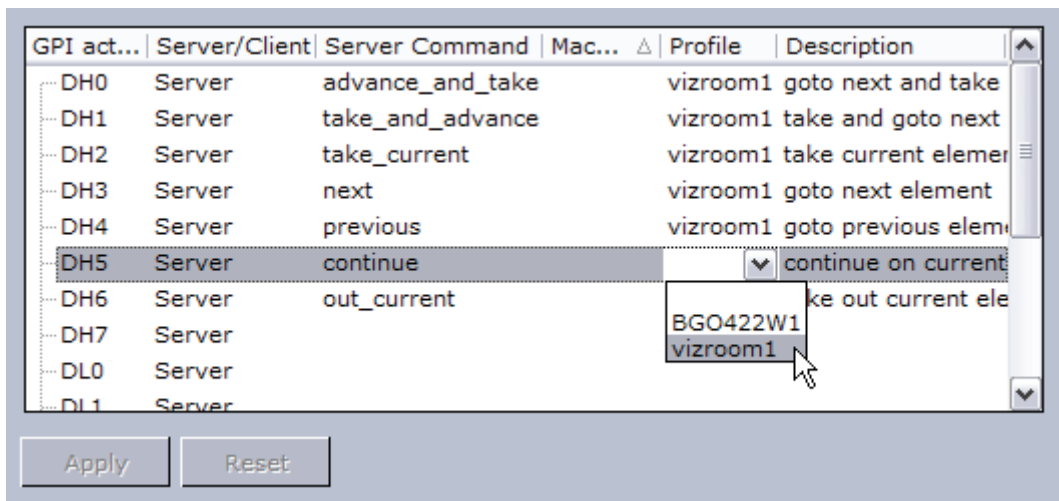
Both server and client side commands are profile specific (see [Output](#)), meaning that the profile determines which Viz Trio client and potentially which Viz Engine(s) that should execute a client and/or server command. It is therefore very important to assign the correct profile for each GPI action as they refer to unique profiles configured per client.

For example; A Viz Trio client has its own profile with two Viz Engines (program and preview) and receives a client command from a GPI on the Media Sequencer. The command executes some logic on the Viz Trio client that issues commands to the program renderer. If the correct profile is not set, such commands might end up on the wrong Viz Trio client and potentially on the wrong program renderer. The same would happen if the GPI action was defined as a server command; however, it would then trigger commands from the Media Sequencer to the Viz directly and not through the Viz Trio client.

- **Show advanced:** Displays all the settings in the GPI Settings frame.
- **Box Type:** Select the type of GPI box that is being configured. The Box Type can be set to SRC-8, SRC-8 III or SeaLevel.
- **Port:** Sets the port that the GPI box is connected to. The Port can be set to COM1-COM17, or None.

- **Base Entry:** This is the node in the Media Sequencer's data structure where the systems look for the GPI actions. The base entry is by default set to `/sys/gpi`.
- **Baudrate:** Sets the maximum rate of bits per second (bps) that you want data to be transmitted through this port. The Baud rate can be set to 110–921600. It is recommended to use the highest rate that is supported by the computer or device that is being used.
- **Stopbits:** Sets the interval (bps) for when characters should be transmitted. Stop bits can be set to 1, 1.5, or 2.
- **Databits:** Sets the number of data bits that should be used for each transmitted and received character. The communicating computer or device must have the same setting. The number of data bits can be set to 5, 6, 7 or 8.
- **Parity:** Changes the type of error checking that is used for the selected port. The communicating computer or device must have the same setting. The parity can be set to:
  - **Even:** A parity bit may be added to make the number of 1's in the data bits even. This will enable error checking.
  - **Odd:** A parity bit may be added to make the number of 1's in the data bits odd. This will enable error checking.
  - **None:** No parity bit will be added to the data bits sent from this port. This will disable error checking.
  - **Mark:** A parity bit set to 0 will be added.
  - **Space:** A parity bit set to 1 will be added.
- **Flowcontrol:** Changes how the flow of data is controlled. The flow control can be set to:
  - **None:** No control of data flow.
  - **XonXoff:** Standard method of controlling the flow of data between two modems. XonXoff flow control is sometimes referred to as software handshaking
  - **Hardware:** Standard method of controlling the flow of data between a computer and a serial device. Hardware flow control is sometimes referred to as hardware handshaking.
- **Verbose:** If enabled, Media Sequencer's GPI handler outputs log information. This information is useful for debugging.
- **Reversed Input Order:** Note that this check box is only available if Box Type is set to SRC–8. If enabled, the signal line that originally triggered GPI action DL0/DH0 will now trigger GPI action DL7/DH7, the signal line that originally triggered GPI action DL1/DH1 will now trigger GPI action DL6/DH6, and so on.

**Figure 55:** Commands and actions list



- **GPI action:** Shows a list of the available GPI actions.
- **Server/Client:** Shows a drop-down list box in every row, where the selected GPI action should apply to either the Media Sequencer (Server option) or the local Viz Trio client (Client option).

**Note:** The server and client actions are reciprocally exclusive.

- **Server Command:** Shows a drop-down list box in every row, where the action to be performed on this GPI line can be selected. Server commands are GPI actions that apply to the Media Sequencer. When right-clicking an item in a [Create Playlist](#) or [Playlist Modes](#), a context menu opens. In this menu, select Set External Cursor. A red arrow appears next to the selected element in the playlist, which indicates that this is the current GPI cursor. The server commands can be set to:
  - **advance\_and\_take:** The cursor shifts to the next element in the playlist, and then runs the Start operation.
  - **take\_and\_advance:** Runs the Start operation on the current element, and then shifts to the next element in the playlist.
  - **take\_current:** Runs the Start operation on the current element (the element with the cursor).
  - **next:** Shifts to the next element in the playlist.
  - **previous:** Shifts to the previous element in the playlist.
  - **continue:** Runs the Continue operation on the current element.
  - **out\_current:** Runs the Take Out operation on the current element.
- **Macro Command:** Macro commands are silent GPI actions. Clicking the ellipsis (...) button opens the Add Command window.
- **Profile:** Sets the profile to be used for the GPI action. This profile must match the profile set for the playlist that is to be triggered by the GPI actions. The drop-down list shows the profiles configured on the Media Sequencer.
- **Description:** Shows the description of the GPI action, as it was specified in the Add Command window.

This section contains information on the following topics:

- [To assign a Server Command](#)

- [To assign a Client Command](#)
- [To add a macro command](#)

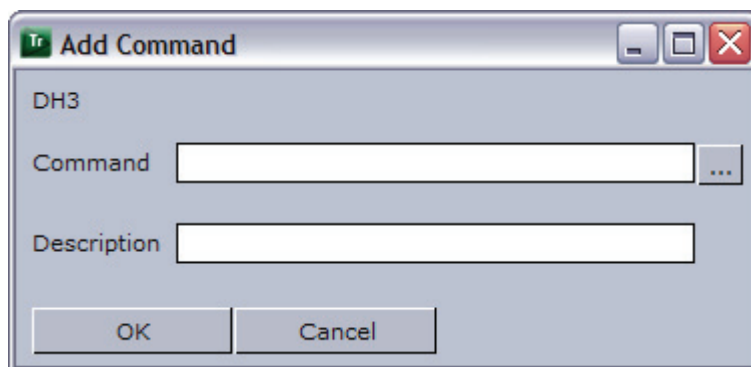
#### To assign a Server Command

1. Select a GPI action, and select **Server** in the Server/Client column.
2. Select a **Server Command**.
3. Select a **Profile**.
4. Click **Apply**.

#### To assign a Client Command

1. Select a GPI action, and select **Client** in the Server/Client column.
2. Select or create a **Macro Command**.
3. Select a **Profile**.
4. Click **Apply**.

#### To add a macro command



1. Select the **Macro Command** column, and click the small ellipse (...) button to open the Add Command dialog box.
2. Enter the command in the **Command** text field, or alternatively click the ellipse (...) button to open the [Predefined Functions](#) window.

#### See Also

- [Profile Setups](#)

## 8.6.4 Video Disk Communication Protocol (VDCP)

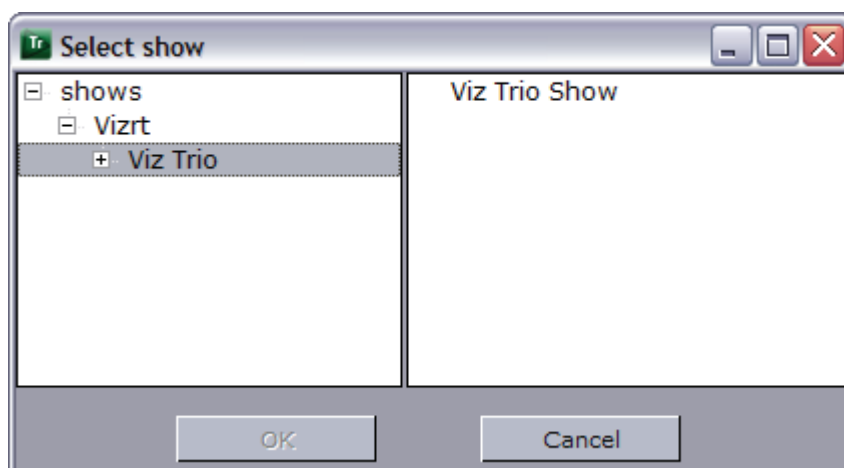


The Video Disk Control Protocol (VDCP) configuration allows a VDCP connection for the Media Sequencer to be established in order to externally control a Viz Content Pilot playlist or Viz Trio show.

With VDCP the Media Sequencer acts like a server that controls the graphics through the VDCP protocol. It sets up a serial connection, and on the other end of the connection typically a video controller is placed. Over this connection VDCP commands are sent, and in this way the video controller is able to control a playlist/show.

The configuration of the Media Sequencer is twofold. There are the general VDCP settings, and there is the configuration for which playlist to control.

The VDCP protocol defines recommended serial settings, but if you for some reason need to use different settings please refer to the Media Sequencer manual's VDCP section, and in particular the section on "Electrical and Mechanical Specifications", for information on how to configure this.



- **Port:** Select the appropriate COM port for the communication.
- **Profile:** Select the profile to use.
- **Select Mode:** Select a Viz Trio show or a Viz Content Pilot playlist path mode.
- **Trio path / Pilot playlist:** Sets the base directory for the VDCP integration. Video clips will be placed here.

## 8.6.5 MCU/AVS

Figure 56: MCU/AVS

MCU/AVS Settings

Port: COM2

Baudrate: 9600

Show path: Vizrt/Viz|Trio/TrioShow

Range begin: 1000

Range end: 2000

Offset: 100

Verbose: no

Replace delimiter: ,

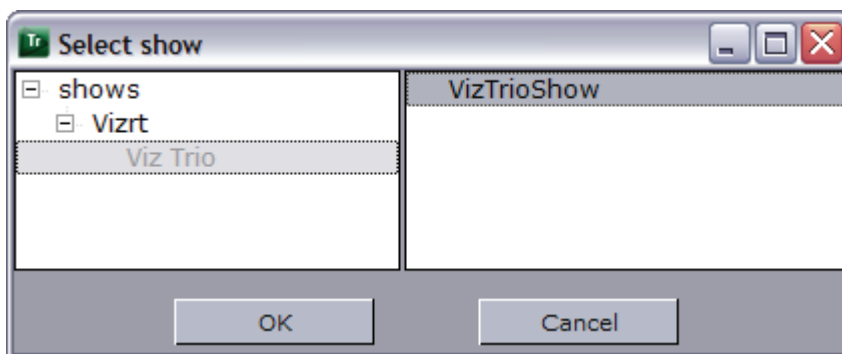
Super-delimiter: /

Ignore incorrect events

Unconditional deletes

This configuration is primarily used for configuring settings for a Newstar newsroom system connection.

- **Port:** Select the appropriate Com port for the communication.
- **Baudrate:** Select the appropriate baud rate.



- **Show path:** Base directory for the MCU/AVS integration. Newsroom stories will be placed here.
- **Range begin and Range end:** Pages received from the newsroom system will be numbered within this range.
  - Example: If Range Begin is 1000 and Range End is 5000 and Offset is 100. Then Callup pages in story 1 will be numbered 1000, 1001, 1002, and so on. Callup pages in story 2 will be numbered 1100, 1101, 1102, and so on.



- **Offset:** Sets the offset value that will be used to separate pages from different stores.
- **Verbose:** Enables notification messages from the MCU/AVS driver. The messages will show on the Media Sequencer console. This feature is useful when setting up the system.
- **Replace delimiter:** Replaces a character with the super-delimiter. For example if set to '|', slashes '/' can be put into tab-fields by entering '|'.  
|
- **Super-delimiter:** Delimiter used in News\* to separate tab-fields. Standard value is '|'.  
|
- **Ignore incorrect events:** Incomplete scripts from News\* will send invalid protocol data. Typically template specifications and supers could be missing. If this option is set to "Yes", then the system accepts incomplete/incorrect story descriptions. Should be set to "Yes" in most cases.
- **Unconditional deletes:** Removed in version 1.10. On older versions of Viz Trio, this option should be enabled. This means that a transfer of messages from Newstar starts with deleting existing messages within the specified range. To maintain a proper state handling this method must be used.

### 8.6.6 Socket Object Settings

Figure 57: Socket object settings

The Socket Object Settings section allows for socket connections to be defined for the Viz Trio client. It can either act as client or a server in a socket connection. A serial port connection which uses the same command set can also be set up.

- **Socket type:**
  - **Client socket:** When Viz Trio is set up to be a client socket it will connect to the specified server socket. Specify a host name and a port for the machine that runs the server socket.

- **Server socket:** When Viz Trio is set up to be a server socket it will wait for a client socket connection. Specify the port it should listen on.
- **Serial port (Com port):** The serial port connections work in principle as a socket connection, but instead of using TCP/IP it uses a null modem cable. The command-set is the same.
- **Text encoding:** Select a text encoding for the connection.
- **Serial port config:** If a serial port connection is to be set up the COM port number and the data baud rate must be specified here.
- **Autoconnect:** With this option set, a server socket or a serial port connection will be open/active at all times, but a client socket will first be active when data is sent. If this option is not checked, a “connect\_socket” command must be sent before sending the data.
- **Socket Host-** The socket server’s host name must be set here if the Viz Trio client is to function as a client socket.
- **Socket Port:** For client and server socket a port number must be specified here.
- **Data separator (escaped)-** Specifies a data separator for incoming data.
  - Receiving data separator will trigger the `OnSocketDataReceived` event in Scripts. If unprintable ASCII characters are used as data separators, special character sequences called “escape codes” are needed. All escape codes start with a backslash. For details, see [10.2.3 Escape Sequences](#).
  - When the socket object receives the data separator, the `OnSocketDataReceived` event is triggered, and the data received since the last separator is issued (not including the separator itself). If Data separator has no value, the `OnSocketDataReceived` is called continuously as long as data is received.

## 8.6.7 Viz Video Hub

**Figure 58:** Viz Video Hub Connectivity settings

	Host:	Port:
Main Messagebus Host:	<input type="text"/>	<input type="text" value="6555"/>
Backup Messagebus Host:	<input type="text"/>	<input type="text" value="6555"/>
<input checked="" type="checkbox"/> Same As Messagebus		
Main Soap Host:	<input type="text"/>	<input type="text" value="80"/>
Backup Soap Host:	<input type="text"/>	<input type="text" value="80"/>
Connection timeout (ms):	<input type="text" value="10000"/>	
Username:	<input type="text" value="quattro"/>	
Password:	<input type="text" value="*****"/>	

The Viz Video Hub section is used to apply Viz Video Hub settings to the Media Sequencer which will enable the Media Sequencer to initiate transfer of Viz Video Hub elements in Viz Trio shows and playlists to the Viz Engine.

The Media Sequencer's communication with Viz Video Hub is related to all Media Sequencer show and playlist elements that contain Viz Video Hub elements residing on the Viz Video Hub that are initialized for playback.

- **Main and Backup Messagebus Host:** Sets the message bus host.
- **Messagebus port:** Sets the message bus port.
- **Main and Backup SOAP Host:** Sets the SOAP host. Check the "Same as messagebus" option if the host is the same as for the message bus host.
- **SOAP port:** Sets the SOAP port.
- **Connection Timeout:** Sets the timeout to Viz Video Hub in milliseconds.
- **Username:** Sets the username for Viz Video Hub.
- **Password:** Sets the password for Viz Video Hub.

---

**Note:** If both main and backup message bus host are empty you will not have a configured Viz Video Hub system; hence, you will not see the status bar icon or receive error messages.

---

### See Also

- Viz Video Hub User's Guide
- Viz Link Administrator's Guide

---

## 8.7 Miscellaneous

The miscellaneous section describes various settings. The section spans from Viz Trio No Video (NV) configurations to setting of log levels. It also covers configurations of external preview and local preview, running of Viz Trio as a proxy for Viz Engine commands and user restrictions for the Viz Trio operator.

The following sections contain information on the following topics:

- [General](#)
- [Local Viz Renderer](#)
- [Viz Proxy](#)
- [User Restrictions](#)

### 8.7.1 General

Figure 59: General

Autoupdate External Preview

Single Machine Mode

Autoupdate cache for image browser

Disable initial loading of icons in the pooleditor

Reload the current page if it is changed in the MSE (from another client or II)

Log level

Script Loglevel

Onair Password

Simple Viz Curious Maps Editor

Create page icons from preview when saving Trio pages.

Ignore Nameless tabfields during read

Help file path  ...

- **Autoupdate external preview:** When selected, the external preview channel will be automatically updated when a page is edited in the Viz Trio client. When not selected, the external preview channel will only update its preview when the page is saved.
- **Single Machine Mode:** This should be set on Viz Trio NV (no video) systems where the machine both functions as client- and output machine. This is to separate the control commands that goes to the output machine from those going to the local preview.
- **Autoupdate cache for image browser:** When enabled (default), the Viz Trio client will update the local image browser cache automatically and add any images that is new and update those who have changed. With very large image trees this update process might slow down the system response. To achieve better system response when the image tree is very large, disable this option.

- **Disable initial loading of icons in the pooleditor:** When enabled, this setting will prevent the Viz pool browser from being loaded the first time a tab field, with for example Viz geometries or images, is selected. The icons are loaded when clicking in the browser or changing the path.
- **Reload the current page if it is changed on the MSE (from another client or II):** If this setting is on, the currently read page is reloaded (read again) if it changes on the server.
- **Loglevel:** Defines what kind of log messages that should be logged. Possible levels are: 0, 1, 2, 5, and 9.
  - **Loglevel 0:** Only error messages will be logged.
  - **Loglevel 1:** Warning messages are added.
  - **Loglevel 2:** Timer messages are added. These show how long time different operations take in the program.
  - **Loglevel 5:** Status messages are added. These show ordinary program events such as “page loaded”, “Page taken” etc.
  - **Loglevel 9:** All commands sent to the Viz Engine rendering process are added.
- **Script Loglevel:** Sets which log level to use when running scripts. It defaults to -1 , which is no logging. So if a script triggers a viz command, the script log level must be set to 9 to see the Viz Engine command in the log file.
- **Onair Password:** Sets an on-air password. When set, all users will be prompted for the password when trying to take a Viz Trio client off- or on-air.
- **Simple Viz Curious Maps Editor:** When enabled, Viz Trio uses the simple Viz Curious Maps Editor with reduced functionality for map selection.
- **Create page icons from preview when saving Trio pages:** When enabled, this setting will create page icons (thumbnails) from the local preview when creating pages of templates.
- **Ignore Nameless tabfields during read:** When enabled, this setting will discard error messages for tab fields with no name.
- **Help file path:** Sets the path for the help file. By default this is the HTML version of the help file. The help file is opened by clicking the Help button in the user interface (next to Config), or by pressing the F1 key unless it has been assigned to another action (e.g. macro).

## 8.7.2 Local Viz Renderer

Figure 60: Local Viz Renderer

General

Renderer Force Sleep

Viz Engine Connection Timeout

Font Encoding

Single Layer Transition Logic Preview

Use Viz for video-preview

Local Preview Resolution

PAL (720x576)

NTSC (720x480)

16:9 (720x405)

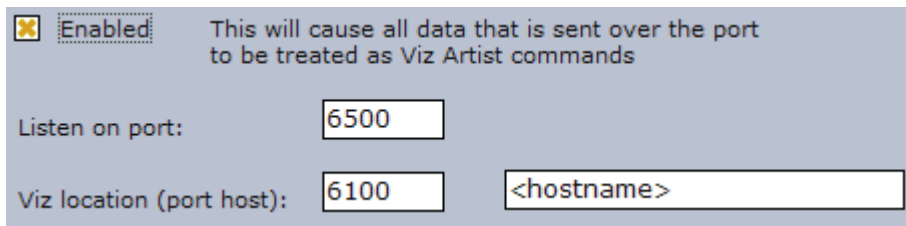
Custom (WxH)   x  Refresh Rate:

- **Renderer Force Sleep:** Check this option if a NVIDIA G-Force graphics card is installed on the Viz Trio Client machine. Otherwise the machine might, in some cases, use a lot of CPU power.
- **Viz Connection Timeout:** Sets the number of milliseconds until the local Viz Engine renderer times out. Changes to this setting requires a restart of Viz Trio.
- **Font Encoding:** Sets the font encoding format of text sent to the local preview. This setting overrides the local Viz preview and requires a restart of Viz Trio to take effect. Note that this does not have an effect on remote Viz Engines; hence, the font options for Viz must be set accordingly. Also note that the Viz Trio interface only supports the font encoding set for the operating system under the *Regional and Language Options* settings (i.e. support for complex script, right-to-left and East Asian languages).
- **Single Layer Transition Logic Preview:** Enable previewing of single-layer pages in the background scene. If this setting is disabled, only Combo pages will be previewed in the background scene.
- **Use Viz Engine for video preview:** Enables Viz Engine (default) to stream video clips from Viz Video Hub and play them for local preview instead of using a standard Microsoft Media Player. Changes requires a restart of Viz Trio.

In the *Local Preview Resolution* section the aspect ratio for the local preview window can be set, and the systems refresh rate defined (normally 50 for PAL and 60 for NTSC).

### 8.7.3 Viz Proxy

Figure 61: Viz Proxy



**Enabled** This will cause all data that is sent over the port to be treated as Viz Artist commands

Listen on port:

Viz location (port host):

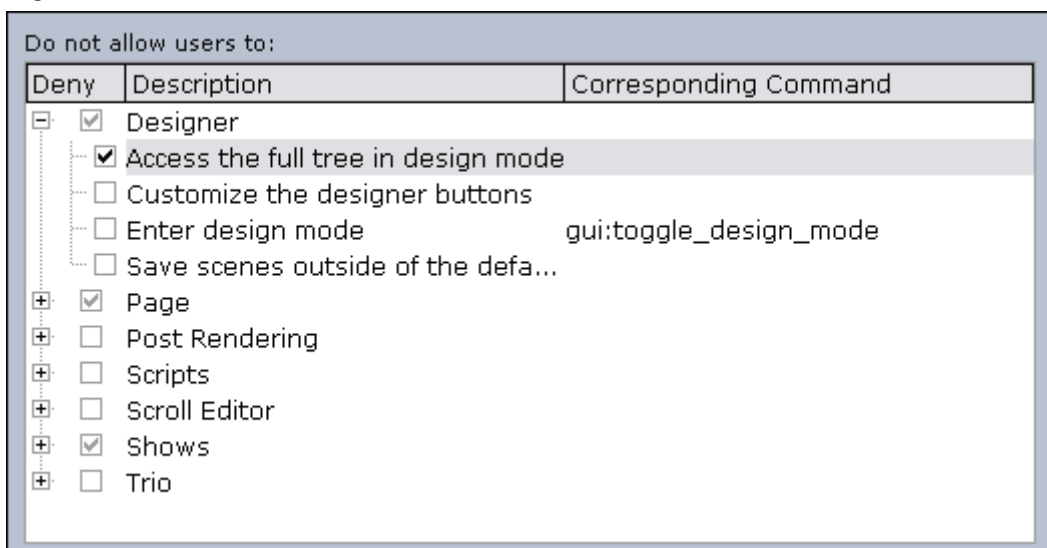
The proxy function is mostly relevant when Viz Trio and Viz Engine are set up with a direct network link to be controlled by an external control application. A dedicated network card on the Viz Trio client together with a crossed network cable directly to Viz is the normal way to make such a link.

All data sent over the configured listener port will be treated as Viz commands and forwarded by the Viz Trio client to Viz. With the proxy function enabled, the data sent over the configured listener port will be treated as Viz commands by the Viz Trio client. Viz Trio does this by acting as a proxy transmitting commands back and forth between Viz and the external control application.

- **Enabled:** Enables the proxy feature.
- **Listen on port:** Sets the listener port for Viz Trio that the sender will use.
- **Viz location (port host):** Sets the listener port for Viz Engine and the hostname (default is 6100 and localhost).

### 8.7.4 User Restrictions

Figure 62: User restrictions



Deny	Description	Corresponding Command
<input checked="" type="checkbox"/>	Designer	
<input checked="" type="checkbox"/>	Access the full tree in design mode	
<input type="checkbox"/>	Customize the designer buttons	
<input type="checkbox"/>	Enter design mode	gui:toggle_design_mode
<input type="checkbox"/>	Save scenes outside of the defa...	
<input checked="" type="checkbox"/>	Page	
<input type="checkbox"/>	Post Rendering	
<input type="checkbox"/>	Scripts	
<input type="checkbox"/>	Scroll Editor	
<input checked="" type="checkbox"/>	Shows	
<input type="checkbox"/>	Trio	

The User Restrictions section allows an administrator to deny the operator to perform certain actions in Viz Trio.

- **Designer:**
  - **Access the full tree in design mode:** When checked, prevents the operator from accessing the Full Tree view when using Viz Trio Designer.

- **Customize the designer buttons:** When checked, disables the option to customize and create new resource buttons.
- **Enter design mode:** When checked, disables the Design button (upper right) preventing the operator from going into Designer mode.
- **Save scenes outside of the default vizpath:** When checked, prevents the operator from saving scenes outside the default Viz scene path set under [Show Settings](#).
- **Page:**
  - **Delete Templates:** When checked, prevents the operator from deleting templates from the show. Command: `show:delete_templates`.
  - **Direct Takeout Pages:** When checked, prevents the operator from issuing a direct take out command on pages in a show. This also disables the context menu option for the show. Command: `page:direct_takeout`.
  - **Direct Takeout Pages from a Play List:** When checked, prevents the operator from issuing a direct take out command on pages in a playlist. Command: `playlist:direct_take_out_selected`.
  - **Save values to linked databases:** When checked (default), prevents the operator from saving values back to the database.
  - **Takeout Pages:** When checked, prevents the operator from using the Take Out command. This also disables the Take Out button in the Page editor. Command: `page:takeout`.
- **Post Rendering:**
  - **Enter new hostnames in post render view:** When checked, prevents the operator from adding new hostnames to the Render Videoclip editor.
- **Scripts:**
  - **Edit scripts:** When checked, prevents the operator from editing the script using the Edit Script button in the Page editor. Command: `gui:show_scripteditor`.
- **Scroll Editor:**
  - **Create new scroll templates:** When checked, prevents the operator from creating new scroll templates. Command: `gui_show_scroll_template_creator`.
- **Shows:**
  - **Access show settings:** When checked, prevents the operator from accessing the Show settings. Command: `gui:show_settings`.
  - **Browse external playlists:** When checked, disables the [Playlists](#) tab under [Show Control](#), and consequently preventing the operator from browsing for playlists on the Media Sequencer.
  - **Browse viz directories when changing shows:** When checked, disables the [Viz Directories](#) tab under [Show Control](#), and consequently preventing the operator from browsing the Viz directory to set a show path.
  - **Create or rename shows:** When checked, prevents the operator from creating new [Show Types](#) or renaming existing shows.
  - **Delete all pages in a show:** When checked, prevents the operator from selecting and deleting all pages in the show in one operation. Normal page by page and page group deletion is still possible.

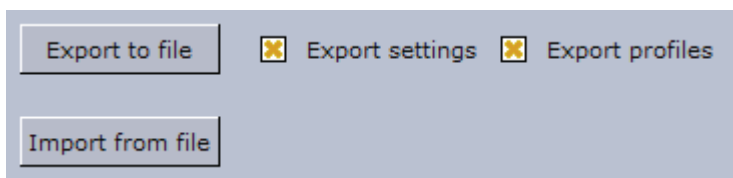


- **Open the window for importing templates into a show:** When checked, prevents the operator from importing new scenes from Viz Graphic Hub (Viz 3.x) or the data root (Viz 2.x).
- **Trio:**
  - **Call Viz Trio Commands:** When checked, prevents the operator from manually calling Trio Commands. This does not prevent scripts from doing the same. Command: `gui:show_triocommands`. See [Working with Macro Commands](#).
  - **Cleanup External Renderers:** When checked, prevents the operator from issuing a cleanup renderer command. Command: `trio_cleanup_renderers`.
  - **Switch to Viz Artist:** When checked, prevents the operator from switching to Viz Artist mode. Command: `gui:artist_mode`.

---

## 8.8 Import/Export Settings

**Figure 63:** Export and import



The Import and Export Settings function is used to import and export the configuration settings from and to an XML-file. This allows customized settings to be applied to one or more Viz Trio clients without manually configuring the settings each time.

**Export:** Enables the export of Viz Trio settings and profile configurations to an XML file. The following settings cannot be exported:

- [Keyboard Shortcuts and Macros](#) settings (can be exported separately).
- Word replace settings under the [Text Editor](#) settings (can be exported separately).
- [Viz Content Pilot Database](#) settings.
- All [Connectivity](#) settings.

**Import:** Enables the import of Viz Trio settings and profile configurations from an XML file. Note that such an operation will overwrite existing settings and profiles, and that Viz Trio must be restarted for the new configurations to take effect.

---

## 8.9 About

Figure 64: About Viz Trio



The about information shows which version and build of Viz Trio has been installed on your machine. When reporting issues to Vizrt you should always attach this information.

---

## 9 Scripting

Scripts in Viz Trio can be stored in two ways; on the Media Sequencer on a per show basis, or as files on a drive (preferably shared). A show script is only accessible to the machines connected to the same Media Sequencer using the same show. Scripts can be assigned to templates and shows. Note that it is only possible to assign one script per show or template; however, you can include other scripts as part of the “main” script in order to extend its functionality.

All edits to a script is done using the Viz Trio [Script Editor](#). Changes made to a show script will only affect the selected show and those clients that control the same show. Changes made to a script file on a shared script repository will have an affect on all shows that use the same script.

---

**Note:** File scripts are read into the show each time a show is opened.

---

Only templates can have scripts assigned. All instances of a template will inherit the template script.

This section contains information on the following topics:

- [Viz Trio Scripting](#)
- [Viz Template Wizard Scripting](#)

### See Also

- [Directories](#)
- [Macro Language](#)
- [Macro Commands and Events](#)

---

**Tip:** Always remember to escape backslashes correctly.

---

---

### 9.1 Viz Trio Scripting

In Viz Trio a VBScript can be attached to any show or template. This enables custom functionality to be implemented for that show or template. Typical applications for scripting can be, import of data from database sources, guidance to users, and so on. All template instances with a script will inherit the script.

Viz Trio supports a set of [Macro Commands and Events](#) that may be used as part of the script.

This section contains information on the following topics:

- [Script Directory](#)
- [Script Editor](#)
- [Script Backup](#)

## See Also

- [Template List](#) and the Script context menu options
- [Page List](#) and the Script context menu options
- [Directories](#) settings for local or shared script path configuration
- The [dblink](#) macro commands for use with scripting

### 9.1.1 Script Directory

All show scripts are stored on the Media Sequencer. However, a local directory is still needed for script files that are stored as files. Open the Configuration window, and set the *Script path* under the [Directories](#) section.

### 9.1.2 Script Editor



- **Predefined Functions** – Opens the [Predefined Functions](#) window which contains a set of [Commands](#) and [Events](#) that can be used in the script(s).
- **Show Library Scripts** – Opens the [Library Scripts](#) window displaying scripts which can be attached to a template as a 'USEUNIT reference.
- **Syntax check** – Checks the syntax to verify that the it is correct.
- **Save script** – Saves a script to the server. It is also possible to save the script by right-clicking within the script window and choose *Save to repository* from the appearing [Context Menu](#). On this menu an option that lets units to be saved to a file can also be found.
- **Close unit** – Closes the selected script unit.
- **Manage show scripts** – Opens the [Script Manager](#) for the currently open show.
- **Close all units** – Closes all script units.
- **Change font** – Changes the font settings for the script editor.
- **Combo box** – Jumps between script functions and procedures within the same script.

This section contains information on the following topics:

- [Context Menu](#)
- [Shortcut Commands](#)
- [Predefined Functions](#)
- [Library Scripts](#)
- [Script Manager](#)
- [To assign a script to a show](#)
- [To assign a script to a template](#)
- [To add a predefined function](#)
- [To add a library script unit](#)
- [To edit a show script](#)
- [To edit a script](#)
- [To execute a script](#)

## Context Menu

73	<b>If Not</b> text = "" <b>Then</b>	
74	New edit window	ches a search string
75	Insert from file...	ery
76	Load from file...	ActiveConnection = connVizrt
77		andText = "SELECT * " & _
78	Undo           Ctrl+Z	"FROM dbTable " & _
79	Redo           Ctrl+Y	"WHERE headline LIKE '%" & Replace(txtS
80	Select All      Ctrl+A	"OR subheadline LIKE '%" & Replace(txtS
81		"OR designation LIKE '%" & Replace(txtS
82	Cut            Ctrl+X	andType = adCmdText
83	Copy           Ctrl+C	sVizrtStory = .Execute
84	Paste          Ctrl+V	
85		
86	Save to repository/trl+S	ts all database entries
87	Toggle Bookmarks	ery
88	Goto Bookmarks	ion = connVizrt
89		LECT * " & _
90	Find...        Ctrl+F	Table "
91	Replace...	andText
92	Use script unit...	.Execute
93		
94	Print...	
95	Close unit	connection was successful.
96	Manage Show Scripts	adStateOpen <b>Then</b>
97		ot opened successfully.")
98	MsgBox ("Re	
99	<b>End If</b>	

- **New edit window** – Opens a new edit window.
- **Insert from file** – Inserts script from file.
- **Load from file** – Loads script from file in a new tab window.
- **Undo** – Undoes latest change.
- **Redo** – Redoes latest undo.
- **Select All** – Selects the whole script.
- **Cut** – Cuts selected text.
- **Copy** – Copies selected text.
- **Paste** – Pastes clipboard content.
- **Save to repository** – Saves the script.
- **Toggle Bookmarks** – This option toggles between inserting and removing a bookmark at/from the currently selected row.
- **Go to Bookmarks** – Moves the cursor to the selected bookmark.
- **Find** – Opens a search window. Search for text strings and expressions.
- **Replace** – Opens a search and replace window. Search for a string to be replaced by another.
- **Use script unit** – Opens a window with a list of accessible units. Select one to use with another script or script unit.
- **Print** – Opens a print window.
- **Close unit** – Closes the unit

- **Manage show scripts** – Opens a dialog box to allow for import, export, and deletion of script from a template.

### Shortcut Commands

- **CTRL + TAB** – Switches between the open scripting tabs.
- **CTRL + SPACE** – Displays the Code Insight Window displaying a list of parameters, variables, functions, and so on available to the currently open script or in attached sub scripts.
- **CTRL + SHIFT + SPACE** – Displays the code completion hints.
- **CTRL + Left mouse button** – Opens the highlighted function (works across units). This works across scripts and scripts units, but not within the same script. For this to work the script needs to add the *'USEUNIT <scriptname>* notation in order to link the scripts together.

### Predefined Functions



Almost all script functions in Viz Trio are macro **Commands** that can be wrapped in the *TrioCmd("macro\_command")* format. The predefined functions window also makes available a set of **Events** that can be added to the script(s).

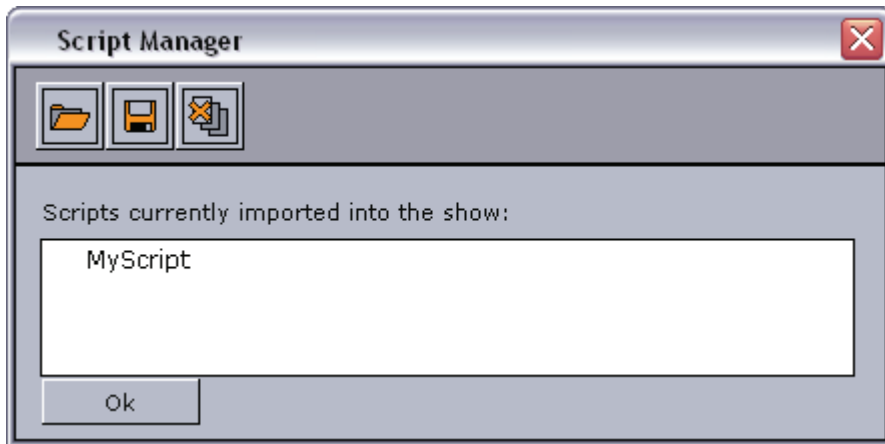
Note that `TrioCmd()` supports UTF16. If you need UTF8 support, you can use `TrioCmdUTF8()`.

### Library Scripts



Script units can be added to an assigned template script. This opens up the possibility to write generic and specialized script units that are useful for more than one template. When using script files a local or shared script path must be configured under the **Directories** settings.

## Script Manager



The Manage Show Scripts button opens a window where it is possible to import, export, and delete VBScript files from the current show.

### To assign a script to a show

1. Open the [Show Settings](#).
2. Click the browse button next to the Show Script field.
3. From the Choose Showscript window select either a file script (local or shared) or a Media Sequencer script.
4. Click OK.

### To assign a script to a template

1. Right-click on a template and select **Script > Assign script > New**.
2. *Optional:* Assign an existing script from the **File** or **Show** options.
3. Enter a **name** for the script, and select to store the script on the Media Sequencer or to a local or shared script path (see [Directories](#)).
4. Click OK.

### To add a predefined function

1. Open a template with an assigned script.
2. Click the **Edit script** button (see [Controls](#)).
3. Click the **Predefined functions** button in the [Script Directory](#).
4. Select a function, and click **Add**.

### To add a library script unit

1. Open a template with an assigned script.
2. Click the **Edit script** button (see [Controls](#)).
3. Click the **Library scripts** button in the [Script Directory](#).
4. Select a script unit to use, and click **OK**.

### To edit a show script

- Right click the Template list pane and from the appearing context menu select **Script** and **Edit Show Script**.

### To edit a script

- Open a template with an assigned script, and click the **Edit Script** button (see [Controls](#)).

### To execute a script



- Open a template with an assigned script, and click the **Execute Script** button (see [Controls](#)).

## 9.1.3 Script Backup

It is recommended that scripts created for a specific show is saved with that show on the Media Sequencer. Thus, a backup of the Media Sequencer files are necessary.

### See Also

- [Media Sequencer](#)

---

## 9.2 Viz Template Wizard Scripting

In Viz Trio, a Viz Template Wizard template can be used to further extend a show's functionality.

This section contains information on the following topics:

- [Dynamically adding components](#)
- [Set and get component values](#)
- [Set and get show values](#)

### Dynamically adding components

Viz Trio has support for the `CreateVTWComponent` function in Viz Template Wizard for dynamically adding components at run-time.

The following is an example where a text box and a button is used to generate a label at run-time.

```
curY = AddButton.Top + AddButton.Height + 5

Sub AddButtonClick(Sender)

    lbl = CreateVTWComponent("TTWUniLabel", Sender.Parent)
    lbl.Parent = Sender.Parent
    lbl.Left = Sender.Left
    lbl.Top = curY
    curY = curY + lbl.Height + 2

SetUnicodeValue lbl, GetUnicodeValue(SampleEdit)

End Sub
```



## Set and get component values

Viz Trio has support for the `SetUnicodeValue` and `GetUnicodeValue` functions in Viz Template Wizard.

The following is an example of a text box being used to get and set a value in another text box without using `TrioCmd`.

```
Sub TWUniEdit1Change(Sender)
    Text = GetUnicodeValue(TWUniEdit1)
    SetUnicodeValue TWUniEdit2, Text
End Sub
```

## Set and get show values

Using Viz Template Wizard to create standard templates for a show is quite useful as it enables the show to execute default commands for an entire show. It is therefore possible to set and get show values; however, there are some subtle differences in how to achieve both.

For example, when issuing a command such as `TrioCmd("page:read 1000")` within a Viz Template Wizard template, the page numbered 1000 will be read and previewed.

However, in order to return (get) values a command must be properly triggered by another event because all Viz Trio commands are queued; hence, the return value will be `QUEUED`. When a top-level command is executed (from the GUI or a macro) it is added to an internal queue and executed after other queued commands are finished.

In order to get return values the code using `TrioCmd()` must be issued by another top-level command. In a VTW template this is achieved by adding Viz Trio commands to events.

```
Function OnMyButtonClick(Sender)
    TrioCmd("page:read 1000")
    TrioCmd("vtwtemplate:run_vtw_script GetDescription")
    TrioCmd("page:read 1100")
    ...
End Function

Function GetDescription()
    returnvalue = TrioCmd("page:getdescription")
    msgbox returnvalue
End Function
```

In the above example the second command `vtwtemplate:run_vtw_script` will be triggered within `GetDescription` and return the description value.

## See Also

- [Show Settings](#) and the **Active VTW template** setting
- [Viz Template Wizard User's Guide](#)
- The [dblink](#) macro commands for use with scripting

---

# 10 Macro Language

With the Viz Trio macro language it is possible to script many of the operations that are normally done in the graphical user interface. The general syntax is:

```
Command [argument]
```

Some commands take several arguments, for instance the command scaling that must have both the x-, y- and z-axis specified. See the command list for reference.

Macro commands can be used in three ways: As part of a shortcut key using the [Keyboard Shortcuts and Macros](#) window, as part of a script using the [Script Editor](#) or as part of an external application where commands are executed over a socket connection.

When macros are used, it is often desirable to use commands that normally would trigger a dialog for user input such as *page:delete 1000*. The latter example would normally ask a user to confirm whether to delete the page or not. To avoid this issue it is possible to set the following modes:

- **gui:set\_silent\_mode** – Sets silent mode which controls whether to show dialogs to the user or not.
- **gui:set\_interactive\_mode** – Sets interactive mode which controls whether to show dialogs to the user or not.

This section contains information on the following topics:

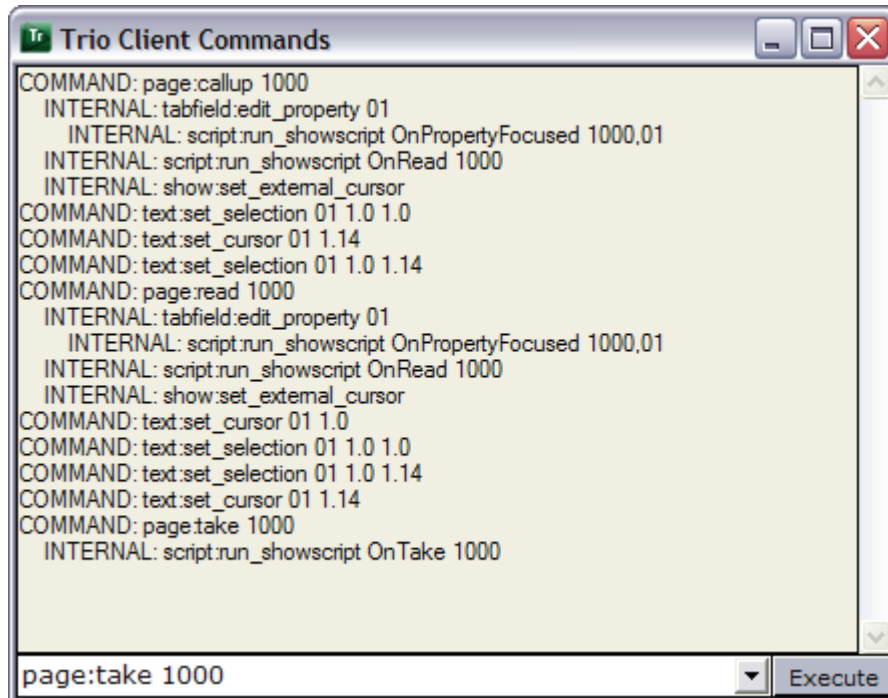
- [Working with Macro Commands](#)
- [Working with Macro Commands over a Socket Connection](#)
- [Working with Shortcut Keys](#)

## See Also

- [Scripting](#)
- [Macro Commands and Events](#)

## 10.1 Working with Macro Commands

Figure 65: Trio Client Commands window



The Viz Trio user interface basically uses the same macro commands as the scripting support uses. To help find the commands that are executed when different user interface operations are performed, use the commands window and look at which commands are sent.

The actual commands are what can be seen after the colon.

```
page:read 1000
```

The Trio Client Commands window can be very helpful when learning the system and the macro commands. Enter commands in the text field at the bottom. Use this to test a customized command, and click the Execute button to run it.

### To enable the Trio Client Commands button

- Click Viz Trio's **Config** button, and in the **Miscellaneous > User Restrictions** section, clear the **Call Viz Trio Commands** check box.

### To open the Trio Client Commands window

- Click the **Trio Commands** button in the lower-right corner of the application window.

This section contains information on the following topics:

- [Apropos and Help Commands](#)
- [Show, Context and Tab-field Commands](#)

## 10.1.1 Apropos and Help Commands

The Trio Client Commands window has some features that is helpful when searching for information about a command.

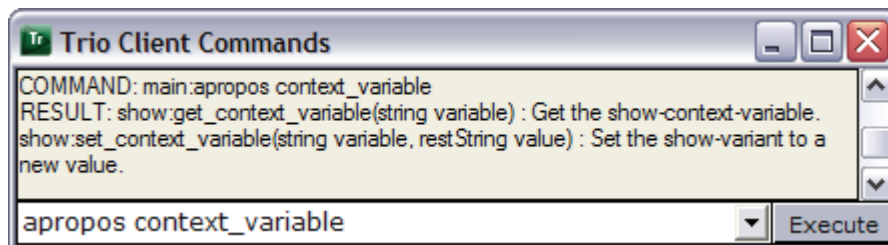
The keywords **apropos** and **help** can be used to do lookups in the available set of commands.

- **apropos** – Searches command names and descriptions for a string.
- **help** – Shows help for the specified command, or lists commands.

This section contains information on the following topics:

- [Example I: Apropos](#)
- [Example II: Help](#)

### Example I: Apropos

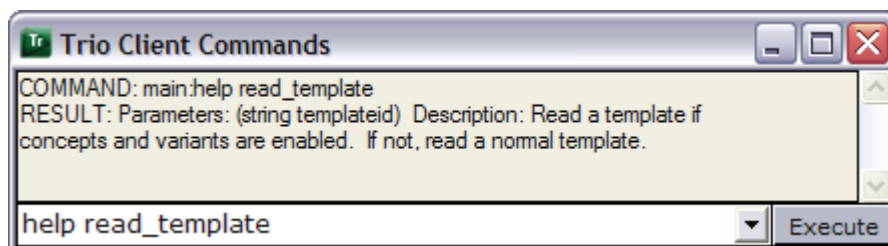


```
COMMAND: main:apropos context_variable
```

```
RESULT: show:get_context_variable(string variable) : Get the show-  
context-variable.
```

```
show:set_context_variable(string variable, restString value) : Set the  
show-variant to a new value.
```

### Example II: Help



```
COMMAND: main:help read_template
```

```
RESULT: Parameters: (string templateid) Description: Read a template if  
concepts and variants are enabled. If not, read a normal template.
```

## 10.1.2 Show, Context and Tab-field Commands

Show variables are global variables as all variables are stored on the [Media Sequencer](#). Meaning that all Viz Trio applications using the same MSE can set and get the variables, and any script within a show can access the variable.

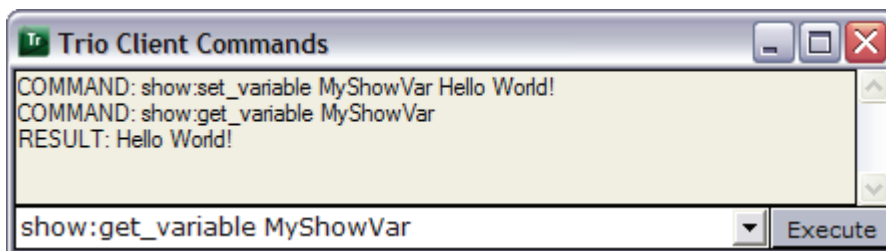
This section contains information on the following topics:

- [Show Variables](#)
- [Concept, Context and Variant Variables](#)
- [Commands](#)
- [Tab-field Variables](#)

### See Also

- [Macro Commands and Events](#)

### Show Variables



Use the Trio Client Commands window to set show variables. The variables can be used by scripts as a means of storing intermediate values such as counters.

#### Commands:

- `show:set_variable` – Set the value of a show variable
- `show:get_variable` – Get the value of a show variable

#### Example commands:

```
COMMAND: show:set_variable MyShowVar Hello World!  
COMMAND: show:get_variable MyShowVar  
RESULT: Hello World!
```

#### Script example:

```
Sub OnUserClick()  
    dim myLocalShowVar  
  
    myLocalShowVar = TrioCmd("show:get_variable MyShowVar")  
    MsgBox("Variable value is: " & myLocalShowVar)  
    myLocalShowVar = InputBox("Enter a new Value:", "New Value")  
  
    TrioCmd("show:set_variable MyShowVar " & myLocalShowVar)  
End Sub
```

### Concept, Context and Variant Variables

Use the Trio Client Commands window to set context variables. Context variables are different than [Show Variables](#) as context variables are used to select the context a page is taken on-air with. The following contexts are available to Viz Trio users:

- **Concept** – A concept can be News or Sports or any other concept. Graphics in the same concept are created as individual scenes belonging to a specific concept.

- **Variant** – A concept can have different variants of the same scene. This can be a scene with the same tab-fields, but displayed as a top or lower third.
- **Context** – Additional user-defined contexts can be used to further differentiate a concept and its variants. For example the context Platform could have SD, HD, Mobile phones, web and so on defined within the concepts News and Sport.

## Commands

If a concept, context or variant for a show is set using one of the show commands, for example `show:set_variant top`, this will override the selection done in the page list's user interface, marking the variant with square brackets. Issuing the same command, without the variant name (`show:set_variant`) resets the variant to the initial selection done using the page list user interface.

**Note:** A `get` command cannot retrieve a variable's value if not a `set` command is issued first.

Available commands:

- **show:enable\_context()** – Enables concept and variant for the current show. This will also separate templates and pages into their own lists.
- **show:set\_concept(restString concept)** – Sets the show's concept to a new value.
- **show:get\_concept()** – Gets the show's concept if set by a set command.
- **show:set\_context\_variable (string variable, restString value)** – Sets the show's variant to a new value.
- **show:get\_context\_variable (string variable)** – Gets the show's context variable if set by a set command.
- **show:set\_variant (restString variant)** – Sets the show's variant to a new value.
- **show:get\_variant()** – Get the show's variant if set by a set command.
- **playlist:set\_concept(string value)** – Sets the concept for a show playlist.

**Note:** The `set_variant` and `get_variant` commands are aliases for the context commands when setting and getting variants, and are not applicable for other contexts.

Example commands:

```
COMMAND: show:set_context_variable concept Sport
COMMAND: show:set_context_variable variant Lower
COMMAND: show:set_context_variable platform HD

COMMAND: show:get_context_variable concept
RESULT: Sport
COMMAND: show:get_context_variable variant
RESULT: Lower
COMMAND: show:get_context_variable <contextname>
RESULT: HD
```

A parameter must be set using the commands, before it can be shown using the commands. An output will normally differ depending on the scene design, therefore not all the output is shown.

**IMPORTANT!** Command parameters are case sensitive.

### Tab-field Variables

Use the Viz Trio's commands window to set custom properties for tab-fields. The property can be used for scripting to set intermediate properties for tab-fields.

Commands:

- `set_custom_property` (string tabfieldName, string value)
- `get_custom_property` (string tabfieldName)

Example commands:

```
COMMAND: tabfield:set_custom_property 01 string
COMMAND: tabfield:get_custom_property 01
RESULT: string
```

---

## 10.2 Working with Macro Commands over a Socket Connection

Viz Trio can be controlled using custom made control applications over a TCP/IP socket connection on port 6200. This feature gives the possibility to use macro commands that can trigger actions in Viz Trio.

This section contains information on the following topics:

- [Sending Macro Commands](#)
- [Unicode Support](#)
- [Escape Sequences](#)
- [Error Messages](#)

**See Also**

- [Commands](#)

### 10.2.1 Sending Macro Commands

This section gives a quick overview of how to establish a connection with Viz Trio over a socket connection in order to send Macro Commands to control Viz Trio. The following procedures will use Telnet as an example.

This section contains information on the following topics:

- [To configure Viz Trio for socket connections](#)
- [To connect to Viz Trio using telnet](#)
- [To send commands to Viz Trio using telnet](#)
- [To enable telnet for Windows 7](#)

**To configure Viz Trio for socket connections**

1. Open the [Configuration Interface](#) and select [Socket Object Settings](#).
2. In the appearing pane set the following:
  - Set Socket type to Server socket
  - Set Text encoding to UTF-8
  - Check Autoconnect and set Socket Host to localhost and Socket Port to 6200.

3. Click OK.

#### To connect to Viz Trio using telnet

1. Open Telnet by clicking the Start button, typing Telnet in the Search box, and then clicking OK.
2. Type `telnet localhost 6200` and press Enter.

-----  
**Note:** Windows 7 users must turn on Telnet as it is by default turned off.  
-----

#### To send commands to Viz Trio using telnet

1. Create a page in Viz Trio with the page number 1000.
2. Open a telnet connection to Viz Trio, type `page:read 1000` and press Enter.

#### To enable telnet for Windows 7

1. Click the Start button, click Control Panel, click Programs, and then click Turn Windows features on or off. If you are prompted for an administrator password or confirmation, type the password or provide confirmation.
2. In the Windows Features dialog box, select the Telnet Client check box.
3. Click OK. The installation might take several minutes.

### 10.2.2 Unicode Support

Commands that can set and get text values (such as *set\_value* and *getproperty*) supports the full set of Unicode characters. The internal format is UTF-16 (WideString). However, when communicating with Viz Trio over a socket connection on port 6200, an 8-bit string must be used, and the default (and preferred) encoding is therefore UTF-8. It is also possible to use latin1 (ISO-8859-1, western european) encoding.

#### To toggle the socket encoding

- To toggle the socket encoding, send one of the following commands to the socket: `socket_encoding latin1` or `socket_encoding utf8`.

-----  
**Note:** The *socket\_encoding* command is not a macro command, and can only be used with socket communication.  
-----

#### See Also

- [sock](#) commands

### 10.2.3 Escape Sequences

A socket command is terminated by a new line. To support setting and getting multi-line parameters and results, some characters have to be escaped; that is, represented by sequences of other characters. Backslash (\) is used as the escape character for socket commands.



See the table below for currently supported escape sequences.

**Table 22:** Escape Sequences

Escape Sequence	Meaning
<code>\n</code>	New line
<code>\r</code>	Carriage return
<code>\xHH</code>	Two-digit hexadecimal character code. For instance, <code>\x00</code> is an ASCII zero and <code>\x0A</code> is a new line (the same as <code>\n</code> ).
<code>\\</code>	Back slash ( <code>\</code> )

### 10.2.4 Error Messages

When controlling Viz Trio over a socket connection you can enable error messages using the `commandserver` command.

```
commandserver:enable_error_messages true  
  
commandserver:enable_error_messages false
```

-----  
**Note:** Typing `commandserver` is optional  
-----

Enabling error messages will send error messages to the connected client as well as being displayed in the GUI (on a per socket connection). The client will receive all error messages that is normally just displayed in the Viz Trio user interface.

Note that not only the error messages generated by the client will be sent – all errors will be sent. For example an operator using Viz Trio at the same time and tries to read a non-existing page will result in error messages being sent to all clients connected over a socket connection.

Errors have this format:

```
ERROR: <Error message>
```

If error handling is turned off, the client will still get errors caused by commands not being found and commands that return an error.

This command is only implemented for the command server on port 6200, and is not related to the [Socket Object Settings](#) in the [Configuration Interface](#).

---

## 10.3 Working with Shortcut Keys

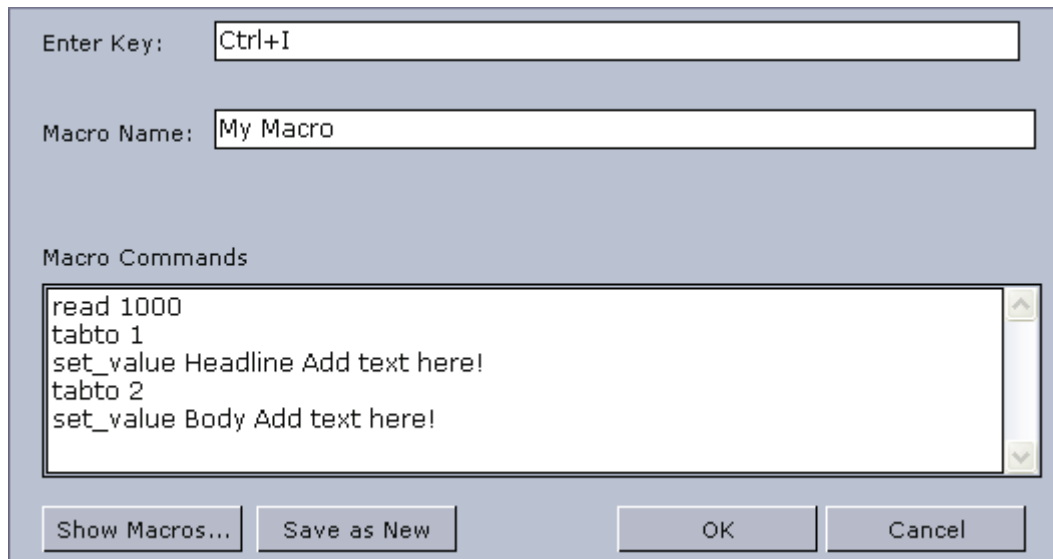
Viz Trio allows you to assign macro commands and scripts to shortcut keys; however, there are two ways of doing this; On a per show or global basis. Shortcut keys assigned on a global level are assigned through the [Configuration Interface](#), whereas show specific shortcut keys are assigned through the [Show Settings](#).

This section contains information on the following topics:

- To assign a macro or script to a shortcut key
- To reassign a shortcut key
- To remove a shortcut key
- To add a predefined function to a Macro or Script

### To assign a macro or script to a shortcut key

**Figure 66:** The Macro Commands editor



1. Click the **Add Macro** or **Add Script** button.
2. Select the keyboard keys for the shortcut (e.g. combinations of CTRL, SHIFT, ALT, ALT GR with other characters).
  - If the selected shortcut key is already in use you may override it, removing it from the other macro or script.
3. Enter a Macro or Script Name and Description of the macro or script.
  - If the macro name is already in use, the **Save as New** button appears.
4. Add the macro or script commands to the Macro or Script text area.
5. Click **OK** to confirm the selected key combination.

### To reassign a shortcut key

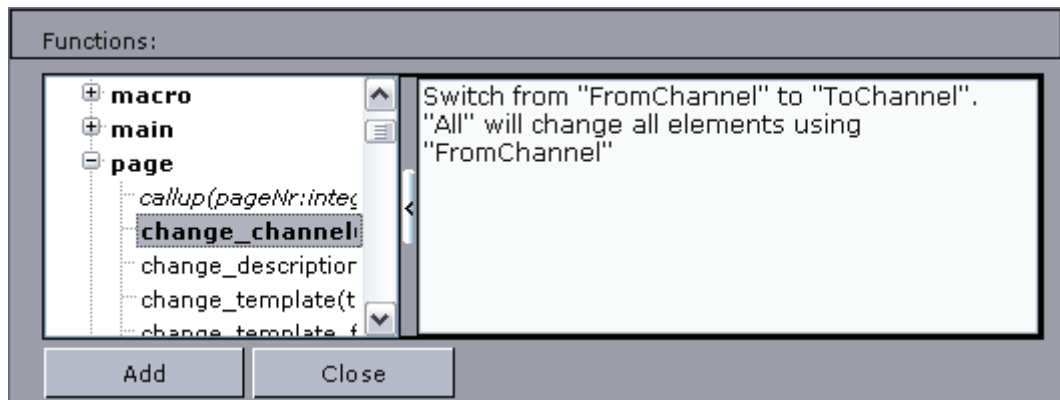
- Double-click the Command in the list and perform the changes.

### To remove a shortcut key

1. Right-click a Macro or Script that has been manually added and from the appearing context menu select **Remove**, or
2. Select a Macro or Script and click the **Remove** button, or
3. Select a Macro or Script and press the **Delete** button on the keyboard.

## To add a predefined function to a Macro or Script

**Figure 67:** Add functions to a macro or script



1. Open a Macro or Script for editing.
2. Click the **Show Macros ...** button in the [The Macro Commands editor](#) to open a list of predefined functions.
3. Select the function and click **Add**

-----  
**Note:** Predefined macro commands can also be applied to scripts.  
-----

### See Also

- [Keyboard Shortcuts and Macros](#)
- [Macro Commands and Events](#)

---

# 11 Macro Commands and Events

This section contains reference information on the following topics:

- [Commands](#)
- [Events](#)

## See Also

- [Scripting](#)
- [Macro Language](#)

---

## 11.1 Commands

The commands are grouped as follows:

- [channelcontrol](#)
- [config\\_settings](#)
- [gui](#)
- [macro](#)
- [main](#)
- [page](#)
- [playlist](#)
- [proxy](#)
- [rundown](#)
- [script](#)
- [scroll](#)
- [scroll2](#)
- [settings](#)
- [show](#)
- [sock](#)
- [tabfield](#)
- [table](#)
- [text](#)
- [trio](#)
- [util](#)
- [viz](#)
- [vtwtemplate](#)

**Table 23:** Commands

Command name and Arguments	Description
<b>channelcontrol</b>	
add_channel (restString channelName)	Adds a new channel to the active profile. The named channel must not exist.
get_active_profile	Returns the name of the active profile.

**Table 23:** Commands

<b>Command name and Arguments</b>	<b>Description</b>
get_preview_channel	Gets the name of the currently configured Viz preview channel. Returns an empty string if no Viz preview channel is defined.
get_program_channel	Gets the name of the currently configured Viz program channel. Returns an empty string if no Viz program channel is defined.
get_video_preview_channel	Gets the name of the currently configured video preview channel. Returns an empty string if no video preview channel is defined.
get_video_program_channel(restString channelName)	Gets the name of the currently configured video program channel. Returns an empty string if no video program channel is defined.
list_profiles	Returns a list of existing profiles, separated by line breaks.
remove_channel (restString channelName)	Removes the channel with the given name from the active profile. The channel name must be valid.
set_active_profile (restString profileName)	Change the active channel profile.
set_channel_hosts (string channelname, restString hostlist)	Changes the list of Viz hosts for a channel. The referenced channel must exist. The list of hosts can be space/comma/semi-colon separated.
set_preview_channel (restString channelName)	Changes the Viz preview channel. The channel name must be valid. If empty the Viz preview channel will be undefined.
set_program_channel (restString channelName)	Changes the Viz program channel. The channel name must be valid. If empty the Viz program channel will be undefined.
set_video_channel_hosts(string channelname, restString hostlist)	Changes the list of video hosts for a channel. The referenced channel must exist. The list of hosts can be space/comma/semi-colon separated.
set_video_preview_channel(restString channelName)	Changes the video preview channel. The channel name must be valid. If empty the video preview channel will be undefined.
set_video_program_channel(restString channelName)	Changes the video program channel. The channel name must be valid. If empty the video program channel will be undefined.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
wait_vizcmd_to_channel (string channel, restString cmd)	Sends a Viz Engine command to a named channel and waits for the returned answer.
<b>commandserver</b>	
enable_error_messages true <i>or</i> false	Enables error messages when Viz Trio is controlled over a socket connection. See <a href="#">Working with Macro Commands over a Socket Connection</a> .
<b>config_settings</b>	
set_channel_mode (string channel, restString mode)	Set channel mode.
single_machine_mode (boolean doSetSingleMachineMode)	Make trio go to single machine mode.
<b>dblink</b>	
link_database (string property key, string connection name, string table name)	<p>Links a property of a page to a table of a database connection.</p> <p><b>Example 1:</b>  <code>TrioCmd("dblink:link_database 1  ExcelTest ""['Scalar Tests\$']"")</code></p> <p><b>Example 2:</b>  <code>TrioCmd("dblink:link_database 1  ExcelTest ""['Table Tests\$']"")</code></p> <p>1 refers to the Control List, Chart or Text's Field Identifier that in Viz Trio can be identified as the template's tab field 1.</p> <p>ExcelTest refers to the name of the configured database connection made in <a href="#">Database Config</a>.</p> <p>Scalar Tests and Table Tests refers to the table that is chosen in order to select the lookup column.</p>

**Table 23: Commands**

Command name and Arguments	Description
map_database_column (string table key, string property column key, string database field name)	<p>Maps a column of a table property to a database field.</p> <p>Example: <code>TrioCmd ("dblink:map_database_column 1 3 Name")</code>.</p> <p>1 refers to the Control List's or Control Chart's Field identifier property that in Viz Trio can be identified as the template's tab field 1.</p> <p>3 refers to the Control Text Field identifiers in the scene that make out column number 3 that in Viz Trio can be identified as the table column in the page editor.</p> <p>Name refers to the column the data should be read from.</p>
select_value (string property key, string value column, string key column, string key value)	<p>Selects a single value (scalar linking) of a database and links it to the property. The value is selected from the &lt;value column&gt; where the &lt;key column&gt; matches the &lt;key value&gt;.</p> <p>Example:  <code>TrioCmd ("dblink:select_value 1 Color Key 2")</code></p> <p>1 refers to the Control Text's Field identifier property that in Viz Trio can be identified as the template's tab field 1.</p> <p>Color refers to the column the data should be read from.</p> <p>Key refers to the lookup column defined in the <a href="#">Database Config</a>.</p> <p>2 refers to the row the data should be read from.</p>
store_to_database (string property key)	<p>Stores a linked property of a page to its value in the database.</p> <p>Example:  <code>TrioCmd ("dblink:store_to_database 1")</code></p> <p>1 refers to the Control Text's Field identifier property that in Viz Trio can be identified as the template's tab field 1.</p>

**Table 23: Commands**

Command name and Arguments	Description
update_from_database (string property key)	<p>Updates a linked property of a page with its value in the database.</p> <p>Example:  <code>TrioCmd("dblink:update_from_database 1")</code></p> <p>1 refers to the Control Text's Field identifier property that in Viz Trio can be identified as the template's tab field 1.</p>
use_custom_sql (string property key, restString custom SQL)	<p>Uses a custom SQL clause to select the database value. An empty SQL clause disables the usage of custom SQL.</p> <p>Example:  <code>TrioCmd("dblink:use_custom_sql 1 SELECT Text FROM ['Table Tests\$']")</code></p> <p>1 refers to the Control List's or Control Chart's Field identifier property that in Viz Trio can be identified as the template's tab field 1.</p> <p>Text refers to the column the data should be read from.</p> <p>Table Tests refers to the database table (i.e. a single spreadsheet in Microsoft Excel).</p>
<b>gui</b>	
add_view	Brings up the add view dialog box.
artist_mode	Enter design mode (Viz Artist).
browse_image	Browse local file system for images.
change_path	Shows the directory selector.
clear_page_field	Clears the content of the page edit's text field.
copy_selected_pages_to_number (restString destnumber)	Copy the selected pages to number or show a dialog if no number is supplied.
copy_selected_pages_with_offset (restString destnumber)	Copy the selected pages with an offset or show a dialog if no number is supplied.
create_new_show	Create a new show in the currently selected directory. A unique name is generated and the new show path is returned.
design_mode	Enter Trio design mode.



**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
edit_begin_time	Shows the in place editor for the selected page.
edit_end_time	Shows the in place editor for the selected page.
edit_group_name	Shows the in place editor for the selected page.
edit_page_description	Shows the in place editor for the selected page.
edit_page_program_channel	Shows the in place editor for the selected page.
edit_scroll	Show the scroll editor.
edit_template (restString templateName)	Edit the specified template in a dialog. This only works for templates in concept-enabled shows.
error_message (restString message)	Display an error message in the status bar. The message will also be logged.
find	Shows the find dialog.
focus_active_view	
focus_next_view	
focus_prev_focused_view	Focus the previous focused view.
focus_prev_view	
get_selected_pages	Retrieves the currently selected pages as a space separated list.
goto_config	Opens Viz Trio's <a href="#">Configuration Interface</a> .
hide_timecode_monitor	Hides the TimeCode monitor. See also <a href="#">show_timecode_monitor</a> .
hit_test (x:integer, y:integer)	Test if a tab field exists on the X,Y coordinate.
load_image_from_db	Show the image database browser for the current tab field.
load_image_from_viz	Show the viz image browser for the current tab field.
log_message (restString message)	Add a message to the logfile.
move_selected_pages_to_number (restString destnumber)	Move the selected pages to number or show a dialog if no number is supplied.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
move_selected_pages_with_offset (restString destnumber)	Move the selected pages with an offset or show a dialog if no number is supplied.
open_effect_dialog	This command opens the effect dialog. Will not open if current page is not set.
playout_mode	Enter playout mode.
post_render	Opens the post render editor.
prepare_snapshot	Open the prepare snapshot editor.
print_selected_pages	Print the selected pages in the active view.
read_page_following_last_taken	Read the page succeeding the page that was last taken.
reimport_selected_pages	Reimport the selected pages in the active view.
select_last_taken_page	Select the page that was last taken.
select_nextpage	Select the next page in the active view.
select_prevpage	Select the previous page in the active view.
select_variant	Select the variant for the focused element in the page view.
set_autopreview (boolean autopreview)	Sets the auto-preview state (single-click read) for page views.
set_bounding_box (visible:boolean)	Toggles the bounding boxes for the elements in the preview (same as the BB button).
set_column_visible (string columnname, boolean active)	Sets the visibility of the column. Applies to all pageviews.
set_focus_to_editor	Sets focus to the currently active tab field editor.
set_interactive_mode (boolean active)	Sets interactive mode which controls whether to show dialogs to the user or not.
set_run_script_button_enabled (boolean isEnabled)	Sets the run script button to enabled/disabled.
set_safearea (visible:boolean)	Toggle safe area.
set_silent_mode (boolean active)	Sets silent mode which controls whether to show dialogs to the user or not.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
set_statusbar_color (int R, int G, int B)	Sets the color of the statusbar based on the RGB values given.
set_titlearea (visible:boolean)	Toggles the title area in the preview window.
show_custom_prop_column (boolean value)	Shows or hides custom property column in tab field list.
show_export_dialog	Show the export show dialog.
show_external_cursor (boolean value)	Show or hide the external cursor in the page view.
show_import_dialog	Show the import show dialog.
show_pageeditor	Shows the page editor.
show_pages_in_rundown (boolean doShow)	Shows or hides pages in the rundown for the current show.
show_pageview	Show the page view.
show_playlist (string refPath, boolean requestMos, restString description)	Shows the playlist with the path specified. Can choose to request a MOS playlist.
show_scripteditor	Shows the script editor.
show_scroll_template_creator	Shows the scroll template creator view.
show_settings	Open the show settings view.
show_shortcut_list	Show the view of local shortcuts and macros.
show_showscript_editor	Shows the show script editor.
show_status_icons (boolean value)	Shows or hides status cells in the callup grid.
show_templates_in_rundown (boolean doShow)	Shows or hides templates in the rundown for the current show.
show_timecode_monitor	Shows the TimeCode Monitor. See also <a href="#">hide_timecode_monitor</a> .
show_triocommands	Toggles the Trio Commands window.
status_message (restString message)	Display a text message in the status bar. The message will also be logged.
take_callup	Take the current callup number directly, without a read.

**Table 23:** Commands

<b>Command name and Arguments</b>	<b>Description</b>
take_snapshot (string path, string handlerName)	Takes snapshot from the specified handler in RGB format.
take_snapshot_rgba (string path, string handlerName)	Takes snapshot from the specified handler in RGBA format.
toggle_design_mode	Toggle between Trio design and playout mode.
toggle_error_message_window	Toggle the visibility of the error message window.
toggle_hide_empty_locked_groups (visible:boolean)	Toggle hide/show empty and locked groups.
toggle_hide_nameless_groups (visible:boolean)	Toggle hide/show groups with no name.
<b>macro</b>	
export_macros_to_xml (string filename)	Exports the macro commands to a XML file.
<b>main</b>	
apropos (search:string)	Search command names and descriptions for a string.
call_macro (restString name)	Call (invoke) a named macro.
define_local_macro (string name, restString commands)	Define a named, local macro.
define_local_script (string name, restString commands)	Define a named, local script.
define_macro (string name, restString commands)	Define a named, global macro.
define_script (string name, restString commands)	Define a named, global script.
delete_global_macro (name:string)	Delete a global macro.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
delete_local_macro (name:string)	Delete a local macro.
get_global_macros	List the names of global macros (separated by new lines).
get_local_macros	List the names of local macros (separated by new lines).
get_shows	Get the names of all the shows in the client
get_version	Get a version string.
help (command:string)	Shows help for the specified command, or lists commands.
macro	
<b>page</b>	
change_channel ("string FromChannel, string ToChannel, boolean All)	Switch from "FromChannel" to "ToChannel". "All" will change all elements using "FromChannel".
change_description (string pageNr, restString desc)	Sets the page description.
change_template (string templateName)	Changes the template of the selected pages. Empty parameter("") will take the number in the callup field and use that as the template ID.
change_template_for_pages (string templateName, restString pages)	Changes the template of the given pages.
change_variable (string variable, restString value)	Change the variant for the currently edited page.
continue (restString pageNr)	Continues the page specified as parameter or the selected if the parameter is empty.
cue (restString pageNr)	Cues the page specified as parameter or the selected if the parameter is empty.
cut (pageNr:integer)	Cuts the current template or page or the page specified.
delete_page	Delete selected page(s) in active view.
delete_pagerange (int start, int end)	Deletes the page inside from the begin range to (including) the end range parameter.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
direct_continue	Continue selected page in active view.
direct_cut	Cut selected page in active view.
direct_take	Take selected page in active view.
direct_takeout	Take out selected page in active view.
export_property (string key, string filename)	Export a property value to file (UTF-8 encoded).
get_current_propertykey	Get the current property-key.
get_current_tabfield_name	Get the name of the current tab field.
get_layers (string page)	Will get the layer name(s) the combination or transition logic page is based on, hence, this command is not valid for pages based on standalone scenes (returns an empty string).
get_program_pagechannel (restString pageid)	Gets the program channel for the page.
get_property (string name)	
get_tabfield_count	Gets the number of tab field on the current page.
get_template_description (string page)	Gets the template description for the page specified.
get_viz_layer (restString page)	Gets the viz layer for a page. If no page name is supplied, current page is used. Only valid for scene-based pages.
getpagedescription	Get the description of the current page.
getpagename	Get the name of the current page.
getpagetemplate	Get the template of the current page.
import_property (string key, string filename)	Import a property value from file (UTF-8 encoded).
load_page_data (filename:string (optional))	Load page data from XML.
print	Print the current page. Use settings printer/left_margin, printer/right_margin, printer/top_margin, printer/bottom_margin to control the size, where the value is the percent of the page width/height.

**Table 23:** Commands

<b>Command name and Arguments</b>	<b>Description</b>
read (pageNr:integer)	Reads the specified page number.
read_parentpage	
read_rootpage	
read_subpage (string string)	
read_template (string templateid)	Reads a template if concepts and variants are enabled. If not, read a normal template.
read_template_with_context (string template, string contextPairs)	Reads a template with context variables that can override the show settings. The context is given a whitespace-separated list of keys and values.
readnext	Read next element in active view.
readprevious	Read previous element in active view.
redo	Redo edit in current page history.
run_command_with_variables_on_channel (string command, string channel, restString variables)	Run a command (for example a take) on a page with variables to a specific channel.
save	Saves current page.
save_page_data (filename:string (optional), version:string (optional, defaults to 2.6))	Save page data as XML in specified version (2.5 or 2.6).
save_scene	Save current page as a new Viz scene.
saveas ([pageNr:integer])	Triggers the save as action for a template or page, saving it with the page number as parameter. If no parameter is given, the callup code will be set to the next page number available; however, the page will not be saved.
set_default_effect (string effect)	Sets the default transition effect to be used between (scene-based) pages. Empty string specifies hard cuts (default).
set_effect (restString effect)	Sets the transition effect to be used between (scene-based) pages. Empty string specifies hard cuts (default).

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
set_effect_on_selected (string effect)	Sets the transition effect to be used between (scene-based) pages on the currently selected one. Empty string specifies hard cuts (default).
set_expert_mode_enabled (boolean enabled)	Enable or disable "expert mode" for the current page. This enables all tab field properties to be accessed instead of a filtered subset.
set_pagestatus (status: string)	Set the status of the selected pages. Valid arguments: FINISHED/UNFINISHED/NONE.
set_pagestatus_one (pagenr: integer, status: string)	Set the status of page. Valid status arguments: FINISHED/UNFINISHED/NONE.
set_pagetime (string pageid, string type, string value)	Sets the begin/end time for the page specified.
set_program_pagechannel (string pageid, restString channelname)	Sets the program channel for the page.
set_property (string name, restString value)	Set the value of a named tab field property.
setpagetemplate (string templateName)	Change the template of the current page.
store	Save the page with the name given in the read page edit field (the left one).
tab	Tab to the next tab field.
tab_backwards	Tab to the previous tab field.
tabto (tabfieldnr:integer)	Go to a specific tab-field number.
tabtotabfield (tabfieldname:string)	Go to a specific tab-field number.
take (pageNr:integer)	Does a take on the current template or page or on page specified.
take_exchange	Take current page and read the last page that was taken.
take_with_variables (restString variables)	Take a page with variables.



**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
take_with_variables_on_channel	Take current page and read next page in active view.
takenext	Take current page and read next page in active view.
takeout (pageNr:integer)	Take out current template or page or template or page specified.
undo	Undo edit in current page history.
update	Update the program renderer (do a take without animating it in).
update_preview	Update the external preview renderer with the contents of the local preview.
<b>playlist</b>	
add_page (string pagename, restString duration)	Add a page to the end of the playlist. The duration argument is optional. If not given, the page duration will be used.
create_playlist (string name)	Add a new empty playlist.
direct_take_out_selected	
direct_take_selected	
go_to_next_story	Select the next story (group) in the active playlist.
go_to_previous_story	Select the previous story (group) in the active playlist.
goto_manual_graphics_mode (restString enable)	Enable/disable manual playout of graphic elements. Use "true" or "" for manual and "false" for automatic mode.
initialize_playlist	Initialize the playlist (load graphics) on the external output channels.
jump_to_ncs_cursor	Jump to the element that the NCS-cursor (newsroom control system) is pointing to.
pause_carousel	Pause the carousel.
read_next_item	Read the next playlist item.
read_prev_item	Read the previous playlist item.
read_selected	Read the currently selected playlist element.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
run_selected_with_custom_action (string Action)	<p>This macro command is generic and can append any Media Sequencer command as a parameter for playing out selected graphics and/or video clip elements in any playlist.</p> <p>Example:</p> <pre>playlist:run_selected_with_custom_action "prepare"</pre> <pre>playlist:run_selected_with_custom_action "take"</pre> <p>Note that if you have a version of the Media Sequencer that is newer than the version released with your version of Viz Trio, you can still use new commands supported by the Media Sequencer by using this macro command.</p>
select_first	Select the first element in the playlist.
select_next	Select the next element in the active playlist.
select_previous	Select the previous element in the active playlist.
set_attribute_on_selected (string attribute, restString value)	Sets an attribute on all the selected items in a playlist.
set_concept (string value)	Set the concept for a show.
set_looping (restString looping TRUE or FALSE)	Set to true if the carousel shall be looping.
start_carousel (play list name or id (optional, defaults to active play list))	Run the elements in the playlist as a carousel.
stop_carousel (play list name or id (optional, defaults to active play list))	Stop a playlist running as a carousel.
<b>proxy</b>	
enable_viz_proxy (boolean enable)	Enables/disables Trio acting as a Viz Proxy on the port configured.
set_proxy_port (int port)	Set the port that Trio should listen on.
set_viz_host (string host)	Set the host where the proxy can locate Viz.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
set_viz_port (int port)	Set the port the proxy should use to access Viz.
<b>rundown</b>	
go_to_next_story	Sets focus on the next story/group in the active rundown.
go_to_previous_story	Sets focus on the previous story/group in the active rundown.
reload_template_images (restString templates)	Comma separated list of template IDs.
<b>script</b>	
eval (restString scriptExpression)	Evaluate the VBScript expression given as a parameter, and return the value converted to a string.
execute_script	Execute the script of the current page.
get_showscript_name	Gets the name of the current show's script.
import_script (string scriptName, restString scriptFilename)	Imports the script into the current show.
run_macro_script (restString scriptCode)	Run the VBScript script code given as a parameter. The snippet can contain multiple lines of code. The macro will return either an empty string or an error message.
run_script (string scriptFunction, restString argumentList)	Run the specified script function. The argument list is a list of strings separated by commas or white space. Arguments that can contain commas or white space must be enclosed by double quotes.
run_showscript (string scriptFunction, restString argumentList)	Run the script associated with the current show. The argument list is a list of strings separated by commas or white space. Arguments that can contain commas or white space must be enclosed by double quotes.
<b>scroll</b>	
scroll_add_range (index:integer, ranges:string)	Insert one or more ranges of elements into scroll at index. Use index -1 to insert at end.
scroll_clear	Clear all scroll data, including scroll elements.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
scroll_element_parameter (index:integer,name:string,value(s):variant)	Set a parameter on the scroll element at index. Supported parameters are: [page (string): set the page name, will refresh element] [space_before (float): extra space before this element, accumulates with padding] [space_after (float): extra space after this element, accumulates with padding] [scaling (two floats): set scaling in X and Y direction] [position (two floats): position adjustment in X and Y direction].
scroll_insert_elements (index:integer, pages:string)	Insert elements into scroll at index. Use index -1 to insert at end.
scroll_move_element (from_index,to_index:integer)	Move scroll element from from_index to to_index.
scroll_refresh_all	Refresh all scroll elements. Also refreshes end page and background page.
scroll_refresh_element (index:integer)	Refresh scroll element at index. This will refresh both the viz object and the data.
scroll_remove_all	Remove all scroll elements. This will not clear other scroll parameters.
scroll_remove_element (index:integer)	Remove scroll element at index.
scroll_select_element (index:integer)	Select a scroll element. Use index -1 to clear the selection.
scroll_set_endpage (page:string,offset:float(optional))	Set the scroll end page, and (optionally) the end page offset.
scroll_set_endpage_offset (offset:float)	Set the end page offset, in seconds from end of scroll animation.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
set_scroll_parameter (name:string,value(s):variant)	Set a scroll parameter. The supported parameters are: [direction (up/down/left/right): the scroll direction] [padding (float): padding between elements] [alignment (none/left/right/bottom/top): left/right or bottom/top alignment, depending on direction] [alignment_offset (float): alignment offset in percentage of screen size] [background_page (string): name of background page] [use_pixel_speed (boolean): if true, scroll will use pixel speed] [pixel_speed (float): number of pixels per frame (takes effect if use_pixel_speed is true)] [duration (float): duration of scroll animation (takes effect if use_pixel_speed is false)] [start_offset (float): start offset of scroll in percentage of screen size] [pause_at_start (boolean): pause scroll animation at start] [viz_layer (front/main/back): viz layer for scroll scene].
<b>scroll2</b>	
add_easepoint (string pagename)	
clear_elements	Clear scroll elements (remove all).
copy_element (int sourceIndex, int targetIndex)	Add a copy of scroll element at sourceIndex to targetIndex.
create_scroll_template (string sceneName, string templateDescription, float width, float height, boolean showBounds)	Create a new scroll template with the given paramters.
edit_easepoint (string subpage, string easepoint)	Edit an easepoint in the current scroll.
insert_objects (int index, restString objects)	Insert SCENE or GEOM objects from viz into the current scroll.
insert_pages (int index, restString pages)	Insert pages from current show into current scroll at index.
insert_range (int index, string range)	Insert an inclusive numerical range of pages (e.g 1000–2000) from the current show, or a single page.
move_element (int sourceIndex, int targetIndex)	Move scroll element from sourceIndex to targetIndex.

**Table 23:** Commands

<b>Command name and Arguments</b>	<b>Description</b>
next_easepoint	Go to next ease point editor.
next_element_data	Go to next element data (extra spacing) editor.
preview_continue	Continue the current scroll in preview.
preview_start	Start the current scroll in preview.
preview_stop	Stop the current scroll in preview.
previous_easepoint	Go to previous ease point editor.
previous_element_data	Go to previous element data (extra spacing) editor.
program_continue	Continue the current scroll on the program channel (must be taken on-air first).
program_start	Start the current scroll on the program channel (must be taken on-air first).
program_stop	Stop the current scroll on the program channel (must be taken on-air first).
reload_all_page_data	Reload all scroll page data by copying from original pages.
reload_page_data (int index)	Reload scroll page data at index by copying from original page.
reload_template_graphics	Reload all scroll elements (reload geometry objects created during import of templates).
reload_templates	Reload all scroll elements by merging with the current show templates.
remove_easepoint (string subpage, string easepoint)	Remove a named easepoint in the current scroll.
remove_element (int index)	Remove scroll element at index.
rename_easepoint (string subpage, string oldName, string newName)	Rename an easepoint in the current scroll.
set_focus (int index)	Focus on scroll element.
<b>settings</b>	
get_setting (string path)	Gets a Viz setting.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
set_global_setting	Sets a global Viz setting (e.g. settings:set_global_setting encoding ISO-8859-1).
set_setting (string path, restString value)	Sets a Viz setting.
<b>show</b>	
append_pages_from_xml (boolean showDialogs, string filename)	Loads pages from XML and replaces any pages that will be overwritten.
copy_pages (page list)	Copies the given page(s) to the clipboard
copy_pages_to_number (int destination, restString CommaSeperatedList)	Copy the pages to a destination number.
copy_pages_with_offset (int offset, int CommaSeperatedList)	Copy the pages to a destination number.
create_combo_page (string pagelist)	Create a combo page.
create_dir (restString path)	Creates the specified show directory.
create_group (string groupname)	Create a page group in the current show, before the focused node.
create_playlist (restString name)	Add a new playlist to the show.
create_show (string path, restString EnableContext)	Create a new show. If EnableContext is not "false" then a context-enabled show is created.
cut_pages (page list)	Cuts the given page(s) to the clipboard.
delete_all_pages	Deletes all the pages in the page view.
delete_templates (restString templateList)	Delete the given templates from the current show. Does nothing if the show does not have concepts and variants enabled.
enable_context	Enable concept/variant for the current show.
export_setting (string name, restString value)	Configure an export setting, use this before export_show.

**Table 23:** Commands

<b>Command name and Arguments</b>	<b>Description</b>
export_show (filename:string)	Export a show (trio archive).
get_all_timing_handlers	Returns a comma separated list of all timing handlers.
get_associated_files	Get the list of associated files for the show
get_background_scenes	Get a space-separated list of background scenes in the show.
get_concept	Get the show-concept.
get_context_variable (string variable)	Get the show-context-variable.
get_custom_colors	Get the custom colors for the current show.
get_default_viz_path	Get the default viz scene path for the current show.
get_group_pages	Returns a list of pages in the specified group.
get_groups	Returns a list of groups in the current show.
get_library_path	Get the default viz scene path for the current show.
get_page_xml(restString page_path)	Get the xml for the specified page. The path is given as showpath/<page-id> (e.g. get_page_xml myshow/1000). For "untitled show" use empty showpath (e.g. get_page_xml /1002). Note that show names are case sensitive.
get_pages	Returns a list of pages in the current show.
get_preview_background_image	Get the Preview background image for the current show.
get_scenes_for_pages (restString pagelist)	Get the list of scenes for the given pages.
get_setting (string path)	Get a show setting.
get_settings	Get all settings.
get_show_description	Get the description of the current show.
get_show_name	Get the name of the current show.



**Table 23:** Commands

<b>Command name and Arguments</b>	<b>Description</b>
<code>get_timing_handler</code>	Returns the timing handler for the current show.
<code>get_variable</code> (string propName)	Get the value of a show variable.
<code>get_variant</code>	Get the show-variant.
<code>goto_viz_dir</code> (path:string)	Change to a show with the same name as a viz directory, creating it if necessary.
<code>import</code> (scenes:string)	Shows the <a href="#">Import Scenes</a> pane or imports specified scene(s).
<code>import_and_initialize</code> (scenes:string)	Import scenes to templates, then reload on external renderers.
<code>import_no_gui</code> (restString sceneNames)	Imports the scenes given as parameter without showing/closing the import GUI.
<code>import_pages_from_xml</code> (string filename, int offset, boolean interactive, boolean pageNames)	Import the named pages from an XML file. An offset different from zero will cause pages with numerical names to be offset by this number, unless they are templates. Pages are overwritten automatically if interactive is false.
<code>import_recursively</code> (string path)	Import all the scenes recursively from this folder.
<code>import_setting</code> (string name, restString value)	Configure an import setting, use this before <code>import_show</code> .
<code>import_show</code> (filename:string)	Import a show (Viz Trio archive).
<code>load_locally</code>	Loads all scenes referenced by pages in the current show on the local viz preview.
<code>load_show</code>	Load show (scenes, textures, fonts) on external output channels. Replaces the old initialize command.
<code>load_show_on_renderers</code> (string ProgramChannelName, restString PreviewChannelName)	Load current show on the specified channels. First argument is treated as a program channel, second as preview.
<code>move_pages_to_group</code> (string group, restString pages)	Move pages into the specified group. If the group name is empty, move to top level.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
move_pages_to_number (int destination, restString CommaSeperatedList)	Copy the pages to a destination number.
move_pages_with_offset (int destination, int CommaSeperatedList)	Move the pages with an offset.
paste_pages	Pastes the given page(s) from the clipboard to the current show.
page_exists (restString page)	Checks if a page with the specified name exists in the current show. Returns "true" or "false".
reload_on_renderers (restString pages)	Reload (re-initialize) graphics for the specified pages on program and preview channels.
rename_page (string old_name, restString new_name)	Sets a new name for the page. If a new name is not defined, a dialog appears asking for the new name.
run_show (boolean b)	Runs/stops the show.
save_pages (string FileName, restString CommaSeperatedList)	Saves the pages given as parameter to XML.
save_pages_to_xml (string filename)	Saves pages to XML.
save_selected_pages (restString fileName)	Saves the selected pages to XML.
set_associated_files (restString filelist)	Set the associated files for the show. ("file1" "file2" "file3").
set_concept (restString concept)	Set the show-concept to a new value.
set_context_variable (string variable, restString value)	Set the show-variant to a new value.
set_custom_colors (restString colors)	Set the custom colors for the current show.
set_default_viz_path (path:string)	Set the default viz scene path for the current show.

**Table 23:** Commands

<b>Command name and Arguments</b>	<b>Description</b>
set_dir (restString path)	Changes the show directory to the path specified.
set_external_cursor (restString pagename)	Set the external cursor to page with page name or to current page if page name is empty.
set_library_path (restString path)	Set the design-library path.
set_preview_background_image (path:string)	Set the preview background image for the current show.
set_setting (string path, restString value)	Set a show setting.
set_show_description (description:string)	Set the description of the current show.
set_show_name (name:string)	Set the name of the current show.
set_timing_handler (string handlerName)	Set the timing handler for the current show.
set_variable (string propName, restString value)	Set the value of a show variable. The value is stored on the Media Sequencer.
set_variant (restString variant)	Set the show-variant to a new value.
show_export_dialog_for_path (restString showpath)	Exports the show.
<b>sock</b>	
connect_socket	Explicitly connect (open) the socket, if not already connected.
disconnect_socket	Disconnect (close) the socket, if connected.
send_socket_data (restString text)	Send UTF-8 encoded text over the socket. If it is a server socket, send to all clients.
set_com_baud (int comBaudRate)	Set baud rate to use for the serial port.
set_com_number (int comPortNumber)	Set COM port number to use for the serial port.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
set_socket_autoconnect (boolean autoconnect)	If auto-connect is true, a client socket will be connected the first time data is sent, or a server socket will be opened automatically.
set_socket_data_separator (string separator)	Set the separator for the OnSocketDataReceived callback.
set_socket_encoding (string encoding)	Set text encoding for socket to either UTF-8 or ISO-8859-1.
set_socket_host (string hostname)	Set the socket hostname. Does nothing if the socket type is server.
set_socket_port (int port)	Set the socket port number.
set_socket_type (string type)	Set the socket type to client or server.
socket_is_connected	Returns true if the socket is connected, false otherwise.
<b>tabfield</b>	
active (active:string)	Toggle active flag of current tab field or set its value (0 or 1).
browse_file	
clock	Change to clock editor of current tab field or set its value (string).
downcase_tabfield	Convert current tab field to lower case, if it contains text.
edit_property (string propertyName)	Changes editor to the specified property i.e 1.image.
geom (vizPath:string)	Change to geometry editor of current tab field or set its value (GEOM*path).
get_custom_property (string tabfieldName)	Gets the custom property of the tab field. "" will get it for the current tab field.
get_tabfield_property (tabfieldnr:integer,property:string)	Gets a property value for a given tab field.
image (vizPath:string)	Change to image editor of current tab field or set its value (IMAGE*path or filename).
image_position	Change to image position editor of current tab field or set its value (two floats).

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
image_scaling	Change to image scaling editor of current tab field or set its value (two floats).
import_value_from_file (restString filename)	Load a tab field value from a text file (UTF-8 encoded).
kerning (value:float)	Change to kerning editor of current tab field or set its value (float).
material (vizPath:string)	Change to material editor of current tab field or set its value (MATERIAL*path).
next_property	Select the next property of the current tab field, or the first if the last one is currently selected.
position (value:float)	Change to position editor of current tab field or set its value (floats).
previous_property	Select the previous property of the current tab field, or the last if the first one is currently selected.
rotation (value:float)	Change to rotation editor of current tab field or set its value (floats).
scaling (value:float)	Change to scaling editor of current tab field or set its value (float).
set_custom_property (string tabfieldName, string value)	Sets the custom property of the tab field. "" as tab-field name will use the current tab field.
set_tabfield_property (prop:string, value:string)	Set a property value.
set_value (value:string)	Sets the value of the current tab-field property.
text (value:string)	Change to text editor of current tab field or set its value (unicode string).
toggle_current_custom_property (string value1, string value2)	Toggle the custom property of the current tab field between the two given values.
upcase_tabfield	Convert current tab field to upper case, if it contains text.
value	Change to default editor.
<b>table</b>	

**Table 23:** Commands

<b>Command name and Arguments</b>	<b>Description</b>
clear_sorting (string table key)	Clear the sorted state of a table property.
delete_rows (string table key, string row index list)	Delete a number of rows from a table property.
get_cell_value (string table key, string column key, int row num)	Get a cell value of a table property.
insert_rows (string table key, int row index, int number of rows to insert)	Insert a number of rows into a table property.
move_current_row_down	Move the selected row in the current table editor one step downwards.
move_current_row_up	Move the selected row in the current table editor one step upwards.
move_row (string table key, int row index to move, int target row index)	Move a row downwards behind the target row or upwards before the target row.
set_cell_value (string table key, string column key, int row num, restString new value)	Set a cell value of a table property.
sort_column (<table key> <column key> [ascending or descending] (optional, default: ascending))	Sort a table ascending or descending after a column.
<b>text</b>	
incr_alpha (string key, string selectionStart, string selectionEnd, float alpha)	Increment or decrement the transparency of selected text.
incr_kerning (string key, string selectionStart, string selectionEnd, float kerning)	Increment or decrement the kerning of selected text.
incr_position (string key, string selectionStart, string selectionEnd, float posX, float posY, float posZ)	Increment or decrement the position of selected text.

**Table 23:** Commands

<b>Command name and Arguments</b>	<b>Description</b>
incr_rotation (string key, string selectionStart, string selectionEnd, float rotX, float rotY, float rotZ)	Increment or decrement the rotation of selected text.
incr_scale (string key, string selectionStart, string selectionEnd, float scaleX, float scaleY, float scaleZ)	Increment or decrement the scaling of selected text.
revert (string key)	Revert current text to saved value.
set_alpha_mode	Set text manipulation mode to Alpha.
set_color (string key, string selectionStart, string selectionEnd, string color)	Change the color of selected text.
set_color_mode	Set text manipulation mode to Color.
set_justification (string key, string justification)	Set the text justification (LEFT, CENTER, RIGHT).
set_kerning_mode	Set text manipulation mode to Kerning.
set_position_mode	Set text manipulation mode to Kerning.
set_rotate_xy_mode	Set text manipulation mode to Rotate x and y.
set_rotate_z_mode	Set text manipulation mode to Rotate z.
set_scale_mode	Set text manipulation mode to Scale.
show_replace_list	Shows the pre-configured character palette for the character to the left of the cursor.
<b>trio</b>	
assign_script (string page, boolean fileScript, restString scriptunit)	Assign or change the script for the specified page.
cleanup_channel (restString channelname)	Cleans up the specified channel by clearing Viz layers and freeing memory. <b>Note:</b> The <code>trio:clear_channel</code> command duplicated the <code>trio:cleanup_channel</code> command and is therefore as of version 2.10 deprecated.

**Table 23:** Commands

<b>Command name and Arguments</b>	<b>Description</b>
cleanup_renderers	Cleans up Viz program and preview channels by clearing Viz layers and freeing memory.
clear_preview	Cleans up Viz preview channel by clearing Viz layers and freeing memory.
clear_program	Cleans up Viz program channel by clearing Viz layers and freeing memory.
continue_last_taken_page	Continue the page that was last taken.
create_snapshot_page	Takes a snapshot and creates a page with that snapshot.
display_imported_archives	Display a list of the imported trio-archives.
exchange	Exchange the content of the program and external preview.
exit	Close the Trio.
freeze_program_clock (string string)	Freeze a clock on the program machine.
get_global_variable (string propName)	Get the value of a global variable.
get_next_savenumber	Get the number that will be proposed on the next save as or with a create_snapshot_page.
help	Opens the Viz Trio user's guide. The help file is also opened by clicking the Help button in the user interface (next to Config), or by pressing the F1 key unless it has been assigned to another action (e.g. macro). See also <a href="#">General</a> settings.
initialize_show_or_playlist	Initialize current show or playlist on external renderers.
invoke_with_current_page (string commandName)	Invoke the given command with the name of the current page as a parameter (useful for macro shortcuts).
invoke_with_current_scene (string commandName)	Invoke the given command with the name of the current scene as a parameter (useful for macro shortcuts).
offair	Set client in off-air mode.
onair	Set client in on-air mode.
open_printsettings_dialog (restString pages)	Open the printer-settings dialog.



**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
print_and_save_snapshots (restString pages)	Print the pages. Use settings printer/left_margin, printer/right_margin, printer/top_margin, printer/bottom_margin to control the size, where the value is the percent of the page width/height.
print_snapshots (restString pages)	Print the pages. Use settings printer/left_margin, printer/right_margin, printer/top_margin, printer/bottom_margin to control the size, where the value is the percent of the page width/height. With no argument, the content of the preview is printed.
process_ii_string (restString value)	Sends a string to the intelligent interface handler on the Media Sequencer and lets it process it.
restart_local_viz	Restarts the local Viz preview channel.
restart_local_viz_program	Restarts the local Viz program channel. Only available when local program channel has been started.
save_snapshots (restString pages)	Print the pages. Use settings printer/left_margin, printer/right_margin, printer/top_margin, printer/bottom_margin to control the size, where the value is the percent of the page width/height.
send_local_vizcmd (command:string)	Send a Viz command to the local viz preview.
send_preview_vizcmd (command:string)	Send a Viz command to the preview renderer.
send_preview_vizcmd_noerror (restString command)	Send a Viz command to the preview renderer if possible, otherwise do nothing (no error message).
send_vizcmd (command:string)	Send a Viz command to the program renderer.
send_vizcmd_to_channel (string channelName, restString command)	Send a Viz command to a specified channel.
set_global_variable (string proptime, restString value)	Set the value of a global property. The value is stored on the Media Sequencer.
set_mcu_folder (string string)	Sets the folder of the MCU/AVS plug-in.

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
set_vizconnection_timeout (int timeout)	Set the Viz connection timeout.
show_trio_settings_dialog	Show dialog containing a tree view of Trio settings.
sleep (int millisec)	Sleeps a page a certain amount of milliseconds. For example: TrioCmd("page:take 1000") TrioCmd("trio:sleep 10000") TrioCmd("page:take 1001") TrioCmd("trio:sleep 10000") TrioCmd("page:takeout")
swap_channels	Swap the program and preview channels.
synchronize_clocks	Synchronize the clocks to the clocks on the program renderer.
takeout_last_taken_page	Continue the page that was last taken.
unfreeze_program_clock (string string)	Freeze a clock on the program machine.
<b>util</b>	
get_node_text (string string)	Gets the node text of a Media Sequencer node.
list_archive_files (filename:string)	Returns a list of files in a trio archive (will also work for other tar files).
list_archive_pages (filename:string)	Returns a list of pages in a trio archive (creates a list of all_pages.xml within the Viz Trio archive).
list_archive_parameters (restString filename)	Returns a list of parameters of a trio archive.
list_xml_pages (restString filename)	Returns a list of pages in an XML file.
run_mse_entry (string path)	Runs the node at the specified path. No context is given.
set_node_text (string path, restString value)	Sets the node text of a Media Sequencer node.
<b>viz</b>	

**Table 23: Commands**

<b>Command name and Arguments</b>	<b>Description</b>
continue	Continue the animation in viz preview (if stopped).
import_image_from_file (string filename, restString pool folder)	Import the given image file into the given viz image pool folder.
play	Start (play) the animation in viz preview.
reload_data_locally (boolean paths)	Reload viz resources locally. The space-separated list of paths must include the pool prefix and full path, e.g. SCENE*folder/name.
set_bounding_box (boolean visible)	Toggle bounding box visibility.
set_key_alpha	Viz Artist setting.
set_key_preview	Viz Artist setting.
set_rgb	Viz Artist setting.
set_safearea (boolean visible)	Toggle safe area visibility.
set_stage_pos (restString pos)	Set the current stage position.
set_titlearea (boolean visible)	Toggle title area visibility.
stop	Stop the animation in viz preview.
<b>vtwtemplate</b>	
get_template	Gets the VTW template for the current show.
import_template (string ScriptName, restString FileName)	Imports a template from file to the current show. Overwrites any template with the same name.
reimport_active_template	Reimports the active VTW template for the current show.
run_vtw_function (string functionName, restString argumentList)	Runs a script function in the active VTW template. The argument list is a list of strings separated by commas or white space. Arguments that can contain commas or white space must be enclosed by double quotes.

Table 23: Commands

Command name and Arguments	Description
run_vtw_script (string FunctionName, restString Data)	Runs a script function in the active VTW template. Only a single parameter is supported by the Data parameter, so use vtwtemplate:run_vtw_function instead if possible.
set_template (restString templateName)	Sets the VTW template for the current show.

**See Also**

- [Scripting](#)

---

## 11.2 Events

- **OnBeforeSave()** – Called before Viz Trio saves the current page.
- **OnContinue** – Called when the Continue button is pressed or a continue macro command is executed.
- **OnCopy(OldName, NewName)** – Called when a page is copied. The event is only available to show scripts.
- **OnDelete(PageName)** – Called when a page is deleted. The script can abort the deletion by returning false (`OnDelete = false`). Note that this event is a function, hence, it can return a value. If you set it to `false` it will not delete the page in question. If you set it to anything else or do not return a value, the page in question will be deleted. The event is only available to show scripts (see [Using the OnDelete script event](#)).
- **OnInitialize** – Called when the Initialize button is pressed or an initialize macro command is executed.
- **OnMove(OldName, NewName)** – Called when a page is moved or renamed. The event is only available to show scripts.
- **OnPropertyFocused(PropertyName)** – Called when a tab-field editor gets focus in the Viz Trio user interface. It sends the property name into the function that uses the event.
- **OnRead()** – Called when the Read button is pressed or a read macro command is executed.
- **OnSave** – Called when the Save button is pressed or a save macro command is executed.
- **OnSocketDataReceived** – Called when socket data is received on the defined port. A socket object must be configured on the configuration page.
- **OnTake()** – Called when the Take button is pressed or a take macro command is executed.
- **OnTakeOut** – Called when the Take Out button is pressed or a take out macro command is executed.
- **OnUpdate()** – Called when an update action is executed in Viz Trio (See keyboard shortcut list on control page).
- **OnUserClick()** – Called when the user clicks the Run macro button or when the Viz Trio macro command `execute_script` is executed.

- **OnValueChanged(PropertyName,NewPropertyValue)** – Used with template scripts. The event is called when a property in the page is changed. It sends in the name of the property and the new value.
- **OnValueChanged(PageName, PropertyName,NewPropertyValue)** – Used with show scripts to specify the page (PageName) that is called. The event is called when a property in a show's page is changed. It sends in the name of the page, the name of the property and the new value.

### Using the OnDelete script event

The following script will cancel the delete operation of any page named 2000.

```
function OnDelete(PageName)

  if PageName = "2000" then
    OnDelete = false
  else
    OnDelete = true
  end if

end function
```

The following are examples of commands that will trigger this event:

- [gui:copy\\_selected\\_pages\\_to\\_number](#)
- [gui:copy\\_selected\\_pages\\_with\\_offset](#)
- [gui:move\\_selected\\_pages\\_to\\_number](#)
- [gui:move\\_selected\\_pages\\_with\\_offset](#)
- [show:copy\\_pages\\_to\\_number](#)
- [show:copy\\_pages\\_with\\_offset](#)
- [show:delete\\_all\\_pages](#)
- [show:delete\\_templates](#)
- [show:move\\_pages\\_to\\_number](#)
- [show:move\\_pages\\_with\\_offset](#)
- [show:rename\\_page](#)
- [page:delete\\_page](#)
- [page:delete\\_pagerange](#)

### See Also

- [Scripting](#)
- [Commands](#)

---

## 12 Designer

There are two ways of doing scene design for Viz Trio. One is using Viz Artist, which in most cases is the preferred way, and another is using the built-in designer. The Designer enables you to use pre-made scene elements or building blocks for creating a standalone scene.

-----  
**Note:** Transition logic is not supported.  
-----

This section contains information on the following topics:

- [About Viz Trio Designer](#)
- [Creating Scene Elements in Viz Artist](#)
- [To start the Designer](#)

### To start the Designer

- Click the Design button in the upper right corner of Viz Trio's playout interface.

### See Also

- Viz Artist User's Guide

---

### 12.1 About Viz Trio Designer

Design

Viz Trio Designer will display the designer's library of pre-defined scene elements. The scene elements are created in Viz Artist and includes the required control plug-ins for exposing the scene properties the operator can adjust. The user can easily group the different scene elements together in order to create more advanced scene. In the Designer it is also possible to modify the scene element's properties, provided that they have been exposed for editing (ref. Control Plug-ins).

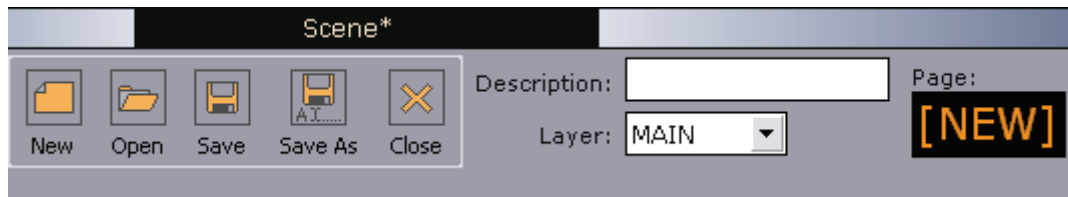
This section contains information on the following topics:

- [Scene](#)
- [Resources](#)
- [Scene Tree](#)
- [Tab Fields](#)
- [Page Editor](#)
- [Properties](#)

### See Also

- Viz Artist 3.3 User's Guide or later on Control Plug-ins

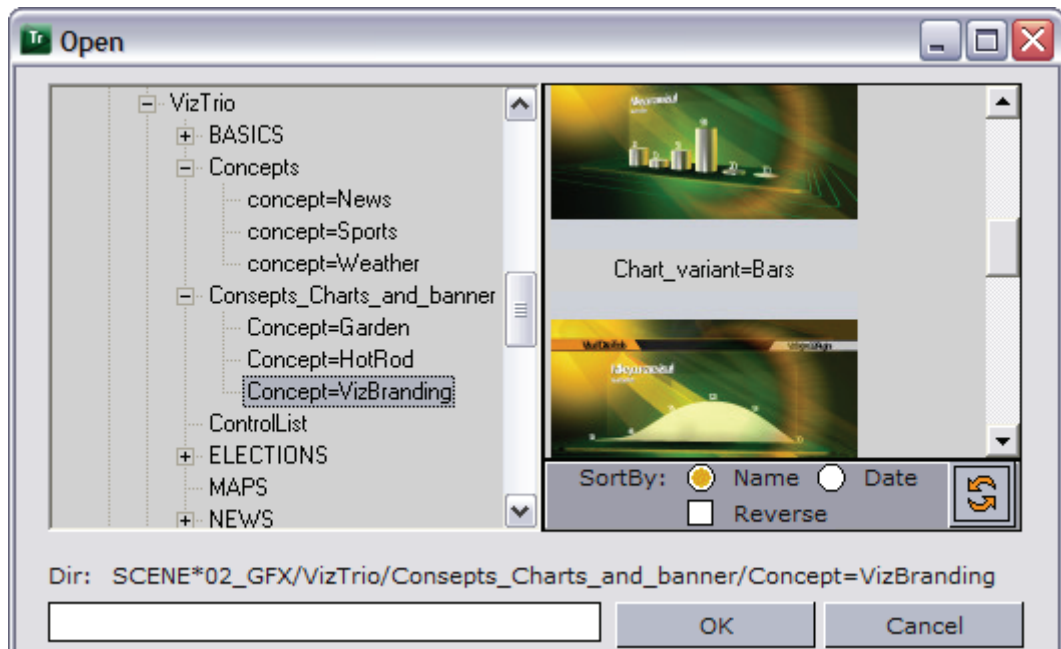
## 12.1.1 Scene



The scene pane is located in the upper left corner of the Designer. It contains functions for scene management and control, such as New, Open, Save, Save As, and Close buttons.

- **New Scene** – Click the New button to create a new scene.

**Figure 68:** Viz browser



- **Open Scenes** – Displays the Viz browser, which provides access to the Viz scene database. Each scene is represented as an icon reflecting the appearance of the scene as it was when last saved. The scene icons are displayed to the right in the scene database pane. The scenes are organized in folders which can be accessed through the folder tree to the left. The tree structure is the same as available in the SCENE folder in the Viz Artist data directory. The scenes can be sorted by Name or Date. The Refresh button refreshes the content for the selected folder.
- **Save** – Saves changes applied to the currently selected scene.
- **Save As** – Saves changes applied to the currently selected scene to a new name.
- **Close** – Closes the currently selected scene. If the scene contains unsaved changes, the user will be prompted for a confirmation.
- **Description** – Can be used to enter a description for the scene which will be displayed in the Template Description column in Viz Trio.
- **Layer** – Allows the renderer layer for the new scene to be specified. The graphics can be played in the FRONT, MIDDLE and BACK layer. The MIDDLE

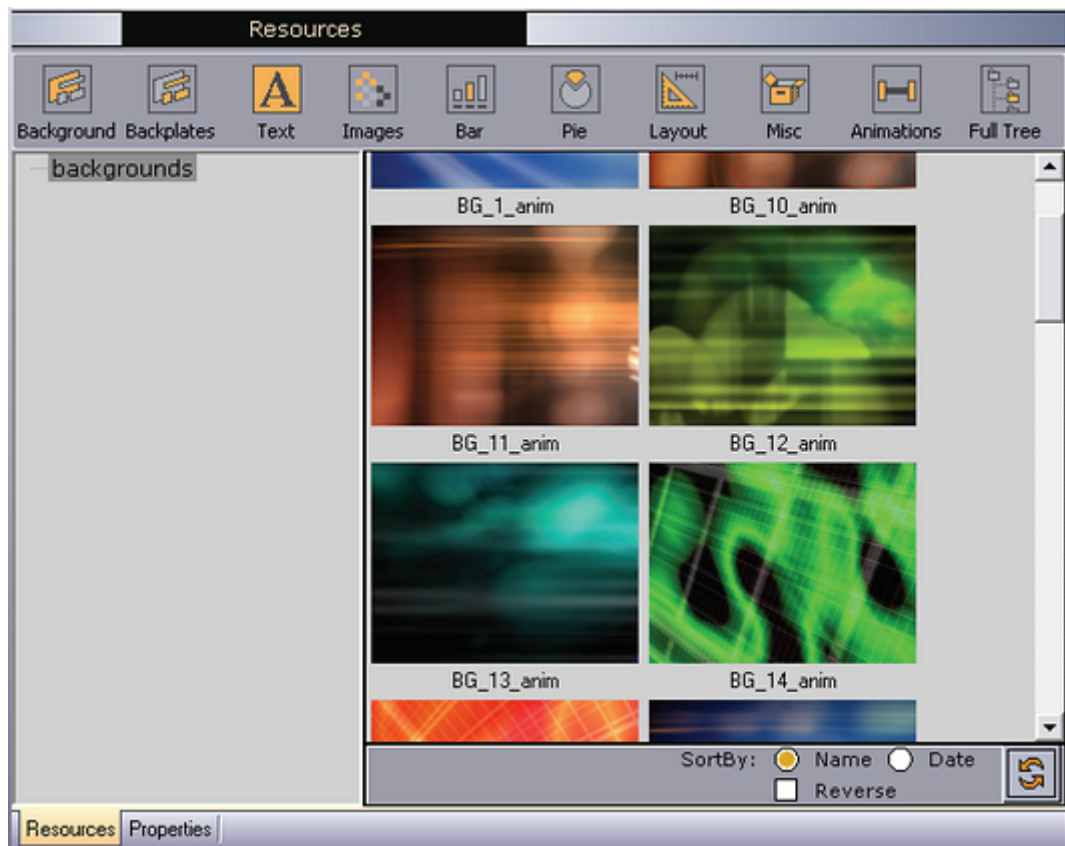
layer, also called MAIN, is the default layer. By using different layers it is possible to have up to three elements on screen simultaneously. However, if they are positioned at the same X and Y positions, graphical elements in the front layer will cover graphical elements in the middle and back layers. Likewise, graphical elements in the middle layer will obscure graphics in the back layer.

- **Callup Code** – Displays the callup code of the currently selected page. New pages, that have not been saved, will have the page name **[NEW]**. Click the **Save As** button to save it with a specific callup code.

### See Also

- Viz Artist 3.3 User's Guide on transition logic

## 12.1.2 Resources



**Note:** The Library path is set in [Show Settings](#), and should be named LIBRARY.

The Resources pane contains all the scene elements available to Viz Trio, and is automatically displayed when opening the Designer. The scene elements are created in Viz Artist, and include the required control plug-ins.

Scene elements are basically scenes created in Viz Artist, however, they are not standalone scenes. They are made more like building blocks such that they can be pieced together with other scene elements in order for a Viz Trio operator to create a standalone scene on the fly.



Scene elements are organized under a `LIBRARY` folder in Viz with the following sub folders; `BACKGROUNDS`, `BACKPLATES`, `TEXT`, `IMAGES`, `BARS`, `PIES`, `LAYOUT`, `MISC` and `ANIMATIONS`. The various folders become available when selecting the corresponding categories using the buttons in the Resources pane.

In addition to the pre-defined scene elements, it is possible to use ordinary Viz Artist scenes, if a group with a placeholder has been added to it, when creating new scenes in Viz Trio's Designer. The Viz Artist folder tree is displayed when clicking the **Full Tree** button.

Each scene element is represented by an icon in the right part of the Resources pane. The scene elements in each category are organized in folders, accessible through the folder tree to the left. The tree structure is the same as the `SCENE` folder tree in the Viz Artist data directory. Browse the folder tree to locate the scene elements to use. Scene elements can be sorted by Name or Date and in reverse order. Click the Refresh button to update the content of the selected folder.

Also, the storage location distinguishes scene elements from ordinary standalone scenes as the scenes must be stored in the designated `LIBRARY` sub-folders. This will ensure that they become available as scene elements in the Designer interface, see [To create a new scene element](#) for further details. Furthermore, the scene elements contain control plug-ins that enable the user to edit exposed properties and organize them in parent-child hierarchies.

---

**IMPORTANT!** Standalone scenes cannot be edited in the Designer. They can only be used as is, and must be added to separate placeholders in the Scene tree. Furthermore, standalone scenes cannot parent other scene elements.

---

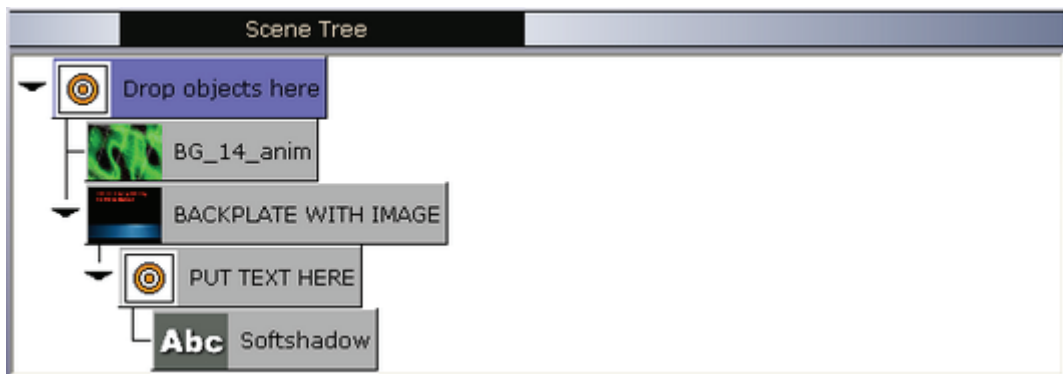
- **Background** – Shows the background scene elements available in `BACKGROUNDS` library folder. Select a background, and add it to the scene by drag and drop onto the root placeholder.
- **Backplate** – Shows the backplate scene elements available in the `BACKPLATES` library folder. Select a backplate, and add it to the scene by drag and drop onto the root placeholder.
- **Text** – Shows the fonts scene elements in the `TEXT` library folder. Adding text to a scene adds a text tab-field that is used by the operator for editing in the Page Editor. Font scene elements may have special effects such as alpha, mask, and shadow. Select a font scene element, and add it by drag and drop onto the appropriate placeholder. This will automatically enable the tab-field for text editing.
- **Images** – Shows the images available in the `IMAGES` library folder. As with the Text button, the Image button adds a tab-field for editing by an operator. Clicking the Image button opens the Images library where an image can be selected and added by drag and drop onto the appropriate placeholder.
- **Bar** – Shows the bar scene objects available in the `BARS` library folder. Select a bar scene element, and add it to the scene by drag and drop onto the appropriate placeholder. To create a scene with several bars a layout scene element should be created that can hold the individual bar scene elements.
- **Pie** – Shows the pie scene objects available in the `PIES` library folder. Select a pie, and add it to the scene by drag and drop onto the appropriate placeholder.
- **Layout** – Shows the layout scene elements available in the `LAYOUT` library folder. The folder contains design elements which allow for easy alignment of

other scene elements, by organizing them in for instance lines or rows. The layout element themselves usually do not contain any graphics. They should ideally function as placeholders in which other types of scene elements can be added. Select a layout, and add it to the scene by drag and drop onto the root placeholder.

- **Miscellaneous** – Shows the miscellaneous scene elements available in the `MISC` library folder. These can be anything you decide to put there that does not fit in the other categories, such as 3D objects, bullets, and placeholders. Select a scene element, and add it to the scene tree by drag and drop onto the appropriate placeholder.
- **Animation** – Shows the animation scene elements in the `ANIMATIONS` library folder. Add an animation to the scene by drag and drop onto the placeholder holding the scene element to animate.
- **Full Tree** – Shows a full Viz directory structure that provides an overview of all available scenes.

### 12.1.3 Scene Tree

**Figure 69:** Placeholders containing objects



The scene tree provides a logical visualization of all the graphical elements in a scene. The tree consists of placeholders containing the different types of scene elements.

A standalone scene is built by selecting various scene elements from the Resources libraries and dragging them onto the scene tree. In the scene tree it is possible to create a hierarchy between elements and groups of elements in logical divisions. This allows for new properties to be assigned to more than one scene element in a single operation.

The black arrow to the left indicates that the placeholder has sub-containers. If the arrow points to the right the sub-containers are not displayed. Click on the arrow to expand the sub-containers. The arrow will then point downwards.

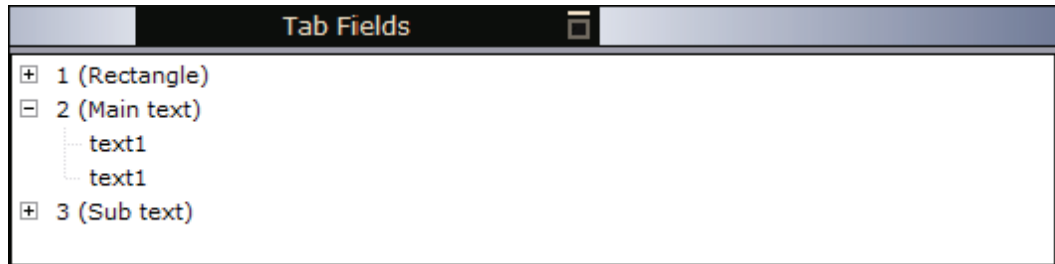
#### To delete an item in the scene tree

- Select the appropriate scene element and then press the **Delete** key. Note that this will also delete sub nodes.

-----  
**CAUTION!** You cannot undo a delete operation.  
-----

## 12.1.4 Tab Fields

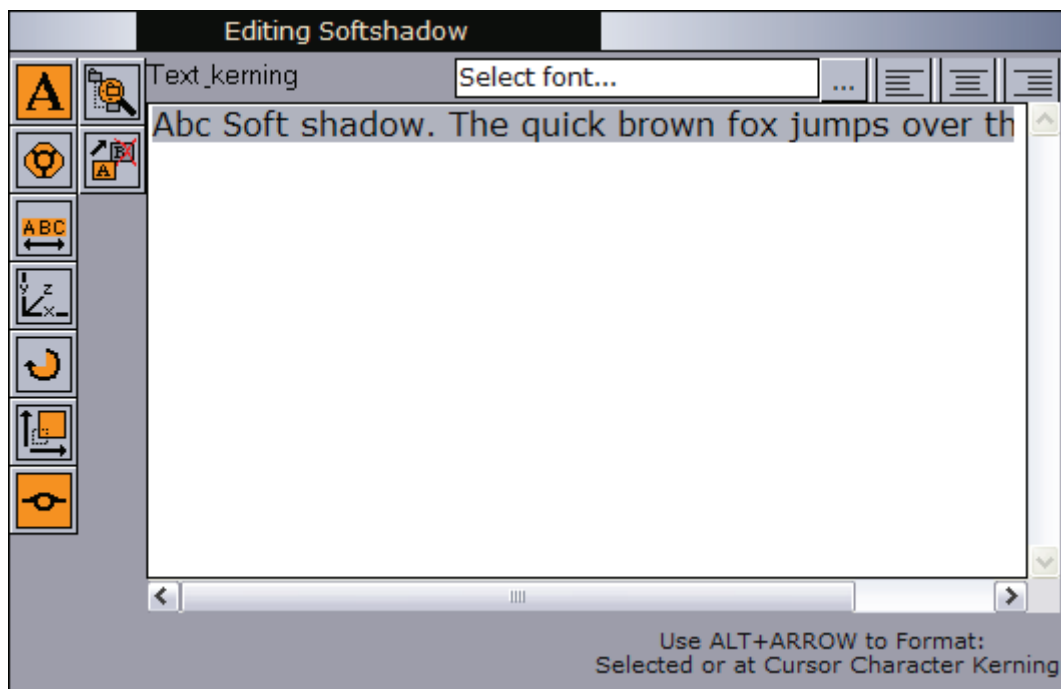
Figure 70: Tab fields



Scene elements can have one or more tab fields. When selecting a scene element in the [Scene Tree](#), the associated tab fields are displayed in the tab fields pane in the lower left corner of the designer tool. In the tab fields pane it is possible to see which [Properties](#) that can be exposed for editing in Viz Trio. Click the '+' sign to expand a tab field's list of exposable properties.

## 12.1.5 Page Editor

Figure 71: Page editor

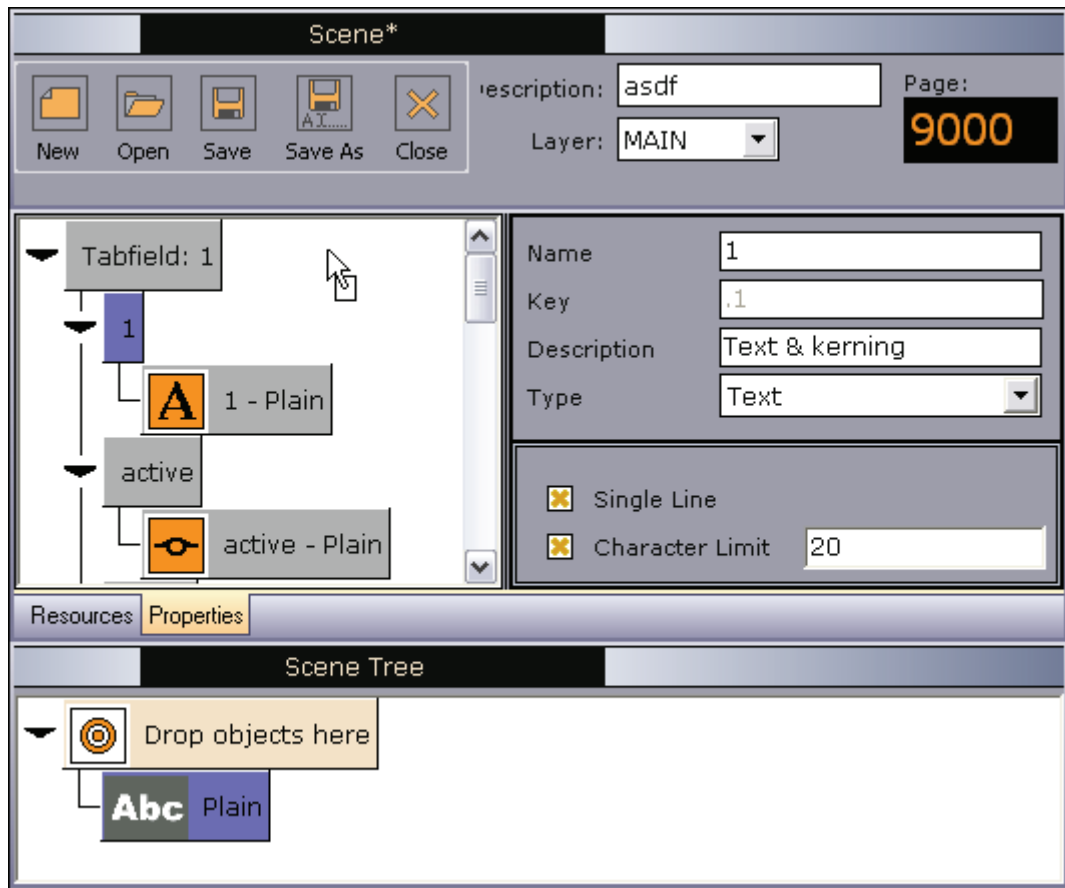


When selecting a tab-field from the [Tab Fields](#) pane, the corresponding page editor is displayed in the upper right corner of the Designer. Use the page editor to edit the selected tab field's properties, like for instance changing material, position, rotation, text, and so on, provided they have been exposed for editing.

The type of page editor displayed varies according to the type of element selected, e.g. text, images, etc. The buttons in the left part of the page editor area represent the various exposable properties of the currently selected tab field. To display the corresponding editor, click on the appropriate button or select the property in the [Tab Fields](#) pane.

## 12.1.6 Properties

Figure 72: Property editors and scene tree editor



The Properties pane becomes available when selecting the Properties tab in the lower left corner of the Resources pane.

Use the Properties pane to define which **Tab Fields** should be exposed for editing in Viz Trio. The properties pane consists of three parts: a **Property Tree**, a **Property Description Editor**, and a **Property Type Editor**.

This section contains information on the following topics:

- [Property Tree](#)
- [Property Description Editor](#)
- [Property Type Editor](#)
- [To expose properties](#)
- [To map multiple property receivers to one exposed property](#)
- [To remove properties](#)

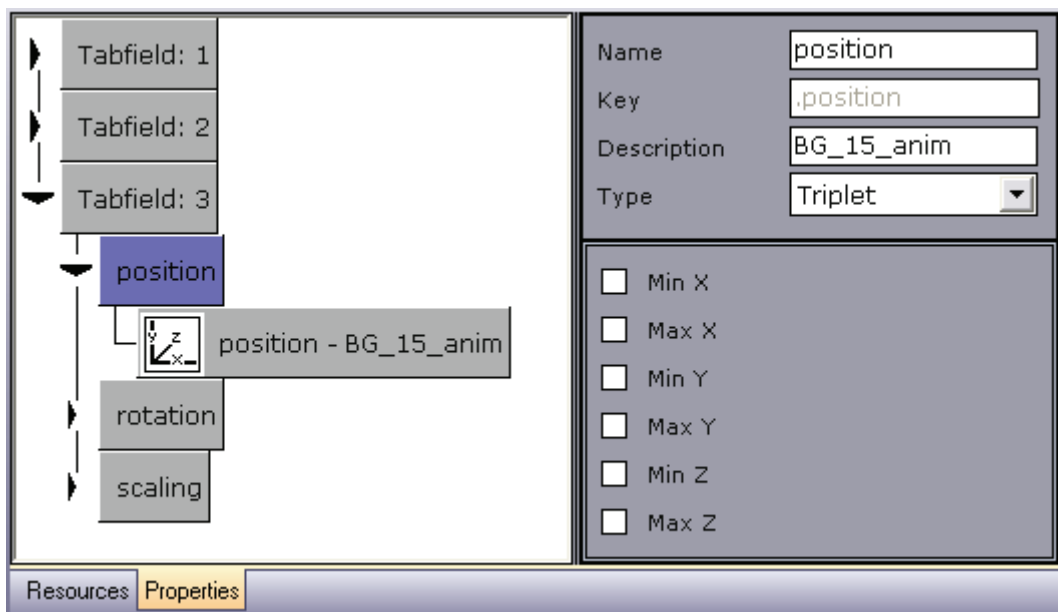
### Property Tree

Initially, when creating a new scene, the properties pane is empty. Tab field properties are exposed for editing by adding them to the properties pane (top left).

The Viz Trio Designer provides two different approaches [To expose properties](#): exposing properties in a tab-field one by one, or exposing all properties in one operation.

Use the properties pane to define tab field properties that should be exposed for editing in Viz Trio. The [Tab Fields](#) pane provides a list of properties that can be exposed for editing for each of the tab fields in a scene. Expose as many of them as needed for editing.

A Viz Artist designer, that creates the resources, can define the properties to be exposed for editing when creating the scene element in Viz Artist. Exposed properties may vary from scene to scene. Note that additional properties must be exposed through Viz Artist.

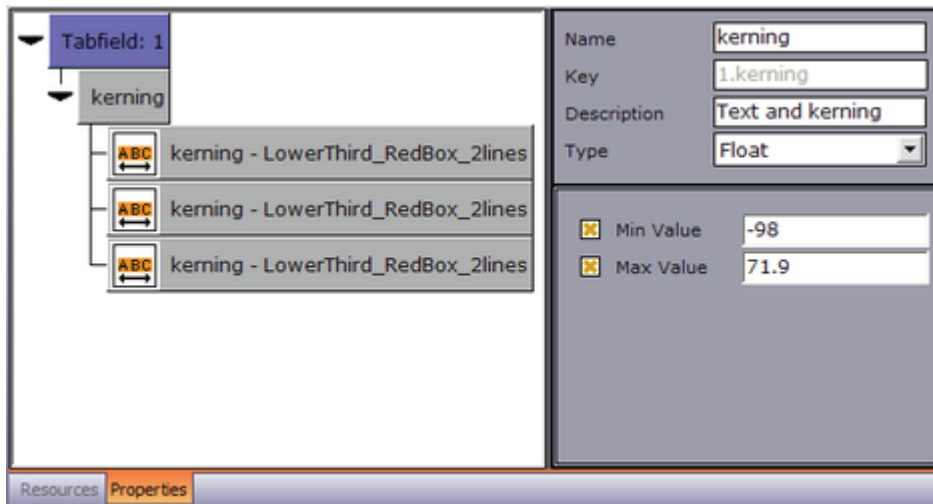


The nodes in the properties tree are named *Tabfield: 1*, *Tabfield: 2*, *Tabfield: 3*, and so on as they are added. The order of the tab fields can be rearranged by drag and drop.

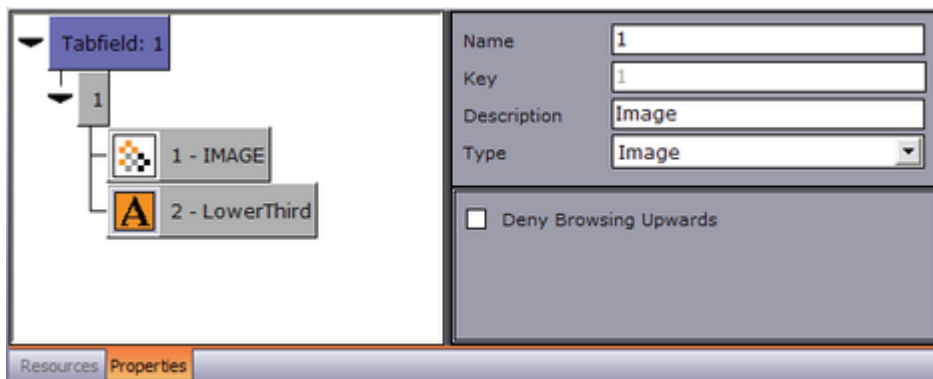
Appended to each tab field are sub-nodes representing the exposed properties (e.g. position and rotation). The sub-nodes have child nodes that represents the receiver of the exposed property's data value, for example the selected scene element. The child node typically displays a property name and a symbol, in addition to the receiving scene element's name (e.g. background).

It is possible to expose several properties for one tab field by dragging the property icons onto the same tab field container from the [Page Editor](#).

It is also possible to map two or more property receivers to one exposed property. This will control the value of several property receivers through one exposed property. This is for instance useful if wanting to change a property, such as kerning or material, for several scene elements in a single operation. This is illustrated in the example below, where the exposed property *kerning* controls the kerning value of the three attached property receivers called *LowerThird\_RedBox\_2lines*.

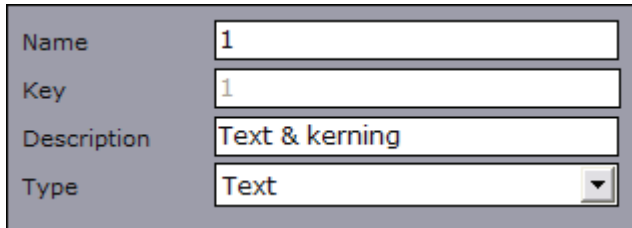


Furthermore, property receivers of various types, like for instance an image and a text object, can be mapped to the same exposed property, even when located in different tab fields. This will allow edits to all the associated property receivers in one single operation. In the example below, the exposed property *Image*, controls the content of the associated image and text property receivers. This implies that when selecting a new image (see [Search Media](#)), the text related to the image will be added accordingly.



Another example is when connected to a newsroom system and changes are made to a text and an image in the same tab field by entering a text string instead of the whole path. This implies that there is a folder of images (e.g. flags) and the Viz Trio system receives 'US' from the external system, it will then load the image with the path: 'image location prefix + US', provided that the image location prefix has been specified.

## Property Description Editor



The screenshot shows a dialog box with four rows. The first row is 'Name' with a text input field containing '1'. The second row is 'Key' with a text input field containing '1'. The third row is 'Description' with a text input field containing 'Text & kerning'. The fourth row is 'Type' with a dropdown menu showing 'Text' and a small downward arrow on the right.

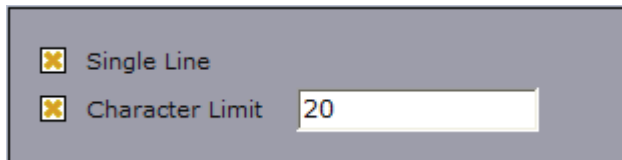
The Property Description editor is displayed in the upper right corner of the Properties pane. It is used to edit the description parameters of a tab field. The fields become available for editing when selecting a property in the [Property Tree](#). All fields, apart from the Key field, are editable.

- **Name** – Shows the name of the selected tab field property.
- **Key** – Displays the key identifier of the property.
- **Description** – Shows the description for the selected tab field. The description is displayed in the [Page List](#).
- **Type** – Shows the data input type defined for the selected property. The value selected in the Type field determines which [Page Editor](#) will be displayed.

## Property Type Editor

The Property Type editor is closely linked to the value selected in the Type field in the [Property Description Editor](#). When selecting a value in the Type field, the fields in the Property Type editor will be updated accordingly, such that values for the associated parameters can be specified.

**Figure 73:** Text

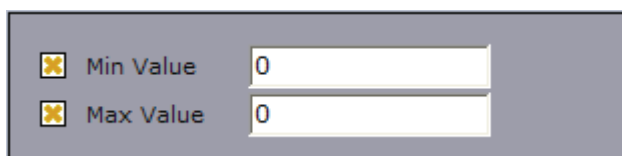


The screenshot shows a dialog box with two rows. The first row has a checked checkbox (indicated by an 'x') followed by the text 'Single Line'. The second row has a checked checkbox followed by the text 'Character Limit' and a text input field containing the number '20'.

Allows the designer to set text parameters.

- **Single Line** – Disables text wrapping displaying the text as a single line.
- **Character Limit** – Allows for setting a maximum number of characters for the selected text tab field.

**Figure 74:** Float

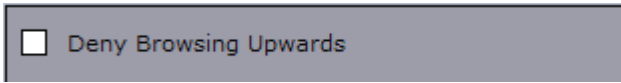


The screenshot shows a dialog box with two rows. The first row has a checked checkbox followed by the text 'Min Value' and a text input field containing the number '0'. The second row has a checked checkbox followed by the text 'Max Value' and a text input field containing the number '0'.

Allows the designer to specify a minimum and a maximum value for the selected property. Select one or more of the check boxes to display the associated input fields.

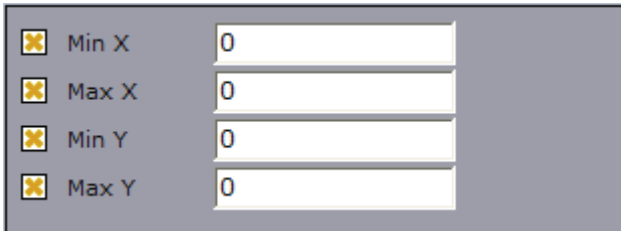
- **Min Value** – Sets the minimum input value for the selected property.
- **Max Value** – Sets the maximum input value for the selected property.

**Figure 75:** Deny browsing



Allows the designer to deny the Viz Trio user from browsing for images, geometries, and materials in folders other than the selected folder and its sub-folders.

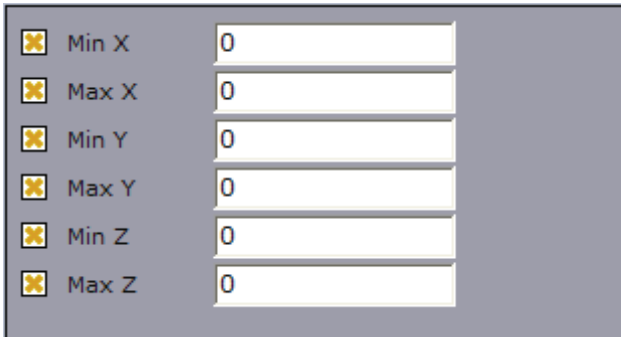
**Figure 76:** Duplet



Allows the designer to specify a minimum and a maximum value for the X and Y axes of the selected property. Select one or more of the check boxes to display the associated input fields.

- **Min X** – Sets the minimum value for the X axis of the selected property.
- **Max X** – Sets the maximum value for the X axis of the selected property.
- **Min Y** – Sets the minimum value for the Y axis of the selected property.
- **Max Y** – Sets the maximum value for the Y axis of the selected property.

**Figure 77:** Triplet



Allows the designer to specify a minimum and a maximum value for the X and Y axes of the selected property, as well as the Z axis. Select one or more of the check boxes to display the associated input fields.

- **Min X** – Sets the minimum value for the X axis of the selected property.
- **Max X** – Sets the maximum value for the X axis of the selected property.
- **Min Y** – Sets the minimum value for the Y axis of the selected property.
- **Max Y** – Sets the maximum value for the Y axis of the selected property.
- **Min Z** – Sets the minimum value for the Z axis of the selected property.
- **Max Z** – Sets the maximum value for the Z axis of the selected property.

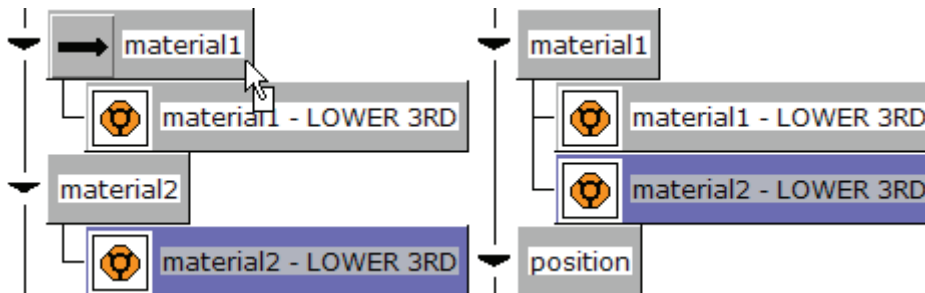
### To expose properties

1. In the [Page Editor](#), select the icon or design element representing the property to expose for editing, and drag and drop it onto the [Property Tree](#), or



2. Drag and drop a container from the [Scene Tree](#) onto the [Property Tree](#) to generate a complete list of exposed properties.

#### To map multiple property receivers to one exposed property



- Drag and drop the property receiver onto the exposed property that should control both properties.

#### To remove properties

- Select a property and press the Delete key on the keyboard. Note that this will also delete sub-nodes.

---

## 12.2 Creating Scene Elements in Viz Artist

This section provides a basic introduction on how to create scene elements in Viz Artist. Creating scene elements basically consist of two main activities; creating the graphics and adding different Control plug-ins. Control plug-ins are the binding links between Viz Artist and Viz Trio that allow properties to be exposed to the Viz Trio operator.

Scene elements can be of various complexities with one or more tab fields, different types of animations, and so on. The scene elements may also include placeholders which are basically non-graphical objects. These can in turn hold other scene elements, like for instance text objects.

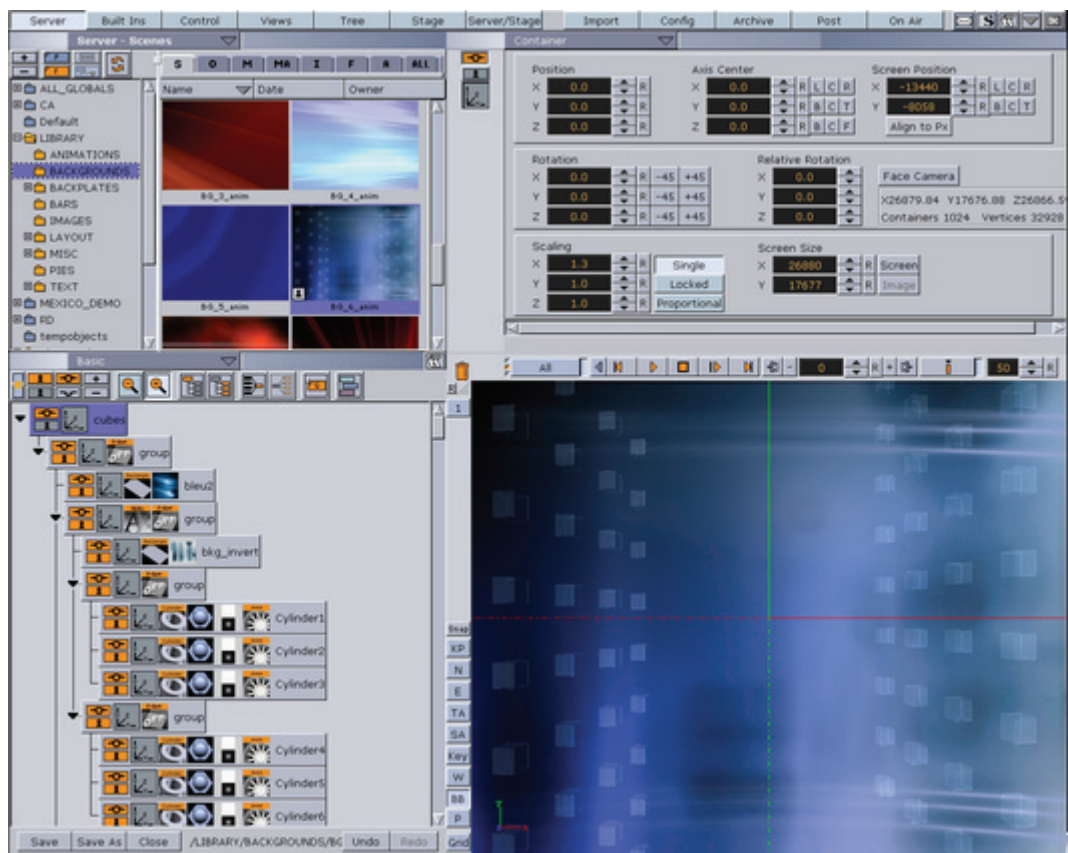
A brief introduction to the Viz Artist GUI is provided initially. Only functions and features required to create scene elements will be addressed in this chapter.

#### See Also

- [Viz Artist User Interface](#)
- [Creating Graphics](#)
- [Adding Control Plug-ins](#)
- [Creating Backgrounds](#)
- [Creating Backplates](#)
- [Creating Text Objects](#)
- [Creating Image Objects](#)
- [Creating Animations](#)
- [Viz Artist User's Guide](#)

## 12.2.1 Viz Artist User Interface

Figure 78: Viz 3.x



Open Viz Artist by clicking the Viz Artist button in the upper right corner of Viz Trio. The Viz Artist user interface at startup displays the following:

- **Resources** – Contains all the different objects stored on Viz Graphic Hub (Viz 3.x) or on the data root (Viz 2.x). The resources are available in the upper left corner of Viz Artist under the Server tab.
- **Editors** – Allows the user to modify the properties of an object. The various editors are displayed in the upper right corner.
- **Scene tree** – Provides an overview of all the elements in a scene and their hierarchy. The scene tree is displayed in the lower left corner.
- **Preview** – Displays a WYSIWYG representation of the graphics and what it will look like when on-air. The renderer is located in the lower right corner.

Interaction with the scene builder tools is generally based on drag and drop actions.

## 12.2.2 Creating Graphics

Create scene elements by using the various building blocks available in the Viz Artist database. The building blocks are basically placeholders which can hold different types of graphical elements and/or plug-ins. They are thus also known as containers.

Build the graphics by organizing the containers hierarchically in the scene tree pane and adjust the associated properties in the corresponding property editors.

The tree structure constitutes a parent–child hierarchy in which all child containers will inherit properties of the parent container.

Selection of the building blocks to use depends on the type and complexity of the design. The following sub sections will provide a basic description on how to create scene elements in each of the various categories. Some steps are common for all categories, like for instance how to create a new scene and add a group container. These procedures are addressed in the initial sections, while steps specific to each of the categories are described in separate sections.

This section contains information on the following topics:

- [To create a new scene element](#)
- [To add group containers](#)
- [To add geometries](#)
- [To edit scale and position](#)
- [To copy containers](#)
- [To delete containers](#)
- [To add materials](#)
- [To add fonts](#)
- [To add images](#)
- [To add key](#)
- [To define tab fields](#)
- [To save scenes](#)

### To create a new scene element

1. Start Viz Artist
2. Select the Scene view.
  - *For Viz Artist 3.x:* Click the **Server** button and select Scene from the drop–list in the upper left corner of the Database window, og click the **S** tab.
  - *For Viz Artist 2.x:* Click the **Scene** button in the upper left corner of the Database window.
3. Locate and expand the `LIBRARY` folder in the folder tree.
  - The sub–folders that appear correspond to the various library categories available in the Designer and contains the different scene elements created by the designer.
  - Note that scenes must be stored in the designated `LIBRARY` sub–folders in order to be available as scene elements in the Designer’s Resources pane.
4. Select the appropriate sub–folder (e.g. `BACKGROUNDS`).
5. Create a new scene.
  - *For Viz Artist 3.x:* Select **Create...** from the Scene database drop–menu, or press the CTRL+A keys.
  - *For Viz Artist 2.x:* Click the **Add...** button in the lower left corner of the Scene database pane.
6. Enter a name for the scene in the appearing dialog box, and click OK.
  - The new scene becomes available in the right part of the Server view.

### To add group containers



The Group function is a container which allows for elements to be grouped in the scene tree, and thereby create a parent-child hierarchy in which the child containers inherit the properties of its parent(s).



1. Open the scene (see [To create a new scene element](#)).
2. Add a group container to the scene tree.

### To add geometries

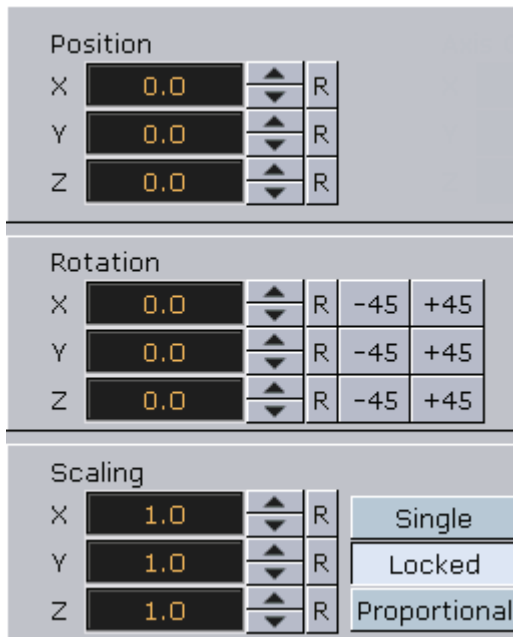


1. Add a geometry object (e.g. Rectangle), and add it as a sub-container to the group container in the Scene Tree.
2. Rename the rectangle's sub-container to for example *background* in order to describe its function.

### To edit scale and position



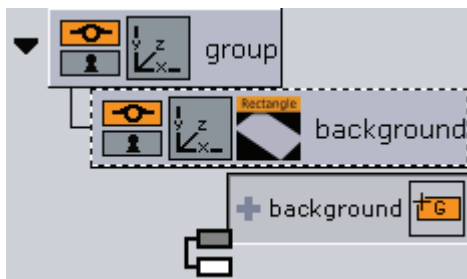
1. Display the Transformation editor by selecting the Transformation icon on the container.



2. Modify the objects's size and position.
  - The size and position of an object depends on the type of scene element you want to create. For example a lower third should cover the lower third of the screen, while an over the shoulder should cover parts of the screen over the should of the anchor/presenter.
3. Specify the order of layered objects by using the Z field in the Position section.
  - Some objects must be displayed behind the other elements in a scene, like for instance a background or a backplate, in order not to cover for elements which should be displayed in the front.
  - Keep in mind when creating scene elements that the object with the highest z-position will be displayed in the front, while the object with the lowest z-position will be displayed in the back (unless you are rotating objects on the y-axis which in case you might need to manually sort objects on the Z-axis using the Z-sort plug-in).

### To copy containers

1. Press the CTRL key and the left mouse button simultaneously and drag the container to its new location.



- *For Viz Artist 3.x:* A node symbol will appear to the left of the copied container, indicating the position which the new container will be added when releasing the mouse button.

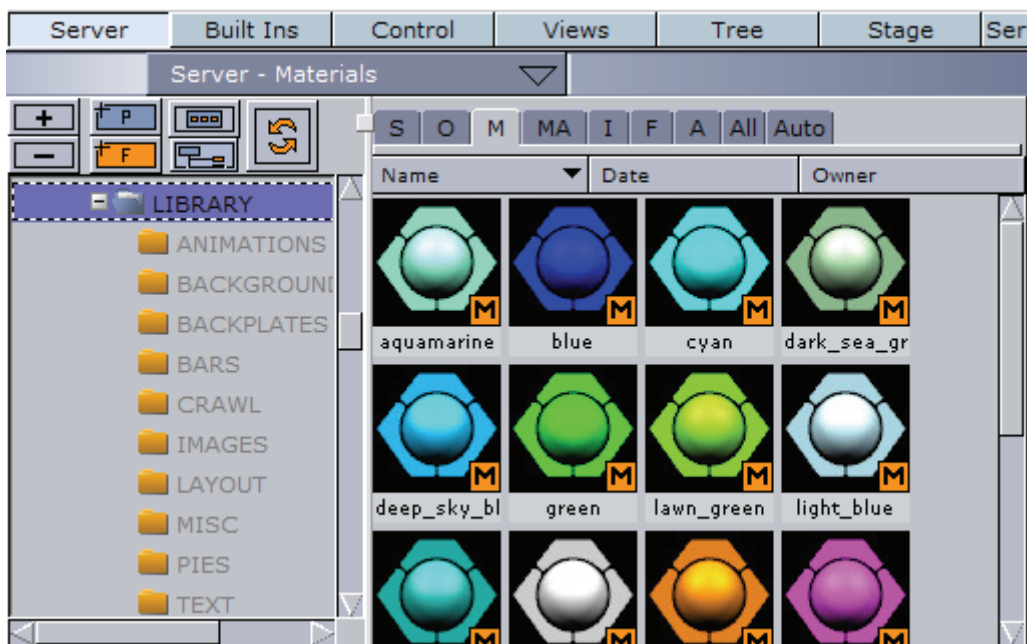
- *For Viz Artist 2.x:* An arrow will appear to the very left of the scene tree, indicating the position in which the new container will be added when releasing the mouse button. Release the mouse button when at the appropriate position.
2. To copy multiple containers, press the CTRL key and select the containers with the left mouse button. Drag the containers to the new location and release the mouse button.

### To delete containers



1. Select the container(s) to delete with the left mouse button and drag it/them onto the trash bin above the Renderer window, or
2. Right-click the container, and select **Delete Container <name>** from the appearing context menu (*Viz Artist 3.x only*).

### To add materials



1. Select the material (M) tab under the server view (Material button in Viz Artist 2.x), and drag and drop a material object onto the container.

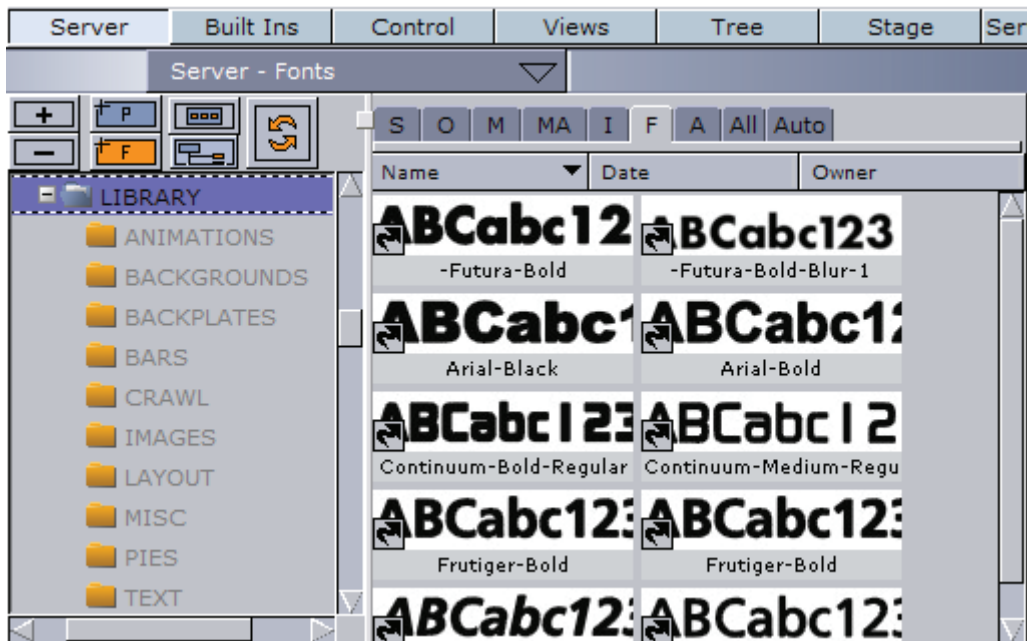


2. Double-click the material to open the Material Editor to change the effects, or types of lights, of the material (e.g. Ambient, Diffuse, Specular and Emission).

#### See Also

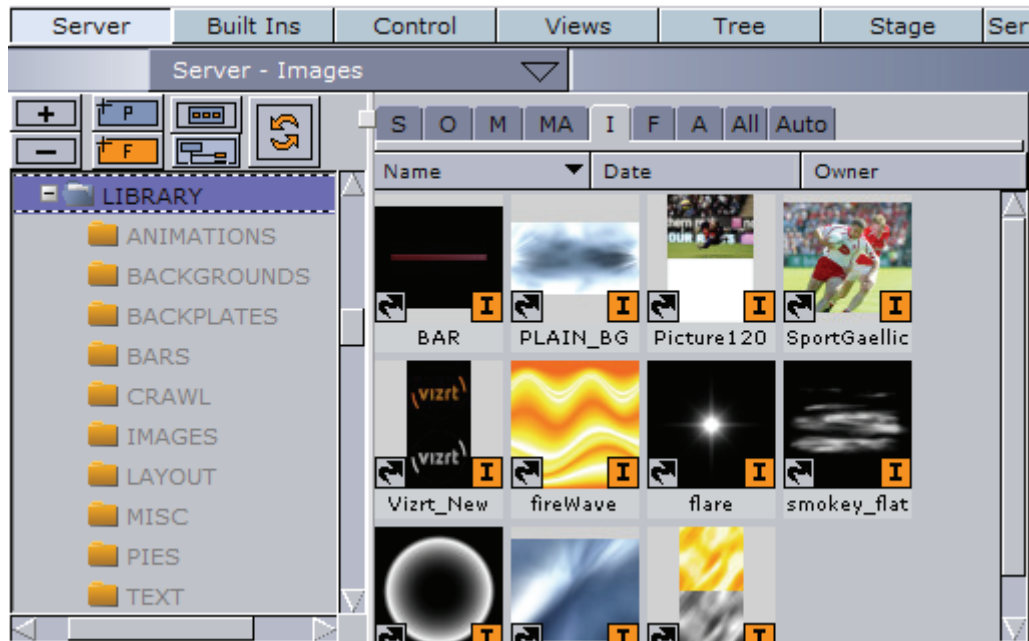
- Viz Artist User's Guide

#### To add fonts



1. Select the font (F) tab under the server view (Font button in Viz Artist 2.x), and drag and drop a font onto the container.
2. Open the transformation editor to change the size and position of the text.

## To add images

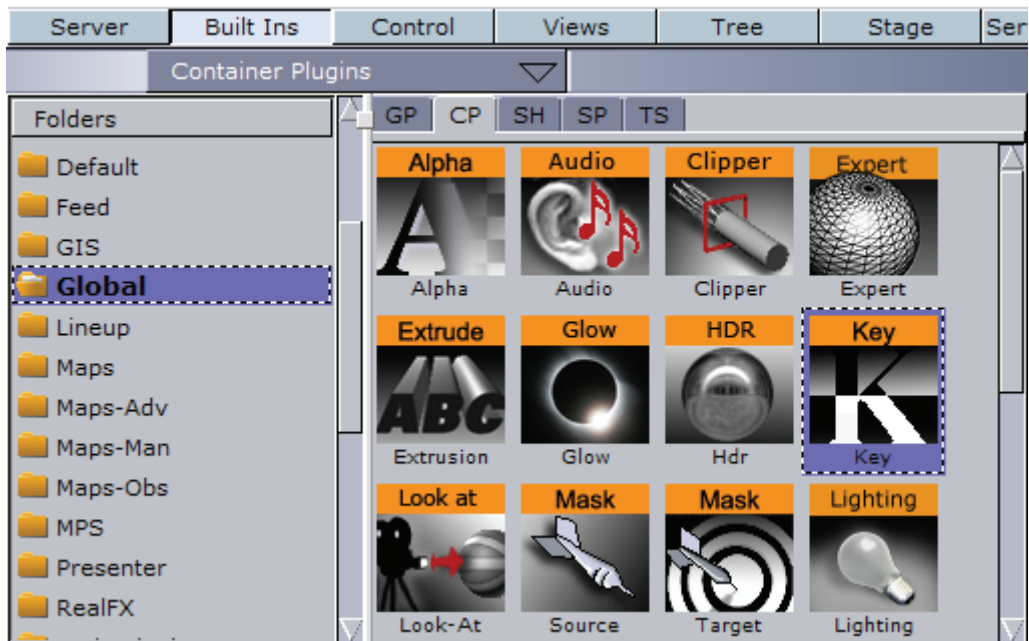


1. Select the image (I) tab under the server view (Image button in Viz Artist 2.x), and drag and drop an image onto the container.
2. Open the transformation editor to change the size and position of the text.

## To add key

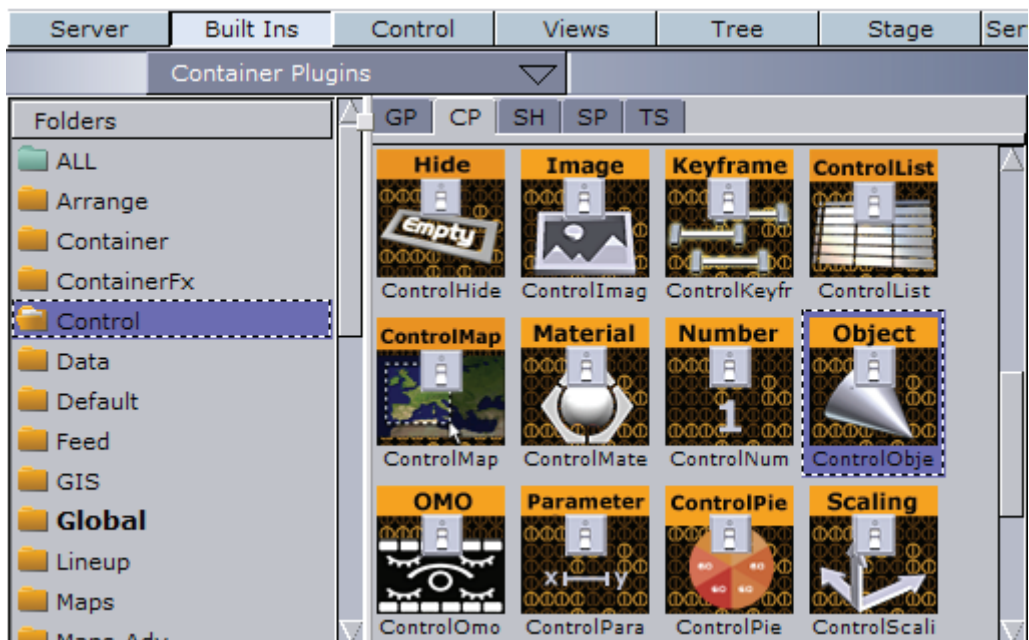
In order to enable Viz Engine to produce a key signal, key must be added to individual containers that hold objects that should draw key or the scene in general.





1. Click the Built Ins button, and from the Container Plugins (CTRL+2) select the Global folder.
  - For Viz Artist 2.x: Click Function.
2. Drag and drop the Key plugin onto the containers that should have key.

### 12.2.3 Adding Control Plug-ins



In order to enable Viz Trio operators to edit the graphical elements, the containers must hold the necessary control plug-ins. These are container plug-ins that must be added for each of the graphical elements that should be editable. Which control plug-in to choose, depends on the type of property that needs to be exposed for editing.

In addition to the container specific plug-ins, the scene tree must have a plug-in called the Control Object plug-in. It is actually by means of the Control Object plug-in that Viz Trio is able to identify a scene as a Viz Trio template. Note that only containers placed under the Control Object plug-in in a scene tree can be imported to Viz Trio.

The control plug-ins are added manually to the containers using drag and drop. The interaction principles are the same for all control plug-ins except the Control Object plug-in that is added automatically if another control plug-in is added to the scene tree. Control Object can also be added manually.

### To add a control plug-in

1. Click the Built Ins button, and select the Container Plugins (CTRL+2) option from the drop-menu or simply click the CP tab.
  - For Viz Artist 2.x: Click Function, Container and select the Control folder.
2. Select a control plug-in and add it to the appropriate container.
3. Click on the control plug-in icon to display the associated editor.

### To define tab fields

Viz Trio uses the control plug-in's Field Identifier ID to identify the container as an editable element. Furthermore, the field's ID defines a tab-field, and is used to specify the tab-order between the editable elements of a scene.

All control plug-ins located in the same container must have the same field ID. This will ensure that when choosing a tab-field, all associated property editors are enabled for selection.

To group properties located in different containers to the same tab-field, the property name must be added as an extension to the field ID, using the following format: *ID.propertyname*, like for instance *2.scaling*.

1. Add a control plug-in (e.g. Control Text) to the scene tree.
2. Open the control plug-in's editor and set the **Field Identifier**.
  - The field identifier should be a numeric value (range 1-n).

### To save scenes

Options for saving and closing scenes are provided in the lower left corner of the Viz Artist GUI.



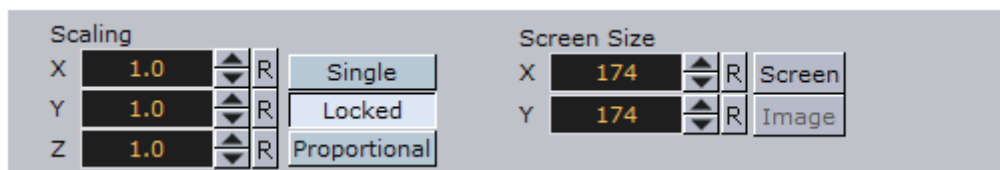
1. Click the **Save** button to save a new scene or to overwrite an open scene, or
2. Select the **Save As** button to save it as a new scene.

## 12.2.4 Creating Backgrounds

This section is a quick tutorial on how to create a basic background with an image.

**Note:** Screenshots used are from Viz Artist 3. The user interface in Viz Artist 3 is similar to Viz Artist 2. Where needed additional explanation is given.

1. Start by creating a new scene in the `BACKGROUNDS` directory in the Viz Artist scene tree (see how [To create a new scene element](#)).
2. Double-click on the new scene to open it, and add a group container to the scene tree (see how [To add group containers](#)).
3. Add a new group as a sub-container to the first group, and name it *background*.
4. Add an object, in this case a rectangle, to the *background* container. The container should now have four icons; a show/hide, lock/unlock, transformation editor, and a rectangle icon (see how [To add geometries](#)).
5. Adjust the rectangle's size and position (see how [To edit scale and position](#)).
  - Select the Transformation button to open the Transformation editor.
  - Change the position in the Position section, available in the upper left corner of the Transformation editor.
  - In order to ensure that the background is displayed behind all other elements in a scene, adjust the Z position in the Position section. Remember that the Z position value of backgrounds must always be lower than the Z position value of the other scene element categories.
  - Furthermore, if the background contains animations, verify that it will not at any time overlay the other objects. A low z-position value, like for instance -500, will normally be sufficient, but increase if needed.



- Change the size of the rectangle to cover the whole area of the Renderer window in order to create a full-screen background. This is easiest done by pressing the Screen button available in the Screen Size section in the lower right corner of the Transformation editor. The rectangle will then be re-sized to cover the screen.
6. Add an image to the scene tree (see [To add images](#)).
    - Select an image and add it onto the *background* container using drag and drop.



7. Add the key function to the *background* container (see how [To add key](#)).
  - In order for Viz Engine to produce a key signal, a key function must be added.
8. Add a *Control Image* plug-in to the *background* group and the Control Object plug-in will be added automatically (see [Adding Control Plug-ins](#)).

- Adding the Control Image plug-in will allow the background object to be modified in Viz Trio in payout or design mode.
9. Define the tab-field. Select the Image icon on the container to display the Control Image editor. Enter an ID in the *Field Identifier* field, for example 1.
    - This creates a tab-field, which enables the user to edit any exposed properties. If needed, the image position and scaling can be exposed for editing.
  10. Add a description for the background object. Select the Control Object plug-in icon on the *group* container. Enter a description for the background object in the Description field in the upper part of the editor (e.g. Background Still Image).
    - The description entered in the Description field is the same as displayed when adding a scene element to the scene tree in Viz Trio's Designer. Although optional, it is recommended to use the scene name as a description in order to provide a better overview.
  11. Click the Save button to save the background object. Select the appropriate BACKGROUND directory in the Scene database to see the new scene icon.

Figure 79: Viz Artist scene tree

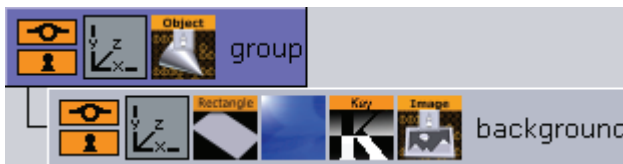
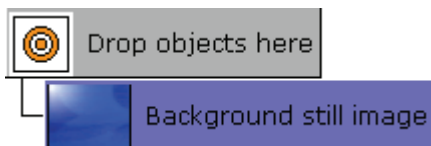


Figure 80: Viz Trio Designer scene tree



For reference: After finishing this quick tutorial, the Viz Artist and Designer scene tree should look like the images above.

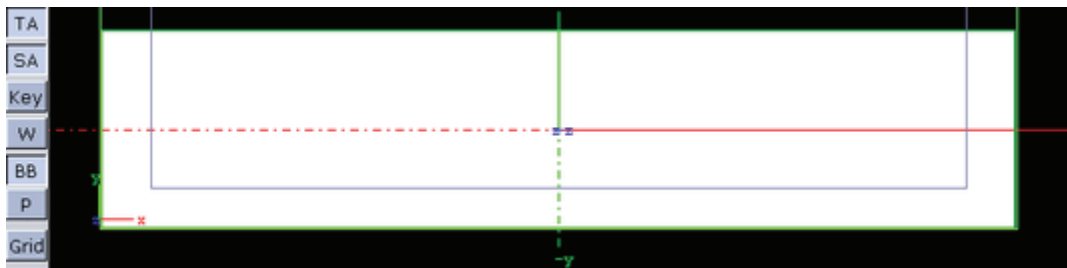
### 12.2.5 Creating Backplates

This section is a quick tutorial on how to create a basic backplate with two text tab fields. For details on how to add animation to a backplate, see also [Creating Animations](#).

-----  
**Note:** Screenshots used are from Viz Artist 3. The user interface in Viz Artist 3 is similar to Viz Artist 2. Where needed additional explanation is given.  
 -----

1. Start by creating a new scene in the `BACKPLATES` directory in the Viz Artist scene tree (see how [To create a new scene element](#)).
2. Double-click on the new scene to open it. Add a group container to the scene tree (see how [To add group containers](#)).
  - Rename the container, by double-clicking the label, to *object*.
3. Add a new group as a sub-container to the first group, and name it *backplate*.
4. Add an object, in this case a rectangle, to the *backplate* container. The container should now have four icons; a show/hide, lock/unlock, transformation editor, and a rectangle icon. For more information, see [To add geometries](#).

5. Adjust the rectangle's size and position. For more information, see [To edit scale and position](#).
  - Scale and position the rectangle to create a backplate. Use the Title and Safe Areas as reference when positioning the rectangle. The Title Area (the blue box) is displayed by clicking on the TA button in the lower left corner of the Renderer window, while the Safe Area (the green box) is shown when selecting the SA button.



- Display the Transformation editor by selecting the Transformation button, located on the *backplate* container.
- Scale the rectangle along the X and Y axes, and place it in the lower third area of the screen as illustrated by the images.
- The backplate should be able to be repositioned and scaled in the Designer and/or in playout mode. Hence, a 100% accuracy with regard to scale and position is not required in this case.

-----  
**Note:** If you have rounded/beveled corners they will be stretched if scaled.  
 -----

6. Set a Z-value slightly below zero (e.g.  $-10$ ) for the backplate.
  - For objects to be displayed in the front, always select zero or a higher value for the z-position. This will ensure that the scene elements are displayed in the correct layers and do not cover each other.
  - Depending on the animations added later, this value could be less.
7. Add a material to the scene tree (see [To add materials](#)).
  - Select a material and add it onto the *backplate* container using drag and drop.
  - To modify the color, display the Material editor by clicking the icon.
8. Add the key function to the *background* container. For more information, see [To add key](#).
  - In order for Viz Engine to produce a key signal, a key function must be added.
9. Add a new group as a sub-container to the *object* container, and name it *placeholder*. This will create a new empty group over the *backplate* container.
10. Add the *Placeholder* plug-in to the *Placeholder* container (see [Adding Control Plug-ins](#)).
  - The *Placeholder* for the object allows users to add scene elements such as text and images to the backplate.



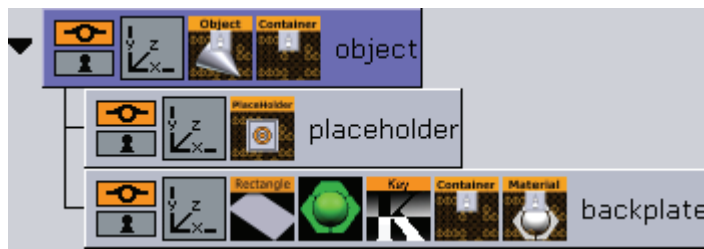
11. *Optional*: Add a font object as a sub-container to the *placeholder* container to provide a visual cue while adjusting the position of the *placeholder* container.

**Note:** The font is a temporary visual cue that must be removed once you have finished adjusting the *placeholder* container.

12. Open the *placeholder* container's transformation editor, and scale and reposition the placeholder object to form a headline as illustrated in the example below.

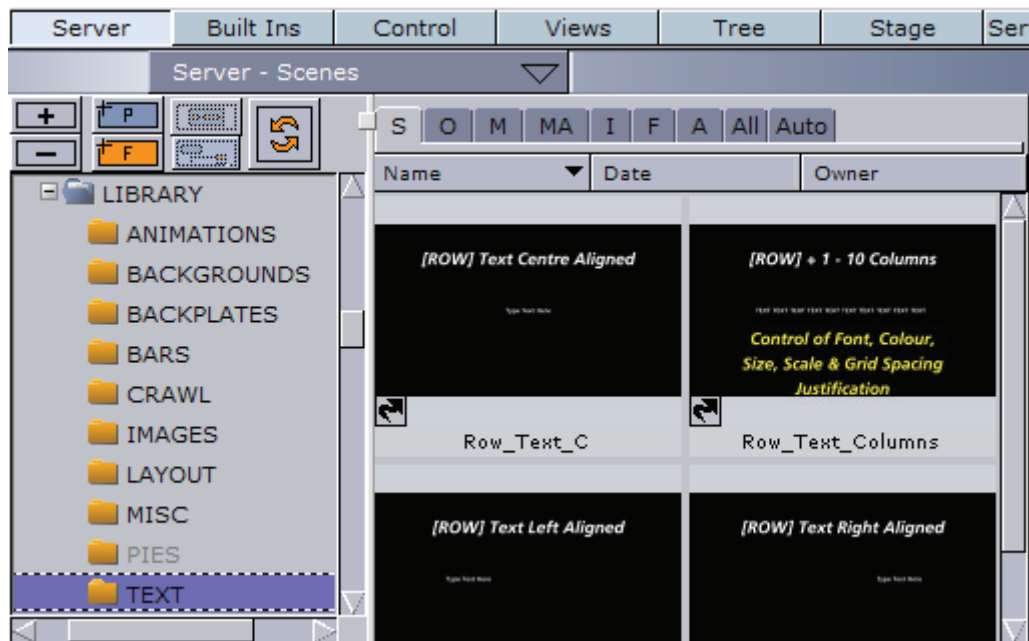


13. Adjust the Z-position in order to ensure that it will be displayed in front of the backplate. Generally a value slightly above zero will be sufficient.
14. When finished adjusting the placeholder object delete the font object as it is no longer needed (see how [To delete containers](#)).
15. Add the *Control Container* plug-in to the node labeled *object* (see [Adding Control Plug-ins](#)).
  - This enables basic positioning control for the entire backplate, including the placeholder object.
  - The *Control Container* plug-in allows all child objects in the group to move simultaneously.
  - Select the *Control Container* icon to open the associated editor. Expose the X, Y, and Z position. The Field Identifier (1) is added automatically.
16. Add the *Control Material* plug-in to the node labeled *backplate* (see [Adding Control Plug-ins](#)).
  - Open the plug-in's editor, and change the *Field Identifier* field to *1.material*.
  - This will group the material property to tab-field 1 and allow the user to modify it together with the other exposed properties of the tab-field.



17. Add the *Control Container* plug-in to the node labeled *backplate* (see [Adding Control Plug-ins](#)).
  - Open the plug-in's editor, and change the *Field Identifier* field to *1.scaling*.
  - This will group the material property to tab-field 1 and allow the user to scale the backplate.
  - Expose the *X* and *Y* scaling.
  - Open the Transformation editor, and enable the Single option in the Scaling section to allow *X* and *Y* axes to be scaled separately.
18. Select the *Control Object* icon in the *object* container to open the Control Object editor. Enter a description in the Description field, for example *Basic backplate*.
19. Click the Save button to save the backplate object.
20. Select the appropriate `BACKPLATES` directory to see the newly created scene and the scene icon the Viz Trio operator will see in Viz Trio's Designer.

## 12.2.6 Creating Text Objects



This section is a quick tutorial on how to create text objects.

**Note:** Screen shots used are from Viz Artist 3. The user interface in Viz Artist 3 is similar to Viz Artist 2. Where needed additional explanation is given.

1. Create a new scene in the `TEXT` directory in the Viz Artist scene tree (see how [To create a new scene element](#)).



2. Rename the group container to *object* (see how [To add group containers](#)).
3. Add a font object to the scene tree as a sub-container of the *object* container, and rename the new container to *text*.
4. Open the Text editor, and enter a common phrase (e.g. *Abc*).
5. Add the key plugin to the *object* element (see how [To add key](#)).
  - In order for Viz Engine to produce a key signal, the key plugin must be added.
6. Add a material to the *text* container (see how [To add materials](#)).
7. Add the *Control Text* and the *Control Material* plug-in to the *text* container.
  - This will allow the Viz Trio operator to edit the text (see [Adding Control Plug-ins](#)).

---

**Note:** *Control Object* is automatically added to the scene tree's root container.

---

8. Select the icon to display the *Control Text* editor.
  - The *Field Identifier* (1) has been added automatically, thereby creating a tab-field.
  - Expose the  *Kerning*, *font*, and *justification* by clicking the corresponding buttons. Set other options if needed.
9. Select the icon to display the *Control Material* editor.
  - The *Field Identifier* (1) has been added automatically, thereby creating a tab-field.
10. Provide a description for the *text* object. This must be done in the *Control Object* editor.
  - Select the *Control Object* icon in the *object* container to open the editor.
  - Enter a description for the text object in the *Description* field, for example *Plain Text*.

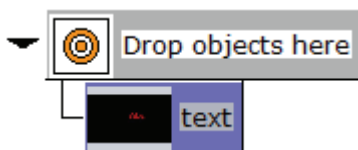
All scene elements available in Viz Trio Designer's resource library are represented with icons reflecting the appearance of the saved scene. All graphical elements are reduced proportionally in size, and will therefore appear very small on the icon.

In order to create an icon with a good enough visualization of the scene, a dummy scaling group can be created at the root level of the scene tree. This will allow the graphical representation of the scene to be enlarged without changing the actual size of the actual scene objects.

11. Add a group to the root level of the scene tree.
  - This will create an empty group container at the same level as the *object* container.

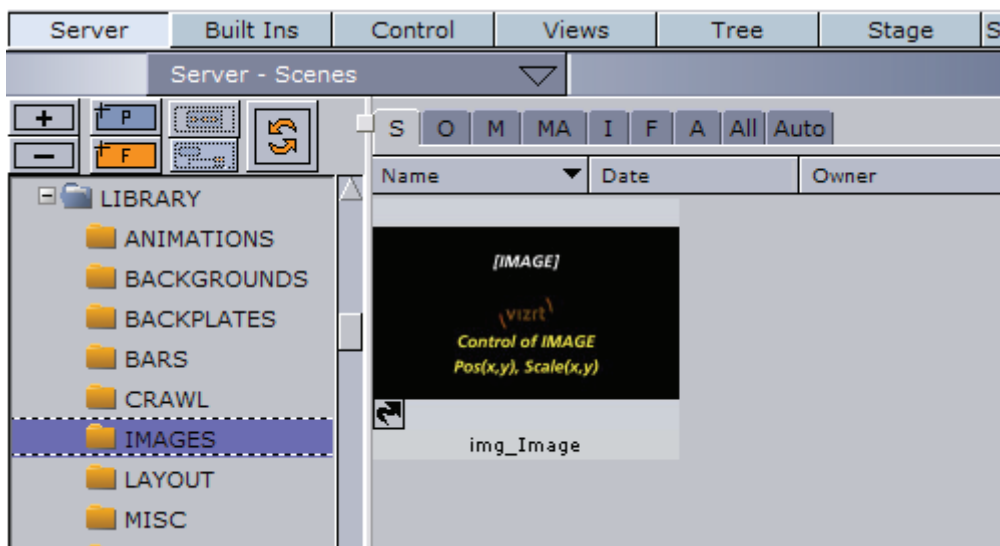


12. Select the *object* container and append it as a sub-container to the newly added *group* container.
  - As the object group with its Control Object is moved, it will not be used in the Viz Trio Designer. This will allow you to create a good visual representation of the library element.
13. Open the transformation editor for the *group* container.
14. Position the text object to the lower left corner of the Renderer window.
15. Increase the scale of the text object until it fills the whole Renderer window.
16. Click the Save button to save the text object. Select the appropriate TEXT directory in the Scene database to see the new scene icon.



For reference: After finishing this quick tutorial, the Viz Trio Designer scene tree should look like the image above.

## 12.2.7 Creating Image Objects



This section is a quick tutorial on how to create image objects.

**Note:** Screen shots used are from Viz Artist 3. The user interface in Viz Artist 3 is similar to Viz Artist 2. Where needed additional explanation is given.

1. Create a new scene in the IMAGE directory (see how [To create a new scene element](#)).



2. Add a group container to the Scene Tree (see how [To add group containers](#)).
3. Rename the group container to *object*.
4. Add an image to the scene tree as a sub-container of the *object* container. The container is automatically named the same name as the image. Rename the container to *image*.
5. Add the key function to the *object* element (see how [To add key](#)).
  - In order for Viz Engine to produce a key signal, a key function must be added.
6. Add the *Control Image* plug-in to the *image* container, which will allow for image changes (see [Adding Control Plug-ins](#)).

---

**Note:** *Control Object* is automatically added to the scene tree's root container.

---

7. Add text or an image that will act as an icon for the container with the *Control Image* editor.
  - The *Field Identifier* (1) has been added automatically, thereby creating a tab-field.

In order to create a scene icon with a large and good enough visualization of the image object, a dummy scaling group can be added. This allows the graphical representation of the icon in the scene tree to be large enough without changing the actual size of the image object to be used in the template.

8. Add a Group to the root level of the scene tree. This will create an empty group container at the same level as the *object* container.
  - Select the *object* container and append it as a sub-container to the newly added *group* container (now top node).
  - Select the Transformation icon in the *group* container to display the Transformation editor.
  - Increase the scale of the image object until it fills the whole Renderer window.
9. Open the Control Image editor, and expose image *position* and *scaling* by clicking the corresponding buttons. Set other options if needed.

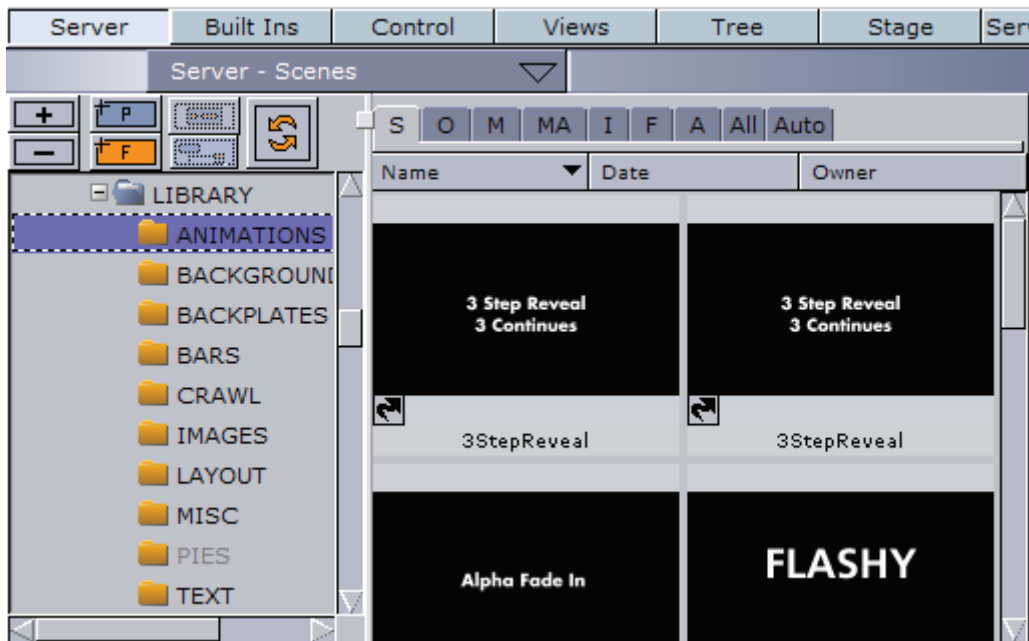
When having specified the parameters, various additional control plug-ins can be added to the image object. Selection of the control plug-ins depends on the effect to be created. Refer the Viz Artist User's Guide for further details on the different control plug-ins.

10. Select the *Control Object* icon in the *object* container to open the editor. Enter a description for the image object in the *Description* field. For example *New Image Object*.
11. Click the Save button to save the image object. Select the appropriate `IMAGE` directory in the Scene database to see the new scene icon.



For reference: After finishing this quick tutorial, the Viz Trio Designer scene tree should look like the image above.

## 12.2.8 Creating Animations



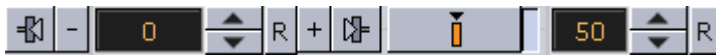
This section is a quick tutorial on how to create a basic animation sliding in from the left. The process of creating an animation object follows the same principles as when adding an animation to graphic elements in ordinary Viz Artist scenes. Basically the only difference is that the animation is added to an empty container without any graphic elements.

**Note:** Screen shots used are from Viz Artist 3. The user interface in Viz Artist 3 is similar to Viz Artist 2. Where needed additional explanation is given.

1. Create a new scene in the `ANIMATIONS` directory (see how [To create a new scene element](#)).



2. Add a group container to the scene tree (see how [To add group containers](#)).
3. Add a group container to the scene tree as a sub-container of the *group* container, and rename the new container to *object*.
4. Create a slide-in animation on the **object** container.
  - To create a slide-in animation the *object* container's position must be changed. Initially the graphics can be off screen by moving it to the far left.
5. Open the **object** container's transformation editor, and specify the container's initial *X-position* in the Position X field (e.g. *-600*).



6. Verify that the **timeline** field in the timeline editor, seen above the Scene Editor, is set to *0* (fields).
7. Click the **Set Key** button (aka Update in Viz Artist 2).
  - This will create a keyframe reflecting the scene as is. The number of fields specified in the timeline field is automatically updated to *50*, which is the default interval.
8. In the Transformation editor, reset the Position X to *0*.

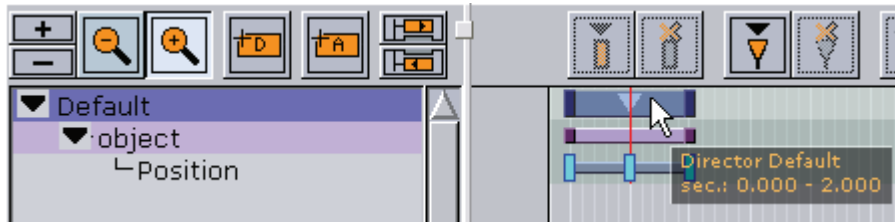
-----  
**Tip:** Click the R (*reset*) button to the right of the Position X field.  
 -----

9. Click the **Set Key** button again to create a new keyframe.
  - This creates a simple in-animation.
  - The values at these keyframes are called keys. Viz Artist calculates the interpolated values between each key to produce the completed animation.
10. Create an out-animation that slides the object out the same way it came in (e.g. Position X set to *-600*).
11. Click the **Set Key** button again to create a new keyframe.
  - Viz Artist calculates the interpolated values between the second and third keyframes to produce the completed out-animation.

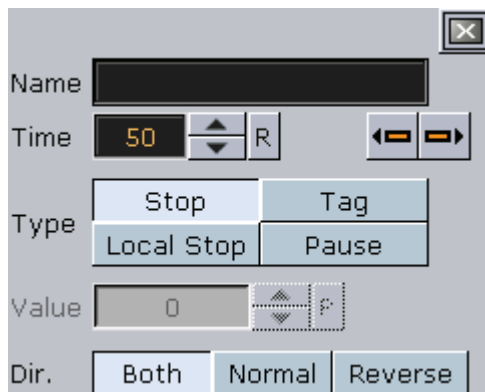
Creating the animation adds an animation icon to the object container. Clicking the icon displays the Stage editor, which allows for further tuning of the animation.

In order to ensure that the animation stays on screen for as long as required, a stop point at the end of the in-animation can be added, which in this case would be at field 50 in the timeline. The stop point will prevent the out-animation from playing before a Continue command is issued.

12. Open the **Stage** editor by selecting the animation icon (wheel) on the **object** container.
  - The animation channel is represented by a gray (Viz Artist 3) or green (Viz Artist 2) line in the right part of the Stage editor.
  - The keyframes are indicated by small rectangles.
13. Move the red **Timeline Marker**, represented as a vertical red line, to the second keyframe (the in point).



14. Select the **Default** director as illustrated in the example above.
15. Click the **Add a Stop/Tag** button (aka Add Stop button in Viz Artist 2.x).
  - It is now possible to see the stop point just added to the Default director.



16. Select the Stop point to open the editor to see, and if needed, adjust the parameters for the stop point.
  - Verify that the stop point has the same time settings as the second keyframe.
17. Add *Control Object* plug-in to the *object* container (see [Adding Control Plug-ins](#)).

---

**Note:** Given that the animation object is created without the need for graphic elements, there is no need to expose properties for editing; however, *Control Object* is needed as it provides a description for the animation object.

---

18. Open the Control Object editor, and enter an appropriate description to the *Description* field.
19. Add a Group container to the scene tree as a sub-container of the *object* container, and rename the new container to *placeholder*.
20. Add the *Placeholder* plug-in to the *Placeholder* container (see [Adding Control Plug-ins](#)).

- The *Placeholder* for the object allows users to add scene elements such as backplates, text, and images to the animation.

**Note:** The animation object contains no graphical elements, hence, saving the scene will create a blank scene icon when viewed in Viz Trio.

21. Create a dummy container with graphics illustrating the animation and apply this as a scene icon.

- This will easily distinguish one animation scene from another in Viz Trio.

**Tip:** In order to provide a graphical illusion of the animation sliding in from the left, a built-in arrow object can be used.



22. Add a geometry, in this case an arrow, to the scene tree (see [To add geometries](#)).

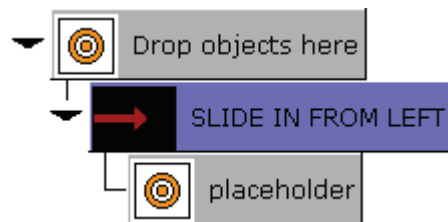
23. Level the *Arrow* container with the *group* container to ensure that the arrow graphics are not displayed on screen during Payout.

24. Open the Arrow editor to set the style and size of the geometry.

25. Open the Arrow container's **transformation editor** to scale and reposition the arrow to illustrate it coming in from the left.

- Add material, images and text if needed.

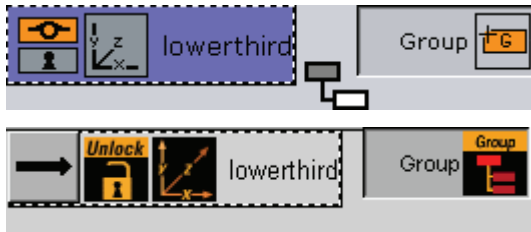
26. Click the Save button to save the animation object. Select the appropriate ANIMATIONS directory in the Scene database to see the new scene icon.



For reference: After finishing this quick tutorial, the Viz Trio Designer scene tree should look alike the image below.

**Note:** SLIDE IN FROM LEFT refers to the control object plugin's description field, and placeholder to the placeholder plugin's description field.



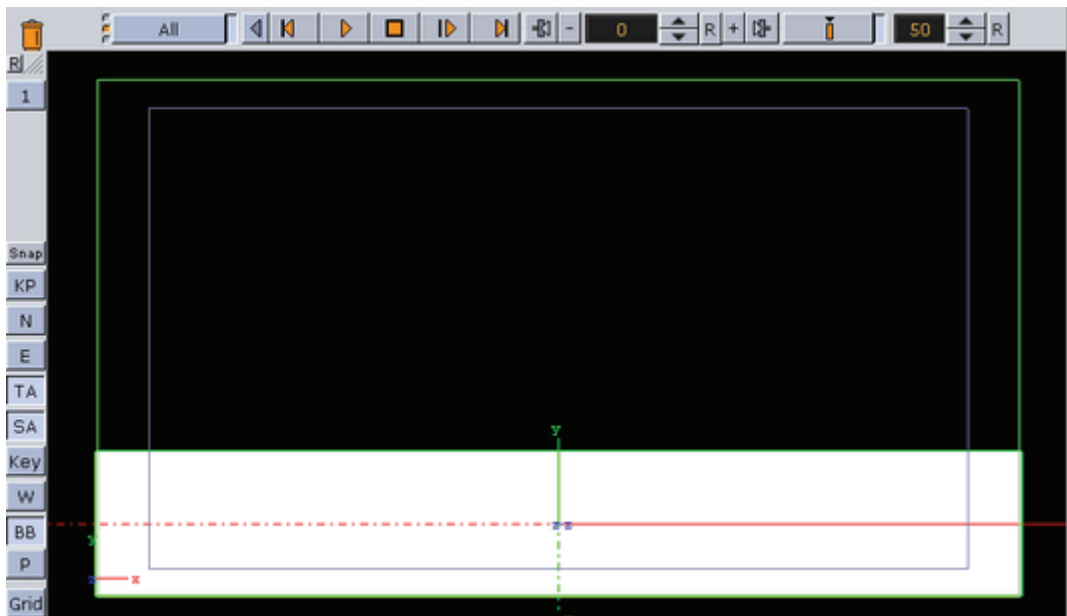


4. Add another group container as a sub-container to the *lowerthird* container.
  - To add a sub-container drag the group object to the right of the existing container.
5. Name the second group *object*.

#### To add a background



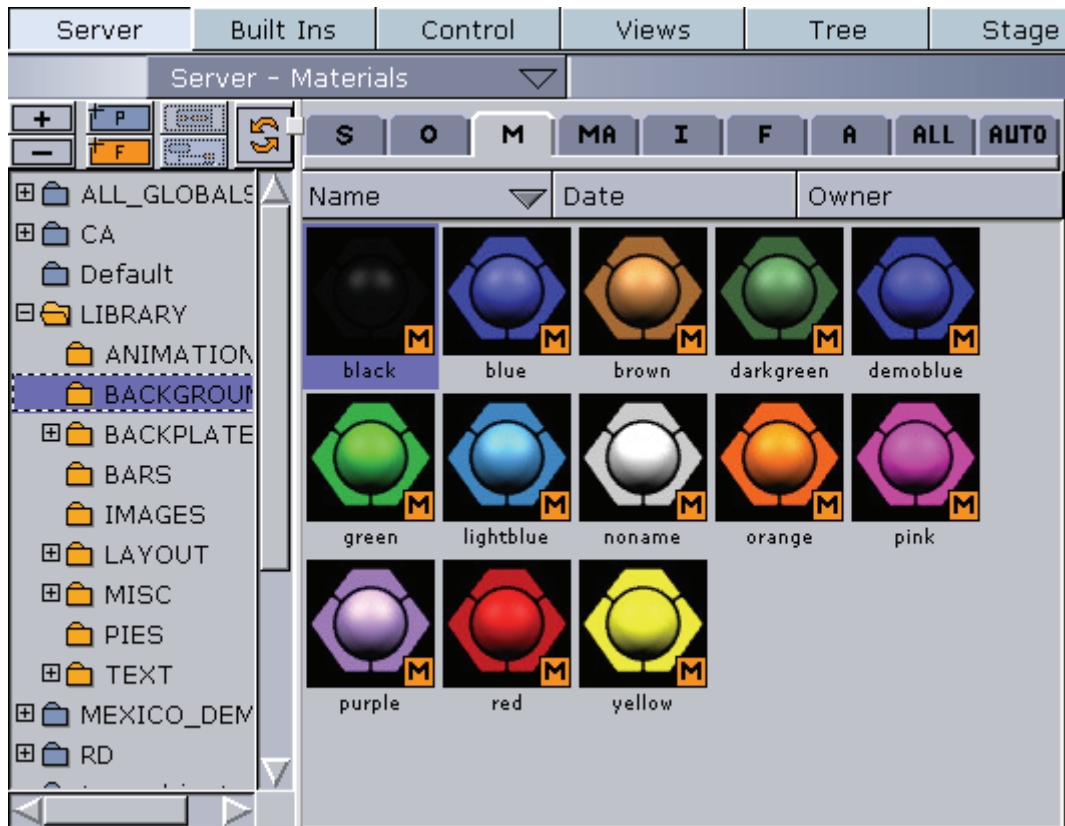
6. Add a rectangle from the *Built Ins* object pool and drop it as a sub-container of the *object* container, and rename it to *Background*.
  - For Viz Artist 3.x: Click the Built Ins button, and select *Primitives* from the drop-list. The object can be found in the *Default* folder.
  - For Viz Artist 2.x: Open the *Object* pool and click on the button labeled *Built in*.



7. Open the rectangle's editor and the background container's transformation editor to scale and then position the object so that it covers the "lower third" of the screen.
  - Rectangle editor: Width *800* and Height *125*.
  - Transformation editor: Position Y *-160*.



8. Since this element is a background element it should be a little bit behind the other objects on the Z-axis.
  - Transformation editor: Position  $Z -100$ .



9. Add a material (see [To add materials](#)) object to give the rectangle color.
  - For Viz Artist 3.x: Click the Server button and select Materials on its drop-down menu (CTRL+3 or the M tab).
  - For Viz Artist 2.x: Click the Material button to open the material database, and select a color.
10. Drag the desired material from the material pool and drop it onto the *Background* container.

### To add text

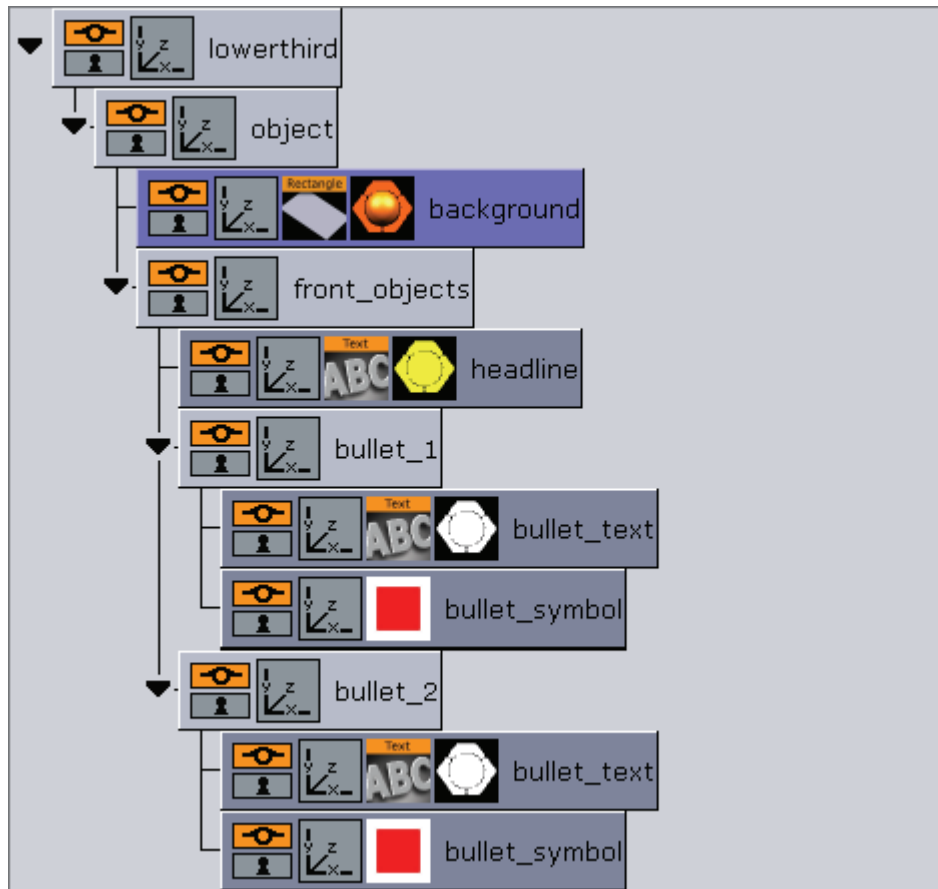


To further improve the basic lower third a headline element and two text lines, lead by a bullet symbol, can be added.

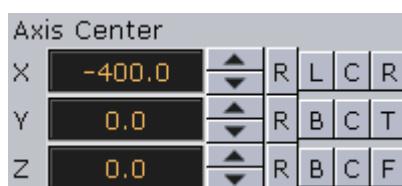
11. Add a group as a sub-container to the object container, and name the group *front\_objects*.
12. Add a font as a sub-container of the *front\_objects* group container, and name it *headline*.
  - Open the text editor by clicking the text symbol of that container.

- In the text editor enter the word *Headline* in the text input field and set the Horizontal orientation to *Left*. This positions the text on the right side of the Y axis to allow the text to be written from left to right.
13. Add a material to the font container.
  14. Open the transformation editor to scale and position the font so that the headline is placed in the upper left corner of the lower third background.
  15. Add a group container as a sub-container of the *front\_objects* group, and name it *bullet\_1*.
  16. As sub-containers of *bullet\_1*, add a font and any image or object that can function as a bullet symbol.
    - Name the text container *bullet\_text* and the image *bullet\_symbol*.
  17. In *bullet\_text*'s text editor enter the text "Bullet 1 text here".
    - Set the horizontal orientation to left as for the headline container.
    - Add a material to the *bullet\_text* container.
    - Scale and position the two containers reciprocally, and then position the *bullet\_1* container under the headline.
  18. Make a copy of the *bullet\_1* container by dragging the container while holding down the CTRL key. Position it as a sub-container of the *front\_objects* container.
    - Rename the copy to *bullet\_2*.
    - Adjust the Y-position of *bullet\_2*.

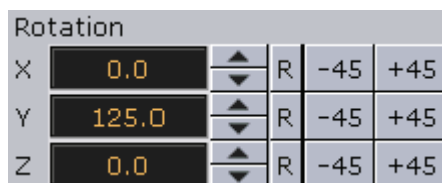
## To create an in animation



To make the whole lower third rotate in from the left side of the screen, the *object* container must be animated. To make the rotation look right, the X-centre of the container must be set to the left.



- Open the transformation editor for the *object* container and click the L button to the right of the Axis Center X property.



- Change the Rotation value for the Y-axis so the container moves out to the left side and is hidden.



21. Press the Set Key (Viz Artist 3.x) or Update (Viz Artist 2.x) button.



22. Set the Rotation value for the Y-axis back to zero by pressing the R (reset) button. Perform the previous step to set a new key. An animation object should now be visible on the *object* container. Play the animation in the render window.

### To create an out animation



To create an out animation, a simple fade out animation of the whole object can be created.

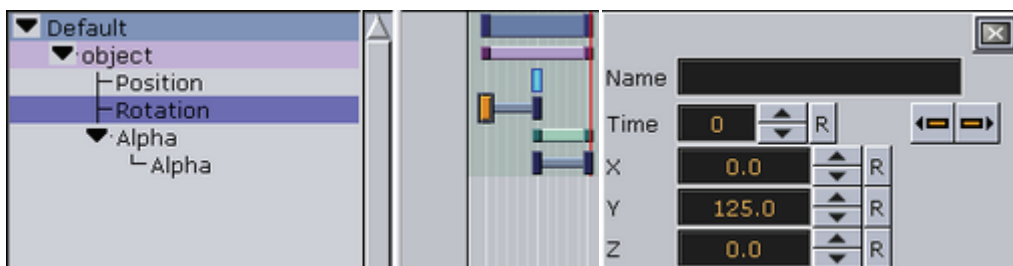
23. Add an *Alpha* function to the *object* container.

- For Viz Artist 3.x: Click the Built Ins button, and select *Function Container* from the drop-list. The *Alpha* function can be found in the *Global* folder.
- For Viz Artist 2.x: Click the Function button.

24. Open the alpha editor.

- Set the alpha value to 100% and update the animation by setting a new key.
- Set the alpha value to 0.0% and update the set key again.
- This has created an alpha animation in addition to the rotation animation.0

25. Click the Stage button and look at the animation channels:



- It is possible to move the keyframes to achieve a different timing. Move keyframes either by dragging them with the mouse or by clicking on them and alter the time settings in the keyframe editor

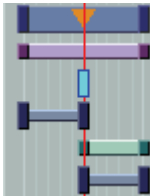
### To add stop tags

When playing out the scene, the objects should move in from the left and then fade out again. In order to stop the animation after the rotation, a stop point must be added before the animation can continue and fade out the scene.

26. Move the timeline (the thin red vertical line) to the second keyframe of the rotation channel (an exact spot is not needed as it is easy to adjust afterwards).



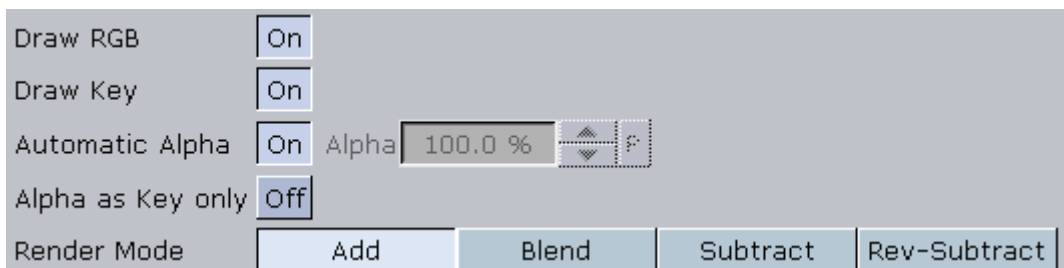
27. Add a Stop keyframe to the Default director.
  - For Viz Artist 3.x: Click the Add a Stop/Tag button above the stage editor.
  - For Viz Artist 2.x: Click the *Default* director label to open the Default director editor. Click the Add Stop button.



28. Move the stop-point with the mouse or move it by using its editor that opens when it is selected. Check that it has the same time settings as the second keyframe on the rotation channel.
29. Play out the animation to verify that it stops when the rotation has finished. Click the Continue button to proceed with the alpha fade.

### To add key functions to the container

30. Add a key function to the *background* container, and click the key icon to open the key editor.
  - For Viz Artist 3.x: Click the Built Ins button, and select *Function Container* from the drop-list. The *Alpha* function can be found in the *Global* folder.
  - For Viz Artist 2.x: Click the Function button.



31. The key for the *background* must have the *Alpha as key only* setting enabled (On). This is to avoid a “dirty key” if the background has some level of transparency. Set the *Render mode* to *Blend*.
32. Add a key function to the *front\_objects* container. On this key signal the *Alpha as key only* setting must be disabled (Off), and the *Render mode* must be set

to *Add*. This is to avoid the foreground objects to cut a hole in the key signal of the background object.

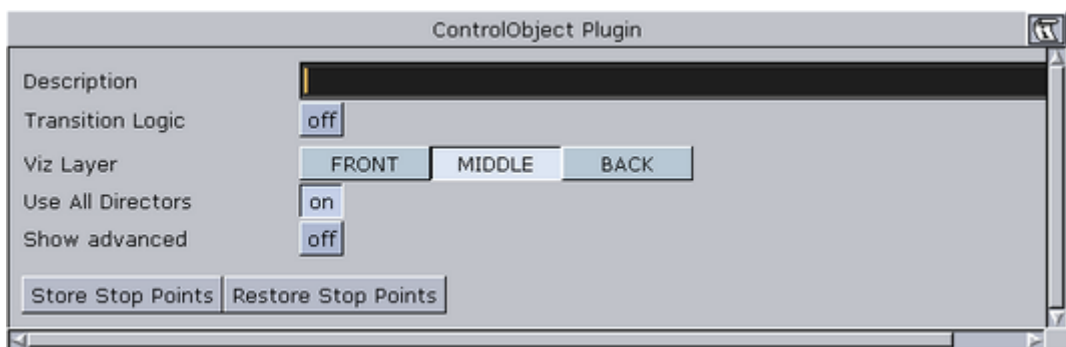
### To add expose properties



To make the scene ready for import into Viz Trio, Control plug-ins must be added. The plug-ins enables the designer to expose scene properties to the Viz Trio operator.

33. Add the Control Object plug-in to the *object* container.

- For Viz Artist 3.x: Click the Built Ins button, and select *Function Container* from the drop-list. The Control Object plug-in can be found in the *Control* folder.
- For Viz Artist 2.x: Click the Function button, and in the *Control* folder select the Control Object plug-in.

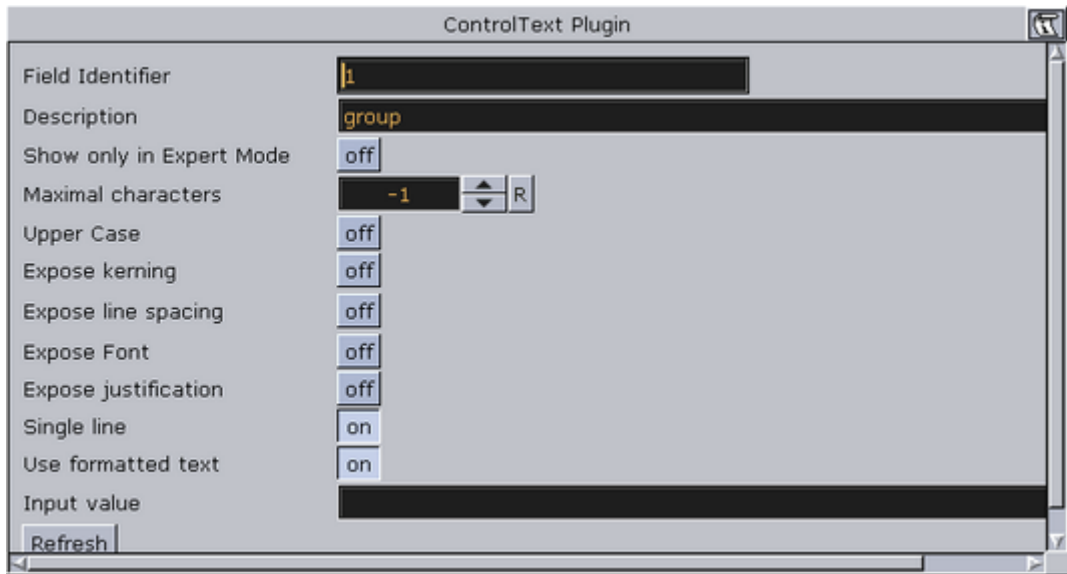


34. Click on the Control Object icon on the *object* container to open its editor.

- Enter a description for the Viz Trio template (e.g. *Lower Third*).



35. Add the Control Text plug-in to the *Headline* and both *bullet\_text* containers.



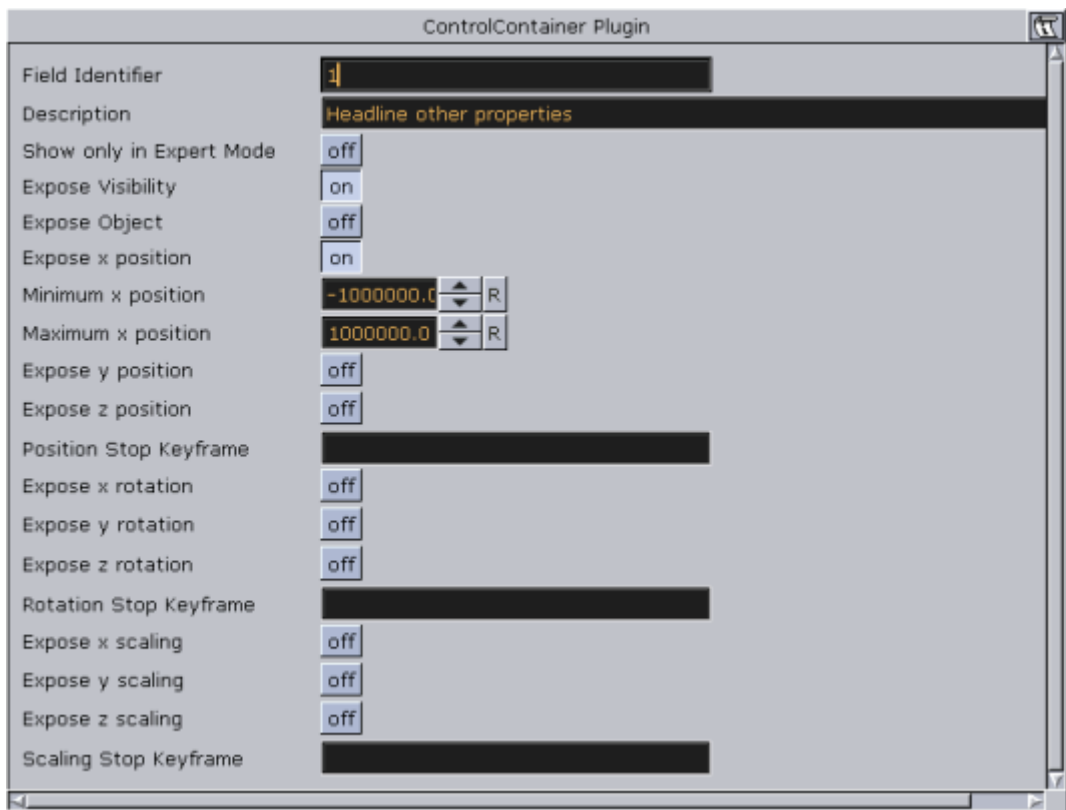
36. Click on the Control Text icon on the containers to open its editor.
- Set the *Field identifier* and *Description*. The Field identifier must be a numeric value and the value will be used to give the Viz Trio page a tab order.
  - In this scene the Headline container will typically get identifier 1, and the two bullet\_text containers will get identifier 2 and 3. There are other parameters that can be set.



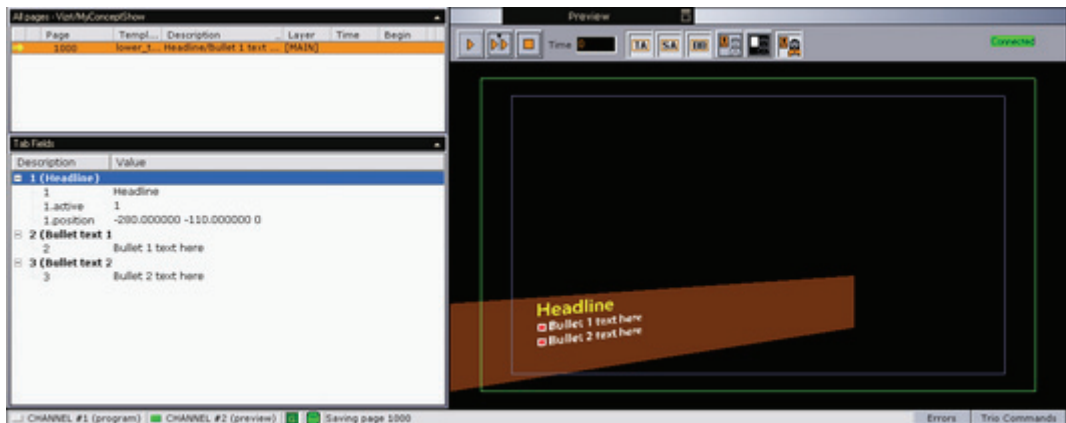
It is possible to expose more properties to the Viz Trio operator. By exposing the position, rotation, and scaling properties the operator can hide or show an object and add material.

37. Add the Control Container plug-in to the container to be exposed (e.g. *headline*).
38. Click the Container plug-in icon to open its editor.





39. Enter a field identifier. If there is another control plug-in on that container, like a Control Text or Image, use the same Field identifier.
- X/Y/Z properties for a keyframe in an animation must be specified by a stop keyframe name.



### To edit multiple elements with a single value

- Use the same field identifier for multiple control plug-ins to treat two or more fields as a single editing element that will receive the same value.

This can for example be used with a bar that scales and a text label that shows its value. The scale value and the text can then be set in a single operation, ensuring that there is no mismatch between the bar size and the written text value.



---

## 14 Creating Transition Effects

The following section will describe how to design custom transition effects for Viz Trio. The scene transition mechanism uses a hardware feature called Dynamic Scene (Dynamic Texture in Viz 2.x).

In Viz you have [To create dynamic textures](#) using the **Dyncamic Scene** plugin found under the Texture Sources tab. You may also find these textures as part of an already existing transition effect scenes in the dynamic folder (at root level).

The textures must be named **layer1** and **layer2**. These two are not ordinary images as they have the possibility to load into them an actual scene.

- **layer1** – Represents the first scene.
- **layer2** – Represents the scene to transit to.

It is possible to make any kind of animation, but there are a few design conventions to follow.

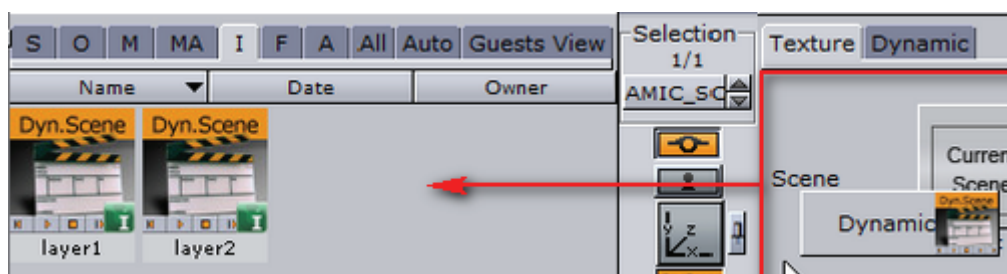
The scene should start with a full screen view of the dynamic texture **layer1**, and end with a full screen display of the dynamic texture **layer2**. Use the screen size section in the transformation editor to set the size of the texture container to fit the screen exactly.

A transition scene is a scene that is created to move from one scene to the next in a dynamic way (not by cut).

-----  
**Note:** All scenes must be placed in a **transitions** folder in the scene database to make it visible in the control application.  
-----

### To create dynamic textures

1. Drag a Dynamic Scene (image build in) into any empty scene.
2. Open theDynamic Scene's **Dynamic** tab and configure the properties (e.g. setting width and height).
3. Open the tranformation properties and set the **Screen Size** to **Screen**.



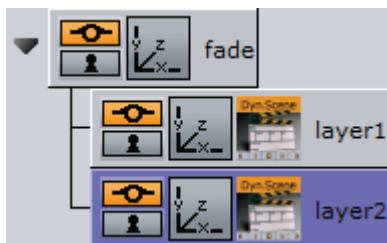
4. Drag the Scene icon into the pool under the correct folder and save two textures as **layer1** and **layer2**.

-----  
**Note:** Depending on the Viz version you use, the icons may be different.  
-----

### To create a transition scene

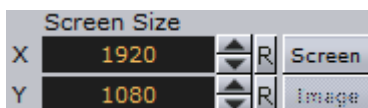
1. Add or create a new scene in the transition scene folder.
2. Add a group to the scene tree, and rename the container to **Fade**.

Figure 81: Viz 3.2.x



3. Add the dynamic texture **Layer 1** and **Layer 2** as sub-containers to the **Fade** group container.

**Note:** There might not be images in the preview window, but only the bounding box.



4. Open the **Fade** group container's transformation editor, and resize the container to fit the entire screen by clicking the Screen button.

The scaling of the main group (**Fade**) to the screen size will ensure that the dynamic textures (layer 1 and layer2) brought under it will be the correct full frame size.



5. For Viz Artist 3.x: Click the Built Ins button, select Function Container from the drop-list and then the Global folder.
6. For Viz Artist 2.x: Click the Function button.
7. Add the Alpha plug-in to the **Layer 1** and **Layer 2** containers.



8. Create a fade out animation for the **Layer 1** container.
  - Click the Alpha plug-in on the container, and set the Alpha to 100%.

- Click the Set Key or Update button above the render window.
  - Set the Alpha value to 0% and click the Set Key or Update button again.
9. Create a fade in animation for the **Layer 2** container.
    - Repeat the steps for **Layer 1** in reverse order.
  10. Save the scene to the transitions folder. Place the timeline somewhere in the middle before the scene is saved as it usually gives the best representation of the transition effect.

---

**Note:** Transition scenes are commonly stored in a **transitions** scene folder at the root of the database directory structure.

---

### See Also

- Viz Artist User's Guide
- Viz Trio User's Guide

---

## 15 Command Line Parameters

Use the command line parameters below to customize the Viz Trio startup.

### To add a command line parameter

- Right-click the program shortcut, and edit the program target path.

-----  
**Example:** "C:\Program Files\vizrt\Trio Client\trio.exe" -mse localhost -control -viz3  
-force\_gpu\_count 2  
-----

**Table 24:** Command line parameters

Parameter	Value Syntax	Description
-control		Allows you to configure Viz Trio.
-folder	showpath	Causes the client to start in the specified show.
-force_gpu_count	integer	Allows the user to force/override the available (detected by Viz) number of GPUs used for preview and program channel.
-hideversion		Hide the version number in the window caption.
-local_program_channel_port	integer	Overrides the configured port number set for the local program channel. This is useful if another port, than the default 6800, needs to be set.
-logfile-path	directory	Folder in which log files are stored.
-loglevel	integer	Controls how much is logged to the log file.
-mse	hostname	Media Sequencer hostname.
-nle-mode		Start Viz Trio in "NLE mode".
-no-nle-mode		Start an NLE-compiled Viz Trio in normal (non-NLE) mode.
-scriptloglevel	integer	Controls how much is logged when executing scripts.
-socket		Use socket (TreeTalk) for Media Sequencer communication.
-t	quoted string	Put a string in the header of the Viz Trio window. %h will be replaced with the Media Sequencer-host

**Table 24:** Command line parameters

Parameter	Value Syntax	Description
-usersetting	settingsname	Use something other than the hostname as an identifier for user settings.
-viz3		Start Viz Engine 3 instead of Viz Engine 2 for local preview.
-viz-console-delay	<time in seconds>	Starts Viz Trio independetly of the Viz Engine process, enabling use of Viz Trio before Viz Engine is running. Note that the Viz Engine console is displayed in this mode (during startup) which can be used to debug Viz Engine start-up problems.
-vizdb	host:db:user:pw	Configure the Viz Engine 3 database login for the local preview engine.
-vizparams	quoted string	Extra parameters to pass on when starting the local viz engine.
-vizpreview	hostname: port<:protocol>	Connect to an external Viz Engine for preview. Viz Engine 2 must have a Viz Multiplexer.

---

## 16 Logging

Log files for Viz Trio are important to Vizrt support in order to troubleshoot any issues our customers might experience; however, log files on the Media Sequencer and Viz Engine is also highly relevant.

This section contains information on the following topics:

- [Viz Trio Log Files](#)
- [Viz Engine Log Files](#)
- [Media Sequencer Log Files](#)
- [Log Level](#)
- [Command Line Parameters](#)

---

### 16.1 Viz Trio Log Files

By setting Viz Trio's log level in the target path to 9, Viz Trio will log a wide range of events. If some error occurs, try and start the application with log level 9 and reproduce the error. The log file generated can then be sent to Vizrt for support together with a description of how the error can be reproduced.

Viz Trio log files are located in the application data folder per user:

- C:\Documents and Settings\\Application Data\Vizrt\Viz Trio

-----  
**Example:** trio-2008-09-08-140923.log  
-----

**CAUTION!** Remember to set a low [Log Level](#), preferably to zero, because a high level of logging might affect the performance when running in a production environment.  
-----

---

### 16.2 Viz Engine Log Files

In addition to Viz Trio's log files created on the client side, it is recommended to read, and if needed send, log files to Vizrt support that were created on the Media Sequencer and Viz Engine.

Viz Engine log files are located in the program folder:

- Viz Engine 2: C:\Program Files\vizrt\viz
- Viz Engine 3: C:\Program Files\vizrt\viz3

-----  
**Example:** VizRender\_1159345015.log  
-----



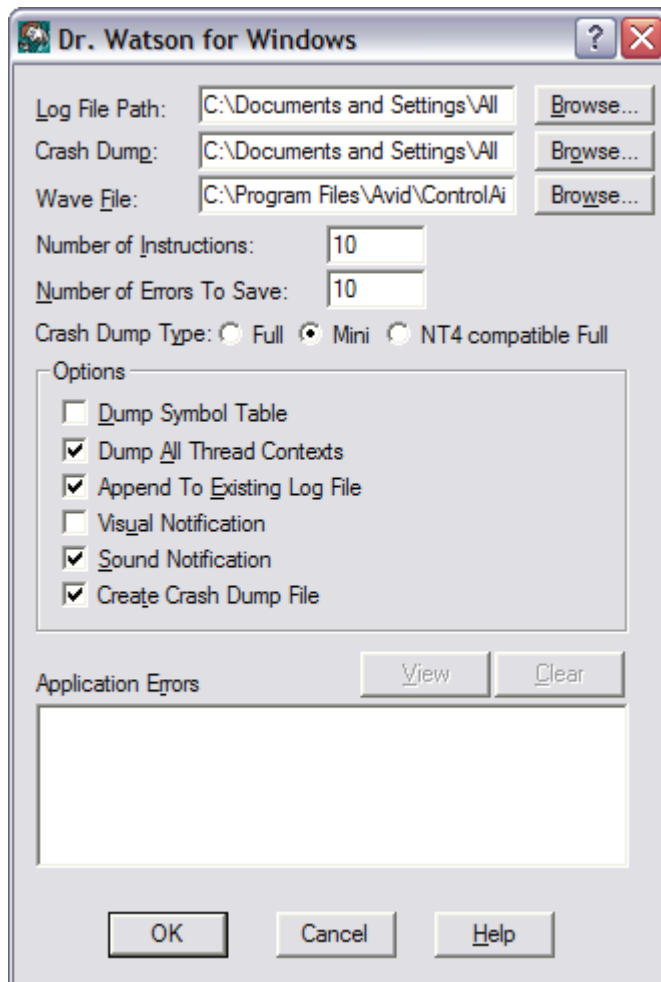
## 16.3 Media Sequencer Log Files

In addition to Viz Trio's log files created on the client side, it is recommended to read, and if needed send, log files to Vizrt support that were created on the Media Sequencer and Viz Engine.

Log files created by a program called Dr. Watson (drwtsn32) can be used to trace problems with the Media Sequencer and Viz Trio. The program is normally a part of the Microsoft Windows default installation. It can be used to create crash dumps whenever a crash occurs as it is able to detect and log most crashes.

The first step after experiencing a Media Sequencer crash is to see whether Dr. Watson has created a crash dump.

**Figure 82:** Media Sequencer crash dump files, Dr Watson



There are several reasons why Dr. Watson might not create a crash dump. The most obvious ones are:

- Dr. Watson was not registered as the default debugger.
  - This can be fixed by starting Dr. Watson with the following command line option: `drwtsn32 -i`.
- The user account the software is running under has not been given debugging privileges.

### To check for crash dumps

1. Start Windows' **Run...** dialog box, and enter **drwtsn32**, and click **OK**.
2. In the appearing window look at the **Applications Errors** list. The name of the program that crashed should be in the list. If it is not, then Dr. Watson has not captured a crash dump.
3. Verify that the **Create Crash Dump File** option is selected.
4. Look at the path specified for **Crash Dump** files.

---

**Example:** C:\Documents and Settings\All Users\Application Data\Microsoft\Dr Watson\user.dmp.

---

5. Send relevant crash dump file to Vizrt for support.

---

## 16.4 Log Level

Viz Trio supports using Intelligent Interface (IIF), and through that interface the log level for Intelligent Interface specific messages on the Media Sequencer can be set. For more information on how to set log levels for the Media Sequencer, see the Media Sequencer documentation.

All logging of output can be assigned a log level. Messages are only written to the log if the current log level is greater than the log level of the message. The log levels are defined as integers in the range 0–100. At a given log level, all the information specified for lower log levels are also logged.

The log level can be changed while the Media Sequencer is running, and the logging output will immediately reflect the new log level.

---

**CAUTION!** Remember to set a low log level, preferably to zero, because a high level of logging might affect the performance when running in a production environment.

---

See the table for a complete list of predefined log levels.

**Table 25:** Log levels

Name	Level	What will be logged
Never	0	Nothing
Bug	5	Bugs detected in the software.
Failure	10	A permanent error, for example part of the software was disabled and no further automatic retries will be attempted.
Lost link	15	A network connection or other link or precious resource was unexpectedly lost. The software might attempt to reestablish the link automatically.
Error	20	Something is incorrect. An internal or external operation could not be completed successfully.
Warning	30	A cause for concern has been detected.
Notice	35	An infrequent but expected event occurred.

**Table 25:** Log levels

<b>Name</b>	<b>Level</b>	<b>What will be logged</b>
Connection	40	A network connection or connection to another precious resource was intentionally opened or closed.
Operation	50	High level operations that are executed as requested or planned.
Input	60	Data that is received into the system from external connections.
Output	70	Data sent to from the system through external connections.
State	80	A change in the significant state of the system.
Analysis	90	Derived information and meta information generated during execution, such as the running times of the executed actions.
Trace	95	The various stages the internal operations of the system goes through during execution.
All	100	All possible logging information, including operation scheduling and loop iteration, and running state statistics.

---

# 17 Appendix

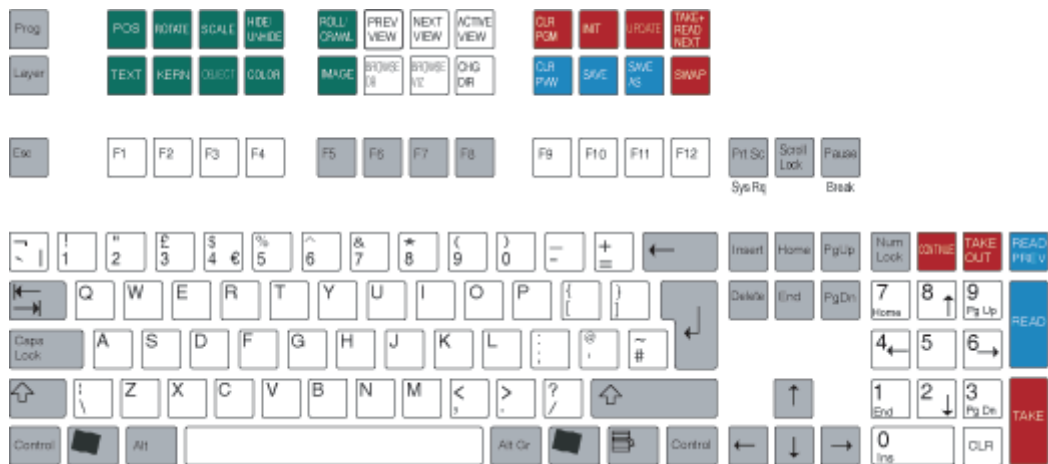
This section contains supplementary information related to Viz Trio:

- [Cherry Keyboard](#)

---

## 17.1 Cherry Keyboard

**Figure 83:** Cherry Keyboard



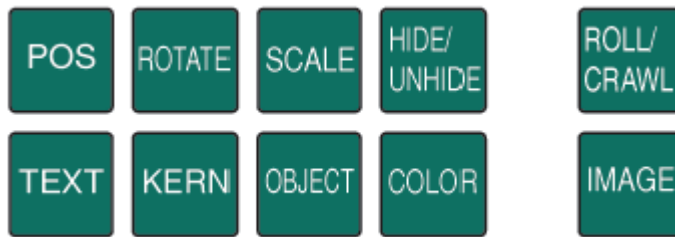
Early versions of Viz Trio was shipped with an now old version of the Cherry Keyboard. If you are still using this keyboard you should be able to use it with current versions of Viz Trio.

The keyboard contains two rows with extra function keys which have been assigned to different Viz Trio actions. The keyboard has its own configuration software. A Viz Trio configuration file must be loaded to create the correct keyboard map. In the Viz Trio client a keyboard mapping file must be imported to assign the correct actions to the keys. This is pre-installed on all Viz Trio clients, so normally there is no need to change these settings.

This section contains information on the following topics:

- [Editing Keys \(green\)](#)
- [Navigation Keys \(white\)](#)
- [Program Channel Keys \(red\)](#)
- [Preview Channel Keys \(blue\)](#)
- [Program and Preview Channel Keys \(red and blue\)](#)
- [Viz Trio Keyboard](#)
- [Keyboard Shortcuts and Macros](#)

### 17.1.1 Editing Keys (green)



The green keys all perform editing operations. The current tab-field must have the property of the key exposed for editing. If not, the key will have no effect and an error message will be written to the log file when the key is pressed.

- **POS** – Displays the position editor.
- **ROTATE** – Displays the rotation editor.
- **TEXT** – Displays the text editor.
- **KERN** – Displays the character kerning editor.
- **SCALE** – Displays the scaling editor
- **OBJECT** – Displays the object pool where 2D and 3D objects can be browsed for.
- **HIDE/UNHIDE** – Hides/shows the tab-field.
- **COLOR** – Shows the pool of colors.
- **ROLL/CRAWL** – Opens the scroller editor
- **IMAGE** – Opens the image pool.

### 17.1.2 Navigation Keys (white)



The white keys shift between different views and editors in the program.

- **PRE VIEW** – If extra page views have been defined, this key shifts the view to the one above the currently active view, see [Add Page List View](#).
- **NEXT VIEW** – If extra page views have been defined, this key shifts the view to the one below the currently active view.
- **ACTIVE VIEW** – When the program has the active focus on some part outside the current page view, hitting this key will bring back the page view in an active state and it is possible to navigate between the pages with the arrow keys.
- **BROWSE DB** – When on an image tab-field, hitting this key will open the [Search Media](#) frame to allow for media searches
- **BROWSE VIZ** – When on an image tab-field, hitting this key will open Viz Engine's image database.
- **CHG DIR** – Displays the change directory or show window.

### 17.1.3 Program Channel Keys (red)



The red function keys all affect actions on the program channel.

- **CLR PGM** – Clears all loaded content on the program channel.
- **INIT** – Initializes the current show on both the program and preview channel.
- **UPDATE** – When a change has been done to a page that is already On-air, hitting update will merge in the changes without running any animations. This is typically used for fixing typing errors. If a page is changed and Take is used instead of Update, all animation directors in the scene will be executed and this normally creates an unwanted effect.
- **TAKE+ READ NEXT** – Takes the page currently read, and reads the next one in the list.
- **SWAP** – The swap key takes to air what is currently read and visible in preview, and it takes off what is currently On-Air and reads that page again.

### 17.1.4 Preview Channel Keys (blue)



The blue keys all affect actions on the preview channel.

- **CLR PVW** – Clears the preview channel
- **SAVE** – Saves the page currently shown on the preview channel.
- **SAVE AS** – Saves the page currently shown in preview to the page number typed in.

### 17.1.5 Program and Preview Channel Keys (red and blue)



The blue keys affect actions on the preview channel, and the red on the program channel.

- **CONTINUE** – When a scene based page halts at a stop point, hitting the Continue key will make the animation continue.
- **TAKE OUT** – If transition logic is used, the Take Out key will take out any page loaded in the layer that is currently read. If transition logic is not used, the Take Out key will perform a “clear” which will be a “hard cut”. To obtain a smooth out animation, the scene must be designed with a stop point and an “out animation”, and the Continue key must be used to take out the page.
- **READ PREV** – Reads the previous page in the page list.
- **READ** – Reads the page currently highlighted by the cursor.
- **TAKE** – Takes the page that is currently read.