



Viz Trio User Guide

Version 3.2



Viz Trio



Copyright © 2021 Vizrt. All rights reserved.

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

Disclaimer

Vizrt provides this publication “as is” without warranty of any kind, either expressed or implied. This publication may contain technical inaccuracies or typographical errors. While every precaution has been taken in the preparation of this document to ensure that it contains accurate and up-to-date information, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained in this document. Vizrt’s policy is one of continual development, so the content of this document is periodically subject to be modified without notice. These changes will be incorporated in new editions of the publication. Vizrt may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time. Vizrt may have patents or pending patent applications covering subject matters in this document. The furnishing of this document does not give you any license to these patents.

Technical Support

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

Created on

2021/06/29

Contents

1	Introduction.....	10
1.1	Related Documents.....	10
1.2	Terminology.....	10
1.3	Feedback.....	11
2	System Overview.....	12
2.1	System Overview - Advanced Setup.....	12
3	Requirements and Installation.....	14
3.1	Requirements.....	14
3.1.1	Prerequisites.....	14
3.1.2	General Requirements.....	14
3.1.3	Hardware Requirements.....	15
3.1.4	Software Requirements.....	15
3.1.5	Viz Engine.....	16
3.1.6	Media Sequencer.....	16
3.1.7	Viz Pilot.....	16
3.1.8	Viz Trio Keyboard.....	17
3.2	Installation.....	17
3.2.1	Installing Viz Trio.....	17
4	Configuration.....	18
4.1	Basic Configuration and Startup.....	18
4.1.1	Connecting to Media Sequencer.....	18
4.1.2	Connecting to Viz Engine.....	19
4.1.3	Starting Viz Trio.....	19
4.2	Hardware Configuration.....	19
4.2.1	Conventional Configuration.....	20
4.2.2	Viz Trio OneBox Configuration.....	20
4.3	Settings.....	20
4.3.1	General Configuration Settings.....	21
4.3.2	Keyboard Shortcuts and Macros.....	22
4.3.3	Text Editor.....	24
4.3.4	Page List/Playlist.....	26
4.3.5	Paths.....	27
4.3.6	Local Preview.....	29
4.3.7	User Restrictions.....	29

4.3.8	Video Codecs	32
4.3.9	Codecs Installation Options	32
4.3.10	Installing Codecs for Local Preview	32
4.3.11	Setting a Preferred Decoder	33
4.3.12	Shared Data	33
4.3.13	Ports	33
4.4	Media Sequencer Configuration	34
4.4.1	Allowing Media Sequencer to Log On with a User Account.....	35
4.5	Viz Pilot Database	35
4.5.1	Database Settings	35
4.5.2	Picture Database Settings.....	36
4.6	Import and Export Settings	36
4.6.1	Import/Export Configuration Settings	37
4.7	Connectivity.....	37
4.7.1	MOS.....	38
4.7.2	Intelligent Interface.....	40
4.7.3	General Purpose Input (GPI)	42
4.7.4	Video Disk Communication Protocol (VDCP).....	45
4.7.5	MCU/AVS	47
4.7.6	Socket Object Settings	48
4.7.7	Proxy.....	50
4.7.8	Viz One Configuration	50
4.8	Graphic Hub	53
4.8.1	Graphic Hub Configuration Settings	53
4.8.2	Configuring the Viz 3.x Database Using Viz Config.....	54
4.8.3	Configuring the Viz 3.x Database Using the Viz Engine Login Window	54
4.8.4	Configuring the Viz 3.x Database Using the Viz Trio Configuration Window.....	55
4.8.5	Configuring the Viz 3.x Database Using the Viz Trio Target Path	55
4.9	Output.....	55
4.9.1	Output Configuration Settings	56
4.9.2	Valid Methods for Resolving the Local Program Channel Name.....	57
4.9.3	Profile Setups.....	58
4.9.4	Working With the Profile Configuration	61
4.9.5	Profile Configuration.....	61
4.9.6	Channel Configuration	62
4.9.7	Output Device Configuration.....	63
4.9.8	Upgrading Old Profiles	68

4.10	Command Line Parameters	68
4.10.1	Adding a Command Line Parameter	68
4.10.2	Command Line Parameters	69
4.11	Settings	71
4.11.1	Video Codecs	71
4.11.2	Codecs Installation Options	71
4.11.3	Installing Codecs for Local Preview	72
4.11.4	Setting a Preferred Decoder	72
4.11.5	Shared Data	72
4.11.6	Ports	73
5	User Interface	74
5.1	Interface Overview	74
5.2	Main Menu	75
5.2.1	File	75
5.2.2	Page	76
5.2.3	Playout	77
5.2.4	View	78
5.2.5	Tools	79
5.2.6	Help	79
5.3	Modes	80
5.3.1	Playout and Design Mode	80
5.3.2	Viz Artist Mode	80
5.3.3	On Air Mode	80
5.3.4	Slave Mode	81
5.3.5	Show Modes	81
5.4	Show Control	83
5.4.1	Show Directories	84
5.4.2	Add Page List View	91
5.4.3	Creating Playlists	92
5.4.4	Show Properties	93
5.4.5	Cleanup Channels	97
5.4.6	Initializing Channels	97
5.4.7	Show Concept	98
5.4.8	Callup Page	98
5.5	Template List	99
5.5.1	Template Context Menu	100
5.5.2	Columns	102

5.5.3	Combination Templates	102
5.5.4	Creating a Combination Template.....	102
5.6	Page List	103
5.6.1	Page Content Filling.....	103
5.6.2	Page List Context Menu	104
5.6.3	Page List Columns	106
5.6.4	Templates and Groups.....	108
5.6.5	Arm and Fire.....	111
5.7	Playlist Modes.....	113
5.7.1	Activating and Deactivating a Playlist.....	113
5.7.2	Taking Pages On-air from a Show Playlist.....	114
5.7.3	Showing and Hiding Playlist Columns.....	114
5.8	Undo and Redo	122
5.9	Tab Fields Window	122
5.9.1	Adding and Editing Custom Values	122
5.10	Status Bar	124
5.10.1	Active Tasks	124
5.10.2	Configuring a Profile.....	125
5.10.3	Changing a Profile	125
5.10.4	Changing Channel Assignment	126
5.10.5	Checking Memory Usage.....	126
5.11	Page Editor	126
5.11.1	Edit a Page.....	127
5.11.2	Text.....	132
5.11.3	Database Linking	137
5.11.4	Image Property Editor	140
5.11.5	Transformation Properties	141
5.11.6	Tables	142
5.11.7	Clock.....	145
5.11.8	Maps	146
5.12	Create New Scroll.....	149
5.12.1	Create New Scroll Editor.....	150
5.12.2	Scroll Elements Editor	151
5.12.3	Scroll Configuration	152
5.12.4	Scroll Live Controls	152
5.12.5	Scroll Control.....	153
5.12.6	Element Spacing	153

5.12.7	Easepoint Editor.....	154
5.12.8	Working with Scrolls	155
5.13	Edit Show Script	157
5.14	Snapshot	157
5.15	Render Videoclip.....	158
5.16	Search Media	159
5.16.1	Media Context Menu.....	160
5.16.2	Search and Filter Options	160
5.16.3	Ordering Metadata Fields for Video and Images	161
5.17	Import Scenes.....	161
5.17.1	Importing Recursively	162
5.17.2	Importing Scenes with Toggle or Scroller Plugin.....	162
5.18	Graphics Preview.....	163
5.18.1	Real-time Rendering of Graphics and Video.....	163
5.18.2	Floating or Moving the Preview Window	163
5.18.3	Connection Status.....	165
5.18.4	OnPreview Script for a Graphical Element	166
5.19	Video Preview	167
5.19.1	Video Control Buttons and Timeline Display.....	167
5.20	TimeCode Monitor	167
5.20.1	TC Monitor	168
5.20.2	TimeCode Monitor Options	168
5.20.3	Enabling the TimeCode Monitor	168
5.20.4	Disabling the TimeCode Monitor.....	169
5.20.5	Monitoring a Video Time Code	169
5.21	Field Linking and Feed Browsing in Viz Trio	169
5.21.1	Workflow	169
5.21.2	Overview.....	170
5.21.3	Technical	171
5.21.4	Field Linking	171
5.21.5	Feed Browsing	176
5.22	Timeline Editor	178
5.22.1	Timeline Editor Functions	179
5.22.2	Working with the Timeline Editor	183
5.22.3	Troubleshooting and Known Limitations	184
5.23	Viz Trio Keyboard	184
5.23.1	Editing Keys.....	185

5.23.2	Navigation Keys	186
5.23.3	Program and Preview Keys	186
6	Macro Commands and Events	188
6.1	Events.....	188
6.1.1	Template Event Callbacks	188
6.1.2	Show Event Callbacks.....	189
6.2	Macro Language	191
6.2.1	Working with Macro Commands.....	191
7	Designing Scenes.....	248
7.1	The Viz Trio Designer	248
7.1.1	The Viz Trio Designer	248
7.1.2	Starting the Designer	248
7.1.3	Resources	250
7.1.4	Scene Tree.....	252
7.1.5	Tab Fields.....	252
7.1.6	Page Editor	253
7.1.7	Properties	254
7.2	Creating Scene Elements in Viz Artist.....	259
7.2.1	Viz Artist User Interface	260
7.2.2	Creating Graphics	261
7.2.3	Adding Control Plug-ins	267
7.2.4	Creating Backgrounds.....	268
7.2.5	Creating Backplates	270
7.2.6	Creating Text Objects	273
7.2.7	Creating Animations	276
8	Creating Standalone Scenes	280
8.1	Creating a Scene	280
8.2	Adding a Background.....	281
8.3	Adding Text.....	282
8.4	Creating an In Animation	283
8.5	Creating an Out Animation	284
8.6	Adding Stop Tags	285
8.7	Adding Key Functions to the Container	286
8.8	Adding Exposed Properties	287
8.9	Editing Multiple Elements with a Single Value.....	290
9	Creating Transition Effects.....	292

9.1	Using Built-in Transition Effects	292
9.1.1	Configuring Built-in Transition Effects	292
9.1.2	Using Built-in Transition Effects	292
9.2	Creating Transition Effect Scenes	293
9.2.1	Creating Dynamic Textures	293
9.2.2	Creating a Transition Scene	294
9.3	TransitionLayers	295
9.3.1	TransitionLayers Properties	295
10	Scripting	297
10.1	Notes About Scripts	297
10.2	Viz Trio Scripting	297
10.2.1	Script Directory	298
10.2.2	Script Editor	298
10.2.3	Script Backup	302
10.2.4	Script Errors	302
10.3	Viz Template Wizard Scripting	302
10.3.1	Dynamically Adding Components	302
10.3.2	Setting and Getting Component Values	303
10.3.3	Setting and Getting Show Values	303
11	Appendix	304
11.1	Enabling Windows Crash Dumps	304
11.2	Logging	304
11.2.1	Viz Trio Log Files	304
11.2.2	Viz Trio Error Messages	305
11.2.3	Viz Engine and Media Sequencer Log Files	306
11.3	Cherry Keyboard	308
11.3.1	Editing Keys (green)	309
11.3.2	Navigation Keys (white)	309
11.3.3	Program Channel Keys (red)	310
11.3.4	Preview Channel Keys (blue)	310
11.3.5	Program and Preview Channel Keys (red and blue)	311

1 Introduction

Welcome to Viz Trio. This guide covers how to configure and operate Viz Trio.


Viz Trio contains all the features of a typical CG system and more.

- Viz Engine is the graphics rendering output system
- Media Sequencer is the control the playout of media elements

You can

- Trigger graphical elements stored as *pages* in a directory structure, with each page utilizing a unique call-up code.
- Enter content in a WYSIWYG (What You See Is What You Get) manner.

Advanced features let you connect to multiple newsroom systems, do seamless context switches on graphics, use specialized editors that can change almost any property of a graphic, produce on-the-fly graphics with a built-in design tool, and more.

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

1.1 Related Documents

- *Viz Artist User Guide*: How to create standalone and transition logic scenes.
- *Viz Pilot User Guide*: How to create a playlist in Viz Pilot that can be monitored in Viz Trio, and use the Newsroom Component to create newsroom data elements for a newsroom system playlist.
- *Viz Engine Administrator Guide*: How to configure your Viz Engines.
- *Viz One Administrator Guide*: How to configure your Viz One system.
- *Viz One - Studio User Guide*: How to work with your Viz One system.

For more information on all Vizrt products, please visit:

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

1.2 Terminology

The following terms are 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.
- **Forked Execution:** With the new profile configuration, Viz Trio now supports forked execution by having more than one graphics render engine per channel. Simply put, you can trigger the same scene on multiple render engines where one can act as your backup.
- **MAM-system:** A Media Asset Management (MAM) system takes care of ingestion, annotation, cataloging, storage, retrieval and distribution of digital media assets. Viz Trio works together with Vizrt's MAM-system Viz One.
- **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. It is only the values of the tab fields that 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 can 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. A transition logic scene 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 render output engine used for playout of graphics and video.

1.3 Feedback

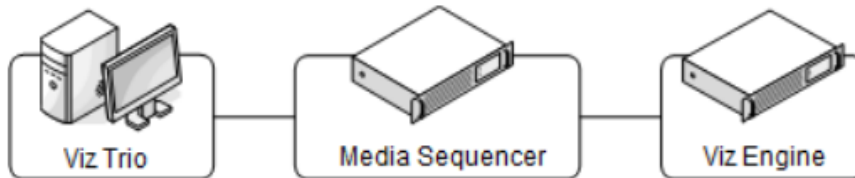
We encourage feedback about our products and documentation. Please contact your local Vizrt customer support team at www.vizrt.com.

2 System Overview

Choose the best setup for your workflow:

- [System Overview - Advanced Setup](#)

System Overview - Basic Setup



Info: The system overview above shows a basic setup. A Viz Engine is typically used for local preview.

- The Trio operator creates *pages*, most often from pre-made *templates*. A page is an instance of a template that can be customized with data values, like sports results or election data.
- When the page is ready, the Trio operator can easily send pages with the necessary information to air.
- Viz Trio automatically uses Media Sequencer to trigger final rendering of the output by the Viz Engine.

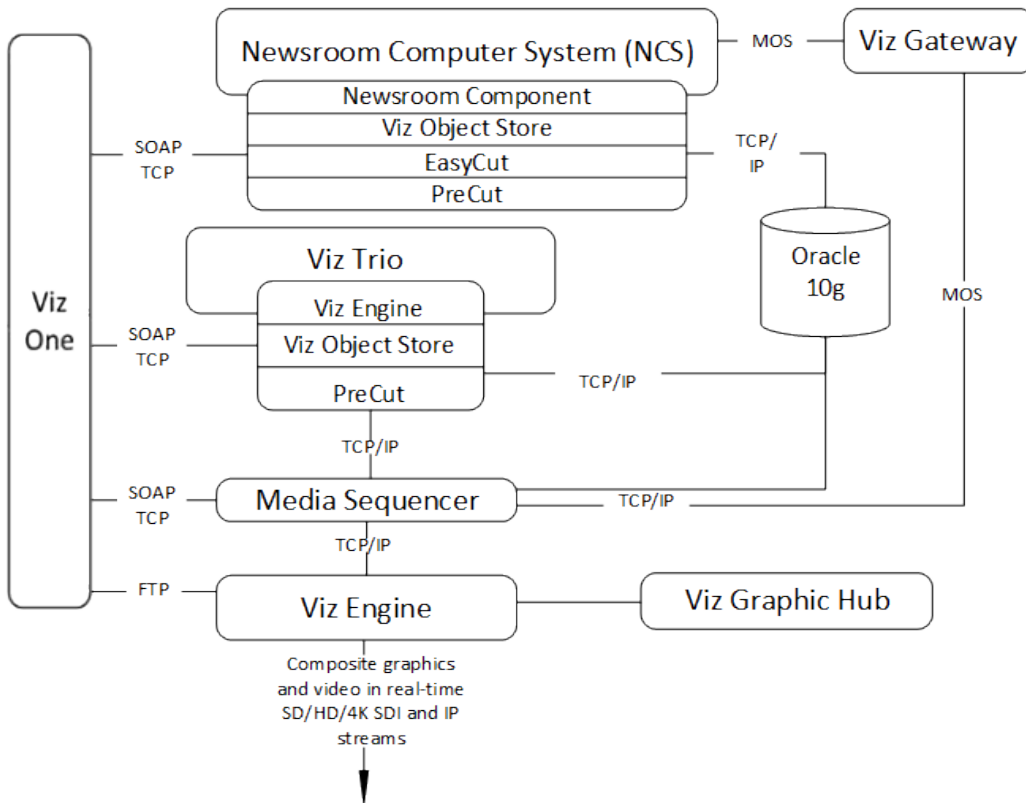
Note: Although Viz Trio, Media Sequencer and Viz Engine can be installed and operated on a single machine, they are mostly installed on separate servers for performance and security reasons.

- Viz Trio can also interface with other systems for news feed and media resources, for instance MAM systems such as Viz One for video and media resources and Viz Pilot.
- Social Media feeds and news can be interfaced and ingested in the workflow with the Viz Social TV solution.

2.1 System Overview – Advanced Setup

- An advanced setup with newsroom and video integration involves several other Vizrt products such as a Vizrt MAM system, Viz Pilot, Viz Gateway and third-party systems such as a newsroom and database system.
- Viz Gateway can be used to connect Viz Trio to most Newsroom systems using the MOS protocol.

Note: Viz Pilot can also be used as the control application. Viz Pilot also has a TCP/IP connection to the Oracle database.



3 Requirements And Installation

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

⚠ IMPORTANT! Always check release notes for information on supported versions of software, hardware requirements and last-minute updates.

This section contains information on the following topics:

- [Requirements](#)
- [Installation](#)

3.1 Requirements

This section covers the following topics:

- [Prerequisites](#)
- [General Requirements](#)
- [Hardware Requirements](#)
- [Software Requirements](#)
- [Viz Engine](#)
- [Media Sequencer](#)
- [Viz Pilot](#)
- [Viz Trio Keyboard](#)

3.1.1 Prerequisites


Before installing a Viz Trio system, make sure that the correct hardware and latest software is available. All Vizrt software required is accessible from Vizrt's FTP server at download.vizrt.com. Contact your local Vizrt representative for credentials.

3.1.2 General Requirements

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

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


- Vizrt recommends that customers who use remote shares for storing data use UNC (Universal Naming Convention) paths directly in the configuration, rather than mapped drives. Example: \\vosstore\images
- 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 a USB or a parallel port dongle on the renderer machine is required.

 **Caution:** Third party systems that provide Vizrt systems with files must only use Microsoft Windows operating system compatible characters in file names.

3.1.3 Hardware Requirements

Hardware requirements vary depending on the system purchased; however, all Vizrt systems are delivered with a hardware specification sheet that matches the Software Requirements for new systems. If you are using pre-existing hardware, such as Viz Engine, with newer versions of Vizrt software, check the new hardware specifications to make sure the software can run on the pre-existing hardware specification.

Additional hardware must always be checked for compatibility with existing hardware. For example, GPI cards supported by Vizrt must fit in the Media Sequencer servers.

 **Note:** For more information on hardware specifications, please contact your local Vizrt customer support team.

3.1.4 Software Requirements

Recommended Versions

- Windows 7
- MS .NET Framework 4.5 or later
- Internet Explorer 11
- Media Sequencer 3.1 or later
- Viz Engine 3.8 or later

Minimum Required Versions

- Windows Vista
- MS .NET Framework 4.5
- Internet Explorer 10
- Media Sequencer 3.0
- Viz Engine 3.3

Optional

- Viz One 5.10 (recommended), Viz One 5.4 (minimum)
- Preview Server 3.0 (recommended), Preview Server 1.0 (minimum)
- Viz Social TV 1.2 (recommended), Viz Social TV 1.0 (minimum)

Not Supported

- Windows XP
- MS .NET Framework 4.0
- Viz Engine 3.2

3.1.5 Viz Engine

The Viz Engine is the output service (renderer) for Viz Trio, and is a separate installer. A Viz Engine is required in order for Viz Trio to preview graphics. Check that Viz Engine is installed, configured and working before installing and using Viz Trio.

A licensed Vizrt dongle must be attached to the USB or printer port on the machine. After installing the Viz Engine, make sure to also install any additional plug-ins, such as Viz Datapool. The plug-ins required will vary depending on the scene design and integrations.

To install the Viz Engine, refer to the instructions in the [Viz Engine Administrator Guide](#).

3.1.6 Media Sequencer

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

Installing 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 Pilot database.

3.1.7 Viz Pilot


In order to connect to Viz Pilot's Oracle database, a runtime installation of the Oracle database client is required. It is recommended to use the same client version as the Oracle database uses. After the client is installed, Viz Trio's database connection can be configured. See [Viz Pilot User Guide](#) for installation instructions.

3.1.8 Viz Trio 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 required keyboard drivers.

Note: To learn more about the keyboard's technical specifications, visit <http://www.devlin.co.uk/keyboards/semicust.html> *DevlinGroup - Devlin Electronics Limited.*

Importing the Keyboard Mapping File

1. Open Viz Trio and click the **Show properties** button.
2. In the **Show properties** window that opens, click the **Keyboard** button. 
3. In the **Macros for Current Show** window that opens, click **Import**.
4. In the **Import keyboard shortcuts** dialog that opens, click the folder button to browse for and import the *keyboard* file (extension *.kbd).
5. *Optional:* Select to merge the existing keyboard shortcuts with the new ones.
6. Click **Import**.

See Also

- [Viz Trio Keyboard](#)
- [Viz Pilot Database](#)

3.2 Installation


3.2.1 Installing Viz Trio

1. Make sure you have the correct hardware and software platform.
2. Ensure that Viz Engine is installed and that you have a working local Graphic Hub and Media Sequencer or have network credentials to the Graphic Hub and Media Sequencer.
3. Download the latest Viz Trio installer and Release Notes with FTP from download.vizrt.com/products/VizTrio/LatestVersion/
4. Read the **Release Notes** carefully. They may contain important last minute information.
5. Double-click the downloaded installer `TrioClient-<VERSION>.msi` and follow the installation instructions.
6. Visit docs.vizrt.com to learn more about the software or press F1 / Help > View Help to browse the local help.

IMPORTANT! You can only have one Trio installed on a server. If you have a previous version of Trio installed, the previous version will be overwritten by the new Trio installer, which will attempt to preserve most data/configuration settings.

4 Configuration

- Connect Viz Trio to the Media Sequencer and Viz Engine, and set up Viz Trio for playout.

 **Note:** Before starting the Viz Trio client, make sure that a Media Sequencer, Viz Engine and Graphic Hub have been installed, and are running and configured.

Select **File > Configuration** to show the configuration window:

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

Select the menu item you want to display or change from the list on the left. Use the panel on the right to review or change setting values:

This section covers the following topics:

- [Basic Configuration and Startup](#)
- [Hardware Configuration](#)
- [Settings](#)
- [Media Sequencer Configuration](#)
- [Viz Pilot Database](#)
- [Import and Export Settings](#)
- [Connectivity](#)
- [Graphic Hub](#)
- [Output](#)
- [Command Line Parameters](#)
- [Settings](#)

4.1 Basic Configuration And Startup

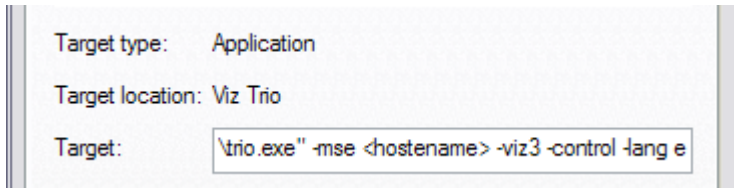
You need to connect Viz Trio to Media Sequencer and a local Viz Engine before you can start using it. This section covers the following topics:

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

4.1.1 Connecting to Media Sequencer

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

Adding Media Sequencer



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

Example: C:\Program Files (x86)\vizrt\Viz Trio\trio.exe -mse localhost -control -lang en -loglevel 5

4.1.2 Connecting to Viz Engine

Viz Trio needs a local Viz Engine in order to run previews and import scenes. Viz Trio automatically detects and runs Viz Engine.

Note: The same Viz Engine version must be installed on all machines and they must share the same data.

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

4.1.3 Starting Viz Trio

Do either of the following:



1. Double-click the **Viz Trio icon** on the desktop, OR
2. Select the program from the Start menu (**All Programs > Vizrt > Viz Trio > Viz Trio**), OR
3. Select the Viz Trio `.exe` file from the program installation directory.

See Also

- [Graphic Hub Connection](#)
- [Media Sequencer Configuration](#)

4.2 Hardware Configuration


Two standard Viz Trio setups are detailed below.

4.2.1 Conventional Configuration


Traditionally, each Viz Trio system has required two standard desktop PCs to operate: one for the Viz Trio client and one for its companion renderer Viz Engine.

4.2.2 Viz Trio OneBox Configuration

You can also run a complete Viz Trio system, including the Viz Engine, on a single standard PC (desktop or rack mountable) with all the features of a conventional Viz Trio setup. Two powerful graphics cards ensure the same graphics quality and rendering speed. Both the VGA preview and final program signals (playout) can be viewed on the same machine.

 **Tip:** The single PC setup is suitable when you have limited space, such as for OB vans, remote broadcasts, and small studios.

4.3 Settings

 **Note:** These UI settings are specific to the local Viz Trio client and do not affect any other Viz Trio clients using the same Media Sequencer.

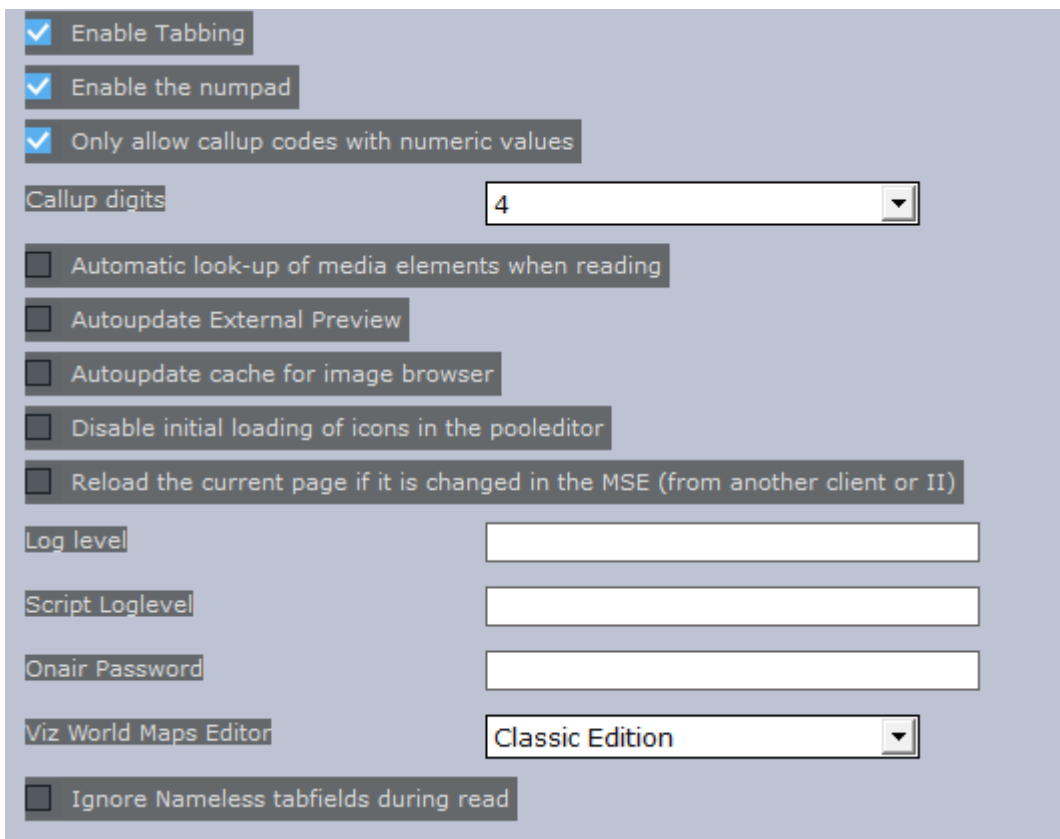
This section covers the following topics:

- [General Configuration Settings](#)
- [Keyboard Shortcuts and Macros](#)
 - [Keyboard Shortcuts and Macros Configuration Settings](#)
 - [Default Keyboard Bindings](#)
- [Text Editor](#)
 - [General](#)
 - [Word Replace](#)
 - [Example](#)
- [Page List/Playlist](#)
 - [General](#)
 - [Cursor](#)
- [Paths](#)
 - [Paths Configuration Settings](#)
- [Local Preview](#)
 - [Local Preview Configuration Settings](#)
- [User Restrictions](#)
 - [User Restrictions Configuration Settings](#)
- [Video Codecs](#)
- [Codecs Installation Options](#)
- [Installing Codecs for Local Preview](#)
- [Setting a Preferred Decoder](#)

- [Shared Data](#)
- [Ports](#)

To view the General configuration settings, click **User Interface > General** in the Trio Configuration.


4.3.1 General Configuration Settings



Enable Tabbing
 Enable the numpad
 Only allow callup codes with numeric values
 Callup digits:
 Automatic look-up of media elements when reading
 Autoupdate External Preview
 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:
 Viz World Maps Editor:
 Ignore Nameless tabfields during read

- **Allow Alphanumeric Imports:** 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, these codes are only allowed when a page is read or saved. When unchecked, this setting will allow all values.
- **Show Tabfield List:** Select this option to make the list visible.
- **Callup digits:** Sets the width of the two callup code edit fields at the top of the main window. For example, selecting 8 will produce a wider code field while selecting 2 will produce a narrow code field. This setting has no impact on page name length.
- **Automatic look-up of media 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.

- **Autoupdate external preview:** When selected, the external preview channel will be updated automatically 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.
- **Autoupdate cache for image browser:** When enabled (default), the Viz Trio client will update the local image browser cache automatically, add any new images and updating images that have been modified. With very large image trees this update process might slow down the system response - disable this option to improve response.
- **Disable initial loading of icons in the pool browser:** 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 enabled, the currently read page is reloaded (read again) if it changes on the server.
- **LogLevel:** Defines what kind of log messages should be logged. Possible levels are: 0, 1, 2, 5, and 9. See Viz Trio Log Levels.
- **Script LogLevel:** Sets the log level to use when running scripts. It defaults to -1 , which is no logging. 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-air or on-air.
- **Viz World Maps Editor:** Defines the Viz World Map Editor (WME) to be launched when a map is added to the template. For more information on how to work with maps, see the *Viz World User Guide*.

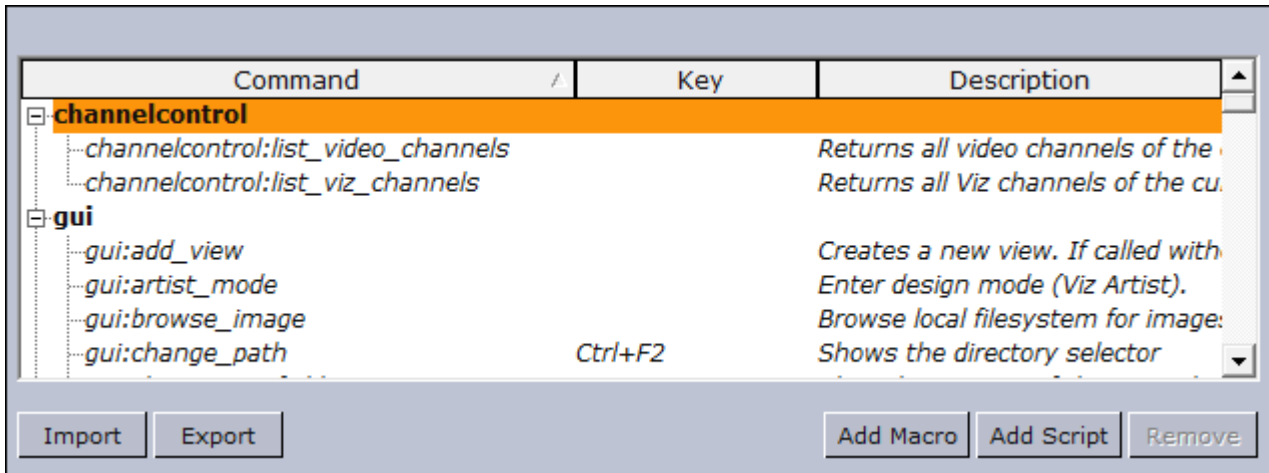
 **Note:** The WME component requires a Viz World Client installation locally and a connection to one or more Viz World Servers or a Server Allocator.

- **Lite Edition (Inline):** an embedded Map Editor with reduced functionality for map selection.
- **Classic Edition:** the default editor. All map features are exposed and the user has full control over the map selected.
- **Second Edition (SE):** the next generation map editor that's more user friendly.
- **Ignore Nameless tabfields during read:** When enabled, this setting discards error messages for tab fields with no name.

4.3.2 Keyboard Shortcuts and Macros

Click **User Interface > Keyboard Shortcuts and Macros** in Trio Configuration:

Keyboard Shortcuts and Macros Configuration Settings



Keyboard shortcuts can be defined for various operations.

- Shortcuts are global - valid for all page folders.
- The keyboard file extension is *.kbd.
- 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.

Note: Local keyboard shortcuts can be specified per show by adding keyboard shortcuts and macros under [Show Properties](#).

- **Import and Export:** can be used to import and export shortcut settings to and from an XML-file for backup and reuse of typical configurations.
- **Add Macro:** allows an operator to write a Viz Trio macro command, and link it to a shortcut key.
- **Add Script:** allows an operator to write a Visual Basic script, and link it to a shortcut key.
- **Remove:** Macros and scripts that are custom made can be removed. Note that no warning appears when this operation is performed.

IMPORTANT! Avoid using **SHIFT** as part of a command as this will mean that you can't use the assigned key when writing characters (for example, in upper case).

Default Keyboard Bindings

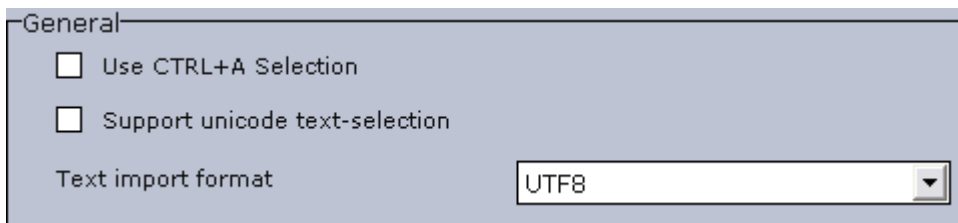
- **CTRL + N** and **CTRL + P** are default bound to *tabfield:next_property* and *tabfield:previous_property*, and are used to navigate between multiple tab-field properties under one tab-field, for example when working with text and kerning.
- **CTRL + TAB** and **CTRL +SHIFT + TAB** are used to move keyboard focus around inside the tab-field editors.

- **TAB** and **SHIFT + TAB** are used to move keyboard focus around if the page editor is *not* active or *no page is read*.

4.3.3 Text Editor

To view configuration settings, click **User Interface > Text Editor** in the Trio Configuration. The text editor contains general settings for text formats and key character replacement.

General

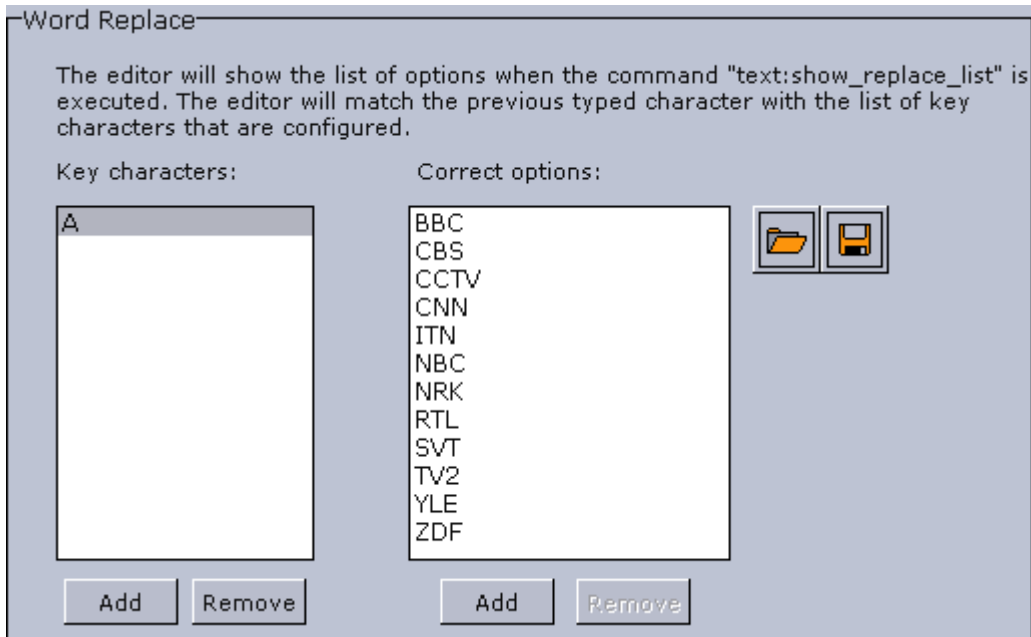


The screenshot shows a configuration window titled "General" with the following settings:

- Use CTRL+A Selection
- Support unicode text-selection
- Text import format: UTF8

- **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 you are copying text from a text editor that by default does not support Unicode language into Viz Trio, the text representation would be incorrect.
- **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



This 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 [User Interface](#) 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, so you can 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 first. Select the key character, and click **Add** below.
- **Remove:** Select an item to remove, and click **Remove** below.
- **Folder:** Imports a predefined list exported to an XML file.
- **Save:** Exports the predefined list to an XML file.

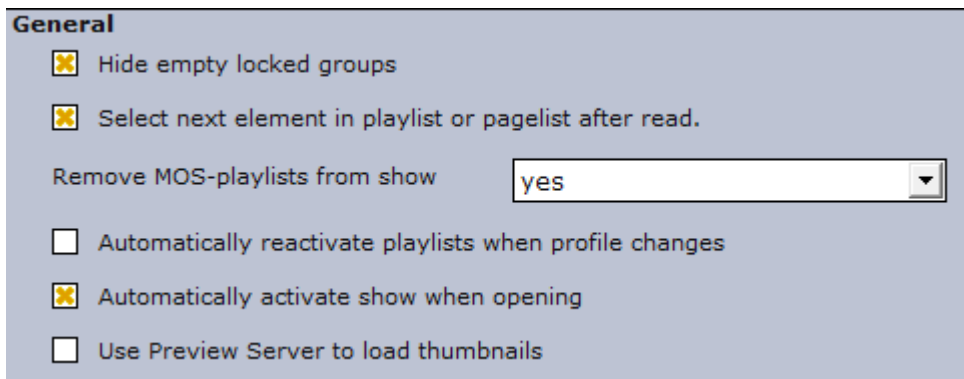
Example

```
<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>
```

4.3.4 Page List/Playlist

To view the Page List/Playlist configuration settings, click **User Interface > Page List/Playlist** in the Trio Configuration.

General



General

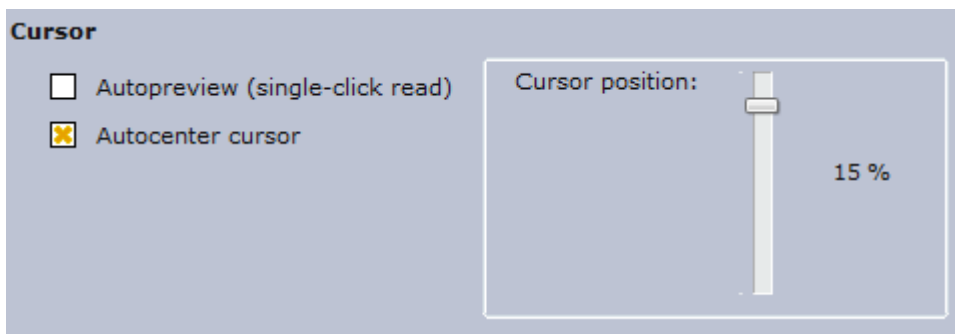
- Hide empty locked groups
- Select next element in playlist or pagelist after read.
- Remove MOS-playlists from show
- Automatically reactivate playlists when profile changes
- Automatically activate show when opening
- Use Preview Server to load thumbnails

- **Hide empty locked groups:** This setting will prevent locked groups with no pages from being displayed in the page list.
- **Select next element in playlist after read:** When enabled, the next element in the playlist will automatically be selected after a read operation.
- **Remove MOS-playlist 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.
- **Automatically reactivate playlists when profile changes:** If enabled, active playlists are reactivated when the current profile is changed.

- **Automatically activate show when opening:** If selected, automatically activate the show when opened. Please be aware that this option, if used, could put a high load on the Media Sequencer.
- **Use Preview Server to load thumbnails:** If selected, use Preview Server to load thumbnails.

⚠ Note: Use a filter with *Description Equals* with *Value* = blank/nothing to automatically hide a group without a name from being displayed in the page list. This replaces the older configuration option “Hide nameless group”.

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.

4.3.5 Paths

To view the Paths configuration settings, open **File > Configuration > User Interface > Paths** in the Trio Configuration. Directories settings are described below All settings are local to the specific Viz Trio client.

Paths Configuration Settings

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"/>	...
Trio Logfile Path	<input type="text"/>	...
MSE As-Run Logfile	<input type="text"/>	
Shared Temp Path	<input type="text"/>	...
Help Path	C:\Program Files (x86)\Vizrt\Viz Tri	...

- **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 default directory to file-based scripts that may be assigned to a Viz Trio show and/or template.
- **Trio Logfile Path:** Sets the path to the directory where the Viz Trio client should store log files. For more information, see [To change the Viz Trio log file path](#).
- **MSE As-Run Logfile:** Sets the path to the Media Sequencer *as-run* log file. This log file offers a Media Sequencer logging mechanism for all elements that have been taken on air. The as-run log file is an additional debugging tool which a Viz Trio administrator may want to enable to verify that the communication between the Media Sequencer and the playout engine is working properly. The file name may be given with either relative or absolute path.

⚠ Note: This is a global setting, affecting all clients connecting to the specified Media Sequencer. Also note that the as-run log file ends up on the Media Sequencer host, not on the local Viz Trio machine. Read more about the as-run log file in the Media Sequencer documentation at http://<mse-host>:8580/mse_manual.html#as-run-log.

- **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.
- **Help Path:** Sets the path for the main help file. Help will be opened when pressing **F1** or when clicking *View Help* under **Help** in the main menu.

4.3.6 Local Preview

To view the Local Preview configuration settings, click **User Interface > Local Preview** in the Trio Configuration.

Local Preview Configuration Settings

General

Viz Engine connection timeout: 30000

Font encoding: UTF-8

Single layer transition logic preview

Use animation when previewing transitions logic elements

Local Preview Resolution

PAL (720x576)

NTSC (720x480)

16:9 (720x405)

Custom (WxH) 730 x 411 Refresh Rate: 60

- **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 animation when previewing transitions logic elements:** If this setting is off transition logic animation will be skipped in the local preview. But this requires that each foreground scene defines a "pilot1" tag on a director. By default, this setting is on. 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).

4.3.7 User Restrictions

To view the User Restrictions configuration settings, click **User Interface > User Restrictions** in the Trio Configuration. Use these settings to administer user permission.

User Restrictions Configuration Settings

Do not allow users to:		
Deny	Description	Corresponding Command
<input type="checkbox"/>	Scripts	
<input type="checkbox"/>	Edit scripts	gui:show_scripteditor
<input type="checkbox"/>	Post Rendering	
<input type="checkbox"/>	Enter new hostnames in post render view	
<input checked="" type="checkbox"/>	Shows	
<input checked="" type="checkbox"/>	Browse viz directories when changing shows	
<input type="checkbox"/>	Browse external playlists	
<input type="checkbox"/>	Create or rename shows	gui:create_new_show
<input type="checkbox"/>	Access show settings	gui:show_settings
<input type="checkbox"/>	Open the window for importing templates into a show	
<input type="checkbox"/>	Delete all pages in a show	show:delete_all_pages
<input checked="" type="checkbox"/>	Page	
<input type="checkbox"/>	Takeout Pages	page:takeout
<input type="checkbox"/>	Direct Takeout Pages	page:direct_takeout
<input type="checkbox"/>	Direct Takeout Pages from a Playlist	playlist:direct_take_out_selected
<input type="checkbox"/>	Delete Templates	show:delete_templates
<input checked="" type="checkbox"/>	Save values to linked databases	
<input type="checkbox"/>	Rename pages	show:rename_page
<input checked="" type="checkbox"/>	Trio	
<input type="checkbox"/>	Cleanup External Renderers	trio:cleanup_renderers
<input checked="" type="checkbox"/>	Call Viz Trio Commands	gui:show_triocommands
<input type="checkbox"/>	Switch to Viz Artist	gui:artist_mode
<input type="checkbox"/>	Designer	
<input type="checkbox"/>	Enter design mode	gui:toggle_design_mode
<input type="checkbox"/>	Access the full tree in design mode	
<input type="checkbox"/>	Customize the designer buttons	
<input type="checkbox"/>	Save scenes outside of the default vizpath	
<input type="checkbox"/>	Scroll Editor	
<input type="checkbox"/>	Create new scroll templates	gui:show_scroll_template_creator

- **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 Properties](#).
- **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 Playlist:** When checked, prevents the operator from issuing a direct take out command on pages in a playlist. Command:
`playlist:direct_take_out_selected.`
 - **Rename pages:** When checked, it is not possible to rename saved pages in the page list.
 - **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 Properties:** When checked, prevents the operator from accessing the Show Properties. 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 [Shows](#) 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 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.`

4.3.8 Video Codecs

Video codecs are only required to preview videos embedded in graphics. Full screen videos can be viewed by using Timeline editor without using a codec.

⚠ IMPORTANT! Due to licensing requirements, Vizrt does not provide the codecs required for local preview. Users must obtain and install their own codecs.

⚠ Note: High resolution playout on Viz Engine does not require these video codecs.

Follow the procedures below to complete installation:

- [Installation Options](#)
- [Installing Codecs for Local Preview](#)
- [Setting a Preferred Decoder](#)

4.3.9 Codecs Installation Options

Codecs are available from several suppliers. Here are a few suggestions:

- FFDSHOW MPEG-4 video decoder and Haali Media Splitter
- LAV Filter video decoder and Splitter
- Main Concept video decoder and splitter

4.3.10 Installing Codecs for Local Preview

The example below shows how to set up support for h.264 playback using the FFDSHOW MPEG-4 codec package and a Matroska Splitter from Haali.

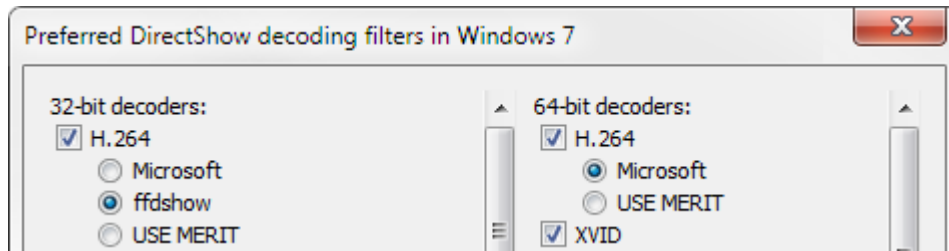
⚠ Note: You need to have your own license for clip playback as FFDSHOW does not come with a decoding license.

1. Make sure you do not have any other codec packages installed on the machine that interfere with FFDSHOW or the media splitter.
2. **Download** the Matroska Splitter from Haali
3. **Download** the Windows 7 DirectShow Filter Tweaker
4. **Download** the FFDSHOW MPEG-4 Video Decoder
 - Make sure you have a license to use the codec.
 - Make sure you download a 32-bit version of the codec.
5. **Uninstall** older 64-bit versions of the MPEG-4 codec.
6. **Install** the Matroska Splitter from Haali.
7. **Install** the Windows 7 DirectShow Filter Tweaker.
8. **Install** the FFDSHOW MPEG-4 codec.
 - After installing the FFDSHOW codec package make sure that no applications are excluded, especially Viz Engine (there is an inclusion and exclusion list in FFDSHOW).
9. Set your MPEG-4 32-bit decoder to FFDSHOW.

- You should now be able to preview video clips from Viz One.

4.3.11 Setting a Preferred Decoder

1. **Run** the Windows 7 DirectShow Filter Tweaker.
2. Click **Preferred decoders** in the dialog.



3. Set your MPEG-4/H.264 32-bit decoder to ffdshow and click **Apply & Close**. Click **Exit**.

4.3.12 Shared Data

Vizrt recommends that customers who use remote shares for storing data use UNC (Universal Naming Convention) paths directly in the configuration, instead of mapped drives.

Example: `\\vosstore\images`

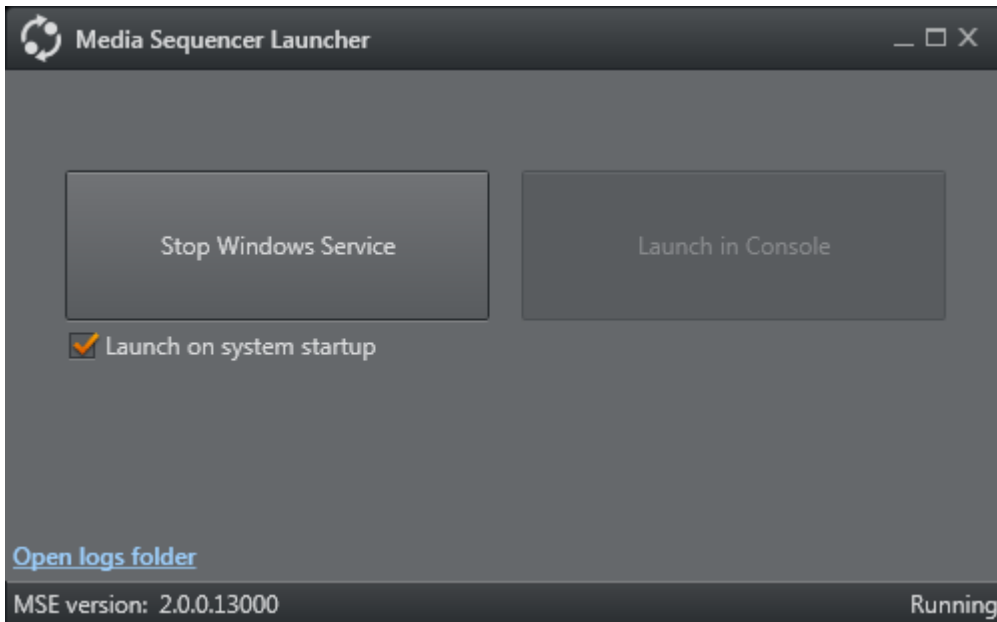
4.3.13 Ports

Component	Protocol	Port	Description
<i>trio.exe</i>	TCP	6200	Used for executing macro commands of the Viz Trio client over a socket connection.
<i>trionle.exe</i>	TCP	6210	Used by the Graphics Plugin for NLE to utilize Viz Trio for effect editing.

See Also

- [Timeline Editor](#)
- [Video Preview](#)

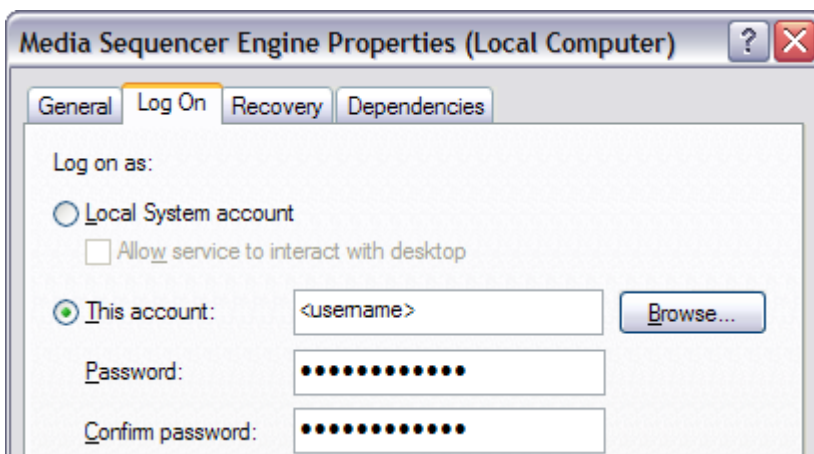
4.4 Media Sequencer Configuration



Media Sequencer is a core component for managing shows and playlists, that can run in either service or console mode (console is mostly used for configuration or debugging/testing). It's generally recommended to run Media Sequencer as an automatically started service process:

- Select this mode by ticking the **Launch on system startup** checkbox on the Media Sequencer startup-screen.

Note: For optimal performance, it's recommended to run Media Sequencer on a dedicated server.



An Oracle database client is needed when using Viz Pilot's Oracle database. If Media Sequencer is

running as a service, it's recommended for the service to log on with a user account, and not with the default Local System account (SYSTEM). This is because Oracle's OCI library is installed per user, and the System user is therefore not able to read environment variables set for a user.

4.4.1 Allowing Media Sequencer to Log On with a User Account

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

4.5 Viz Pilot Database

This section is used for Viz 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 covers the following topics:

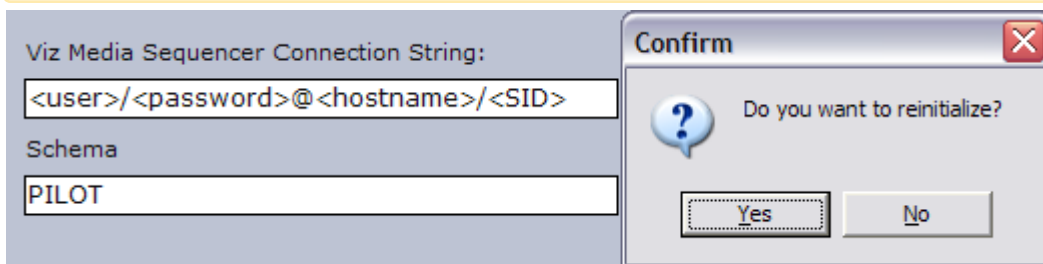
- [Database Settings](#)
- [Picture Database Settings](#)

4.5.1 Database Settings

To view the Database configuration settings, click **VCP Database > Database Settings** in the Trio Configuration.

The Viz Pilot Database pane is used to configure a Viz Pilot database connection.

Note: Before configuring a Viz Pilot (VCP) database connection, an Oracle 10g Runtime client must be installed.



- **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.

Configuring the Viz Pilot Database Connection

1. Click the **Config** button (upper left) to open the Configuration Window.
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.

4.5.2 Picture Database Settings

To view the Picture Database configuration settings, click **VCP Database > Picture Database** in the Trio Configuration.

The picture database is a connection to Viz 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.



Use Picture Database

Picture Database Connection String

- **Use Picture Database:** Enables or disables the VOS picture database.
- **Picture Database Connection String:** Sets the database connection string or TNS name alias.

Configuring the Picture Database

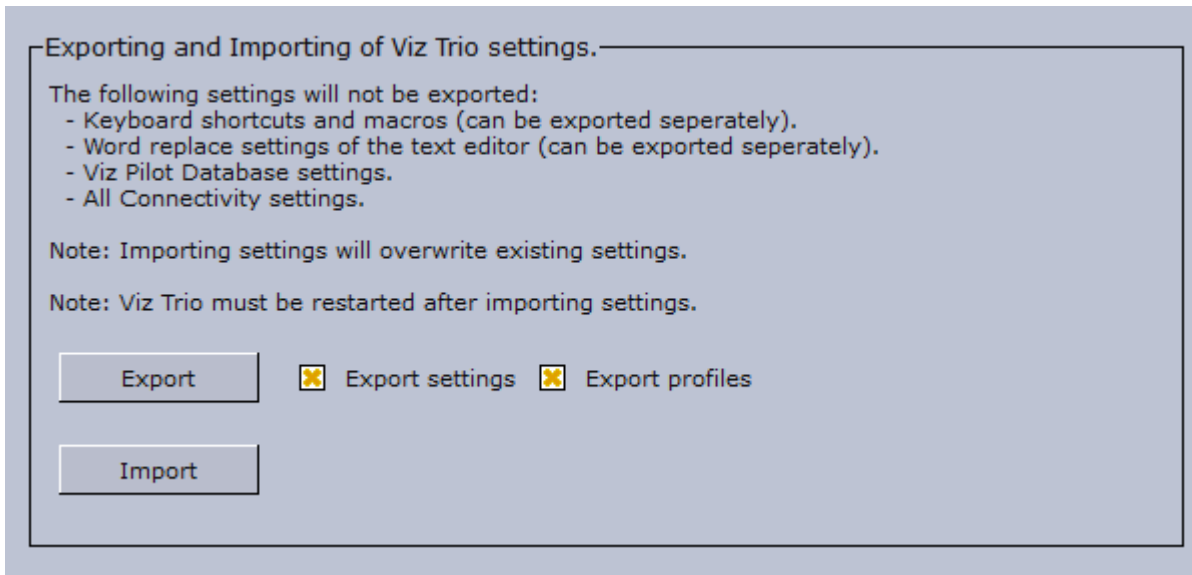
1. Map the shared image drive used by Viz Object Store (VOS) on the Viz Trio and Viz Engine machines.
2. Enable the **Use Picture Database** option, and add the VCP database connection string.
3. Click **Apply**.

Note: If a TNS name alias is configured it can be used as a replacement for the hostname and SID in the connection string.

4.6 Import And Export Settings

To view the Import/Export configuration settings, click **Import/Export Settings** in the Trio Configuration.

4.6.1 Import/Export Configuration Settings



This 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 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.

4.7 Connectivity

The connectivity section is used to configure various interfaces: the Media Object Server (MOS) protocol, Intelligent Interface (IIF), General Purpose Input (GPI), Video Disk Communication Protocol (VDCP), Newstar MCU/AVS, and socket object settings that allow Viz Trio to act as client or a server in a socket connection.

This section covers the following topics:

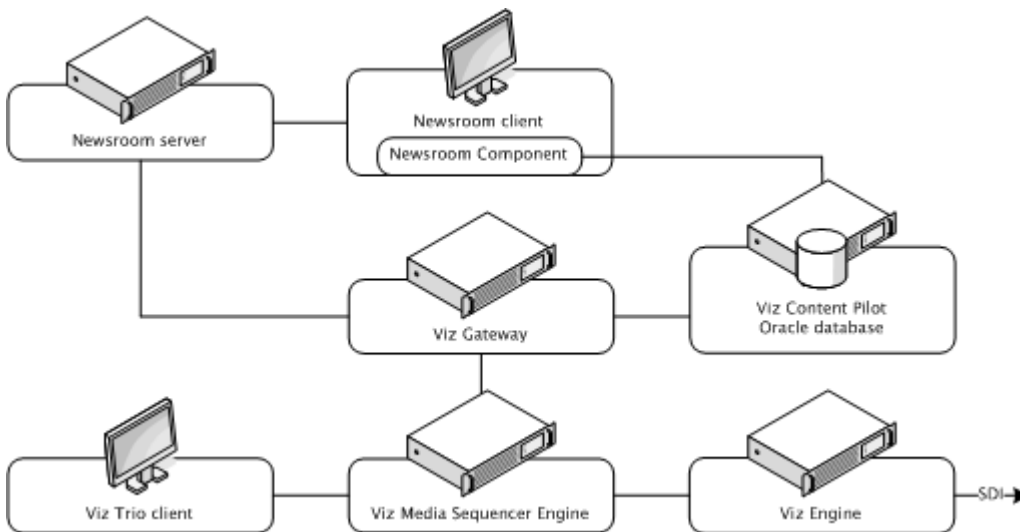
- [MOS](#)
- [Intelligent Interface](#)
- [General Purpose Input \(GPI\)](#)
- [Video Disk Communication Protocol \(VDCP\)](#)

- [MCU/AVS](#)
- [Socket Object Settings](#)
- [Proxy](#)
- [Viz One Configuration](#)

4.7.1 MOS

To view the MOS configuration settings, click **Connectivity** > **MOS** in the Trio Configuration.

MOS Newsroom Integration




A Viz Gateway (MOS) connection lets Viz Trio monitor playlists (see [Playlist Modes](#)) from any Newsroom Computer System (NCS) that supports the Media Object Server (MOS) protocol.

NCSs that support the MOS protocol can deliver newsroom playlists to Viz Trio through the Viz Gateway and Viz Pilot's database; hence, a MOS integration requires a Viz Gateway, an Oracle database and Viz 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 MOS protocol and/or [Intelligent Interface \(IIF\)](#) protocols (for example, Avid iNEWS ControlAir).

Viz Gateway, MOS Configuration Settings

 Trio Configuration

User Interface

- General
- Keyboard Shortcuts and Macros
- Text Editor
- Page List/Playlist
- Paths
- Local Preview
- User Restrictions

Output

Graphic Hub

Viz Pilot Database

- Database Settings
- Picture Database

Connectivity

- MOS
- Intelligent Interface
- GPI
- VDCP
- MCU/AVS
- Socket Object Settings
- Proxy
- Viz One

Import/Export Settings

Enable Gateway Configuration

Viz Gateway Connection

Start
Stop
Restart

Viz Gateway Connection Config

0 Messages per second limit (0=disabled)

Viz Gateway Host	Viz Gateway Port
<input type="text" value="10.211.112.37"/>	<input type="text" value="10640"/>
MOS ID	NCS ID
<input type="text" value="PILOT"/>	<input type="text" value="COLUMBUS"/>

Send status changes back to NCS


Send channel assignment changes back to NCS

OK
Cancel
Apply
Reset

- **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 Config:** Define Viz Gateway connection settings.


- **Messages per second limit:** 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.
- **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.
- **Send status changes back to NCS:** Sends new status changes to the NCS.
- **Send channel assignment changes back to NCS:** If enabled, will instruct the Media Sequencer to update channel assignments back to the Newsroom system. Hence the next time the MOS playlist gets updated the changed channel assignment stays at the changed element in Trio and is not overwritten by the previous channel value.

The [Database Settings](#) for the [Viz Pilot Database](#) on the Media Sequencer must be established for the Viz Gateway integration to work.

 **Note:** MOS and NCS ID are not needed for Viz Gateway versions 2 and newer.

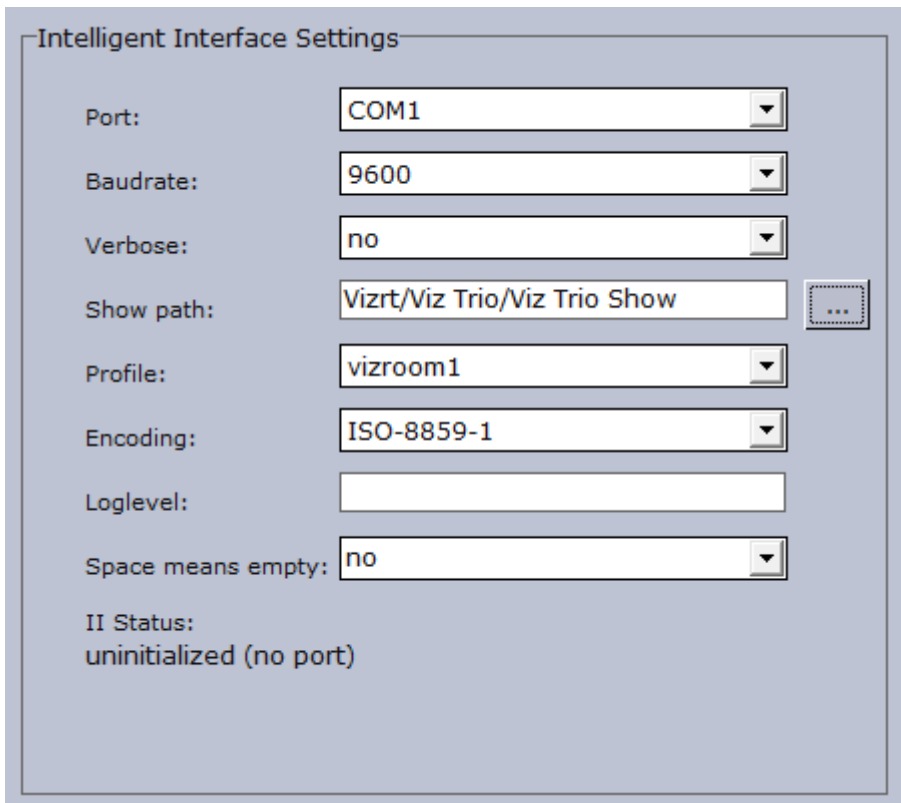
Configuring a Viz Gateway Connection

1. Click the **Config** button to open the Configuration Window.
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 Window 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.

4.7.2 Intelligent Interface

To view the Intelligent Interface configuration settings, click **Connectivity > Intelligent Interface** in the Trio Configuration.



Intelligent Interface Settings

Port:

Baudrate:

Verbose:

Show path:

Profile:

Encoding:

Loglevel:

Space means empty:

II Status:
uninitialized (no port)

The intelligent interface (IIF) configuration is used to set the parameters the automation system uses to connect to the Media Sequencer over the intelligent interface protocol. Further it defines the Viz Pilot playlist or Viz Trio show the automation system can control and on which output profile.

The Media Sequencer listens to a communications port and acts as the Character Generator (CG) device side of the intelligent interface. 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:

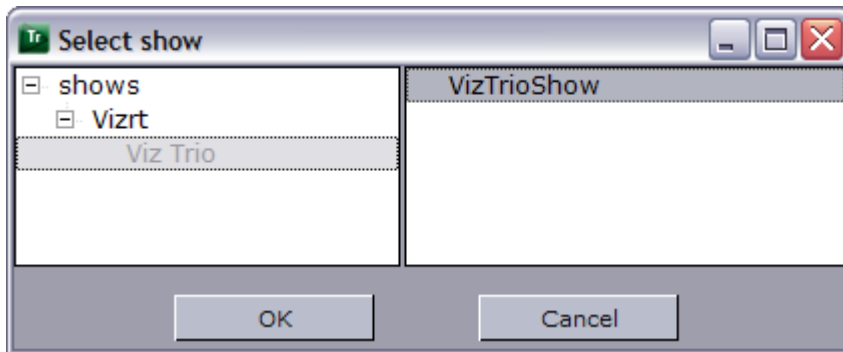
- Receive callup data from a newsroom system and store it in the schedule.
- Trigger actions based on simulated keyboard commands it receives.

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 get displayed in the Media Sequencer console. This feature is useful when setting up the system.

- **Show path:** Sets the path of the playlist or show.

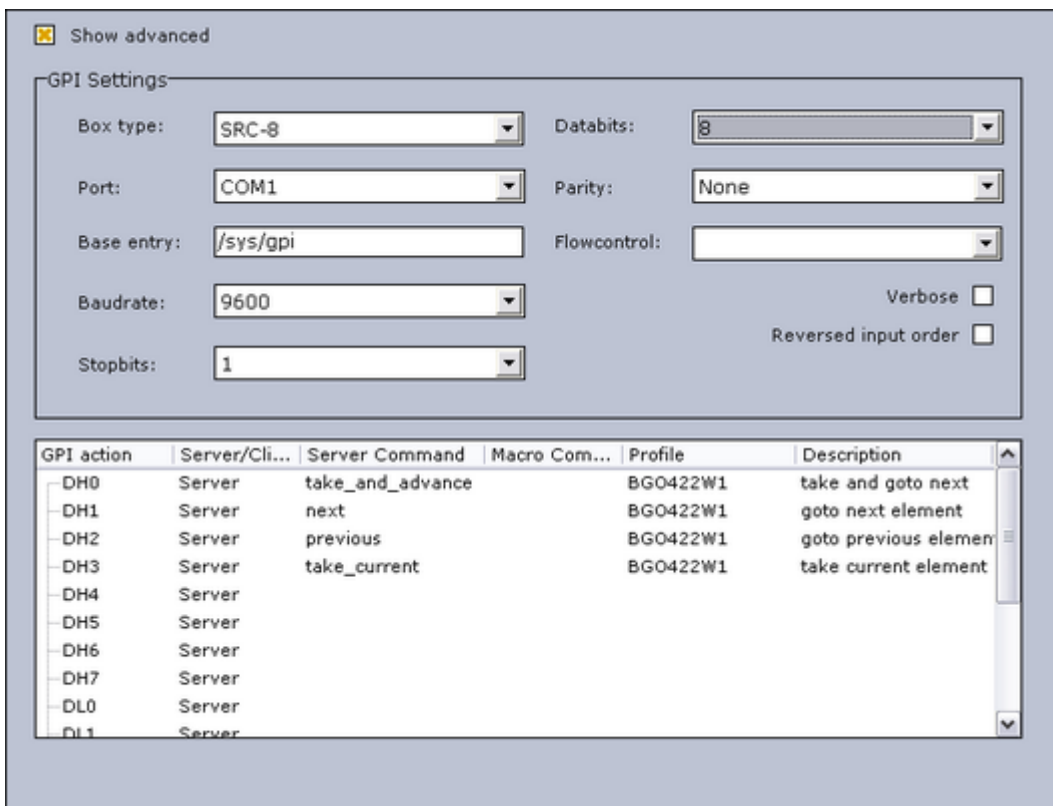
Select Show:



- **Profile:** Sets the appropriate profile to be used.
- **Encoding:** Sets the appropriate font encoding for the connection.
- **Loglevel:** Sets the Media Sequencer Logging 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.

4.7.3 General Purpose Input (GPI)

To view the GPI configuration settings, click **Connectivity > GPI** in the Trio Configuration.



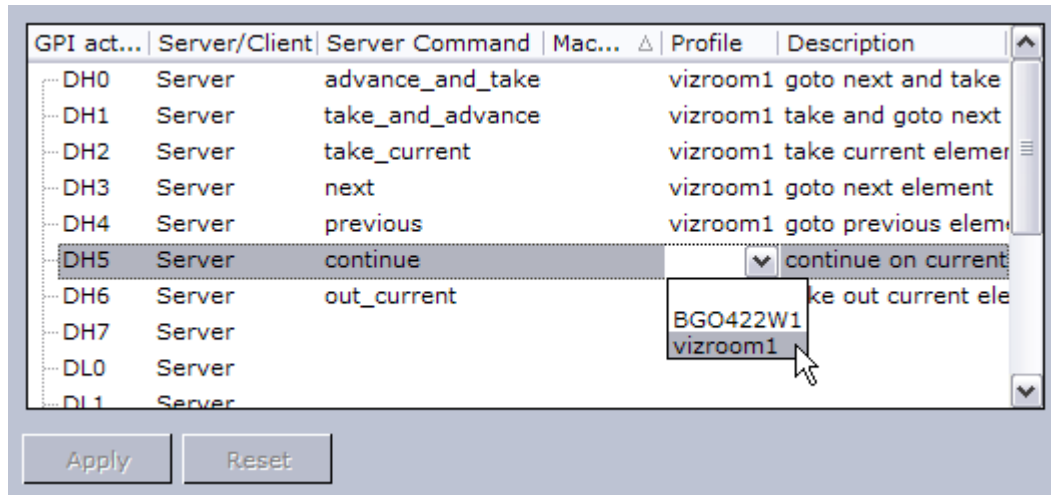
The GPI settings allows the Media Sequencer to be configured to handle GPI commands. The commands can be handled by the Media Sequencer itself (Server Command) or forwarded to the Viz Trio client (Macro Command).

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) 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, the 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.
- **GPI action:** Shows a list of the available GPI actions.
Commands and actions list:



- **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.

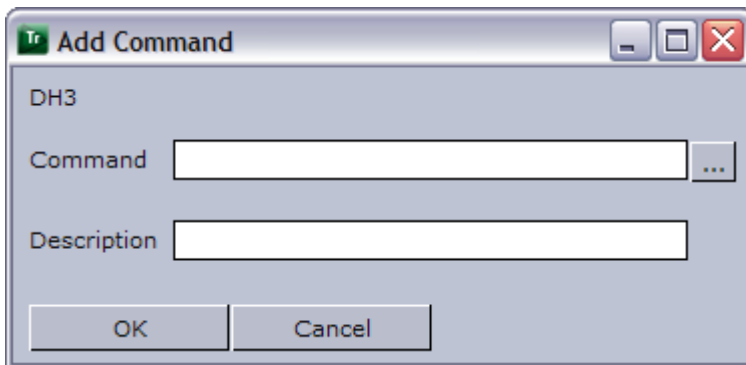
Assigning a Server Command

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**.

Assigning 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**.

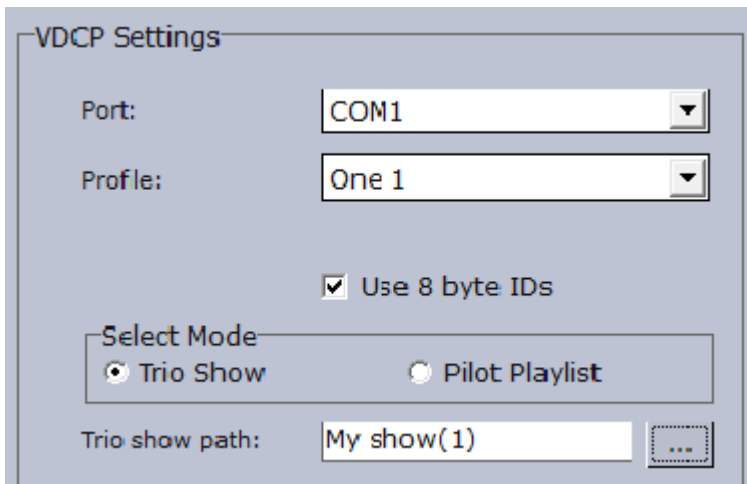
Adding a Macro Command



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.

4.7.4 Video Disk Communication Protocol (VDCP)

To view the VDCP configuration settings, click **Connectivity > VDCP** in the Trio Configuration.

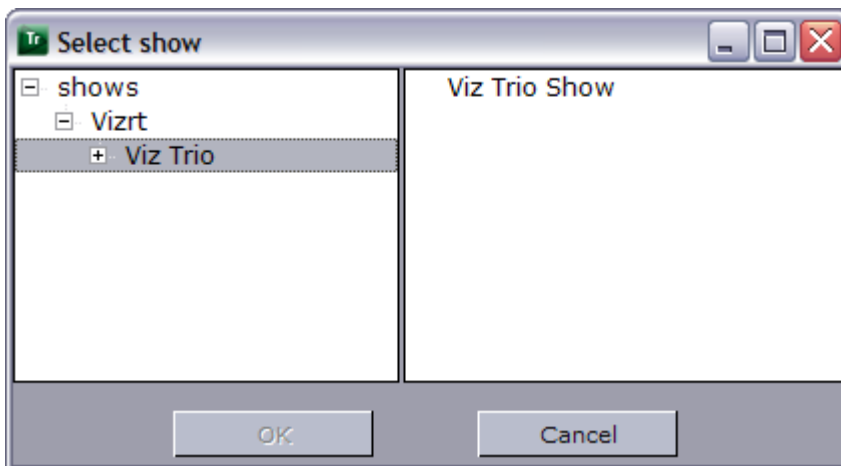


The Video Disk Control Protocol (VDCP) configuration allows a VDCP connection for Media Sequencer to be established in order to externally control a Viz Pilot playlist or Viz Trio show.

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.
- **Use 8 byte IDs:** Allow the use of 8 character IDs for VDCP protocol.
- **Select Mode:** Select a Viz Trio show or a Viz Pilot playlist path mode.
- **Trio path / Pilot playlist:** Sets the base directory for the VDCP integration. Video clips will be placed here.

4.7.5 MCU/AVS

To view the MCU/AVS configuration settings, click **Connectivity > MCU/AVS** in the Trio Configuration.

MCU/AVS Configuration Settings

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: /

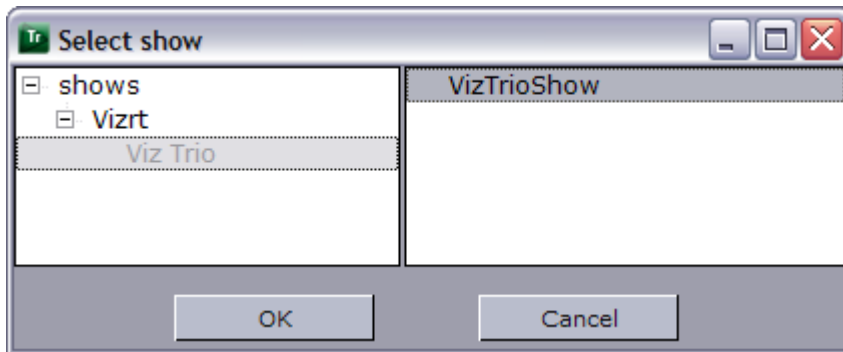
Ignore incorrect events

Unconditional deletes

This configuration is primarily used for configuring settings for a Newstar (News*) 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.

4.7.6 Socket Object Settings

To view the Socket Object configuration settings, click **Connectivity > Socket Object Settings** in the Trio Configuration.

Socket Object Configuration Settings

The Socket Object Settings section allow 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.

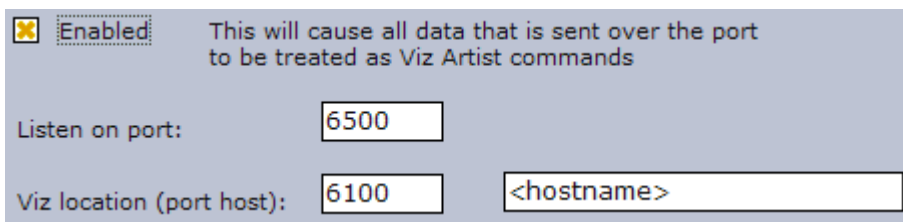
- **Socket type:**
 - **Client socket:** When Viz Trio is set up to be a client socket it will connect to the specified server socket. Specify a host name and a port for the machine that runs the server socket.
 - **Server socket:** When Viz Trio is set up to be a server socket it will wait for a client 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 [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.

4.7.7 Proxy

To view the Proxy configuration settings, click **Connectivity > Proxy** in the Trio Configuration.

Proxy Configuration Settings



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).

4.7.8 Viz One Configuration

To view the Viz One connectivity settings, click **Connectivity > Media Engine** in the Trio Configuration.

Viz One Connectivity Settings

The Media Sequencer's communication with Viz One is related to all Media Sequencer show and playlist elements that contain Viz One elements residing on the Viz One that are initialized for playback.

- **Enable the Viz One handler:** If selected, enables the Viz One handler.
- **Service document URI:** Defines the Viz One instance to use.

⚠ Note: When the service document URI of the Viz One is changed, remember to manually change the Viz One storages for each video handler in the [Output](#) section of the configuration. A Viz One storage points to a Viz Engine where the Viz One files are sent for playback. For more information, see [To add a video device](#).

It's possible to define the Viz One server instance based on either an IP address or hostname. It is recommended to use hostname. Host comparisons in the system are generally done by string comparison, not by lookup. This is why *either* IP or hostname must be selected. This also means that you should not mix hostnames and fully qualified domain names; http://vizione01.vizrt.com/thirdparty_ is not the same as http://vizione01/thirdparty_, even if it is possible to ping both. The system allows using both *http* and *https*, although *http* is recommended. Whatever options being selected (IP vs. hostname, hostname vs. domain name, http vs. https), choose one or the other, and stick with it throughout the entire setup process.

⚠ Note: Viz One version 5.3 and later uses */api* and not */thirdparty*.

- **Username:** Searches Viz One using the following Viz One username.
- **Password:** Searches Viz One using the following Viz One password.

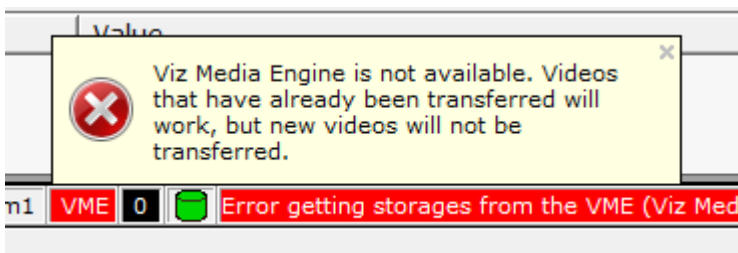
- **Viz Preview Server Host:** Specify the host of the Preview Server (*hostname:port*). This service is used to generate preview graphics in the timeline editor. If the *ZeroConf* service is installed and running, you can also select servers on the local network.

Configuring a Viz One Connection

1. Make sure that the **Enable the Viz One handler** check box is selected.
2. Type the hostname of the relevant Viz One in the **Uri to service document** box.
3. Enter working login credentials in the **Username** and **Password** boxes.
If the Viz One configuration is successful, a green Viz One icon will appear in the **Status Bar**.

Viz One Error Message

If the configuration is not working properly, the Viz One icon will be red. Scenarios where the icon is red can for example be a Viz One-search that fails, or a more serious error; a non-responsive Viz One with no video playout. In such serious situations where the Viz One is unavailable, the following Viz One error message will appear, stating that videos that have already been transferred to Viz will still work, but any new videos will not be transferred.



Configuring Viz One Server Time-out Handling

In the event of a network failure or other unforeseen events the Trio client can be instructed by scripting to switch to a backup Viz One server. In order to prepare for this you should:

1. Use the macro commands `settings:set_mam_service_document_uri [uri [username password]]` and `settings:set_media_search_credentials username password` to instruct the switch to an alternate Viz One server.
2. Update the active configuration (will also update the Media Sequencer VDOM) with the new publishing point using the command: `channelcontrol:set_asset_storage <host-name> <storage-name>`
3. Make sure that the timeout value for the Media Sequencer VDOM value `mam/get_publishing_points_timeout` is appropriate for your network environment. By default it is 10000 (ms), i.e. 10 seconds. If the value is too low, the `settings:set_mam_service_document_uri` command could possibly return with error without making the required change due to perhaps network glitches. Note that you will have to restart your Trio client for a change to this parameter to take effect.

In order to get or set the timeout value you can:

- Get the current value with: `settings:get_setting mam/get_publishing_points_timeout`
- Set the timeout with: `settings:set_setting mam/get_publishing_points_timeout [VALUE IN MS]`. To allow for (max) 15 seconds : `settings:set_setting mam/`

```
get_publishing_points_timeout 15000
```

A basic switchover-script example:

```
settings:set_mam_service_document_uri http://MyNewVizOneServer/api
settings:set_media_search_credentials MyUserName MyPassword
channelcontrol:set_asset_storage MyNewVizOneServer The Video server
```

4.8 Graphic Hub

Viz Trio must be configured to access the same graphics data as the Viz Engine program and preview channels in order to enable local preview.

Note: Since all Viz Artist and Viz Engine machines in a Viz Trio system must use the same data, it's recommended that the data be located on a high performance machine.

Configure the Graphic Hub database:

- at startup using the Viz Engine login window
- through Viz Trio's own configuration interface
- by setting command line options in Viz Trio's target path

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

To view Graphic Hub configuration settings, click **Graphic Hub** in the Trio Configuration.

4.8.1 Graphic Hub Configuration Settings

Nameserver Host	<input type="text" value="<hostname>"/>
Database	<input type="text" value="VizDb"/>
User Name	<input type="text" value="Guest"/>
Password	<input type="password" value="*****"/>
<input checked="" type="checkbox"/> Auto Login	

The Graphic Hub category contains settings for logging onto the Graphic Hub. All settings are local to the specific Viz Trio client.

- **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.

4.8.2 Configuring the Viz 3.x Database Using Viz Config

The screenshot shows the following configuration parameters in the Viz Config tool:

- Host Name: slartibartfast
- Hub: VizDbServer
- Port Number: 19396
- User: Guest
- Replication Timeout (sec): 15 (with a spin button and a 'R' icon)
- Show Auto Login: No
- Auto Login: Yes
- Import by Name: No
- Popup Server Messages: Enabled (with a note: 'Disabled' works on localhost only!)
- Date Format: EU (US is also visible)
- Archive Bit-Mode: 32 Bit (64 Bit is also visible, with a note: Enable 32 Bit only if you want to ensure that archives created by this version of Viz Artist can be imported by 3.0.0 versions.)
- Temporary Folder: c:/temp (with a browse button)


1. Start the **Viz Config** tool and select the Database section.
2. Enter the following connection parameters:
 - **Host Name:** Sets the database hostname.
 - **Hub:** Sets the database name.
 - **Port Number:** Sets the database port number. Default port is **84932**.
 - **User:** Sets the database user.
 - **Auto Login:** Sets login automatically 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.

4.8.3 Configuring the Viz 3.x Database Using the Viz Engine Login Window

1. Start Viz Trio.
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. Select the **Auto Login** check box.
5. Click **OK** to continue.


4.8.4 Configuring the Viz 3.x Database Using the Viz Trio Configuration Window


1. Start Viz Trio with the following target parameter: *-control*
2. Click the **Config** button (upper left) to open the Configuration Window.
3. Select the **Graphic Hub** section, and enter the following parameters:
 - Name server host, database, username, and password
4. Select the **Auto Login** check box.
5. Click **Apply** to save the settings, and exit the Configuration Window. The change of database is immediate, however, it might be necessary to refresh the *Import Scenes* view to see the change when importing scenes.

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

4.8.5 Configuring the Viz 3.x Database Using the Viz Trio Target Path

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

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

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

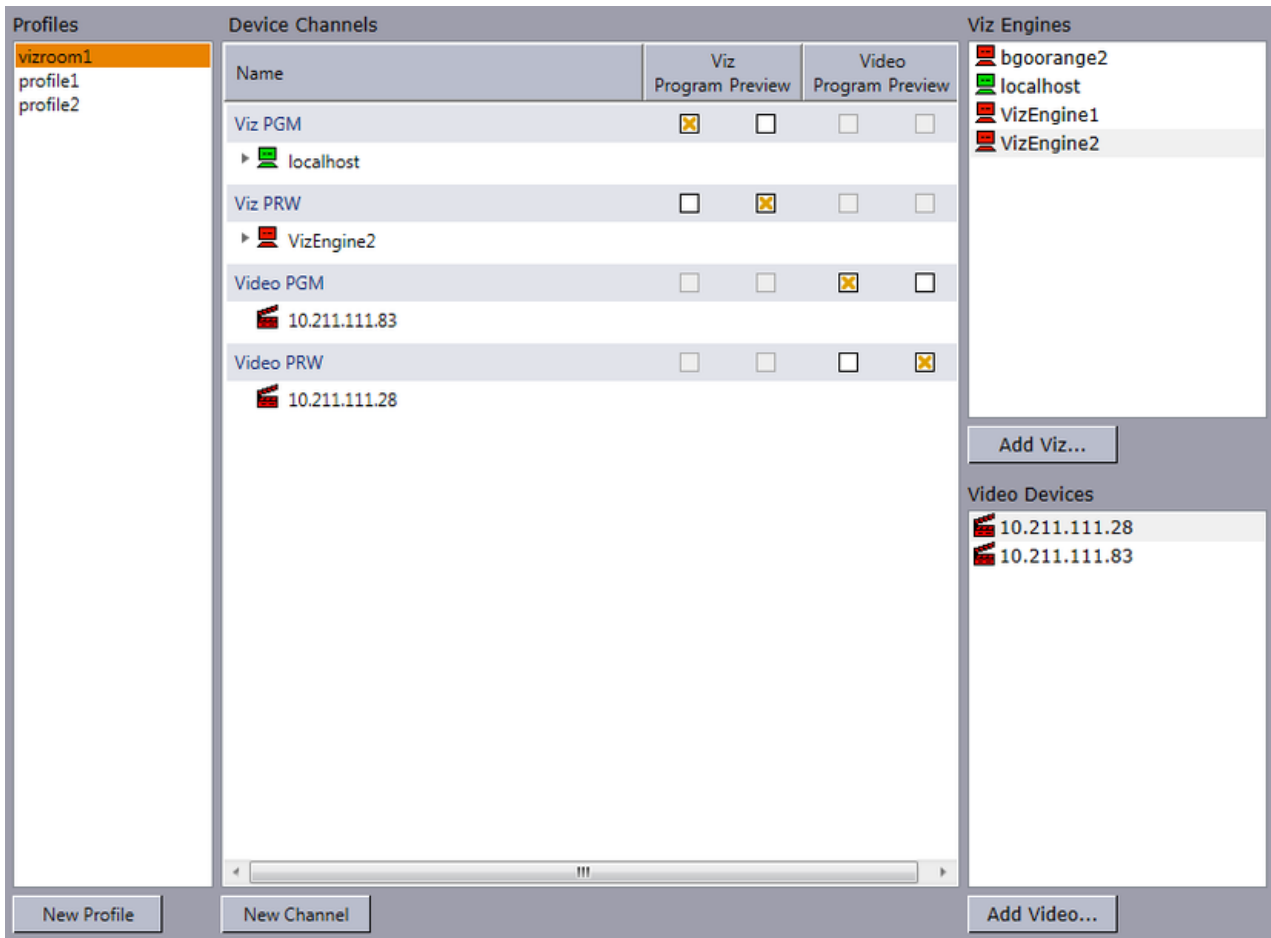
See Also

- [Viz Engine Administrator Guide](#)

4.9 Output

To view the Output configuration settings, click **Output** in the Trio Configuration.

4.9.1 Output Configuration Settings



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. The most common step is to define a studio with channels mapped to renderer machines (Viz Engine hosts).

Viz Trio setups 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 Engine host named *LocalProgramChannel* with host *localhost:6800*. See [Viz Trio OneBox Configuration](#).

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

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

4.9.2 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.
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 (for example in /etc/hosts), but only the NetBIOS name can be used (the name obtained when running the "hostname" command). The domain name cannot be used (for example: <hostname>.domain.internal).

This section covers the following topics:

- [Program and Preview Channels](#)
- [Profile Setups](#)
- [Forked Execution](#)
- [Standalone Scenes](#)
- [Transition Logic Scenes](#)
- [Visible Containers](#)

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 are updated immediately when text and images are added to the page. Select which elements to edit by pointing and clicking on them with your cursor 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.

4.9.3 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 (for example Viz One), 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.

IMPORTANT! Current limitations require both the graphics and video channels to use an IP address.

Forked Execution

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

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, by defining channels with different render engine setups, Transition Logic scenes 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 another.

For example, if 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


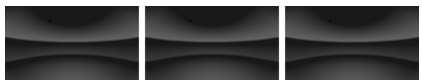
If you have a scene with four layers, each layer can be controlled separately from the other layers (see table below). By setting a state per layer you can achieve a varied output depending on the channel used and how that channel is configured in terms of render engines (see table above).

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

Layer 1 States	Layer 2 States	Layer 3 States	Layer 4 States
Show cube	Show text	Position left	Next image
Hide cube	Hide text	Position right	Show logo
<ignore>	<ignore>	Position center	<ignore>
		<ignore>	

The scene layers and configured channels above provide 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	Layer state	Output on Viz Engine 1, 2 and 3
Channel B(Viz 1,3)	Hide cube	
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.

4.9.4 Working With the Profile Configuration

This section covers the following topics:

- [Profile Configuration](#)
- [Channel Configuration](#)
- [Output Device Configuration](#)

4.9.5 Profile Configuration

Opening Profile Configuration

1. Click the **Config** button (upper left), and then select **Output**.
2. Right-click the profile on the status bar (lower-left) and select **Profile Configuration...** from the appearing context menu.

Adding a New Profile

- In the Profile Configuration window click the **New profile** button, and in the appearing text field enter a new unique profile name, and press **ENTER**.

Renaming a Profile

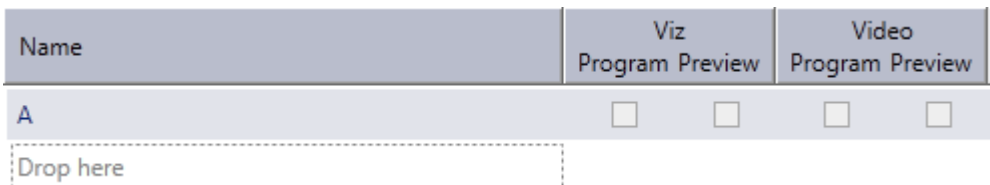
1. Right-click the profile and from the appearing context menu select **Edit Profile Name**, or double-click it and enter the new name.
2. When finished editing the name, press **ENTER** or click the cursor outside the Profiles list.

Deleting a Profile

- Right-click the profile and from the appearing context menu select **Delete Profile**, or select it and press the **DELETE** button.

4.9.6 Channel Configuration

Adding an Output Channel to the Channels List

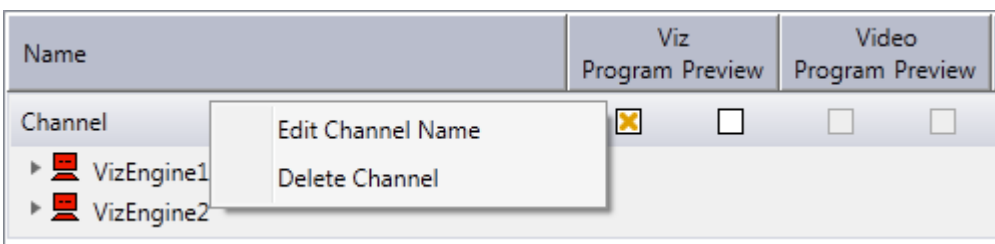


- Click the **New channel** button, or drag and drop a Viz Engine or video device to the Channels list.

Renaming an Output Channel in the Channels List

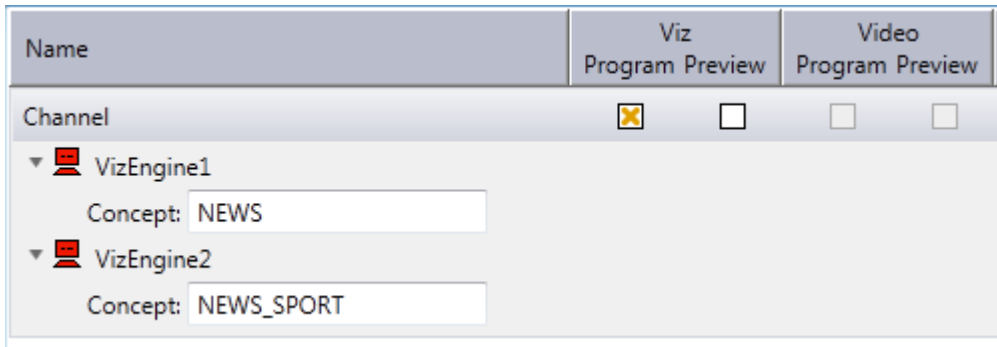
- Right-click the channel and select **Edit Channel Name** from the context menu, or double-click the name.

Removing an Output Channel from the Channels List



- Right-click the channel entry and select **Delete Channel** from the context menu, or select the channel and press the **DELETE** button.

Adding a Concept Override for a Channel Output Device

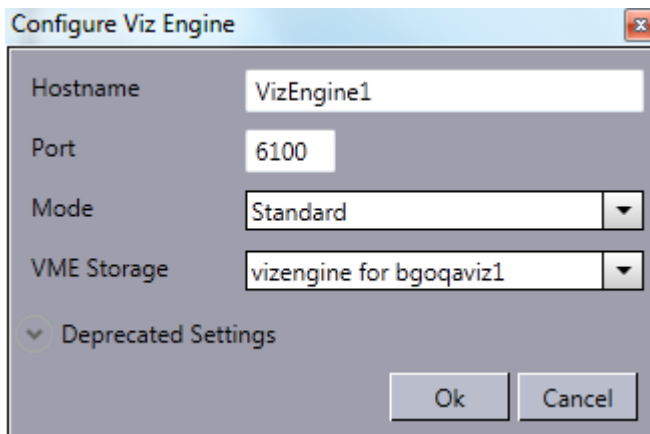


1. **Expand** the channel's output device and append the concept name. This overrides any concepts set elsewhere.
2. Click **OK**.

Caution: Note that concept names are case sensitive.

4.9.7 Output Device Configuration

Adding 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.
3. Default port for Viz Engine is 6100.
4. Optional: Select Mode.
5. Optional: Select an Asset Storage location. Asset Storage lists available Viz Engine storage for clip transfer and playout.
6. Click **Ok**.
 - A status indicator will show if the renderer is On Air.

Adding 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. Type the **Host or IP** address.
4. Enter the **Port** number (optional).
5. Select a publishing point from the **Asset Storage** list, so that stand-alone clips and clips used in pages or data elements are transferred to the right location (the specified Viz Engine) for payout.
6. If the **Fullscreen** check box is selected, you can define additional values (**Use Transitions** and **Fullscreen Scene**) to provide transition effects from one video to another.


Note: The fullscreen values are only available for Viz Engines, not MVCP video devices.

IMPORTANT! Trio by default sets the "Fullscreen Scene" in Media Sequencer to "Vizrt/System/transitions" when you switch on "Use Transitions". Using this default, clip channel must be 1 and the alternate clip channel must be 2. These settings can only be changed when using other transition scenes.

7. Select the **Clip Channel** to use.

Note: Viz Engine 3.6 and above supports up to 16 clip channels for payout. A clip channel might be unavailable if it's configured to be inactive in Viz Config or the license only covers a limited number of clip channels., for example. Note that Media Sequencer prior to version 2.0.1 imposed some restrictions, see note below.

8. **Alternate Clip Channel:** When using videoclip transitions the clip from "Clip Channel" will transition to the clip from "Alternate Clip Channel" and the next clip will transition from "Alternate Clip Channel" back to the "Clip Channel". The transition clip channel is enabled in the configuration window only if **Use Transitions** is enabled.
9. Click **OK**.
 - A status indicator will show if the video device is online.

 **IMPORTANT!** Media Sequencer up to and including version 2.0.0 imposes the following limitations: Clip channels 3-16 not supported when using *fullscreen* clips. Clip channels 2-16 not supported when using *transitions*. Using *alternate* clip channel setting not supported. These restrictions are lifted starting with Media Sequencer version 2.0.1 and higher.

 **Tip:** If you to transfer video from Vizrt's MAM systems you can use the *default* profile (unless you configure a profile manually in both systems yourself).

Editing a Viz Engine or Video Device

- Right-click the device and select **Edit** from the context menu, or **double-click** it.

Deleting a Viz Engine or Video Device

- Right-click the device and select **Delete** from the context menu, or select it and press the **DELETE** button.

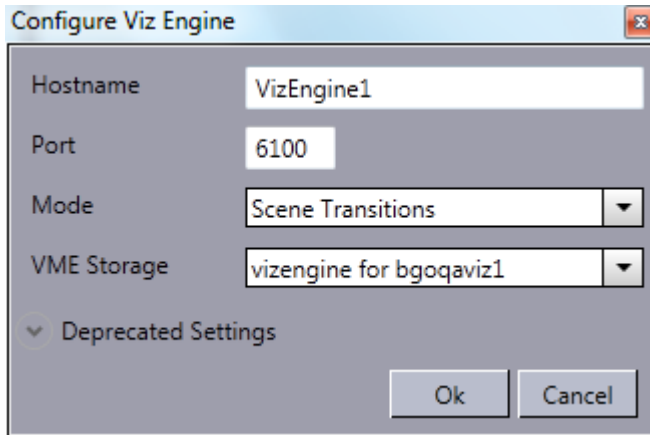
Adding an Output Device to the Channels List

- **Drag and drop** a Viz Engine or video device onto the channel in the Channels list, or select it and select **Add to profile** (creating a new channel) or **Add to selected channel** from the context menu.

Removing an Output Device from the Channels List

- Right-click the Viz Engine or video device and from the context menu select **Delete Output**, or select it and press **DELETE**.

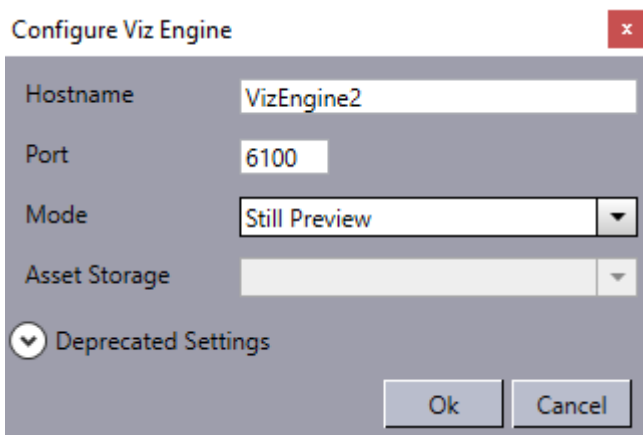
Enabling Scene Transitions



1. Configure the Viz Engine settings as seen in how [Adding 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 Properties](#), 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.

Enabling Still Preview



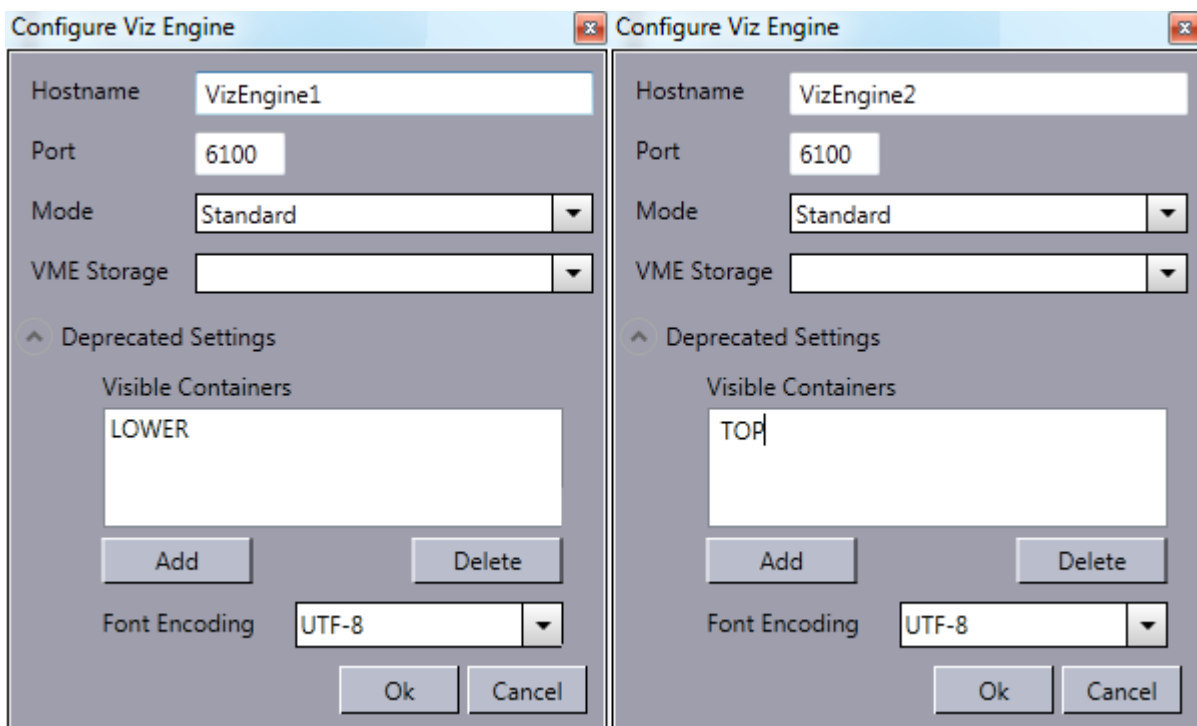
1. Create two channels, one for program and one for preview.
2. Configure the same render engine twice as seen in how [To add a Viz Engine](#).
3. For the second render engine set the **Mode** to **Still Preview**.
 - **Still Preview:** Allows you to use the program output to get a still preview. To 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. Note that the port used for still preview is **6100**.

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.

Rendering 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**.

Changing 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**.

4.9.8 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.

4.10 Command Line Parameters

Use the command line parameters below to customize Viz Trio startup.

4.10.1 Adding a Command Line Parameter

- Right-click the program shortcut, and edit the program target path.

Example:

```
C:\Program Files (x86)\vizrt\Viz Trio\trio.exe -mse localhost -control -force_gpu_count 2
```

4.10.2 Command Line Parameters

Parameter	Value Syntax	Description
-attach_to_viz		Attaches to a running viz.exe without asking the user to terminate it. If a Viz Engine process is already running locally then this command line option avoids the notification "viz.exe is already running. Terminate?" So it always keeps the viz.exe process and attaches to it without asking. If there is no Viz Engine process running Trio start up its own process as before no matter if the command is present or not. This command is just a flag and does not have any parameter value.
-control		Allows you to configure Viz Trio.
-folder	show path	Causes the client to start in the specified show.
-force_gpu_count	integer	Lets the user force/override the available (detected by Viz) number of GPUs used for preview and program channel.
-hide-auto-login		Hides the option to auto login into the Graphic Hub database in the Viz Engine startup dialog box.
-local_program_channel_port	integer	Overrides the configured port number set for the local program channel. This is useful if a different port than default port 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.
-macro-port	integer	Overrides the port of the macro command server. The default port is 6200.
-mse	hostname	Media Sequencer hostname.

Parameter	Value Syntax	Description
-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	Redefine the main window title of Trio. %v will be replaced by the version string. %h will be replaced by the MSE-host. %s will be replaced by the current show path. Default, if -t not specified: "Viz Trio %v - MSE: %h - Show: %s"
-usersetting	settings name	Use something other than the hostname as an identifier for user settings.
-viz-console-delay	<time in seconds>	Starts Viz Trio independently 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.

Parameter	Value Syntax	Description
-viz-startup-timeout	integer	Specifies the time-out to use during start-up of Viz Trio when communicating with Viz Engine. The default time-out value is set to 30 seconds. Time-outs during start-up will often leave Viz Trio unusable. After Viz Trio has successfully started up, the time-out setting in the Local Preview is used.

4.11 Settings

This section covers the following settings:

- [Video Codecs](#)
- [Codecs Installation Options](#)
- [Installing Codecs for Local Preview](#)
- [Setting a Preferred Decoder](#)
- [Shared Data](#)
- [Ports](#)

4.11.1 Video Codecs

Video codecs are only required to preview videos embedded in graphics. Full screen videos can be viewed by using Timeline editor without using a codec.

⚠ IMPORTANT! Due to licensing requirements, Vizrt does not provide the codecs required for local preview. Users must obtain and install their own codecs.

⚠ Note: High resolution playout on Viz Engine does not require these video codecs.

Follow the procedures below to complete installation:

- [Installation Options](#)
- [Installing Codecs for Local Preview](#)
- [Setting a Preferred Decoder](#)

4.11.2 Codecs Installation Options

Codecs are available from several suppliers. Here are a few suggestions:

- FFDSHOW MPEG-4 video decoder and Haali Media Splitter
- LAV Filter video decoder and Splitter

- Main Concept video decoder and splitter

4.11.3 Installing Codecs for Local Preview

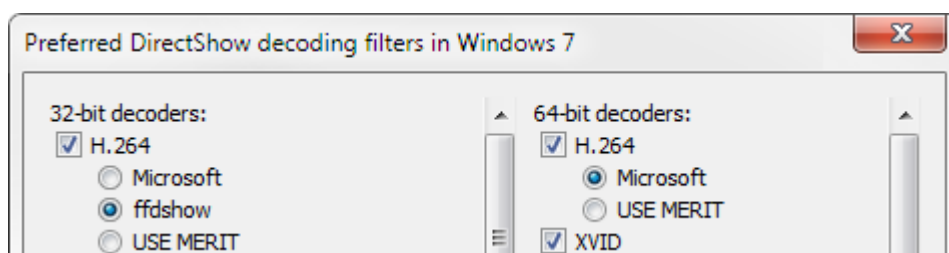
The example below shows how to set up support for h.264 playback using the FFDSHOW MPEG-4 codec package and a Matroska Splitter from Haali.

Note: You need to have your own license for clip playback as FFDSHOW does not come with a decoding license.

1. Make sure you do not have any other codec packages installed on the machine that interfere with FFDSHOW or the media splitter.
2. **Download** the Matroska Splitter from Haali
3. **Download** the Windows 7 DirectShow Filter Tweaker
4. **Download** the FFDSHOW MPEG-4 Video Decoder
 - Make sure you have a license to use the codec.
 - Make sure you download a 32-bit version of the codec.
5. **Uninstall** older 64-bit versions of the MPEG-4 codec.
6. **Install** the Matroska Splitter from Haali.
7. **Install** the Windows 7 DirectShow Filter Tweaker.
8. **Install** the FFDSHOW MPEG-4 codec.
 - After installing the FFDSHOW codec package make sure that no applications are excluded, especially Viz Engine (there is an inclusion and exclusion list in FFDSHOW).
9. Set your MPEG-4 32-bit decoder to FFDSHOW.
 - You should now be able to preview video clips from Viz One.

4.11.4 Setting a Preferred Decoder

1. **Run** the Windows 7 DirectShow Filter Tweaker.
2. Click **Preferred decoders** in the dialog.



3. Set your MPEG-4/H.264 32-bit decoder to ffdshow and click **Apply & Close**. Click **Exit**.

4.11.5 Shared Data

Vizrt recommends that customers who use remote shares for storing data use UNC (Universal Naming Convention) paths directly in the configuration, instead of mapped drives.

Example: `\\vosstore\images`

4.11.6 Ports

Component	Protocol	Port	Description
<i>trio.exe</i>	TCP	6200	Used for executing macro commands of the Viz Trio client over a socket connection.
<i>trionle.exe</i>	TCP	6210	Used by the Graphics Plugin for NLE to utilize Viz Trio for effect editing.

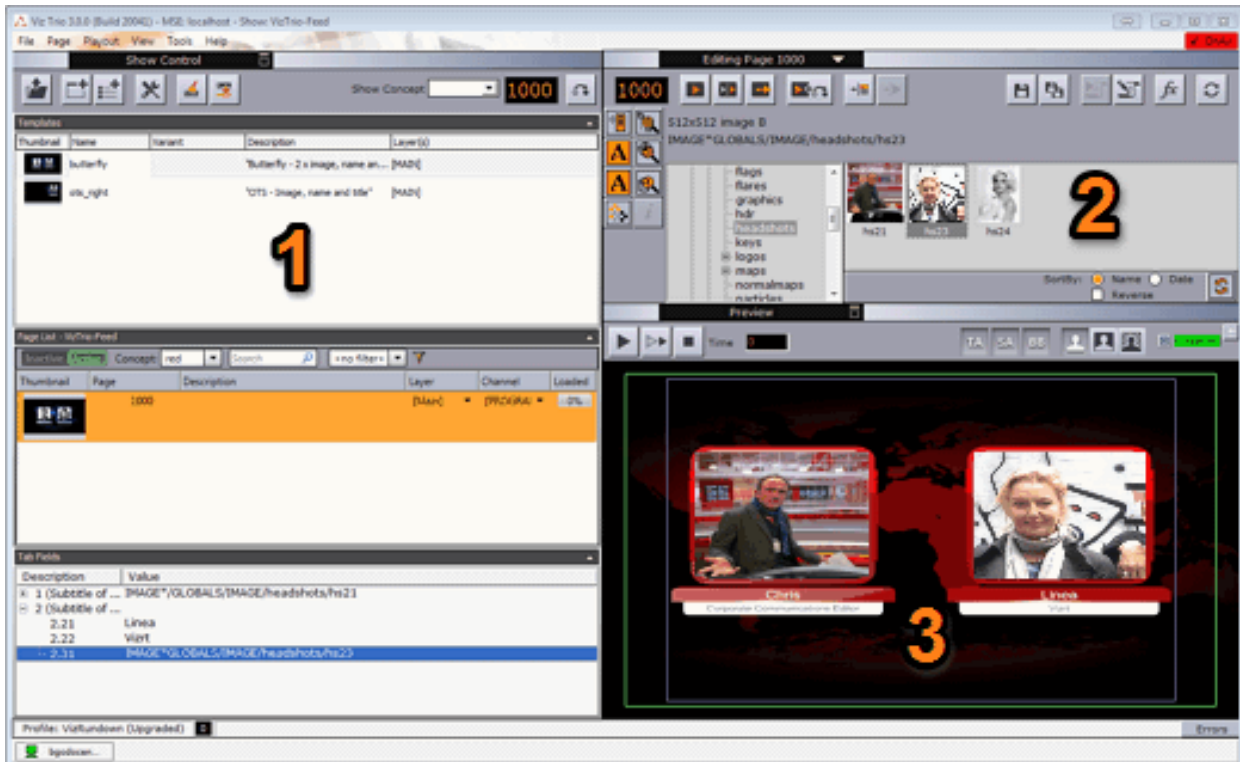
See Also

- [Timeline Editor](#)
- [Video Preview](#)

5 User Interface

This section describes the playout interface. The Viz Trio client's [Designing Scenes](#) and [Configuration Window](#) are not described in this section.

5.1 Interface Overview



Trio has three main areas (1,2,3 in the figure above) and a top main menu and status lines with options at the bottom of the main window.

- At the top, the main menu is grouped into sections [Menu option File](#), [Menu option Page](#), [Menu option Layout](#), [Menu option View](#), [Menu option Tools](#) and [Menu option Help](#).
- [1] On the left section of the screen you select and work with Templates, Pages, Pageview and edit tab values. Tab values are data entry points in templates and pages you can edit before the finished pages are played out.
- [2] The top right area contains editors and inspectors to view, select and change various properties for the objects you are working with.
- [3] The [preview window](#) is in the bottom right area. Trio 3 and newer versions allow you to detach the preview window so that you can have it on a second monitor, for example. Note: Viz Template Wizard pages can also be detached from the main Trio window.
- At the bottom, status bars display status information.
- At the bottom, status bars display status information.

5.2 Main Menu

The main menu at the top of the main window provides easy access to functions and workflows:



A brief description of the main menu sections is listed below. For certain menu items, a more detailed explanation can be found later in this chapter.

5.2.1 File

Option	Function
Open Show...	Show a directory panel to open a TrioShow.
Open External Playlist...	Show a directory panel to open an external playlist.
New Page View...	Add a new pageview.
New Show Playlist...	Add a new playlist.
Configuration	Show the Configuration window where you configure Trio.
Show Properties	View and configure properties for a TrioShow.
Import Viz Engine Archive...	Import Viz Engine Archive (.via) files from disk and re-import any affected templates in the current show.
Import Show Archive...	Import a TrioShow(.trioshow file) from disk.
Export Show Archive...	Export Show to disk as .trioshow file.
Export Selected Pages Archive...	Export Show with only selected pages including associated items, see Export Selected Pages Archive .
Slave Mode	Set Trio in Slave Mode, see Slave Mode .
Quit	Quit Trio.

5.2.2 Page

Option	Function
Save	Save the current page you're working on.
Save As...	Save the current page you're working on with a new name.
Rename...	Rename the current page you're working on.
Change Template...	Change the template for a page.
Refresh Linked Data	Asynchronously refresh the data for a page linked to a feed.
Reimport	Reimport the template design for a page.
Delete	Delete the currently selected page. You will be asked for confirmation.
Delete All	Delete all pages. You'll be asked to confirm.
Save to XML...	Save page to an XML-file on disk.
Load from XML...	Load page from an XML-file on disk.
Print/Save Snapshot...	Take a Snapshot, which can be printed or saved to disk.
Move to Number...	Move page to a specific number.
Move with Offset...	Move page with offset.
Copy to Number...	Copy page to a specific number.
Copy with Offset...	Copy page with offset.

5.2.3 Playout

Option	Function
Take	Performs a direct take on the selected page.
Pause	Pauses the current element.
Continue	Continues the animation on the selected page, or any page that is loaded in the same transition logic layer as the selected page.
Out	Takes out the selected page, or any page that is loaded in the same transition logic layer as the selected page. Hard cut.
Cue	Prepares the clip for playout so the first frame is ready in the player.
Cue Append	Prepares the clip for playout. The clip will start automatically when the current clip ends.
Take and Read Next	Takes the current element and reads the next one.
Cleanup Channels	Cleans up the renderers (Viz Engines) currently in use by the open show. Asks for confirmation. This frees up renderer memory and realtime capacity. See Cleanup Channels .
Initialize Show on Channels	Loads the selected show on the program and preview renderers. See Initialize Channels .
Initialize Show on Local Preview	Loads the selected show on the Preview renderers. See Initialize Channels .
Update on Program	Updates on Program renderer without reloading or rerunning animations. Only for Transition Logic scenes.

Option	Function
Update on External Preview	Updates on External Preview renderer without reloading or rerunning animations. Only for Transition Logic scenes.
Clear Program	Takes out the scene on Program renderer, but does not clean up the renderer (scenes will remain in memory).
Clear external Preview	Takes out the scene on Preview renderer, but does not clean up the renderer (scenes will remain in memory).
Start Scroll	See Scroll Control .
Stop Scroll	See Scroll Control .
Continue Scroll	See Scroll Control .
Arm Current	See Arm and Fire .
Arm Selected	See Arm and Fire .
Unarm Current	See Arm and Fire .
Unarm All	See Arm and Fire .
Fire All	See Arm and Fire .

5.2.4 View

Option	Function
Export Mode	Toggle. Enable Trio to operate in export mode.
Tabfields	Toggle. Enable tabfields.

Option	Function
Commands	Show the command window. In this window you can execute Viz Trio macros and commands using a command-line (cli) interface.
Errors	Show the error message window.
Running Carousels	Show the task window where all running tasks in the active profile will be displayed.
Refresh Thumbnails	Force refresh off all thumbnails.

5.2.5 Tools

Option	Function
Viz Artist	Switch over to Viz Artist. Trio will keep running. While in Artist, click the Trio button to return to Trio again. See Viz Artist Mode .
TimeCode Monitor	Start the TimeCode Monitor. See TimeCode Monitor .
FeedStreamer	Start the Feedstreamer moderation tool. See Field Linking and Feed Browsing in Viz Trio .
Trio Designer	Start the Design Tool. See Modes .

5.2.6 Help

Option	Function
View Help	Show help and documentation.
Legal Notices	Information about third-party codelibs and Viz Trio licensing.

Option	Function
About...	Displays which Viz Trio version and build that has been installed on your machine (essential information to attach when reporting issues to Vizrt), Credits and Legal Notices.

5.3 Modes

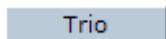
5.3.1 Playout and Design Mode

Clicking **Tools > Trio Designer** from the main menu when in playout mode switches the user interface to design mode, displaying the **Designing Scenes** 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.

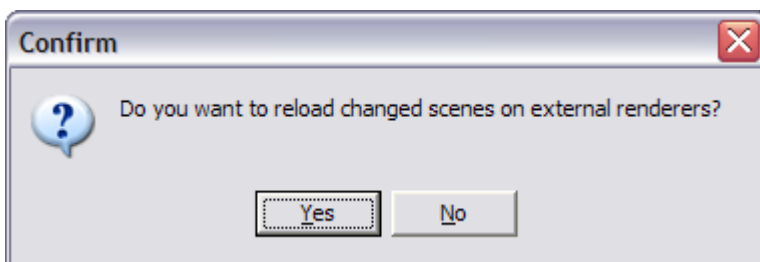
Click **Tools > Trio Designer** again to return to playout mode.

5.3.2 Viz Artist Mode

Viz Trio templates are based on scenes created in Viz Artist. From the main menu, click **Tools > Viz Artist** to start Viz Artist, opening the template's scene that is currently loaded in Viz Trio's preview window. You can edit the scene, and save the changes in Viz Artist.



When in Viz Artist mode, Artist displays a button labelled **Trio** in the upper right corner. Click **Trio** to return to Viz Trio.



When you are back in playout mode, the confirmation dialog above appears. The template is re-imported automatically.

5.3.3 On Air Mode



The top bar to the contains the button above, which takes Viz Trio on and off-air. **OnAir** is *red* in on-air mode.

5.3.4 Slave Mode

- Select the main menu switch **File > Slave Mode** to enable/disable slave mode.
- Slave mode 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 typically makes sense when the commands to change the folder are issued from a newsroom system - the newsroom system is then master.
- To exit slave mode, click the Main menu option **File > Slave Mode** again.

5.3.5 Show Modes

Prior to version 2.11, Viz Trio supported two show modes: traditional shows where templates and pages were part of the same view, and [Context Enabled Shows](#) where templates and pages are split into two views. From version 2.11, Viz Trio only supports [Context Enabled Shows](#).

This section covers the following topics:

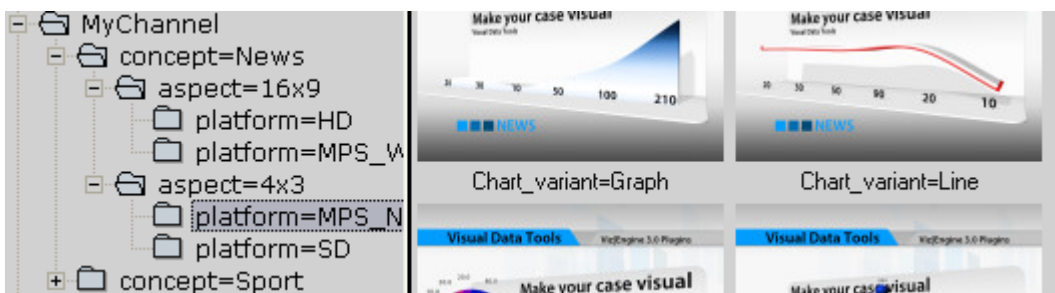
- [Context Enabled Shows](#)
- [User-defined Contexts](#)

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 (such as lower and top third).

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

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



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 naming conventions are as follows:

Concept Naming Conventions

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! It's important to add an underscore "_" to separate the scene name from the variant's keyword in the names of variants.

User-defined Contexts

Viz Trio supports the creation of user-defined contexts. The number of user-defined contexts is unlimited. However, user-defined contexts are only controllable through the Viz Trio 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 user-defined macro commands for use with scripts and shortcut keys.

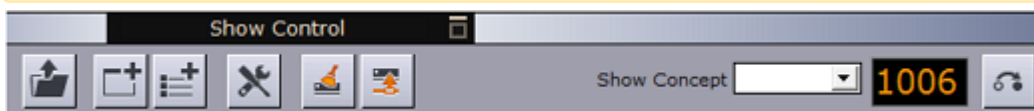
See Also

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

5.4 Show Control

Manage a show before, during and after it's taken on-air with the Show Control interface. Show Control gives you access to previous shows, Viz Pilot playlists, newsroom playlists and lets you create new shows and show playlists.

⚠ **Note:** there are differences between a show, a show playlist and a playlist, especially in the way they are managed and monitored; this chapter describes 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 Properties:** Opens the [Show Properties](#) window.
- **Cleanup Renderers:** Cleans up all data on the renderers (see the [Cleanup Channels](#) section).
- **Initialize:** Initializes the show on the renderers (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 be read from the page list.
- **Read Page:** Reads the page that has the callup code shown in the **Page To Read** window.

This section covers the following topics:

- [Show Directories](#)
- [Add Page List View](#)

- [Create Playlist](#)
- [Show Properties](#)
- [Cleanup Channels](#)
- [Initialize Channels](#)
- [Show Concept](#)
- [Callup Page](#)

5.4.1 Show Directories



Clicking the **Change Show** button displays the Show Directories view.

A [Shows](#) tab and a [Playlists](#) tab are displayed by default. However, a [Viz Directories](#) view is also available for compatibility with previous Viz Trio versions.

Viz Directories

The Show logic is the recommended way of organizing pages. However, it's 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 disabled by default. Open **Configuration**, and in the **User Interface/User Restrictions** section, enable **Browse viz directories when changing shows**.

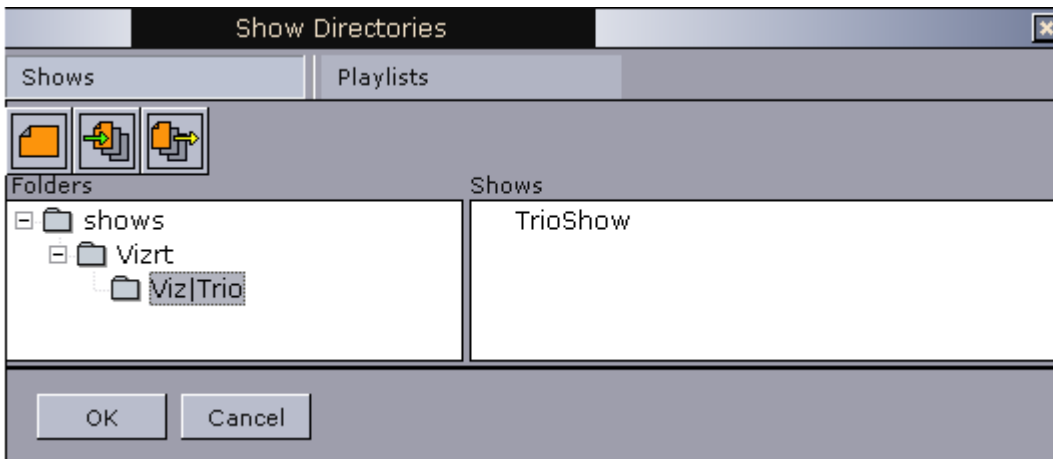


When the Viz Artist folder is set and pages are imported, a show folder with the same path and name will be created in the Shows view. To keep using the Viz Artist directory workflow, where each show is tied to a Viz Artist folder, simply ignore the Shows view.

It's possible to switch at any time to the Shows view logic. However, these shows can only be accessed using the Shows view once show folders with paths that differ from the Viz Artist scene data structure have been created or if there is more than one show per folder.

Shows

A show folder is used to organize shows that belong to a certain show or production. Scenes from the whole Viz Artist scene tree can be imported into a show. From version 2.11, Viz Trio only supports [Context Enabled Shows](#). You can also click [Create Playlist](#) to play out pages in a carousel-like manner. All scenes added to a show are added to the **Templates** list.



The **Folders** and **Shows** panes contain context menus that provide options for show management.



The buttons are used for creating new shows, and to import and export shows. The **Import** and **Export Show** buttons let you export and import shows 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

- **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:** Make sure to export or delete all shows and sub-folders before deleting a root folder.

Shows

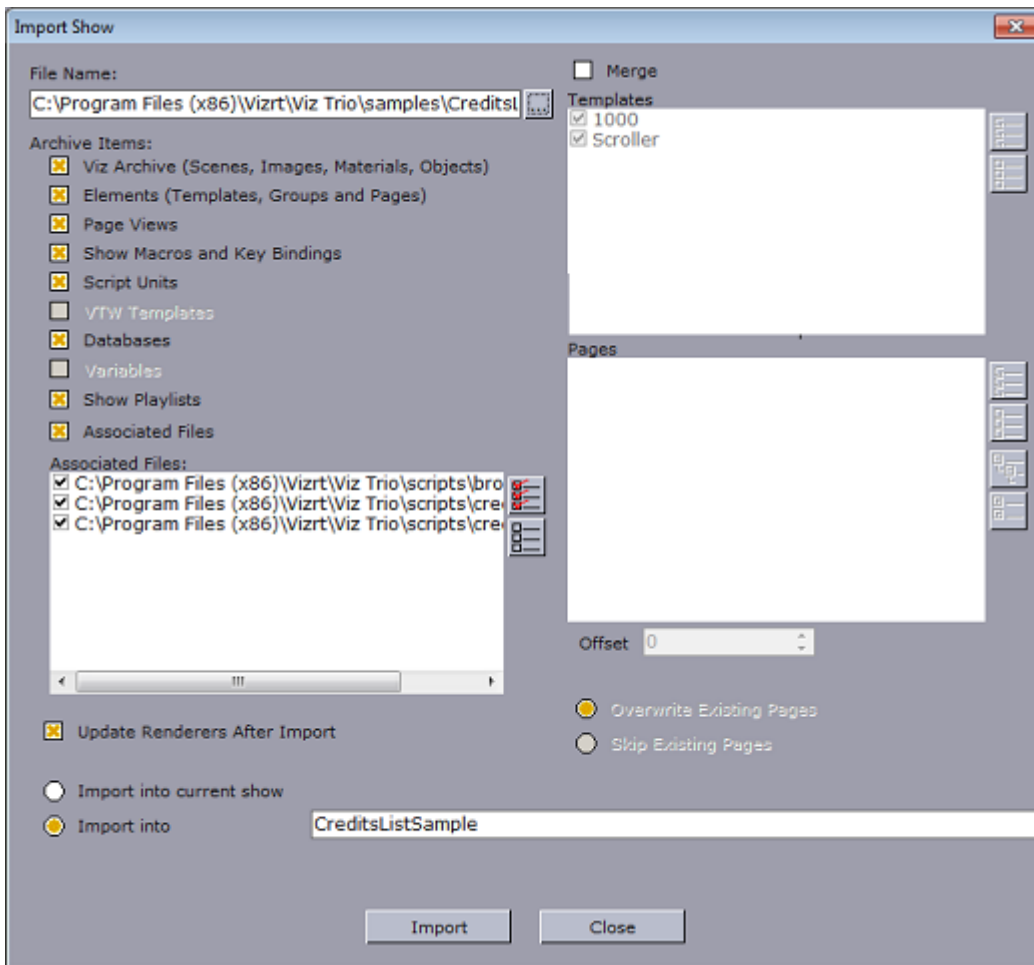
- **Rename Show:** Renames an existing show.
- **Create Show:** Creates a new show.
- **Delete Show:** Deletes an existing show.
- **Export Show:** Exports an existing show. Selecting this option opens the [Export Show](#) window.

⚠ **Note:** It's possible to **move a show** to a different Show folder using 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.



When importing a non context-enabled show that was created in a previous Viz Trio version, a dialog will appear during import, asking if the show should be converted into a context-enabled show type. When confirming this question, the show will automatically be converted to context-enabled upon import.

- **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.
- **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.

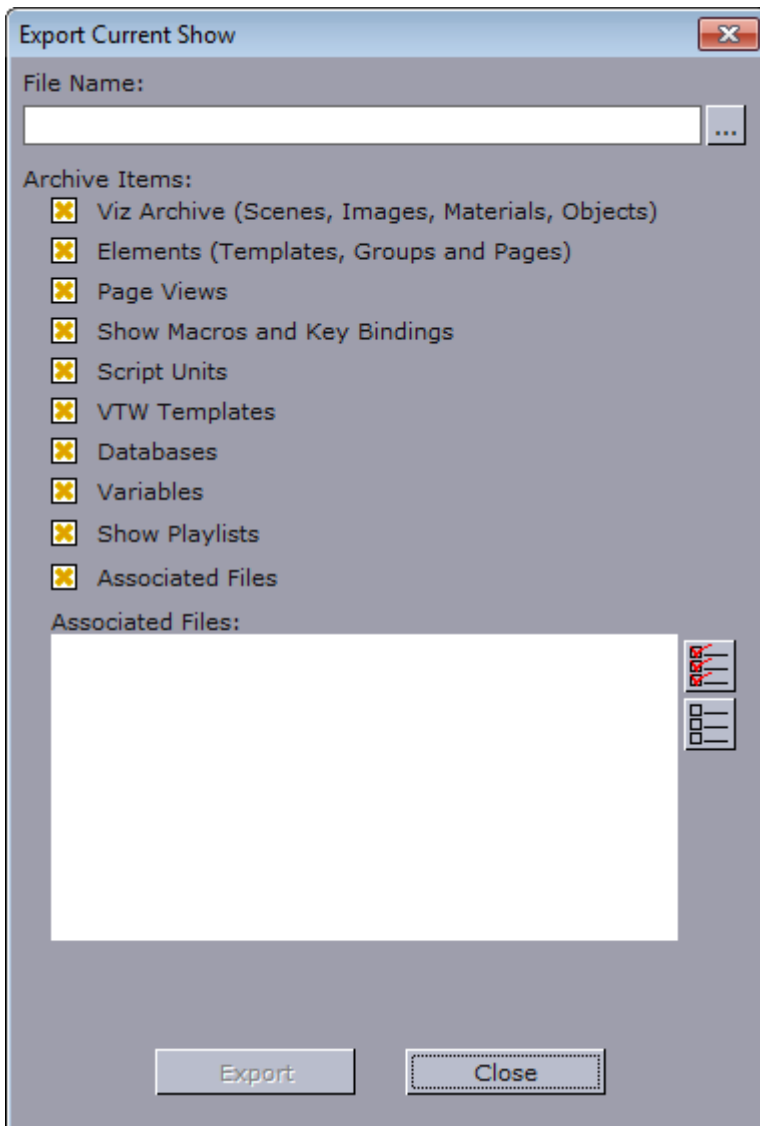
- **Offset:** Sets the numeric value used for pages that are imported with offset values. If a page has callup code 1000, and the offset is 100, the new code will be 1100. Note that only pages can be merged with an offset.
- **Overwrite Existing Pages:** If the callup codes already exists in the show, 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 is to import all elements present in the show archive.

Export Show

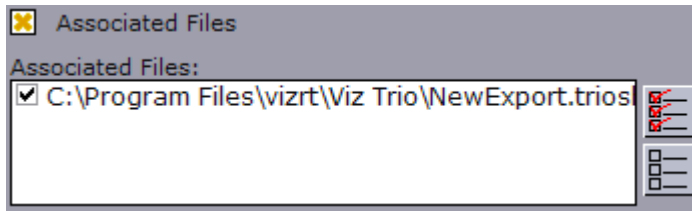


Clicking the **Export show** button opens the Export Current Show dialog.



- **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.
- **Show Macros and Key Bindings:** Includes the show's macro and key shortcut specifications (see the [Show Properties](#) 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 Properties](#) 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 Pilot and newsroom playlists.

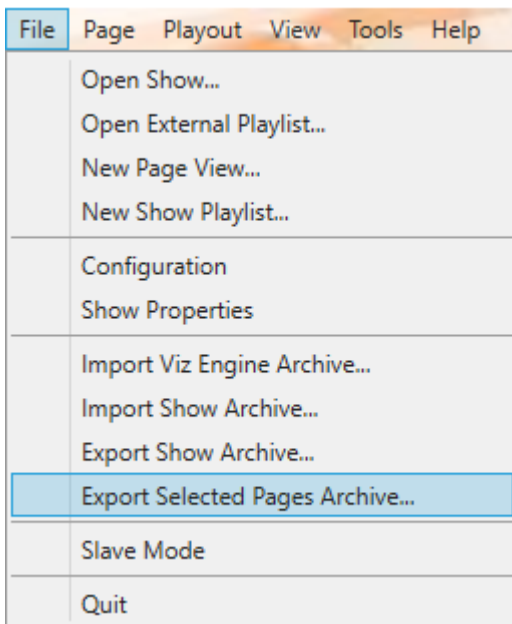


- **Associated Files:** Includes the show's associated files (see the [Show Properties](#) section).
 - **Select all (button):** Selects all associated files.
 - **Deselect all (button):** Deselects all associated files.

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

Export Selected Pages Archive

To access this feature click on **File > Export Selected Pages Archive...**. This process can also be triggered by using macro command `gui:show_export_selected_page_dialog`.



This menu is functioning only when there are pages selected in the trio page list. Clicking on this menu will trigger the **Export Show** dialog to export selected pages. It is almost the same as the [Export Show](#) feature, but it will export only selected pages and all its associated items.

Creating 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**.

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

Importing 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 only has 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.

Exporting 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 being exported must be loaded in Viz Trio.

Playlists

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

Show Directories						
Shows			Playlists			
Name	Start	Durati... △	Host	Status	Active	
[-] MOS Playlists						
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"/>	

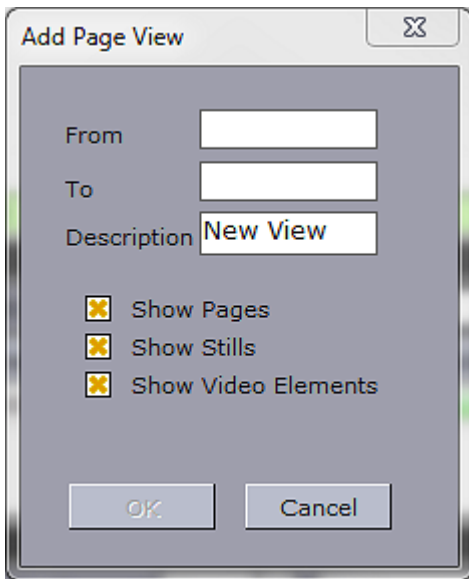
- **Name:** Shows the names of the playlists provided by the newsroom control system.
- **Start:** Specifies the start time of the various playlists.
- **Duration:** Shows the duration of each playlist.
- **Host:** Indicates the host of a determined playlist.
- **Channel ID:** Shows the ID of the channel 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 [Import and Export Settings](#) setting that can be disabled to allow editing of MOS playlists.

5.4.2 Add Page List View



Clicking the **Create page 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.

5.4.3 Creating Playlists



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

Creating a Show Playlist




1. Click the **Create Playlist** button, and in the dialog that opens, enter a playlist description.
2. Drag pages from the current show to the Show Playlist.

5.4.4 Show Properties

Show name: VizTrio-Feed
 Description: My show description here
 Default viz scene path: ...
 Library Path: ...
 Timing handler: ▾
 Preview background image: ...
 Active VTW template: ▾ ...
 Show Script: ...
 Transition Path: transitions ...
 Default Effects: ...
 Databases: ...
 Associated Files: Add File(s)
 Associated Viz Folders (Not recursive): Add Folder
 Customize colors by doubleclicking the color boxes.
 OK Cancel

- **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 section).
- **Preview background image:** Sets the image path for the preview background image. For more information about preview modes, see [Preview modes RGB, Key and RGB with 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. The resulting VTW window created can also be

un-docked (or floated) from the main Trio window by clicking the pin-button: . When the VTW window is floated, clicking the pin-button again will re-attach the VTW window to the main Trio window.

Note: For information on how to export Viz Pilot templates, see the Template Wizard section in the [Viz Pilot User 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.
- **Default Effects (ellipse button):** Opens the **Default Effects** window where you can select *Default Video/Still Effect* and *_Default Scene Effect _* from a dropdown. If no effects are selected, default effect settings will be used, which will be sufficient for most usages.
- **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).
- **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 Properties.
 - **Delete File(s):** Removes the selected file from the current Show Properties.
 - **Delete All:** Removes all associated files from the current Show Properties.
- **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 Properties.
 - **Delete Folder:** Removes the selected folder from the current Show Properties.
 - **Delete All:** Removes all associated folders from the current Show Properties.
- **Customize colors by double-clicking the color boxes:** Double-click on the color boxes to customize show specific character colors (see also how [To edit the text color](#)).

This section covers the following topics:

- [Opening Show Properties](#)
- [Setting the Transition Effects Path](#)
- [Creating a New Database Connection](#)

- [Editing a Connection](#)
- [Removing a Connection](#)

Opening Show Properties



- Click the **Show Properties** button.

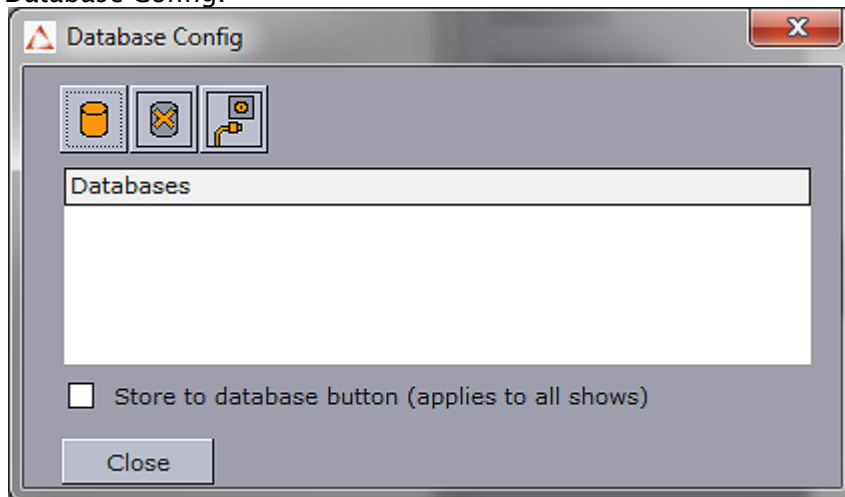
Setting the Transition Effects Path

1. Click the **Show Properties** button to open the [Show Properties](#) 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 Properties window.

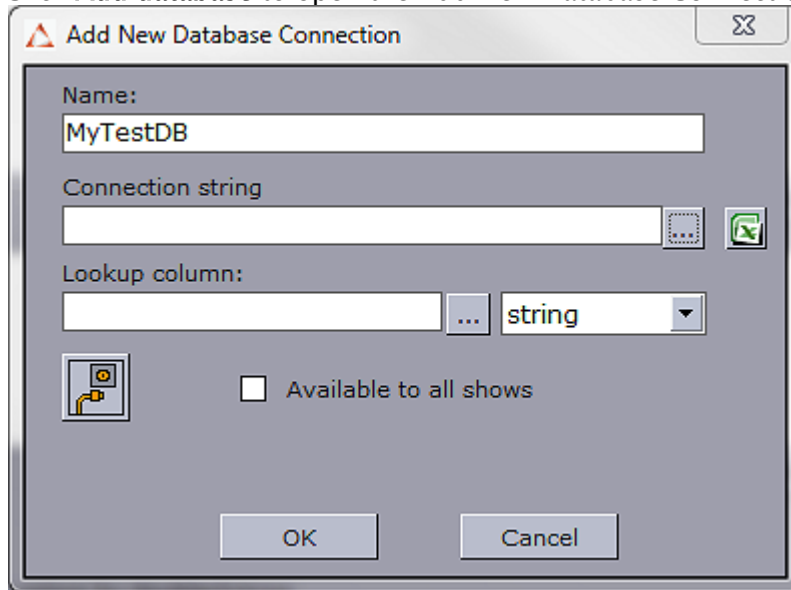
Note: The effects folder should ideally be placed according to the Viz Pilot preferences which is at the root of the Viz database.

Creating a New Database Connection

1. Click the Databases settings' ellipse (...) button in the **Show Properties** section to open the Database Config.

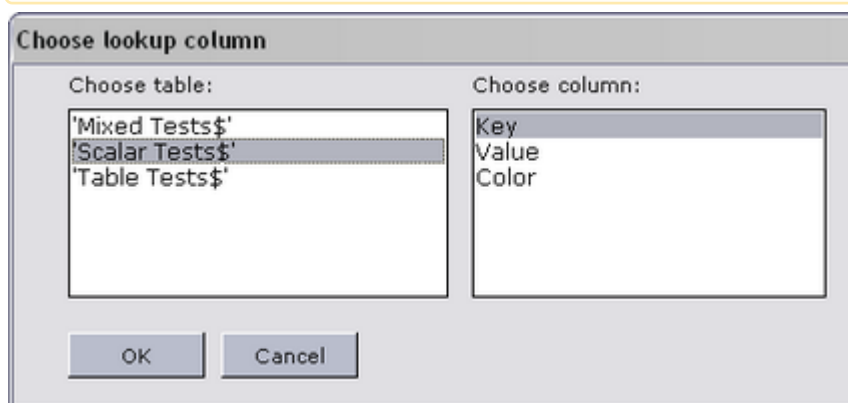


- Click **Add database** to open the Add New Database Connection window.



- Enter a **Name** for the connection and select one of the following options:
- Option 1: OLE DB*
 - Click the **Connection string**'s ellipse (...) button to configure the Data Link Properties
 - Select an **OLE DB** provider, and click **Next**.
 - Add the connection parameters.
 - Click the **Test Connection** button.
- 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.
- Click the **Test connection** button.
- Click the **Lookup column**'s ellipse (...) button to **choose lookup column**.

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



- 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.

9. Specify whether 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.
10. Click **OK** to save the database connection.

Editing a Connection

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

Removing a Connection



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

5.4.5 Cleanup Channels



This function clears all loaded graphics from memory for the page list and all the playlists 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.

5.4.6 Initializing 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.

Each playlist and pagelist elements have its own initialize indicator. The Loaded column will for each element indicate status with percentage and a color-code; Yellow: loading and Green: loaded. A progress indicator icon will indicate the load for all elements in the pagelist or playlist with a tooltip indicating in percent total load.

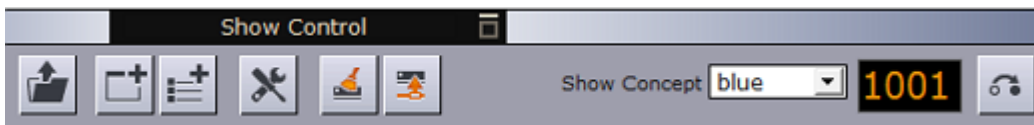


Several shows can be initialized if necessary. After having initialized the first, another show can be switched to and initialized. The graphics for the first one will still be there, ready for use.

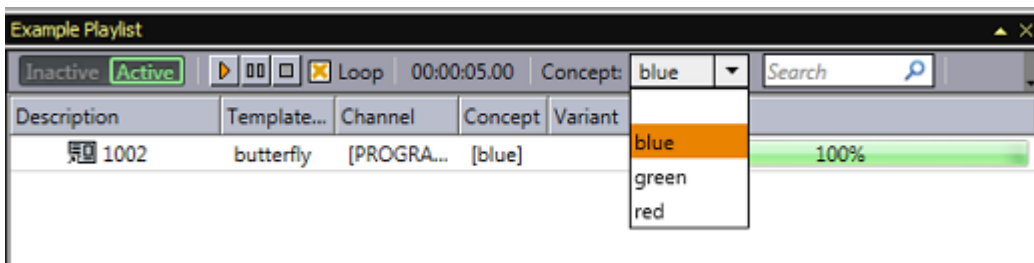
When initializing shows that require a lot of memory, please take the memory use on the program and preview renderers into consideration when loading the graphics. Too much graphics on the renderer(s) will use up all physical memory, causing the performance to drop below real-time, which in turn may 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.

5.4.7 Show Concept



A concept can be switched directly from the [Show Control](#). 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. If none of the available concepts is chosen one menu option will be blank.

5.4.8 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](#).

Reading a Page

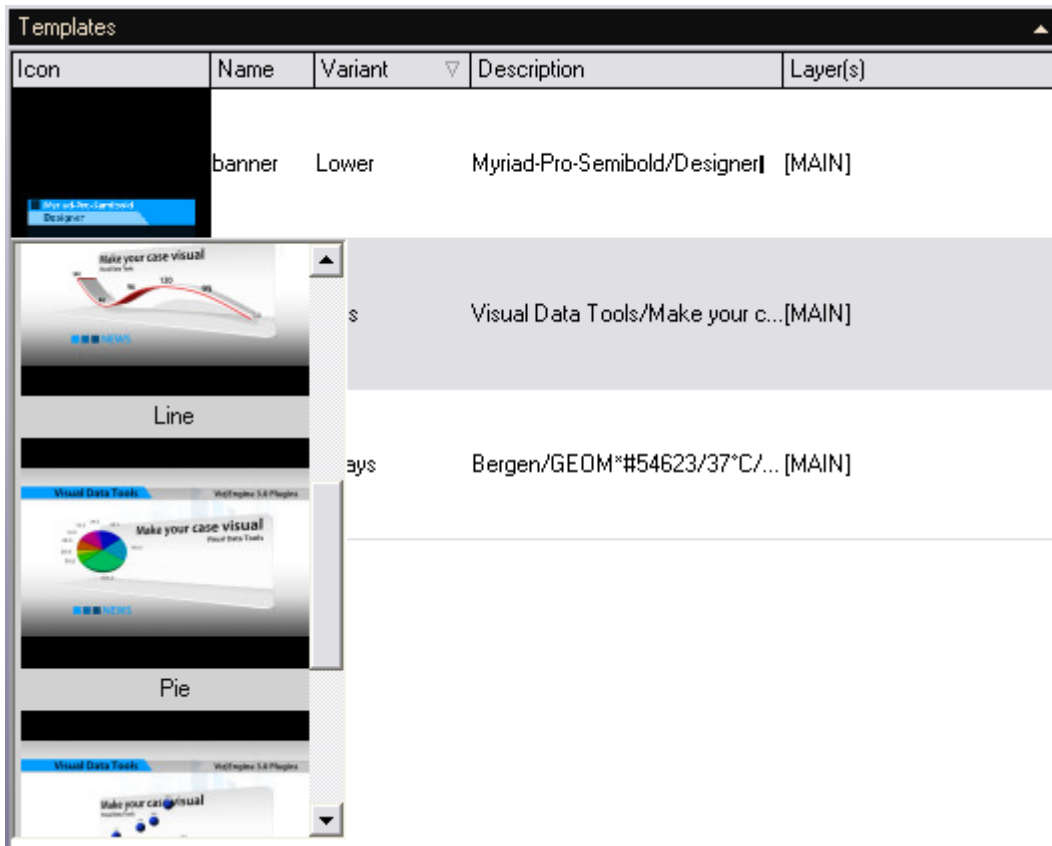
- Enter the page's **callup code** (number) using the keyboard's **numeric pad**, and press the **plus (+)** key or the Read Page button to read the page.

See Also

- [Playlist Modes](#)
 - [Show, Context and Tab-field Commands](#)
 - [Cleanup Channels](#)
 - [Macro Commands and Events](#)
 - [Initializing Channels](#)
 - [Page Editor section on Database Linking](#)
 - [Keyboard Shortcuts and Macros](#)
 - [Macro Language](#)
 - [Scripting](#)
-

5.5 Template List

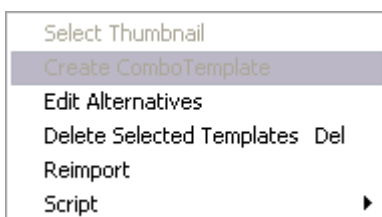
Unlike the traditional Page List, The Template List only contains templates. A separate Page List contains the pages derived from the templates. Template variants can be switched directly from the Template List, or from the [Page Editor](#). As with concepts, variants can also be switched using a macro command in a user-defined macro or script.



This section contains information on the following topics:

- [Template Context Menu](#)
- [Columns](#)
- [Combination Templates](#)
- [Creating a Combination Template](#)

5.5.1 Template Context Menu



The Template List contains a context menu for creating combinations of templates, assigning scripts and so on.

- **Select Thumbnail:** Use this option to select which thumbnail to show for the template.
- **Create ComboTemplate:** Opens the Combo Template Editor, from where it is possible to create a combination template that contains all selected templates. This is only relevant for transition logic templates. All templates that are to be merged into [Combination](#)

Templates 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 associated with a template, such as concept, variant, user-defined contexts (platform) and scene path.
- **Delete Selected Templates:** Deletes single or multiple templates from the Template List.
- **Reimport:** Updates any scenes that have been changed.
- **Script:** Includes all script-related options:
 - **Edit Show Script:** Opens the **Choose Show Script** window, where the script assigned to the show can be edited.

✓ **Tip:** The **Choose Show Script** window can also be opened by selecting **Edit Show Script** from the **Page Editor** drop-down menu, or by clicking the **Show Script** browse button in the **Show Properties** window. See also [To assign a script to a show](#).

- **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. This menu also includes a list of the names of the scripts already assigned to the selected template.
- **Media Sequencer:** Selects scripts placed on the Media Sequencer.
- **Clear:** Clears the assigned script.

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

5.5.2 Columns

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

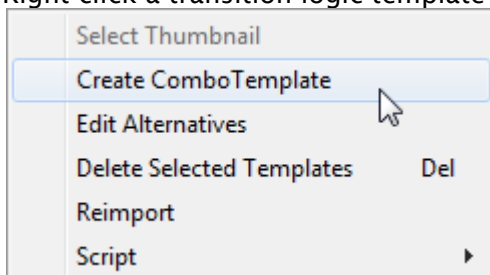
- **Description:** Shows the contents of the template tab fields.
- **Icon:** Shows the thumbnail image of the scene.
- **Layer(s):** Shows which layer, [FRONT], [MAIN], [BACK], the scene belongs to.
- **Name:** Shows the name of the template.
- **Variant:** Shows the variants of the template. For example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

5.5.3 Combination Templates

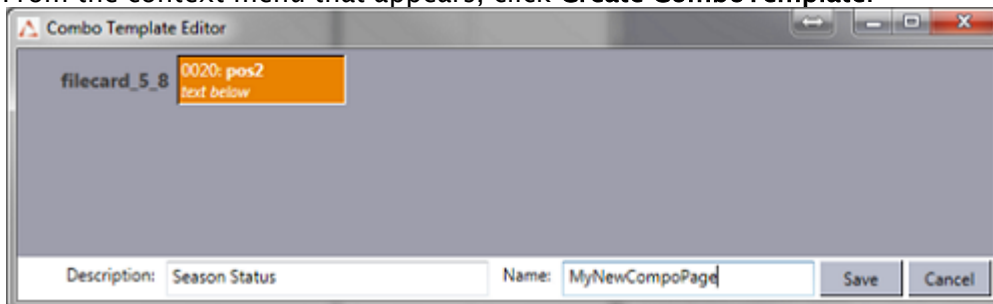
A combination template can only be created using transition logic templates. In order to create a combination template, the templates must be in different layers (see **Layer** information in the **Template List** or **Page List** column).

5.5.4 Creating a Combination Template

1. Right-click a transition logic template in the Template List.



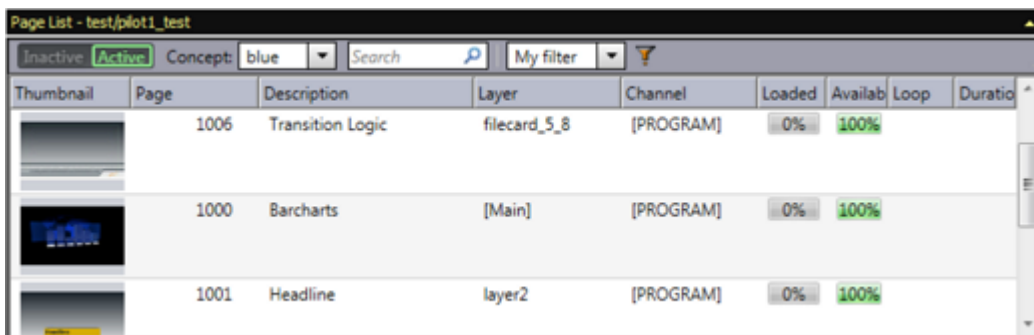
2. From the context menu that appears, click **Create ComboTemplate**.



3. In the **Combo Template Editor** that opens:
 - Select the layers/states that should be activated for this combo template (only one state per layer can be selected).
 - Enter a **Description**.
 - Enter a **Name** (no spaces are permitted in the combo template name).
 - Click **Save**.

5.6 Page List

The Page List contains a list of accessible pages. Pages are created from templates. Templates are, in turn, normally created by importing scenes created with Viz Artist. You can think of a template as a blueprint while pages are instances or real objects created from templates. You can create as many pages as you like from one template, and each of the pages you create is unique with its own page ID.



Thumbnail	Page	Description	Layer	Channel	Loaded	Availab	Loop	Duratio
	1006	Transition Logic	filecard_5_8	[PROGRAM]	0%	100%		
	1000	Barcharts	[Main]	[PROGRAM]	0%	100%		
	1001	Headline	layer2	[PROGRAM]	0%	100%		

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 Properties](#) window or by using more advanced features such as [Macro Commands and Events](#).

This chapter contains the following sections:


- [Page Content Filling](#)
- [Page list Context Menu](#)
- [Page list Columns](#)
- [Template and group](#)
- [Arm and Fire](#)

5.6.1 Page Content Filling

Filling a page with content is a central operation in Viz Trio. In order to create a page, the user must first create a show and import scenes to that show. A template is then converted into a page, and once the page is created it can be populated with media from Viz One.

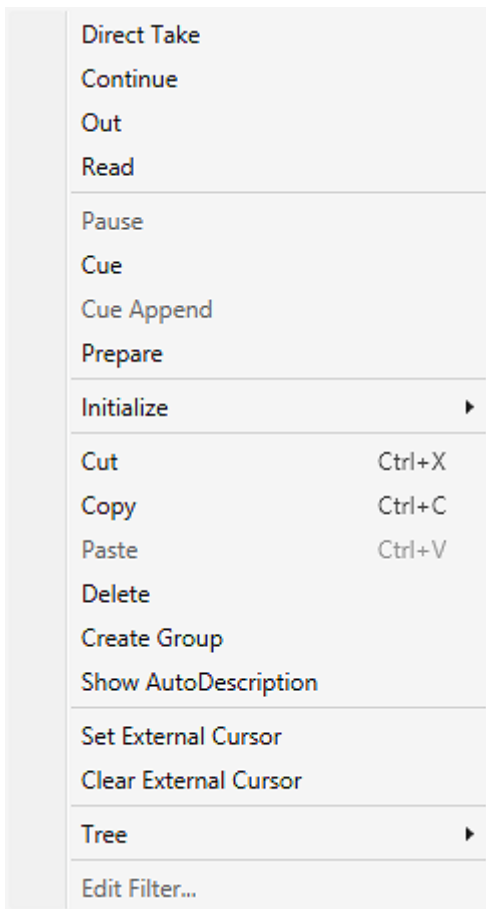
To create a page and fill it with content, follow this procedure:

- Create a show under Show Directories from **File > Open Show**. Alternatively, access the Show Directories menu by clicking the Change Show icon.
- From here, you can import a show. If there is no show to import, click the **Create show** icon
- A field opens in the Shows panel. Give the new show a title and click **OK**.
- From the drop down menu in the Page Editor, select Import Scenes. Select the desired scenes from the Graphic Hub and import them by clicking **Import** at the bottom of the panel.
- The scene now appears in the Template List
- Now save the template as a page element. Double click the desired template in the Template List.
- In the Page Editor, click **Save As** (the icon with two floppy disks)
- The page number title field is highlighted. Accept the title or edit it and press **Enter**. The new page now appears in the Page list. Pages can be searched for and filtered by page number from the Page list. From here, fill up the page with, for example, images and video by dragging and dropping media from Viz One into the Page Editor. It's also possible to take the element in a page to air from the Page list, or add pages to different playlists.

 **Note:** Composite elements cannot be displayed from the page list.

5.6.2 Page List Context Menu

Right-clicking an element in the page list activates the context menu. Select a menu item to perform an action or define a setting for a page. Clicking an image icon will open a drop-list of available template variants.



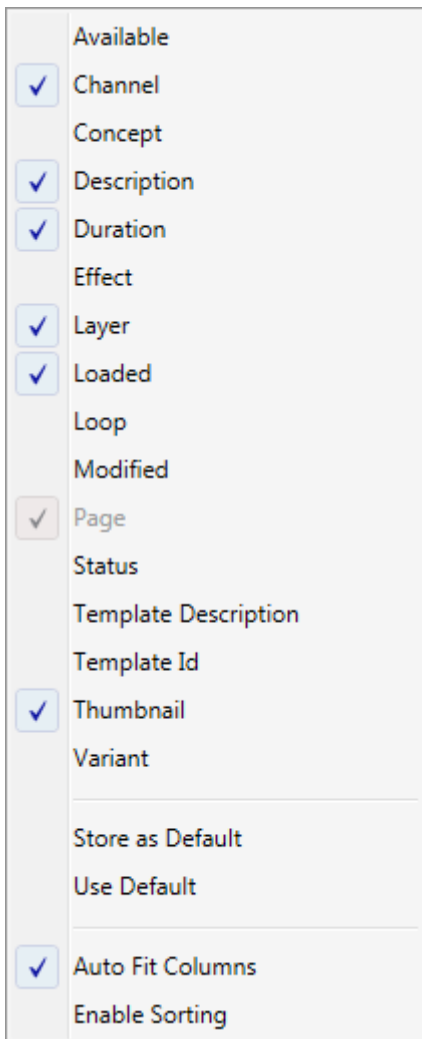
Note: The menu options **Create Combo Page** and **Script** are part of the **Template List's** context menu.

- **Direct Take:** Performs a direct take on the selected page.
- **Continue:** Continues the animation on the selected page, or any that is loaded in the same transition logic layer as the selected page.
- **Out:** Takes out the selected page, or any that is loaded in the same transition logic layer as the selected page.
- **Read:** Performs a read on the selected template or page.
- **Pause:** Pauses the current element.
- **Cue:** Prepares the clip for playout so the *first frame is ready* in the player.
- **Cue Append:** Prepares the clip for playout. The clip will *start automatically* when the current clip ends.
- **Prepare:** Prepares the clip in the pending player without affecting the current clip, so it is ready to be played out. The clip will *not start automatically* after the current clip. Requires Viz Engine 3.3 or above.
- **Initialize:** Loads the selected page(s) on the program and preview renderer.
- **Cut:** Cuts an element to the pasteboard (shortcut: *Ctrl+x*)
- **Copy:** Copies an element to the pasteboard (shortcut: *Ctrl+c*)
- **Paste:** Pastes in the current pasteboard element (shortcut: *Ctrl+v*)

- **Delete:** Deletes the current selected element.
- **Create Group:** Creates a new group.
- **Show AutoDescription:** Switch. Shows the Auto Description for item.
- **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.
- **Tree:** The Tree option contains sub-menus for selecting, expanding, collapsing, and moving data elements.
 - **Select All:** Selects all data elements in the playlist.
 - **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. This setting only affects the current playlist. In order to make this setting global for all playlists, enable the **Store as Default** option in the playlist column context menu.
 - **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.
 - **Show Row Lines:** Switch. Select to show Row Lines.
 - **Show Alternating Row Colors:** Switch to show alternating row colors.
- **Edit Filter:** If using a filter, select this to display the **Edit Filter** menu window where you can edit a filter.

5.6.3 Page List Columns

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



Choose the columns to add or remove.

- **Available:** Shows the availability of the element as a percentage.
- **Channel:** Selects the playout channel on which the graphics will be rendered.
- **Concept:** Shows which concepts are associated with the data element.
- **Description:** Shows by default the text entered in the editable tab fields (or the element name, if the element is a stand-alone media file). Can be edited inline. Only a page's Description field can be changed.
- **Duration:** Shows the duration of the element.
- **Effect:** Opens the Choose Effect dialog when clicking the browse button.


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

- **Layer:** Shows the layer information. This can be [FRONT], [MAIN] and [BACK].
- **Loaded:** Shows (in percent) load status of the element
- **Loop:** Loop enables a loop mode for 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) 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.

Looping in a page list is aligned with the looping in a timeline editor. However, when “Reload the current page if it is changed in the MSE” is disabled from **Configuration>>General** then looping between page list and timeline editor is asynchronous.

- **Modified:** Displays the time when the element (graphic, still image, or video) was modified/created. This feature is mostly used for sorting, and note that sorting must be enabled in the page list context menu in order for this to work.
- **Page:** Shows the page ID that usually is a numeric callup code. Can be edited inline.
- **Status:** Shows the status of the selected page. Page status indicators are Finished, Unfinished and None. **None** means: no certain page status indicated. Trio handles these pages regularly. **Finished** means: the page content is edited to completion and page should not be changed. Trio prohibits editing pages with this status in the page editor. **Unfinished** means the page needs further editing before playout. This is a simple reminder to the journalist to finish editing the page before playout. Also see the macro [commands](#) page in the [Macro Commands and Events](#) section.
- **Template Description:** Shows the template’s description; usually what kind of template it is, for example lower third, bug and full screen. Descriptions are predefined by the scene designer.
- **Template id:** A page references a template, thus showing the template’s ID. The ID is usually a numeric callup code.

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

- **Thumbnail:** Shows a thumbnail image of the graphics.
- **Variant:** Shows the template variant text. To view a list of variants with thumbnails, left-click the text field.
- **Store as Default:** Stores the current selected column values as Defaults.
- **Use Default:** Uses the stored column Defaults.
- **Auto Fit Columns:** sizes columns automatically.
- **Enable Sorting:** disabled by default. The sort feature is specific per playlist, so if sorting is enabled on one playlist or show, it can still be disabled for another Playlist or show.

5.6.4 Templates and Groups

This section covers the following topics:

- [Change Template](#)
- [Create Groups](#)
- [Time Codes](#)


Change Template

For standalone scene templates, it's possible to [change a template](#) that a page is using without having to retype its content. This works 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 as well. This also applies 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 similar in their logical structure.
- The total number of tab-fields should be the same:
 - If the source template has three bullet points (exposed as three tab-fields) and the target has four, for instance, 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 only has three, the last bullet will not be shown. If no new text has been saved or added, it is possible to change back and the original template will be restored.

Changing a Template


1. Select the pages for which a template is to be changed (use CTRL or SHIFT to multi-select).
2. Right-click the page(s), and select **Change template...** from the context menu.
3. Enter the new template name/number.

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

4. Click **OK**.

Changing 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**.

Creating Groups

It's possible to create groups in a page list to organize pages. To create a group, right click in the page list and select **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.

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

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).

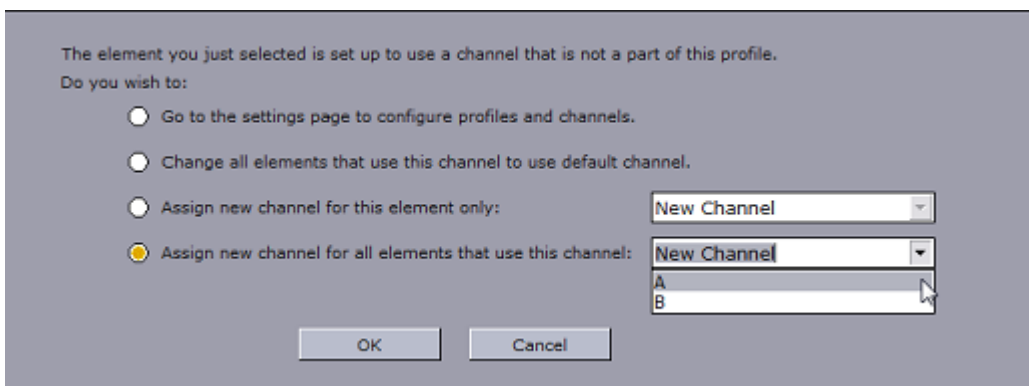
Copying and Pasting Groups

It's possible to copy groups using the copy (**CTRL + C**) and paste (**CTRL + V**) commands in the same show. The pages within the pasted group are given new unique names.

Manually Sorting Pages

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

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 been deleted, for example.

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

Time Codes

A time code reader is a handler that reads time-codes from a given 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 will utilize specific hardware for decoding time-code signals embedded in video signals.

Viz Trio supports the use of time code reader hardware. A card named PCL PCI D from Alpermann+Velte Electronic Engineering GmbH is currently 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.

5.6.5 Arm and Fire


With the Arm and Fire feature it is possible to play out two or more different elements on two different channels simultaneously. By using the commands `page:arm` and `page:arm_current` several elements can be cued up on their respective channel before firing them: channel A, C and D - take. The commands `page:fire` and `page:fire_all` execute actions (take, continue, out, and so on) on all relevant channels that are armed. `page:unarm`, `page:unarm_all` and `page:unarm_current` clears channels from being executed.

This section describes the following related functionalities:

- [Reading a page](#)
- [Taking Pages On-air](#)
- [Deleting a Page](#)
- [Saving to XML](#)
- [Loading from XML](#)
- [Selecting a Variant from the Page List](#)
- [Adding Transition Effects Using the Page List](#)

Reading 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](#).

Taking 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.

Deleting 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.

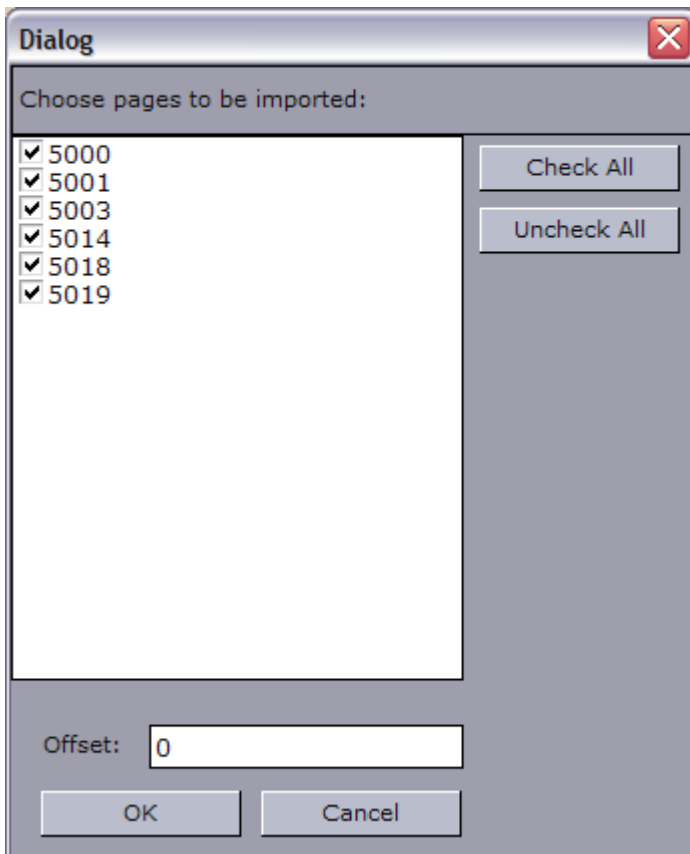
 **Note:** You can multi-select pages by holding in the SHIFT or CTRL button and selecting the range or individual pages to be deleted.

Saving to XML

To save a page to an XML-file on disk, use **File > Export Selected Pages Archive...**

For more information about this option, see [Export Selected Pages Profile](#).

Loading from XML



1. From the main menu click **Page > 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.

Selecting a Variant from the Page List

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

Adding 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** (the latter is for visual reference only). This displays the Effect and Effect Icon columns.
2. Click the **ellipse** button in the **Effect** column and from the appearing dialog box select a transition effect.

See Also

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

5.7 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 Pilot [Playlists](#) are requested by the Viz Trio operator and modified when integrated with a Viz Pilot system. Note that a Viz Pilot playlist element cannot be edited, only the structure of the playlist. The Viz Pilot 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. In most cases, Viz Trio is therefore only used to read and play out elements when monitoring newsroom playlists. For the playout of newsroom playlists to work, the Media Sequencer must establish a connection to Viz Gateway (see [MOS](#) integration).

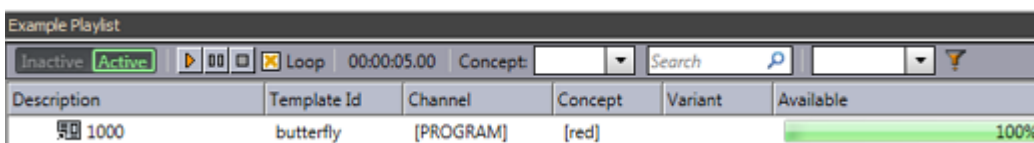
All shows and playlists are stored in the Media Sequencer for playout.

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

This section covers the following topics:

- [Activating and Deactivating a Playlist](#)
- [Taking Pages On-air from a Show Playlist](#)
- [Showing and Hiding Playlist Columns](#)

5.7.1 Activating and Deactivating a Playlist



Description	Template Id	Channel	Concept	Variant	Available
1000	butterfly	[PROGRAM]	[red]		100%

1. [Create Playlist](#) or open [Playlists](#)

2. Select Active or Inactive in the menu bar for the currently selected playlist to activate or deactivate the playlist. Activating a playlist will initialize the elements and start the transfer of video clips to Viz Engine. The duration column shows the duration of a video. The duration column will initially be blank for graphic elements or still elements.

5.7.2 Taking Pages On-air from a Show Playlist

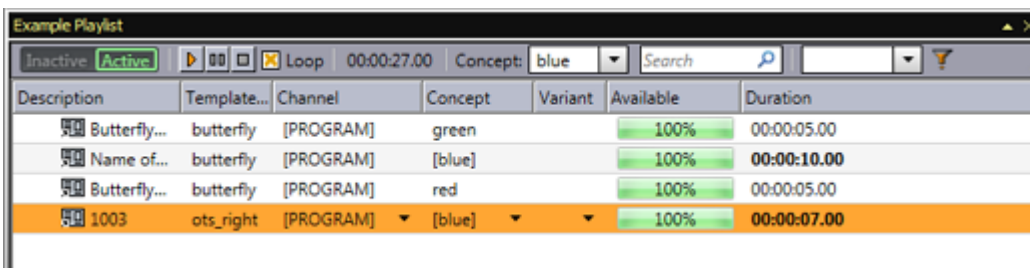
1. Add pages to the playlist.
 - *Optional:* Enable looping.
 - *Optional:* Set the concept for the playlist or for each element.
 - *Optional:* Set the variant for each element.
 - *Optional:* Set the duration for each element.
2. Click the **Play** button to start the carousel.
3. Click the **Pause**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

5.7.3 Showing and Hiding Playlist Columns

- Right-click the playlist header and select columns to display or hide from the menu.
 - The **Auto width** option enables automatic resize of columns within the MOS Playlist area.
 - Disable the **Auto width** option to manually resize the columns.

Playlist



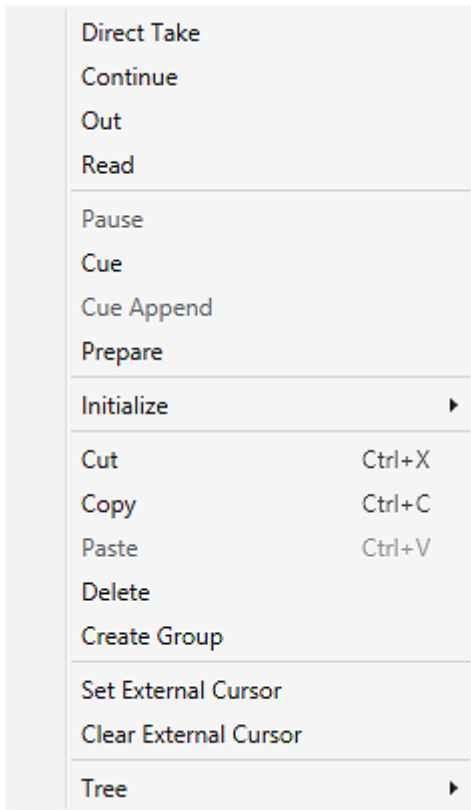
Description	Template...	Channel	Concept	Variant	Available	Duration
Butterfly...	butterfly	[PROGRAM]	green		100%	00:00:05.00
Name of...	butterfly	[PROGRAM]	[blue]		100%	00:00:10.00
Butterfly...	butterfly	[PROGRAM]	red		100%	00:00:05.00
1003	ots_right	[PROGRAM]	[blue]		100%	00:00:07.00

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, nor vice versa.

This section covers the following topics:

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

Playlist Context Menu



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

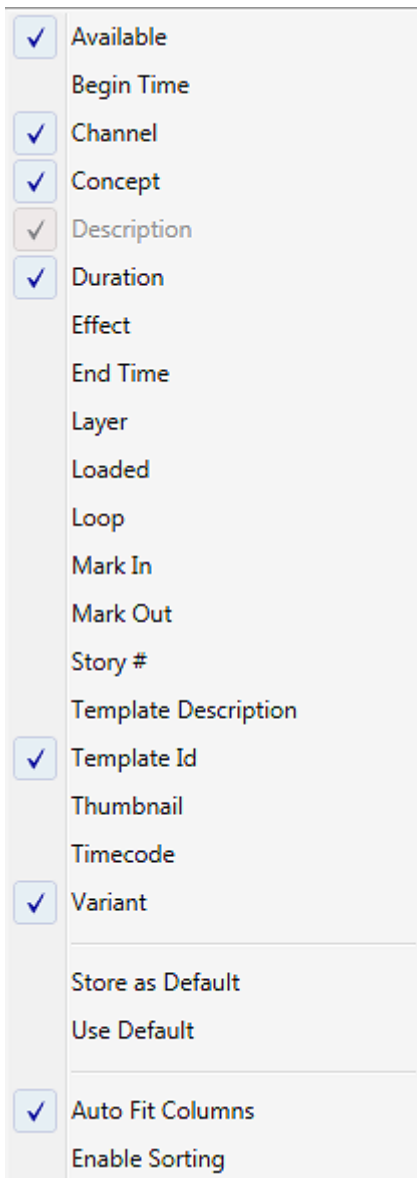
Note: On composite groups containing video, only the following operations can be performed: **Direct Take**, **Read**, and **Out**. On standalone videos, all commands can be performed (*Cue*, *Pause*, *Prepare*, and so on).

- **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.
- **Pause:** Pauses the element.
- **Cue:** Prepares the clip for playout so the *first frame is ready* in the player.
- **Cue Append:** Prepares the clip for playout. The clip will *start automatically* when the current clip ends.
- **Prepare:** Prepares the clip in the pending player without affecting the current clip, so it is ready to be played out. The clip will *not start automatically* after the current clip. Requires Viz Engine 3.3 or above.
- **Initialize:** Initializes the selected playlist or element (loads graphics) on the external output channels. Sub menu items for *Initialize*:
 - **Playlist:** initializes the playlist
 - **Element:** initializes the currently selected element.

- **Cut:** Cuts an element to the pasteboard (shortcut: **Ctrl+x**)
- **Copy:** Copies an element to the pasteboard (shortcut: **Ctrl+c**)
- **Paste:** Pastes in the current pasteboard element (shortcut: **Ctrl+v**)
- **Delete:** Deletes the current selected 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/Playlist](#) and [Cursor](#) sections for further details.
- **Clear External Cursor:** Clears the current global cursor.
- **Tree:** The Tree option contains sub-menus for selecting, expanding, collapsing, and moving data elements in the playlist.
 - **Select All:** Selects all data elements in the playlist.
 - **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. This setting only affects the current playlist. In order to make this setting global for all playlists, enable the **Store as Default** option in the playlist column context menu.
 - **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.
 - **Show Row Lines:** Switch. Select to show Row Lines.
 - **Show Alternating Row Colors:** Switch to show alternating row colors.

Playlist Columns

Right-Click on the playlist columns-bar to select values to display:



- **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*.
- **Begin Time:** Shows the activation start time for a group (format hh:mm:ss).
- **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 for this element only or in a template (that all data elements made from it will have). By creating a group and placing elements within it, all elements in the group will be organized by having the same channel. The Channel column is presented by default. See the [Channel referencing underlying element](#) section for more information about using channels with playlist elements.
- **Concept:** Shows which concept(s) the data element is associated with.

- **Description:** Shows the description of the element. By default this will show the path of the scene (or the element name of a stand-alone media file). May be edited inline in the playlist.
- **Duration:** Shows the length of the element. The duration is often set by Viz Pilot or a newsroom system. However, the individual elements duration can be set in Viz Trio by editing the Duration column directly. The total duration of the elements in the playlist will be shown in the heading for the playlist. The duration of an element is the time before the next element in the list is taken.
- **Effect:** Opens the Choose Effect dialog, which makes it possible to select a transition effect between two pages.
- **End Time:** Shows the activation end time for a group (format hh:mm:ss).
- **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.
- **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.
- **Loop:** Displays a loop information column. Only a playlist or videos can be looped, not groups or individual elements in a playlist.
- **Mark In:** Sets mark in times for video clips.
- **Mark Out:** Sets mark out times for video clips.
- **Story #:** Shows the story number for stories in MOS playlists. This is only supported from the ENPS newsroom system.
- **Template Description:** Shows the template description (e.g. name).
- **Template Id:** Shows the template ID.
- **Thumbnail:** Shows thumbnails of the scenes.
- **Timecode:** The timecode is an offset time on format *hh:mm:ss:ff*. It indicates that an element should be played out relative to the parent group or video. This is used for instance in composite groups with a video and overlay graphics that is played out on a timeline.
- **Variant:** Select a concept's variant from the drop-list (see the Concept column).
- **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.

Channel Referencing Underlying Element

Square brackets in the channel column denote two different scenarios. [PROGRAM] refers to the program channel that is defined in the profile. However, an element with a channel in square brackets in a playlist, for example [A], refers to its underlying element in the pagelist.

Square brackets in a playlist therefore denote a link back to the element in the pagelist. If a channel contains square brackets in a playlist, and the channel is changed in the pagelist, the changed channel in the pagelist will automatically be reflected in the playlist. When a channel designated to a playlist element does not have square brackets, it will play out on that channel regardless of the channel designated to the underlying element.

In this scenario, changing the channel in the pagelist (the top panel) will automatically be reflected in the channel column in the playlist:

Layer	Channel
[Main]	B

Template Id	Channel	Concept	Variant
1000	[B]		

Playing Out an Element from a Playlist and Pagelist

You can designate the channel that an element will play out on either from the pagelist or playlist. Furthermore, it is possible to play out both the underlying element in a pagelist and a playlist element on different channels. You can manage these operations in the channel column.

To play out a playlist element on the same channel as a pagelist element, ensure that the same channel is selected for the pagelist and the playlist. For example, channel B in the pagelist is channel [B] in the playlist. Once a channel appears with square brackets in the playlist it will always sync with the channel in the pagelist. This means that a channel changed in a pagelist will always be automatically reflected in the corresponding item in the playlist.

Alternatively, select different channels for the element in the playlist and its underlying element in the pagelist to play them out on different channels. If a channel is changed in the playlist and not in the pagelist, the channel selection will not be updated in the pagelist.

⚠ Note: The [PROGRAM] channel is designated with the same square brackets that are used to show a reference to an underlying element. Do not confuse the [PROGRAM] channel with the square brackets for an element in the playlist that refers to its underlying element in the pagelist.

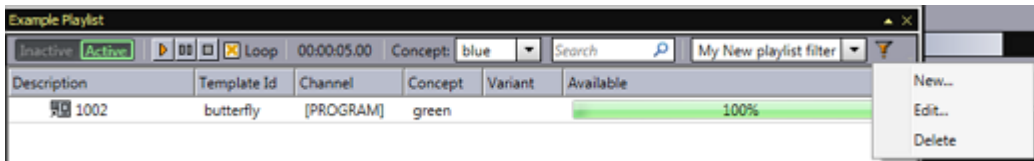
Playlist Filters

You can add a filter to the playlist to narrow down the list of items.

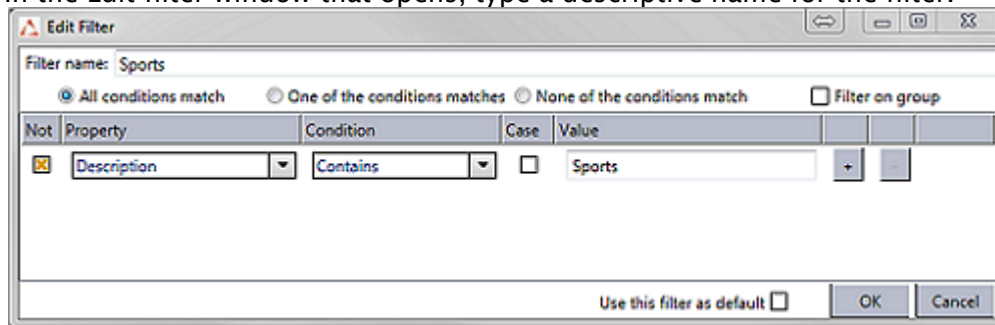
This section covers the following topics:

- [Creating a Playlist Filter](#)
- [Adding a Playlist Filter for Viz Pilot Integration](#)
- [Adding a Default Playlist Filter](#)
- [Adding a Filter to the Playlist](#)

Creating a Playlist Filter



1. From the playlist menu, click the **Manage filter(s)** button.
2. In the menu that opens, click **New**.
3. In the Edit filter window that opens, type a descriptive name for the filter.



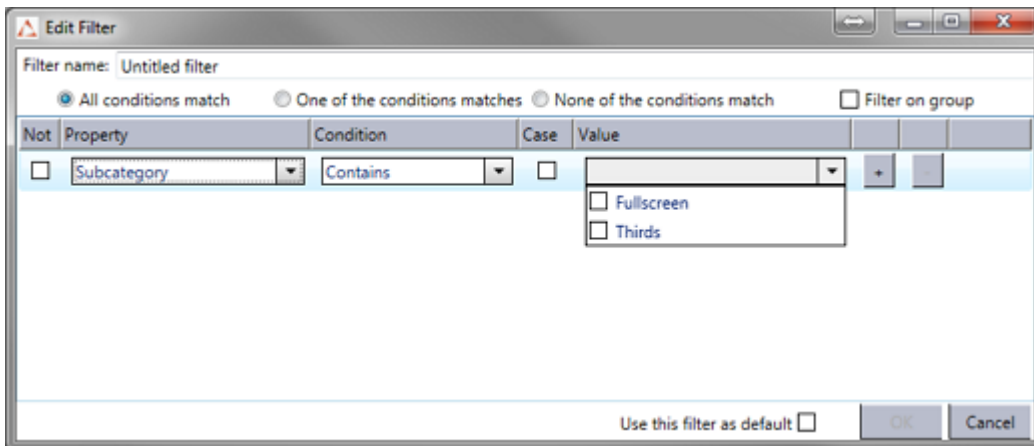
4. Fill in the filter values, for example properties and conditions.
5. Click the **OK** button.
The filter then becomes available in the Playlist filter list.

Note: Starting with Trio 3, composite videos behave like regular videos when filtering on a playlist.

Adding a Playlist Filter for Viz Pilot Integration

Viz Trio can access playlists based on Viz Pilot templates as well as playlists from Newsroom systems via Viz Gateway.

The designer of the Viz Pilot templates can categorize templates by logically dividing them into *Categories* and *Channels*. You can use the Viz Trio filters option to filter playlist views based on these categories created by the Viz Pilot template designer as illustrated in the screenshot below. In this example the playlist can be filtered as **Fullscreen** or **Thirds**.



To use Viz Pilot template categories as filters:

1. Create a new filter or edit an existing filter for a playlist.
2. In the add/edit filter panel set the column **Property** to Subcategory and the **Value** column will be populated with the Categories defined by the Viz Pilot template designer. You can now select values for filtering and save the filter. The playlist will be filtered according to the values selected.

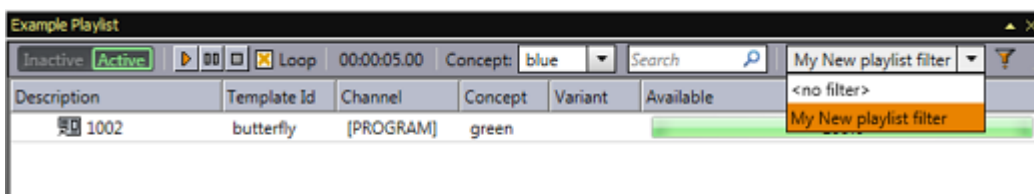
Adding a Default Playlist Filter

1. Create a playlist filter, see [To create a playlist filter](#).
2. Select the **Use this filter as default** check box in the Edit filter window.
3. Click the **OK** button.

When enabling the default filter option, all playlists of the specified Viz Trio client always uses the selected filter. The default filter affects only one Trio client. The default filter is saved and continues to be activated when restarting the Trio client.

These filters are not deleted even if you delete the corresponding page view. However, they can be deleted manually in the filter drop down. If you create two page views with the same filter properties they will share the same filter.

Adding a Filter to the Playlist



- From the Playlist filter list, select a pre-defined filter.

See Also

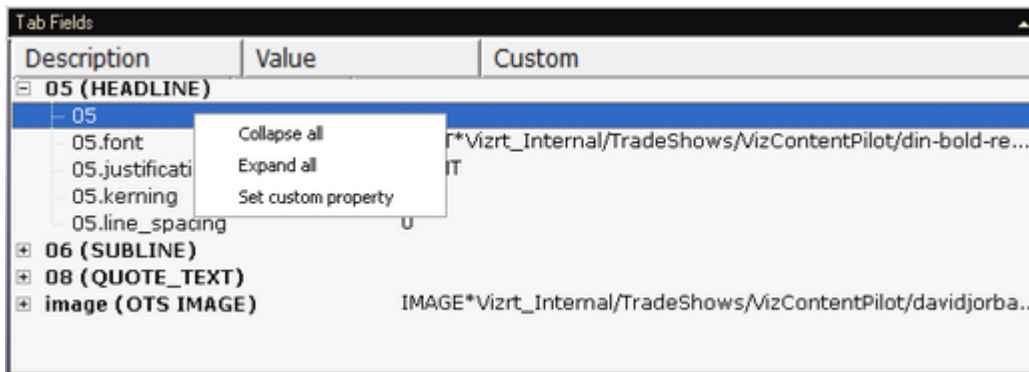
- [Playlists](#)
- [Create Playlist](#)
- [Macro Commands and Events](#)
- [Macro Language](#)
- [Monitoring Playlists](#)

- [MOS configuration](#)
- [Viz Gateway Administrator Guide](#)
- [Viz Pilot User Guide](#)
- [Creating Transition Effects](#)

5.8 Undo And Redo

The global functions *trio:undo* and *trio:redo* are mapped to **CTRL + Z** (undo) and **CTRL + Y** (redo). The undo only applies for deleted playlist entries.

5.9 Tab Fields Window



- The Tab Fields panel 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. For more information on how to use custom properties, see the [Tab field Variables](#) section.

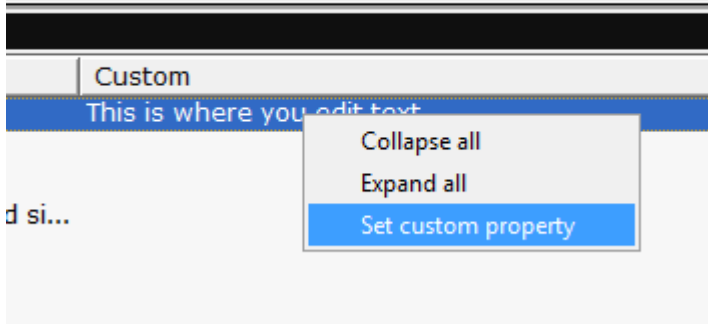
5.9.1 Adding and Editing Custom Values

There are two methods for editing the custom values that correspond to a tab field. However, in order to add custom values, the Custom column must first be activated as follows:

1. Navigate the cursor to the **Tab Fields** panel in the bottom left corner of the main window.
2. Right click in the top menu.
3. Click **Custom**.

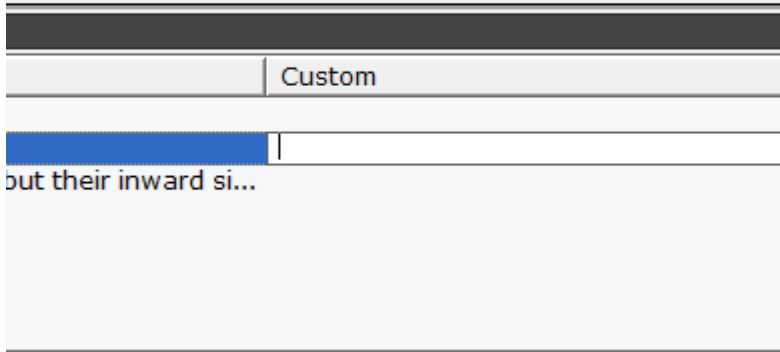
The Custom column now appears in the tab field panel. Now that the custom column is enabled, it is possible to edit or add a tab field's custom value using one of two methods. The first is via the **Set custom property** dialogue box and the second is with inline editing. The two methods are described below.

Set Custom Property Box



1. Right-click the custom value cell and select **Set custom property**.
2. Edit the text in the field that appears.
3. Click **OK** to confirm.

Inline Editing

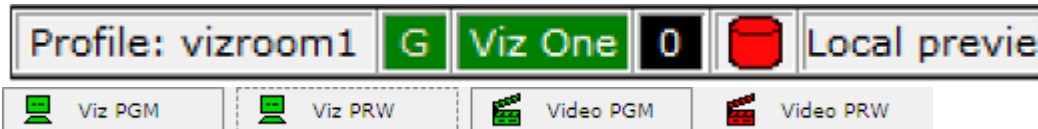


1. Click on the desired tab field, which will highlight it in blue.
2. Click on the custom value cell. A text editor opens, after which you can enter your text.
3. When finished, press **ENTER** and your addition is confirmed.

⚠ Note: Values can be assigned and edited for top-level nodes only. This means that custom values can be added for tab fields but not the tab field properties that appear below them.

5.10 Status Bar

General status information and the Media Sequencer's connection status to devices such as Viz Engine and Viz Gateway are shown in the lower left corner of the interface:



The following icons can be displayed:

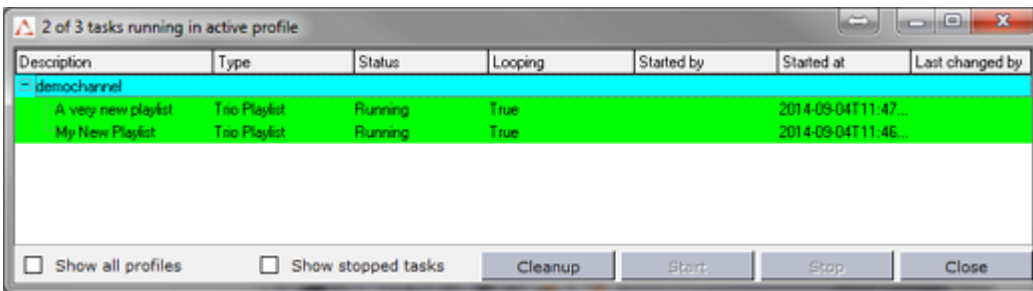
- **Profile:** Shows the currently active profile.
- **G:** Shows the Media Sequencer's connection status to Viz Gateway. The icon indicates that a newsroom system, or several, is available. If no connection is established, the symbol is not visible. A Viz Gateway connection is used to retrieve and control playout of [Create Playlist](#). Double-click the icon to edit the Viz Gateway connection parameters (see the [MOS](#) section).
- **Viz One:** Shows the Media Sequencer's connection status to the MAM system. The icon implies that one or more MAM systems, such as Viz One, are connected. 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 [Proxy](#) connection. The indicator icon is not shown if a MAM system is not configured.
- **Active Tasks (0-n):** Indicates the number of playlist tasks that are active for the current profile(s). For details, see the [Active Tasks](#) section.
- **Database:** The green cylinder shows the Media Sequencer's connection status to the Viz Pilot database. If no connection is established, the symbol is not visible.
- **Messages:** Shows messages to and from the system. For example, error messages and commands sent to the Media Sequencer. Error messages are collected in the Error log. The log can be viewed directly from the status bar by clicking the Errors button (a small red error count label will be displayed on the button if errors occur).
- **Channel:** Displays the channel status for all channels configured for the currently selected profile.

This section covers the following topics:

- [Active Tasks](#)
- [Configuring a Profile](#)
- [Changing a Profile](#)
- [Changing Channel Assignment](#)
- [Checking Memory Usage](#)

5.10.1 Active Tasks

Double-click the Active Tasks icon in the status bar to open the Active Tasks window.

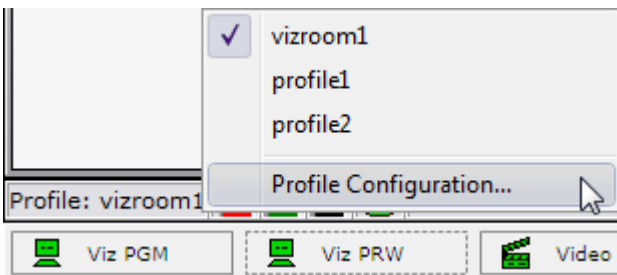


The Active Tasks window displays show-specific playlists that are running in active profiles, showing active tasks running in other profiles as well as different show playlists within a single profile.

The Active Tasks window contains 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.

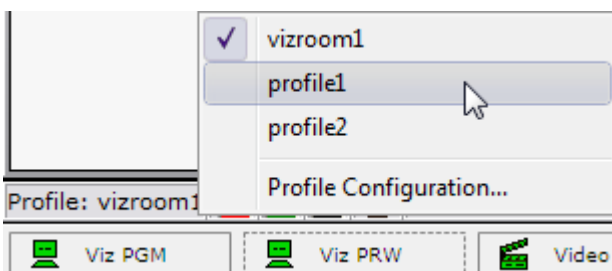
5.10.2 Configuring a Profile



- Right-click the currently active profile (displayed on the status bar) and select Profile Configuration (see [Profile Setups](#)).

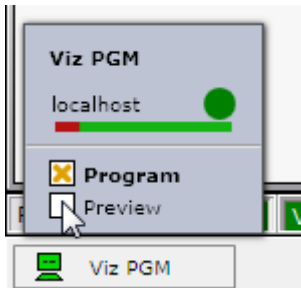
⚠ Note: You cannot change profiles unless you have the control parameter set.

5.10.3 Changing a Profile



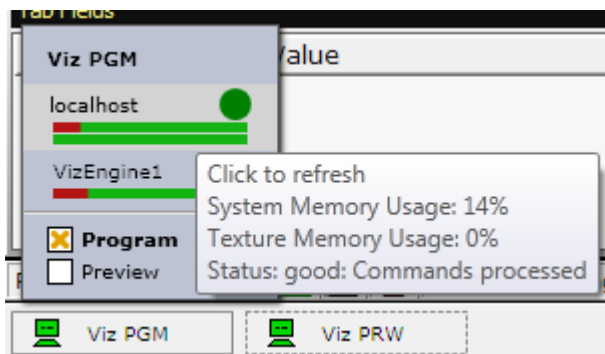
- Right-click the currently active profile (displayed on the status bar) and select the profile you want to use.

5.10.4 Changing Channel Assignment



- Right-click a channel and assign it as a **Program** or **Preview** channel.

5.10.5 Checking Memory Usage



- Right-click a channel and hover your cursor over one of the bars to see the total usage.

See Also

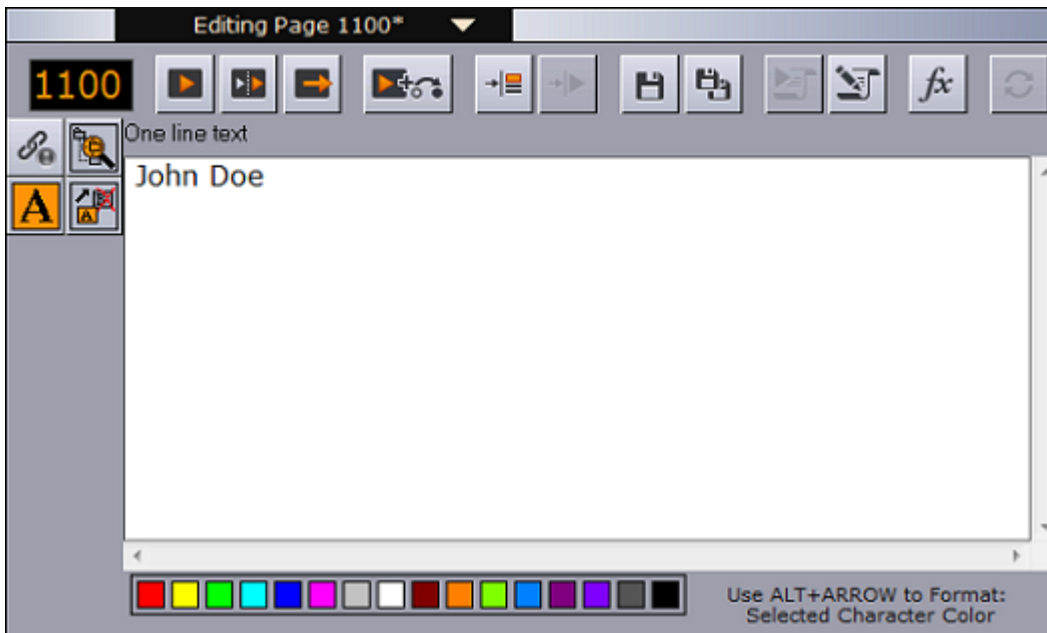
- Configuration Window
- [Playlist Modes](#)

5.11 Page Editor

Depending on the template or page edited, the page editor can display a wide range of specialized editors, including text, database and scroll editors.

When pages have editable elements, it's possible to navigate between elements by pressing **TAB**, or to select elements by clicking them in the local preview window.

The page editor is located in the upper right corner of the UI and is displayed by default when you start Viz Trio.



5.11.1 Edit a Page

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

1. **Double-click** or **Read** a page 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's not necessary to save a page before playing it out on air. However, a template cannot be played out before it has been saved as a page.

This section covers the following topics:

- [Controls](#)
- [Text](#)
- [Database Linking](#)
- [Image Property Editor](#)
- [Transformation Properties](#)
- [Tables](#)
- [Clock](#)
- [Maps](#)

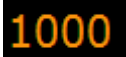
Controls

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

This section covers the following topics:

- [Callup Code](#)
- [Field Linking](#)
- [Playout Buttons](#)
- [Save Buttons](#)
- [Script Buttons](#)
- [Transition Effects](#)
- [Refresh Button](#)
- [Variants](#)
- [Creating a Page](#)
- [Adding Transition Effects Using the Page Editor](#)
- [Applying a Transition Effect](#)
- [Selecting a Variant from the Page Editor](#)

Callup Code

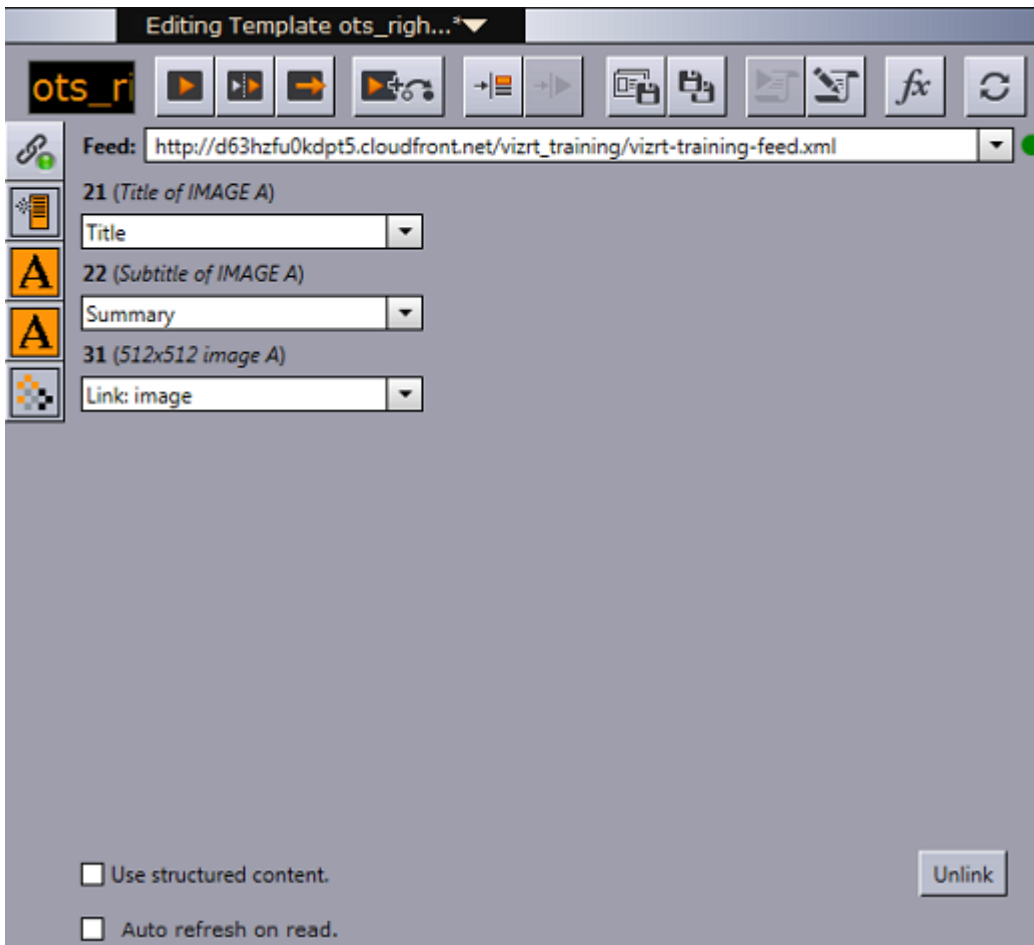
A black rectangular box with the number '1000' written inside in a bold, orange, sans-serif font.

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

Field Linking



When working with templates (not pages), the field linking button is shown in the properties area of the page editor. The button opens a panel, where it is possible to define a feed URI and to define how tab field properties in the template are linked to properties in the external feed.



Feed URIs can link to Twitter, Flickr, and so on. After the feed has been defined, a new property icon (the **select values button**) appears in the page editor:



Clicking the **select values button** displays a panel where you can browse for items from the specified feed, which can be used as tab field values. The list of feed items can be displayed and sorted in various ways, and a filter can be added to narrow down the search.

Options in the Field-Linking Panel

- Check the **Use structured content** checkbox to automatically map tab field values to values in the VDF payload in the selected atom entry. VDF is the Viz Data Format, an ATOM-feed specification. Only check this option if the feed URI is known to be a valid VDF-formatted feed.
- Check the **Auto refresh on read** checkbox to automatically refresh all tab field values when the template is read.

Playback Buttons



Control buttons for page playback are located above the page editor.

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

The **Take** button takes the page that is currently displayed in the local preview On Air, if 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 **Take Out** is clicked when a page that is not on-air is in the preview window, any element that is in the same layer as the page currently displayed in the preview will be taken out.

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

The **Cue** button cues the page specified as parameter or the selected page if the parameter is empty.

The **Cue append** button cues the video after the currently playing video. This button is only enabled for video elements.

The **Pause** button pauses the video. This button is only enabled for video elements.

Save Buttons



When working with a template in the page editor, the **Save template** and **Save as** buttons are available. Clicking **Save template** saves all modifications to the template: feed linking information, feed browser layout, and default tab-field values. Clicking the **Save as** button creates a new page based on the template. The page's callup code is by default set to be the highest callup code (number) in the page list + one (1). The page's callup code is shown in the [Callup Code](#) field. Press **ENTER** to save the new page.

When working with a page in the page editor, the **Save page** and **Save as** buttons are available. Clicking the **Save page** button saves all modifications to the page without any changes to the callup code. Clicking the **Save as** button creates a new page. The page's callup code is by default set as the highest callup code (number) in the page list + one (1). The page's callup code is shown in the callup code field. Press **ENTER** to save the new page.

Script Buttons



The **Execute script** button executes a template's script (see [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 Properties](#) window.

Refresh Button



For pages with linked tab-fields ([Field Linking](#)), the user may want to press this button to refresh the data of all tab-fields in the page. A typical use case is for weather or financial data.

The button is only enabled when at least one tab-field is linked to a feed which provides self links. While refreshing, the icon on the button changes to a cross. Press the button again to cancel the refresh operation.

On a template level, data is automatically refreshed upon read.

The following script commands are related to the refresh procedure:

- `page:refresh_linked_data`: Refreshes the linked data in a blocking way (useful for scripts that manipulate the refreshed data afterwards).
- `page:refresh_linked_data_async`: Refreshes the linked data in a non-blocking way so that the GUI stays responsive.
- `page:cancel_refresh_linked_data`: Cancels a refresh started with `page:refresh_linked_data_async`.


Variants

Template variants can be switched directly from the [Template 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.

Creating a Page

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

Adding Transition Effects Using the Page Editor

1. Open a page in the page editor.
2. Click  and select a transition effect.

Applying a Transition Effect

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

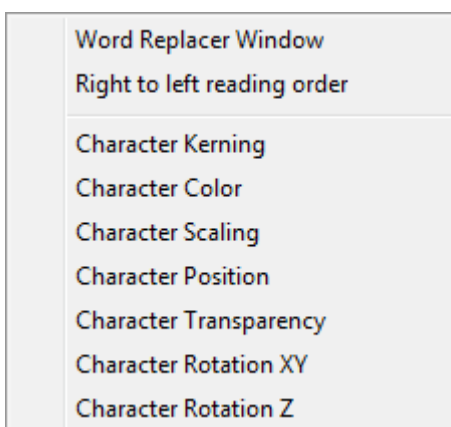
Selecting a Variant From the Page Editor



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

5.11.2 Text

Text Editor Context Menu




The text editor is the main editor for most users. Here, it's possible to do a number of operations if the scene's text editing properties are exposed, such as setting text color or adjusting the position and scale of the text. In order to enable character formatting, the scene's Control Text

plug-in's **Use formatted text** property must be enabled (on). This is done by the Viz Artist designer. The following can be formatted if enabled:

Mode	Command	Procedures
Color	text:set_color_mode	Editing Text Color
Alpha	text:set_alpha_mode	Editing Text Transparency (Alpha)
Kerning	text:set_kerning_mode	Editing Text Kerning
Position	text:set_position_mode	Editing Text Position
Replace character	text:show:replace_list	Replacing a Word
Rotation X, Y and Z	text:set_rotate_xy_mode text:set_rotate_z_mode	Editing Text Rotation
Scaling	text:set_scale_mode	Editing Text Scaling

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. The default mode is Color editing. For more information on text commands, see [Macro Commands and Events](#).

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

In some situations, if text effects are used in graphics there might be conflicts with editing of parameters such as scaling, alpha and position. For example: if the text effect controls the scaling of text, it's 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 such as TFX Color, Scale and Rotation that are used by scene designers.

Word Replacement



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

This list becomes available in the text editor when you press 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 covers the following topics:

- [Editing a Text Item](#)
- [Editing Text Color](#)
- [Editing Text Kerning](#)
- [Editing Text Position](#)
- [Editing Text Rotation](#)
- [Editing Text Scaling](#)
- [Editing Text Transparency \(Alpha\)](#)
- [Replacing a Word](#)

Editing 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 right-click 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.

Editing Text Color

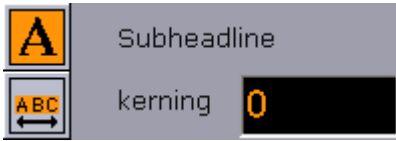


1. Open the appropriate template or page, and select a text element in the Tab Fields or Preview window.
2. Right-click the text editor, and from the [Text editor context menu](#) select **Character Color**.
3. To change the color of a character or selected characters, click a color in the color palette, 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 [Show Properties](#) on how to customize the color palette.

Editing Text Kerning



1. Right-click the text editor and select **Character Kerning** from the [Text editor context menu](#).
2. Select one or more characters, and hold down the **ALT** key while using the right and left arrow keys to change the kerning, or
3. Click the **Kerning for whole text** button.
4. Click the **Text editing mode** 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-in Expose kerning property enables the Page Editor's kerning button.

Editing Text Position

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

✔ **Tip:** The position can also be set using the command `text:set_position_mode` when enabled by the designer.

Editing Text Rotation

1. Right-click the text editor, and from the [Text editor 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.

Editing Text Scaling

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

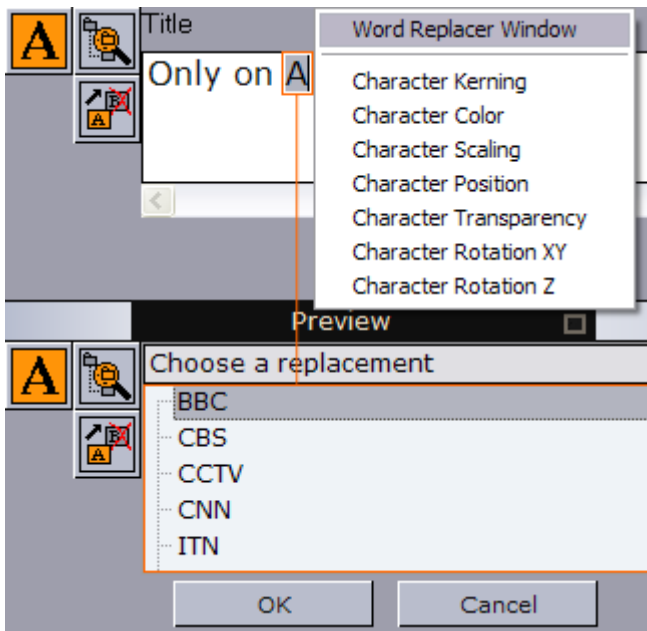
- ✓ **Tip:** Scaling can be also set using the command `text:set_scale_mode` when enabled by the designer.

Editing Text Transparency (Alpha)

1. Right-click the text editor and select **Character Transparency** from the [Text editor context menu](#).
2. Select 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.

Replacing 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.

5.11.3 Database Linking

In order to automate the way a page is filled with content, a page's tab field can be linked at an external data source such as a database or Microsoft Excel spreadsheet (see [Show Properties](#) and [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. Click the Database link button below:



Database link: Opens one of the database link editors so you can link data to a page.



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

This section covers 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 Properties](#) 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

Link to database
 Database connection: Test Table: ['Scalar Tests\$']
 Select column: Headlines where Key column has value: 2
 Custom SQL
 SELECT Headlines FROM ['Scalar Tests\$'] WHERE Key = 2
 Apply

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

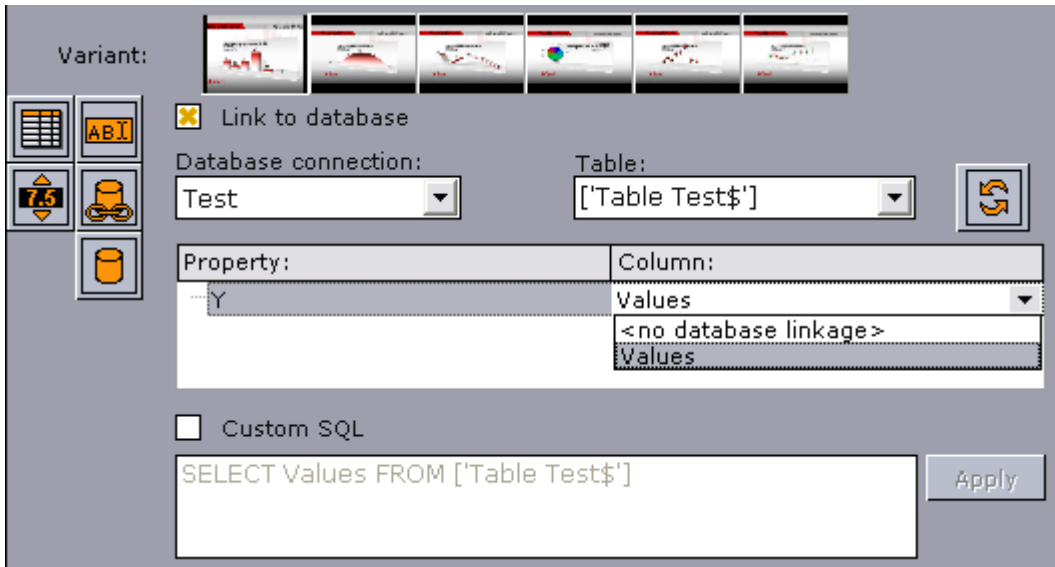
- **Select column:** Sets the Microsoft Excel spreadsheet column, from which the text value is selected.
- **Key column has value:** Specifies the key value to match with the lookup column, which is specified in the Databases settings (see [Show Properties](#)). This mechanism selects the row of the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

To restrict which rows from a Microsoft Excel spreadsheet will 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



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.

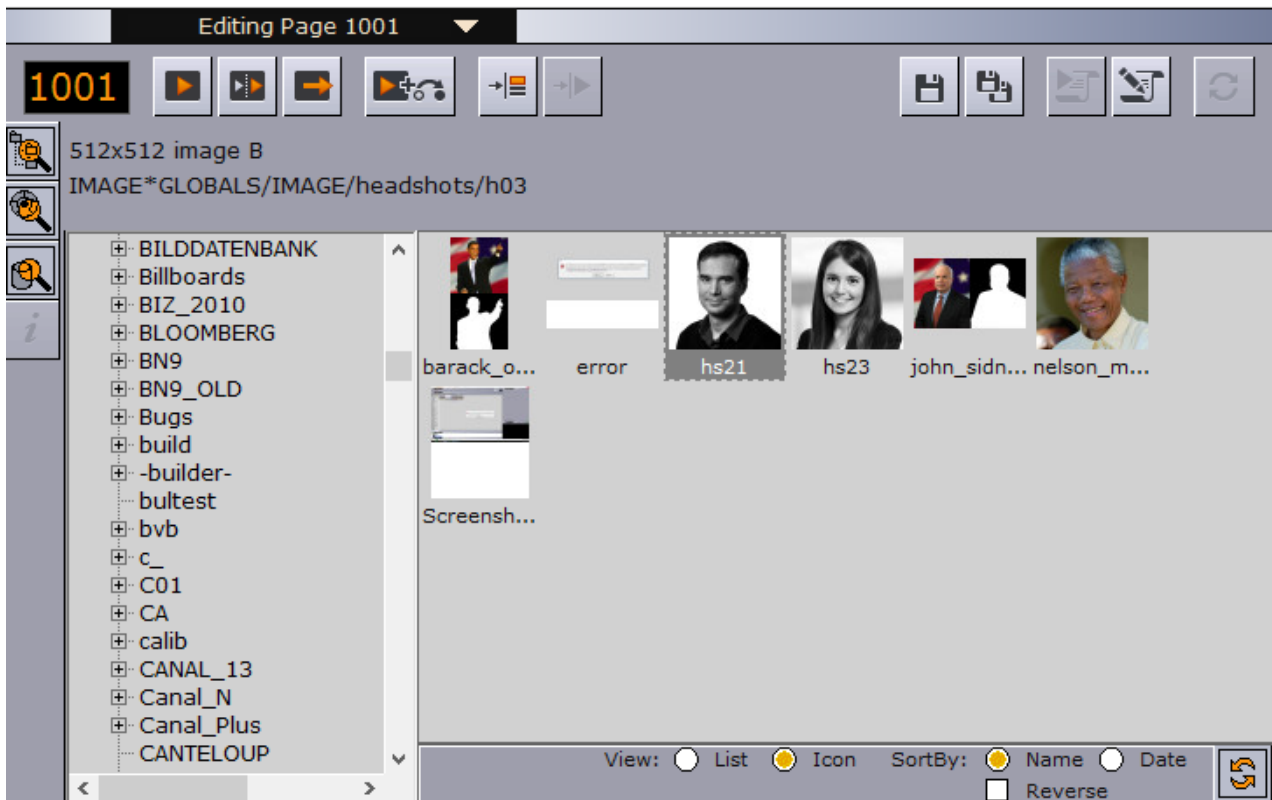
Note: Mapping a tab-field property with a data source immediately updates the graphics.

Microsoft Excel and Unicode Support

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

Info: Unicode characters can be used in Microsoft Excel table names and in cell values of a table, but not in column names.

5.1.1.4 Image Property Editor



The Image Property Editor appears within the Page Editor (as shown above) when selecting an image or item in a page's tab field. When the template contains an image you can load an image from file, Graphic Hub, Viz Object Store or a Vizrt MAM system, if the properties are exposed.



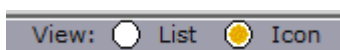
Browse File: Opens a standard Windows file browser.



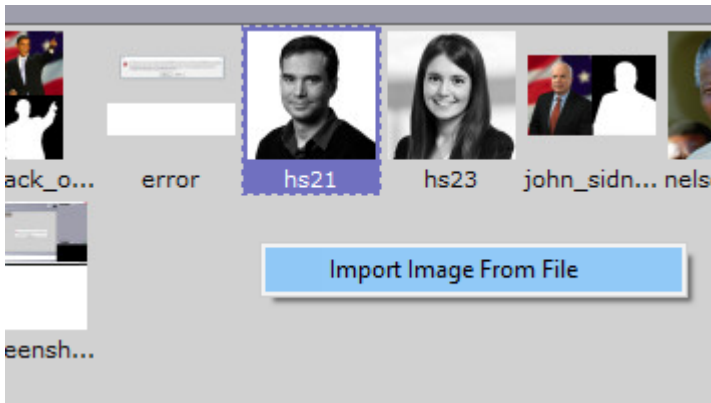
Browse Viz Artist: Opens a browser in the Page Editor for browsing images on the Viz Engine. It's possible to sort by name and date. For convenience both sorting actions can be reversed and refreshed.



Browse: Opens the [Search Media](#) pane for browsing images, video clips and person information. Note that additional Browse buttons will appear if other image repositories are used.



List and icon view: Allows users to view images imported into the image editor as either small thumbnails or in a list displaying only the name of the file. Items are arranged alphabetically in both cases.



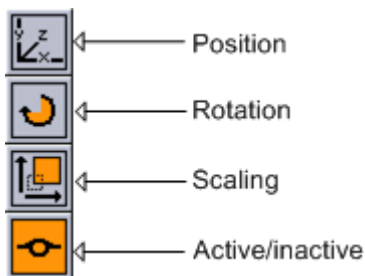
Upload multiple images at a time: Right click in the image property editor and select **Import Image From File**. Hold either the Shift or Ctrl key and left click multiple images to import or Ctrl + A to select and import all images in a folder. Click Open and the images will appear in the editor ready to be imported into a page.

5.11.5 Transformation Properties

It's 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:

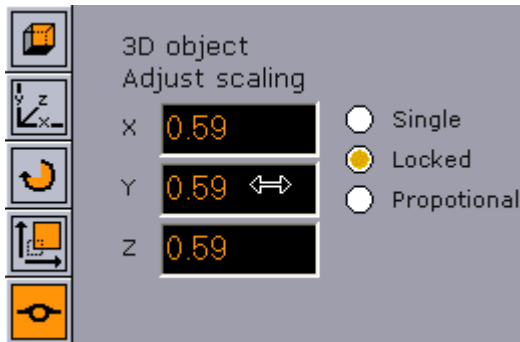
- Active
- Position
- Rotation
- Scaling

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



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

Setting 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 according to the type of object exposed, such as image, text and so on. The screenshot above shows a Geom.

5.11.6 Tables

Editing Template 2560X		
#	text1 (01)	Image 1 (02)
1	Vaar Staude	GLOBALS/IMAGES/HEADSHOTS/vaar
2	Oyvind	GLOBALS/IMAGES/HEADSHOTS/oyvind
3	Gerd	GLOBALS/IMAGES/HEADSHOTS/gerd09
4	Kari	GLOBALS/IMAGES/HEADSHOTS/kari10
5	Wenche	GLOBALS/IMAGES/HEADSHOTS/wenche
6	Beruta	GLOBALS/IMAGES/HEADSHOTS/beruta06
7	Monica Hegg Lund	GLOBALS/IMAGES/HEADSHOTS/monica05
8	Esias	GLOBALS/IMAGES/HEADSHOTS/esias003
9	Blomster Finn	GLOBALS/IMAGES/HEADSHOTS/finn1
10	Eli	GLOBALS/IMAGES/HEADSHOTS/eli
11	Bjørg	GLOBALS/IMAGES/HEADSHOTS/bjorg002

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

The cells contain different editors depending on the data type they display. Changes made in the table editor are instantly 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 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, feature a small text editor with spin button functionality. Click the spin buttons or use arrow keys, or enter numbers directly on the keyboard to change value.



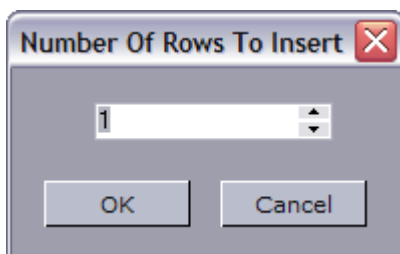
- **Checkbox Cells:** For Boolean values, such as hide and show, a checkbox will be displayed.
- **Table Keyboard Navigation:** To move between the cells, use the arrow keys on the keyboard.
- **Sorting:** The table can be sorted manually by clicking on the header bar of the column to sort by. Rows can 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.

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

- **Customizing a Table View:** Right-click on the table header bar, to make a context menu appear. Columns can be hidden or shown by selecting or unselecting the column. The table editor context menu contains the following columns:

#	a_1 (2)	Games		P...
1	Manchester Utd	38	✓ #	87
2	Chelsea	38	Description	85
3	Arsenal	38	✓ a_1 (2)	83
4	Liverpool	38	✓ Games played (3)	76
5	Everton	38	✓ Won (4)	65
6	Aston Villa	38	✓ Draw (5)	60
7	Blackburn	38	✓ Lost (6)	58
8	Portsmouth	38	✓ Goals against (7)	57
			✓ Goals for (8)	
			✓ Points (9)	
			✓ Auto Width	
			Enable Sorting	
			Clear Sorting	
			Insert Row Ins	
			Insert N Rows ...	
			Edit Row	
			Delete Selected Rows Del	
			Select All Ctrl+A	
			Cut Ctrl+X	
			Copy Ctrl+C	
			Paste Ctrl+V	
			Save Settings	

- #: 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 insertion. The number of rows you can add is 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: The number of 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 Guide.

Populating Tables Using Macro Commands

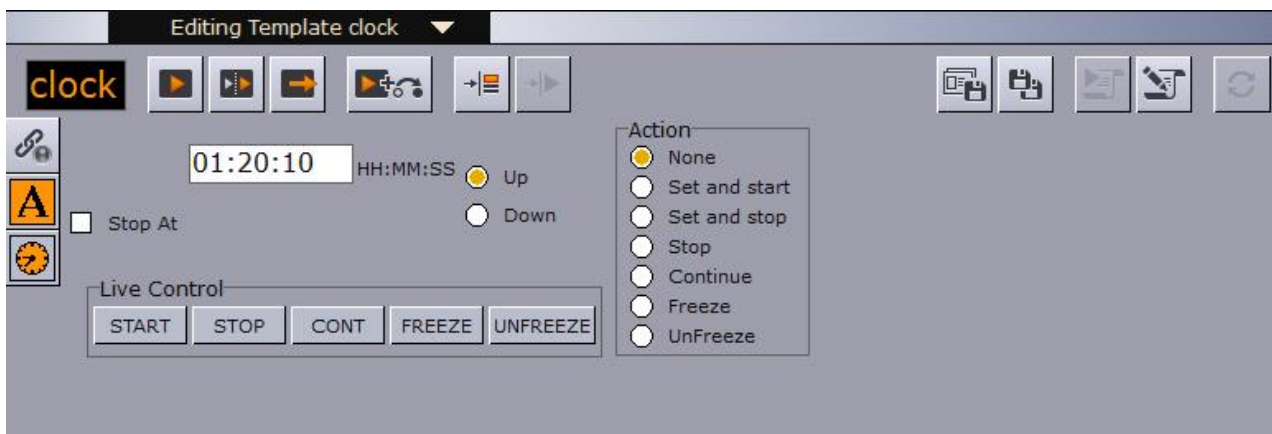
You can use Viz Trio's macro language to populate a table with data through a script, or through an external application that connects via a socket connection. To populate a table, the macro `table:set_cell_value` is used. The macro takes 4 arguments:

- 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
```

5.11.7 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.



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.

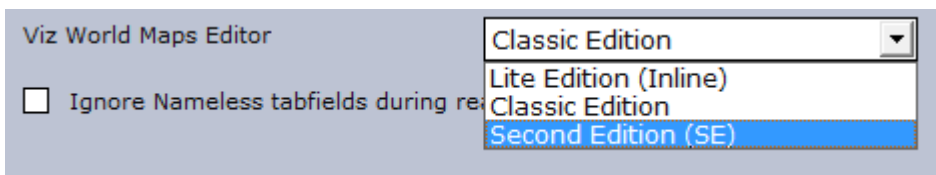
The clock editor's properties are:

- **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.

5.11.8 Maps

Viz World Client is available to Viz Trio in both a simplified and a full version. The simplified version is enabled in the configuration window, see [General](#) settings under the [User Interface](#) section. The full version launches the selected Map client, either in Classic Edition or the new Second Edition (SE).

Select the preferred Map client in **File > Configuration > General > Viz World Maps Editor:**



Note: Make sure that the Viz Maps Client is installed on the Trio client and make sure that the Viz Engines (both local preview and On-Air render Engines) connect to the same Viz World Server. To configure the Viz World Server address in *Viz Engine* go to **Config > Maps** on the Viz Engine server.

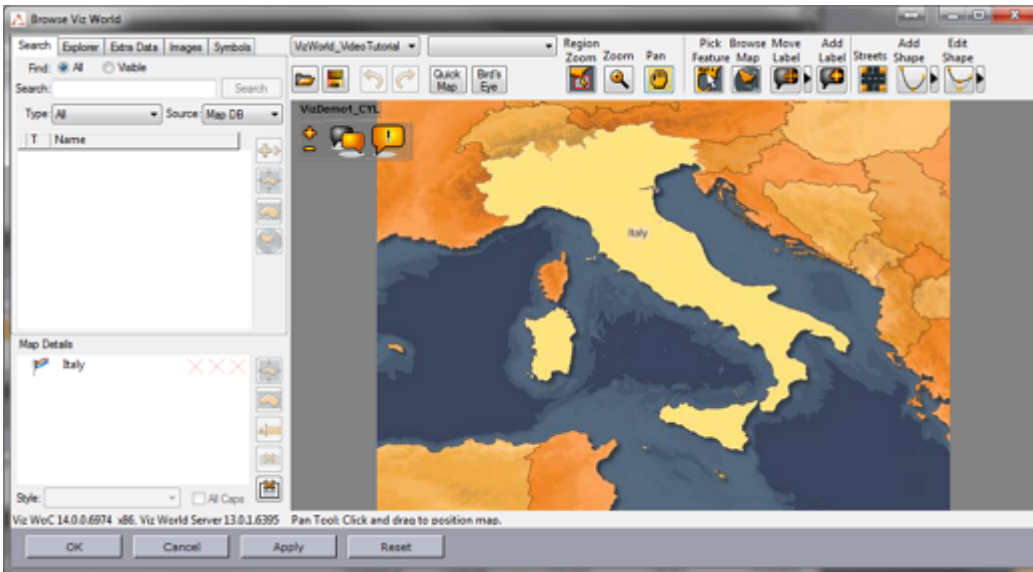


Edit Map: If the map's tab-field property has more than one property exposed, clicking this button will open the corresponding editors (for example text and text kerning).



Browse Viz World: Opens the full version of the Viz World Client. To use the full version, disable the small version. You can also click the **Select Map** button to launch the selected Map client.

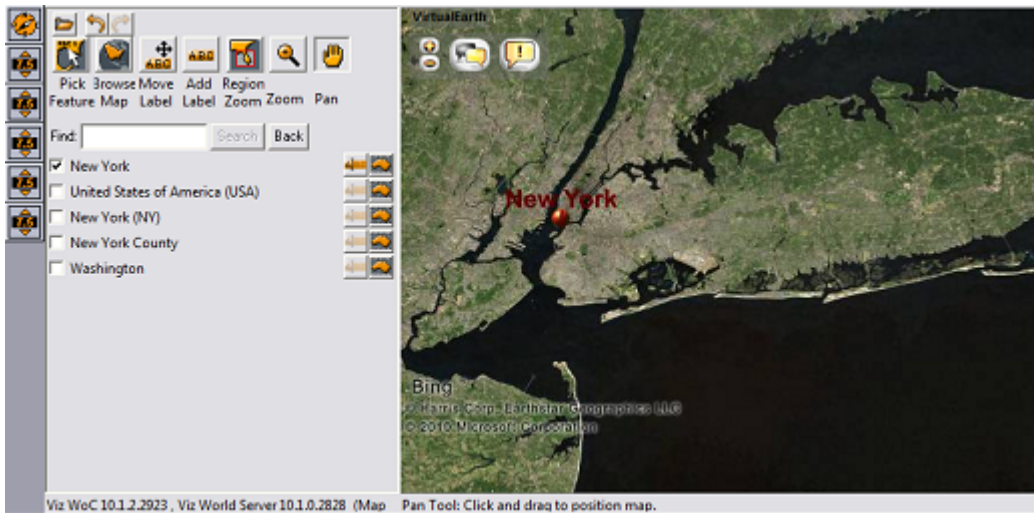
Maps Classic Client



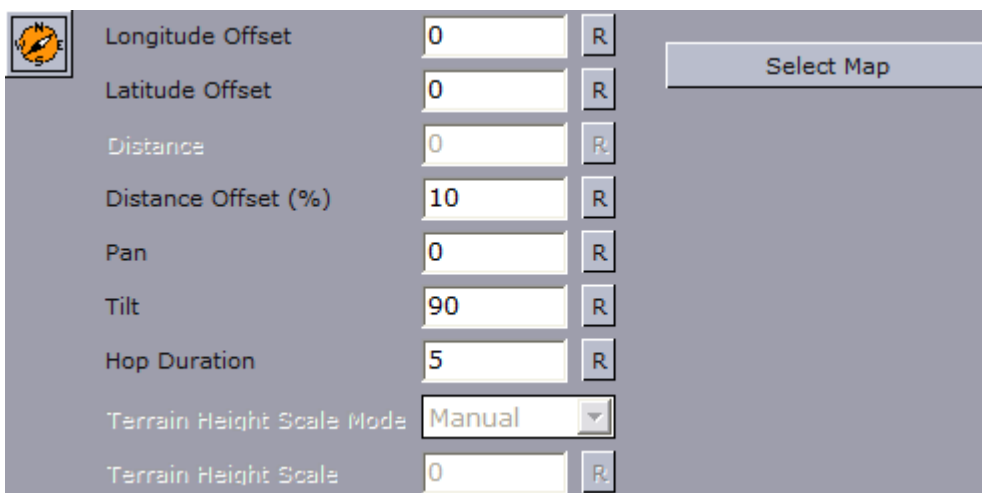
Maps Second Edition Client



Maps Lite Edition (inline)



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 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 Guide for information on Control World
- Viz World Client and Server User Guide
- [Field Linking and Feed Browsing in Viz Trio](#)
- [Creating a New Database Connection](#)
- The `dblink` macro commands for use with scripting.
- [Scripting](#)
- [Setting the Transition Effects Path](#)
- [Adding Transition Effects Using the Page Editor](#)
- [Creating Transition Effects](#) in Viz Artist

5.12 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 covers the following topics:

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

5.12.1 Create New Scroll Editor

Create New Scroll Template:

Template Name

Scene Directory

SCENE*02_GFX/NewScroll

Template description

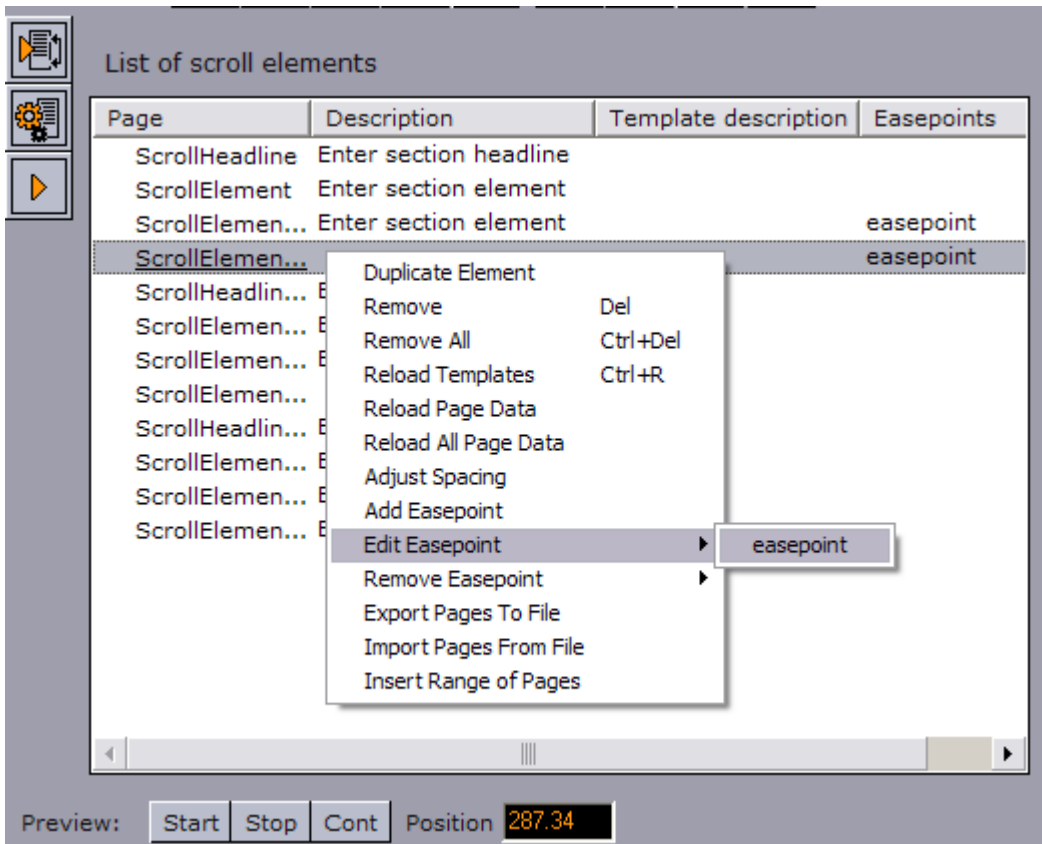
Scroll Area Width

Scroll Area Height

Show Scroll Area

- **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.

5.12.2 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 Scroll list context menu options 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 create new scroll 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.

5.12.3 Scroll Configuration



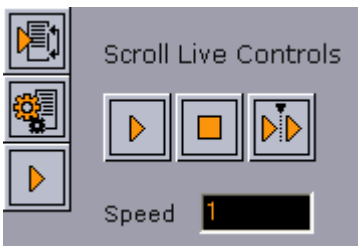
Click the **Scroll Config** button to open the scroll main property page.

- **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.

5.12.4 Scroll Live Controls



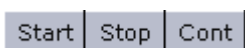
Clicking the Scroll Live Controls button, opens the Scroll Live Controls sub page, which controls the start, stop, and continue actions of a scroll on-air.



To activate the Scroll Live Controls, click the **Take** button in the Scroll Editor. When active, the Live Control buttons are green and can be used as follows:

- **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.

5.12.5 Scroll Control



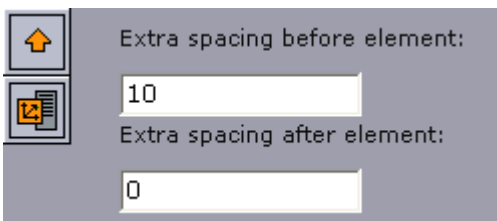
The **Start**, **Stop** and **Continue** buttons in the lower left corner of the Scroll Elements editor enables a preview of the scroll in the local preview renderer window.

- **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.

5.12.6 Element Spacing

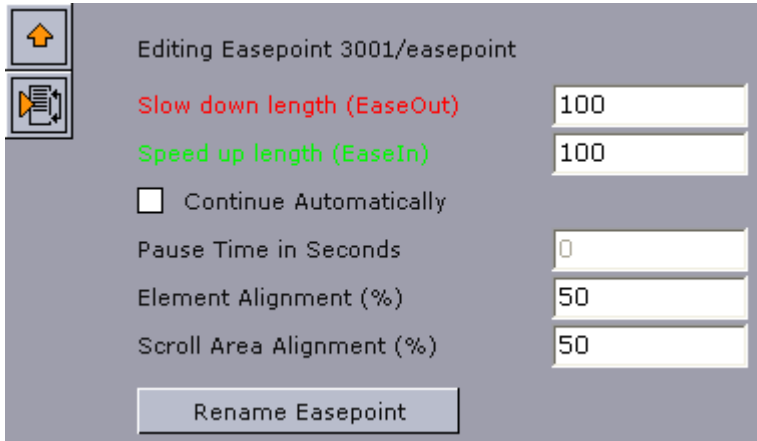
Open the adjust element spacing editor:



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

5.12.7 Easepoint Editor

The Easepoint editor allows 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 as a 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 to 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.

5.12.8 Working with Scrolls

This section covers the following topics:

- [Creating a Scroll](#)
- [Configuring a Scroll](#)
- [Editing a Scroll Element](#)
- [Copying an Element in the Scroll Elements List](#)
- [Testing a Scroll](#)

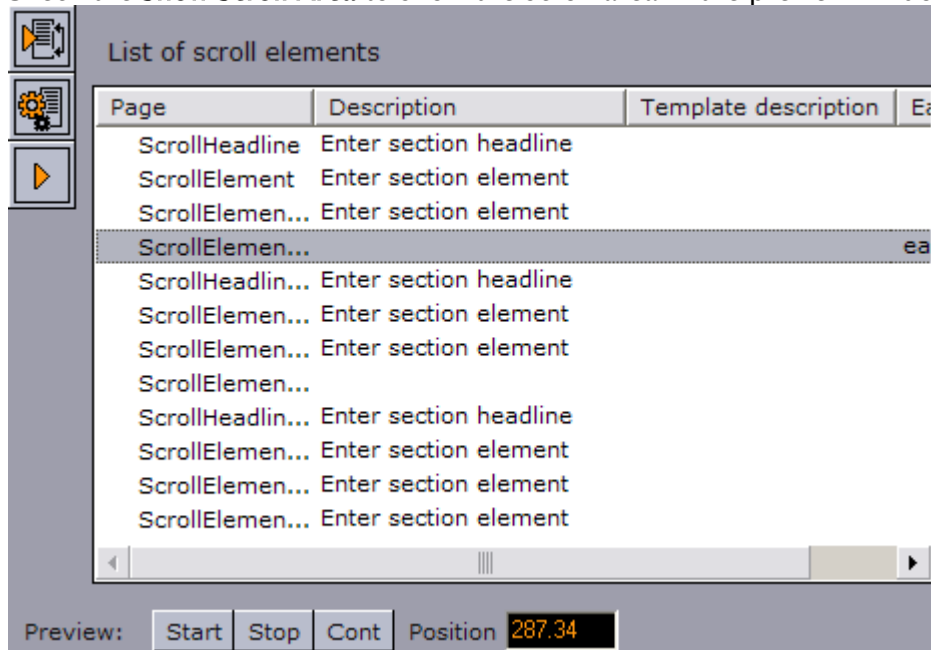
Creating a Scroll

The screenshot shows a dialog box titled "Create New Scroll" with a subtitle "Create New Scroll Template:". The dialog contains the following fields and controls:

- Template Name:** A text input field containing "NewScroll".
- Scene Directory:** A text input field containing "02_GFX" and a browse button (three dots in a square).
- SCENE*02_GFX/NewScroll:** A text input field containing the path.
- Template description:** A text input field containing "Viz Trio scroll".
- Scroll Area Width:** A numeric input field containing "500".
- Scroll Area Height:** A numeric input field containing "375".
- Show Scroll Area:** A checkbox that is checked.
- Save Template:** A button at the bottom of the dialog.

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: Since different templates are required for different scroll elements, the graphics designer should create templates for all scroll elements required.

Configuring a Scroll

1. Click the **Scroll Configuration** button to adjust layout and animation speed parameters.
2. Adjust spacing before 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.

Editing a Scroll Element

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

Copying an Element in the Scroll Elements List

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

Testing 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.

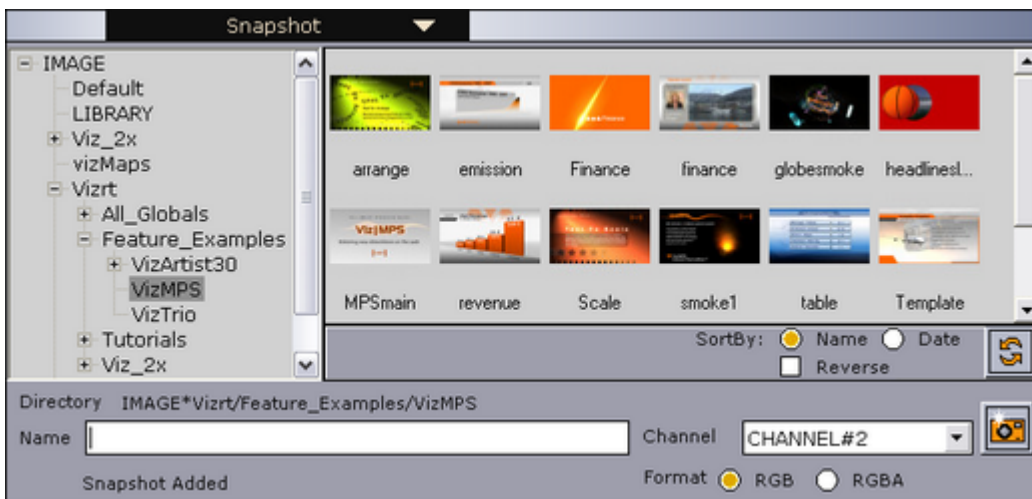
5.13 Edit Show Script

Select **Edit Show Script** from the [Page Editor](#) drop-down menu to open the **Choose Show Script** window, where you can edit the script assigned to the show.

- ✓ **Tip:** The **Choose Show Script** window can also be opened by selecting **Script > Edit Show Script** from the [Template List](#) context menu, or by clicking the **Show Script** browse button in the [Show Properties](#) window.

5.14 Snapshot

Select **Snapshot** from the [Page Editor](#) drop-down menu to open a window where a snapshot of the current view on the on-air renderer can be taken.



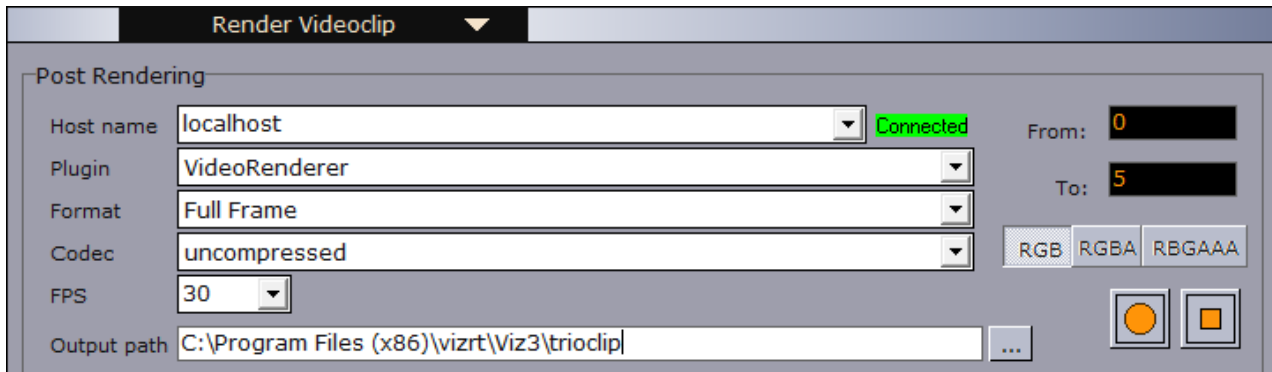
Specify the channel to take a snapshot from, and select the preferred format. Browse the Image database to select the folder where the snapshot is to be stored. Images can be sorted by **Name** or **Date**. Select the **Reverse** check-box to reverse the sorting. To refresh the content of the selected 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 in the selected folder.

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

5.15 Render Videoclip



The **Render Videoclip** option exposes an editor for post rendering 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 ten 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 are automatically set up and do not require configuration.

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

The settings in the post render frame stay the same after restarting Viz Trio.

- **Host name:** Sets the Viz machine that will post render the scene. It is also possible to define the port number (<host-name>:<port-number>).
- **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.
- **FPS:** Sets frame per second. Available options are 30, 60, 29.97 and 59.94.
- **Output path:** Sets the output path. A corresponding browse button is available when localhost is selected as host. As Viz Trio cannot browse the file system on other machines, manually make sure that the output path is valid if defining the path on another machine. 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.
- **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).

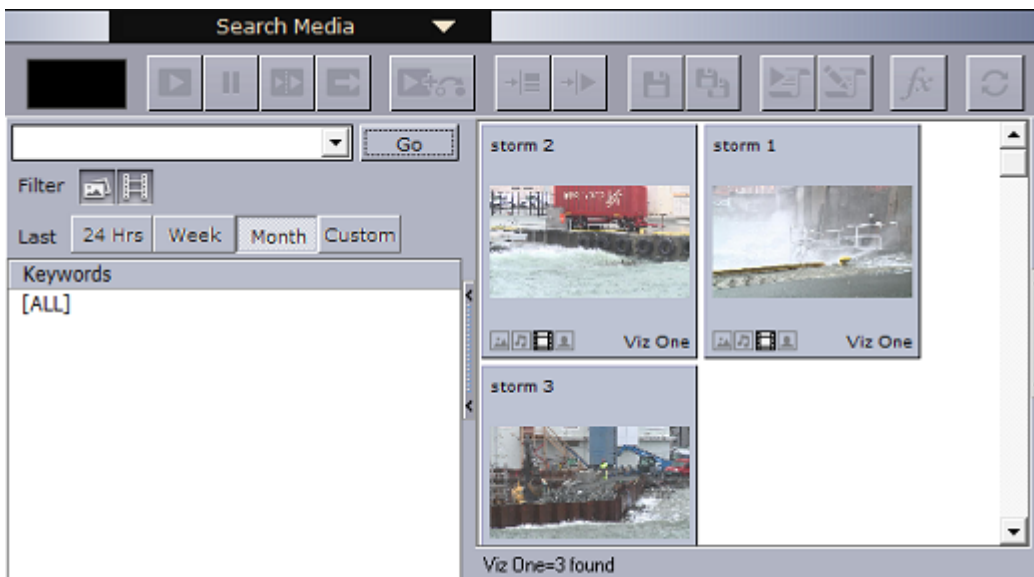
- **Record button (circle):** Starts the rendering process.
- **Stop button (square):** Stops the rendering process before the configured stop time.

5.16 Search Media

Select **Search Media** from the top right drop-down menu. Viz Object Store traditionally stores only still images and person information and Viz One traditionally stores only video, audio and video stills. The Search Media editor lets you search both the Viz Object Store database and the Viz One MAM system for images and video clips.

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

Media items can be dragged and dropped into a show, show playlist, playlist, template or page (for example, full screen or part of the graphics), where they can be double-clicked to open the **Search Media** editor.



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

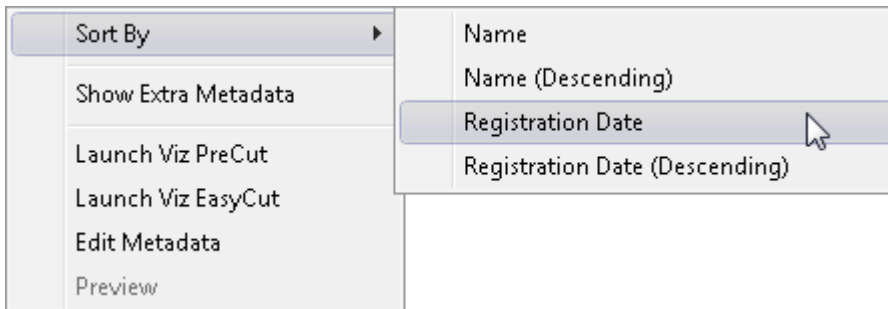
Tip: The number of fetched media files can be specified in the [General](#) section of Trio Configuration. Scroll down to the bottom of the list to fetch additional media files.

When the media search frame is shown and you have many hits only the first N hits are shown. When scrolling down in the result list the next N hits will be fetched. There is no more button. The next chunk of hits get automatically searched when scrolling down to the bottom.

This section covers the following topics:

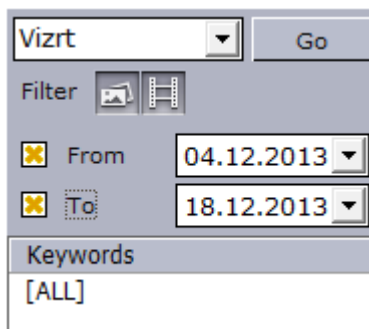
- [Media Context Menu](#)
- [Search and Filter Options](#)
- [Ordering Metadata Fields for Video and Images](#)

5.16.1 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 One).
- **Preview:** Previews images using the *Windows Picture and Fax Viewer*. Only available for Viz Object Store items.

5.16.2 Search and Filter Options

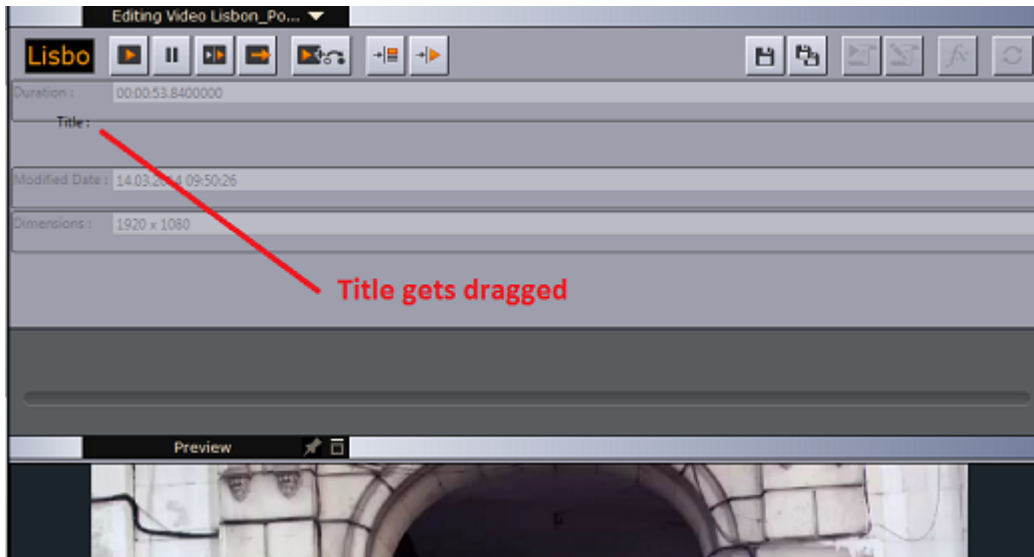


- **Search field:** Combo box for entering a search criterion. Previously entered search criteria are remembered per session.
- **Filter:** Enables/disables the image or video filters.
- **From/To:** Filters the search result based on From and To registration dates.
- **Keywords:** Filters the search result based on keywords.

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

5.16.3 Ordering Metadata Fields for Video and Images

Fields in the metadata view of a video or an image can be dragged and dropped to change their order. This is typically used to display important fields at the top of the list.



5.17 Import Scenes

Select **Import Scenes** in the drop-down menu to open a window where Viz Artist scenes can be selected and enabled for Viz Trio.



Select scenes to import into the template list and click **Import**. It's possible to select multiple scenes and import scenes from any scene folder. To make the scenes ready for playout quickly, click **Import and Initialize**. This loads all scenes on the program and preview renderer (Viz Engine).

Sort scenes by **Name**, **Date**, or **Reverse** sorting preferences or **Refresh** content.

This section covers the following topics:

- [Importing Recursively](#)
- [Importing Scenes with Toggle or Scroller Plugin](#)

5.17.1 Importing Recursively



Import Recursively lets you import a whole folder or folder tree structure. This does not initialize the show on the renderers.

Right-click the folder in the scene tree and select **Import Recursively**.

Caution: If a scene has the same name as an existing template in the current show, an attempt is made to merge it with the existing template (for example, scene 1000 with template 1000).

5.17.2 Importing Scenes with Toggle or Scroller Plugin

A warning appears when you attempt to import scenes with a Toggle or Scroller plugin, since such scenes usually do not need to be imported. However, if you are using a Toggle or Scroller plugin in scenes that are not Transition Logic or ticker-related, you have the option of importing the scenes as templates.

5.18 Graphics Preview

5.18.1 Real-time Rendering of Graphics and Video



The local Preview window displays a WYSIWYG (what you see is what you get) representation of the graphics and video.

- The defined title and safe area and bounding boxes are displayed. Bounding boxes are related to the graphics scene's editable elements (see [Tab Fields Window](#)).
- Timeline for scrubbing full screen video preview.
- Control interactive scenes by clicking the scene inside the local preview window. You can then adjust an offside line in Trio's local preview by reading a template referencing an interactive scene, for example.

5.18.2 Floating or Moving the Preview Window

Use the **pin** button at the upper right corner of the preview window panel (beside the green **Connected** button) to un-dock the preview window.

- If the preview window is docked (attached) to the Trio main window, clicking the pin-button will detach the Preview window.
- If the preview-window is un-docked from the main Trio window, clicking the pin-button will dock the preview window to the main Trio window.

This section covers the following topics:

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

- [OnPreview Script for a Graphical Element](#)

Graphics Control Buttons

At the top of the preview window:



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



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



Stop stops the scene animation.



The **Time slider** lets you scrub the timeline by holding in and moving your cursor in the value field. The cursor changes to an arrow to indicate that the time value can be changed.



Click **TA** to show or hide the Title Area in the Preview window.



Click **SA** to show or hide the Safe Area in the Preview window.



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

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 left button displays the graphics in an RGB mode with a default background color (black).
- The middle button displays the graphics in Key mode, showing how the key signal looks like.
- The right button displays an RGB preview with a background image, illustrating how the graphics appear on top of a video.

The respective preview modes are shown below:



5.18.3 Connection Status

Connected

The **Connection** status of the local Viz renderer is shown in the upper right corner of the Preview window. Click the **arrow** beside it to display a context menu.



These functions are mainly for testing and debugging purposes. The context menu can show these menu items:

- **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 Viz Engine running on the same computer as the Trio client. This menu item is only active for local preview.
- **Reconnect Viz:** Allows Trio to reconnect to a local preview engine after an engine running the local preview has been shut down. This option is visible regardless of whether an external preview is running.
- **Reconnects to the external Viz Engine.** Only the local Trio client reconnects to the remote Viz Engine.

If Trio has been started with an *external* Viz Previewer (and only then), the following menu choice appears in the context menu:

- **Set OnAir:** Sets the external Viz Engine in OnAir Mode, so that it exits Viz Artist's Design Mode, for example, and functions as a renderer engine.

Note: In this context, External Viz Previewer should not be confused with External Viz Preview *settings* in the *profile*. An External Viz Previewer is another way of making an external Viz Engine act as a local Viz preview. You can configure the previewer using the Viz Trio **startup parameter**-*vizpreview your-vizpreview-hostname-here*.

5.18.4 OnPreview Script for a Graphical Element

This feature supports triggering an OnPreview event in a script for a scene. The event is only triggered for the local preview in Trio.

OnPreview scripts can be created using a whole scene or a container.

Creating a Scene Script:

1. Open a scene in Viz Artist.
2. Go to **Scene Settings > Script** and select **Edit** (If it's not already showing in the script editor).
3. Enter a script in order to write a unique ID for the scene, so that it can later be identified in the console output:

```
sub OnPreview(doPreview as Integer) PrintIn "OnPreview in test1 scene script: "
& doPreview end sub
```

4. Press **Compile & Run** button below the script editor.
5. Save the scene.

Creating a Container Script

1. Add a group container to the root of the scene.
2. Go to **Built Ins > Container Plugins** and select the **Global** folder.
3. Drag the **Script** plugin onto the container/group. A script editor opens.
4. Enter a script that writes a unique id for the scene that can be identified in the console output later.

```
sub OnPreview(doPreview as Integer) printIn "OnPreview in test1 container
script: " & doPreview end sub
```

5. Press **Compile & Run**.
6. Save the scene.

Follow the steps below in Viz Trio and open the local preview command window to check that the OnPreview event is triggered:

- Import a scene (Transition Logic scenes, Combo Templates, Concept/Variant scenes, TrioScroll scenes, Stillstore image scene etc.) with an onPreview script into Trio. See how to make an OnPreview script above.
- Open the local preview Viz engine console output window from **Connect > Show Command**.
- Read the imported template and do a page take.

The page editor is used for editing pages. Depending on the template or page edited, page editor can display a wide range of specialized editors. Pages are edited using a wide range of editors: for example text, database and scroll editors. When pages have editable elements, it's possible to jump from element to element by pressing the **Tab** key. It's also possible to select elements by clicking them in the local preview window. The page editor is located in the upper right corner of the UI and is shown by default when starting Viz Trio.

- Save the page.
- Read the page by double-clicking it or issuing a Trio command. The local Viz engine console then triggers an OnPreview script event.

5.19 Video Preview

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

5.19.1 Video Control Buttons and Timeline Display

Located above the preview window:

Play starts and stops the video clip currently loaded in local preview. If a Mark in point has been set, the clip will jump back to the Mark in point. Clips play automatically when loaded in Preview.


Pause freezes and unfreezes the video clip currently loaded in local preview.

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

The **Timeline** has a marker and you can to set Mark in and out points. Clip length in hours, minutes, seconds and milliseconds is displayed.

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 - new clips saved in the MAM system are not created. New clips are made using Viz PreCut (and Viz EasyCut, if applicable).

5.20 TimeCode Monitor

The TimeCode Monitor (TC Monitor) shows the progress of the current video clip on a specific channel.

5.20.1 TC Monitor

Note: The TimeCode Monitor keeps track of the time code and the video server channels. The system must therefore be configured to be integrated with a video server. In order to monitor video clips embedded in graphics, the render engine must also be defined as a video channel (see [Output](#)).

This section covers the following topics:

- [TimeCode Monitor Options](#)
- [Enabling the TimeCode Monitor](#)
- [Disabling the TimeCode Monitor](#)
- [Monitoring a Video Time Code](#)

5.20.2 TimeCode Monitor Options

The TimeCode Monitor displays three columns: Channel, Status and Armed.

- **Channel:** Channel name. Video channels are displayed as light grey rows, with progress information in the **Status** column. Non-video channels are shown as dark grey rows without status.
- **Status:** Name of the current item, followed by any cued item, displayed above the progress bar. Current Location time and Remaining time are shown below the progress bar. The progress bar changes color to indicate status:
 - *Green:* Video playing
 - *Red:* Video playing, less than 10 seconds remaining
 - *Blue:* Video finished
- **Armed:** Displays the armed state of an element on a channel.

List rows can be shown or hidden using the context menu on each item:

- **Hide:** Hide the selected channel
- **Show All:** Show all channels

At the bottom of the window, **zoom** lets you adjust display size. **Profile** displays the current profile name, and contains a list of other profiles to choose from.

The Timecode Monitor opens in full screen by default. Press **ESC** to exit full screen.

5.20.3 Enabling the TimeCode Monitor

- To activate the TimeCode Monitor, go to **Configuration** -> **Keyboard Shortcuts** and use the command:

```
gui: show_timecode_monitor
```


When activated, the time code status will be shown in the [TC Monitor](#) whenever a video is played.

5.20.4 Disabling the TimeCode Monitor

- To disable the TimeCode Monitor, use the command:

```
gui: hide_timecode_monitor
```

5.20.5 Monitoring a Video Time Code

1. Make sure that the TimeCode Monitor is activated, see [Enabling the TimeCode Monitor](#).
2. Double-click a video or composite element (graphics and video combined).
3. Play the video. The progress of the video is displayed in both the TC Monitor, and the inline TC Monitor above the local Preview panel.

5.21 Field Linking And Feed Browsing In Viz Trio

Viz Trio supports browsing tab-field values from external sources. Instead of editing tab-field values manually, values from an external feed can be selected.


 **Note:** Viz Trio only supports the Atom 1.0 feed standard.

This section covers the following topics:

- [Workflow](#)
- [Overview](#)
- [Technical](#)
- [Field Linking](#)
- [Feed Browsing](#)

5.21.1 Workflow

1. Import a regular scene as a template in Trio.
2. Read the template.
3. On top of each tab-field of the template, a link button opens the collection picker, where you can specify a feed URI. You must also specify which part of the feed item should be used as a value: title, content, link with a certain link relation, entry or locator.

 **Note:** The link button is not available for pages. Linking to an Atom feed is a Viz Trio template property that is shared by all pages using that template.

4. Press **Save template**.
5. When establishing a valid feed connection, the tab-field value can be selected from the feed items.

5.21.2 Overview

Feed URIs can be defined in the control object of the scene design in Viz Artist. the feed URI for a tab-field in a template can also be defined inside Viz Trio. Each page then uses the feed browser to edit the tab-field value. The specified feed URI in Viz Trio overwrites any existing feed URI from the control object. Reimporting the scene does not replace the specified feed URI with the field definition specified in control object.

A tab-field contains several properties, each of which relate to a control plug-in in the scene. Properties are grouped under a tab-field using a naming convention.

Example: The ControlText plug-ins specifying the field identifiers 1.name, 1.score and 1.image forms a tab-field 1 with the properties' name, score and image. Selecting a feed item for tab-field 1 instantly applies the parts of the selected item to the properties name, score and image. The individual property values are editable as usual. The feed item selection assists fill-in only.

Feed linking can be used across all tab-field types.

Note: Make sure to define matching object types such as a thumbnail link for control image tab-fields.

Field linking in Viz Trio is a two-step procedure:

1. **Field Linking:** Linking the fields in a template to various parts of data that comes from entries in a feed. This is a setup step that is usually done once for a template.
2. **Feed Browsing:** After field linking has been set up and saved for a template, a new property editor for selecting an entry from a feed will be available. This feed browser lets you select an item in a feed that contains the values that should be applied to the field values.

Note: Viz Trio only supports the Atom 1.0 feed standard.

The feed browser supports two types of feeds:

- Flat feeds
- Hierarchical feeds (folder structure)

Tabfield Grouping

To be able to set multiple tabfield values from the same feed entry, the tabfields must be grouped together. This is done by giving them the same prefix followed by the period character. For example, the following tabfield names will generate two groups of tabfields (candidate1 and candidate2):

```
candidate1.name candidate1.image candidate2.name candidate2.image
```

5.21.3 Technical

The following XML Namespace Prefixes are used when referring to XML elements:

XML Namespace Prefixes

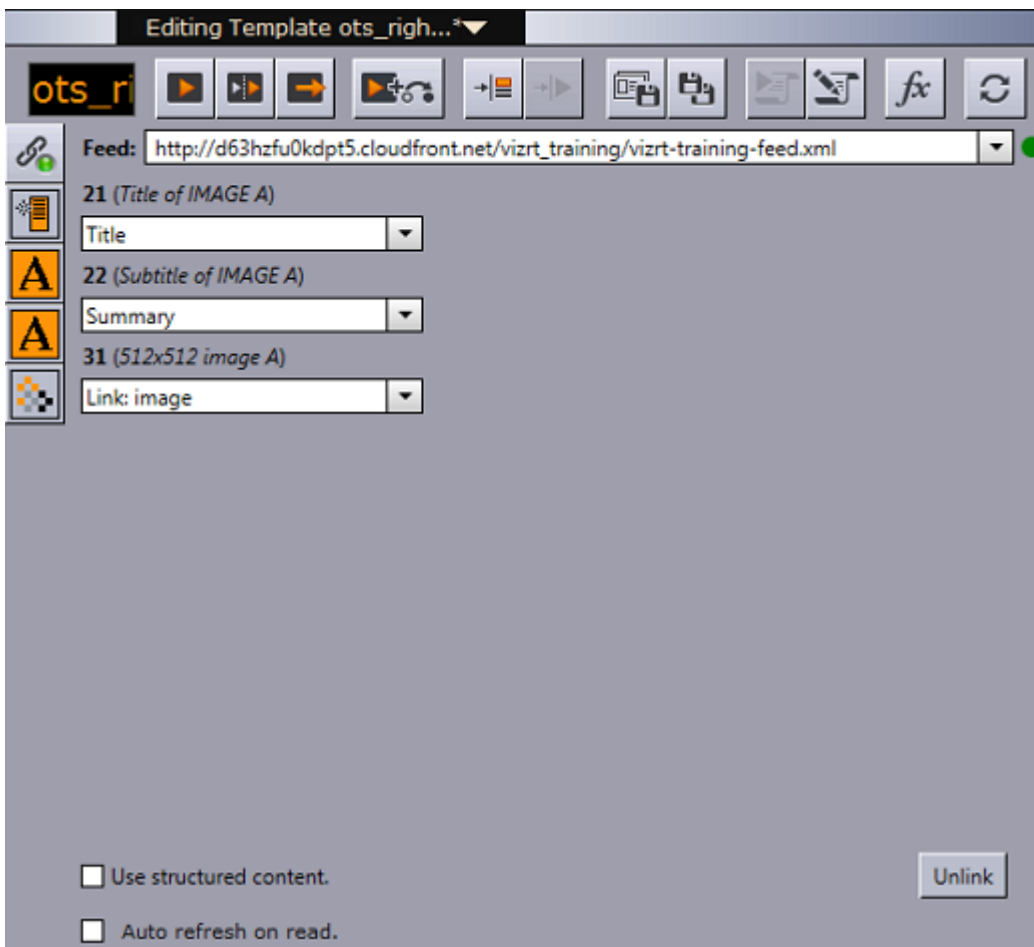
Prefix	URL
atom	http://www.w3.org/2005/Atom
viz	http://www.vizrt.com/types
vaext	http://www.vizrt.com/atom\ext
media	http://search.yahoo.com/mrss/
thr	http://purl.org/syndication/thread/1.0
opensearch	http://a9.com/-/spec/opensearch/1.1/

Example: The notation `<atom:entry>` is to be interpreted as referring to the same element as `<entry xmlns="http://www.w3.org/2005/Atom">`.

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in RFC 2119.

5.21.4 Field Linking

Field linking is the process of linking the fields in a template to various parts of data that comes from entries in a feed.



Fields That Can be Linked

- `<atom:author>`
The personal details of the author of the selected `<atom:entry>`, which might be an individual person or an organization. The data is provided in child elements as follows:
 - `<atom:name>`: The name of the author. The value of the `<atom:name>` element inside the first `<atom:author>` element that applies for the entry is recorded as the field value. This child element is required if you are specifying the `<atom:author>` element.
 - `<atom:uri>`: A URL associated with the author, such as a blog site or a company web site. The value of the `<atom:uri>` element inside the first `<atom:author>` element that applies for the entry is recorded as the field value. This child element is optional.
 - `<atom:email>`: The email address of the author. The value of the `<atom:email>` element inside the first `<atom:author>` element that applies for the entry is recorded as the field value. This child element is optional.
- `<atom:content>`
 - Both inline and URL content is supported.
 - There is no type match, so it is up to the Trio operator to figure out what content can be used.

- For URL content the resource is downloaded first and then applied as the value. If you just want the URL you should use `<atom:link>`.
- `<atom:entry>`
The whole entry XML.
- `<atom:link>`
This element defines a reference from an entry to a Web resource; in other words this is the value of the `href` attribute.
- `<vaext:locator>`
The text of the Vizrt Atom Extension `<vaext:locator>` element which has a `vaext:type` attribute equal to the `mediatype` attribute of `<viz:fielddef>` element in the model.
- `<atom:published>`
This element is a Date construct indicating the initial creation or first availability of the entry. The value of the `<atom:published>` element in the entry is recorded as the field value.
- `<atom:summary>`
This element is a Text construct that conveys a short summary, abstract, or excerpt of an entry. The content of the `<atom:summary>` element in the entry is recorded as the field value.
- `<atom:thumbnail>`
- The value of the `url` attribute of the first thumbnail element in the entry is recorded as the field value.
- `<atom:title>`
This element is a Text construct that conveys a human-readable title for an entry or feed; the title of the selected `<atom:entry>`.
- `<atom:updated>`
This element is a Date construct indicating the most recent instant in time when the selected `<atom:entry>` was modified. The value of the `<atom:updated>` element in the entry is recorded as the field value.

Fields That Cannot be Linked

- `<atom:category>`
- `<atom:id>`
- `<atom:rights>`
- Any other elements.

Elements in Atom Feed

Elements that should be present in the `<atom:feed>` for the full user experience:

- **OpenSearch link:**

```
<atom:link rel="search" type="application/opensearchdescription+xml"
href="http://example.com/opensearchdescription.xml"/>
```

- Optional, required for support for server side searching.
 - The `opensearchdescription` xml file must contain a template node that returns search results as an atom feed:

```
<opensearch:Url type="application/atom+xml" template="http://..." />
```

- See <http://www.opensearch.org/Specifications/OpenSearch/1.1> for more details.
- **Up link:**

```
<atom:link rel="up" type="application/atom+xml;type=feed" href="http://..." />
```

- Optional, required for nested collections:
 - The up link should link to the parent folder feed for hierarchical feeds.
 - The up link must have a type string equal to "application/atom+xml;type=feed".

Elements in Atom Entry

Elements that should be present in the <atom:entry> for the full user experience:

- **Self link:**

```
<atom:link rel="self" type="application/atom+xml;type=entry" href="http://..." />
```

- The self link must link to the URL that will return the <atom:entry> xml.
 - The self link must have a type value equal to "application/atom+xml;type=entry".
 - Needed for refresh/update of values from the item to work.
 - Needed in combination with the up link for remembering and showing the selected feed entry in a hierarchy of feeds (folder structure). Not needed for remembering selection in flat feeds since then the atom:id will be used.

- **Up link:**

```
<atom:link rel="up" type="application/atom+xml;type=feed" href="http://..." />
```

- The up link must have a href value that is the URL of the feed the entry is in.
 - The up link must have a type value equal to "application/atom+xml;type=feed".
 - Needed in combination with the self link for remembering and showing the selected feed entry in a folder structure.

- **Down link:**

```
<atom:link rel="down" type="application/atom+xml;type=feed" href="http://..." />
```

- Required if this <atom:entry> is to be considered a subfolder instead of a normal <atom:entry>.
 - The down link must have a type value equal to "application/atom+xml;type=feed".
 - The down link must have a href value that is the url of the feed (folder) the entry represents.

- Link may contain `thr:count` attribute (RFC4685) indicating how many children there are. If the value of `thr:count` is 0 (zero) then the folder will not be loaded since it is empty. This is an optimization.

- **Thumbnail link:**

```
<media:thumbnail url="http://..." />
```

- Required for thumbnail icons to display on the entries in the feed browser:
 - The URL value must reference a JPEG or PNG image resource.
 - If many thumbnails are defined the first one will be selected as the default. This is according to Media RSS Specification Version 1.5.0.

- **Locator**

```
<vaext:locator type="application/vnd.vizrt.viz.geom">GEOM*Vizrt/Tutorials/VizTrio/...</vaext:locator>
```

- Required to be able to link a tabfield to a locator. A locator is a path that is not representable with the URI `href` in an `<atom:link>`. The selected value is the text node in the locator, so it can in theory be used for any kind of text.
 - For Viz Trio it is meant to be used to select paths to resources that are not URIs, like the resources on the Viz Graphics Hub that is currently in use.
 - The `type` must match the tabfield type.
 - The value from the first locator that matches will be used.
 - If no locators matches an error message will be logged in Trio.
 - The Viz Engine types are:
- **TEXT** : `text/plain`
 - **RICHTEXT** : `application/vnd.vizrt.viz.richtext+xml`
 - **MATERIAL** : `application/vnd.vizrt.viz.material`
 - **DUPLET** : `application/vnd.vizrt.viz.duplet`
 - **TRIPLET** : `application/vnd.vizrt.viz.triplet`
 - **FONT** : `application/vnd.vizrt.viz.font`
 - **CLOCK** : `application/vnd.vizrt.viz.clockcommands`
 - **GEOM** : `application/vnd.vizrt.viz.geom`
 - **IMAGE** : `application/vnd.vizrt.viz.image`
 - **AUDIO** : `application/vnd.vizrt.viz.audio`
 - **VIDEO** : `application/vnd.vizrt.viz.video`
 - **MAP** : `application/vnd.vizrt.curious.map+xml`
 - **TRIOSCROLL** : `application/vnd.vizrt.trio.scrollelements+xml`

- **Other links:**

```
<link rel="..." href="http://..." type="..." />
```

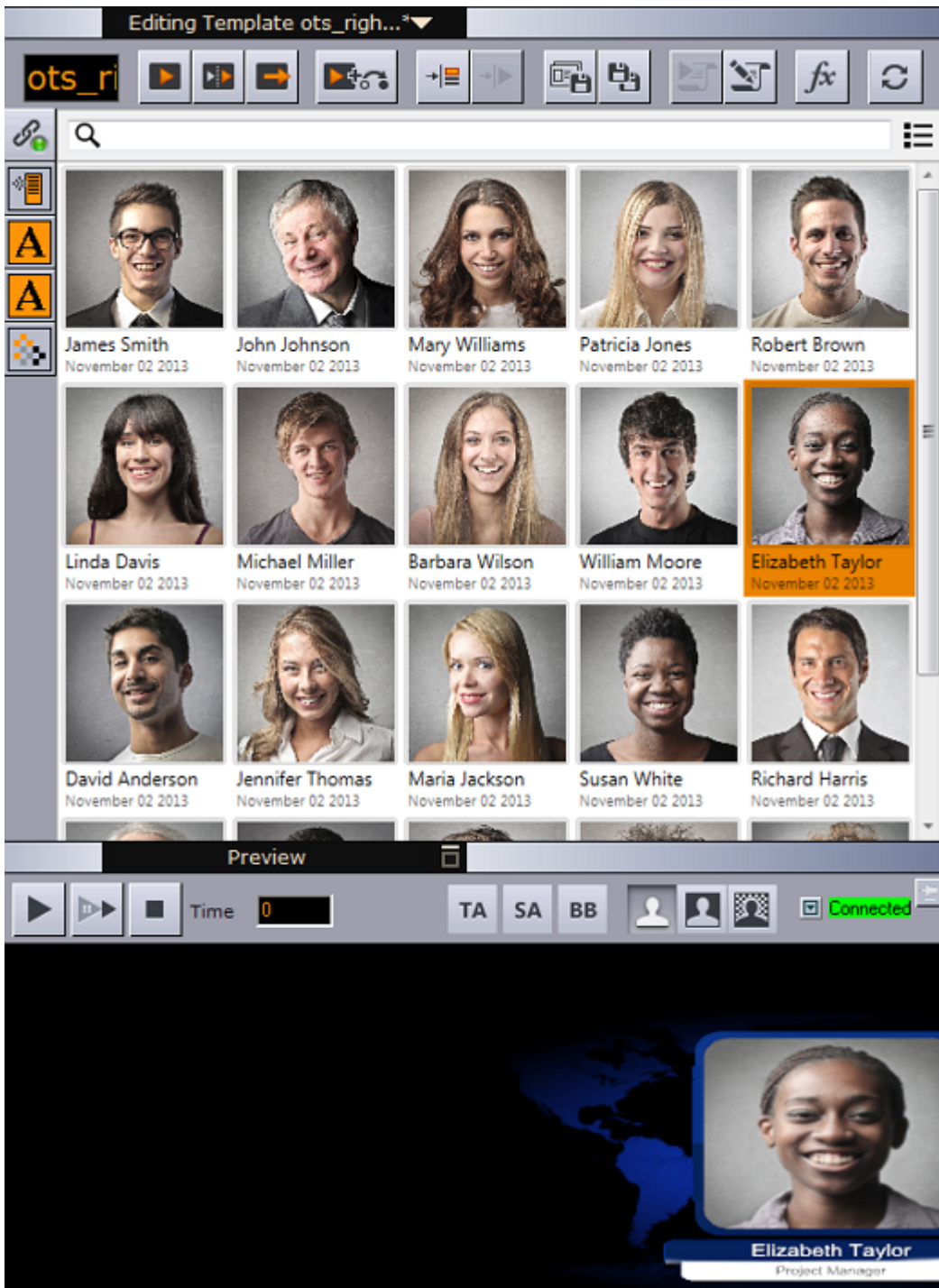
- Required in order to select the URL to external resources. This can be images, videos, text, and so on.

- Limitation 1: The atom 1.0 specs allow multiple links with the same relation, but when linking a tabfield to a link in Trio, only the first link will be selected.
- Limitation 2: Trio can only use the href URL as the value. It cannot download the resource at the URL and use its contents as the value. The Viz Engine control plugin for the property has to be able to understand the URL and download the resource.

5.21.5 Feed Browsing

Given a valid Field Linking URL the elements can now be browsed and selected in the **Feed Browser** window. The example below uses the link:

http://d63hzfu0kdpt5.cloudfront.net/vizrt_training/vizrt\-training\-feed.xml



· **Authentication:**

The feed browser supports basic HTTP authentication. The server must return a HTTP header based on the HTTP standard, like this:

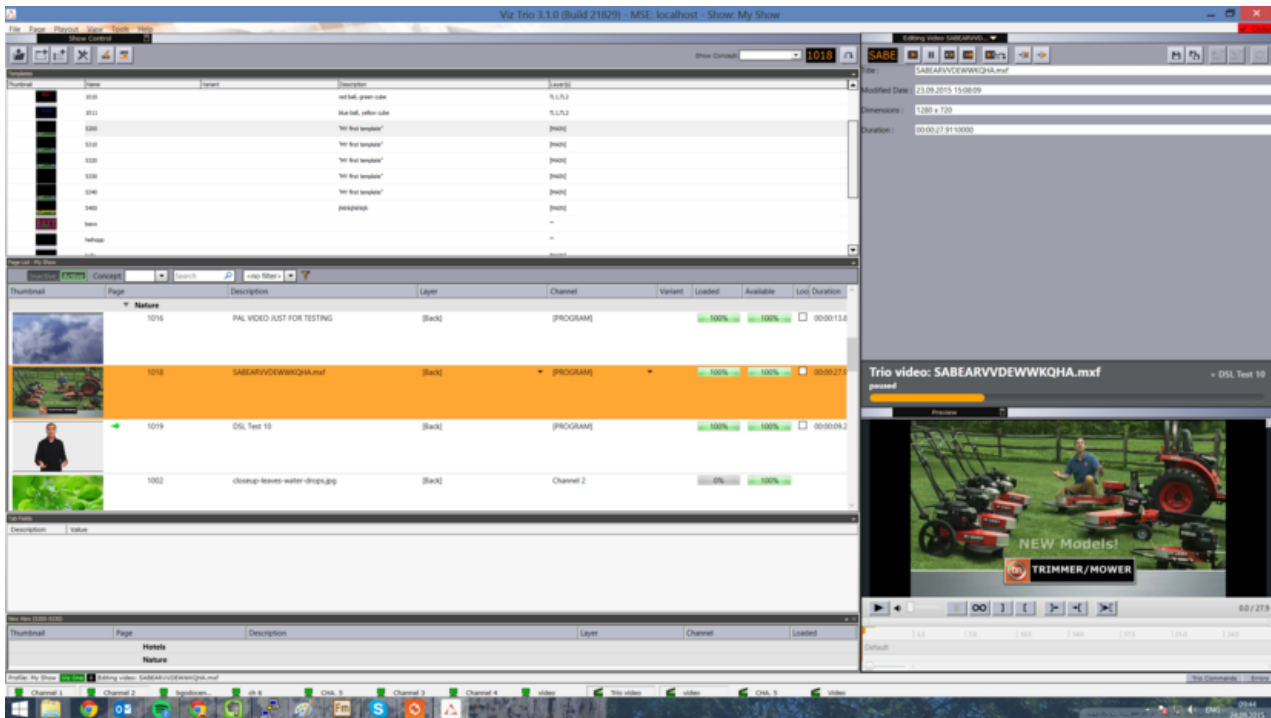
```
WWW-Authenticate: Basic realm="..."
```

- **Searching and Filtering:**

If the feed supports OpenSearch, the search box will be enabled and all searches will be done on the server. If not, the search box will be a local text filter box.

5.22 Timeline Editor

Add graphics to a video timeline with the Timeline Editor.



- Search for a video using the [Search Media](#) editor and open it in the Timeline editor.
- When using data elements (graphics), the Timeline editor will add them as placeholders on the video timeline. The placeholders will display snapshots of the graphics as video overlays (fetched from Preview Server), and can be manually adjusted on the timeline.
- A story item containing clip and graphics references from the Timeline editor appears as a group in the playlist. Issuing a Take command on the group triggers the video clip and timed graphics.

This section covers the following topics:

- [Timeline Editor Functions](#)
- [Working with the Timeline Editor](#)
- [Troubleshooting and Known Limitations](#)

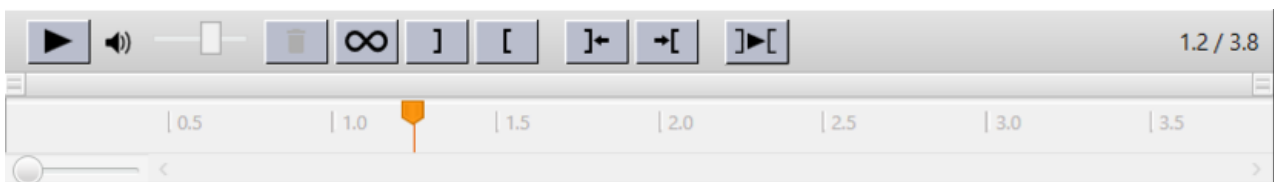
5.22.1 Timeline Editor Functions

This section contains information on the following topics:







- [Editor Control Bar Functions](#)
- [Timeline Editor Functions](#)
- [Timeline Editor Keyboard Shortcuts](#)
- [Inline TimeCode Monitor](#)






Editor Control Bar Functions

The Timeline editor is divided in two; a control bar and a preview window.

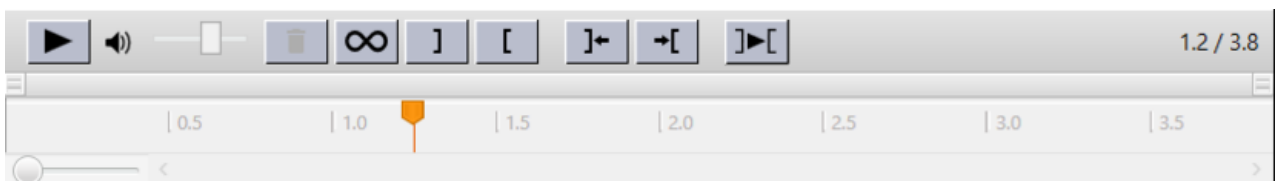


Timeline Editor Functions

Function	Description
	Play or Pause the clip.
	Adjust the volume of the clip.
	Delete the currently selected graphics from the timeline.
	If on, a clip loops between the clip in and the out times. Looping affects the clip only in the playout mode.
	Scrub marker (cursor). Drag to scrub the clip.
	Set the mark-in point to the current position of the cursor.

Function	Description
	Set the mark-out point to the current position of the cursor.
	Jump the cursor to the mark-in point.
	Jump the cursor to mark-out point.
	Play only the area between the mark-in and mark-out points.
	Drag the Zoom control to increase or decrease the time scale shown on the timeline.

Multiple Tracks in the Timeline Editor



- The Timeline editor supports multiple tracks for transition logic scenes. This allows graphics to overlap and play out correctly, but requires that all data elements are based on the same transition logic set.
- The tracks reflect the layers in the transition logic scene. A track is displayed in the Timeline editor if an element using that layer has been added.
- Graphics from the same layer should not be overlapped, and the Timeline editor will indicate a possible conflict by coloring the graphics placeholders red.

Timeline Editor Keyboard Shortcuts

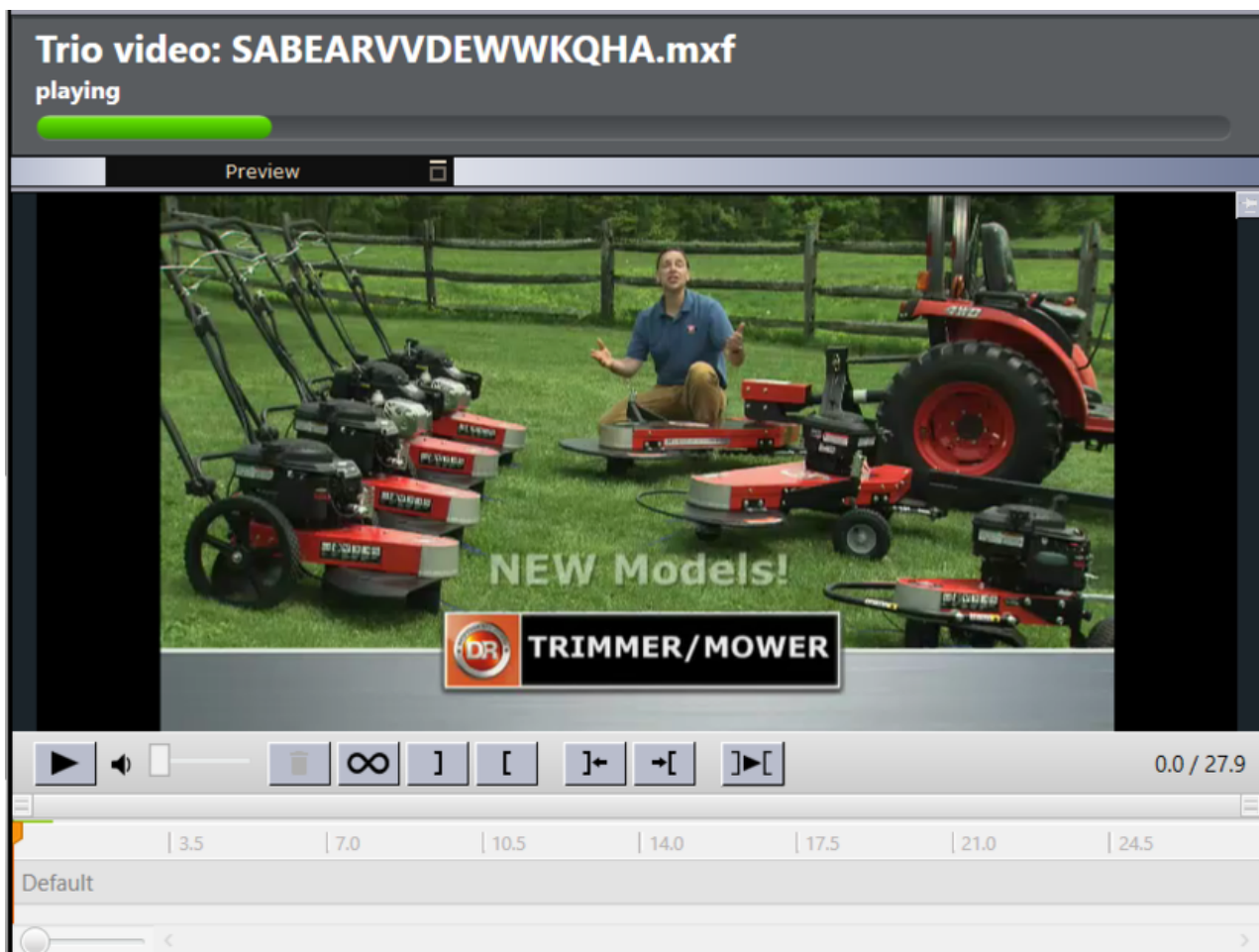
Key	Description
Mouse wheel	Zoom in and out
CTRL + '+'	Zoom in

Key	Description
CTRL + '-'	Zoom out
CTRL + 0	Reset zoom
DELETE	Remove currently selected graphics from timeline
SPACE	Play/Pause
SHIFT + SPACE	Play only the area between the mark-in and mark-out points
CTRL + ARROW	Move graphics in small steps back and forth on the timeline
CTRL + SHIFT + ARROW	Move graphics in big steps back and forth on the timeline
CTRL + ALT + ARROW	Increase or decrease duration by a small amount
CTRL + ALT + SHIFT + ARROW	Increase or decrease duration by a large amount
CTRL + L	Move the selected graphics next to the one to the left
CTRL + R	Move the selected graphics next to the one to the right
CTRL + ALT + L	Stretch the selected graphics next to the one to the left
CTRL + ALT + R	Stretch the selected graphics next to the one to the right
CTRL + H	Sets video position to start of the clip
CTRL + E	Sets video position to end of the clip

Key	Description
CTRL + T	Sets a video to loop. (Note that this shortcut only works if the cursor is first placed in the timeline editor.)

Note: Media keyboard shortcuts for play, pause and mute should also work.

Inline TimeCode Monitor



In addition to the basic [TC Monitor](#) that opens in a separate window, an inline TimeCode Monitor becomes available above the local Preview panel in Viz Trio when playing a video. It displays the channel associated with the video and the video's name in this format: <channel-name>: <clip-name>.

It also displays a progress bar showing how much of the video has played. The progress bar is green when the video is playing, yellow when it is paused and red when 10 seconds remain in the video.

5.22.2 Working with the Timeline Editor

This section covers the following topics:

- [Adding Graphics to a Video Clip Timeline](#)
- Editing a Composite Element
- Setting the Mark-in or Mark-out Point

Adding Graphics to a Video Clip Timeline

1. Drag a video from the [Search Media](#) editor to a playlist.

Note: The video must be dragged to a *playlist* in order to add graphics and create a composite element. Graphics cannot be added to videos that are dragged to a *page list*.

2. Double-click the video item.
 - The item will open in the Timeline editor.
 - If the video does not appear in the Timeline editor as expected, see [Troubleshooting and Known Limitations](#).
3. To add graphics, drag an item from the page list to the timeline. A graphics placeholder will appear on the timeline.
4. Use the mark-in/mark-out points to adjust the duration of the animation.
5. *Optional:* Repeat steps 3 and 4 to add more graphics.



Note: Although graphics may be overlaid on the timeline, graphics that use the same layer (front, middle, back or transition logic layer) will not play out correctly.

6. Click **Save** in the upper right corner to create a new composite element, consisting of both the video and graphics.
 - The composite element is presented in the playlist with the video element as parent, and the graphics items as children.

Editing a Composite Element

1. First, create a composite element, consisting of both video and one or more graphics items, see [To add graphics to a video clip timeline](#).
2. Double-click the composite element in the playlist.
 - The item will open in the Timeline editor.
3. Drag graphics to the timeline.
4. Click **Save as** in the upper right corner to update the composite element.

Setting the Mark-in and Mark-out point

1. In the Timeline editor, scrub the cursor to the desired frame.
2. Click the **Set Mark-in**  or **Set Mark-out**  button.

Click **Jump to Mark-in**  or **Jump to Mark-out**  to view the points.

5.22.3 Troubleshooting and Known Limitations

Video Codecs

If you are previewing proxy versions of video clips from Viz One using the Timeline editor you must install video codecs that are not part of the basic Viz Trio installation. Note that the setup procedures of [Settings](#) are only relevant when using the Timeline editor and not Viz Trio as such. Playout of high resolution versions on Viz Engine does not require these codecs.

Viz Engine Configuration

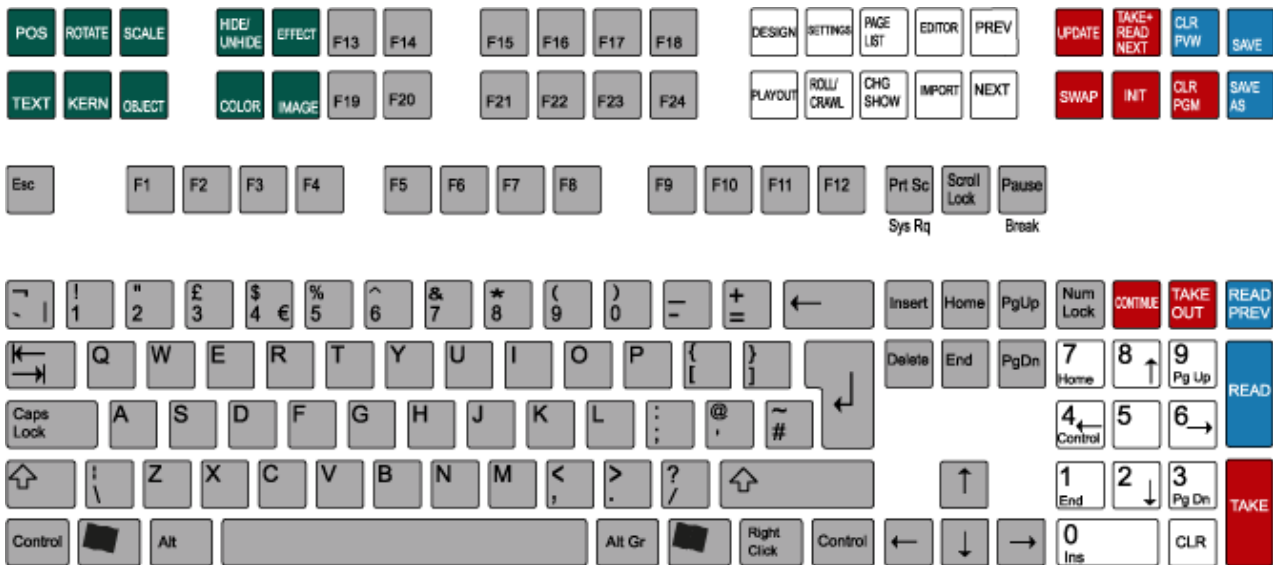
In order to work with video clips from Viz One in the Timeline Editor, Viz Engine must be configured accordingly.

Known Limitations

- Although graphics may be overlaid on the timeline, graphics that use the same layer (front, middle, back or transition logic layer) will not play out correctly. The Timeline editor will always indicate a possible conflict by coloring the graphics placeholders red. To use overlapping graphics based on transition logic scenes, see [Multiple Tracks in the Timeline Editor](#).
- Current limitations on the Media Sequencer will cause the following behavior:
 - a. If you issue a `Take` on a timed group (1), and while that group is being played out issue a `Take` on graphics in another layer (2), the Media Sequencer will instead take out the last taken element (2) when issuing the timed `Take Out` command (for the timed graphics).
 - b. Hence, it is currently not recommended to take other data elements on-air while a timed group is being played out.
- The Timeline editor may behave unexpectedly in some virtual machine or remote desktop environments.

5.23 Viz Trio Keyboard

Viz Trio is designed to be operated with a mouse and keyboard or by keyboard only. An overview of the Viz Trio keyboard is presented below. If you are using a Cherry keyboard, see [Cherry Keyboard](#).



The keyboard contains two rows with extra function keys for 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 these settings usually do not need to be changed.

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

5.23.1 Editing Keys



Green keys perform editing operations.

- **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.

Note: The current tab field must have the property of the key exposed for editing. If not, the key has no effect and an error message is written to the log file when the key is pressed.

5.23.2 Navigation Keys



Use white keys to switch between different views and editors in the program.

- **DESIGN:** Opens Viz Trio in design mode displaying the designer tool UI.
- **PLAYOUT:** Opens Viz Trio in playout mode displaying the playout UI.
- **SETTINGS:** Displays the Show Properties 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 [Add Page List View](#).
- **NEXT:** If extra page views are defined, this key shows the view below the one currently active.

5.23.3 Program and Preview Keys



Red function keys affect actions on the program channel. The blue keys all affect actions on the preview channel.

- **UPDATE:** When a change has been made 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 on the list.
- **SWAP:** The swap key takes to air what is currently read and visible on the preview, and takes off air what's 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, pressing **Continue** will make the animation continue.
- **TAKE OUT:** If transition logic is used, **Take Out** will take out any page loaded in the layer that is currently read. If transition logic is not used, **Take Out** 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 on 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.

6 Macro Commands And Events

This section contains reference information on the following topics:


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

See Also

- [Scripting](#)
 - [Macro Language](#)
-

6.1 Events

There are two types of events in scripts: *template scripts* with **template** events and *show scripts* with **show** events. Although they look similar, they are not the same. When used in scripts, the events will be called only if the specific event described by the event-handler occurs.

 **IMPORTANT!** Template scripts are loaded only when a page/template is *read*. Some template events are triggered only if the template or a page using this template is read.

This section contains reference information on the following topics:

- [Template Event Callbacks](#)
- [Show Event Callbacks](#)
- [Using the OnDelete Script Event](#)

6.1.1 Template Event Callbacks

- **OnContinue()**: Called when the Continue button is pressed or a continue macro command is executed. Only triggered if the template or a page using this template is read.
- **OnInitialize**: Called when the Initialize button is pressed or an initialize macro command is executed.
- **OnBeforeSave()**: This procedure is called before Trio saves the current page. The script can cancel the saving by returning false (`OnBeforeSave = false`).
- **OnSave**: Called when the Save button is pressed or a save macro command is executed.
- **OnBeforeTake()**: This function is called before Trio runs take on the current page. The script can cancel the take action by returning false (`OnBeforeTake = false`). Only triggered if the template or a page using this template is read.
- **OnTake()**: Called when the Take button is pressed or a take macro command is executed. Only triggered if the template or a page using this template is read.
- **OnTakeOut()**: Called when the Take Out button is pressed or a take out macro command is executed. Only triggered if the template or a page using this template is read.
- **OnUpdate()**: Called when an update action is executed in Viz Trio (See keyboard shortcut list on control page).

- **OnClick():** 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.
- **OnSocketDataReceived:** Called when socket data is received on the defined port. A socket object must be configured on the configuration page.
- **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.
- **OnShowVariableChanged(Name, Value):** This procedure is called when a global variable is changed or added. All Trio clients (including the client where this command was issued) that have the same show open will receive the event when the command `show:set_variable` is used .
- **OnGlobalVariableChanged(Name, Value):** This procedure is called when a show variable is changed or added. All Trio clients that are connected to the same Media Sequencer with the same show open will receive the event when the command `trio:set_global_variable` is used.

6.1.2 Show Event Callbacks

- **OnShowOpen():** Called when show is opened.
- **OnShowClose():** Called when show is closed.
- **OnRead():** Called when the Read button is pressed or a read macro command is executed.
- **OnRead(PageName):** Called when a specific page is read.
- **OnContinue(PageName):** Called when the Continue button is pressed or a continue macro command is executed.
- **OnInitializeShow():** same as OnInitialize for template scripts
- **OnCleanupRenderers():** Called when the `trio:cleanup_renderers` or `trio:cleanup_all_channels` commands are executed.
- **OnBeforeSave(PageName):** Called before a specific page is saved.
- **OnSave(PageName):** Called when a specific page is saved.
- **OnBeforeTake(PageName):** Called before a specific page is taken.
- **OnTake():** Called when the Take button is pressed or a take macro command is executed.
- **OnTake(PageName):** Called when a specific page is taken.
- **OnTakeOut:** Called when the Take Out button is pressed or a take out macro command is executed.
- **OnTakeOut(PageName):** Called when a specific page is taken out.
- **OnUpdate(PageName):** Called when a specific page is updated.
- **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.
- **OnSocketDataReceived(data):** Called when data received on socket.
- **OnPropertyFocused(PageName, PropertyName):** Called when property focused.

- **OnCopy(OldName, NewName):** Called when a page is copied. The event is only available to show scripts.
- **OnCopy(OldName, NewName):** Called when page copied from old to new.
- **OnMove(OldName, NewName):** Called when a page is moved or renamed. 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](#)).
- **OnShowVariableChanged(Name, Value):** Called when show variable changed.
- **OnGlobalVariableChanged(Name, Value):** Called when global variable changed.

! **IMPORTANT!** An event with the same name but different argument value(s) are *not* equal. For example `OnRead()` and `OnRead(Pagename)` are not the same event.

Using the OnDelete Script Event

For example, 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 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](#)
- [Script Editor](#)

6.2 Macro Language

Use the Viz Trio macro language to script many of the operations that are normally done in the UI. 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](#)
- As part of an external application where commands are executed over a socket connection.

When macros are used, it's often desirable to use commands that normally would trigger a dialog for user input such as *page:delete 1000*, which would normally ask the user for confirmation before deleting. To avoid this, you can set the following modes:

- **gui:set_silent_mode:** controls whether to show dialogs to the user.
- **gui:set_interactive_mode:** controls whether to show dialogs to the user.

This section covers the following topics:

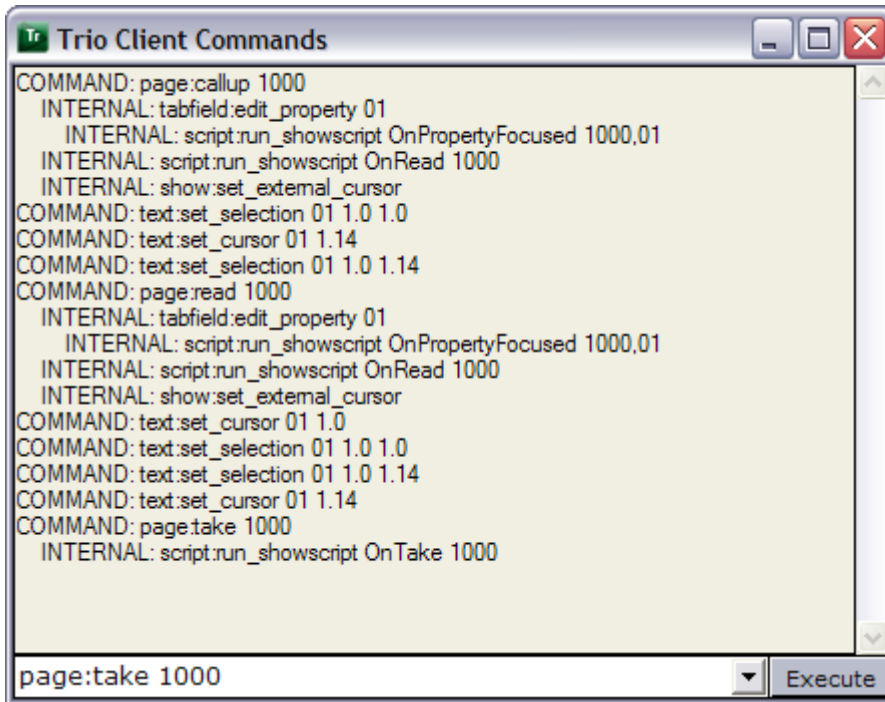
- [Working with Macro Commands](#)

6.2.1 Working with Macro Commands

This section covers the following topics:

- [Trio Client Commands Window](#)
- [Enabling the Trio Client Commands Button](#)
- [Opening the Trio Client Commands Window](#)
- [Show, Context and Tab-field Commands](#)

Trio Client Commands Window



The Viz Trio user interface generally uses the same macro commands as the scripting support uses. To find commands that are executed when different user interface operations are performed, use the commands window to see 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 and click the **Execute** button to run the command to test your customized command.

Enabling the Trio Client Commands Button

Click Viz Trio's **Config** button, and in the **User Interface > User Restrictions** section, clear the **Call Viz Trio Commands** check box.

Opening the Trio Client Commands Window

Click the **Trio Commands** button in the lower-right corner of the application window.

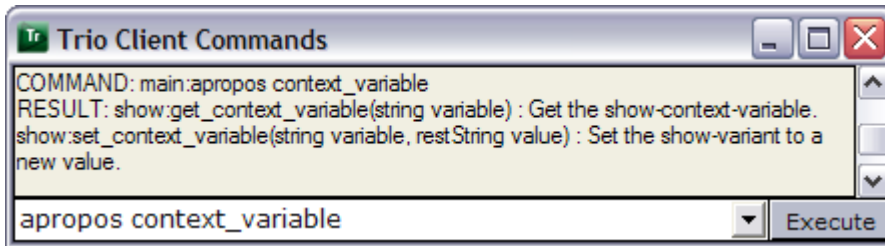
Apropos and Help Commands

Use Trio Client Commands window features when searching for information about a command.

The keywords **apropos** and **help** can be used to perform look-ups in the available set of commands.

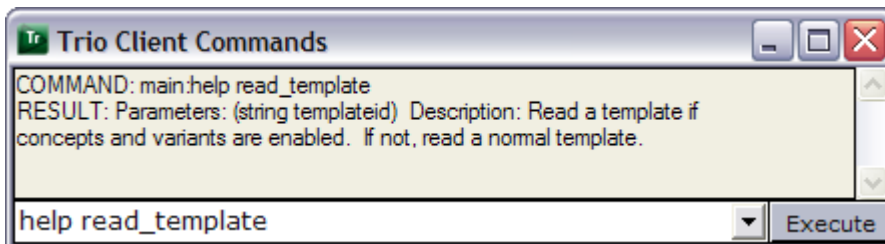
- **apropos**: Searches command names and descriptions for a string.
- **help**: Shows help for the specified command, or lists commands.

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.

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 Media Sequencer can set and get the variables, and any script within a show can access the variable.

This section covers the following topics:

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

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:** Sets the value of a show variable.
- **show:get_variable:** Gets 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_variabl
e 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 context 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.
- **page:arm_copy <page-name> [<channel-name>]:** Arms a copy of a page for firing. If channelName is empty, the default channel is used. If the original page is changed or deleted after arming, it will not affect the armed page.
- **page:arm_copy_of_current [<channel-name>]:** Arms a copy of the current page for firing. If channelName is empty, the default channel is used. If the original page is changed or deleted after arming, it will not affect the armed page.
- **page:arm_copy_of_selected [<channel-name>]:** Arms a copy of the selected page on its assigned channel or on the given channel, if defined. If the original page is changed or deleted after arming, it will not affect the armed page.

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)
- `show_custom_property_editor_for_current_tabfield`

Example commands:

```
COMMAND: tabfield:set_custom_property 01 string COMMAND: tabfield:get_custom_property
01 RESULT: string
```

Commands

The commands are grouped as follows:

- [Channelcontrol](#)
- [dblink](#)
- [Gui](#)
- [Macro](#)
- [Main](#)
- [Page](#)
- [Playlist](#)
- [Proxy](#)
- [Rundown](#)
- [Script](#)
- [Scroll2](#)
- [Settings](#)
- [Show](#)
- [Sock](#)
- [Tabfield](#)
- [Table](#)
- [Text](#)
- [Trio](#)
- [Util](#)
- [Viz](#)
- [vtwtemplate](#)

Channelcontrol

Command Name and Arguments	Description
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.
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) @TODO check args	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.
list_video_channels	Returns all video channels of the current profile as comma separated list.
list_viz_channels	Returns all Viz channels of the current profile as comma separated list.
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.

Command Name and Arguments	Description
set_asset_storage (string host-name, restString storage-name)	Sets the publishing point to be used for all handlers in the active profile. The storage name is the user visible title for the asset storage.
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.
wait_vizcmd_to_channel(string channel, restString cmd)	Sends a Viz Engine command to a named channel and waits for the returned answer.

dblink

Command Name and Arguments	Description
<p>link_database (string property key, string connection name, string table name)</p>	<p>Links a property of a page to a table of a database connection. Example 1: <code>TrioCmd("dblink:link_database 1 ExcelTest "'[Scalar Tests\$]'"")</code> Example 2: <code>{{TrioCmd("dblink:link_database 1 ExcelTest "'[Table Tests\$]'"")}}</code> 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. ExcelTest refers to the name of the configured database connection made in the Database Config window. (See To create a new database connection). Scalar Tests and Table Tests refers to the table that is chosen in order to select the lookup column.</p>
<p>map_database_column (string table key, string property column key, string database field name)</p>	<p>Maps a column of a table property to a database field. Example: <code>TrioCmd("dblink:map_database_column 1 3 Name")</code>. 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. 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. Name refers to the column the data should be read from.</p>
<p>select_value (string property key, string value column, string key column, string key value)</p>	<p>Selects a single value (scalar linking) of a database and links it to the property. The value is selected from the <value column> where the <key column> matches the <key value>. Example: <code>TrioCmd("dblink:select_value 1 Color Key 2")</code> 1 refers to the Control Text's Field identifier property that in Viz Trio can be identified as the template's tab field 1. Color refers to the column the data should be read from. Key refers to the lookup column defined in the Database Config window. (See To create a new database connection). 2 refers to the row the data should be read from.</p>

Command Name and Arguments	Description
store_to_database (string property key)	Stores a linked property of a page to its value in the database. Example: <code>TrioCmd("dblink:store_to_database 1")</code> 1 refers to the Control Text's Field identifier property that in Viz Trio can be identified as the template's tab field 1.
update_from_database (string property key)	Updates a linked property of a page with its value in the database. Example: <code>TrioCmd("dblink:update_from_database 1")</code> 1 refers to the Control Text's Field identifier property that in Viz Trio can be identified as the template's tab field 1.
use_custom_sql (string property key, restString custom SQL)	Uses a custom SQL clause to select the database value. An empty SQL clause disables the usage of custom SQL. Example: <code>TrioCmd("dblink:use_custom_sql 1 SELECT Text FROM ['Table Tests\$']")</code> 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. Text refers to the column the data should be read from. Table Tests refers to the database table (i.e. a single spreadsheet in Microsoft Excel).

Gui

Command Name and Arguments	Description
add_view (title min max elementTypeList)	Creates a new view. If called without any parameters, the PageView dialog will be shown. Min and max may optionally be specified to filter elements with numeric names. Elements with non-numeric names will always be included. elementTypeList is an optional space-separated list of types of elements to include in the view. Possible values are: pages stills videos empty_groups. If none are specified, all will be included. Example 1: <code>gui:add_view "The Title" 2000 3500 pages stills</code> . This will show pages and stills in the range between 2000 and 3500. Videos will not be shown and empty groups will be hidden. Example 2: <code>gui:add_view title2 videos</code> . This will show all videos, nothing else.

Command Name and Arguments	Description
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.
change_path	Opens the browser for either shows or external playlists.
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.
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	This command has been removed and replaced by Ctrl+f and the search box.
focus_next_view	Activate the next page view

Command Name and Arguments	Description
focus_prev_focused_view	Focus the previous focused view.
focus_prev_view	Activate the previous page view
get_callup_code	Gets the value of the callup code window.
get_selected_pages	Retrieves the currently selected pages as a space separated list.
goto_config	Opens Viz Trio's Configuration Window.
hide_timecode_monitor	Hides the TimeCode monitor. See also show_timecode_monitor .
hit_test(x:integer, y:integer)	Test if a tab field exists on the X,Y coordinate.
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(rest String destnumber)	Move the selected pages to number or show a dialog if no number is supplied.
move_selected_pages_with_offset(rest String destnumber)	Move the selected pages with an offset or show a dialog if no number is supplied.
open_playlists	Shows the directory selector for the external playlists. Typically MOS playlists.
open_shows	Opens the show browser.
open_effect_dialog	This command opens the effect dialog. Will not open if current page is not set.
play_video	Play video in video preview

Command Name and Arguments	Description
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.
search_video (restString search_text (optional video id))	Search for video.
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.
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_font (restString fontname)	Set the default GUI font

Command Name and Arguments	Description
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.
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.
set_mam_service_document_uri [uri [username password]]	The parameters are optional. If no parameters are provided the command should disable the Viz One configuration. Otherwise it enables the configuration if necessary and sets the new URI and possibly also the credentials if those parameters are provided.
set_media_search_credentials username password	This command changes the credentials used for when doing media searches on Viz One.
show_viz_preview	Show Viz preview
show_custom_prop_column(boolean value)	Shows or hides custom property column in tab field list.
show_custom_property_editor_for_current_tabfield	Opens the inplace editor for the custom property of the current tab-field if the custom column in the tab-field tree is visible.
show_export_dialog	Show the export show dialog.

Command Name and Arguments	Description
show_export_selected_page_dialog (string showPath)	Display the selected pages export show dialog. <i>showPath</i> parameter is optional. It is almost same as the Export Show feature but it will export only selected pages & all it's associated items.
show_external_cursor(boolea value)	Show or hide the external cursor in the page view.
show_feed_streamer_moderation	Opens the feed streamer moderation tool if Viz SocialTV is installed.
show_import_dialog	Show the import show dialog.
import_viz_archive	Initiate the viz archive import process. This triggers the file open dialog to select the Viz Engine archive (.via) file from disk.
show_pageeditor	Shows the page editor.
show_pages_in_rundown(boolea doShow)	Shows or hides pages in the rundown for the current show.
show_pageview	Show the page view.
show_playlist (string refPath, boolea requestMos, restString description)	Shows the playlist with the specified path or name. Can choose to request a MOS playlist. Description is optional and defaults to the playlist name.
show_scripteditor	Shows the script editor.
show_scroll_template_creator	Shows the scroll template creator view.
show_settings	Open the show properties view
show_search_media	Shows the search media view.
show_search_video	Shows the search video view.
show_shortcut_list	Show the view of local shortcuts and macros.

Command Name and Arguments	Description
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 hide_timecode_monitor .
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.
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_nameless_groups(visible: boolean)	Toggle hide/show groups with no name.

Macro

Command Name and Arguments	Description
export_macros_to_xml(string filename)	Exports the macro commands to a XML file.

Main

Command Name and Arguments	Description
<code>apropos(search:string)</code>	Search command names and descriptions for a string.
<code>call_macro(restString name)</code>	Call (invoke) a named macro.
<code>define_local_macro(string name, restString commands)</code>	Define a named, local macro.
<code>define_local_script(string name, restString commands)</code>	Define a named, local script.
<code>define_macro(string name, restString commands)</code>	Define a named, global macro.
<code>define_script(string name, restString commands)</code>	Define a named, global script.
<code>delete_global_macro(name:string)</code>	Delete a global macro.
<code>delete_local_macro(name:string)</code>	Delete a local macro.
<code>get_global_macros</code>	List the names of global macros (separated by new lines).
<code>get_local_macros</code>	List the names of local macros (separated by new lines).
<code>get_shows</code>	Get the names of all the shows in the client
<code>get_version</code>	Get a version string.
<code>help(command:string)</code>	Shows help for the specified command, or lists commands.

Command Name and Arguments	Description
macro	

Page

Command Name and Arguments	Description
arm(pageName:string, channelName:string)	Arms the specified page on the specified channel. If channelName is empty, the default channel is used.
arm_current(channelName:string)	Arms the currently read page on its specified program channel. If channelName is empty, the default channel is used.
arm_selected	Arm the selected page.
cancel_refresh_linked_data(page name (optional))	Cancel refreshing of linked data, started with the 'page:refresh_linked_data_async' command.
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_thumbnail_uri (string pageName, restString newThumbnailURI)	Changes the thumbnail URI for the referenced page. The URI and the page name must be valid.
change_variable(string variable, restString value)	Change the variant for the currently edited page.

Command Name and Arguments	Description
<code>continue(pageName, channelName)</code>	Only applicable to videos. Continues a video on a named channel. If channel is not mentioned it plays the video on its designated channel. If the clip name is not mentioned the current clip is continued. Such as, <code>page:continue 1000 A</code> continues the clip "1000" on channel "A" while just <code>page:continue 1000</code> would continue clip "1000" on its designated channel.
<code>cue(pageName, channelName)</code>	Cues a page on a named channel. If channel is not mentioned it cues the page on its designated channel. If the page name is omitted the current page is cued. Such as, <code>page:cue 1000 A</code> cue the page "1000" on channel "A" while just <code>page:cue 1000</code> would cue page "1000" on its designated channel.
<code>cue_append(pageName, channelName)</code>	Only applicable to videos. The command interrupts the current playing clip and immediately starts playing the selected video on the named channel. Such as, <code>page:cue_append 1000 A</code> cue the video "1000" on channel "A" while just <code>page:cue_append 1000</code> would cue video "1000" on its designated channel.
<code>cut(pageNumber, channelName)</code>	Cuts a page on the named channel. If channel is omitted it cuts the page on its designated channel. Such as, <code>page:cut 1000 A</code> cuts the page "1000" on channel "A" while just <code>page:cut 1000</code> would cut page "1000" on its designated channel.
<code>delete_page</code>	Delete selected page(s) in active view.
<code>delete_pagerange(int start, int end)</code>	Deletes the page inside from the begin range to (including) the end range parameter.
<code>direct_continue</code>	Continue selected page in active view.
<code>direct_cut</code>	Cut selected page in active view.
<code>direct_take</code>	Take selected page in active view.
<code>direct_takeout</code>	Take out selected page in active view.

Command Name and Arguments	Description
direct_cue	Cue the selected video on the video program channel
direct_cue_append	Cue append the selected video on the video program channel
direct_pause	Pause the currently playing video on the video program channel
export_property(string key, string filename)	Export a property value to file (UTF-8 encoded).
fire(channelName:string, action:string, clearAfterFire:boolean)	Fire a channel with the given action (take, continue, out, and so on). If clearAfterFire is true, the armed element on the channel is un-armed after fire.
fire_all(action:string, clearAfterFire:boolean)	Fire all armed channels with the given action (take, continue, out, and so on). If clearAfterFire is true, the armed elements on the channels are un-armed after fire.
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 the value of the specified tabfield property.
get_tabfield_names (optional: pageName)	Retrieves the tab-field names of the given page or the current page if no page is specified. The returned list is space separated.

Command Name and Arguments	Description
get_tabfield_count	Gets the number of tabfield 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.
get_property_keys (pageName optional)	Retrieves the property keys of the given page or the current page if no page is specified. The returned list is space separated.
get_variable(string variable)	Retrieves the value of a variable of the currently read 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.
pause(pageName, channelName)	Only applicable to videos. Pauses a named video on a named channel. If channel is not mentioned it pauses the element on its designated channel. If the video name is not mentioned then current video is paused. Such as, page:pause 1000 A pause the video "1000" on channel "A" while just page:pause 1000 would pause video "1000" on its designated channel.

Command Name and Arguments	Description
prepare(pageName, channelName)	Prepares a named element on a named channel. If channel is not mentioned it prepares the element on its designated channel. If the element name is not mentioned the current element is prepared. Such as, page:prepare 1000 A prepare the page "1000" on channel "A" while just page:prepare 1000 would prepare page "1000" on its designated channel.
prepare_current_on_channel <channel-name>	Prepares the currently read element to a channel other than the specified channel for that element.
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.
read(pageNr:integer)	Reads the specified page number.
read_parentpage	Read the parent of a sub page.
read_rootpage	Read the root page in a page hierarchy.
read_subpage(string string)	Read a sub page of another page, for instance a scroll element. Argument format: "tabfield key/pag number", eg. "1/1003".
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)	Read a master template with context variables that can override the show properties. 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.

Command Name and Arguments	Description
refresh_linked_data(page name (optional))	Refresh linked data if the page has any tabfields that are linked to data from an external source.
refresh_linked_data_async(page name (optional))	Start refreshing linked data if the page has any tabfields that are linked to data from an external source. The refresh can be cancelled with the 'page:cancel_refresh_linked_data' command.
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(integer)	Goes in saveas mode or saves page to page given as argument. 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_effect(restString effect)	Sets the transition effect to be used between (scene-based) pages. Empty string specifies hard cuts (default).
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. Restarting Trio will reset expert mode.

Command Name and Arguments	Description
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 (pageName, channelName)	Plays the named element on its designated channel. If channel is not mentioned it plays the element on its designated channel. If the page name is omitted then the current page is played. Such as, page:take 1000 A takes the page "1000" on channel "A" while just page:take 1000 takes the page "1000" on its designated channel.

Command Name and Arguments	Description
take_exchange	Take current page and read the last page that was taken.
take_with_variables(restString variables)	Take a page with variables.
take_with_variables_on_channel	Take a page with variables to a specific channel.
takenext	Take current page and read next page in active view.
takeout(pageName, channelName)	Takes out the named element from the named channel. If channel is not mentioned it takes out the element from its designated channel. If the element name is not mentioned then the current element takes out from the designated channel. Such as, page:takeout 1000 A takes out the page "1000" on channel "A" while just page:take 1000 would take out page "1000" from its designated channel.
unarm(channelName:string)	Unarms the specified channel.
unarm_current	Unarms the currently read page on the channel it is armed.
unarm_all	Un-arms all armed pages.
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.

Playlist

Command Name and Arguments	Description
<code>add_page(string pagename, restString duration)</code>	Add a page to the end of the playlist. The duration argument is optional. If not given, the page duration will be used.
<code>create_playlist(string name)</code>	Adds a new empty global playlist and returns the path to the new playlist. The name of the playlist is optional. If not specified, a dialog will pop up to ask for it. The new playlist will show up in the "playlists" in the "Change Show" dialog.
<code>close (restString: id)</code>	Closes a specified playlist. The id parameter is either the path or the unique identifier of the playlist to close. If omitted the current playlist is closed.
<code>copy_to_show (restString start_num)</code>	Copy the elements in the playlist to the current show
<code>clear (restString id)</code>	Clears the specified playlist. The id can be the playlist path or its identifier. If omitted the current playlist gets cleared.
<code>direct_take_out_selected</code>	Take out the currently selected playlist element.
<code>direct_take_selected</code>	Take the selected playlist item onair without previewing it.
<code>get_concept</code>	Retrieves the concept of a playlist.
<code>get_current</code>	Retrieves the path of the playlist which is the current one. May return an empty string.
<code>get_playlists</code>	Retrieves a list of all open playlists in the current show as space separated list of playlist references which can be used to refer to playlists in other commands like <code>playlist:set_current</code> .
<code>go_to_next_story</code>	Select the next story (group) in the active playlist.

Command Name and Arguments	Description
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.
reload_thumbnails (restString: template_names)	Reload the thumbnails for the given templates on all open playlists. Reloads all thumbnails if no templates are given.
run_selected_with_custom_action (string Action)	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. Example: <pre>playlist:run_selected_with_custom_action "prepare"{{playlist:run_selected_with_custom_action "take"}} </pre> 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.
select_first	Select the first element in the playlist.

Command Name and Arguments	Description
select_next	Select the next element in the active playlist.
select_next_by_description(restString phrase)	Selects the next element in the active playlist which has the given phrase in its description.
set_channel_on_selected (restString channelName)	Set the channel for all selected items in the current playlist
set_current (string: id)	Defines the specified playlist as the current one for operations like add_item. The id can be the playlist path or its identifier.
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(restString value)	Set the concept for a playlist.
set_filter (restString: filterName)	Set the filter of the currently active playlist. The filter must already exist. Calling with no filter name will reset the filter of the playlist.
set_looping(restString looping TRUE or FALSE)	Set to true if the carousel shall be looping.
start_carousel (playlist name or id (optional, defaults to active playlist))	Run the elements in the playlist as a carousel.
stop_carousel (playlist name or id (optional, defaults to active playlist))	Stop a playlist running as a carousel.

Proxy

Command Name and Arguments	Description
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.
set_viz_port(int port)	Set the port the proxy should use to access Viz.

Rundown

Command Name and Arguments	Description
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)	Reloads the thumbnails for all listed scene paths. If the scene paths are not given all thumbnails are reloaded.

Script

Command Name and Arguments	Description
eval(restString scriptExpression)	Evaluate the VBScript expression given as a parameter, and return the value converted to a string.
execute_script	Tries to run the OnUserClick event on the current template script. Doesn't report any errors e.g. if the event is not handled in the template script.

Command Name and Arguments	Description
get_showscript_name	Gets the name of the current show script. Specify full name, does not add .vbs to the script name
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.

Scroll2

Command Name and Arguments	Description
add_easepoint(string pagename)	Add an easepoint to the current scroll.
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 parameters.
edit_easepoint(string subpage, string easepoint)	Edit an easepoint in the current scroll.

Command Name and Arguments	Description
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.
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).

Command Name and Arguments	Description
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.

Settings

Command Name and Arguments	Description
apply_gateway_settings	Applies all MOS gateway settings which have been set up previously.
get_setting(string path)	Gets a Viz setting.
get_global_setting (string: path)	Get a global setting.
get_vcp_database_connection_string	Get VCP database connection string
get_vcp_database_schema	Get VCP database schema

Command Name and Arguments	Description
export_to_file (boolean: include_settings, boolean: include_profiles, restString filename)	Export settings to a file. If filename is empty, a file dialog will be shown.
enable_gateway (boolean: flag)	Enables or disables the MOS gateway according to the given flag. After running this command settings:apply_gateway_settings have to be run in order to put the setting operative.
import_from_file (restString: filename)	Export settings to a file using old format for import on Trio 2.6 and earlier. If filename is empty, a file dialog will be shown.
import_from_file (restString filename)	Import settings from a file. If filename is empty, a file dialog will be shown.
start_gateway	Starts the MOS gateway handler of the Viz Media Sequencer Engine.
restart_gateway	Restarts the MOS gateway handler of the Viz Media Sequencer Engine.
set_gateway_host (string hostname [<port_nummer>])	Changes the MOS gateway host name and port. <host-name> must be a valid host name or IP address where the gateway is running. <port-number> must be a valid port number. It is optional and defaults to 10640. After running this command settings:apply_gateway_settings have to be run in order to put the setting operative.
set_gateway_ncs_id (string: id)	Changes the gateway NCS ID. After running this command settings:apply_gateway_settings have to be run in order to put the setting operative.
set_gateway_mos_id (string: id)	Changes the gateway MOS ID. After running this command settings:apply_gateway_settings have to be run in order to put the setting operative.

Command Name and Arguments	Description
set_gateway_messages_per_second (int: seconds)	Changes the number of messages per second to process by the MOS gateway. A zero value disables this setting. After running this command settings:apply_gateway_settings have to be run in order to put the setting operative.
set_vcp_database (string: connection_string, string: Schema)	Set VCP database connection setting
set_global_setting	Sets a global Viz setting (e.g. settings:set_global_setting encoding ISO-8859-1)
set_setting(string path, restString value)	Set a show setting.
set_send_item_stat (boolean: flag)	Enables or disables the item stat sending of the MOS gateway according to the given flag. After running this command settings:apply_gateway_settings have to be run in order to put the setting operative.
stop_gateway	Stops the MOS gateway handler of the Viz Media Sequencer Engine.

Show

Command Name and Arguments	Description
append_pages_from_xml(boolean showDialogs, string filename)	Loads pages from XML and replaces any pages that will be overwritten.
copy_pages (restString: pageList)	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.

Command Name and Arguments	Description
create_combo_template (string: name, string: pagelist)	Create a combo mastertemplate
create_group(string groupname)	Create a page group in the current show, before the focused node.
create_playlist(restString name)	Adds a new playlist to the show and returns the path to the new playlist. If name is omitted, a dialog will pop up asking for a name.
create_show(restString showPath)	Create a new show. The show path may specify a show within a show folder.
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.
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.

Command Name and Arguments	Description
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.
get_templates	Returns a list of templates in the current show.

Command Name and Arguments	Description
get_timing_handler	Returns the timing handler for the current show.
get_variable(string propName)	Get the value of a show variable.
get_variant	Get the show-variant.
goto_viz_dir(path:string)	Change to a show with the same name as a viz directory, creating it if necessary.
import(scenes:string)	Shows the Import Scenes pane or imports specified scene(s).
import_and_initialize(scenes:string)	Import scenes to templates, then reload on external renderers.
import_no_gui(restString sceneNames)	Imports the scenes given as parameter without showing/closing the import GUI.
import_pages_from_xml(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.
import_recursively(string path)	Import all the scenes recursively from this folder.
import_setting(string name, restString value)	Configure an import setting, use this before import_show.
import_show(filename:string)	Import a show (Viz Trio archive).
load_armed	Load all armed elements on their channels.
load_locally	Loads all scenes referenced by pages in the current show on the local viz preview.

Command Name and Arguments	Description
load_show	Load show (scenes, textures, fonts) on external output channels. Replaces the old initialize command.
load_show_on_renderers(string ProgramChannelName, restString PreviewChannelName)	Load current show on the specified channels. First argument is treated as a program channel, second as preview.
move_pages_to_group(string group, restString pages)	Move pages into the specified group. If the group name is empty, move to top level.
move_pages_to_number(int destination, restString CommaSeperatedList)	Move the pages to 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".
reimport_templates (string: templateList)	Reimport the given templates
reload_current_show	Reloads the current show
reload_on_renderers (string: template_l)	Reload (re-initialize) graphics for the specified master templates or scene info on program and preview channels.
load_on_renderers(restString pages)	Load (initialize) graphics for the specified pages on program and preview channels.
rename_page(string old_name, restString new_name)	Rename a page name in the opened show

Command Name and Arguments	Description
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_scene_effect (restString: effect)	Sets the default transition effect to be used between (scene-based) pages. Empty string specifies hard cuts (default).
set_default_video_effect (restString: effect)	Sets the default transition effect to be used between videos and stillstore images. Empty string specifies hard cuts (default).
set_default_viz_path(path:string)	Set the default viz scene path for the current show.
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.

Command Name and Arguments	Description
set_filter	Set the filter of the page list or the currently active page view. The filter must already exist. Calling with no filter name will reset the filter of the page view.
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.

Socket

Command Name and Arguments	Description
connect_socket	Explicitly connect (open) the socket, if not already connected.
disconnect_socket	Disconnect (close) the socket, if connected.

Command Name and Arguments	Description
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.
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.

Tabfield

Command Name and Arguments	Description
active(active:string)	Toggle active flag of current tab field or set its value (0 or 1).
browse_file	Change to the browse file editor of the current tabfield or set the full path to a file.

Command Name and Arguments	Description
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).
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).

Command Name and Arguments	Description
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.

Table

Command Name and Arguments	Description
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_row_num (string: table_key)	Get the number of rows of a table property.
insert_rows_with_xml (string: table_key, int: row_index, restString: XML_data_to_insert)	Insert several rows with their data into a table property.
get_cell_value(string table key, string column key, int row num)	Get a cell value of a table property.
list_columns (string: table_key)	Returns the column IDs of a table property as space separated list.
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.

Text

Command Name and Arguments	Description
<code>incr_alpha(string key, string selectionStart, string selectionEnd, float alpha)</code>	Increment or decrement the transparency of selected text.
<code>incr_kerning(string key, string selectionStart, string selectionEnd, float kerning)</code>	Increment or decrement the kerning of selected text.
<code>incr_position(string key, string selectionStart, string selectionEnd, float posX, float posY, float posZ)</code>	Increment or decrement the position of selected text.
<code>incr_rotation(string key, string selectionStart, string selectionEnd, float rotX, float rotY, float rotZ)</code>	Increment or decrement the rotation of selected text.
<code>incr_scale(string key, string selectionStart, string selectionEnd, float scaleX, float scaleY, float scaleZ)</code>	Increment or decrement the scaling of selected text.
<code>revert(string key)</code>	Revert current text to saved value.
<code>set_alpha_mode</code>	Set text manipulation mode to Alpha.
<code>set_color(string key, string selectionStart, string selectionEnd, string color)</code>	Change the color of selected text.
<code>set_color_mode</code>	Set text manipulation mode to Color.
<code>set_justification(string key, string justification)</code>	Set the text justification (LEFT, CENTER, RIGHT).
<code>set_kerning_mode</code>	Set text manipulation mode to Kerning.
<code>set_position_mode</code>	Set text manipulation mode to Kerning.
<code>set_rotate_xy_mode</code>	Set text manipulation mode to Rotate x and y.

Command Name and Arguments	Description
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.

Trio

Command Name and Arguments	Description
activate_current_playlist	Activate current playlist
assign_script(string page, boolean fileScript, restString scriptunit)	Assign or change the script for the specified page.
cleanup	Performs a clean-up of the current show with all playlists on external renderers by clearing Viz layers and freeing memory.
check_channels_status	Check connection status of program and preview channels in profile. Both for Viz Engine and video channels. Returns "true" if all are good, "false" otherwise.
cleanup_all_channels	Cleans up all channels in the current profile by clearing Viz layers and freeing memory. Does not clear the channel for transition logic pages.
cleanup_channel(restString channelname)	Cleans up the specified channel by clearing Viz layers and freeing memory. Note: The <code>trio:clear_channel</code> command duplicated the <code>trio:cleanup_channel</code> command and is therefore as of version 2.10 deprecated.
cleanup_renderers	Cleans up the renderers of the current playlist (frees graphics and clears output) on the utilized external output channels.

Command Name and Arguments	Description
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.
deactivate_current_playlist	Deactivate current playlist
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 proptype)	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 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).
initialize	Initializes the current show (both page list and all playlists) on external renderers.

Command Name and Arguments	Description
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.
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.
rename_show_folder (string: old_name, restString: new_name)	Rename a folder.
rename_show (string: old_name, string: new_name)	Rename a show.

Command Name and Arguments	Description
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 command or a list of commands separated by newlines to the local viz preview. Returns a list of return values from viz. Newlines within a single command must be escaped (\n).
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 propName, 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.
set_vizconnection_timeout(int timeout)	Set the Viz connection timeout.
set_next_savenumber (int: name)	Set the number that will be proposed on the next saveas or with a create_snapshot_page
show_trio_settings_dialog	Show dialog containing a tree view of Trio settings.

Command Name and Arguments	Description
show_settings_dialog (restString: category)	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.

Util

Command Name and Arguments	Description
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.

Command Name and Arguments	Description
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.

Viz

Command Name and Arguments	Description
cleanup_locally	Cleanup (unload) all viz resources locally. If a scene is active in the local renderer, it is not cleaned up.
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.
load_data_locally (boolean: paths)	Load viz resources locally. The space-separated list of paths must include the pool prefix and full path, e.g. SCENE*folder/name
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.

Command Name and Arguments	Description
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.

vtwtemplate

Command Name and Arguments	Description
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)	Deprecated, use <i>vtwtemplate:invoke_function</i> instead. 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.
invoke_function(string functionName, restString argumentList)	Execute 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.
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 <i>vtwtemplate:invoke_function</i> instead if possible.

Command Name and Arguments	Description
set_template(restString templateName)	Sets the VTW template for the current show.

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 (the port number can be overridden - see [Command Line Parameters](#)), allowing you to use macro commands to trigger actions.

This section contains information on the following topics:

- [Sending Macro Commands](#)
- [Unicode Support](#)
- [Escape Sequences](#)
- [Error Messages](#)

Sending Macro Commands


Establish a connection with Viz Trio over a socket connection to send Macro Commands to control Viz Trio. The following procedures use Telnet as an example.

Configuring Viz Trio For Socket Connections

1. Open the Configuration Window 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**.

Connecting 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 enable Telnet as it's disabled by default.

Sending 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.

Enabling 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 may take several minutes.

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's also possible to use latin1 (ISO-8859-1, western european) encoding.

Toggling 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.

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 supported escape sequences.

Escape Sequences

Escape Sequence	Meaning
<code>\n</code>	New line
<code>\r</code>	Carriage return
<code>\xHH</code>	Two-digit hexadecimal character code.
<code>\</code>	Backslash (\)

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 sends error messages to the connected client as well as being displayed in the GUI (on a per socket connection). The client receives all error messages that are normally just displayed in the Viz Trio user interface.

Note that not only the error messages generated by the client are 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 results 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 Window.

Working with Shortcut Keys

Assign macro commands and scripts to shortcut keys on a per show or global basis. Shortcut keys assigned on a global level are assigned through the Configuration Window, whereas show specific shortcut keys are assigned through the [Show Properties](#).

This section covers the following topics:

- [Assigning a Macro or Script to a Shortcut Key](#)
- [Reassigning a Shortcut Key](#)
- [Removing a Shortcut Key](#)
- [Adding a Predefined Function to a Macro or Script](#)

Assigning a Macro or Script to a Shortcut Key

1. Click the **Add Macro** or **Add Script** button. The **Macro Commands editor** opens:

The image shows a dialog box titled "Macro Commands editor". It contains the following elements:

- Enter Key:** A text input field containing "Ctrl+I".
- Macro Name:** A text input field containing "My Macro".
- Macro Commands:** A text area containing the following commands:


```
read 1000
tabto 1
set_value Headline Add text here!
tabto 2
set_value Body Add text here!
```
- Buttons:** "Show Macros...", "Save as New", "OK", and "Cancel".

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.

Reassigning a Shortcut Key

Double-click the Command in the list and perform the changes.

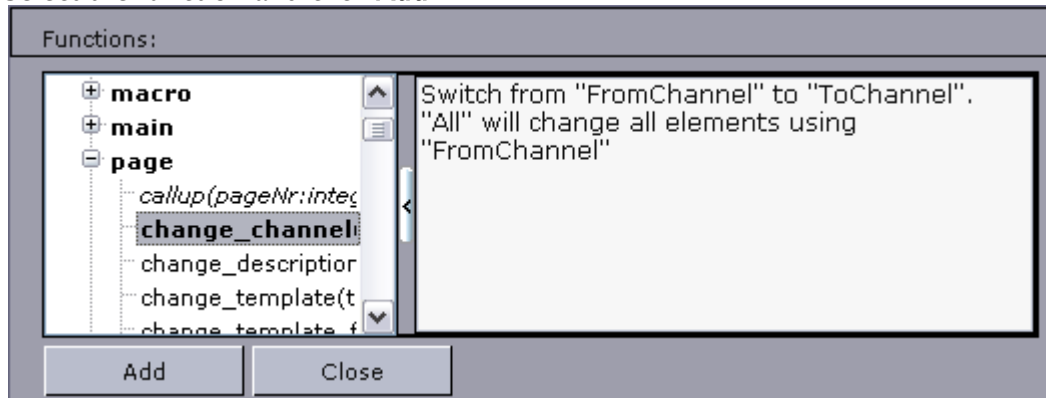
Removing a Shortcut Key

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.

Adding a Predefined Function to a Macro or Script

1. Open a Macro or Script for editing.
2. Click the **Show Macros ...** button in 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.

7 Designing Scenes

There are two ways of designing scenes for Viz Trio:

- [Creating Scene Elements in Viz Artist](#) - in most cases this is the preferred method.
- [The Viz Trio Designer](#) - the built-in designer. The Designer lets you use pre-made scene elements or building blocks to create a standalone scene.

Note: Transition logic is not supported.

7.1 The Viz Trio Designer

7.1.1 The Viz Trio Designer

Design

Viz Trio Designer contains pre-defined scene elements that are created in Viz Artist and include adjustable control plug-ins required to expose the scene properties. You can easily group different scene elements together to create more advanced scenes. It's also possible to modify the scene element's properties, if they have been exposed for editing (see **Control Plug-ins**).

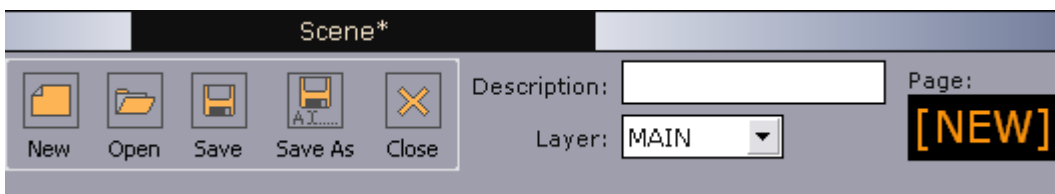
This section covers the following topics:

- [Scene](#)
- [Resources](#)
- [Scene Tree](#)
- [Tab Fields](#)
- [Page Editor](#)
- [Properties](#)

7.1.2 Starting the Designer

- Click **Tools** and select **Trio Designer** from the menu.

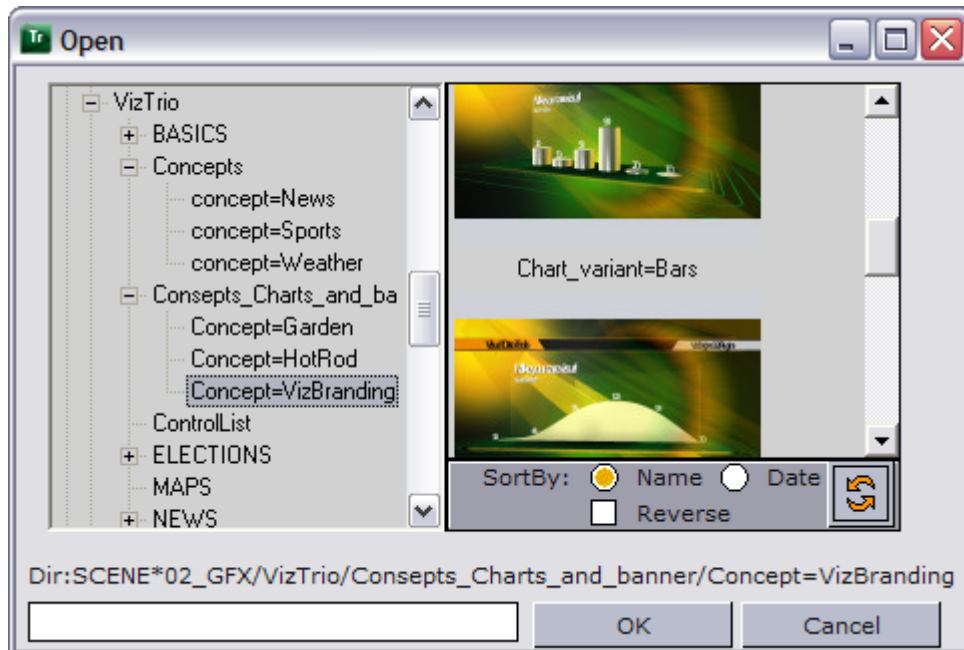
Scene



The scene pane is located in the upper left corner of the Designer; it contains functions for scene management and control.

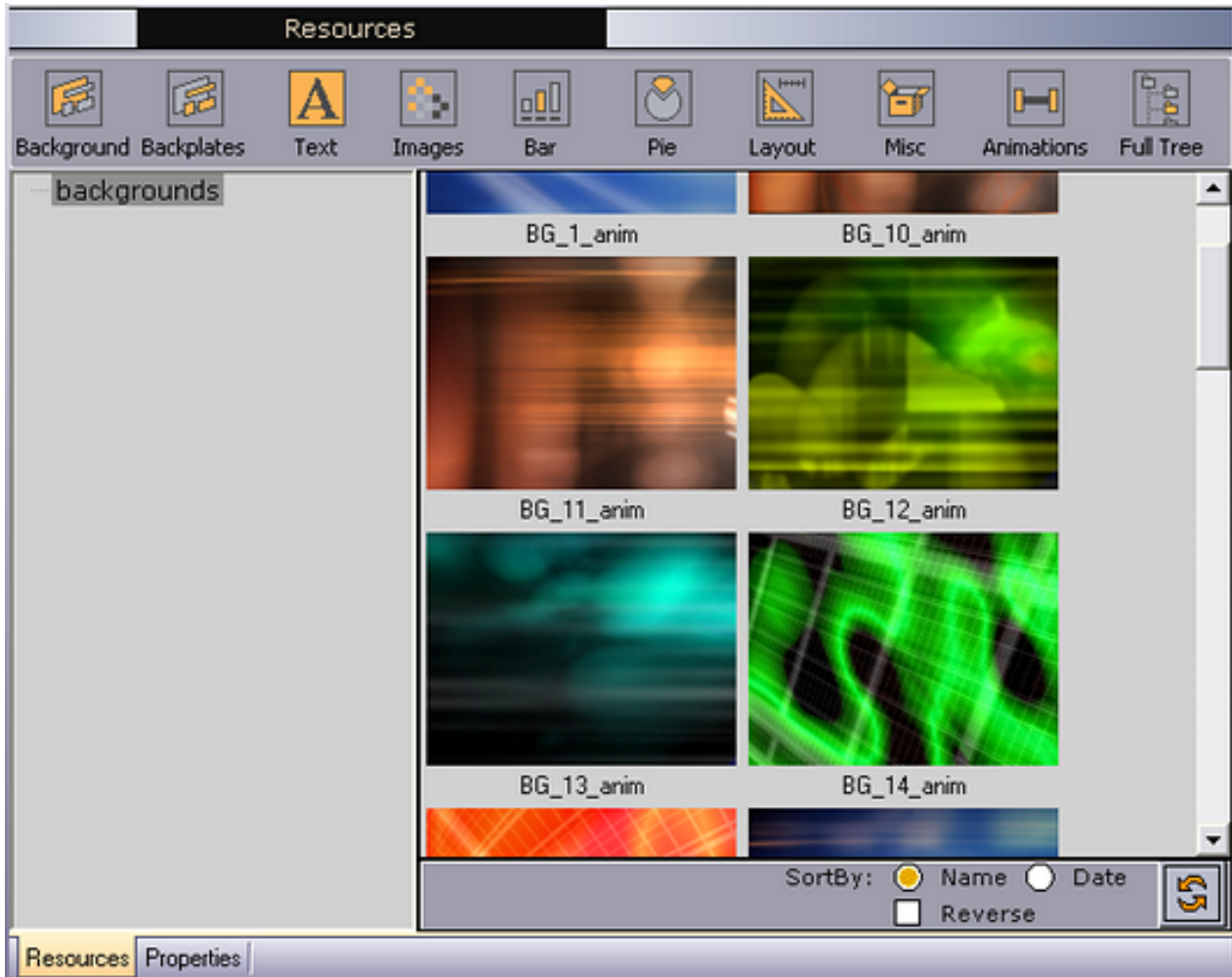
- **New:** Create a new scene.

- **Open:** Displays the Viz browser, which provides access to the Viz scene database. An icon to the right of the scene database pane represents each scene as it was when last saved. Scenes are organized in folders accessible from the folder tree on the left. The tree structure is the same as that in the **SCENE** folder in the Viz Artist data directory. Scenes can be sorted by **Name** or **Date**.



- **Description:** This is displayed in the Template Description column in Viz Trio.
- **Layer:** Specify the renderer layer for the new scene. Graphics can be played in the **FRONT**, **MAIN** and **BACK** layer. The **MAIN** layer, also called **MAIN**, is the default layer. By using different layers, you can 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 main and back layers. Correspondingly, graphical elements in the main layer will block graphics in the back layer.
- **Callup Code:** For the selected page. New pages that have not been saved have the page name **[NEW]**. Click **Save As** to save the page with a specific callup code.

7.1.3 Resources



Note: The library path is set in [Show Properties](#), 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. Scene elements are created in Viz Artist, and include the required control plug-ins.

Info: Scene elements are not standalone scenes, rather they are like building blocks that can be pieced together with other scene elements to create a standalone scene on the fly.

Scene elements are organized under a LIBRARY folder in Viz containing the following sub folders: BACKGROUNDS, BACKPLATES, TEXT, IMAGES, BARS, PIES, LAYOUT, MISC and ANIMATIONS. These folders become available when selecting the corresponding categories using the buttons in the Resources pane.

In addition to pre-defined scene elements, you can use ordinary Viz Artist scenes when creating new scenes in Designer if a group with a placeholder has been added to the scene. The Viz Artist folder tree is displayed when clicking the **Full Tree** button.

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. The scene elements also contain control plug-ins that let you 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. 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 dragging and dropping it into 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 dragging and dropping it into 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 by dragging and dropping it into the appropriate placeholder. This will automatically enable text editing in the tab-field.
- **Images:** Shows the images available in the IMAGES library folder. As with the **Text** button, the **Images** button adds a tab-field for editing by an operator. Clicking **Images** opens the Images library where an image can be selected and added by dragging and dropping it into 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 dragging and dropping it into 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 dragging and dropping it into 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 lines or rows, for example. The layout elements themselves usually do not contain any graphics, but should rather ideally function as placeholders in which other types of scene elements can be added. Select a layout, and add it to the scene by dragging and dropping it into the root placeholder.
- **Miscellaneous:** Shows the scene elements available in the MISC library folder. Store things here that do not fit into other categories, such as 3D objects, bullets, and placeholders. Select a scene element, and add it to the scene tree by dragging and dropping it into the appropriate placeholder.

- **Animation:** Shows animation scene elements in the ANIMATIONS library folder. Add an animation to the scene by dragging and dropping it into the placeholder that contains the scene element you wish to animate.
- **Full Tree:** Shows a full Viz directory structure of all available scenes.

7.1.4 Scene Tree



The scene tree shows all the graphical elements in a scene in a logical way. 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 into the scene tree, where you can create a hierarchy between elements and groups of elements in logical divisions. This lets you assign new properties to more than one scene element in a single operation.

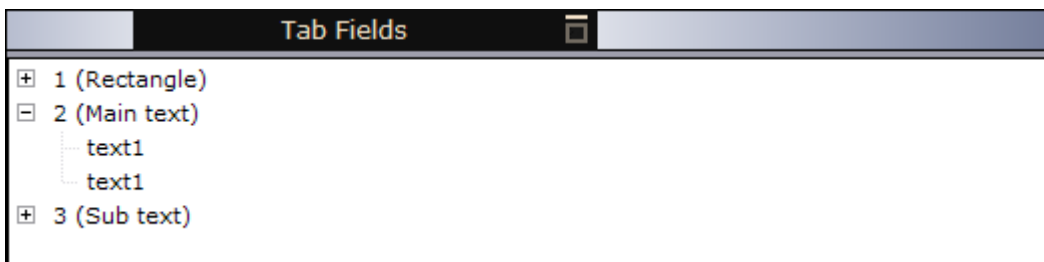
The **black arrow** to the left indicates that the placeholder contains sub-containers. If the arrow points to the right the sub-containers are not displayed. Click on the arrow to expand the sub-containers. The arrow will then point downwards.

Deleting an Item in the Scene Tree

- Select the appropriate scene element and then press the **DELETE** key. Note that this also deletes sub-nodes.

Caution: You cannot undo a delete operation.

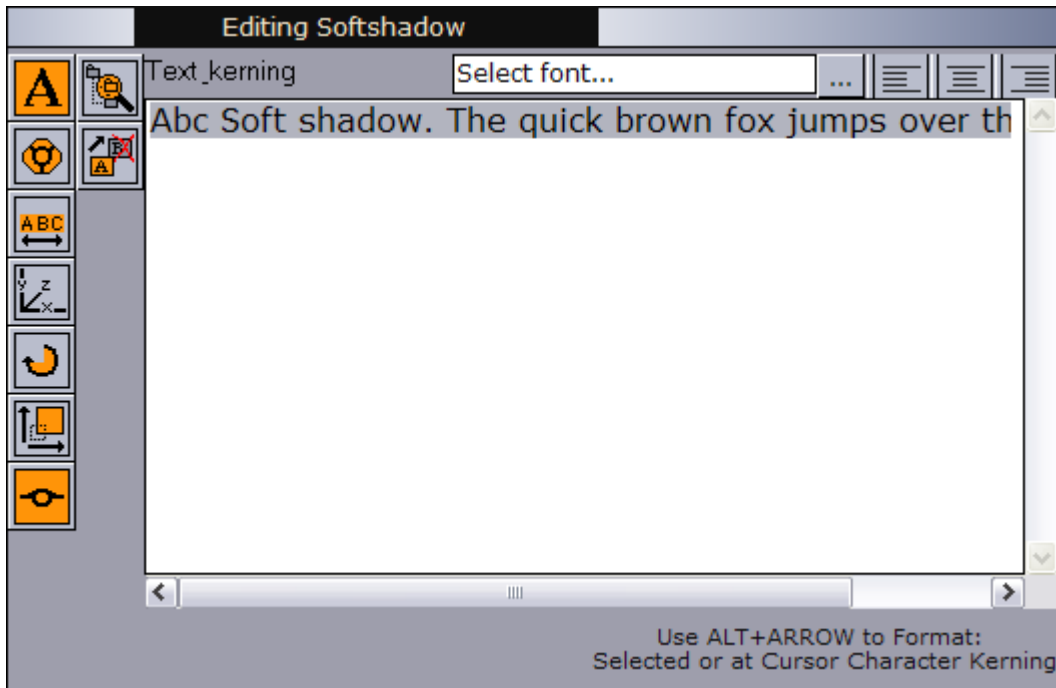
7.1.5 Tab Fields



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. The tab fields pane shows which [Properties](#) can be exposed for editing in Viz Trio. Click + to expand the list.

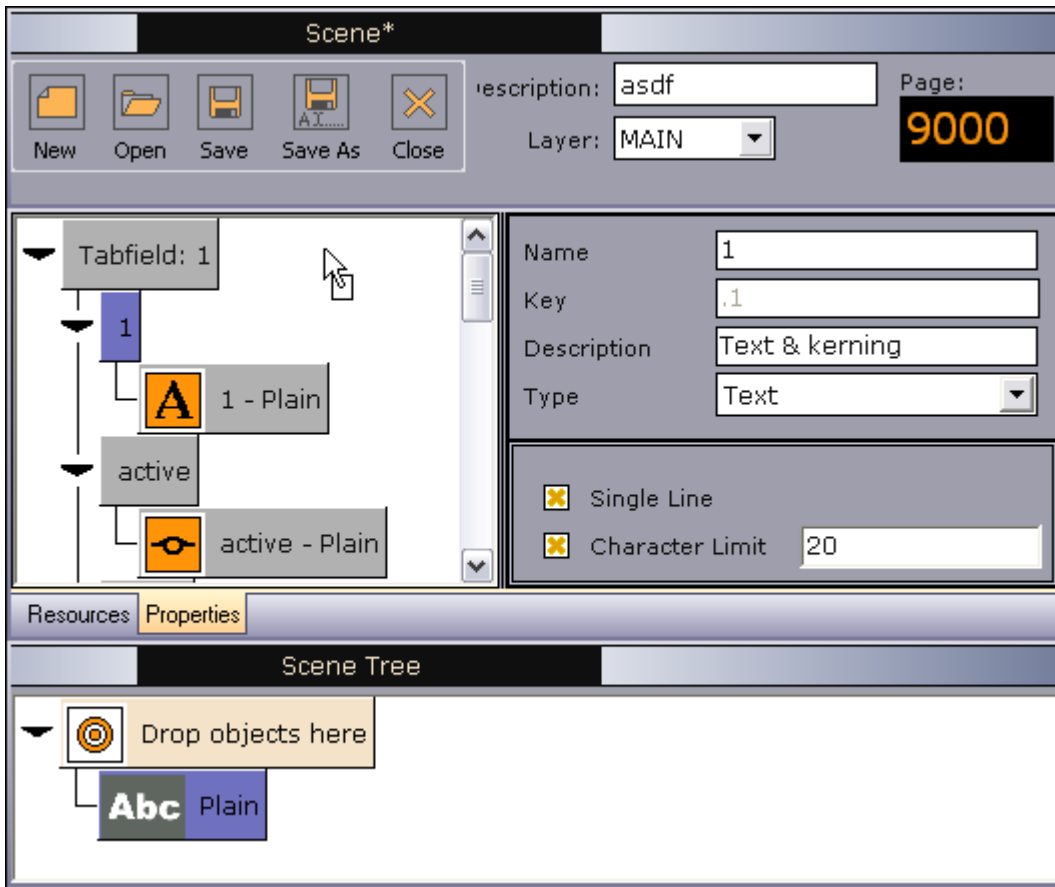
7.1.6 Page Editor



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, such as changing material, position, rotation, text, etc. if they have been exposed for editing.

The type of page editor displayed varies according to the type of element selected, for example text or images. The buttons on the left represent the properties that can be exposed. To display an editor, click on the appropriate button or select the property in the [Tab Fields](#) pane.

7.1.7 Properties



Select the **Properties** tab at the lower left. Here, you can define which **Tab Fields** to expose to editing in Viz Trio.

This section covers the following topics:

- [Property Tree](#)
- [Property Description Editor](#)
- [Property Type Editor](#)
- [Exposing Properties](#)
- [Mapping Multiple Property Receivers to One Exposed Property](#)
- [Removing Properties](#)

Property Tree

When creating a new scene, the properties pane is initially empty. Tab field properties are exposed for editing by adding them to the properties pane (top left).

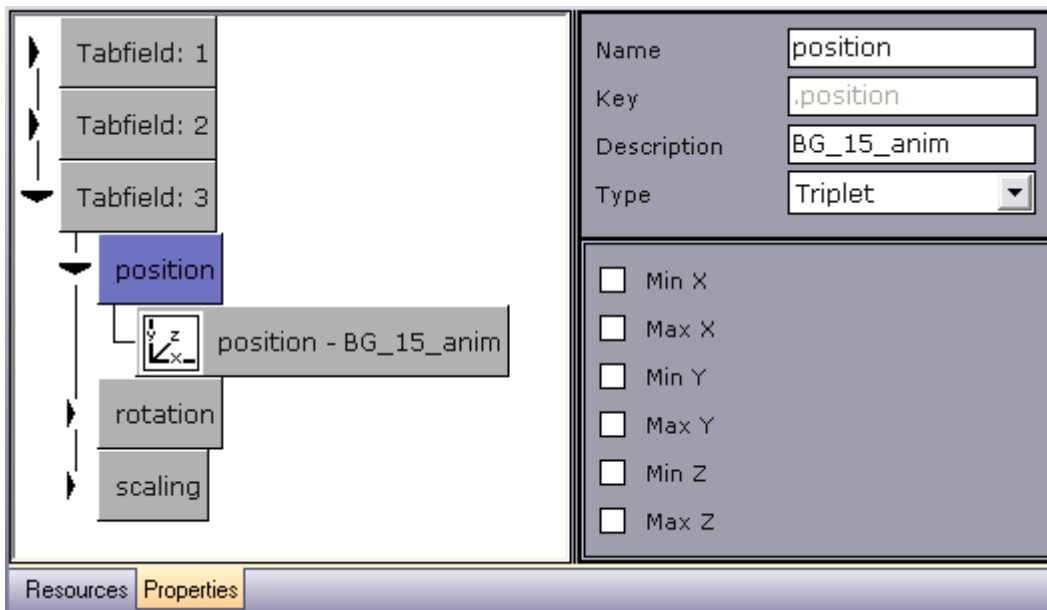
There are two ways of [Exposing Properties](#) in Designer:

- one by one in a tab-field, OR
- exposing all properties in a single operation

Use the **Properties** pane to define tab field properties to expose for editing. 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 as needed for editing.

A designer can define properties to be exposed for editing when creating a scene element in Viz Artist. Exposed properties may vary from scene to scene.

Note: Additional properties must be exposed through Viz Artist.

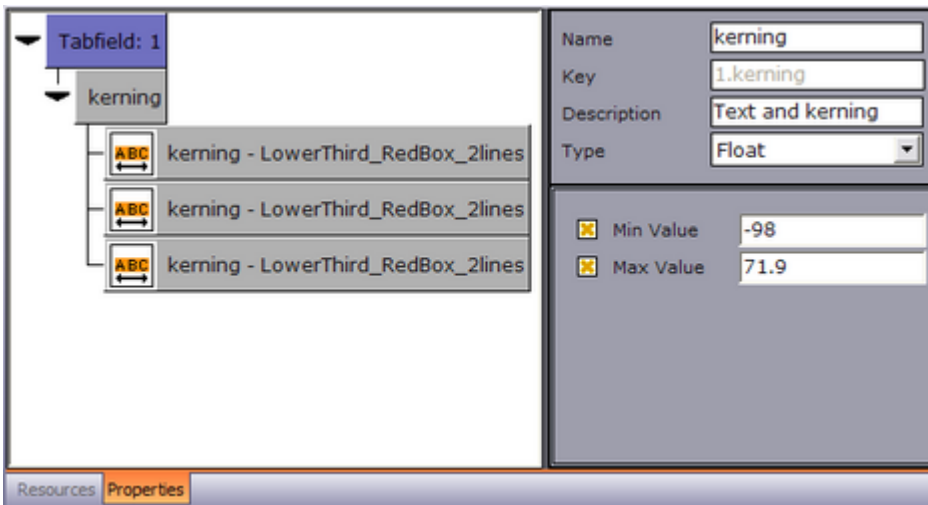


The nodes in the properties tree are named **Tabfield: 1**, **Tabfield: 2**, **Tabfield: 3**, etc. as they are added. The order of the tab fields can be rearranged by drag and drop.

Each tab field has sub-nodes representing the exposed properties (such as 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, and the receiving scene element's name (for example, background).

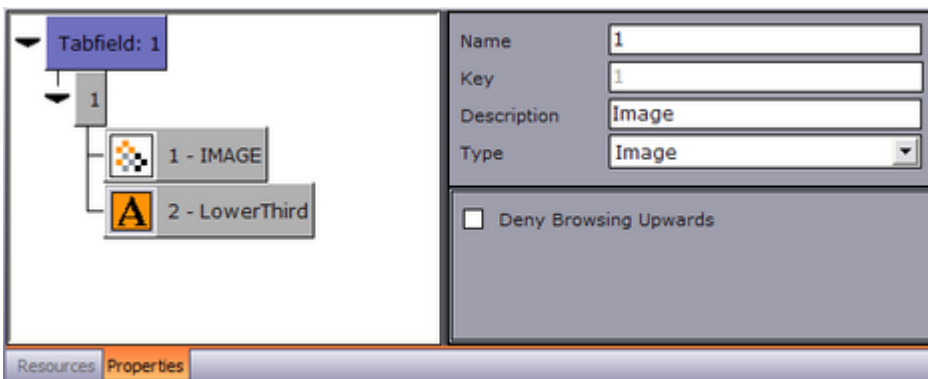
It's possible to expose several properties for one tab field by dragging the property icons into the same tab field container from the [Page Editor](#).

It's also possible to map two or more property receivers to one exposed property in order to control the value of several property receivers through a single exposed property. This is useful for changing a property such as kerning or material for several scene elements in a single operation, for example. This is shown in the example below, where the exposed property **kerning** controls the kerning value of the three attached property receivers **LowerThird_RedBox_2lines**.



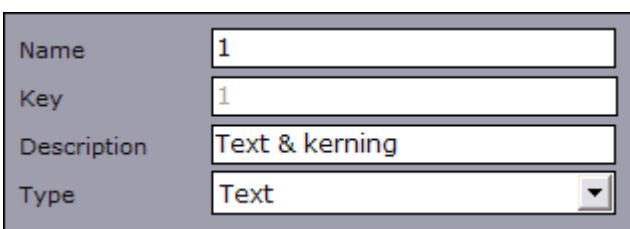
Different types of property receivers such as an image and a text object can be mapped to the same exposed property, even when located in different tab fields. This lets you edit all the associated property receivers in one single operation.

In the example below, the exposed property **Image** controls the content of the associated image and text property receivers. This means that when a new image is selected (see [Search Media](#)), the text related to the image will also be added.



Another example is when you are connected to a newsroom system and you change a text and an image in the same tab field by entering a text string rather than the whole path. This indicates that there is a folder of images (such as 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 Property Description editor is in the upper right corner of the Properties pane. The fields

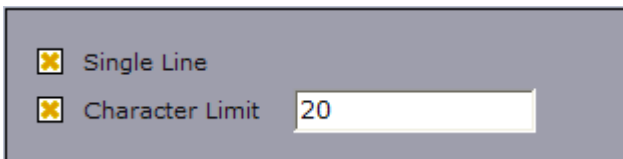
become available for editing when selecting a property in the [Property Tree](#). All fields are editable, apart from the **Key** field:

- **Name:** Name of the selected tab field property.
- **Key:** The key identifier of the property.
- **Description:** Description of the selected tab field.
- **Type:** The data input type defined for the selected property. This value determines which [Page Editor](#) will be displayed.

Property Type Editor

The Property Type editor is 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 are also updated so that values for the associated parameters can be specified.

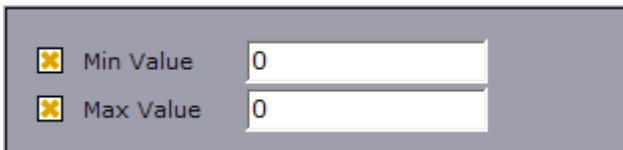
Text



Set text parameters:

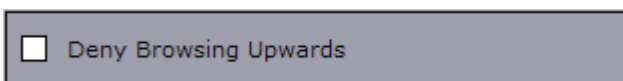
- **Single Line:** Disables text wrapping, displaying the text as a single line.
- **Character Limit:** Sets a maximum number of characters for the text tab field.

Float



Specify a minimum (**Min Value**) and a maximum (**Max Value**) value for the selected property. Select a check boxes to display its input field.

Deny browsing



Block the user from browsing for images, geometries, and materials in folders other than the selected folder and its sub-folders.

Duplet

<input checked="" type="checkbox"/>	Min X	0
<input checked="" type="checkbox"/>	Max X	0
<input checked="" type="checkbox"/>	Min Y	0
<input checked="" type="checkbox"/>	Max Y	0

Specify minimum and maximum values for the X and Y axes of the selected property. Select a check boxes to display its input field.

Triplet

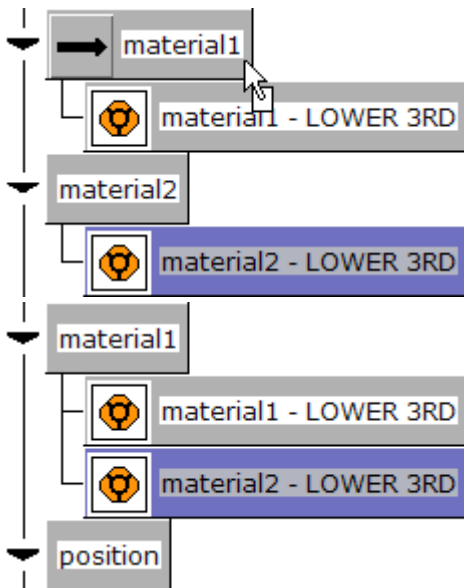
<input checked="" type="checkbox"/>	Min X	0
<input checked="" type="checkbox"/>	Max X	0
<input checked="" type="checkbox"/>	Min Y	0
<input checked="" type="checkbox"/>	Max Y	0
<input checked="" type="checkbox"/>	Min Z	0
<input checked="" type="checkbox"/>	Max Z	0

Specify minimum and a maximum value for the X, Y and Z axes of the selected property. Select a check boxes to display its input field.

Exposing Properties

1. In the [Page Editor](#), select the icon or design element representing the property to expose for editing, and drag and drop it into 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.

Mapping Multiple Property Receivers to One Exposed Property



- Drag and drop the property receiver into the exposed property that is to control both properties.

Removing Properties

Select a property and press the **Delete** key on your keyboard. Note that this will also delete sub-nodes.

See Also

- [Viz Artist User Guide](#) on transition logic.

7.2 Creating Scene Elements In Viz Artist

Creating scene elements primarily consists of two 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.

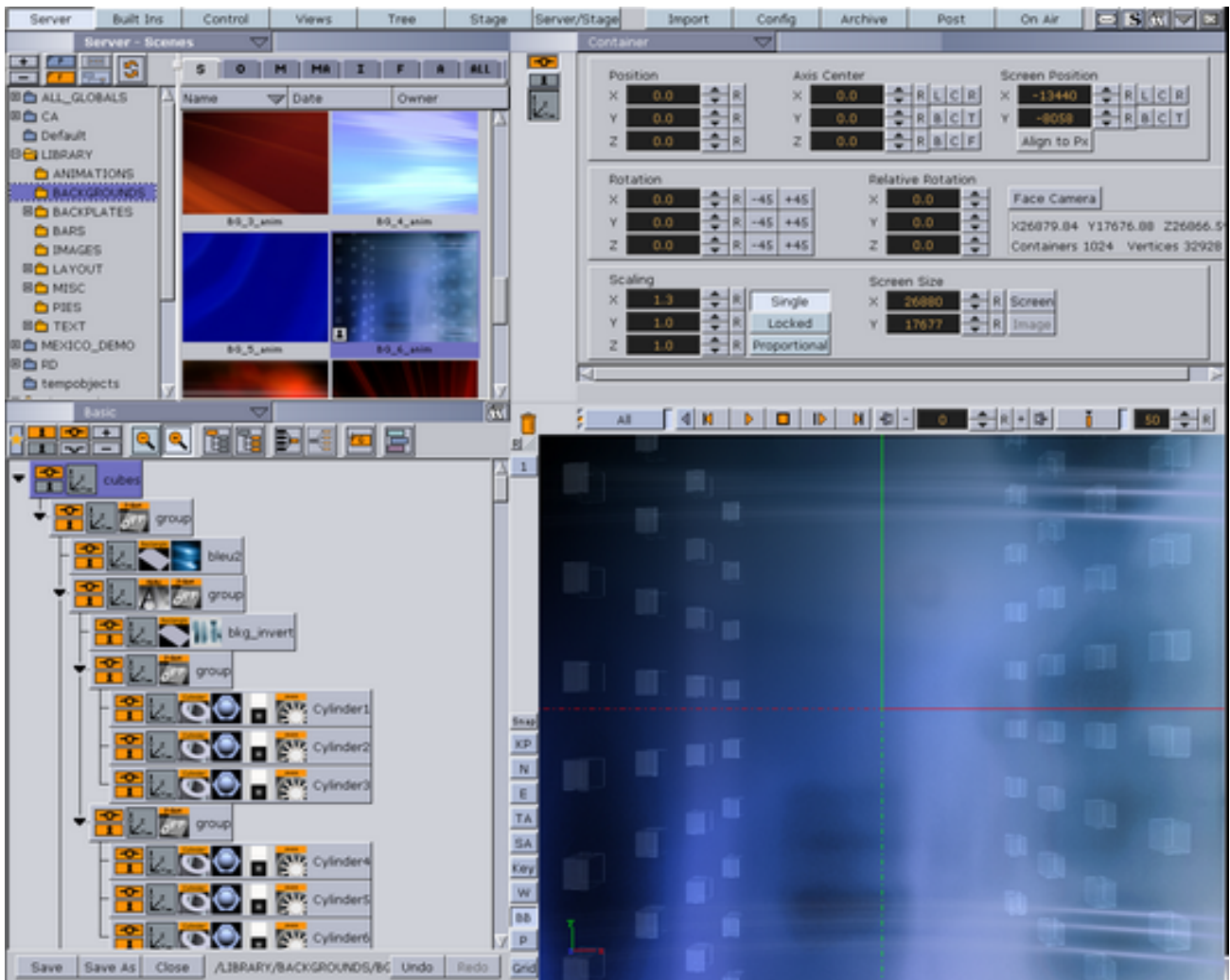
Scene elements can have varying complexity, with one or more tab fields, different types of animations, etc. Scene elements may also include placeholders that are non-graphical objects. These can in turn contain other scene elements, such as text objects.

This section covers the following topics:

- [Viz Artist User Interface](#)
- [Creating Scene Elements in Viz Artist](#)
- [Adding Control Plug-ins](#)

- Creating Backgrounds
- Creating Backplates
- Creating Text Objects
- Creating Image Objects
- Creating Animations


7.2.1 Viz Artist User Interface



Open Viz Artist by clicking the **Viz Artist** button in the upper right corner of Viz Trio. At startup, the Viz Artist user interface contains:


- **Resources:** Contains all the different objects stored on Graphic Hub (Viz 3.x); located in the upper left corner of Viz Artist under the **Server** tab.
- **Editors:** Lets you modify the properties of an object; displayed in the upper right corner.
- **Scene tree:** Displays all the elements in a scene and their hierarchy; displayed in the lower left corner.

- **Preview:** Displays a WYSIWYG representation of what the graphics will look like on-air; located in the lower right corner.

 **Note:** Scene builder tools generally use drag and drop actions.

7.2.2 Creating Graphics

Create scene elements with the building blocks available in the Viz Artist database.

 **Note:** Building blocks are placeholders that contain different types of graphical elements and/or plug-ins. They are therefore also called *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 inherit the properties of the parent container.

The building blocks you select will depend on the type and complexity of the design. A basic description on how to create scene elements in each of the categories follows below.

This section covers the following topics:

- [Creating a New Scene Element](#)
- [Adding Group Containers](#)
- [Adding Geometries](#)
- [Editing Scale and Position](#)
- [Copying Containers](#)
- [Deleting Containers](#)
- [Adding Materials](#)
- [Adding Fonts](#)
- [Adding Images](#)
- [Adding Key](#)
- [Defining Tab Fields](#)
- [Saving Scenes](#)

Creating a New Scene Element

1. Start Viz Artist
2. Select Scene view.
 - *For Viz Artist 3.x:* Click **Server** and select **Scene** from the drop-down in the upper left corner of the Database window and click the **S** tab.
 - *For Viz Artist 2.x:* Click **Scene** in the upper left corner of the Database window.
3. Locate and expand the LIBRARY folder in the folder tree.
 - The sub-folders displayed correspond to the various library categories in the Designer and contain 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 dialog box, and click **OK**.
 - The new scene becomes available at the right of the Server view.

Adding Group Containers



The Group function is a container that lets you group elements in the scene tree, and thereby create a parent-child hierarchy in which the child containers inherit the properties of its parents.



1. Open the scene (see [To create a new scene element](#)).
2. Add a group container to the scene tree.

Adding Geometries

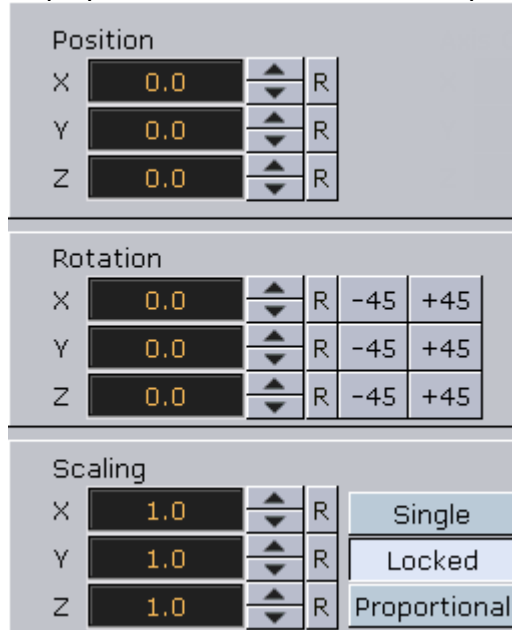


1. Add a geometry object (e.g. Rectangle), and add it as a sub-container of the group container in the Scene Tree.
2. Rename the rectangle's sub-container to for example *background* in order to describe its function.

Editing Scale and Position



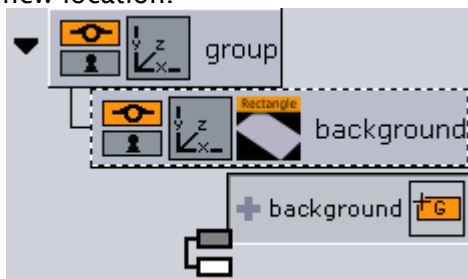
1. Display the Transformation editor by selecting the **Transformation** icon on the container.



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 or over the shoulder..
3. Specify the order of layered objects by using the Z field in the **Position** section.
 - Some objects, such as a background or a backplate, must be displayed behind the other elements in a scene in order to prevent blocking them.
 - Keep in mind that the object with the highest z-position will be displayed at the front, while the object with the lowest z-position will be displayed at the back (unless you are rotating objects on the y-axis, in which case you might need to manually sort objects on the Z-axis using the Z-sort plug-in).

Copying Containers

1. Press the CTRL key and the left cursor button simultaneously and drag the container to its new location.



- *For Viz Artist 3.x:* A node symbol appears to the left of the copied container, indicating the position where the new container will be added when releasing the cursor button.
- *For Viz Artist 2.x:* An arrow appears at the far left of the scene tree, indicating the position where the new container will be added when releasing the cursor button.

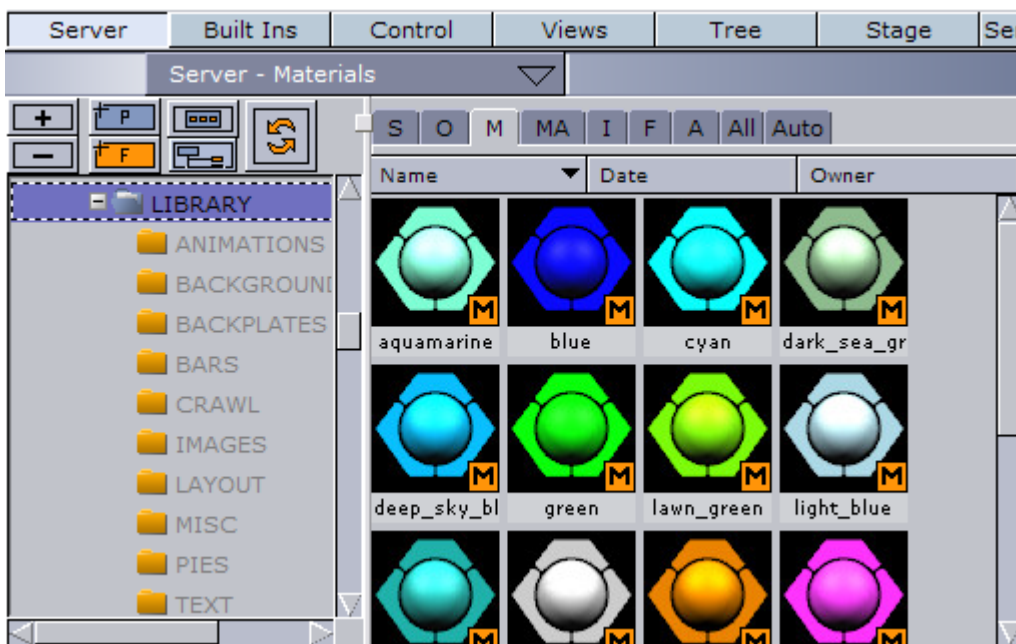
2. To copy multiple containers, press the **CTRL** key and select containers with the left cursor button. Drag and drop the containers to the new location.

Deleting Containers

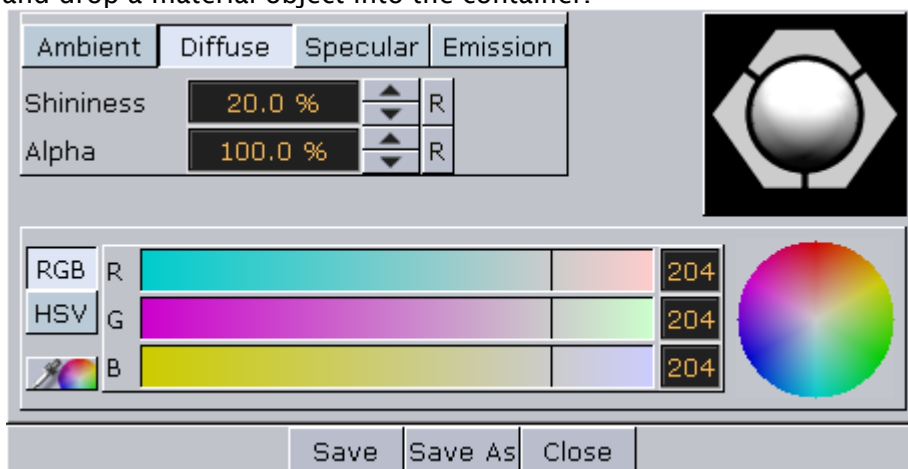


1. Select containers to delete with the left cursor button and drag them into the trash can above the Renderer window, or
2. Right-click the container and select **Delete Container <name>** from the menu (*Viz Artist 3.x only*).

Adding Materials

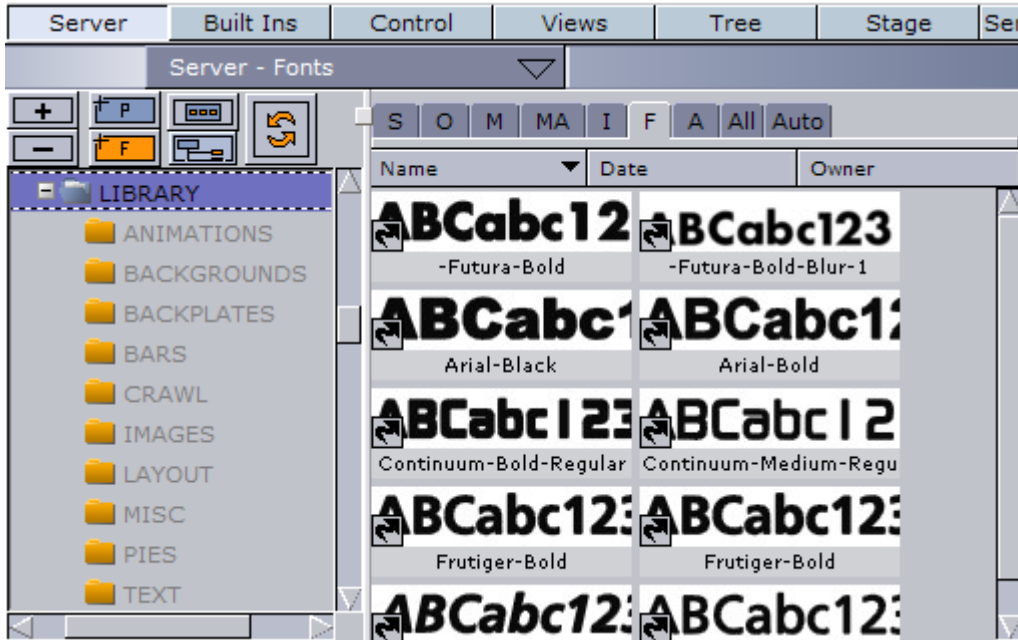


1. Select the material (**M**) tab under the server view (**Material** button in Viz Artist 2.x), and drag and drop a material object into the container.



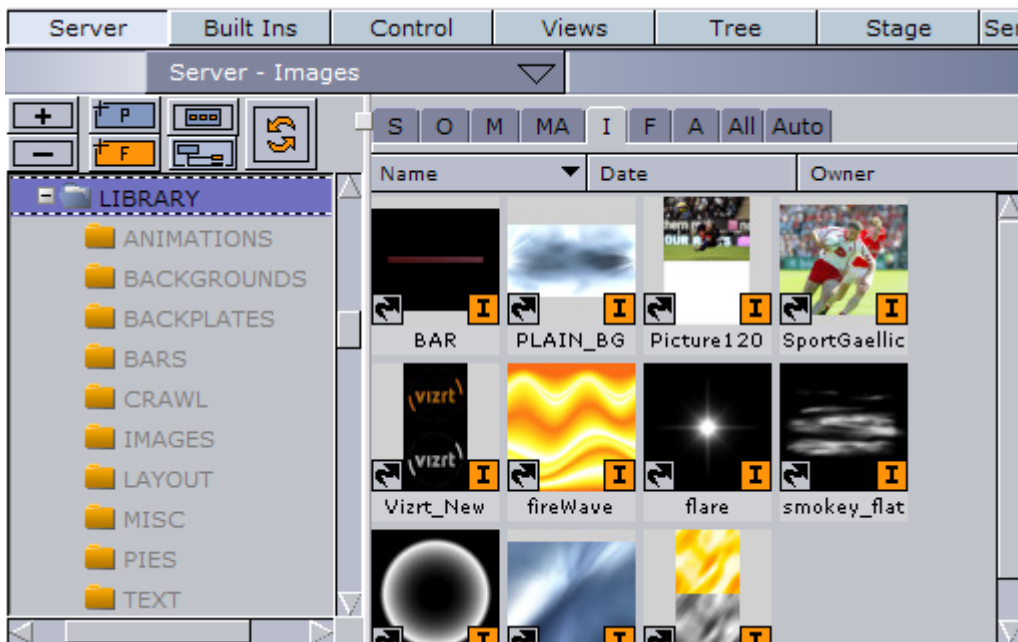
2. Double-click the material to open the Material Editor to change effects or lights (e.g. Ambient, Diffuse, Specular and Emission).

Adding Fonts



1. Select the font (F) tab under the server view (Font button in Viz Artist 2.x), and drag and drop a font into the container.
2. Open the transformation editor to change the size and position of the text.

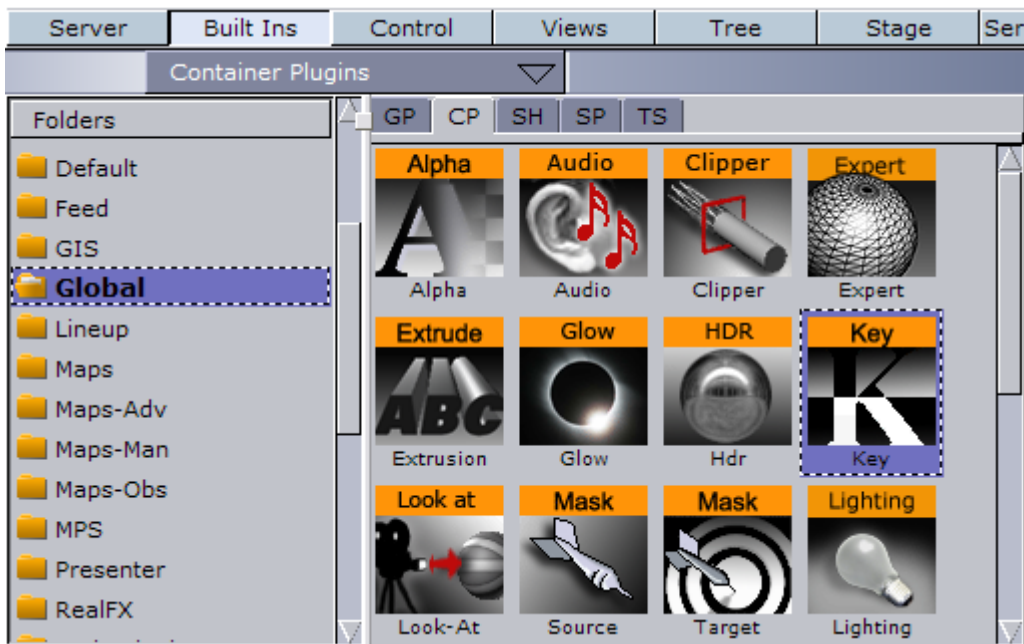
Adding Images



1. Select the image (I) tab under the server view (**Image** button in Viz Artist 2.x), and drag and drop an image into the container.
2. Open the transformation editor to change the size and position of the text.

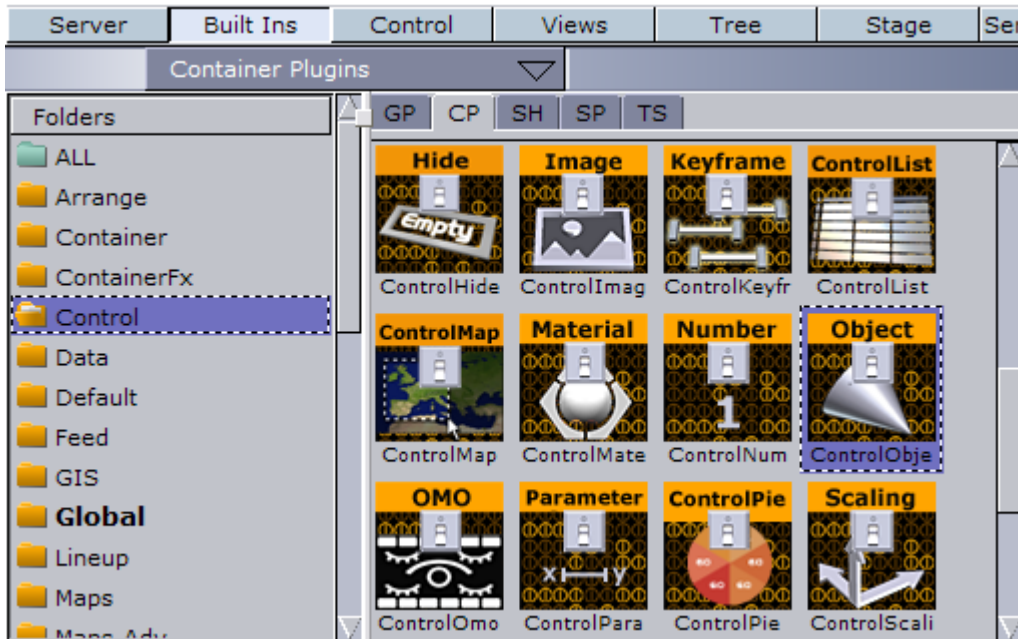
Adding Key

In order to enable Viz Engine to produce a key signal, key must be added to individual containers that contain objects that should draw key or the scene in general.



1. Click the **Built Ins** button, and select the **Global** folder from the **Container Plugins** (CTRL + 2).
 - For Viz Artist 2.x: Click **Function**.
2. Drag and drop the **Key** plugin into the containers that should have key.

7.2.3 Adding Control Plug-ins



To enable Viz Trio operators to edit the graphical elements, the containers must contain the necessary control plug-ins, which must be added for each of the graphical elements that are to be edited. The plug-in selected depends on the type of property that needs to be exposed for editing.

In addition to container specific plug-ins, the scene tree must contain a plug-in called the Control Object plug-in, which allows Viz Trio to identify a scene as a Viz Trio template.

Note: Only containers placed under the Control Object plug-in in a scene tree can be imported to Viz Trio.

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 which is added automatically if another control plug-in is added to the scene tree. Control Object can also be added manually.

Adding a Control Plug-in

1. Click **Built Ins**, and select the **Container Plugins (CTRL + 2)** option from the drop-menu or 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 the control plug-in icon to display the associated editor.

Defining Tab Fields

Viz Trio uses the control plug-in's Field Identifier ID to identify the container as an editable element. In addition, the field's ID defines a tab-field and is used to specify the tab-order between editable elements in a scene.

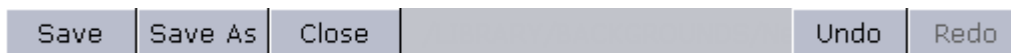
Note: All control plug-ins located in the same container must have the same field ID; this ensures that all associated property editors are enabled for selection when a tab-field is chosen.

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*. For example, *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).

Saving Scenes

Save and close scenes at the lower left corner of the Viz Artist interface.



1. Click **Save** to save a new scene or to overwrite an open scene, or
2. Click **Save As** to save it as a new scene.

7.2.4 Creating Backgrounds

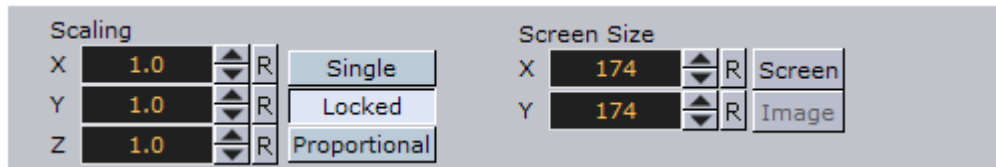
This section shows how to create a basic background with an image.

Note: The screenshots below are taken from Viz Artist 3. The user interface in Viz Artist 3 is similar to Viz Artist 2. Additional descriptions are provided where necessary.

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 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](#)).
 - Click the **Transformation** button to open the Transformation editor.
 - Change the position in the Position section in the upper left corner of the Transformation editor.
 - Adjust the Z position in the Position section to ensure that the background is displayed behind all other elements in a scene. Remember that the Z position value of

backgrounds must always be lower than the Z position value of the other scene element categories.

- If the background contains animations, verify that they will not overlay the other objects at any point. A low z-position value such as -500 will normally be sufficient, but increase this as required.

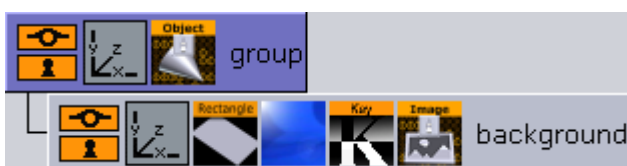


6. Change the size of the rectangle to cover the whole area of the Renderer window to create a full-screen background. This is most easily done by pressing the **Screen** button available in the Screen Size section in the lower right corner of the Transformation editor. The rectangle will then be re-sized to cover the screen.
7. Add an image to the scene tree (see [To add images](#)).
 - Select an image and add it into the **background** container using drag and drop.

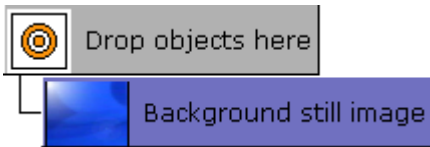


8. Add the key function to the **background** container (see how [To add key](#)).
 - In order for Viz Engine to produce a key signal, a key function must be added.
9. Add a **Control Image** plug-in to the **background** group; the Control Object plug-in will then be added automatically (see [Adding Control Plug-ins](#)).
 - Adding the Control Image plug-in allows the background object to be modified in Viz Trio in playback or design mode.
10. 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. The image position and scaling can be exposed for editing if required.
11. Add a description of the background object. Select the Control Object plug-in icon on the **group** container. Enter a description of 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 that displayed when adding a scene element to the scene tree in Viz Trio's Designer. Although optional, it's recommended to use the scene name as a description in order to provide a better overview.
12. Click **Save** to save the background object. Select the appropriate BACKGROUND directory in the Scene database to see the new scene icon.

Viz Artist Scene Tree



Viz Trio Designer Scene Tree



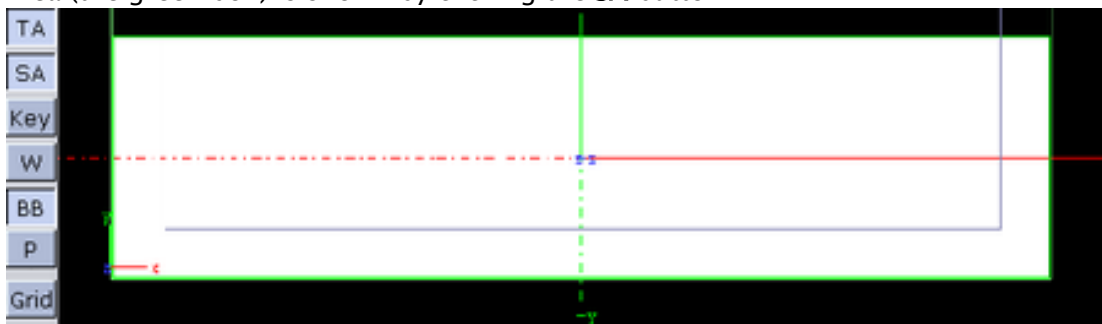
Info: After finishing the tutorial, the Viz Artist and Designer scene tree should look like the screenshots above.

7.2.5 Creating Backplates

Create a basic backplate with two text tab fields. For details on how to add animation to a backplate, see [Creating Animations](#).

Note: The screenshots used below are taken from Viz Artist 3. The user interface in Viz Artist 3 is similar to Viz Artist 2. Additional descriptions are provided where necessary.

1. Start by creating a new scene in the `BACKPLATES` directory in the Viz Artist scene tree (see how [Creating a New Scene Element](#)).
2. Double-click on the new scene to open it. Add a group container to the scene tree (see how [Adding Group Containers](#)). Rename the container to *object* by double-clicking the label.
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: **show/hide**, **lock/unlock**, **transformation editor**, and **rectangle**. For more information, see [Adding Geometries](#).
5. Adjust the rectangle's size and position. For more information, see [Editing Scale and Position](#).
 - Scale and position the rectangle to create a backplate. Use the Title and Safe Areas as references when positioning the rectangle. The Title Area (blue box) is displayed by clicking the **TA** button in the lower left corner of the Renderer window, while the Safe Area (the green box) is shown by clicking the **SA** button.



6. 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 shown in the screenshots.

- The backplate should be able to be repositioned and scaled in the Designer and/or in playout mode. Complete accuracy in terms of scale and position are therefore not required here.

Note: Any rounded/beveled corners will be stretched if scaled.

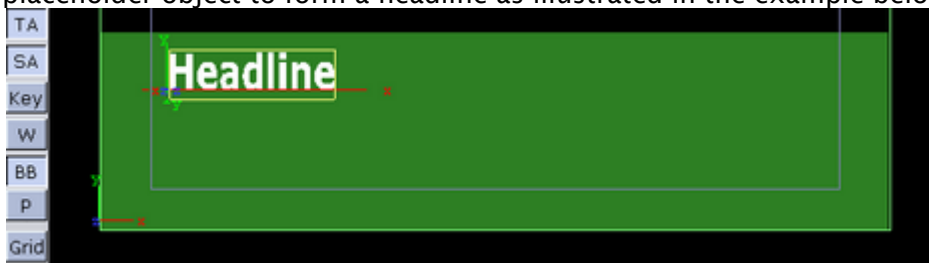
- Set a Z-value slightly below zero (e.g. 10) for the backplate.
 - For objects to be displayed in the front, always select zero or a higher value for the z-position. This will ensure that the scene elements are displayed in the correct layers and do not cover each other.
 - This value could be lower depending on the animations added later.
- Add a material to the scene tree (see [Adding Materials](#)).
 - Select a material and add it to the **backplate** container using drag and drop.
 - To modify the color, display the Material editor by clicking the icon.
- Add the key function to the **background** container. For more information, see [Adding Key](#). In order for Viz Engine to produce a key signal, a key function must be added.
- Add a new group as a sub-container to the **object** container, and name it **placeholder**. This will create a new empty group above the **backplate** container.
- 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.



- 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.

- Open the *placeholder* container's transformation editor, and scale and reposition the placeholder object to form a headline as illustrated in the example below.



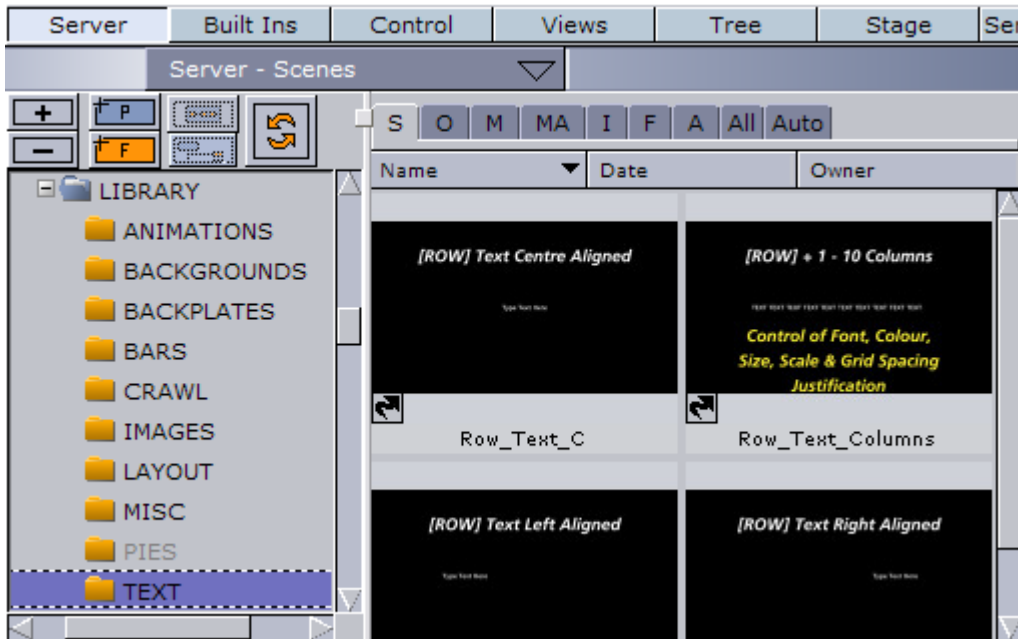
- Adjust the Z-position in order to ensure that it will be displayed in front of the backplate. Generally a value slightly above zero will be sufficient.

15. When finished adjusting the placeholder object delete the font object as it is no longer needed (see [Deleting Containers](#)).
16. 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.
17. 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.



18. 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.
19. 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*.
20. Click the Save button to save the backplate object.
21. 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.

7.2.6 Creating Text Objects



⚠ The screen shots used below are taken from Viz Artist 3. The user interface in Viz Artist 3 is similar to Viz Artist 2. Additional descriptions are provided where necessary.

1. Create a new scene in the TEXT directory in the Viz Artist scene tree (see how [Creating a New Scene Element](#)).



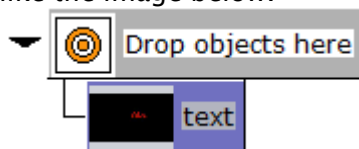
2. Rename the group container `_object_` (see [Adding Group Containers](#)).
3. Add a font object to the scene tree as a sub-container of the `object` container, and rename the new container `text`.
4. Open the Text editor, and enter a common phrase (e.g. *Abc*).
5. Add the key plugin to the `object` element (see [Adding 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 [Adding Materials](#)).
7. Add the **Control Text** and the **Control Material** plug-in to the `text` container. This allows 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.

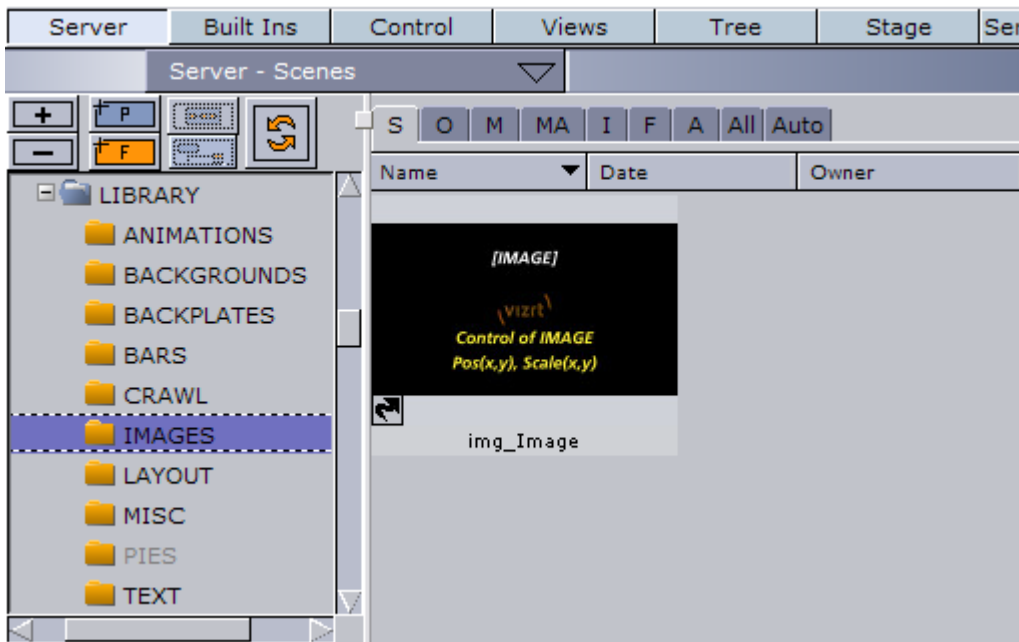
8. Select the icon to display the **Control Text** editor.

- 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 below.



Creating Image Objects



Note: The screen shots used below are taken from Viz Artist 3. The user interface in Viz Artist 3 is similar to Viz Artist 2. Additional descriptions are provided where necessary.

1. Create a new scene in the IMAGE directory (see [Creating a New Scene Element](#)).



2. Add a group container to the Scene Tree (see [Adding 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 *image*.
5. Add the key function to the **object** element (see [Adding 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 Guide for further details on the different control plug-ins.

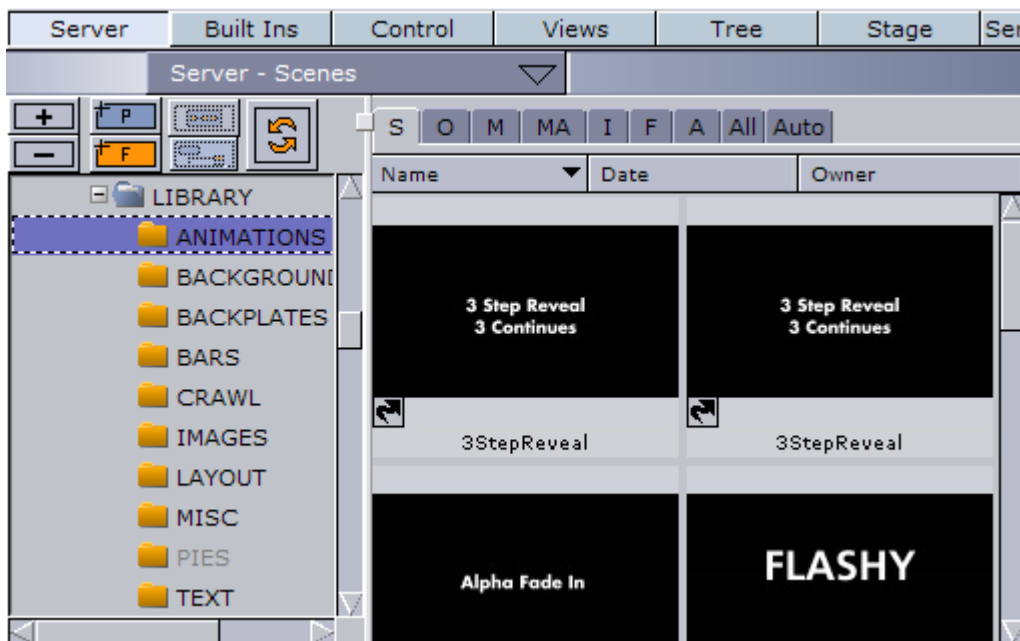
10. Select the *Control Object* icon in the *object* container to open the editor. Enter a description for the image object in the *Description* field. For example *New Image Object*.

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 below.



7.2.7 Creating Animations



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).

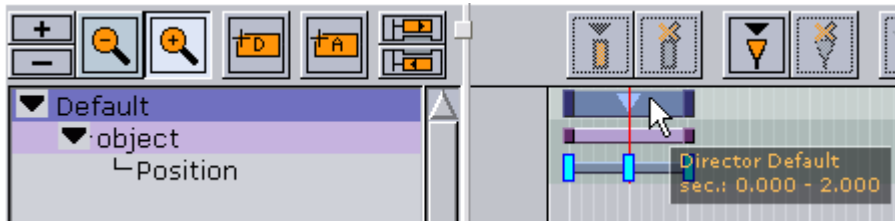


6. Verify that the **timeline** field in the timeline editor, seen above the Scene Editor, is set to 0 (fields).
7. Click the **Set Key** button (aka Update in Viz Artist 2). This creates a keyframe reflecting the scene as is. The number of fields specified in the timeline field is automatically updated to 50, which is the default interval.
8. In the Transformation editor, reset the Position X to 0.

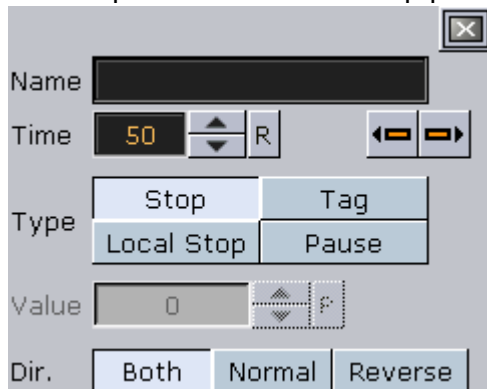
Tip: Click the **R** (*reset*) button to the right of the Position X field.

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's 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: Since 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 Playout.
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.

See Also

- [Viz Artist User Guide](#)

8 Creating Standalone Scenes

This section shows how to create a basic lower third as a standalone scene, which can be imported to Viz Trio to create a template for creating pages used for layout.

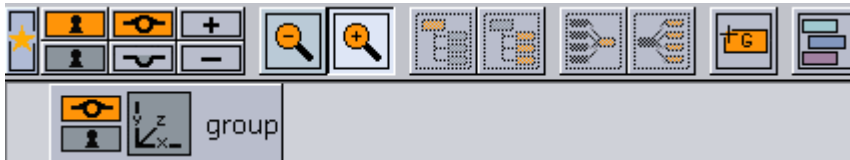
In order for the example below to work, follow each individual procedure in chronological order.

This section covers the following topics:

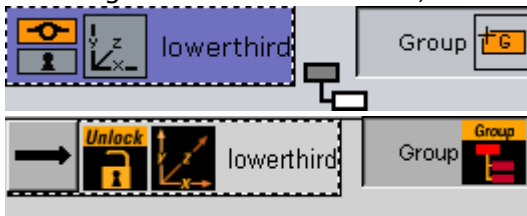
- [Creating a Scene](#)
- [Adding a Background](#)
- [Adding Text](#)
- [Creating an In Animation](#)
- [Creating an Out Animation](#)
- [Adding Stop Tags](#)
- [Adding Key Functions to the Container](#)
- [Adding Exposed Properties](#)
- [Editing Multiple Elements with a Single Value](#)

8.1 Creating A Scene

1. Start Viz Artist 2.x or 3.x, and create a blank scene.
 - For Viz Artist 3.x: Start Viz Artist and start creating the new scene.
 - For Viz Artist 2.x: Open the directory in the Viz Artist scene tree where the lower third scene will be saved. Create a new blank scene by pressing the **Add** button, and open the scene.



2. Add a group container to the tree.
 - For Viz Artist 3.x: Click or drag the Group icon labeled G (see image above).
 - For Viz Artist 2.x: Click the Function button, and click or drag the **Group** icon.
3. Rename the group, to *lowerthird* for example
 - To change the container's name, double click on the name section of the container.

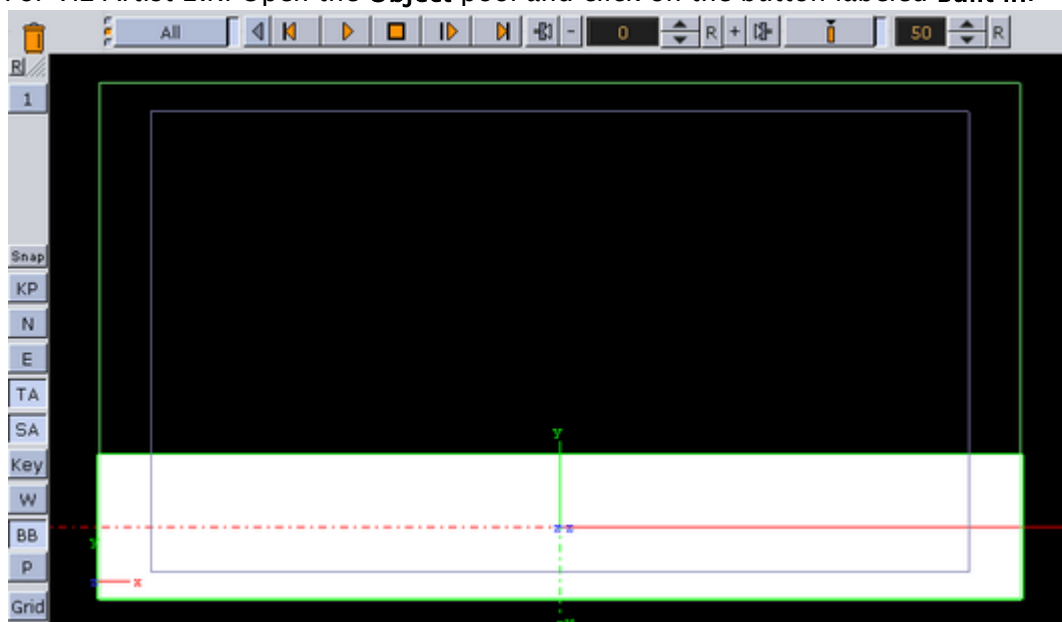


4. Add another group container as a sub-container to the **lowerthird** container.
 - To add a sub-container drag the group object to the right of the existing container.
5. Name the second group *object*.

8.2 Adding A Background

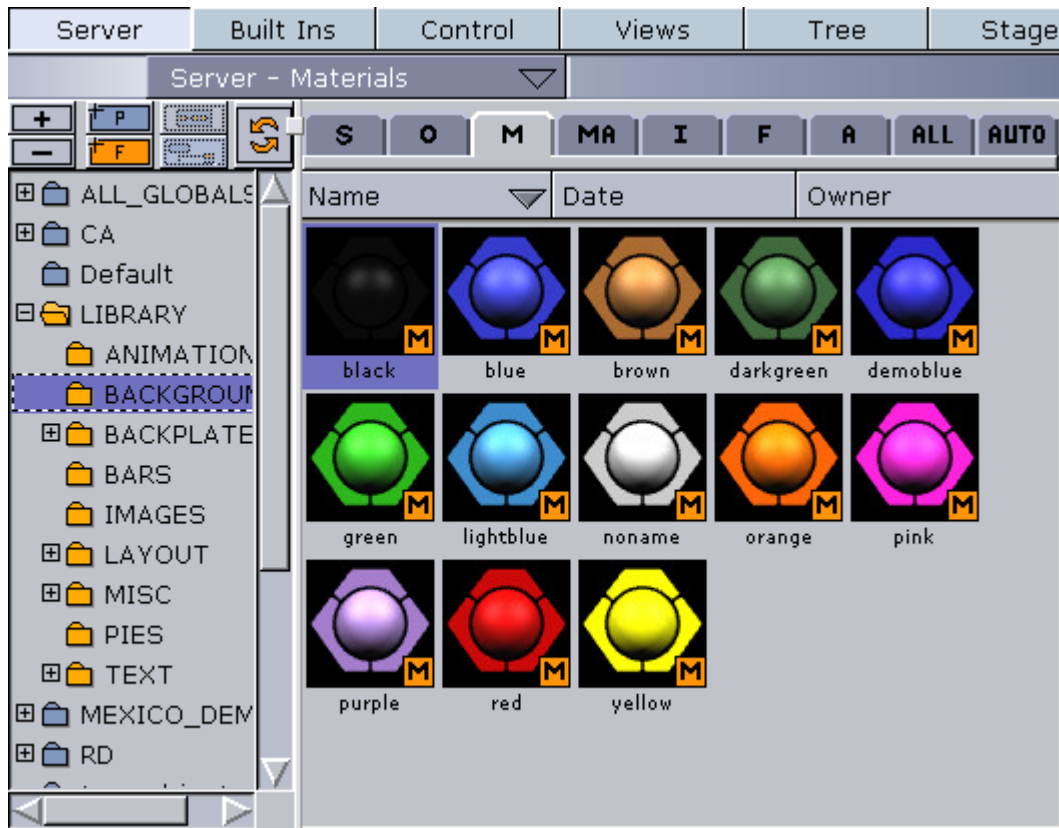


1. Add a rectangle from the **Built Ins** object pool and drop it as a sub-container of the **object** container. Rename it *Background*.
 - For Viz Artist 3.x: Click the **Built Ins** button, and select **Primitives** from the drop-list. The object can be found in the **Default** folder.
 - For Viz Artist 2.x: Open the **Object** pool and click on the button labeled **Built in**.



2. 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 part of the screen.
 - Rectangle editor: Width **800** and Height **125**.
 - Transformation editor: Position Y **-160**.
3. 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**.



- Add a material (see [To add materials](#)) object to give the rectangle a 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.
- Drag the desired material from the material pool and drop it onto the **Background** container.

8.3 Adding Text

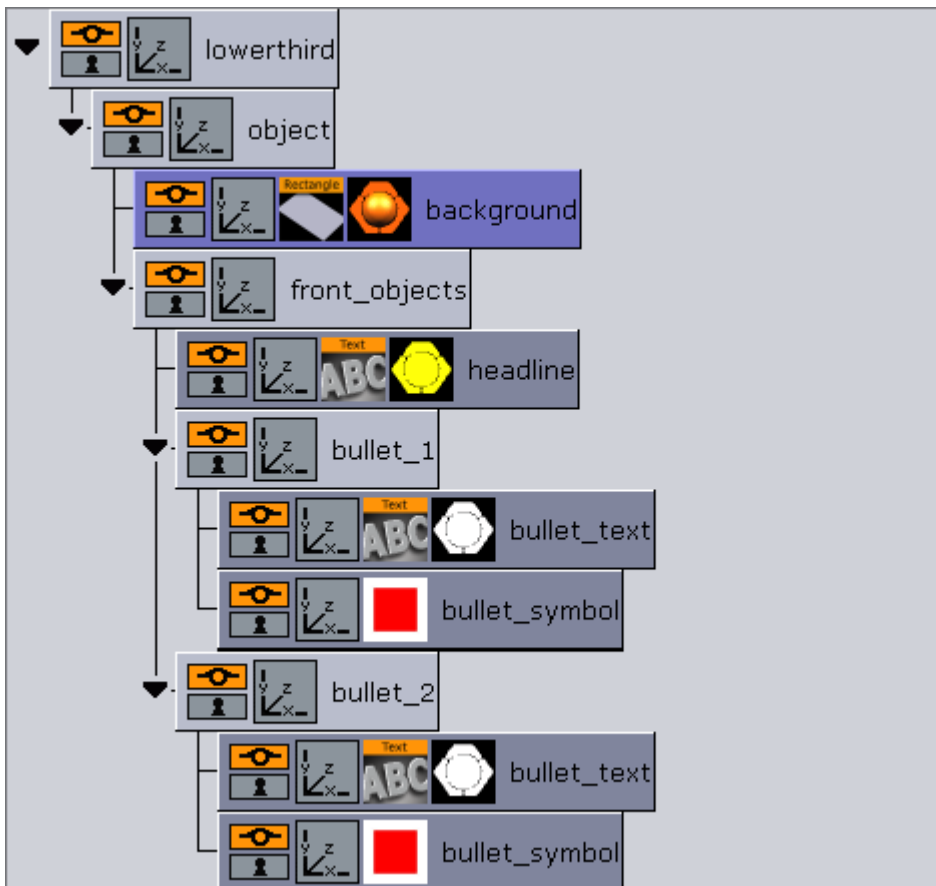


To enhance a basic lower third, you can add a headline element and two text lines, lead by a bullet symbol:

- Add a group as a sub-container to the object container, and name the group *front_objects*.
- 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 the 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.
- Add a material to the font container.
- Open the transformation editor. Scale and position the font so that the headline is placed in the upper left corner of the lower third background.

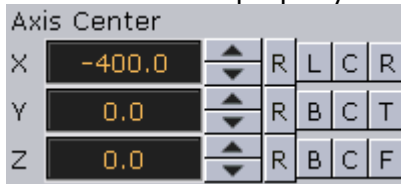
5. Add a group container as a sub-container of the **front_objects** group, and name it *bullet_1*.
6. As sub-containers of **bullet_1**, add a font and an image or object to act as a bullet symbol.
 - Name the text container *bullet_text* and the image *bullet_symbol*.
7. In **bullet_text**'s text editor enter the text "Bullet 1 text here".
 - Set the horizontal orientation to *Left*.
 - Add a material to the **bullet_text** container.
 - Scale and position the two containers reciprocally. Position the **bullet_1** container under the headline.
8. Copy the **bullet_1** container by dragging it 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**.

8.4 Creating An In Animation

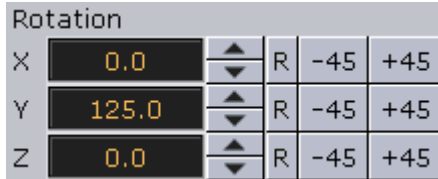


The **object** container must be animated in order to make the whole lower third rotate in from the left side of the screen. To make the rotation look right, the X-center of the container must be set to the left.

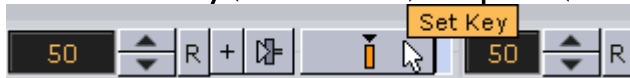
1. Open the transformation editor for the **object** container and click the **L** button to the right of the Axis Center X property.



2. Change the Rotation value for the Y-axis so the container moves out to the left side and is hidden.



3. Press the **Set Key** (Viz Artist 3.x) or **Update** (Viz Artist 2.x) button.



4. Set the Rotation value for the Y-axis back to zero by pressing the **R** (reset) button. Set a new key by repeating step 3 above. An animation object should now be visible on the **object** container. Play the animation in the render window.



8.5 Creating An Out Animation

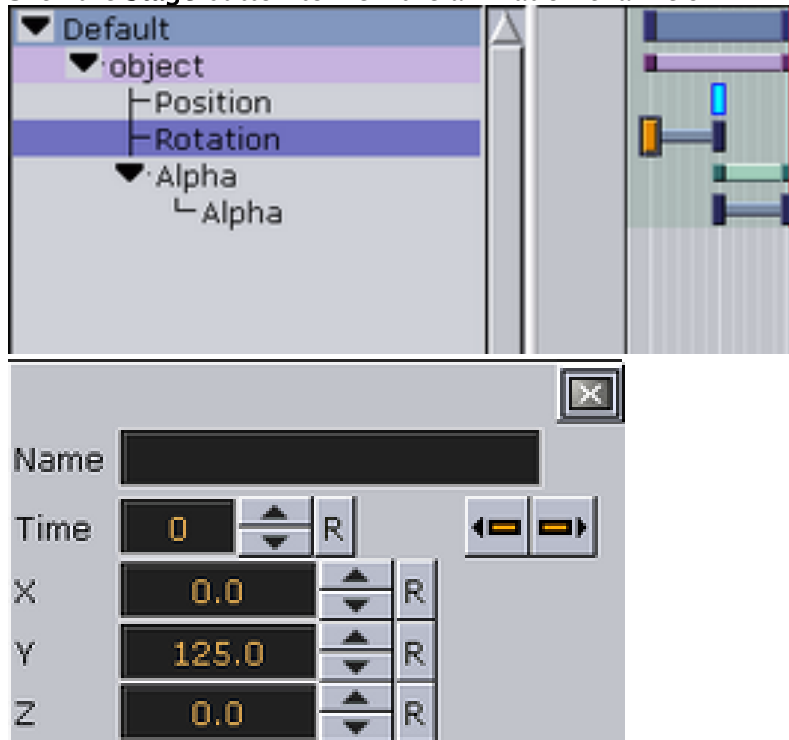


A simple fade out animation of the whole object can be created to make an out animation.

1. 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.
2. 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

1. Click the **Stage** button to view the animation channels:



2. You can move the keyframes to change the timing, drag them with your cursor, or click them to alter the time settings in the keyframe editor.

8.6 Adding Stop Tags

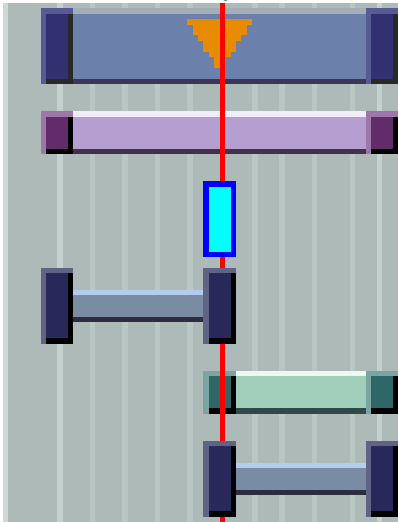
When playing out the scene, the objects should move in from the left and then fade out again. In order to stop the animation after the rotation, a stop point must be added before the animation can continue and fade out the scene.

1. Move the timeline (the thin red vertical line) to the second keyframe of the rotation channel (you do not need to be precise as this is easy to adjust afterwards).



2. Add a Stop keyframe to the Default director.
 - For Viz Artist 3.x: Click the **Add Stop/Tag** button above the stage editor.

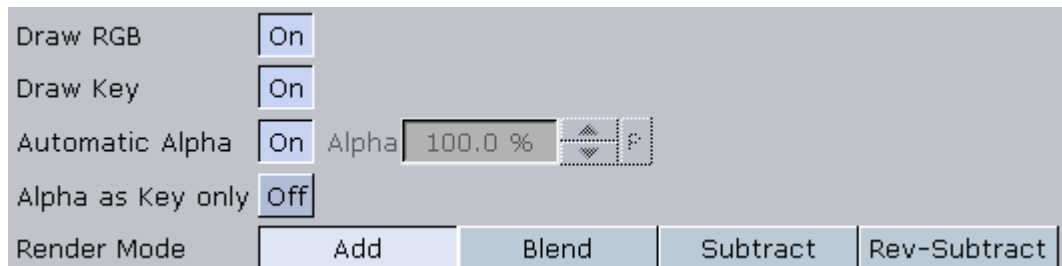
- For Viz Artist 2.x: Click the **Default** director label to open the Default director editor. Click the **Add Stop** button.



- Move the stop-point with your cursor or select it the stop-point and use the editor to move it.
- Ensure that it has the same time settings as the second keyframe on the rotation channel.
- Play out the animation to verify that it stops when the rotation has finished.
- Click the **Continue** button to proceed with the alpha fade.

8.7 Adding Key Functions To The Container

- Add a key function to the **background** container.
- 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 is in the **Global** folder.
 - For Viz Artist 2.x: Click the **Function** button.



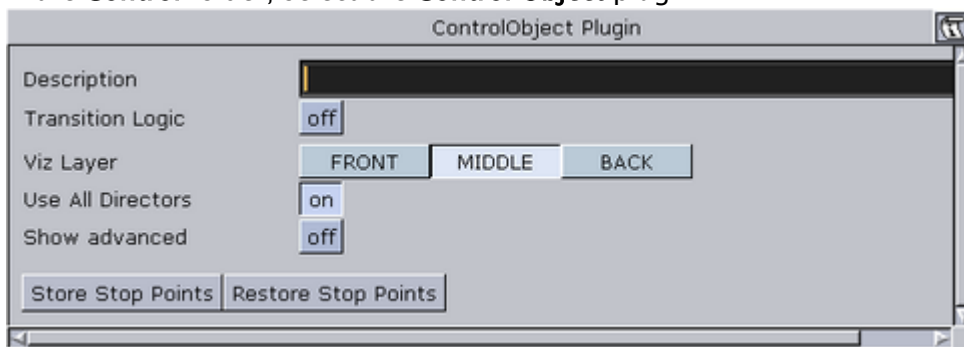
- The key for the **background** must have the **Alpha as key only** setting enabled; this prevents a "dirty" key if the background has some level of transparency.
- Set **Render mode** to **Blend**.
- Add a key function to the **front_objects** container. The **Alpha as key only** setting must be disabled, and **Render mode** must be set to **Add**, to prevent foreground objects from creating a hole in the background object's key signal.

8.8 Adding Exposed Properties



Control plug-ins must be added to make the scene ready for import into Viz Trio. Plug-ins allow scene properties to be visible to the Viz Trio operator.

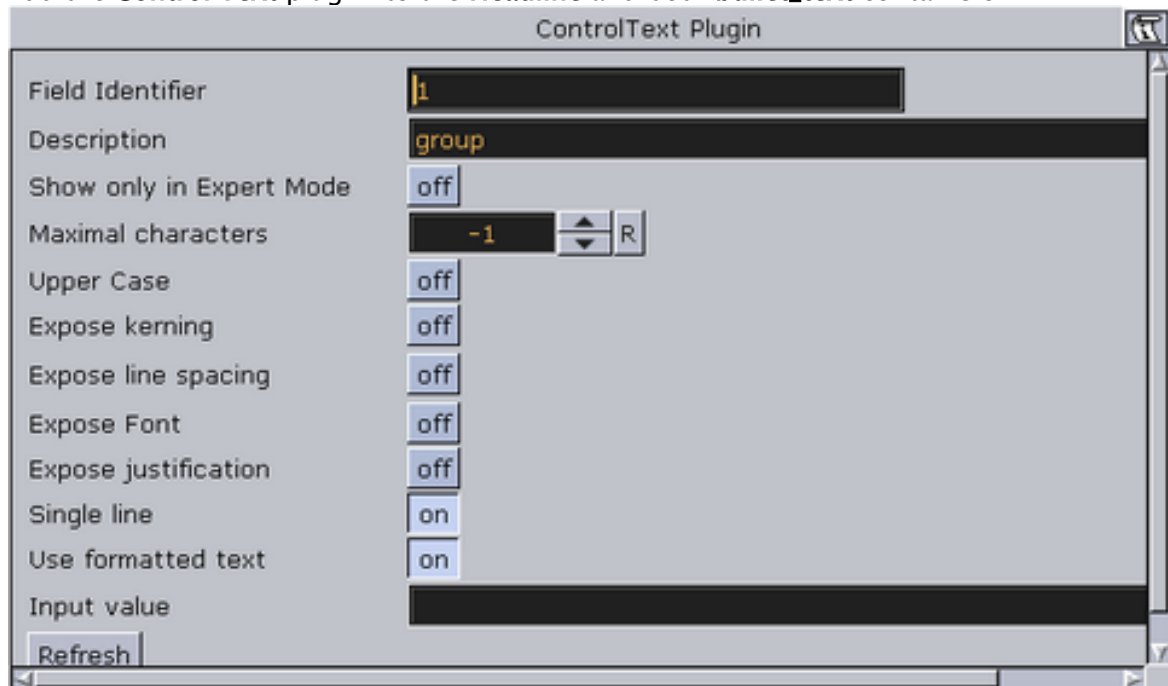
1. Add the **Control Object** plug-in to the **object** container.
 - For Viz Artist 3.x: Click the **Built Ins** button, and select **Function Container_** from the drop-list. The **Control Object** plug-in is in the **_Control** folder.
 - For Viz Artist 2.x: Click the **Function** button.
 - In the **Control** folder, select the **Control Object** plug-in.



2. Click the **Control Object** icon on the **object** container to open its editor.
 - Enter a description of the Viz Trio template (for example *Lower Third*).

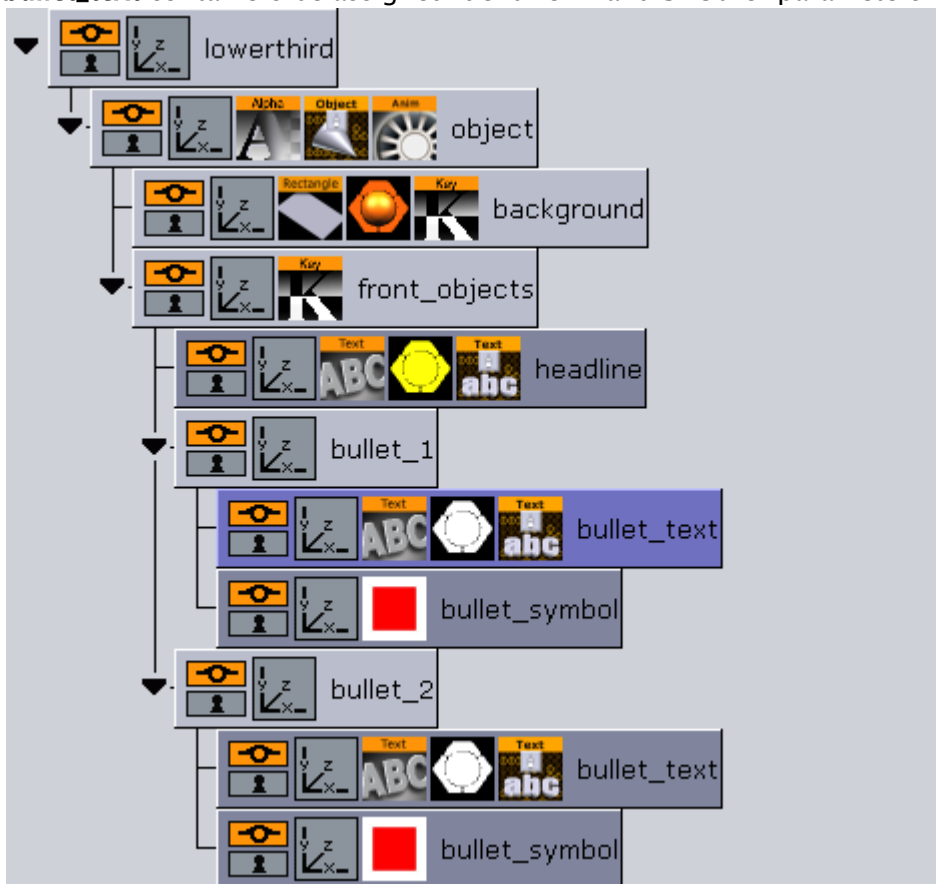


3. Add the **Control Text** plug-in to the **Headline** and both **bullet_text** containers.



4. Click the **Control Text** icon on the container to open its editor.
 - Set the **Field identifier** and **Description**. The Field identifier must be a numeric value; the value will be used to give the Viz Trio page a tab order.

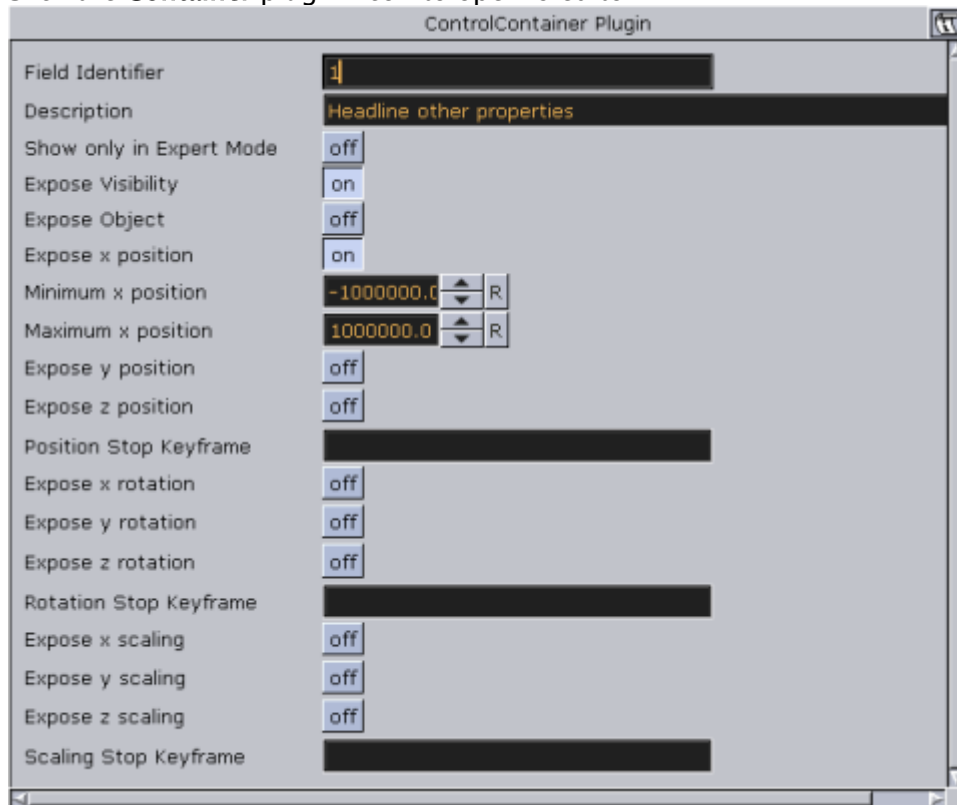
- In this scene, the **headline** container will typically be assigned identifier 1, and the two **bullet_text** containers be assigned identifier 2 and 3. Other parameters can be set.



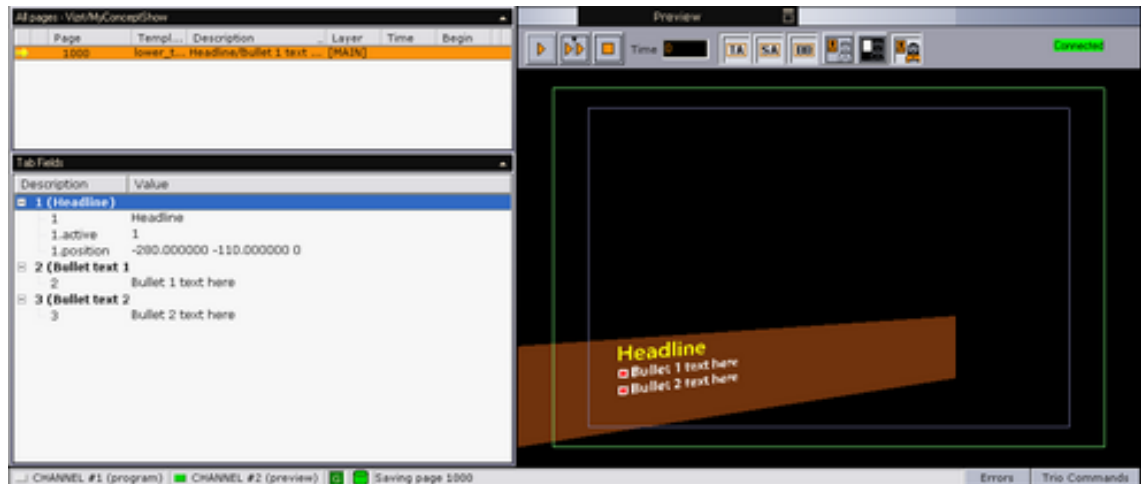
It's possible to expose more properties. By exposing the position, rotation, and scaling properties, the operator can hide or show an object and add material.

1. Add the **Control Container** plug-in to the container to be exposed (e.g. **headline**).

- Click the **Container** plug-in icon to open its editor.



- Enter a field identifier. If there is another control plug-in on that container, such as 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.



8.9 Editing Multiple Elements With A Single Value

Use the same field identifier for multiple control plug-ins; two or more fields will then be handled as a single editing element and will receive the same value.

⚠ Note: This can be used with a bar that scales and a text label that shows its value, for example. 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.

9 Creating Transition Effects

Use the built-in transition effect feature for stills and videos or create custom transition effects by designing scenes in Viz Artist.

This chapter covers the following topics:

- [Using Built-in Transition Effects](#)
- [Creating Transition Effect Scenes](#)
- [TransitionLayers](#)

9.1 Using Built-In Transition Effects

Built-in transition effects provide easy access to basic effects like cut and fade.

9.1.1 Configuring Built-in Transition Effects

- **For videos and stills:** select the **Use Transitions** checkbox in Configuration settings. Go to **File > Configuration > Output > Add Video**. For more information, see [Working With the Profile Configuration](#).
- **For graphic elements:** the Viz output mode in Configuration settings must be changed from "Standard" to "Scene Transitions". This menu option can be set in **File > Configuration > Output > Add Viz... "Mode"-option**.
- **For transitions between stills and videos:** the profile needs to define a channel for *both* video and graphic elements (Viz and Video output) or ensure that stills and video elements are played out on the same Viz Engine; transitions between video and graphic elements will otherwise not be possible.

⚠ Note: If one dual channel machine is being used for both program and external preview, the machine may not have enough capacity to show transitions on both channels. Should this occur, use a simpler preview scene for stills that does not support transitions. To enable this preview, the override concept **PREVIEW** must be specified for the external preview channel.

9.1.2 Using Built-in Transition Effects

To view the effects column, right-click the playlist column and select **Effect** in the menu to make the column visible. Transition effects can be chosen individually per element or as a default for the playlist. Select default for a playlist from the default effect dialog by clicking the **Fx..** button.

i Info: Default values are shown in the effect column with bracket values [...]. Individual effects are shown without brackets.

The effect column in a playlist shows different effect sets depending on the element type:

- still images and video elements (currently Cut and Fade).
- graphic elements (the scene transition effects).

In Trio version 3 and later, there are two additional macro commands for setting default effects for a show:

- `show:set_default_scene_effect` and
- `show:set_default_video_effect`

See Also

- [Show](#) in the Macro Commands and Events section.

9.2 Creating Transition Effect Scenes

The following section describes how to design custom transition effects for Viz Trio. The scene transition mechanism uses a hardware feature called Dynamic Scene. A transition scene is a scene that is created to move from one scene to the next in a dynamic way (and not by a cut).

In Viz Artist, dynamic textures are made using the **Dynamic Scene** plugin found under the **Media Assets** 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 since an actual scene can be loaded into them.

- **layer1**: Represents the first scene.
- **layer2**: Represents the destination scene for the transition.

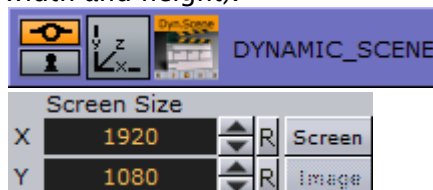
Although, it's possible to make any kind of animation, a few design conventions must be followed:

- 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.

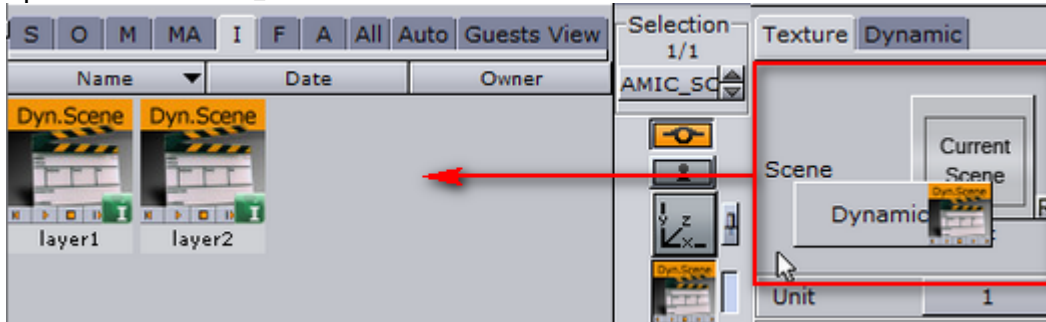
Note: All scenes must be placed in a **transitions** folder in the scene database to make them visible in the control application.

9.2.1 Creating Dynamic Textures

1. Start **Viz Artist**
2. Add the **Dynamic Scene** plugin to your Scene Tree.
3. Open the Dynamic Scene plugin's **Dynamic** tab and configure the properties (e.g. setting width and height).



4. Open the DYNAMIC_SCENE container's transformation editor and set **Screen Size** to **Screen**.



5. Open the **Server Area**
6. Create a folder at the root level and name it **dynamic**
7. Select the **Image (I)** tab
8. Drag the **Current Scene** placeholder icon into the Server Area and save two textures as layer1 and layer2.

Note: Icons may differ depending on which software version you are using.

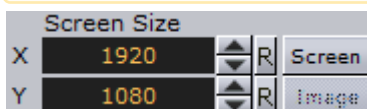
9.2.2 Creating a Transition Scene

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**.



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 ensures that the dynamic textures (layer 1 and layer2) brought under it are the correct full frame size.



5. Click the **Built Ins** button, select **Container Plugins (CP)** from the drop-list and then the **Global** folder.

6. Add the Alpha plug-in to the **Layer 1** and **Layer 2** containers.



7. 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 frame button above the Scene Editor.
 - Set the Alpha value to 0% and click the Set key frame button again.
8. Create a fade in animation for the **Layer 2** container. Repeat the steps for *Layer 1* in reverse order.
9. Open the **Server Area**
10. Create a folder at the root level and name it **transitions**
11. Save the scene to the transitions folder.

✓ **Tip:** Place the timeline somewhere in the middle before the scene is saved, as this usually provides the best representation of the transition effect. Transition scenes are commonly stored in a **transitions** scene folder at the root of the database directory structure.

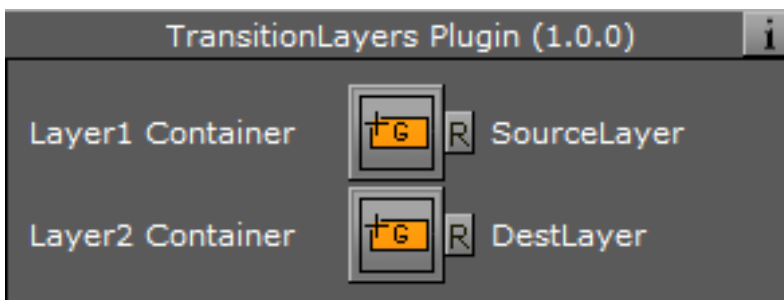
See Also

- [Viz Artist User Guide](#)

9.3 TransitionLayers

The TransitionLayers plug-in lets you define the two dynamic images of a scene-transition scene using the Dynamic Scene Media Asset (see [Creating Transition Effect Scenes](#)).

9.3.1 TransitionLayers Properties



- **Layer1 Container:** contains the scene *from* which to transition. The name of the referenced container is on the right.
- **Layer2 Container:** contains the scene *from to* which to transition. The name of the referenced container is on the right.

- **Reset button (R):** Each layer container has a reset button (R) that empties the container reference.

✓ **Tip:** Press the **group** button for the source of destination layers to select the corresponding container in the scene tree.

When adding the TransitionLayers plug-in to a container, the plug-in will automatically create the two sub-containers required and add these to the plug-in properties



1. Parent container containing the **TransitionLayers** plug-in.
2. Sub-container containing the scene *from* which to transition (or Layer1 Container).
3. Sub-container containing the scene *to* which to transition (or Layer2 Container).

In addition, a default two second alpha in animation will be added for the Layer2 Container to the *Default *direction in the Stage. The default animation can be changed as desired.

✓ **Tip:** Containers containing dynamic images for the transition scene can also be dragged onto the desired Layer container in the plug-in properties pane.

See Also

- [Viz Artist User Guide](#)

10 Scripting

Scripts 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 available to machines connected to the same Media Sequencer using the same show. Scripts can be assigned to templates and shows. Although it's only possible to assign one script per show or template you can include other scripts as part of the main script to extend its functionality.

10.1 Notes About Scripts

- All edits are done with the [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 affect all shows that use the same script.
- File scripts are read into the show each time a show is opened.
- Only templates can have scripts assigned. All instances of a template inherit the template script.

This section covers the following topics:

- [Viz Trio Scripting](#)
- [Viz Template Wizard Scripting](#)

 **Tip:** Always remember to escape backslashes correctly.

10.2 Viz Trio Scripting

A VBScript can be attached to any show or template to enable custom functionality. Typical applications for scripting include importing data from database sources and giving users guidance. All template instances with a script will inherit the script. Viz Trio supports [Macro Commands and Events](#) that may be used as part of the script.

This section covers the following topics:

- [Script Directory](#)
- [Script Editor](#)
- [Script Backup](#)
- [Script Errors](#)

10.2.1 Script Directory

All show scripts are stored on the Media Sequencer. However, a local directory is still needed for script that are stored as files:

- Open the Configuration window, and set the *Script Path* under the [User Interface/Paths](#) section.

The default script directory is %ProgramData%\Vizrt\Trio\scripts normally C:\ProgramData\Vizrt\Trio\scripts .

If a show is imported without its underlying scripted templates in the script folder, Viz Trio will create an empty script file. Since the script files are empty, you must reassign the scripts and restart Viz Trio to load the updated script memory.

Note: Versions of Trio prior to 3.0 store scripts in C:\Program Files (x86)\Vizrt\Viz Trio\scripts. Script files for a scripted show or template are copied to the Viz Trio script folder.

10.2.2 Script Editor



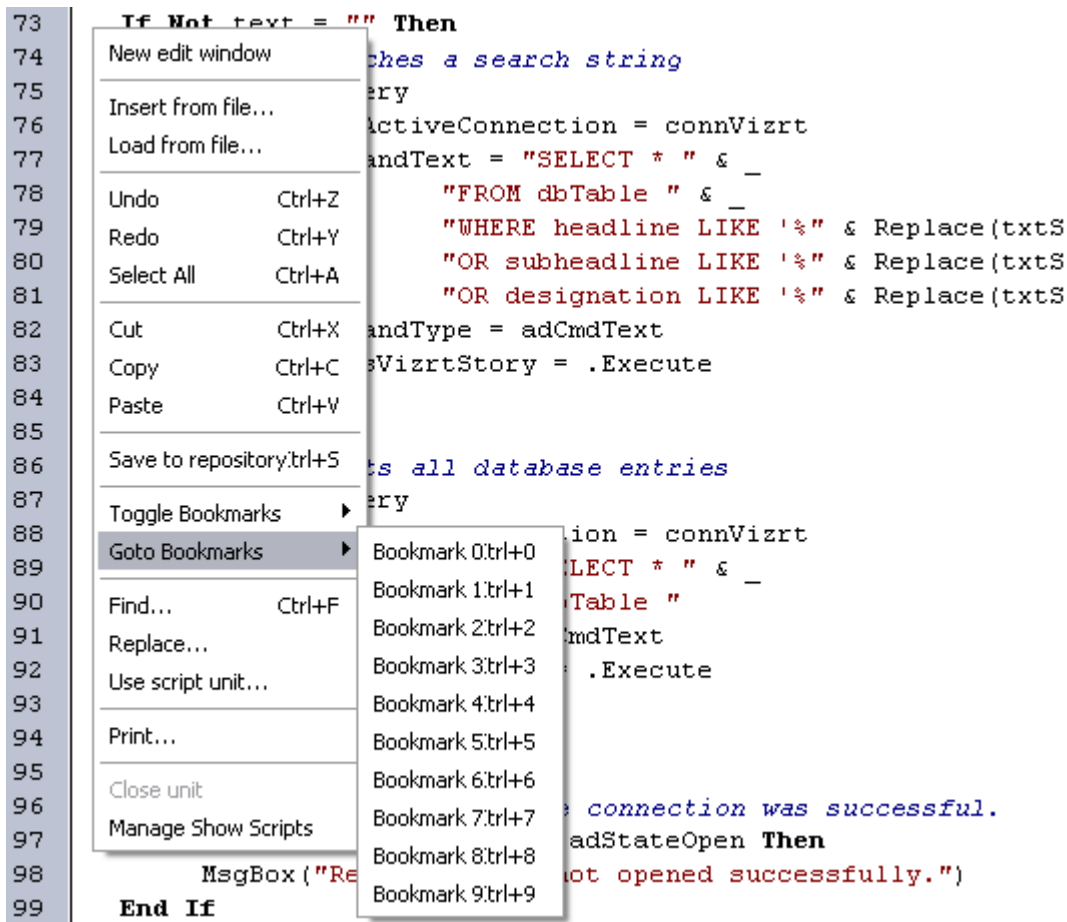
- **Predefined Functions** - Opens the [Predefined Functions](#) window containing [Commands](#) and [Events](#) that can be used in scripts.
- **Show Library Scripts** - Opens the [Library Scripts](#) window displaying scripts which can be attached to a template as a 'USEUNIT reference.
- **Syntax check** - Checks syntax correctness.
- **Save script** - Saves a script to the server. You can also save the script by right-clicking inside the script window and selecting **Save to repository** from the [Context Menu](#), where you can also save units to file.
- **Manage show scripts** - Opens the [Script Manager](#) for the show that's open.
- **Close all units** - Closes all script units.
- **Change font** - Changes font settings for the script editor.
- **Combo box** - Jumps between script functions and procedures within the same script.

This section covers the following topics:

- [Context Menu](#)
- [Shortcut Commands](#)
- [Predefined Functions](#)
- [Library Scripts](#)
- [Script Manager](#)
- [Assigning a Script to a Show](#)
- [Assigning a Script to a Template](#)
- [Adding a Predefined Function](#)

- [Adding a Library Script Unit](#)
- [Editing a Show Script](#)
- [Editing a Script](#)
- [Executing a Script](#)

Context Menu



- **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:** Toggles between inserting and removing a bookmark at the 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 where you can import, export, and delete a script from a template.

Shortcut Commands

- **CTRL + TAB:** Switches between the open scripting tabs.
- **CTRL + SPACE** - Displays the Code Insight Window containing a list of parameters, variables, functions, etc. available to the currently open script or in attached sub scripts.
- **CTRL + SHIFT + SPACE** - Displays code completion hints.
- **CTRL + Left mouse button** - Opens the highlighted function; works across scripts and scripts units, but not within the same script. For this to work, the ' *USEUNIT <scriptname>* notation must be added to the script 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 contains a set of [Events](#) that can be added to scripts.

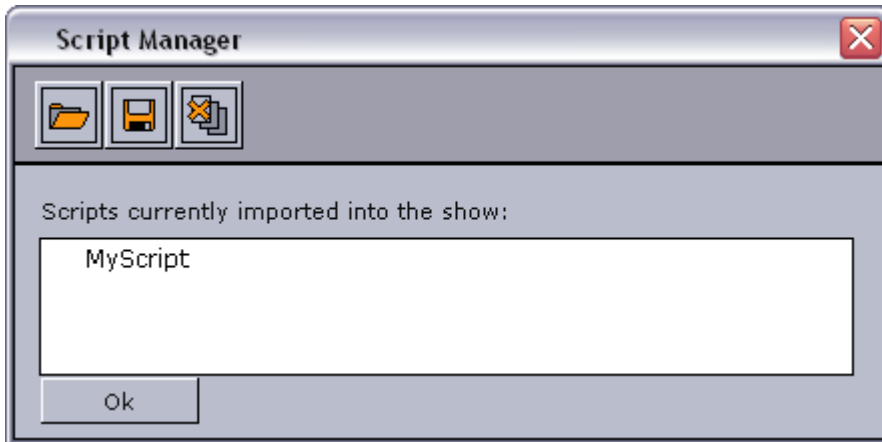
Note: TrioCmd() supports UTF16. If you need UTF8 support, you can use TrioCmdUTF8().

Library Scripts



Script units can be added to an assigned template script, letting you 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 Graphic Hub settings.

Script Manager



The **Manage Show Scripts** button opens a window where you can import, export, and delete VBScript files from the current show.

Assigning a Script to a Show

1. Open the [Show Properties](#).
2. Click the **browse** button next to the Show Script field.
3. From the Choose Show Script window select either a file script (local or shared) or a Media Sequencer script.
4. Click **OK**.

Assigning a Script to a Template

1. Right-click 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 store the script on the Media Sequencer or to a local or shared script path (see Graphic Hub).
4. Click **OK**.

Adding a Predefined Function

1. Open a template with an assigned script.
2. Click **Edit script** (see [Controls](#)).
3. Click **Predefined functions** in the [Script Directory](#).
4. Select a function, and click **Add**.

Adding a Library Script Unit

1. Open a template with an assigned script.
2. Click **Edit script** (see [Controls](#)).
3. Click **Library scripts** in the [Script Directory](#).
4. Select a script unit to use, and click **OK**.

Editing a Show Script

- Right click the Template list pane and select **Script** and **Edit Show Script** from the menu.

Editing a Script

Open a template with an assigned script, and click **Edit Script** (see [Controls](#)).

Executing a Script



Open a template with an assigned script, and click **Execute Script** (see [Controls](#)).

10.2.3 Script Backup

It's recommended to save scripts created for a specific show together with the show on Media Sequencer. A backup of the Media Sequencer files is therefore necessary.

10.2.4 Script Errors

If the script contains an error, the script window will automatically open and highlight the script error location. The script error is also reported in the [Error Messages Window](#).

10.3 Viz Template Wizard Scripting

In Viz Trio, a Viz Template Wizard template can be used to extend a show's functionality.

This section covers the following topics:

- [Dynamically Adding Components](#)
- [Setting and Getting Component Values](#)
- [Setting and Getting Show Values](#)

10.3.1 Dynamically Adding Components

Viz Trio supports the `CreateVTWComponent` function in Viz Template Wizard for dynamically adding components at run-time.

An example of a text box and a button being used to generate a label at run-time follows below:

```

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

```

10.3.2 Setting and Getting Component Values

Viz Trio has support for the `SetUnicodeValue` and `GetUnicodeValue` functions in Viz Template Wizard.

An example of a text box being used to get and set a value in another text box without using `TrioCmd` follows below:

```

Sub TWUniEdit1Change(Sender)   Text = GetUnicodeValue(TWUniEdit1)   SetUnicodeValue
TWUniEdit2, Text End Sub

```

10.3.3 Setting and Getting 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's 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:g
etdescription")   msgbox returnvalue End Function

```

In the example about, the second command `vtwtemplate:run_vtw_script` will be triggered within `GetDescription` and return the description value.

11 Appendix

This section covers the following:

- [Enabling Windows Crash Dumps](#)
- [Logging](#)
- [Cherry Keyboard](#)

11.1 Enabling Windows Crash Dumps

Log files can be a valuable tool in understanding and analyzing unexpected program behavior. In addition to Vizrt log files, it's recommended to allow Microsoft Windows to generate User-mode crash dumps. This can make debugging easier, particularly if there are any hardware or general Windows issues affecting program behavior.

Enabling User-mode Dumps requires Windows 7 or higher. To enable, configure the following registry setting:

```
HKEY\LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps
```

For more information, see the Microsoft page [Collecting User-Mode Dumps](#).

11.2 Logging

Viz Trio log files, together with log files on the Media Sequencer and Viz Engine, are crucial in helping Vizrt support troubleshoot any issues related to our products.

This section covers the following topics:

- [Viz Trio Log Files](#)
- [Viz Trio Error Messages](#)
- [Viz Engine and Media Sequencer Log Files](#)

11.2.1 Viz Trio Log Files

Viz Trio Log Levels

Viz Trio Log Levels can be specified to define the level of detail. For example, log level 0 provides information about critical errors only, while log level 9 will provides information about a wide range of events.

Specify log levels in the [General](#) section of Configuration:

- **Loglevel 0:** Only error messages will be logged.
- **Loglevel 1:** Warning messages will be added.

- **Loglevel 2:** Timer messages will be added. These show the amount of time different operations take in the program.
- **Loglevel 5:** Status messages will be added. These show ordinary program events such as "page loaded", "page taken", and so on.
- **Loglevel 9:** All commands sent to the Viz Engine rendering process will be added.

✓ **Tip:** Messages on log level 0 and 1 (error and warning messages) can also be viewed in the [Error Messages Window](#).

⚠ **Caution:** In many cases it may be beneficial to set a low Media Sequencer logging level, preferably to 0, since a high level of logging might affect performance when running in a production environment.

Changing Log Level

1. Click the menu option **File > Configuration** to open the Trio Configuration.
2. Select [General](#) from the [User Interface](#) section.
3. In the **Log level** box, enter a log level according to the available [Viz Trio Log Levels](#).

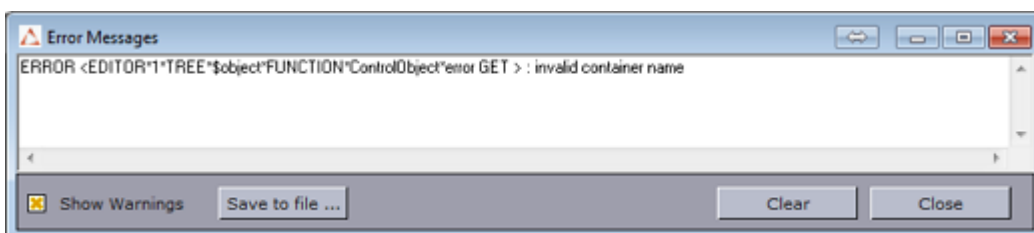
Changing the Log File Path

1. Click the **Config** button in Viz Trio's main window to open Trio Configuration.
2. Select [Paths](#) from the [User Interface](#) section.
3. In the **Logfile path** box, enter the path to where the log files should be located from now on.

11.2.2 Viz Trio Error Messages

Messages on log level 0 and 1 (error and warning messages) can be viewed in the Error Messages Window as well as in log files.

Error Messages Window



This section covers the following topics:

- [Viewing Error Messages](#)
- [Viewing Warnings](#)
- [Saving Error Messages Window Content](#)

Viewing Error Messages

- Click the **Errors** button in the lower right corner of the Viz Trio main window. The Error Messages Window then opens.

Viewing Warnings

1. Click the **Errors** button in the lower right corner of the Viz Trio main window.
2. In the Error Messages Window that opens, select the **Show Warnings** check box.

Saving Error Messages Window Content

1. Click the **Errors** button in the lower right corner of the Viz Trio main window.
2. In the Error Messages Window that opens, click **Save to file**.
3. In the **Save As** dialog that opens, define the path and enter a descriptive name for the .txt file.


11.2.3 Viz Engine and Media Sequencer Log Files

In addition to Viz Trio's log files created on the client side, it's recommended to read and, if required, send log files to Vizrt support that were created on Media Sequencer and Viz Engine.

Viz Engine Logs

Viz Engine log files are located in the program folder:

Viz Engine 3: C:\ProgramData\Vizrt\viz3

 **Example:** VizRender_1159345015.log

Media Sequencer Config Files

The default.xml files are stored here:

Win Vista and 7/8: C:\ProgramData\Vizrt\Media Sequencer Engine

Media Sequencer Logs

The log level for Intelligent Interface specific messages on the Media Sequencer can be set using Intelligent Interface (IIF).

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. Log levels are defined as integers in the range 0-100. At a given log level, all the information specified for lower log levels is also logged.

The log level can be changed while the Media Sequencer is running, and the logging output immediately reflects the new log level.

Info: For more information on how to set log levels for the Media Sequencer, see the Media Sequencer documentation.

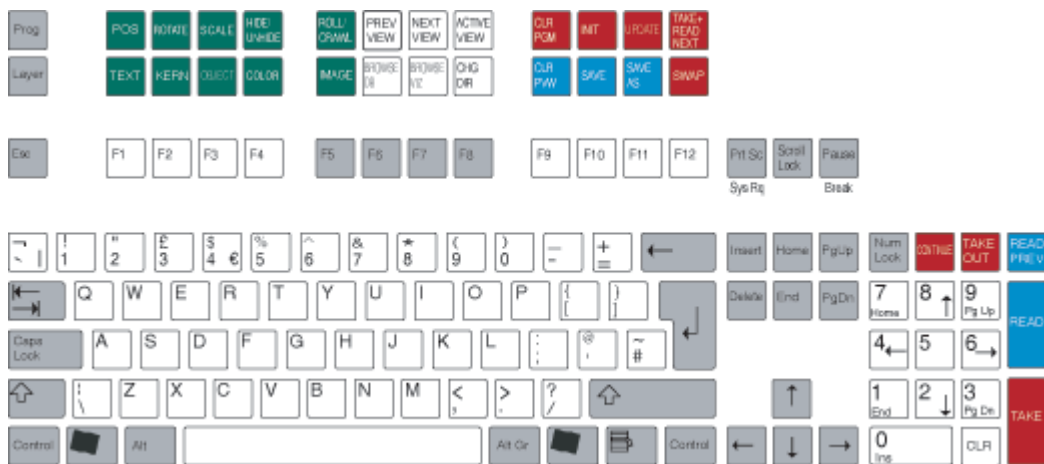
Caution: Remember to set a low log level, preferably to zero, since a high level of logging might affect performance when running in a production environment.

All Predefined Log Levels

Name	Level	What's 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.
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.

Name	Level	What's logged
Output	70	Data sent 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 running times of the executed actions.
Trace	95	The various stages the internal operations of the system perform during execution.
All	100	All possible logging information, including operation scheduling and loop iteration, and running state statistics.

11.3 Cherry Keyboard



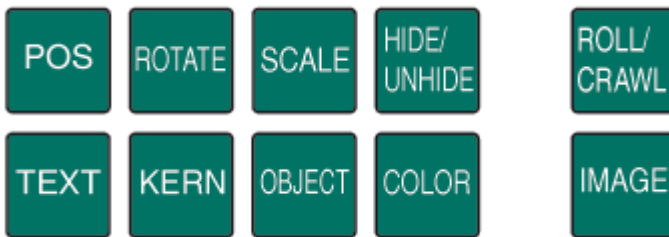
Early versions of Viz Trio shipped with an old version of the Cherry Keyboard, which can still be used with current versions of Viz Trio. The keyboard contains two rows with extra function keys which have been assigned to different Viz Trio actions.

Note: 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 there is normally there is no need to change these settings.

This section covers the following topics:

- [Editing Keys \(green\)](#)
- [Navigation Keys \(white\)](#)
- [Program Channel Keys \(red\)](#)
- [Preview Channel Keys \(blue\)](#)
- [Program and Preview Channel Keys \(red and blue\)](#)

11.3.1 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/CRAVL:** Opens the scroller editor
- **IMAGE:** Opens the image pool.

11.3.2 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.

11.3.3 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, pressing 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 of what is currently On-Air and reads that page again.

11.3.4 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.

11.3.5 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.

See Also

- [Viz Trio Keyboard](#)
- [Keyboard Shortcuts and Macros](#)