Viz Mosart Administrator Guide

Version 5.6

Viz Mosart

Copyright ©2024 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. his publication may contain technical inaccuracies or typographical errors. While every precaution has been taken in the preparation of this document to ensure that it contains accurate and up-to-date information, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained in this document. Vizrt’s policy is one of continual development, so the content of this document is periodically subject to be modified without notice. These changes will be incorporated in new editions of the publication. Vizrt may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.
Vizrt may have patents or pending patent applications covering subject matters in this document. The furnishing of this document does not give you any license to these patents.

Antivirus

Vizrt does not recommend or test antivirus systems in combination with Vizrt products, as the use of such systems can potentially lead to performance losses. The decision for the use of antivirus software and thus the risk of impairments of the system is solely at the customer's own risk.

There are general best-practice solutions, these include setting the antivirus software to not scan the systems during operating hours and that the Vizrt components, as well as drives on which clips and data are stored, are excluded from their scans (as previously stated, these measures cannot be guaranteed).

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

2024/03/27

Contents

1 Introduction 13
1.1 Related Documents 13
1.2 Feedback and Suggestions 13
2 About Viz Mosart 14
2.1 Naming Convention for Components 15
2.2 The Viz Mosart System 17
2.2.1 Viz Mosart Applications 18
2.3 Viz Mosart Server 19
2.3.1 Viz Mosart Server 19
2.3.2 Viz Mosart Server Components 20
2.4 Viz Mosart Client 29
2.4.1 Viz Mosart GUI 29
2.4.2 NRCS Plugin 30
2.4.3 Timing Display 31
2.4.4 Rundown Viewer 32
2.4.5 Audio Panel 34
2.4.6 Audio Player 35
2.4.7 iNews Timer 35
2.5 Other Viz Mosart Applications 36
2.5.1 Media Router and Mosart Template Database 36
2.5.2 Omnibus Controller 36
2.5.3 SNMP Service 36
2.5.4 Test Suite 36
3 Installation 38
3.1 Prerequisites 39
3.1.1 System Requirements 39
3.1.2 Hardware Requirements 41
3.1.3 Licensed and Optional Features 44
3.2 Viz Mosart Installation Files 46
3.2.1 Viz Mosart Installer Files 46
3.2.2 Additional Mosart Files 47
3.2.3 Installation and Upgrade 48
3.2.4 Version Numbering 49
3.3 Licensing 51

3.3.1 License Management 51
3.3.2 License Configuration status 51
3.3.3 License Notifications 52
4 AV Automation 53
4.1 Overview 53
4.1.1 AV Automation Tasks 53
4.1.2 AV Automation Characteristics 54
4.2 Menu Selections 55
4.3 Logging and Status 56
4.3.1 Logging Panel 56
4.4 AV Automation Device Properties 58
4.4.1 Vision Mixer 60
4.4.2 Video Servers 64
4.4.3 Graphics 87
4.4.4 Audio 94
4.4.5 Camera Robotics 102
4.4.6 Light Control 104
4.4.7 Router 105
4.4.8 Subtitles 106
4.4.9 GPIO 107
4.4.10 Tally 112
4.4.11 Virtual Set 113
4.4.12 Weather 114
4.4.13 Video Wall 115
4.4.14 General 116
4.4.15 Vizrt Graphics 118
4.5 Audio and Video Setup 119
4.5.1 Working with Audio and Video Setup 119
4.6 Device Connection Strings 126
4.6.1 Video Server, MAM and Database Connection Strings 127
4.6.2 Video Wall Connection Strings 173
4.6.3 Device Configuration Files 176
4.7 Template Editor 201
4.8 Mosart Instrumentation Panel 202
4.9 Recording 203
4.9.1 Working with Recordings 203
4.9.2 Supported Video Servers 205

5 Manus Administrator 207
5.1 NRCS Software Compatibility 207
5.2 Configuration Utilities 209
5.2.1 Centralized Settings Menu 209
5.2.2 Newsroom Settings 209
5.3 Settings Editor iNews 210
5.3.1 Working with iNEWS Settings 210
5.4 Settings Editor MOS 221
5.4.1 Working with MOS Settings 221
5.4.2 Notes 232
5.5 Configuring the MSE Playlist Panel 233
5.5.1 Managing the MSE Playlist Panel 233
5.6 Newsroom Settings Editor 234
5.6.1 Working with the Newsroom settings 235
5.6.2 Viz Mosart Object Structure 244
5.6.3 Viz Mosart and Viz Content Pilot (VCP) MOS Item Integration 248
5.6.4 MOS ID Mapping 252
5.6.5 Additional Newsroom Settings Examples 253
5.7 Field Mapping 259
5.7.1 The Tool Set 259
5.7.2 What to Configure 259
5.7.3 Target Fields 259
5.7.4 Mapping Source Fields to Target Fields 260
5.7.5 Examples 261
5.8 Manus Administrator Commands 265
6 Media Administrator 267
6.1 Media Administrator Commands 268
6.2 Media Administrator Properties Editor 269
6.2.1 Sections of the Mosart Media Administrator Properties File 271
6.2.2 Dynamic Video Server Configuration 273
7 Overlay Graphics 275
7.1 Overlay Graphics Interface 276
7.1.1 [1] List of Overlay Graphics in the Rundown 277
7.1.2 [2] Server Status 279
7.1.3 [3] General Log 279
7.1.4 [4] Event Log 279

7.1.5 System Information ... 280
7.2 Menu Bar ... 280
7.3 Overlay Graphics Configuration and Use ... 281
7.4 Managing Overlay Graphics Controllers and Engines ... 282
7.4.1 Overlay Graphics Controllers and Engines ... 282
7.5 Overlay Graphics Configuration Property Tabs ... 287
7.5.1 Working with Overlay Graphics Configurations ... 287
7.6 Mosart Graphics Destinations and Takeout Logic ... 293
7.6.1 Mosart Graphics Destinations ... 293
7.6.2 Takeout Logic ... 296
7.7 Overlay Graphics Types ... 299
7.7.1 VIZRT ... 299
7.7.2 VIZ FLOWICS ... 301
7.7.3 CHYRON ... 302
7.7.4 DEKO ... 303
7.7.5 ORAD ... 303
7.7.6 PIXEL POWER CC ... 304
7.7.7 XPRESSION ... 305
7.7.8 BIGVIZ ... 305
7.8 Sending Lower Third Graphics MOS Object to an NRCS ... 307
7.8.1 Managing Destinations ... 307
7.8.2 Detailed Description of MOS Objects ... 308
8 Show Design and Template Editor ... 309
8.1 Overview ... 310
8.1.1 Template Sets ... 310
8.1.2 Working with Templates ... 311
8.1.3 Template Terms and Definitions ... 312
8.2 Template Examples ... 314
8.2.1 Studio Camera ... 314
8.2.2 Package: Video Clip with Full Sound ... 315
8.2.3 Voice-over ... 316
8.2.4 Live External Source ... 317
8.2.5 DVE ... 318
8.2.6 Graphic: Full Screen Graphics ... 321
8.2.7 Accessory Templates ... 322
8.2.8 Switcher Plugin ... 324
8.3 Working with Templates ... 333

8.3.1 Template Operations 334
8.3.2 Template Properties 336
8.3.3 Control Commands in Templates 341
8.3.4 Destination Panel 363
8.3.5 Mosart Templates Feedback to the NRCS 364
8.3.6 Controlling a REST-based Driver 371
8.4 Other Template Functionality 379
8.4.1 Additional Template Features 379
8.5 Working with Template Sets 383
8.5.1 Template Set Types 383
8.5.2 Template Set Operations 385
8.5.3 Template Set Properties 388
8.6 Newsroom Tags 391
8.6.1 Adding a Newsroom tag to your Viz Mosart Template 392
8.6.2 Newsroom Tags Order 394
8.6.3 Advanced Features for Specific NRCSs 395
8.7 Template Device Functions 402
8.7.1 Working with Device Functions 402
8.7.2 Testing a Template 404
8.7.3 Video Switcher Crosspoint 405
8.7.4 Video Switcher Transition 407
8.7.5 Video Switcher Register/Timeline Recall 409
8.7.6 Macro Recall 411
8.7.7 Video Switcher Auxiliary Bus Delegation 414
8.7.8 Keyfill: Video Switcher Key Bus Delegation 417
8.7.9 Video Server 421
8.7.10 Graphics 423
8.7.11 Video Wall Register Recall 425
8.7.12 Audio Player 426
8.7.13 Virtual Set 427
8.7.14 Robotic Camera Control 428
8.7.15 GPO 432
8.7.16 Light Control 432
8.7.17 Router Control 436
8.7.18 Audio Faders 437
8.7.19 Generic REST commands 443
9 Operational Examples 444

9.1 Viz Mosart Settings and Configuration 445
9.1.1 Working with the Settings Menu 445
9.1.2 Generic REST Drivers 447
9.1.3 Vizrt Graphics 454
9.2 Story Recorder Mode 457
9.2.1 Enabling Story Recorder mode 457
9.2.2 Prerequisites 457
9.2.3 Overview 457
9.2.4 Activating Story Recorder Mode 458
9.2.5 Verification 472
9.2.6 Troubleshooting 472
9.2.7 Housekeeping 478
9.3 Viz Engine as a Switcher 480
9.3.1 Engine Switcher Mode 480
9.3.2 Features of the Engine Switcher 481
9.3.3 Requirements and Limitations 482
9.3.4 Installation 483
9.3.5 Configuration 483
9.3.6 Template Design 490
9.3.7 Custom Content Transitions 498
9.3.8 Labels 501
9.3.9 NDI Clip Preview 504
9.3.10 Summary of Configuration Points 506
9.3.11 Clean and Dirty-Feed Setup 508
9.4 Media Sequencer to Power Vizrt Graphics 517
9.4.1 Configuring Vizrt Graphics 517
9.4.2 Main/Backup Topology 518
9.5 Audio Players 520
9.5.1 Default Viz Mosart Audio Player 520
9.5.2 Third Party Audio Player Devices 521
9.5.3 Working with Audio Players 524
9.6 Viz Arc AR Support in a Mosart Show 529
9.6.1 Configuration and Setup 529
9.7 Frame Accurate System Operations 536
9.7.1 Why Frame Accuracy Matters 536
9.7.2 Requirements 537
9.7.3 Enabling Frame Accuracy Operation 537

9.7.4 Calibrating the System ... 540
9.7.5 Genlock Logging ... 543
9.7.6 Frame Accuracy in Story Recorder ... 544
9.7.7 Limitations ... 544
10 Viz Mosart Template Database ... 545
10.1 Architecture ... 545
10.2 Template Properties ... 545
10.3 Information Stored in the Template Database ... 546
10.4 Prerequisites for Connecting to TDB ... 547
10.5 Installing the Viz Mosart Template Database ... 548
10.5.1 MySQL Database Servers and Clients ... 548
10.5.2 MySQL Users ... 549
10.5.3 Initial Mosart Template Database ... 549
10.6 Creating and Upgrading a Viz Mosart Template Database ... 551
10.6.1 Upgrading MySQL 5.x to MySQL 8.x ... 551
10.6.2 Verifying the WampServer ... 551
10.6.3 Running Database Scripts ... 552
10.6.4 Running Scripts Automatically ... 552
10.6.5 Running Scripts Manually ... 553
10.7 Configuring Viz Mosart Server ... 554
10.7.1 Connecting AV Automation to the Mosart Template Database ... 554
10.7.2 Connecting Manus Administrator to the Mosart Template Database ... 558
10.7.3 Viz Mosart Template Database and Mosart Web Apps NRCS Plugin ... 559
10.7.4 Viz Mosart Template Database and Viz Mosart ActiveX ... 559
10.8 Backup and Recovery ... 561
10.8.1 Working with the Server Database ... 561
10.8.2 Working with Database Clients ... 562
10.9 Database Maintenance ... 565
10.9.1 Upgrading the Database ... 565
10.9.2 Viz Mosart Template Editor ... 566
10.9.3 TemplateSetEditor ... 566
10.10 Viz Mosart Template Database Specification ... 575
10.10.1 Entity Relationships ... 575
10.10.2 Introductions and Notations ... 576
10.10.3 Viz Mosart Database Tables ... 579
11 Maintenance ... 599
11.1 System Logging ... 600

11.1.1 Log Viewer ... 600
11.1.2 Log File Characteristics ... 601
11.1.3 Log Content ... 602
11.1.4 Log Adapters ... 604
11.1.5 Configuring Logging ... 605
11.1.6 Starting-up a Log Adapter ... 609
11.1.7 Troubleshooting Log Adapters ... 610
11.2 Server Maintenance ... 612
11.2.1 Server File Structure ... 612
11.2.2 Files for Backup ... 614
11.2.3 File Purging ... 614
11.3 General Advice on System Operations ... 615
11.3.1 Rebooting and Restarting - General Notes ... 615
11.3.2 Rebooting Machines ... 615
11.3.3 Application Restart ... 615
11.3.4 Viz Mosart Application Updates ... 615
11.3.5 Windows Updates ... 616
11.3.6 Viz Mosart Main/Backup Server Testing ... 616
11.4 Redundancy ... 617
11.4.1 Automation Server Mirroring ... 619
11.4.2 Connected GUIs ... 619
11.4.3 Graphical Interfaces ... 619
11.4.4 Device-specific Redundancy ... 621
12 ActiveX ... 623
13 Appendix ... 624
13.1 Avid iNews Web Service for Status Feedback ... 625
13.1.1 Introduction ... 625
13.1.2 Installation and Configuration ... 625
13.1.3 Operation ... 635
13.1.4 Troubleshooting and Known Issues ... 647
13.1.5 Useful Links ... 650
13.2 GPIO Sample XML Config ... 651
13.3 Jupiter Systems PixelNet Video Wall Control ... 700
13.3.1 Introduction ... 700
13.3.2 Configuration ... 700
13.3.3 Implementation ... 703
13.3.4 Requests to Implement ... 703

13.4 Mosart Remote Control REST API 705
13.4.1 Introduction 705
13.4.2 Working with the REST API 705
13.5 Named Overlay Graphics 708
13.5.1 Named CGs (Named Overlay Graphics) 708
13.5.2 NCS Placeholders 709
13.5.3 Named CG Actions 710
13.5.4 Required Fields in XML 711
13.5.5 Adding the Named CG to a Template 712
13.5.6 Obtaining a valid overlay graphics as a Named Overlay Candidate 712
13.6 Stagetec Driver Configuration 713
13.6.1 AudioMixerStagetecDiamondRas.xml Config File 713
13.6.2 Scenario Configuration 713
13.6.3 Using the Scenarios 716
13.7 Structure of MOS Objects Sent to NRCS 718
13.7.1 MOS Objects with Mosart Templates Grouped by Type 719
13.7.2 MOS Object with Mosart Templates not Grouped by Type 727
13.7.3 MOS Object with Clip Templates Merged in a Mosart Template 731
13.7.4 MOS Object with Lowerthird Information 734
13.7.5 Octopus 737

This manual is organized as follows:

- Introduction
- About Viz Mosart
- Installation
- AV Automation
- Manus Administrator
- Media Administrator
- Overlay Graphics
- Show Design and Template Editor
- Operational Examples
- Viz Mosart Template Database
- Maintenance
- ActiveX
- Appendix

Copyright © 2024 Vizrt

1 Introduction

This guide takes you through the setup of Viz Mosart and is a reference guide for use during its installation and configuration. The purpose of this document is to help new users become familiar with the system; to illustrate the main workflow, and to show the available options. Familiarity with XML is required.

1.1 Related Documents

- *Viz Mosart User Guide*: Contains information on how to use Viz Mosart in live production.

For more information about all of the Vizrt products, visit:

- www.vizrt.com
- Vizrt Documentation Center
- Vizrt Training Center
- Vizrt Forum

1.2 Feedback And Suggestions

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

Copyright © 2024 Vizrt

2 About Viz Mosart

Viz Mosart is a broadcast solution designed for newsroom environments. Mosart allows a control room to be fully operational with minimal effort and consistency within the production.

Viz Mosart is a collection of server and client applications, along with services, that control equipment within your broadcast environment.

This section contains the following topics:

- Naming Convention for Components
- The Viz Mosart System
- Viz Mosart Server
- Viz Mosart Client
- Other Viz Mosart Applications

Copyright © 2024 Vizrt

2.1 Naming Convention For Components

When contacting support, please use the following standard naming convention for Viz Mosart applications and components:

Short Name	Long Name	Description
AVA	AV Automation	Application controlling attached broadcast equipment.
GUI	Viz Mosart GUI	Main user control application.
KVM	Keyboard, Video, Mouse	Application allowing multiple computers to share the same monitor, keyboard and mouse.
Manus Admin	Manus Administrator	Application controlling the Viz Mosart Rundown.
Media Admin	Media Administrator	Media Administrator. Application for monitoring media objects (clips).
MMR	Media Router	Application that shares broadcast devices between control rooms. See Media Router and Mosart Template Database.
NCS, NRCS	Newsroom Computer System	
OGI	Overlay Graphics Interface	Application that controls Overlay graphics systems.
RCC	Robotic Camera Controller	
RCS	Remote Control Service	Service used for all external PCs, like GUI and Timing Display, to connect to Viz Mosart Server. Sometimes referred to as RCPS.
Trio Interface		Application that controls Vizrt overlay graphics. (Kept for backward compatibility only - use Overlay Graphics instead).
URL	Uniform Resource Locator	In this context, address of web service.

Copyright © 2024 Vizrt

Short Name	Long Name	Description
Viz Mosart	Viz Mosart Newsroom Automation System	

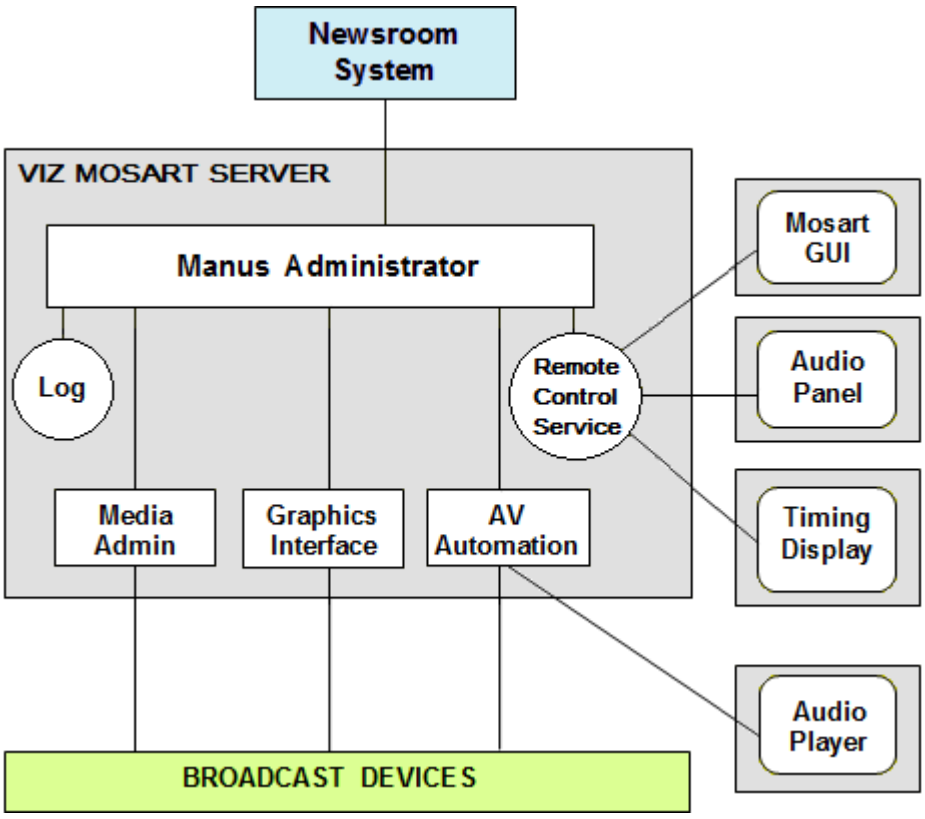
Copyright © 2024 Vizrt

2.2 The Viz Mosart System

The Viz Mosart system is a collection of software applications that join to make your production come together. The applications used may vary between installations and are dependent on your broadcast environment.

The following is a general and simplified overview of the Viz Mosart system.

Viz Mosart comprises two main applications: the **Viz Mosart Server** and the **Viz Mosart GUI**. There are also several supporting applications, such as the **Timing Display** and **Audio Panel**. You can have more than one Viz Mosart GUI running at workstations connected to the Mosart server. Normally there are two Mosart servers: one *active* (live) and one *backup* server.



Note: The shaded boxes in the diagram signify a physical device such as a workstation, server or a piece of hardware. White rectangles represent Viz Mosart applications, white circles represent Viz Mosart services, and rounded rectangles represent user displays.

The figure above shows a typical Viz Mosart installation with one **Viz Mosart Server** and four Viz Mosart workstations to run the user control software. There is also provision for a physically attached remote audio fader panel attached to a Viz Mosart workstation via the **Audio Panel** application. Redundancy is not shown.

All Viz Mosart applications and services are connected via TCP/IP, allowing them to run on any computer in a common logical network.

Copyright © 2024 Vizrt

2.2.1 Viz Mosart Applications

The Viz Mosart software package consists of the following applications:

Viz Mosart Server

- Viz Mosart Server
  - Manus Administrator
  - Media Administrator
  - Overlay Graphics
  - AV Automation
  - Log Service and Log Viewer
  - Remote Control Service

Viz Mosart Client Applications

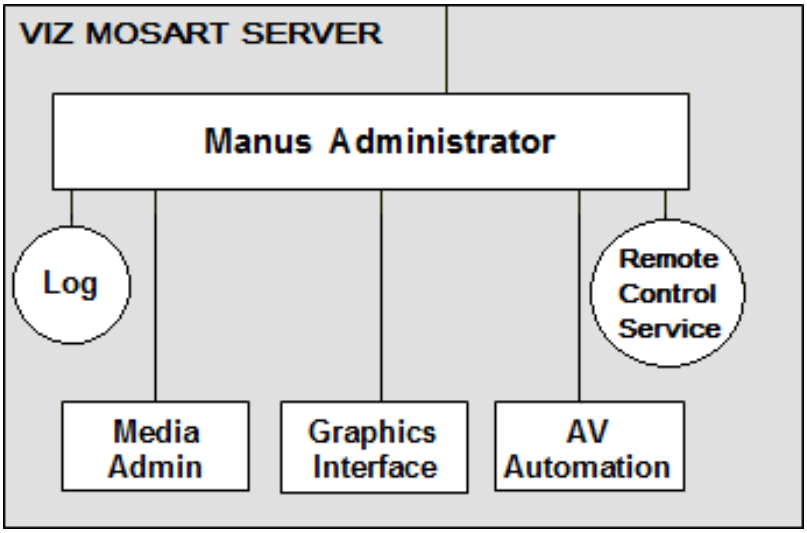
- Viz Mosart Client
  - Viz Mosart GUI
  - Audio Panel (see *Viz Mosart User Guide*)
  - Timing Display (see *Viz Mosart User Guide*)
  - Audio Player
  - ActiveX
  - iNews Timer

Other Viz Mosart Applications

- Other Viz Mosart Applications
  - Media Router and Mosart Template Database
  - Omnibus Controller
  - SNMP Service
  - Test Suite

Copyright © 2024 Vizrt

2.3 Viz Mosart Server



- Viz Mosart Server
- Viz Mosart Server Components
  - Manus Administrator
  - Media Administrator
  - Overlay Graphics Interface
  - AV Automation
  - Log Service and Log Viewer
  - Remote Control Service

2.3.1 Viz Mosart Server

The Viz Mosart server:

- Comprises a suite of dedicated server applications, relevant to your broadcast environment, running independently of one another on the same workstation.
- Supports multiple GUI workstations connected to it at any time. The control room has two Viz Mosart GUIs, one for redundancy, while others can be used for information purposes and locked to *Browse Mode* for monitoring the rundown outside the control room.
- Supports multiple timing displays. Various timing information can be seen at a glance wherever the application is installed. It may also be helpful to have a different display for the control room and another for the studio floor.
  The Timing Display is connected to the Viz Mosart rundown that is currently On Air.

Copyright © 2024 Vizrt

2.3.2 Viz Mosart Server Components

Icon	Application
	Mosart Manus Administrator Executables either: - (Default) *Vizrt.Mosart.ManusAdministrator.exe* (A combined iNEWS/ MOS program). - *MMConsoleAdmin_2007.exe* (FTP, for iNEWS). - MMConsoleAdmin_MOS.exe (for MOS workflow Newsroom Control Systems).
	Mosart Media Administrator Executable: *MMMediaAdministrator.exe*
	Mosart AV Automation Executable: *MMAVAutomation.exe*
	Mosart Overlay Graphics Executable: *MMOverLayGraphicsInterface.exe*

The Viz Mosart server is split into four main components plus two services, each handling an important role for the system to function. Splitting the system into components allows greater flexibility in handling a variety of broadcast devices and productions.

A typical Viz Mosart server installation contains the following applications and services.

- Manus Administrator (console application): Handles the connection to the Newsroom Control System (NRCS).
- Media Administrator (console application): Application for monitoring media objects (clips) referenced in the current rundown. Supports the media search option used in the Viz Mosart GUI.
- Overlay Graphics Interface (Windows application): Application for controlling and monitoring overlay graphics.
- AV Automation (Windows application): Application for controlling all other external broadcast equipment.
- Log Service and Log Viewer (Windows service): Service used for generating Viz Mosart command logs. Automatically installed with Viz Mosart Server and the Viz Mosart GUI.
- Remote Control Service (Windows service): Service used for all external PCs, like GUI and Timing Display, to connect to the Viz Mosart Server. Automatically installed with the Viz Mosart Server.

Normally, all of the services listed above should be running at all times on the Mosart server. To ensure this, you could start them from a convenience batch-script such as the one below:

Copyright © 2024 Vizrt

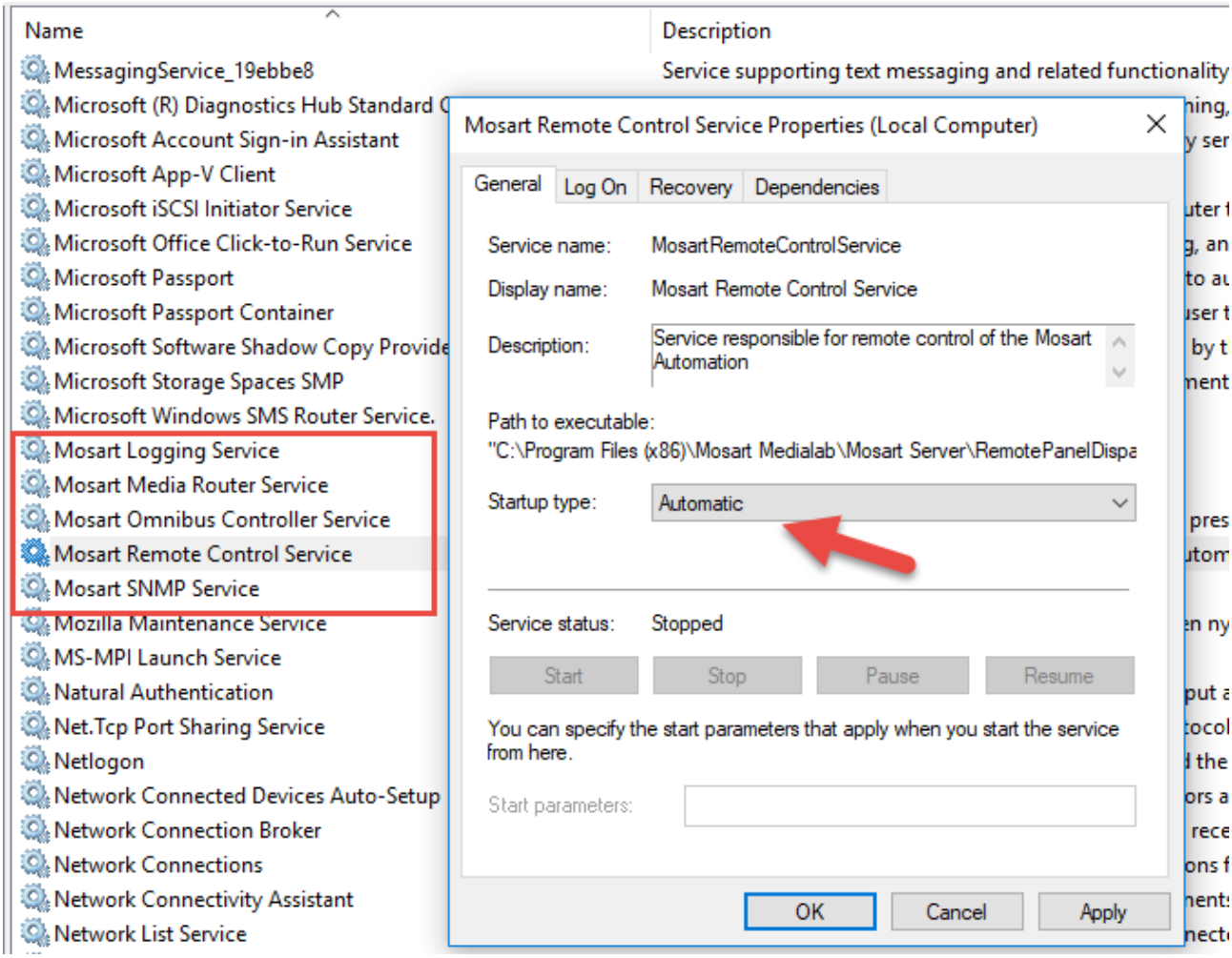
StartMosartServicesAndGUI.cmd

```
@echo off

@REM Assume we are installed at the default location, otherwise fix path
cd "C:\Program Files (x86)\Mosart Medialab\Mosart Server"

start "Manus Admin" /MIN MMConsoleAdmin_MOS.exe
timeout /T 1 > nul
start "Media Admin" /MIN MMMediaAdministrator.exe
timeout /T 1 > nul
start "AvAutomation" /MIN MMAVAutomation.exe
timeout /T 2 > nul
start "OverlayGraphics" /MIN MMOverLayGraphicsInterface.exe
timeout /T 1 > nul
cd "C:\Program Files (x86)\Mosart Medialab\Mosart GUI"
start "Mosart GUI" MosartMultiGui.exe
```

The status of the console applications (Manus Admin and Media Admin) can easily be seen on the Viz Mosart server console. The same applies Windows applications **AvAutomation** and **OverlayGraphics**.

The **RCS (Remote Control Service)** and **Log Service** run as Windows services and are therefore not available for easy visual inspection. It's recommended to make these services start automatically when Windows starts up. One easy way to do this is to use the Windows services application (**Start button > Services**): locate the required services in the **services** window, right-click the service and make sure its properties are set to **Automatic** as shown in the screenshot below:

Copyright © 2024 Vizrt

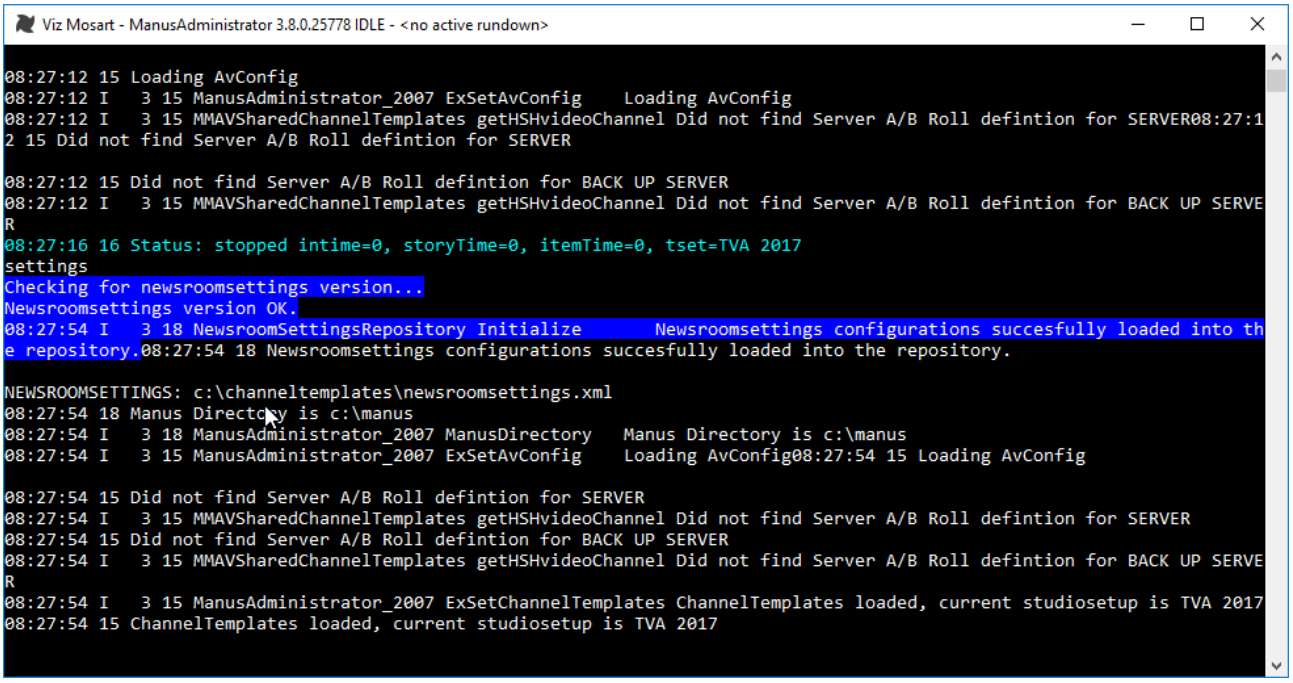


If you prefer the command line, you can easily inspect and start/stop services from Powershell. Start an elevated Powershell (Admin) session via **Start > Powershell** > right-click and select **Run as administrator**.
In the example below, the Mosart Logging Service and RCS are running, while other Viz Mosart services are stopped:

```
PS C:\Users\bva.VIZRTINT> Get-Service | grep Mosart
Running  MosartLogService   Mosart Logging Service
Stopped  MosartMediaRout... Mosart Media Router Service
Stopped  MosartOmnibusCo... Mosart Omnibus Controller Service
Running  MosartRemoteCon... Mosart Remote Control Service
Stopped  MosartSNMPService  Mosart SNMP Service
```

Copyright © 2024 Vizrt

Manus Administrator



Manus is short for *manuscript*, the script originating from the Newsroom Control System (NRCS). Manus Administrator is the central Viz Mosart application that controls the current rundown in the Viz Mosart GUI and receives rundowns from the attached NRCS. It issues events according to the different story items in the rundown and is dynamically updated when changes are made in the NRCS. Although Manus Administrator receives information from the NRCS, it does not send updates back - this prevents rundown control conflicts between the systems.

- Manus Administrator handles the connection to the NRCS, and executes commands at the appropriate time according to GUI input and template configuration.
- Manus Administrator runs as a live console application and responds to free text commands. For improved execution speed, Manus Administrator should be run in the background or be minimized.

Variants

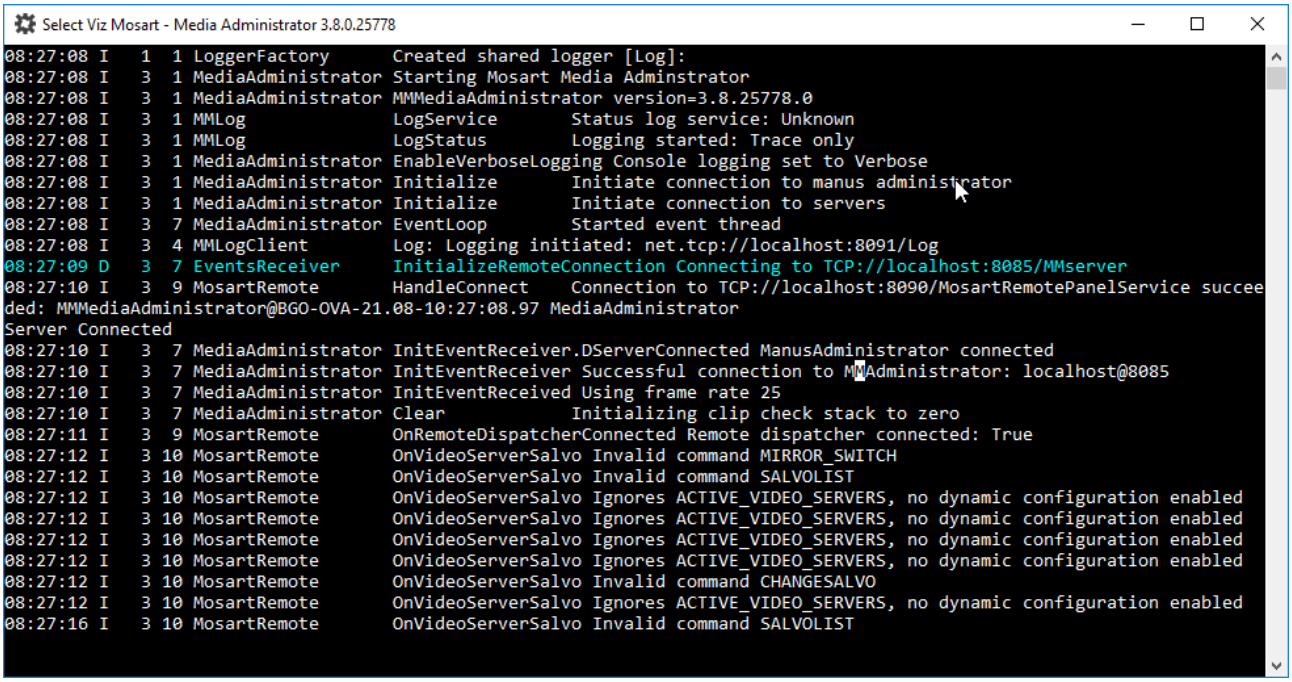
Manus Administrator has variants with slightly different settings that suit different broadcast environments. During installation, you are prompted to select the Manus Administrator appropriate to your environment.
For most users, the newer *combined Manus* is recommended.

- **Combined Manus** (default) for all NRCS connections. Installs the file *Vizrt.Mosart.ManusAdministrator*.
- **iNews FTP Newsroom** connections. Installs the file *MMConsoleAdmin_nnnn*.
- **MOS Newsroom** connections. Installs the file MMConsoleAdmin_MOS.

Copyright © 2024 Vizrt

> **Note:** Manus Administrator is sometimes also referred to as *MMConsoleAdmin*, *ManusAdmin*, *ManusAdministrator* or just *Manus*.

Media Administrator



The Media Administrator (MMMediaAdministrator) handles the database connection to your video servers. This application references the database to provide a current clip list within the Viz Mosart timeline, without making changes.

Mainly used for monitoring and searching video clips on connected video servers, the Media Administrator:

- Is responsible for reporting status for all clips in the current rundown to the Manus Administrator - typically whether a clip is present and properties like clip duration. The clip status for video servers is displayed in the Viz Mosart GUI as horizontal bars. A light-blue bar indicates a clip that is present on the video server, whilst a checkered bar indicates a non-existing clip. Clip length is shown by the length of the horizontal bar in the Viz Mosart GUI. For other media objects, Viz Mosart may present clip status in other ways.
- Makes it possible to **search** for clips on the video server. This functionality is used by the Viz Mosart **media search window**, making it possible to add clips to the rundown without the use of an NRCS. You can also search for other media objects like subtitles, graphic elements, and audio files.
  The Media Administrator runs as a live console application and responds to free-text commands. It should be run in the background or minimized for improved execution speed.

Copyright © 2024 Vizrt

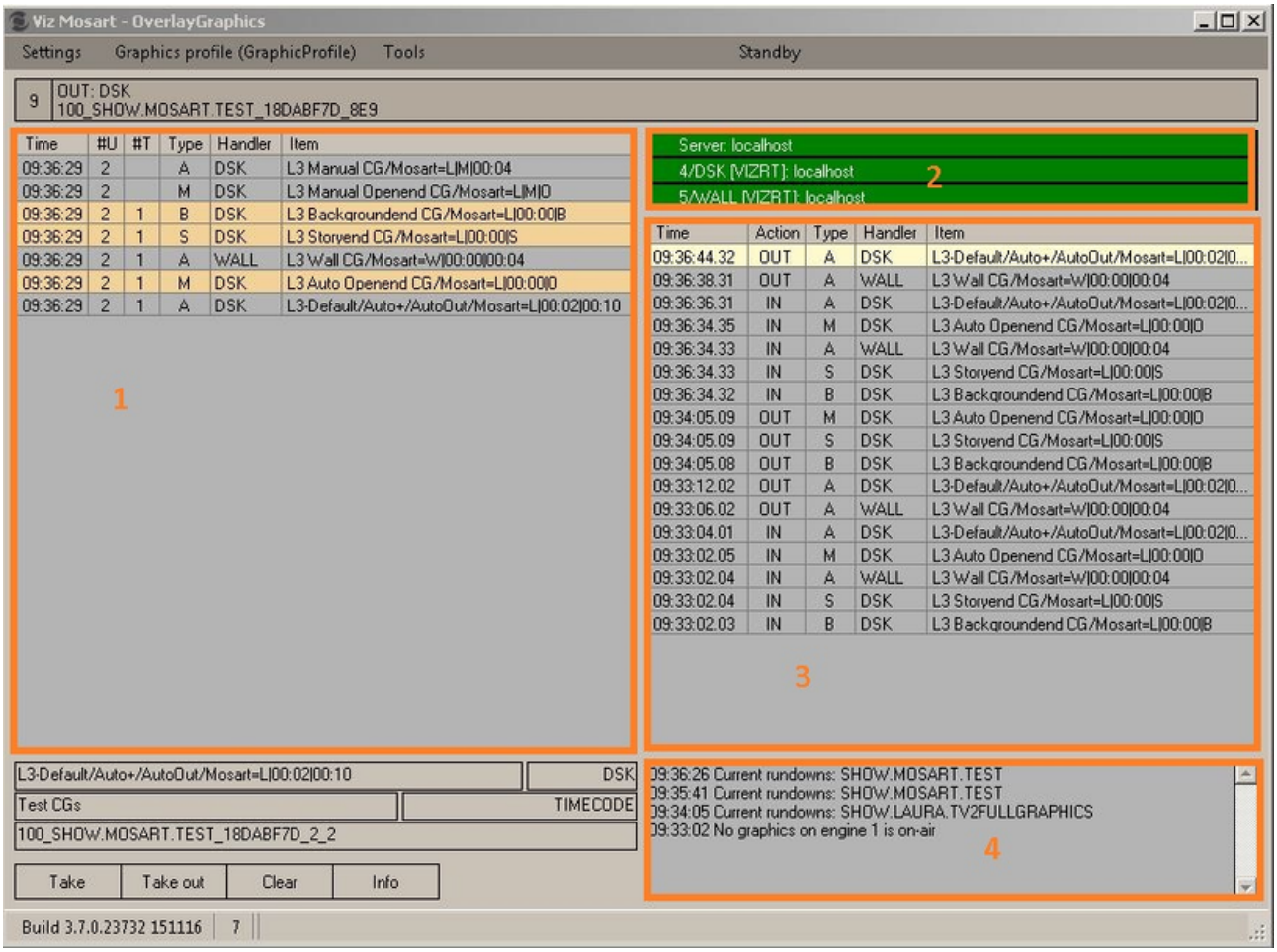
Overlay Graphics Interface

This is used for controlling and monitoring overlay graphics for graphics engines.

The two overlay graphics interfaces are:

- **OverlayGraphics** Interface (recommended).
- **Trio Interface** (deprecated, kept for backward compatibility only).

> ⚠ **Note:** An overlay graphics interface is only used for overlay graphics. Full-frame graphics are controlled through AV Automation.



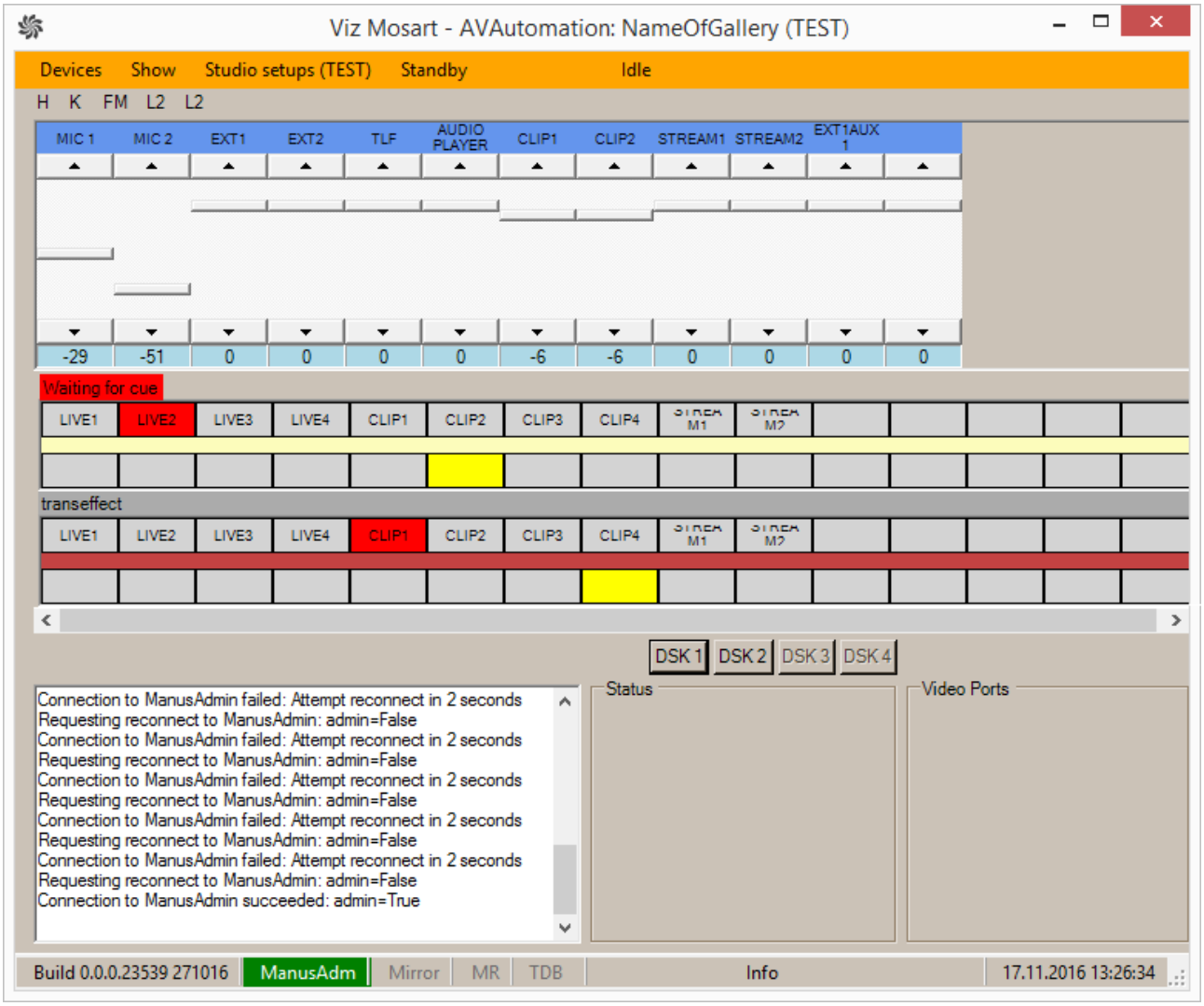
The Overlay Graphics is used for controlling and monitoring the overlay graphics for all Viz Mosart approved graphics engines. Configure it using Overlay Graphics Configuration. For a list of supported graphics devices, see Overlay Graphics Types.

Trio Interface

> ⚠ **Note: Trio Interface** is no longer recommended for use and is kept for backward compatibility only. Vizrt recommends using the Overlay Graphics instead.

Copyright © 2024 Vizrt

AV Automation



AV Automation (MMAVAutomation) controls all Viz Mosart broadcast devices with the exception of Overlay Graphics engines. Full-screen graphics are also controlled here.

Commands are issued to each device either on the fly through the Viz Mosart GUI, or as pre-determined by the rundown submitted from the NRCS.

All device commands are stored as predefined Viz Mosart templates in the **Template Editor** and are saved in `C:\channeltemplates` or in a template database.

Log Service and Log Viewer

Log Service

This is used for generating Viz Mosart command logs and is automatically installed together with the Viz Mosart Server and the Viz Mosart GUI.

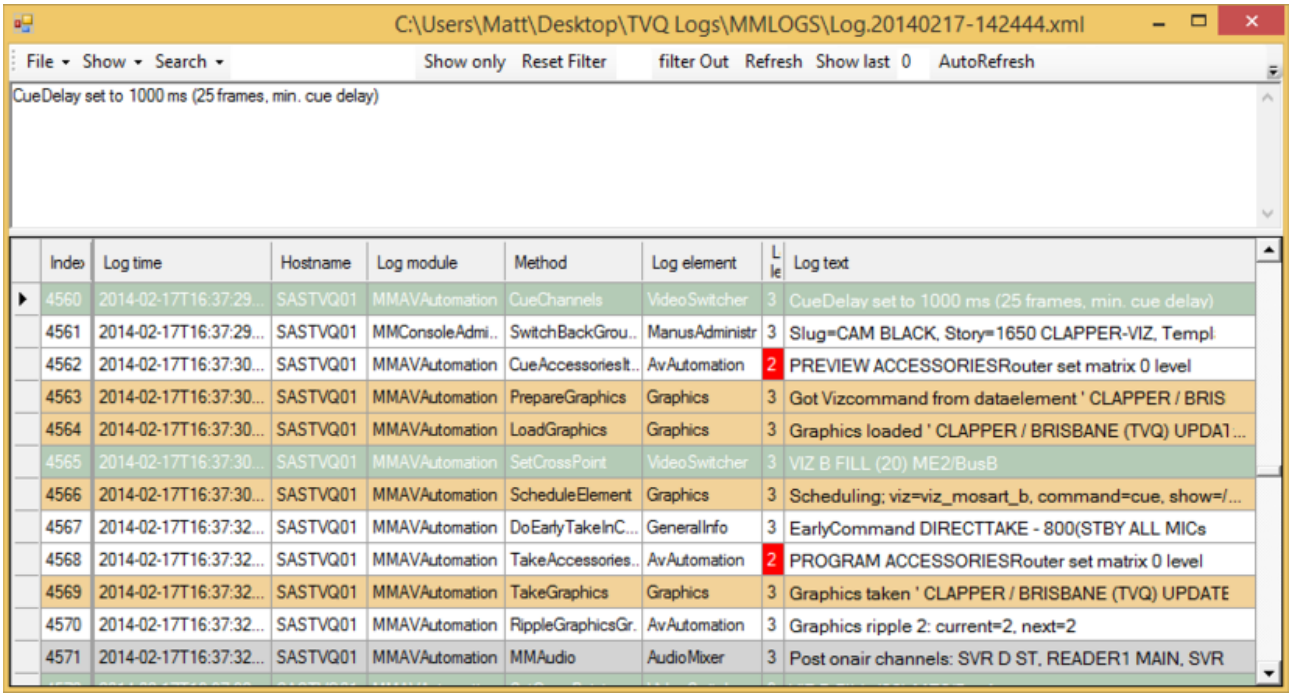
Copyright © 2024 Vizrt

Log Viewer

Logviewer

Log Viewer is used to decipher **Viz Mosart** logs. It is stored as an XML file once published, and as a .log file while actively running.

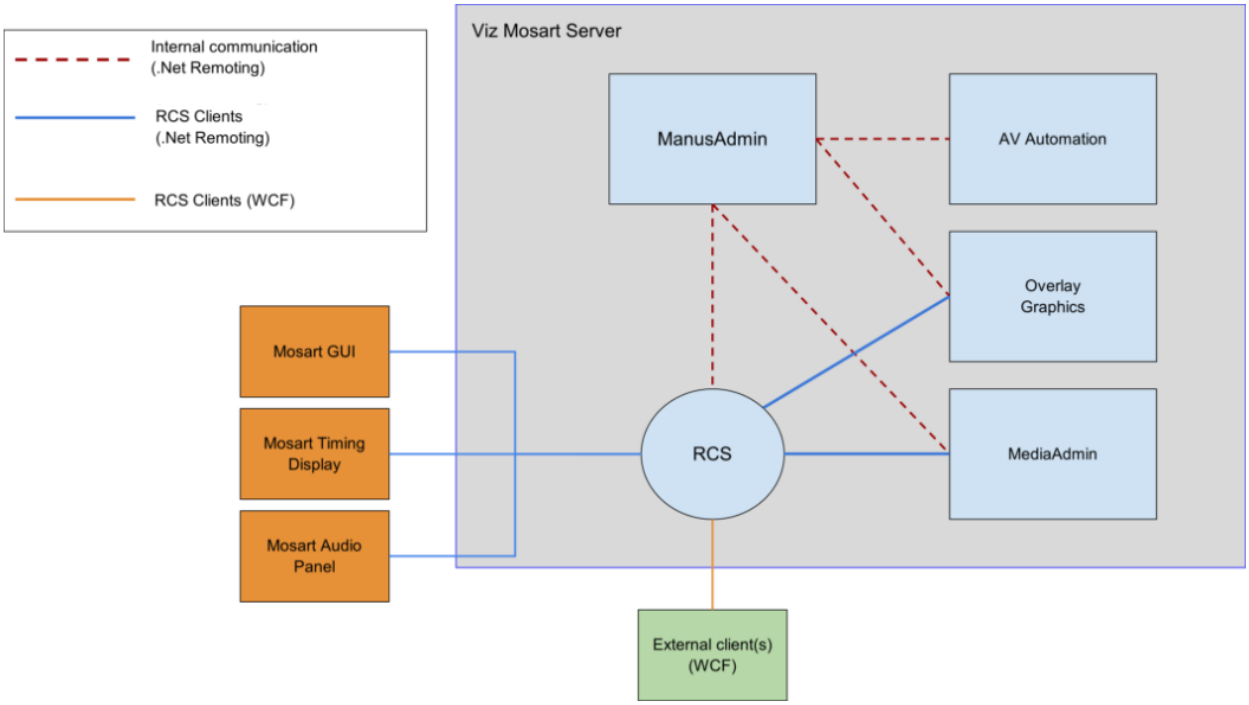
The log entries are color-coded to allow for quick command visualization in the file. Microsoft Excel may also be used to display log files (XML), and lets you use several layers of filtering and plot graphs.



Remote Control Service

The Remote Control Service (RCS, sometimes referred to as RCPS) allows all external PCs, such as the Viz Mosart GUI and Timing Display, to connect to the Viz Mosart server. RCS is automatically installed together with the Viz Mosart server.

Copyright © 2024 Vizrt



RCS provides a WCF interface for remote clients. This interface allows external applications or services to connect to the Viz Mosart server.

For configuration options, see the comments in the associated standard configuration file *RemoteDispatcherServiceConfig.xml* located by default at C:\Program Files (x86)\Mosart Medialab\Mosart Server\ConfigurationFiles\RemoteDispatcherServiceConfig.xml

See Also

- Manus Administrator
- AV Automation
- AV Automation Device Properties
- Media Administrator

Copyright © 2024 Vizrt

2.4 Viz Mosart Client

The main components that make up Viz Mosart Client are:

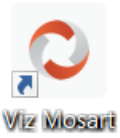
- Viz Mosart GUI: A Windows application for control and monitoring of the Viz Mosart rundown.
- NRCS Plugin: A web application drag and drop tool for reliably adding Viz Mosart template instructions to an NRCS story.
- Timing Display: A web application that displays selected timing details from the active Viz Mosart rundown.
- Rundown Viewer: A web application displaying scalable details from the active Viz Mosart rundown.
- Audio Panel: A Windows application used to control and monitor software audio faders.
- Audio Player: Plays out audio files that are located on Viz Mosart's file system.

Additional Viz Mosart Client applications:

- iNews Timer: An iNEWS-specific utility.

Note:
- Multiple instances of the above applications may run on any workstation connected to the same network as the Viz Mosart Server.
- All instances of Viz Mosart software must run the same Viz Mosart version with a standard OS across all Viz Mosart client machines.

2.4.1 Viz Mosart GUI



The Viz Mosart GUI workstation is a standalone networked PC with a single instance of the Viz Mosart GUI application (MosartMultiGui) run in full-screen mode. More than one PC can run the same software component, for example, you can add a backup GUI PC.

The Viz Mosart GUI is the main control interface for the Viz Mosart Server. The director can run the production from a single key press on the workstation keyboard.

Copyright © 2024 Vizrt

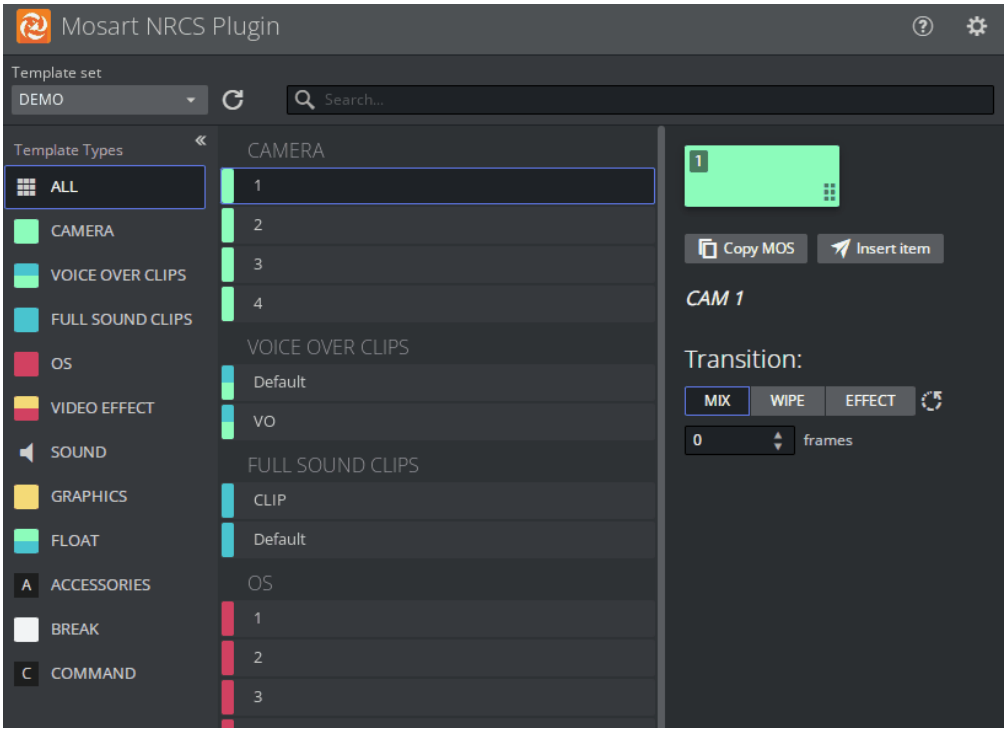
Alternate configurations may also host a fader panel and a Viz Mosart Client. Each Viz Mosart Client application should run on a standalone networked PC.

The Viz Mosart GUI is described in detail in the *Viz Mosart User Guide*.

2.4.2 NRCS Plugin

The NRCS Plugin delivers the functionality previously provided by the legacy Viz Mosart ActiveX Plugin.

Copyright © 2024 Vizrt



Features include:

- Searching for a Template
- Modifying a Template
- Adding Template Details to the NRCS
- Template Change Alert
- Optimizing Your NRCS Plugin Workspace
- Copying the MOS Object Details
- Modifying Existing MOS Objects.

For details, please refer to the Mosart Web Applications Guide.

2.4.3 Timing Display

The web-based Timing Display replaces the legacy standalone utility Timing Display (WPFTimingInfo).

Copyright © 2024 Vizrt

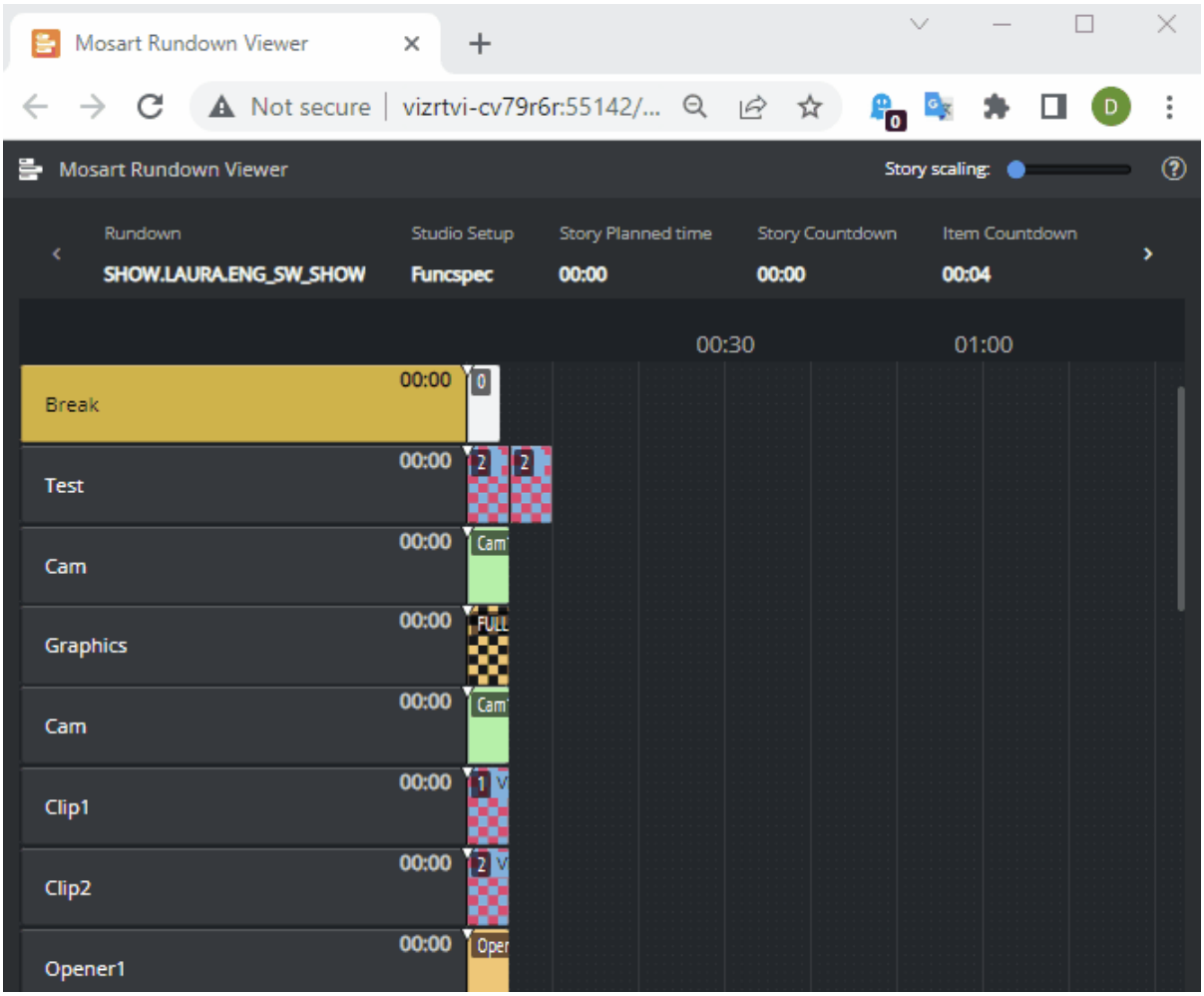
- The Timing Display provides timing information to the studio control room, studio floor or any device connected to the Viz Mosart network.
- Details are synchronized to the current rundown in the Viz Mosart GUI.
- Multiple Timing Displays can be customized to the specific needs of the production staff.

For details, please refer to the Mosart Web Applications Guide.

2.4.4 Rundown Viewer

The Mosart Rundown Viewer provides a convenient web browser view of details from the current Viz Mosart rundown.

Copyright © 2024 Vizrt

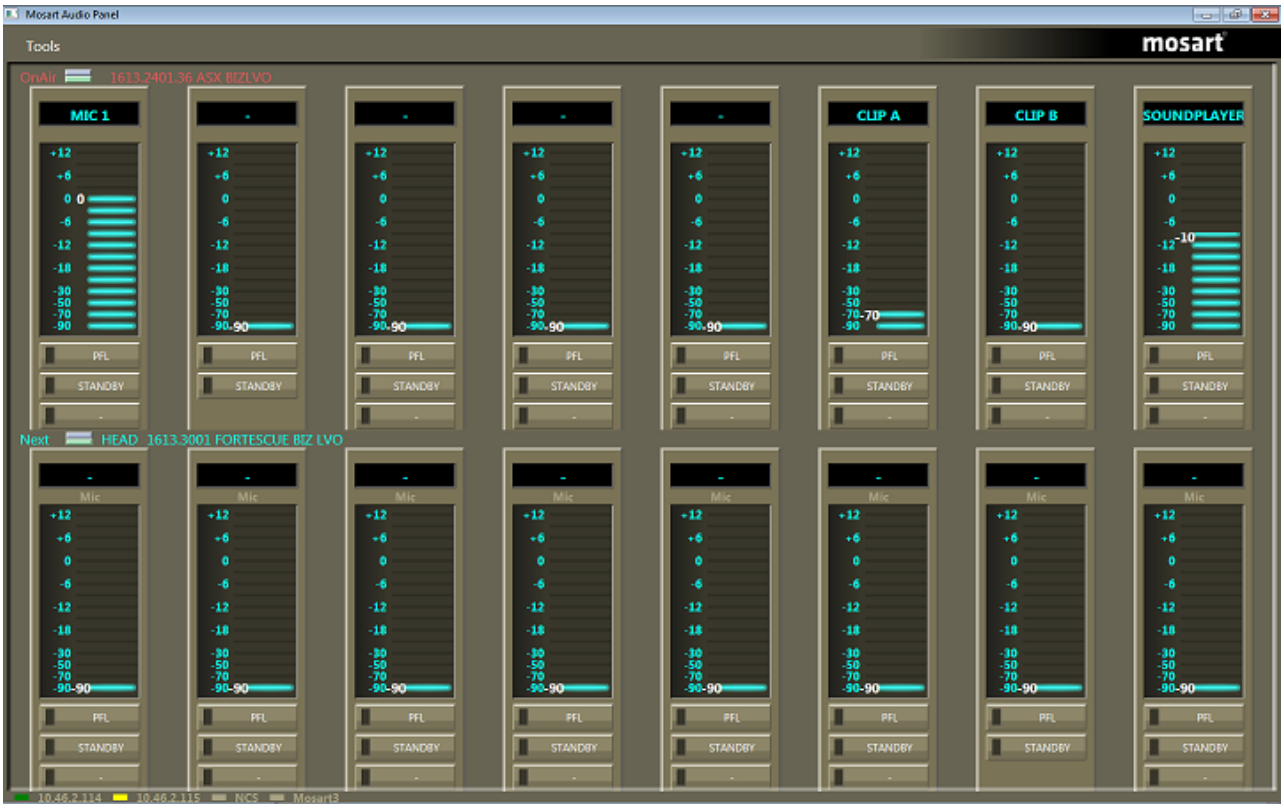


- Displays the active Viz Mosart rundown.
- Summary of story timings.
- Supports a variety of devices. For example an iPad or networked phone.

For details, please refer to the Mosart Web Applications Guide.

Copyright © 2024 Vizrt

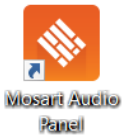
2.4.5 Audio Panel



Audio Panel (Server)

The Audio Panel (AudioPanel) lets you use a Behringer BCF2000 or JL Cooper MXL with a Viz Mosart Server. The panel controls data transmission between the physical hardware panel and AV Automation on the Viz Mosart Server.

Audio Panel (Client)

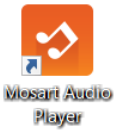


The Audio Panel (Client) is a software representation of the audio mixer connected to the Viz Mosart Server, displaying current faders, on air faders, and faders in preview.

The Audio Panel is described in detail in the *Viz Mosart User Guide*.

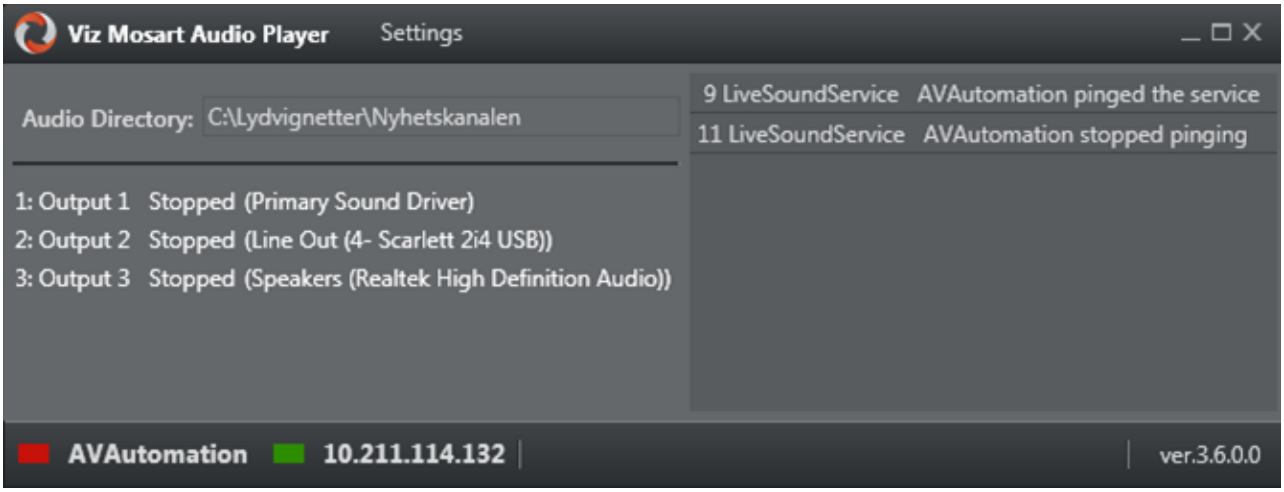
Copyright © 2024 Vizrt

2.4.6 Audio Player



The Audio Player is part of the Viz Mosart installation and may be used to play out audio files located on the file system. This is useful for playing out files that are used on a regular basis, such as openers and audio-beds.

> **Note:** A broadcast sound card is required for audio output. If you're running the Audio Player on a Windows Server, you must install Windows Audio features on the machine in order for the various audio formats to work.



> **Note:** The Audio Player replaces the discontinued Soundfile Player.

The Audio Player is described in detail in the Viz Mosart User Guide.

2.4.7 iNews Timer

The iNews Timer provides accurate timing of the production in iNews, and is used to synchronize the story currently on air in the Viz Mosart GUI to the corresponding entry in iNews.

Copyright © 2024 Vizrt

2.5 Other Viz Mosart Applications

This section contains the following topics:

- Media Router and Mosart Template Database
- Omnibus Controller
- SNMP Service
- Test Suite

2.5.1 Media Router and Mosart Template Database

Media Router and Template Database with a MySQL database enables two features within Viz Mosart:

- Resource sharing: video servers, graphics devices, robotics, and so on.
  Viz Mosart can be configured to use shared broadcast devices between control rooms via the dedicated Media Router (MMR). This device enables Viz Mosart to select other devices before the next show.
  The vision switcher and audio mixer are not included in Media Router control, but may in some situations be shared, depending on the device's capability and/or the planned use of the device.
- Template sharing: Viz Mosart template information between multiple Viz Mosart installations.

These features are a licensed addition to the Viz Mosart Server. Please consult your nearest sales agent for more information.

2.5.2 Omnibus Controller

The Omnibus Controller allows communication with Omnibus systems for Viz Mosart installations running Omnibus equipment.

2.5.3 SNMP Service

This application allows for system monitoring of the Viz Mosart Server with SNMP agents such as Nagios and Navigator. Any SNMP agent application with MIB support is able to use this service.

2.5.4 Test Suite

The Test Suite contains various diagnostic utilities that can be used when installing Viz Mosart for the first time or when connected broadcast devices must be checked.

This is a list of some of the tools available in the Test Suite:

- **AsRunLogTester:** Test AsRun Log publishing
- **MediaRouterTester:** Test Media Router
- **MVCPSimul:** Test MVCP protocol

Copyright © 2024 Vizrt

- **NexioTest:** Test Nexio protocol
- **QuantelTest:** Test Corba protocol
- **RoboTest:** Test camera robotics
- **ShotokuTester:** Test Shotoku camera robotics
- **TestAudioMixer:** Test Audio Mixer communications
- **TestMIDIShowControl:** Test MIDI Lighting control
- **TestRouterControl:** Test Router communications
- **TestVDCP:** Test VDCP protocol
- **TestVisionMixer:** Test Vision Mixer communications
- **VintenTester:** Test Vinten 200 protocol

Copyright © 2024 Vizrt

3 Installation

This section contains an overview of how to install the various Viz Mosart Applications, by running standard Microsoft installers (.MSI files).

- Prerequisites
- Viz Mosart Installation Files
- Licensing

Copyright © 2024 Vizrt

3.1 Prerequisites

There are several system prerequisites that must be in place before the Viz Mosart Applications are installed.

This section contains:

- System Requirements
  - Microsoft .NET Framework
  - Microsoft Visual C++ Redistributable Package
- Hardware Requirements
  - Viz Mosart Automation Servers
  - Viz Mosart GUI PCs
  - Viz Mosart Audio Player
  - Viz Mosart Timing Display and Audio Panel PCs
  - Viz Mosart Shared Template Database
  - Pre-Installation Tasks
  - Connections
- Licensed and Optional Features

3.1.1 System Requirements

The following apply all machines running Viz Mosart software:

Prerequisite	Target	Description
Microsoft Windows Server 2012 R2 or later	Viz Mosart Server	Operating system. Viz Mosart has been verified working with Windows Server versions up to and including Windows Server 2022. Viz Mosart Server will also work with Windows 10 and Windows 11.
Microsoft Windows 10 or later	Viz Mosart Client	Operating system.
Microsoft .NET Framework	All	See the *Viz Mosart Release Notes* for the applicable Microsoft .Net version. Most of the Viz Mosart installers cancel the installation process if the correct .NET version is not detected.
Microsoft Visual C++ Redistributable (both x86 and x64)	Viz Mosart Server	See the *Viz Mosart Release Notes* for the applicable Microsoft Visual C++ Redistributable Package.

Copyright © 2024 Vizrt

Prerequisite	Target	Description
Microsoft Edge WebView2 Runtime (x64)	All	Required for the Mosart Central Settings which can be opened from the combined Manus Administrator (introduced in Viz Mosart 5.2.0) or AV Automation driver properties (since Viz Mosart 5.3.0) on the Viz Mosart Server. From Viz Mosart 5.4.0 also required for the Viz Mosart UI for doing Vizrt Graphics editing and for the web-based Timeline window and Story Script window introduced in that version. ⚠ For Windows Server 2012 and 2012R2, the last supported version is **Microsoft Edge-WebView2 109.** Use the Microsoft online installer to manually install this version if the Windows servers mentioned above are used.
Wibu-Systems CodeMeter Runtime Kit	Viz Mosart Server	Required for licensing
Windows Folder Structure	All	The following folder must be available, containing Viz Mosart installers, before installation: C:\mosart\installers
Windows Firewall	All	For some versions of Windows, it may be necessary to open ports. Either completely disable Windows firewall, or alternatively open for Ping and ports 8080-8099. If Mosart Template Database is used, port 3306 also must be opened.
Windows Audio	Viz Mosart Server	If MIDI devices are connected via direct conversion serial adapters, Windows Audio feature must be enabled on the server. ⚠ **Note:** If using Windows Remote Desktop, audio must be set to play on the server, not the remote PC, as the Audio configuration will otherwise be disturbed.
Windows Server Type	Viz Mosart Server	It may be favorable to configure the Windows Server running the Viz Mosart Server software to **Application Server**, as this automatically pre-configures several features and also automatically sets a detailed .NET configuration.
Server Specifications	All	For more details, see below.

⚠ **Note:** All prerequisites must be completed before installing any Viz Mosart components, both server and client applications.

Copyright © 2024 Vizrt

Microsoft .NET Framework

Microsoft .NET must be installed on the system before installing any Viz Mosart Applications. See the *Viz Mosart Release Notes* for the applicable Microsoft .NET version.

To determine which Microsoft .NET Frameworks are installed

- If you do not know how to determine this, contact your IT System Administrator to learn how to.

Installing the Microsoft .NET Framework

1. Obtain the applicable Microsoft .NET Framework Installer from the Microsoft Download Center.
2. Run the installer and follow the prompts.
3. Restart the machine if prompted on completion.

Microsoft Visual C++ Redistributable Package

A *Microsoft Visual C++ Redistributable Package* must be installed on the system before installing the Viz Mosart Server software. See the *Viz Mosart Release Notes* for details on the Microsoft Visual C++ Redistributable Package.

> **Note:** Viz Mosart is a 32-bit program, but it also includes x64 components.

Obtain the applicable Visual C++ Redistributable Package (both **x86** and **x64**) installers from the Microsoft Download Center.

Installing Microsoft Visual C++ Redistributable Package:

1. When downloading, you have to download both the **32-bit (x86)** version and the **64-bit (x64)** version.
2. Run the installers and follow the prompts.
3. Restart the machine if prompted on completion.

3.1.2 Hardware Requirements

Viz Mosart Automation Servers

Two servers: one main and one backup.

Minimum Recommended Hardware Configuration

- Dual Power Supply for redundancy
- Redundant disk configuration
- Integrated Lights Out (iLO), IPMI or similar with Keyboard-Video-Mouse (KVM) support
- Windows operating system with applicable .NET Framework
- 2 x Gigabit Ethernet ports

Copyright © 2024 Vizrt

- 1280 x 1024 display adapter
- 16 GB Memory

Recommended Performance-related Components

- 2 Processors with performance as 2.6 GHz 6-core and with 12 MB L2 cache
- Minimum 50 GB free storage space
- Windows Server 2008, 2012, 2016, 2019 or 2022 operating systems

Site-specific Requirements

Location-specified external device connections might include the following examples:

- PCI multiport serial card RS-422/RS-232 (short distance connections).
  www.moxa.com CP-118EL-A.htm
- IP connected Terminal Servers with serial output ports RS-422/RS-232 (long distance connections).
  www.moxa.com - 2650
  www.moxa.com - 5150
- IP connected Terminal server with MIDI output port.
  www.kissbox.nl - CM-MIDI2
- Additional Ethernet ports
- IP connected GPIO control box
  https://www.wut.de/e-57730-ww-daus-000.php

Note: W&T is the only GPIO control device vendor supported by Mosart. Up to four boxes can be controlled, giving 48 outputs. Only the first box is used for input, giving you 4 predetermined Mosart rundown functions, and 8 general inputs. Display, keyboard and mouse should be available in the relevant gallery.

Viz Mosart GUI PCs

Two PCs: One main and one backup.

Minimum Recommended Hardware Configuration

- Local PCs physically located in the control room. Alternatively Integrated Lights Out (iLO), IPMI or similar, optionally with Keyboard-Video-Mouse (KVM) support.
- Windows 10 or Windows 11 64 bit operating system with applicable .NET Framework
- Gigabit Ethernet port
- Approved video adapter with resolution 1920 x 1080
- 32 GB Memory
- Displays with minimum 1920 x 1080 resolution, keyboard and mouse should be installed in the gallery.

Recommended Performance-related Components

- Processor with performance equal or better than Intel i7 2.6 GHz 6-core, 12 MB L2 cache

Copyright © 2024 Vizrt

- Minimum 50 GB free storage space

Viz Mosart Audio Player

If you will use the Viz Mosart Audio player, a single dedicated PC is normally placed in the server room.

The degree of hardware redundancy required for this function shall be considered.

The PC running the Viz Mosart Audio Player should be equipped with:

- A broadcast quality audio card registering with Windows driver.
  For example, Lynx Studio Technology AES16e PCI Express AES/EBU Interfaces, www.lynxstudio.com.
- Windows 10 or Windows 11 operating system with applicable .NET Framework.
- Ensure the audio card is supported by the selected Windows operating system.

Remaining hardware configuration and performance parameters are similar to GUI PCs

> **Note**: The Viz Mosart Audio Player replaces the discontinued *Mosart Sound File Player*.

Viz Mosart Timing Display and Audio Panel PCs

The main purpose of these two PCs is to display the output from the respective Viz Mosart applications.

- For the Audio Panel it is also possible to interact with the application via mouse and keyboard.

The configuration and performance parameters shall be similar to the GUI PCs.

Viz Mosart Shared Template Database

The Template Database is built on an Oracle MySQL Server. Actual database traffic will be light – any modern PC will suffice, since Viz Mosart template features are more similar to a shared storage function than a classic database transaction server.

Recommended MySQL Software

- Refer to the *Viz Mosart Release Notes*.

Minimum Recommended Hardware Configuration

- Keyboard-Video-Mouse (KVM) support.
- Windows operating system with applicable .NET Framework.
- Gigabit Ethernet port.
- 1280 x 1024 display adapter.

The above hardware can be increased, for example with dual redundant PSUs, NICs, disks etc. for redundancy.

Copyright © 2024 Vizrt

Note: Some of the hardware components (for example dual PSU) might necessitate an upgrade to a server proper, running a recent version of Windows Server. It is also possible to add another machine as a backup server.

Recommended Performance-related Components

- Processor with performance equal or better than Intel 2.0 GHz Dual-core.
- 8 GB Memory.
- Minimum 50 GB free storage space.

Pre-Installation Tasks

A computer platform as described must be installed by the customer prior to the start of installation.

This includes:

- Dual independent power feeds to Viz Mosart Servers and the GUI PCs, where one of the inputs is from an Uninterruptible Power Supply (UPS) source.
- Set-up of keyboard, video and mouse for Viz Mosart GUIs and Viz Mosart Servers in the relevant gallery.
- Cabling and connectivity-testing from the Viz Mosart servers to the gallery equipment to be automated.
- Tested networks connection from the Viz Mosart servers to the Newsroom system and the Viz Mosart GUIs.
- For all IP connections (see Connections below), the customer must make sure relevant ports and services are opened in any in-between firewalls.
- Viz Mosart Support should be permitted remote access to all Viz Mosart computers for online support.

Connections

All Viz Mosart applications and services are connected via TCP/IP. This allows them to run on any computer in a common logical network.

- Connections to external devices are a combination of TCP/IP and serial communication protocols like RS-232, RS-422 and MIDI.
- Extenders and translators, terminal servers and MIDI over IP, may also be used.

Note: All device connections (such as link layer and application protocols) are dependent on the device itself.

3.1.3 Licensed and Optional Features

For general Licensing guidelines, see section Licensing.

Copyright © 2024 Vizrt

- **Engine Switcher**: If Viz Mosart is licensed to use this optional feature, please refer to the Viz Engine requirements in the *Viz Engine Administrator Guide* (version 5+).

Copyright © 2024 Vizrt

3.2 Viz Mosart Installation Files

Viz Mosart is delivered as a software package comprising various server and client applications, used for connecting to studio devices and managing the NRCS rundown, imported into Viz Mosart.

- Viz Mosart Installer Files
- Additional Mosart Files
- Installation and Upgrade
- Version Numbering

3.2.1 Viz Mosart Installer Files

Obtain the latest version of Viz Mosart at Vizrt's customer FTP:

1. Authenticate to Vizrt's FTP at https://download.vizrt.com/
2. Navigate to products/VizMosart/Latest Version/

Some of the installations files are available both as .msi and .exe.

Installer	Machine Role	Description
VizMosartActiveX	NRCS Client	Legacy installer containing the **ActiveX** utility, for displaying Viz Mosart template data in the NRCS client. Is superseded with the web application, **NRCS Plugin**. See WebApplications.
VizMosartAudioPanel	Audio Client (GUI or standalone PC)	Installer containing the **Audio Panel**, an application for controlling audio faders, both on air and in preview.
VizMosartAudioPlayer	Viz Mosart Server or Audio Client	Installer containing the **Audio Player**, typically installed on a computer, or Viz Mosart Server, attached to the newsroom audio mixer. ⚠ **Note:** Audio Player is a replacement for the discontinued Soundfile Player.
VizMosartGUI	Viz Mosart GUI Client	Installer containing the **Viz Mosart UI** application, which is the Viz Mosart user interface.
VizMosartINEWSTimer	iNews Client	Installer containing the **iNews Timer**, an application used to create accurate timing of the production in iNews.

Copyright © 2024 Vizrt

Installer	Machine Role	Description
VizMosartLogViewer	Viz Mosart Server	Installer containing the **Log Viewer**, an application used to simplify reading of Viz Mosart logs.
VizMosartMediaRouter	Viz Mosart Database Server	Installer containing the **Media Router**. For more details, see *Media Router* and *Mosart Template Database Administrator Guide*.
VizMosartOmnibusController	Viz Mosart Server	Installer containing the **Omnibus Controller**, an application used for Omnibus Server control.
VizMosartServer	Viz Mosart Server	Installer containing the various Viz Mosart Server components: Manus Administrator, Media Administrator, AV Automation, and Overlay Graphics Interface.
VizMosartSNMPService	Viz Mosart Server	Installer containing the **SNMP Service**, used for monitoring the Viz Mosart Server suite via SNMP.
VizMosartTestSuite	Viz Mosart Server	Installer containing the **Test Suite**, test utilities used when installing Viz Mosart.
VizMosartTimingDisplay	Timing Client	Installer containing the **Timing Display**, used to provide timing information to the studio control room and studio floor. Is superseded with the *web application*, **Timing Display**. See WebApplications.

3.2.2 Additional Mosart Files

Obtain the latest version of additional Viz Mosart software at Vizrt's customer FTP:

1. Authenticate to Vizrt's FTP at https://download.vizrt.com/
2. Navigate to products/VizMosart/Latest Version/

Installer Location	Machine Role	Description
WebApplications	Viz Mosart Server	Installer containing a Configuration tool and all Mosart Web Applications. Mosart Web Applications are described here.

Copyright © 2024 Vizrt

Installer Location	Machine Role	Description
WebApplications\MosartDashboard	Viz Mosart Server + Clients	Directory containing both backend service *MosartDashboardService-n.n.n.n.msi* and frontend service *MosartDashboardUIService-n.n.n.n.msi* Details here.
Engine Switcher Scenes	Viz Engine Server	Directory containing sample scenes for Engine Switcher. Located at / products/VizMosart/Engine Switcher Scenes. See also section Viz Engine as a Switcher.

3.2.3 Installation and Upgrade

Installing or upgrading Viz Mosart preserves any stored settings and configurations. Run the Viz Mosart Installer Files manually from Windows Explorer, as presented in To Install Viz Mosart.

Backing-up Viz Mosart

The locations are:

All files in the following folders:

- C:\channeltemplates
- %localappdata%\Mosart_Medialab
- %programdata%\Mosart Medialab\ConfigurationFiles
- C:\Program Files (x86)\Mosart Medialab\<Mosart application>\ConfigurationFiles
- All files with extension .exe.config in folders
  C:\Program Files (x86)\Mosart Medialab\<Mosart application>\
  where <Mosart application> is the relevant Viz Mosart application, for example, Mosart Server, Mosart GUI.

Windows registry settings for

- HKEY_CURRENT_USER\Software\[Wow6432Node\]Mosart Medialab
- HKEY_LOCAL_MACHINE\Software\[Wow6432Node\]Mosart Medialab

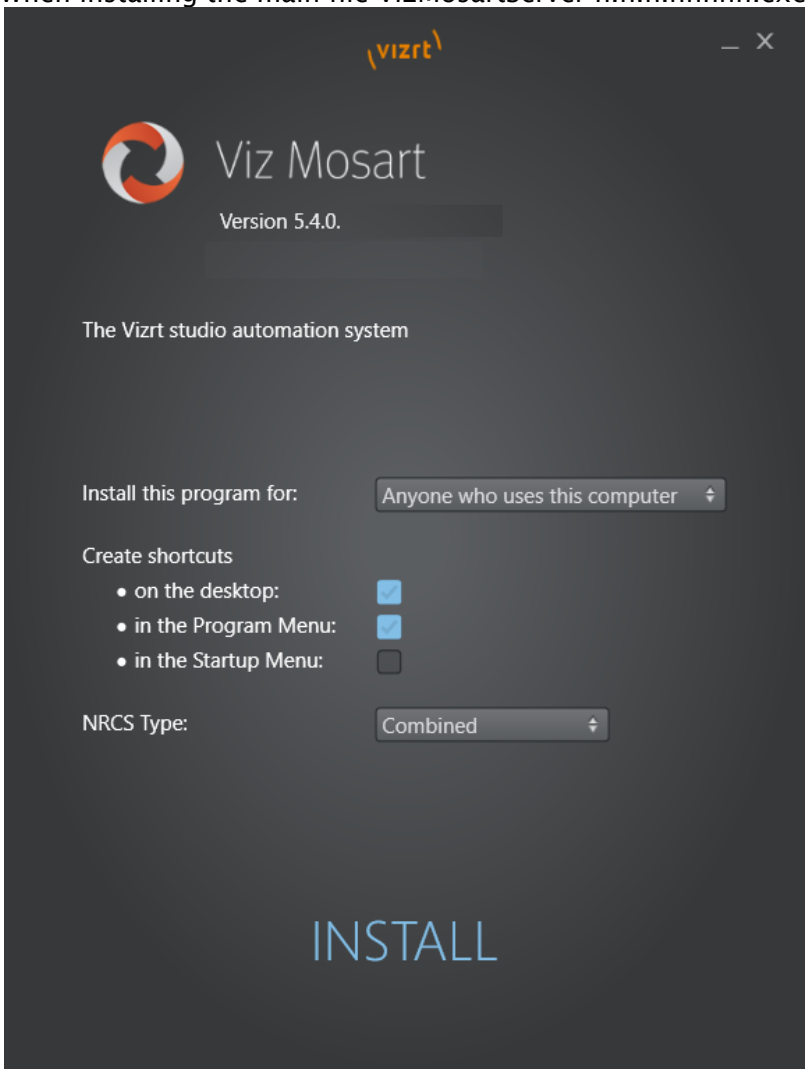
Installation

To Install Viz Mosart

1. Download all relevant Viz Mosart Installation Files to your preferred location.
2. If you are upgrading from an earlier Viz Mosart version it is recommended to take a backup of your current installation settings before you start the installation of the new version. See Backing-up Viz Mosart above.

Copyright © 2024 Vizrt

3. Double-click the installation files (msi or exe).
4. Follow the prompts to complete installation.
5. When installing the main file VizMosartServer-n.n.n.nnnnn.exe, you can specify NRCS type.



⚠ **Note**: All three (MOS, iNEWS and Combined) Viz Mosart Manus Administrator entities are always installed. The selection made with the **NRCS Type** drop-down decides which of these three, the Manus Administrator shortcut shall point to.

3.2.4 Version Numbering

Installation folders are named with the Viz Mosart version number. All files inside that folder also have the corresponding version number attached to the filename.

Version numbers have the following syntax:

- Major.Minor.Revision.Build (for example **5.4.5.12345**)

  Where:

Copyright © 2024 Vizrt

- Major.Minor.Revision denotes the Viz Mosart Version Number
- Build denotes a patch number

Example: `Installers\5.4.5.12345\VizMosartServer-5.4.5.12345.msi`

In this example, the computer has a folder for the Viz Mosart 5.4.5.12345 release, containing associated installers.
Every significant code change of Viz Mosart results in a change to the Viz Mosart version number. This increases either the major, minor, or revision number.

Patches are identified by having a higher build number than the remainder of the files.

Copyright © 2024 Vizrt

3.3 Licensing

In version 5.2.0 a license management backend, a License Configuration menu and associated Notifications were introduced.

In later versions, license-based feature control will be introduced. To ensure service continuity with the correct licenses in place before upgrading, please contact your Vizrt account manager.

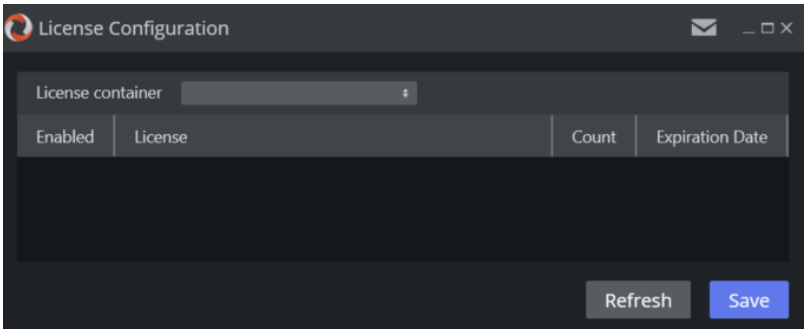
3.3.1 License Management

Vis Mosart uses the same Wibu-based backend licensing architecture as other Vizrt products.

- For further details, please refer to the latest version of the Vizrt Licensing Administrator Guide.
- An orientation course to the Management portal is also available on Viz University.

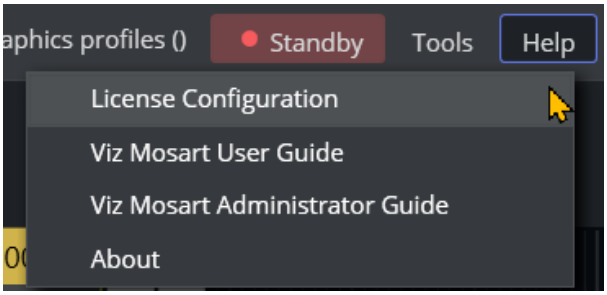
3.3.2 License Configuration status

The configuration status of Viz Mosart licenses can be seen in Viz Mosart UI, using the new License Configuration UI.



In version 5.2.0 this License Configuration UI is usually empty, and is present only for testing, and for administrators to become acquainted with the concept. In later versions it will display the active features that require a license, and the status of the respective licenses.

- To open the License Configuration UI from the Viz Mosart UI menu bar, navigate to **Help > License Configuration**.

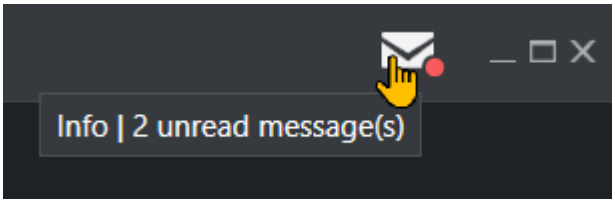


Copyright © 2024 Vizrt

3.3.3 License Notifications

Important licensing notifications are presented to the user through a the Viz Mosart UI.

A notifications icon appears at the top of all menus. A red marker indicates an unread notification.



- Clicking the icon opens the Notification Box menu, listing important system messages that will usually require action.

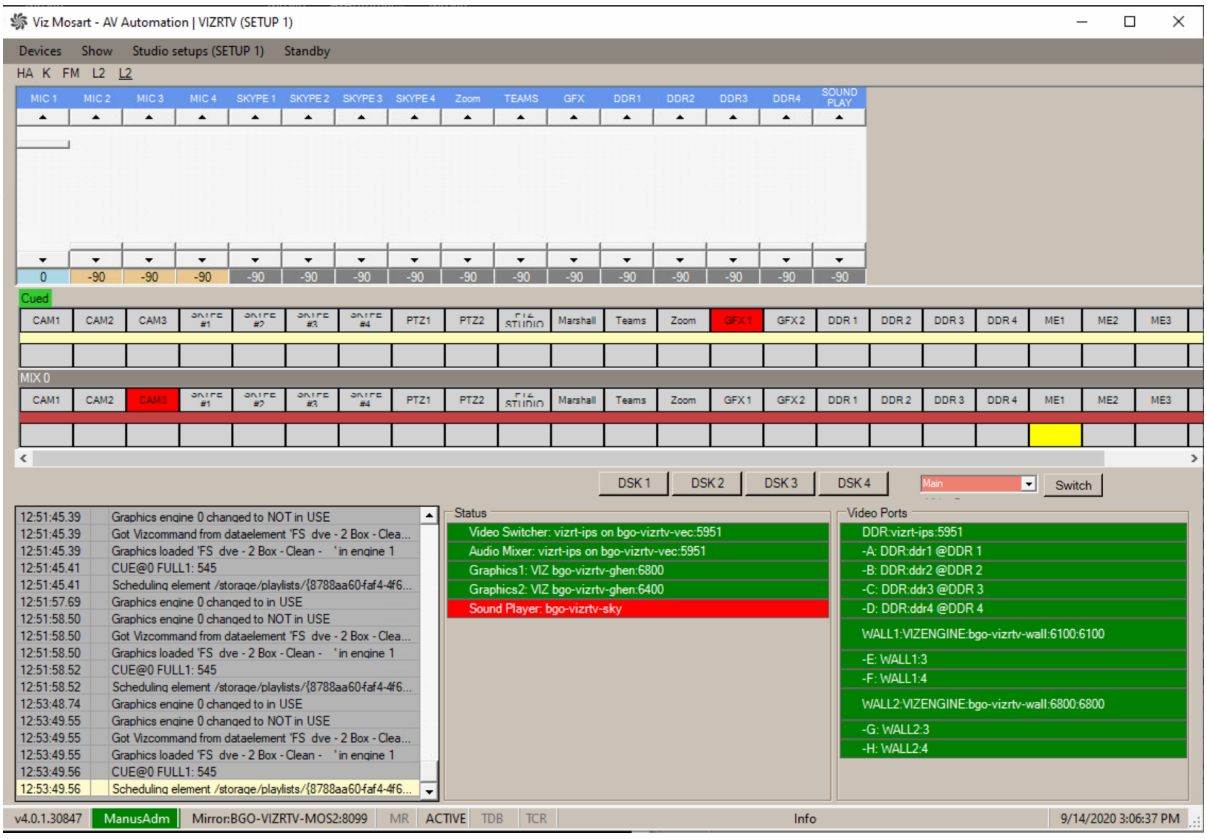
4 AV Automation

AV Automation (program file *MMAVAutomation*) forms the bridge between Viz Mosart and the devices (for example a vision mixer or video server) in the studio. It also controls full-screen graphics (but *not* Overlay Graphics engines), which from a Viz Mosart perspective, are treated as *clips*.

The following AV Automation topics are presented:

- Overview
- Menu Selections
- Logging and Status

4.1 Overview



4.1.1 AV Automation Tasks

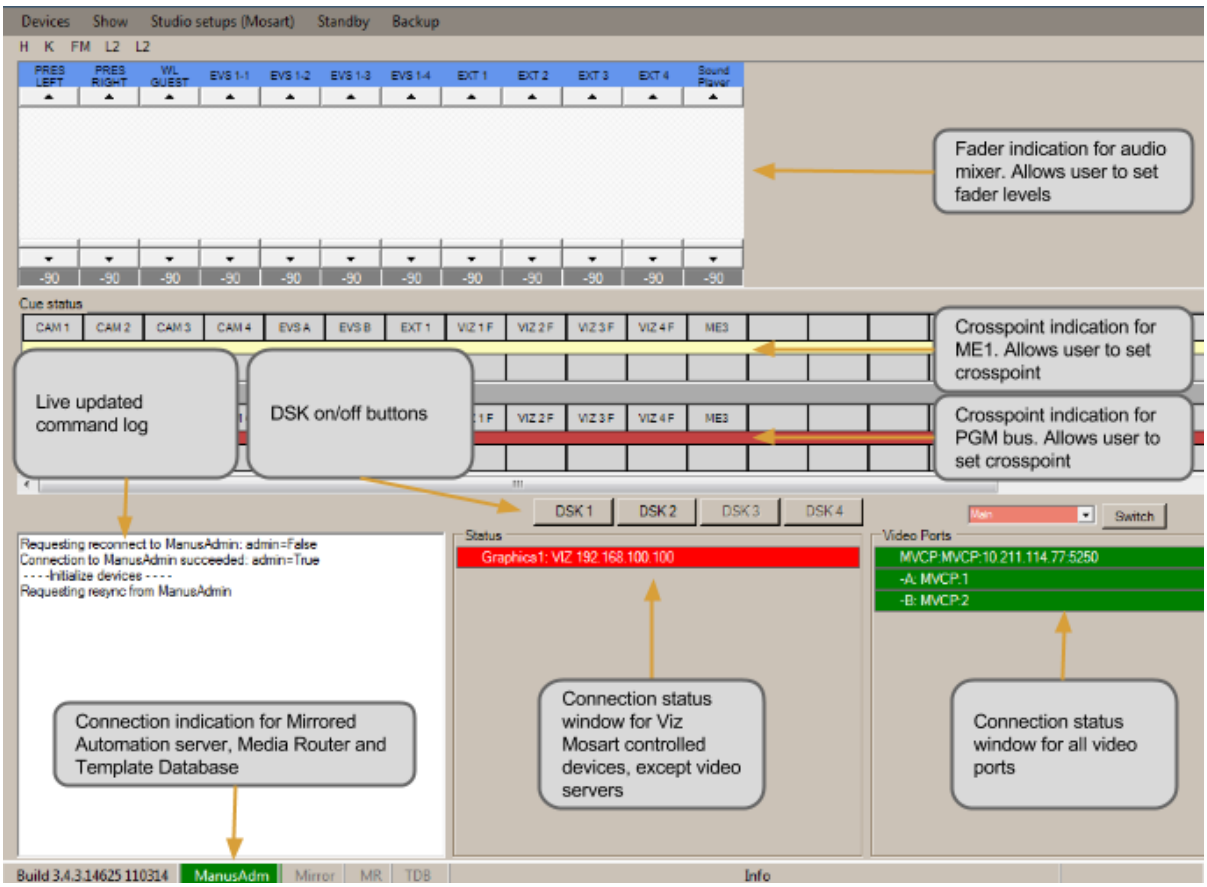
The principal tasks for AV Automation are:

- To *Cue* a clip (prepare) when in preview
- To *Play* a clip when taken on air
- To *Manage control commands* to studio devices.

Copyright © 2024 Vizrt

4.1.2 AV Automation Characteristics

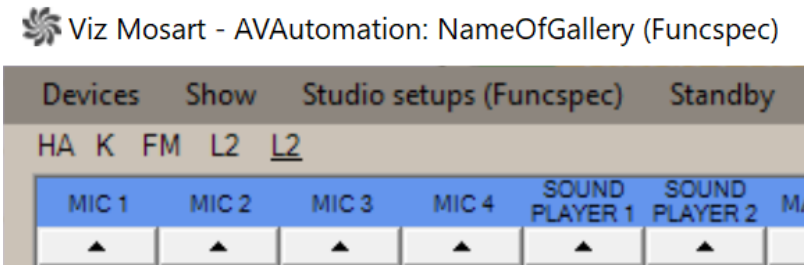
- AV Automation is the device handler of Viz Mosart. It communicates with all connected devices such as cameras, video servers and vision mixers.
- All device drivers are stored in AV Automation, offering the administrator a single point of configuration.
- All device commands are stored as predefined Viz Mosart *templates*. They are managed in the Template Editor, and saved in `C:\channeltemplates` or in a separate template database.
- As well as providing tools for configuring these devices, it includes Template Editor, for defining and customizing templates.
- On the fly commands can then be issued directly to each device through the Viz Mosart GUI, or as pre-programmed operations driven by the rundown of a connected NRCS.



- The **Switch** button switches video server salvo. The drop-down menu to the left of **Switch**, lists the available video server salvos.

Copyright © 2024 Vizrt

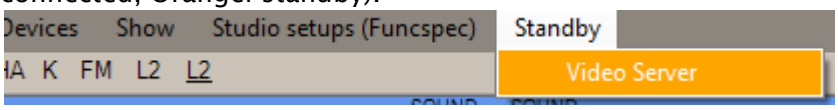
4.2 Menu Selections



- **Devices**:
  - **Preferences**: Opens AV Automation Device Properties.
  - **Settings**: Opens Mosart Administrator Settings.
  - **Template editor**: Opens Template Editor.
  - **A/V Setup**: Opens Audio and Video Setup.
  - **Record**: See Recording.
  - **Change password**: Creates or changes the Other Template Functionality.
- **Show**:
  - **Show MVCP Control**: Deprecated.
  - **Instrumentation Panel**: Opens the Mosart Instrumentation Panel.
- **Studio setups**: Selects the template set to use for the current show. These template sets are *not* available for selection:
  - those marked as **Hide from user**
  - the special template set **DirectTakes**.

> ⚠ **Note**: For more information on the concepts of *template set, Directtakes,* and *direct take* templates, please refer to the *Template Sets* section in the Templates chapter of the Viz Mosart User Guide.

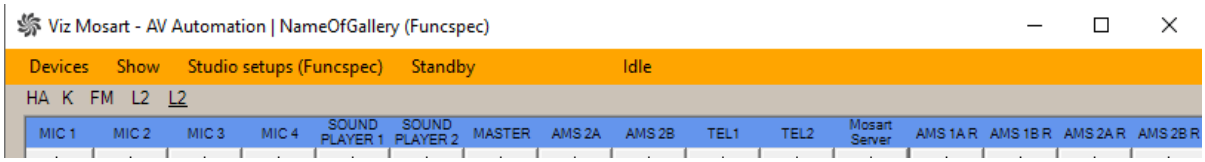
- **Standby**: Lists all configured devices and their connection status (Green: connected, Red: not connected, Orange: standby).



From this menu, a device can be placed in Standby (or taken out of Standby). (*Standby* means the device is not under control of Viz Mosart; no commands will be sent to it from Mosart.)
The Viz Mosart operator can also perform this action from the main UI. See the **Viz Mosart User Interface** section of the *Viz Mosart User Guide*, under *Main Menu*, *Standby* for further details.

- **Idle**: This is an indicator that AV Automation is in idle mode, not a menu option. In this state, the menu bar turns orange.

Copyright © 2024 Vizrt



As long as AV Automation is *not* in idle mode, there is no 'Idle' in the menu bar.

4.3 Logging And Status

Below the fader and switch representations are three monitoring panels.

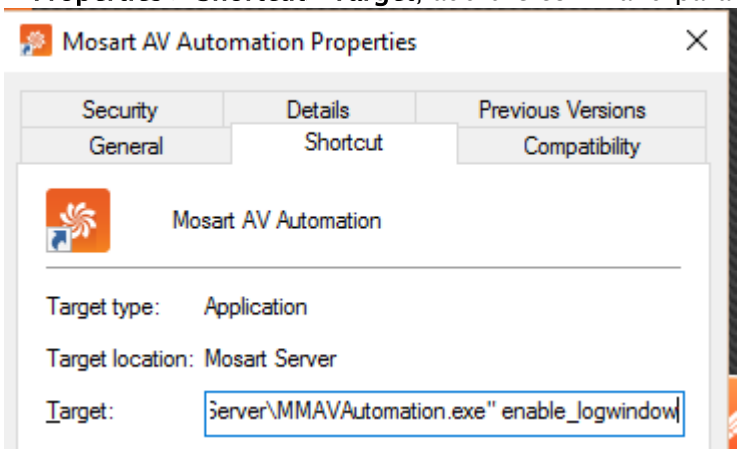
4.3.1 Logging Panel

To the left is a Logging panel with time stamped event details. Logging is disabled by default.

To Enable Logging

There are two options:

- Right-click in the panel and select **Enable**
- In **Properties > Shortcut> Target**, add the command parameter *enable_logwindow*.



Logging will then be enabled at AVAutomation start-up.

To Clear Logging

- Clear the Logging panel by double-clicking in the panel.

The following AV Automation topics are presented:

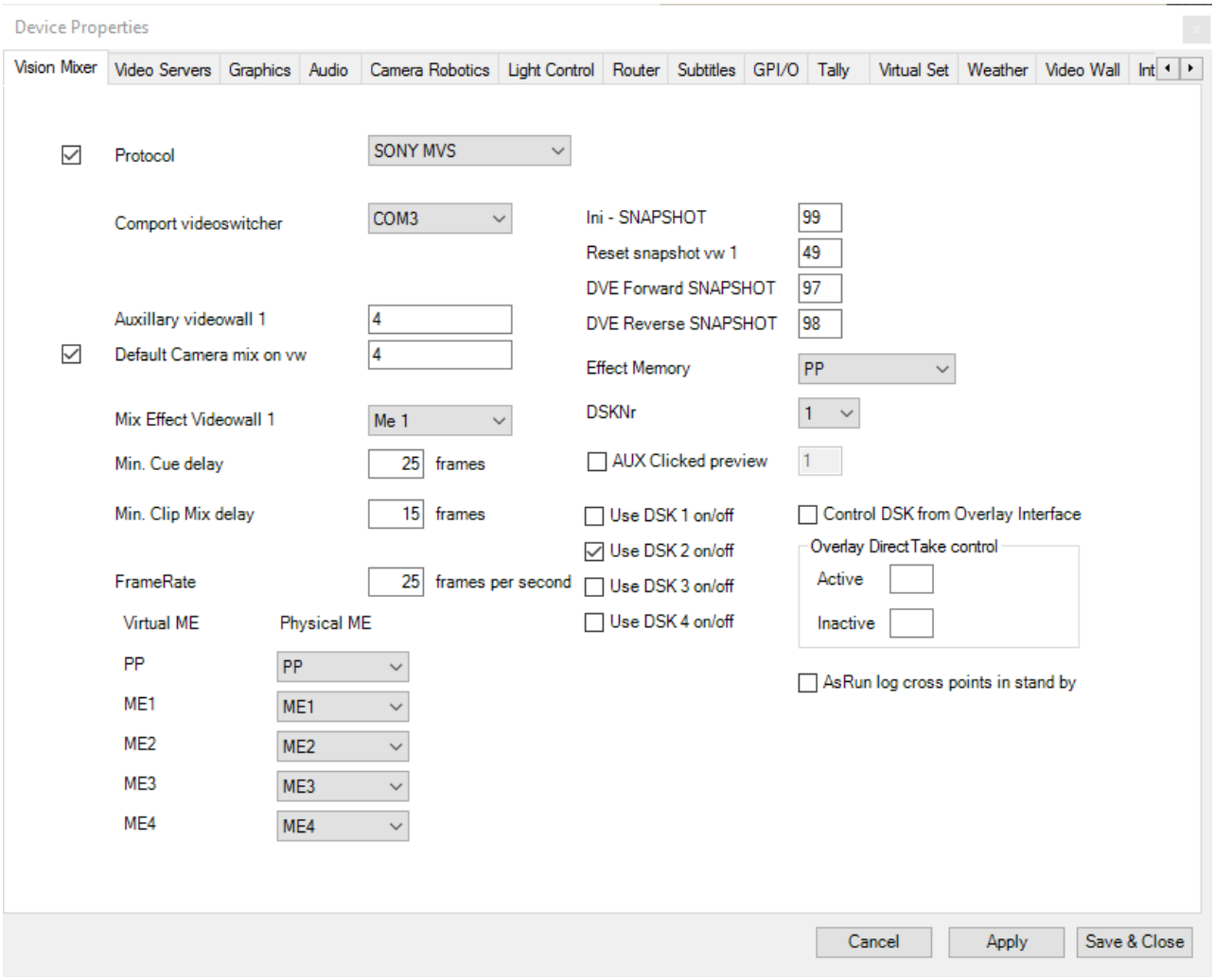
Copyright © 2024 Vizrt

- AV Automation Device Properties
- Audio and Video Setup
- Device Connection Strings
- Template Editor
- Mosart Instrumentation Panel
- Recording

Copyright © 2024 Vizrt

4.4 AV Automation Device Properties

AV Automation can send commands to all connected broadcast devices. For the commands to be received, and interpreted according to the device's specifications, the device must be configured within Viz Mosart.



- To configure device properties in AV Automation, select **Devices > Properties**.

Note: AV Automation must be restarted before any changes will take effect.

The Device Properties menu has three common buttons:

- **Cancel**: Exit this menu without saving any changes.
- **Apply**: Save all changes and reconnect the device on the active tab (only applies to tabs **Vision Mixer**, **Video Servers**, **Graphics**, **Audio**, **Camera Robotics**, **Light Control** and **Subtitles**).
- **Save & Close**: Save all changes and close the menu.

Copyright © 2024 Vizrt

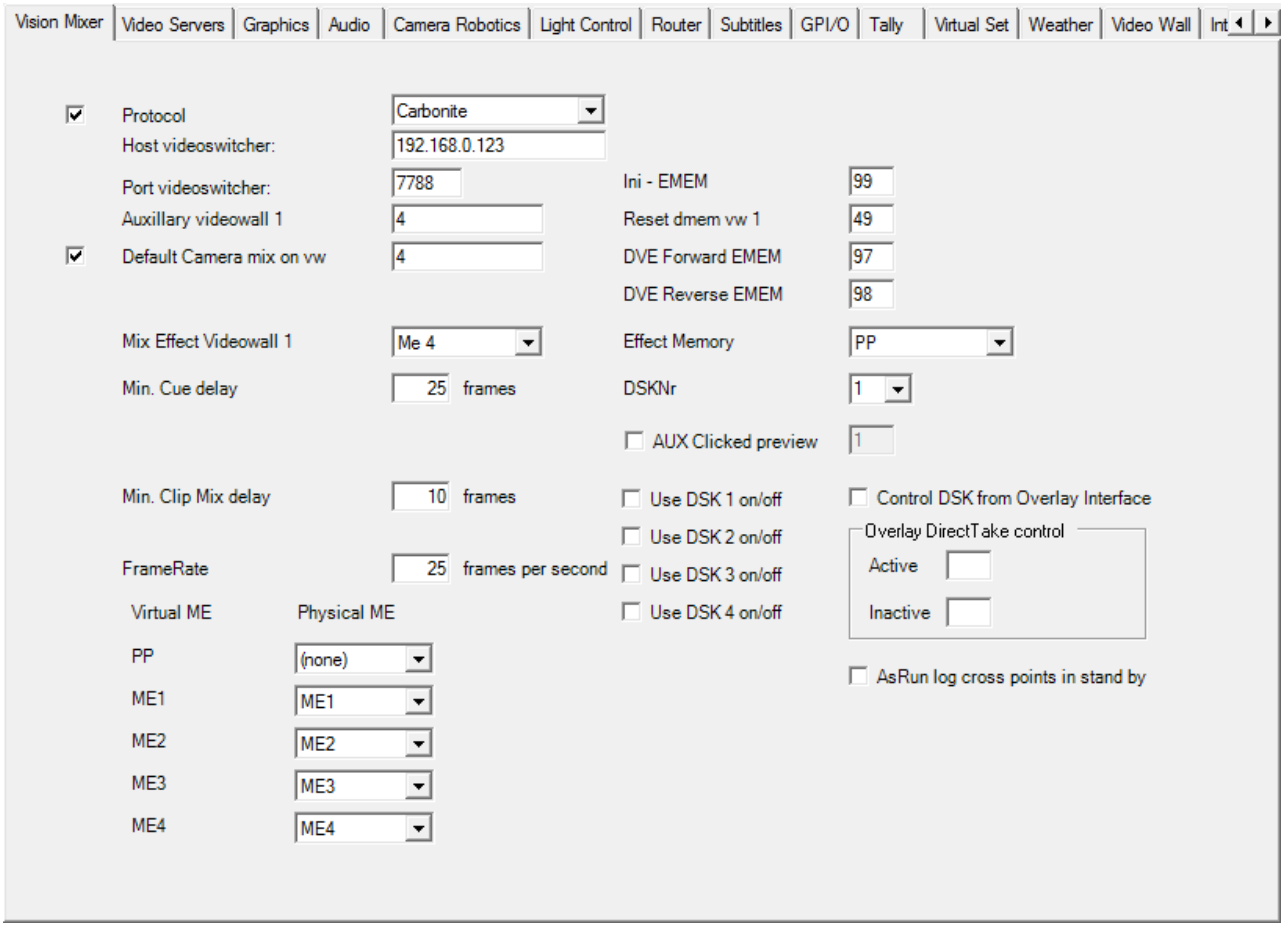
This section presents the following device properties:

- Vision Mixer
- Video Servers
- Graphics
- Audio
- Camera Robotics
- Light Control
- Router
- Subtitles
- GPIO
- Tally
- Virtual Set
- Weather
- Video Wall
- General
- Vizrt Graphics

Copyright © 2024 Vizrt

4.4.1 Vision Mixer

This dialog menu is accessed from the AV Automation menu bar with **Devices > Properties > Tab: Vision Mixer**.



The exact appearance of the **Vision Mixer** tab depends on your selection from the **Protocol** drop-down menu.

- Vision Mixer Settings
  - Description of the Video Mixer Menu
  - Common Video Mixer Connection Settings

Vision Mixer Settings

There are several settings required for Viz Mosart to effectively control a specific video mixer.

Copyright © 2024 Vizrt

Description of the Video Mixer Menu

> ⚠ **Note**: Most devices, especially ones controlled through IP, often have their own config interface (for example, accessible through a web interface) where various parameters such as host/port can be inspected and modified as needed.
> Values found in this device config interface should be used for setting their corresponding properties that are described below.

- **Protocol (check box)**: Check the box to enable video switcher control.
- **Protocol**: Select a protocol supported by your switcher:
  - ACUITY (See **Note** below)
  - CARBONITE

    > ⚠ **Note**: For ACUITY and CARBONITE video switcher protocols, a new physical ME named *ME0* is introduced. *ME0* shall be used to address the P/P on newer/ updated Ross video switchers. This mapping is set up in **AV Automation > Device Properties > Vision Mixer**. Use this mapping to control a Ross Carbonite Ultra video switcher from Viz Mosart.

  - GVG200: GVG/Philips/Thomson
  - GVG4000: Ross Synergy
  - GVG4000 V2
  - GV CPL
  - GV DD35
  - GV KAYENNE PRIMARY
  - GV KAYENNE SECONDARY
  - GV ZODIAK
  - KAHUNA: Snell Kahuna/Kula
  - KAHUNA IP: Grass Valley Kahuna/Kula
  - MASTERPIECE
  - NOVA700: Echolab
  - SONY: Sony compatible video switchers
  - SONY BVS: Support for legacy SONY DVS/BVS 300 series
  - SONY MVS: An extension to the protocol for the SONY MVS8000
  - SONY SERIAL TALLY
  - VIZRT-IPS
- **Comport videoswitcher**: Defines the serial port the video switcher is connected to.
- **Host videoswitcher/Port videoswitcher**: Defines the IP and port the video switcher is connected to.
  - This labeling on the settings page depends on what type of mixer is selected.
  - Host / Port is valid for mixers that are using TCP/IP protocol, or a com-port for serial port communication, or a host- or client-port for UDP.
  - Com-port / Host-port labeling depends on the mixer model. See Vision Mixer below.
- **Auxiliary videowall 1**: This is the auxiliary output from the video switcher to the video wall.

Copyright © 2024 Vizrt

- **Default Camera mix on vw (check box)**: Enables the default camera in program when entering video wall mode.
- **Default Camera mix on vw**: Enter the camera number you want to use as the default camera in program when entering video wall mode.
- **Mix Effect Videowall 1**: Select the ME to use when enabling mixing in video wall mode.
- **Min. Cue delay**: Value in frames which sets the minimum delay before Viz Mosart cues the next template in preview.
- **Min. Clip Mix delay**: Value in frames which sets the minimum delay from starting the server to starting the video switcher transition.
- **FrameRate**: Specify the number of video frames per second for the system.
- **Ini-EMEM**: Initial register to recall when starting the automation. Depending on the Protocol chosen, the caption may alternatively be **Ini - DMEM** or **Ini - SNAPSHOT**.
- **Reset EMEM vw 1**: Register to recall to normalize the ME before entering video wall mode. (Depending on the Protocol chosen, the caption may alternatively be **Reset dmem vw 1** or **Reset snapshot vw 1**).
- **DVE Forward EMEM**: (not supported by Sony) EMEM to recall for running DVE forward. (Depending on the Protocol chosen, the caption may alternatively be **DVE Forward DMEM** or **DVE Forward SNAPSHOT**).
- **DVE Reverse EMEM**: (not supported by Sony) EMEM to recall for running DVE backward. (Depending on the Protocol chosen, the caption may alternatively be **DVE Reverse DMEM** or **DVE Reverse SNAPSHOT**).
- **Effect Memory**: ME to use when recalling registers for Effect use.
- **DSKNr**: Downstream keyer to use for the DSK on/off functionality.
- **AUX Clicked preview (check box and number)**: Check the box to enable AUX clicked preview, and specify the AUX bus connected to the preview monitor. When activated, it is possible to click an element in the Viz Mosart rundown, to preview the source on the monitor, via the specified AUX bus.
- **Use DSK n on/off**: Enable or disable the on/off functionality for DSK 1-N, where N depends on the protocol. Most protocols have 4 DSKs, but there could be as many as 8 DSKs.
- **Control DSK from Overlay Interface**: Check box to enable DSK to be on only when overlay graphics is present.
- **Overlay Direct Take control:**
  - **Active**: Viz Mosart will run the direct take entered here when an overlay graphic goes On Air.
  - **Inactive**: Viz Mosart will run the direct take entered here when an overlay graphic goes Off Air.
- **AsRun log cross points in stand by (check box)**: Check box to set AsRun log in stand-by.

Common Video Mixer Connection Settings

The following table presents some recommended or additional values for a range of video mixer drivers.

Copyright © 2024 Vizrt

Driver	Settings	Configuration Interface Notes
GV CPL	Port video switcher: 5000	This driver uses UDP protocol
	Client Port video switcher: 6004	Additional client port property used for receiving callbacks
Sony	SonyVideoSwitcherConfig.xml setting <item name="ReadTimeout" value="100" />	ReadTimeout is possible to change via the SonyVideoSwitcherConfig.xml configuration file. Default set to 200 milliseconds
VIZRT ENGINE		When working with a Viz Engine in a *switcherless*, or *Engine Switcher* configuration, refer to the specific settings in section Viz Engine as a Switcher.

Note: Configuration files can be used to configure some settings. See section Vision Mixer Configuration Files for examples.

Copyright © 2024 Vizrt

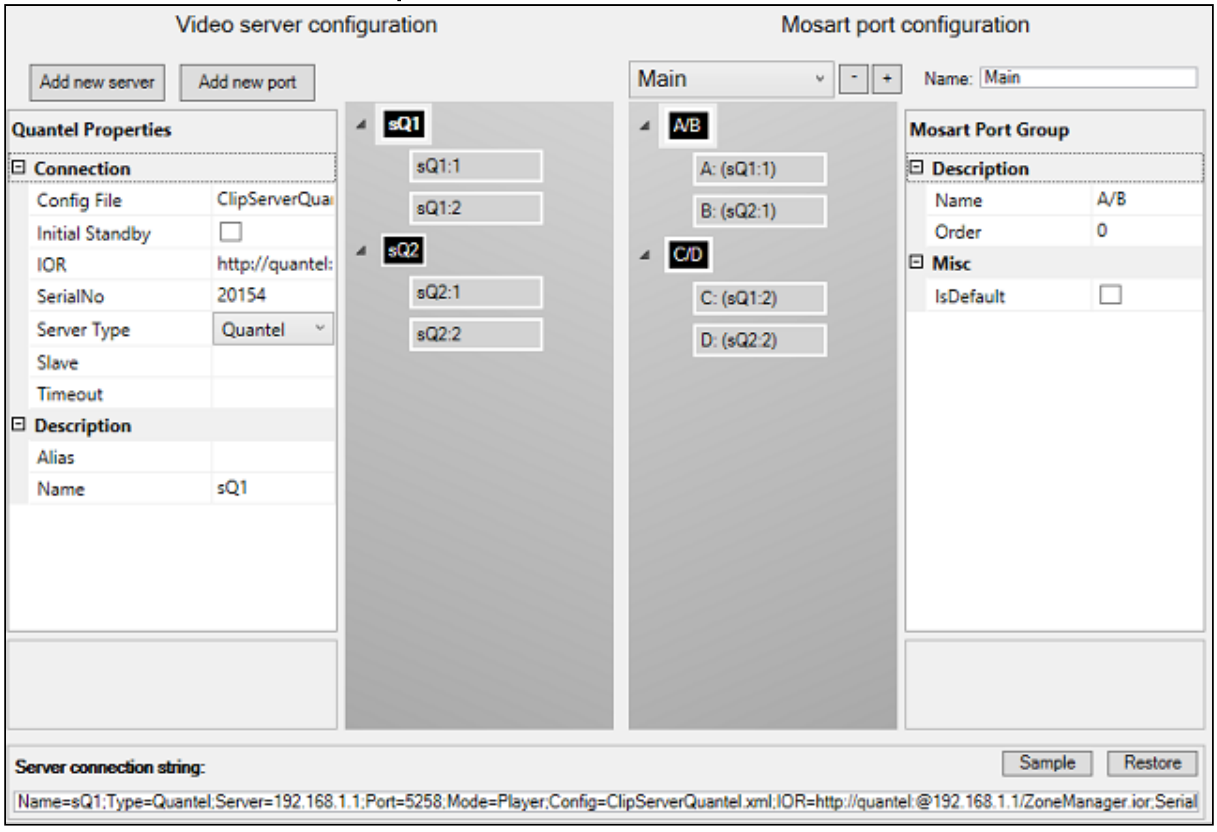
4.4.2 Video Servers

Viz Mosart is capable of controlling video servers from many different manufacturers. Although the end result is the same, each video server type operates in a slightly different way.

- Working with Video Server Configurations
  - Video Server Properties
  - Video Server Port Properties
- Working with Servers and Ports
- Mosart Port Configurations
  - Virtual Server Ports
  - Working with Mosart Port Configurations
  - Working with Salvos
- Video Server Recording Setup Example
  - Configuring Viz Mosart

It is essential that the correct settings are defined in AV Automation for Viz Mosart to effectively and correctly control your video server. You reach these settings with

- **AVAutomation > Devices > Properties> Video Servers** tab.



- The *left* side presents details of the physical video server, and any port configurations. The example above presents a Quantel server, sQ1 (as selected from the **Server Type** drop-down menu), and one other server (sQ2), which could also be a Quantel server.

Copyright © 2024 Vizrt

Server sQ1 has 2 ports defined, sQ1:1 + sQ1:2.
Server sQ2 has 2 ports defined, sQ2:1 + sQ2:2.

- The *right* side presents details of the virtual connection within Viz Mosart, with salvo definitions.
  This is described in Video Servers.

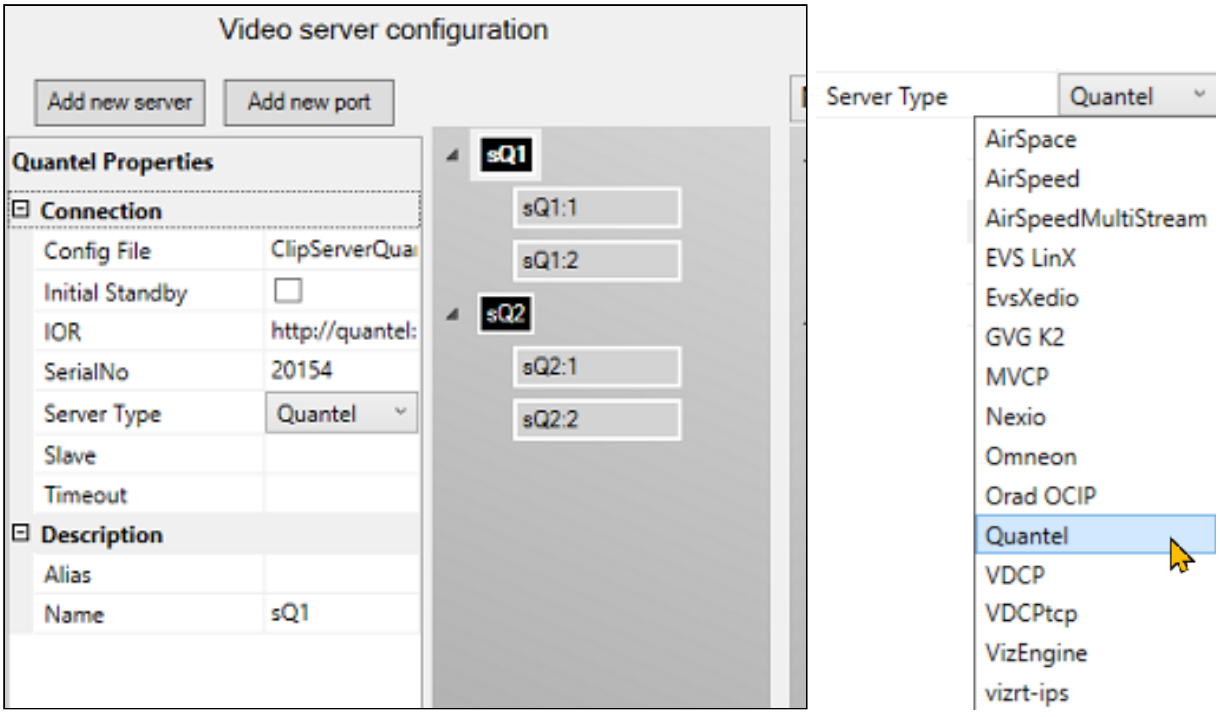
This rest of this section contains the following topics, presented according to the configuration menu illustrated above, from left to right:

- Working with Video Server Configurations
  - Video Server Properties
  - Video Server Port Properties
- Working with Servers and Ports
- Mosart Port Configurations
  - Virtual Server Ports
  - Working with Mosart Port Configurations
  - Working with Salvos
- Video Server Recording Setup Example
  - Configuring Viz Mosart

Working with Video Server Configurations

You can *add* or *remove* video servers to your Viz Mosart system, and *modify* the configuration of existing hardware.

Viz Mosart supports a wide range of video servers. These can be viewed from the **Server Type** drop-down menu.



Copyright © 2024 Vizrt

> **Note:** Only compatible combinations of Media Administrator and AV Automation drivers can be defined. Please contact Vizrt Support for further details.

- The **Name** property links the selected video server to a Media Administrator *properties* line.
- For an explanation of the various configuration properties, please refer to the specific video server model, as presented in Video Servers below.
- After configuration, restart Media Administrator and AV Automation, and ensure that the GUI is connected to the required server.
- Verify that all indicator lights for the selected video server go green.

> **Tip:** If you do not get green lights, and you see a message similar to this in the Mosart log:
> `Failed connecting to video server K2:`
> `The source was not found, but some or all event logs could not be searched.`
> `Inaccessible logs: Security.`
> you can try to *start AVAutomation* as *Administrator*. This is a known issue with Grass Valley servers. More info here: http://www.gvgdevelopers.com/concrete/apis/appserver_api/windows/

Video Server Properties

Here are some guidelines for common video servers that Viz Mosart can work with.

Common Settings

Several of the settings are always required, as listed here

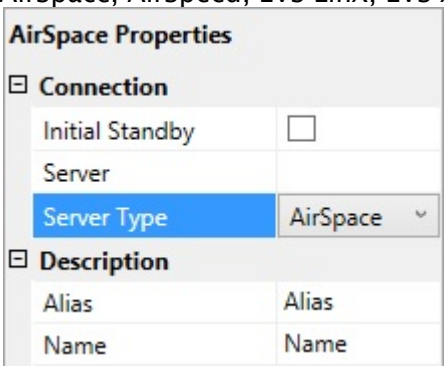
- **Initial Standby:** If selected, the server is forced to start in *Standby* mode.
- **Server:** Defines the hostname or IP address of the video server.

  > **Note**: For *EVSXedio*, a value for **Server** is not required.

- **Server Type:** Name of supported video server. For example, AirSpace, AirSpeed, EVS LinX, Orad OCIP.
- **Alias:** Defines the name of the server to display in the Viz Mosart GUI and Timing Display.
- **Name:** Internal name of the server, displayed in AV Automation.

Copyright © 2024 Vizrt

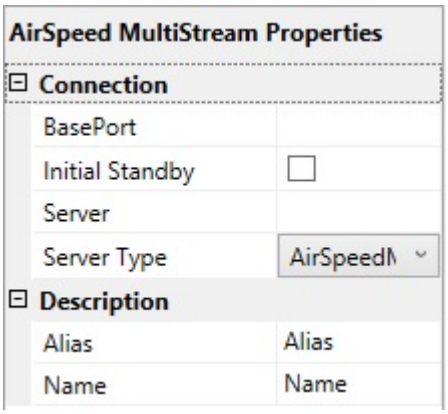
The above settings are typical for
AirSpace, AirSpeed, EVS LinX, EVS Xedio, OradOcip video servers.



Some common video server properties are summarized below:

- AirSpeed MultiStream
- Grass Valley K2
- Nexio
- MVCP
- Omneon
- Quantel
- TriCaster
- VDCP
- Viz Engine
- Vizrt-ips

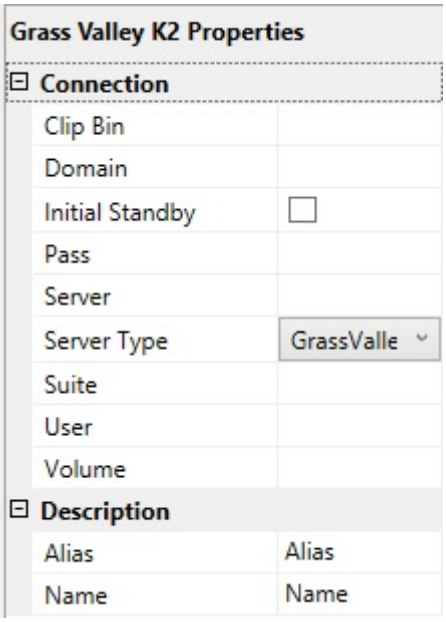
AirSpeed MultiStream



- **BasePort:** The first port Viz Mosart uses to communicate with the AirSpeed MultiStream server.
  Default: `59451.`

Grass Valley K2

Connection to K2 is initialized with a ping (ICMP) from Mosart to the server to confirm the connection. Then it uses port 3811 to communicate with the K2 server. Both must be open for Mosart to be able to connect.
Manus Admin uses a separate port when it connects: 49171 (a range is OK. Use 49168-49172)

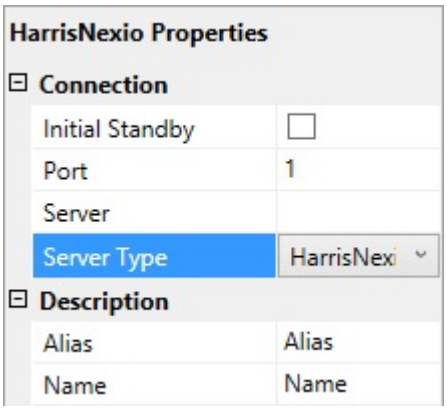


- **Clip Bin**: This value, together with that of *Volume* (see below), defines the default server location of clips to be played. This default location is used only when the location is not provided by other means. (In particular, it is not used when the setting *IgnoreBin* (in Media Administrator) is true for this server.) The value of *Clip Bin* should designate a bin in the volume designated by the value of the *Volume* setting.
- **Domain**: The domain as part of the user credentials.
- **Pass:** The password as part of the user credentials.
- **Suite:** Arbitrary value. However, follow these guidelines:
  - For each K2 server, two or even three unique Suite values must be used:
    - First, the values for Media Administrator and AV Automation must be different.
    - Second, in a redundancy setup with Main and Backup Mosart server, the two values for Media Administrator must be different.

      > ⚠ **Note**: In this constellation, both Main and Backup AV Automation should use the *same* value, different from the two Media Administrator values.
- **User:** The username as part of the user credentials.
- **Volume:** This value, together with that of *Clip Bin* (see above), defines the default server location of clips to be played. This default location is used only when the location is not provided by other means. (In particular, it is not used when the setting *IgnoreBin* (in Media Administrator) is true for this server.) The value of *Volume* should designate a server volume.

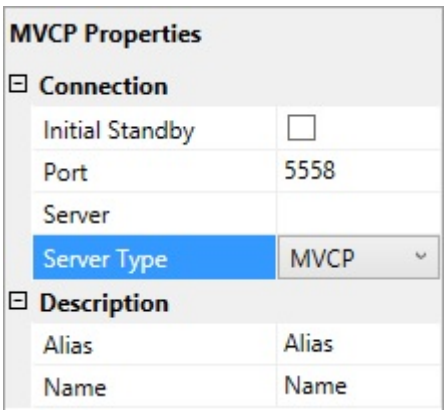
Copyright © 2024 Vizrt

Nexio



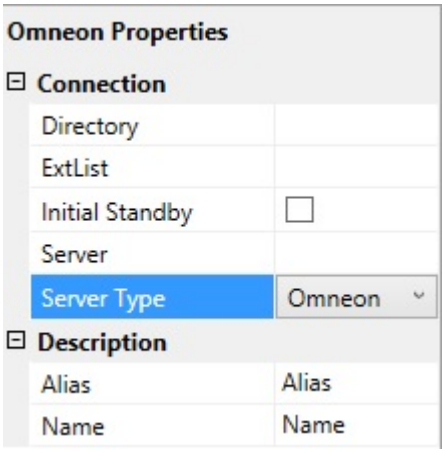
- **Port:** TCP/IP communication port to the Nexio server.

MVCP



- **Port:** TCP/IP communication port to the MVCP server.

Omneon

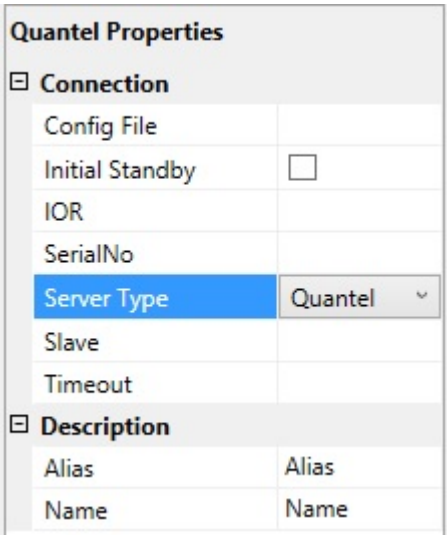


- **Directory:** Defines the directory where the clips are stored.

Copyright © 2024 Vizrt

- **ExtList:** The list of valid file extensions used when listing and querying files on the server. The list is period separated and case sensitive.

Quantel



- **Config File:** The path to the XML configuration file used to define the Viz Mosart - Quantel communication.
- **IOR:** The HTTP link including hostname or IP address to the IOR resource on the **Quantel** ISA manager. Example: http://192.168.100.50:2096/ZoneManager.ior
- **SerialNo:** The serial number of the Quantel playout server.
- **Slave:** The hostname or IP address of the slave/backup IOR.
- **Timeout:** Timeout value for requests from Viz Mosart to ISA manager. If the request exceeds this timeout the server connection is reinitialized. Setting a value here should only be needed for sites experiencing issues with the Quantel connections. Leaving a blank value uses the default timeout.

Recording with Quantel Devices

Quantel requires a duration when preparing a recording. This duration can be set the following ways:

- As part of a template: Specified using *markin* and *markout* properties.
- As part of a keyboard shortcut: Specified by trailing the `ClipName` property with [,duration]. Similar to the examples under the general section Controlling Video Servers from Mosart above.
- As part of template control command: Specified as part of the **Parameter** property. Again, see the general section above.

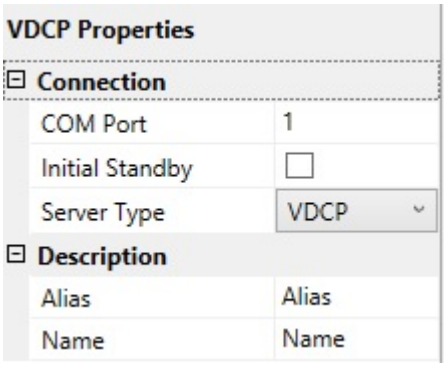
TriCaster

The Vizrt TriCaster uses the Video Servers protocol.

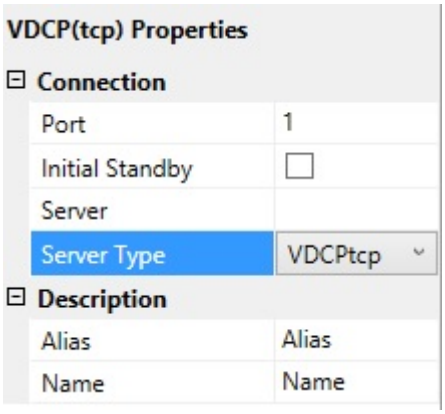
Please refer to the procedure below in Video Servers.

Copyright © 2024 Vizrt

VDCP

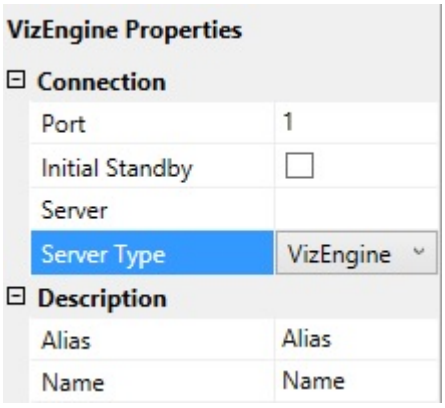


- **COM Port:** The COM port connected to the VDCP video server.



- **Port:** The TCP/IP communication port connected to the VDCP video server.

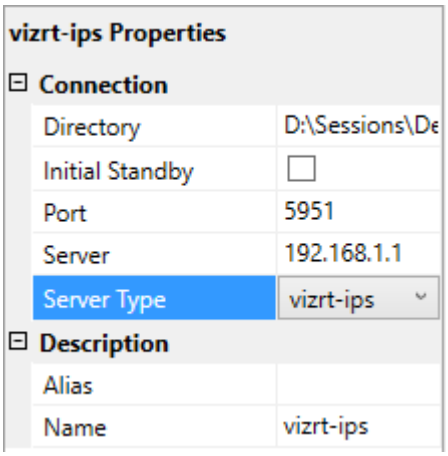
Viz Engine



- **Port:** The TCP/IP communication port connected to the Viz Engine video server.

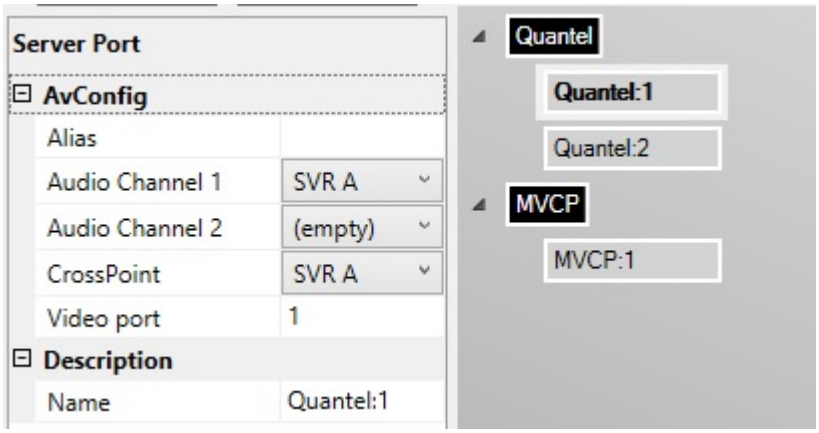
Copyright © 2024 Vizrt

Vizrt-Ips



- **Directory:** Defines the directory where the clips are stored. The following format can be used: `{volume}:/Sessions/{session_name}/Clips/Import`
- **Port:** The TCP/IP communication port connected to the vizrt-ips video server.

Video Server Port Properties



- **Alias**
- **Audio Channel 1:** The audio channel 1 for the selected video port.
- **Audio Channel 2:** The audio channel 2 for the selected video port.
- **CrossPoint:** The video crosspoint for the selected video port.
- **Video port:** The actual port name/number on the video server. Please refer to the documentation of your video server for information on how to find these.

  > ⚠ For the Server Type vizrt-ips, any recording port must have a 'Video port' of the form recN, where N is a number.

- **Name**

Copyright © 2024 Vizrt

Working with Servers and Ports

To Add a Server

1. Click the **Add new server** button
2. In the **Properties** pane on the left, select from **Server Type** and add additional information such as **Port** and **Server host**.

To Remove a Server

- In the **Video Server Configuration** tree, right-click a server node and select **Remove**.

To Add a Port

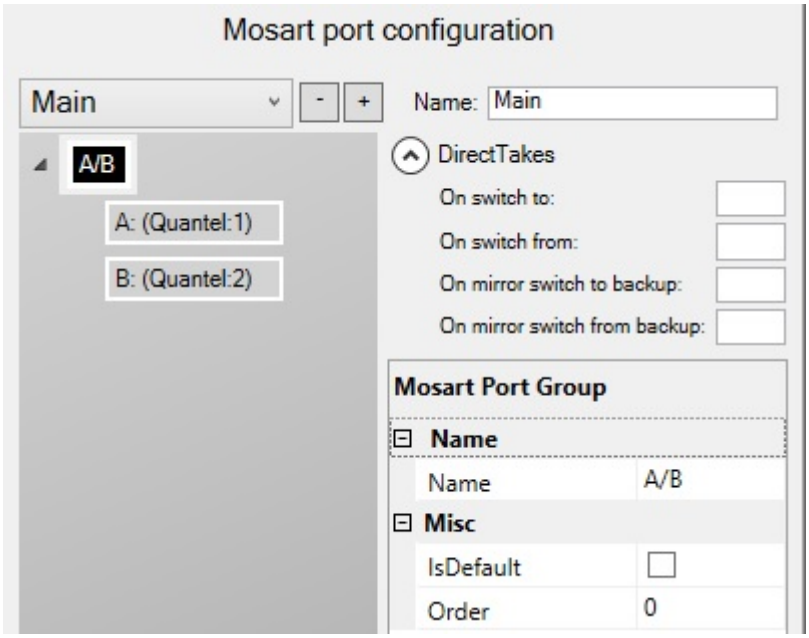
1. In the **Video Server Configuration** tree, select a server node
2. Click the **Add new port** button, or right-click the server node and select **Add new port**
3. In the property editor on the left, enter the port information.

To Remove a Port

- In the **Video Server Configuration** tree, right-click a port node and select **Remove**.

Mosart Port Configurations

Virtual Server Ports



- **Add new salvo:** Use the + icon to add another empty virtual video server salvo.
- **Remove a salvo:** Use the - icon to remove/delete a virtual video server salvo.

Copyright © 2024 Vizrt

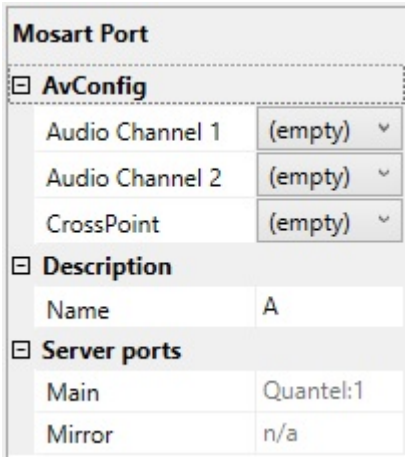
- **Switch between salvos:** Use the drop-down menu to select another virtual video server salvo. If the *Media Administrator* setting **Enable dynamic configuration** (see Media Administrator Properties Editor, Configuration) is *True*, selecting a salvo will also have the effect that *Media Administrator* connects only to
  - those servers that are part of the salvo selected, and
  - those servers which have *Static=True* in the *Media Administrator* connection string. (See Media Administrator Properties Editor, Connection Media Search Servers, Video clip Server <n>.)
- **Name:** Defines the name of the virtual video server salvo.

Direct Takes

- **On switch to:** The Recall Nr (see Template Properties) of the direct take template that should be taken when switching to this virtual server salvo.
- **On switch from:** The Recall Nr of the direct take template that should be taken when switching from this virtual server salvo.
- **On mirror switch to backup:** The Recall Nr of the direct take template that should be taken when switching to a backup server using this virtual server salvo.
- **On mirror switch from backup:** The Recall Nr of the direct take template that should be taken when switching from a backup server using this virtual server salvo.

Mosart Port Group

- **Name:** The name of the virtual server group. This name is displayed in the server part of the templates.
- **IsDefault:** Enable to set the default and preferred video port group for assets residing on multiple systems.
- **Order:** See the documentation of the Media Router. If a Media Router is not used, this value can be left to the default *0*.



Mosart Port

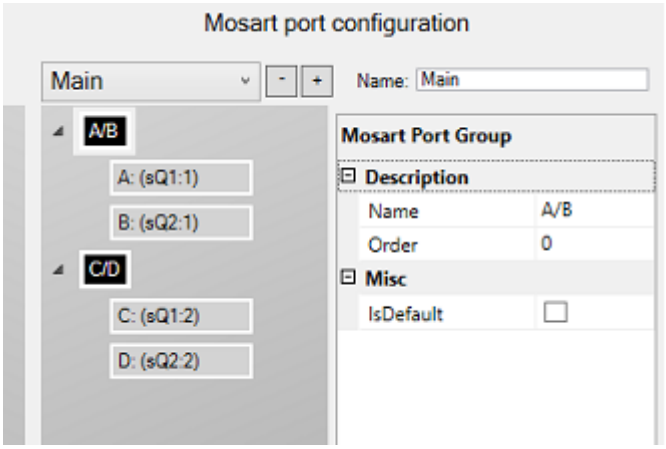
- **Audio Channel 1:** The virtual audio fader 1 for the selected video port. If nothing is selected here, the audio fader from the physical part is used. Default: *Empty*

Copyright © 2024 Vizrt

- **Audio Channel 2:** The virtual audio fader 2 for the selected video port. If nothing is selected here, the audio fader from the physical part is used. Default: *Empty*
- **CrossPoint:** The virtual video crosspoint for the selected video port. If nothing is selected here, the video crosspoint from the physical part is used. Default: *Empty*
- **Name:** The name of the virtual server port, this name is displayed in the server part of the templates.
- **Main:** Auto-generated display name of the main video port.
- **Mirror:** Auto-generated display name of the mirror video port.

Working with Mosart Port Configurations

The panel on the right-side of the details presented in the **Video Servers** tab lists any configured ports. This represents a virtual configuration, presented as Viz Mosart port groupings, or *salvos*. A salvo is a configured constellation of virtual ports (see Video Servers below).



To Add A Virtual Port Group

- Right-click in the **Mosart Port Configuration** tree (not on a node) and select either:
  - **Add virtual ripple group** to add a new virtual ripple group. The default ripple group names are *A/B, C/D, E/F* etc.
  - **Add virtual preview group** to add a new green virtual preview group node to be used for preview ports. The default preview group name is *P*.
  - **Add virtual recording group** to add a new red virtual recording group node to be used for recording ports. The default recording group name is *Rec* .

You can edit or rename the selected group using the **Properties Editor** on the right.

To Remove A Virtual Port Group

You can remove a virtual ripple group, preview group or recording group.

1. In the **Mosart Port Configuration** tree, right-click any ripple/preview/recording group node and select **Remove group <name>**.
   Alternatively, select a node and press the **Delete** key.
2. When prompted, select *Yes* to confirm that you want to remove the group from all salvos.

Copyright © 2024 Vizrt

To Add A Virtual Port

1. In the **Mosart Port Configuration** tree, right-click any group node and select **Add virtual port**.

If the port was added to a *preview* group, the added port will be a preview port with default name *P*. If the port was added to a *recording* group, the added port will be a recording port with default name *Rec*.

> ⚠ **Note:** Do not change the name *Rec*!

To Remove A Virtual Port

1. In the **Mosart Port Configuration** tree, right-click any *virtual port node* and select **Remove <name>**. Alternatively, select a node and press the **Delete** key.
2. When prompted, select *Yes* to confirm that you want to remove the group from all salvos.

Working with Salvos

Salvos are used to switch between server setups. They are used when the operator needs to switch between server parks, for example, when switching to a backup salvo.
Salvos can also be used if different video servers are used in different parts of the show, for instance, sports and news, but you still want to use the same ports.

> ⚠ **Note:** It is recommended that the default *Main* and *Backup* salvos are not renamed.

To Connect A Server Port To A Virtual Port

You can connect a server port to a virtual port:

- Select the server port (in the left-hand, Video server configuration part) you want to connect, and drag it to the virtual port (in the right-hand Mosart port configuration part) that you use to represent the port in the selected salvo.

> ⚠ **Note:** If the virtual port is a *recording* port (in the recording group), the server port to connect must also be a recording port: The physical server must have recording capabilities (and the chosen Server Type must allow for recording).

Mirroring

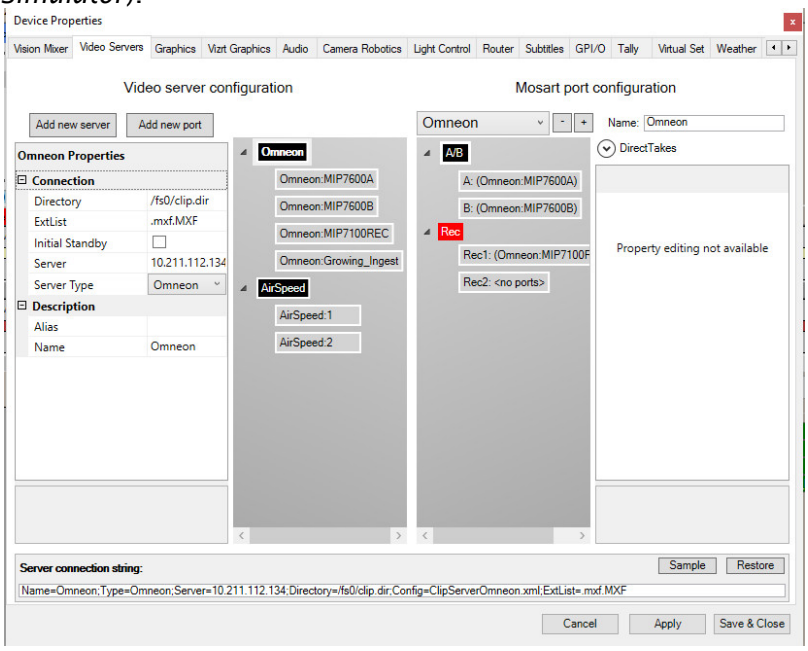
Mirroring is a redundancy setup (with both servers running at the same time) - an effective method for increasing operational security. Mirroring provides the operator a switch to a server with an identical working state, should there be issues with current server.

This method differs from a *Main-Backup* setup, which is a failsafe solution (where either main *or* backup is running, but not both at the same time).

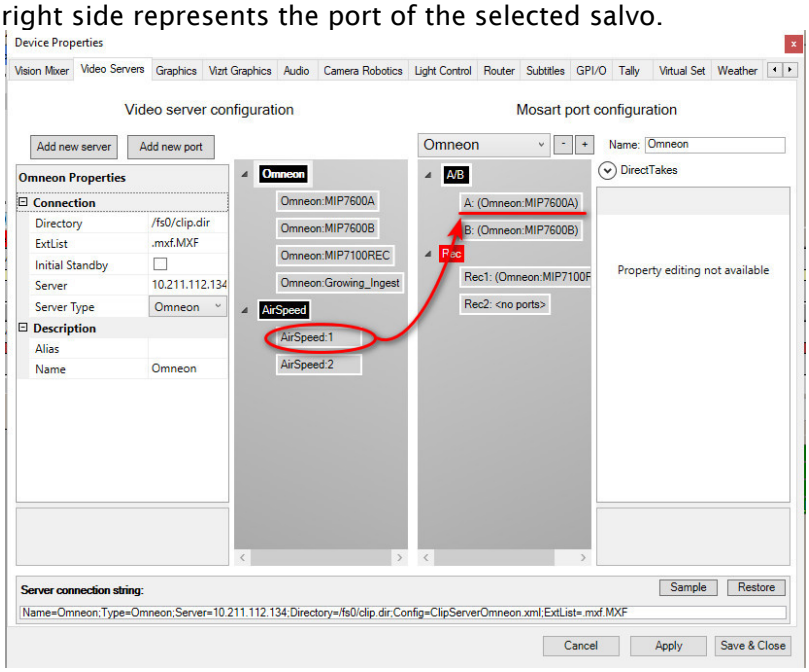
Copyright © 2024 Vizrt

To Connect Two Server Ports To One Virtual Port For Mirroring:

1. You need to have two servers already added (in this example, *Omneon* and *AirSpeed Simulator*).



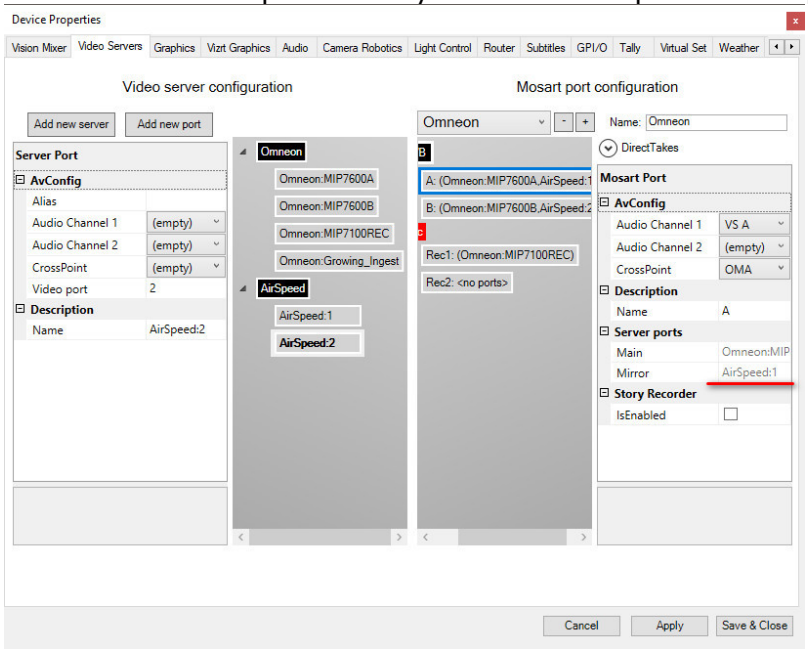
2. Select a server port (left side), then drag and drop it onto the virtual port (right side). The right side represents the port of the selected salvo.



The name changes

Copyright © 2024 Vizrt

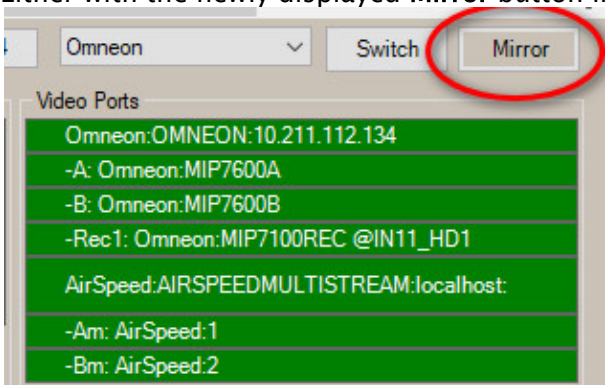
3. Click on the virtual port to verify that the server port from step **2** is now listed as a mirror.



4. Select another server port and drag it to the other virtual port.
5. Click **Apply**, then **Save & Close**.

> **Note:** Only two ports can be mirrored.

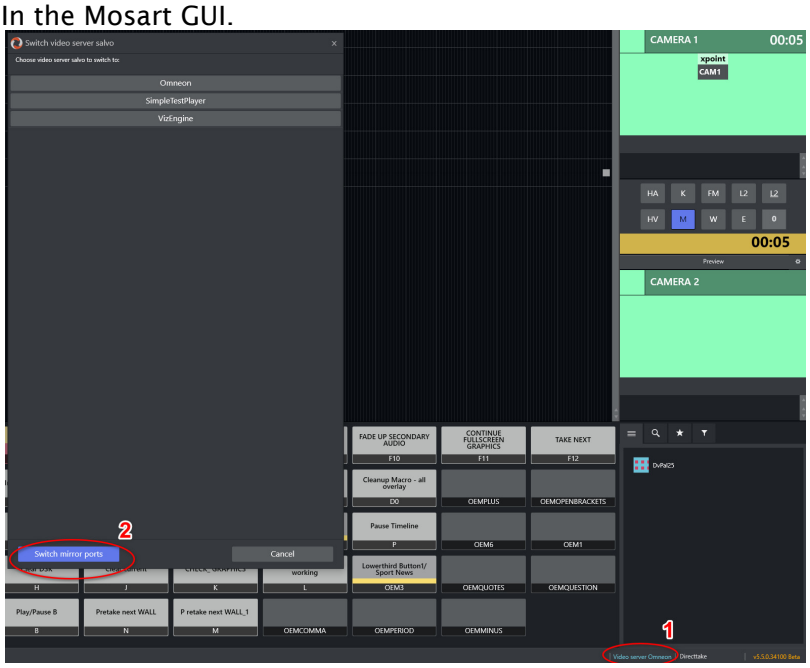
6. Verify that you can now switch between mirrors
   a. Either with the newly displayed **Mirror** button in Av Automation



or

Copyright © 2024 Vizrt

b. In the Mosart GUI.



To Clear Server Links In A Virtual Port Node

Right-click the virtual port node and select **Clear links**.

To Add A Salvo

Click the **Add** button.

To Remove A Salvo

Click the **Delete** button.

To Modify A Salvo

- Click the drop down box above the **Mosart Port Configuration** tree and select the salvo you want to edit.
- Rename the selected salvo in the **Name** textbox.

Video Server Recording Setup Example

As an example of a typical video server workflow, this scenario describes how Viz Mosart can control recording of the output from a TriCaster video mixer to a specified video server.

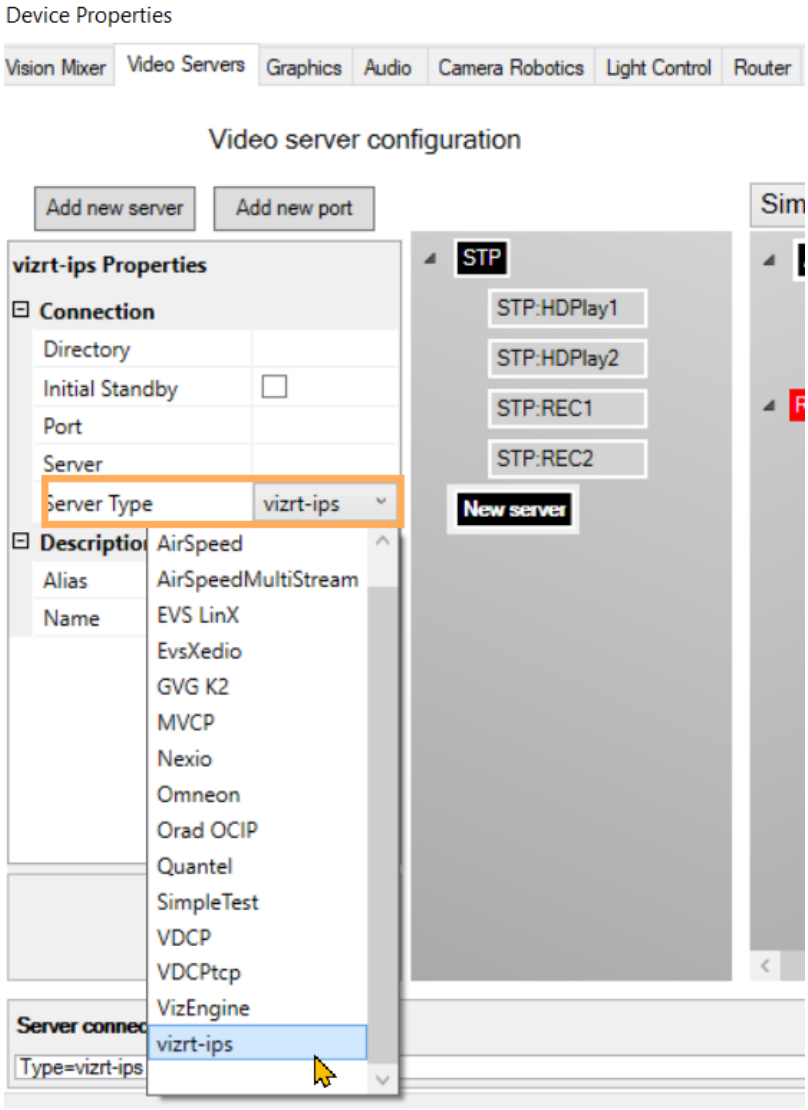
Configuring Viz Mosart

Follow all steps in this section.

Copyright © 2024 Vizrt

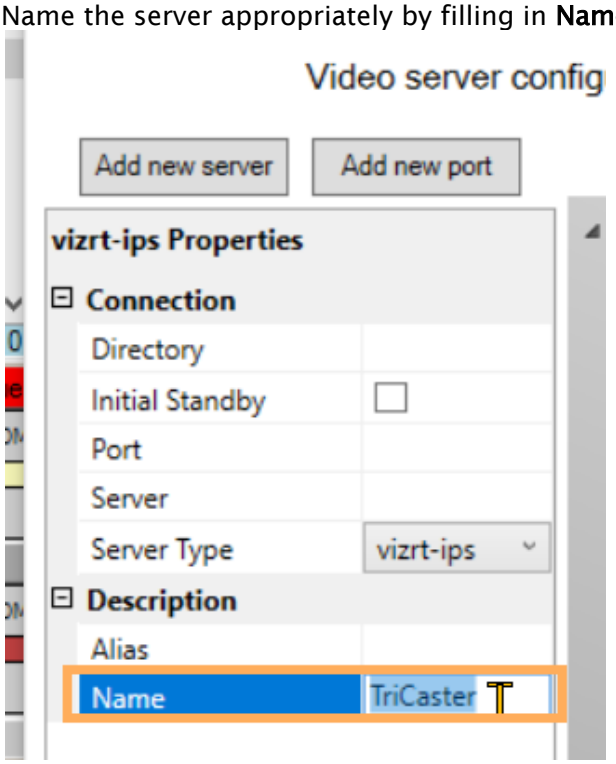
To Configure AV Automation

1. Navigate to **AV Automation > Devices > Properties > Video Servers**.
2. Under **Video server configuration** on the left side, click **Add new server**
3. If not in focus, click on the newly created black box, *New server*.
4. From the **Server Type** drop-down menu, select *vizrt-ips*.

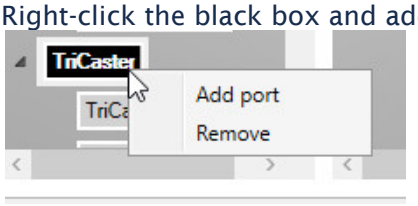


Copyright © 2024 Vizrt

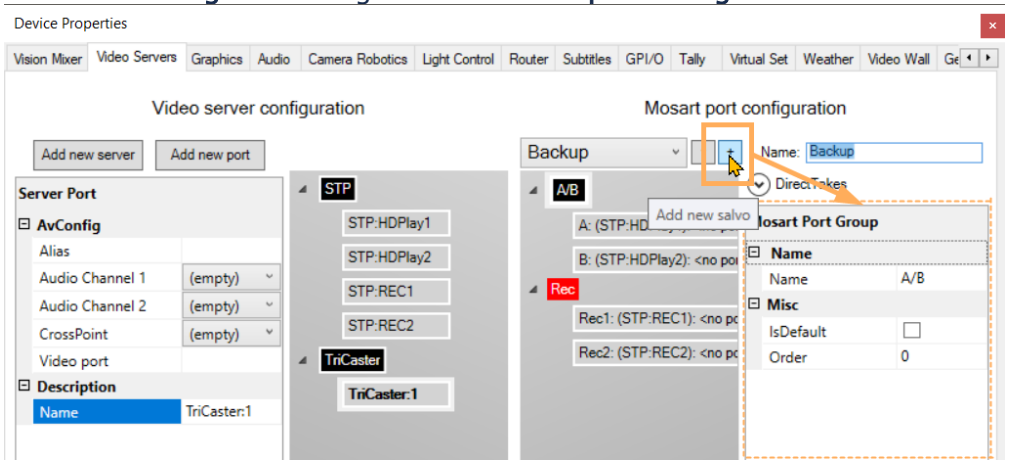
5. Name the server appropriately by filling in **Name**.



6. Right-click the black box and add a port:



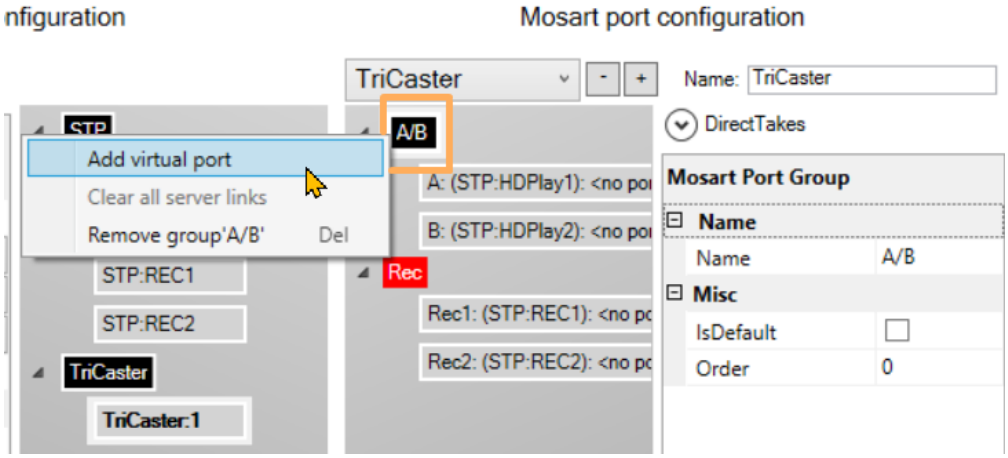
7. Click the **Plus sign** on the right under **Mosart port configuration**.



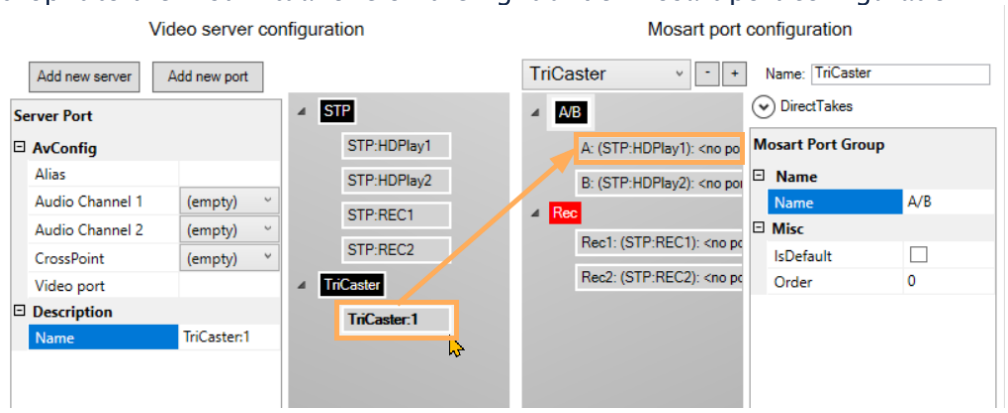
8. In **Name**, give this new virtual server an appropriate identity, like *TriCaster*.

Copyright © 2024 Vizrt

9. Right-click on the *black* box to add a virtual port.



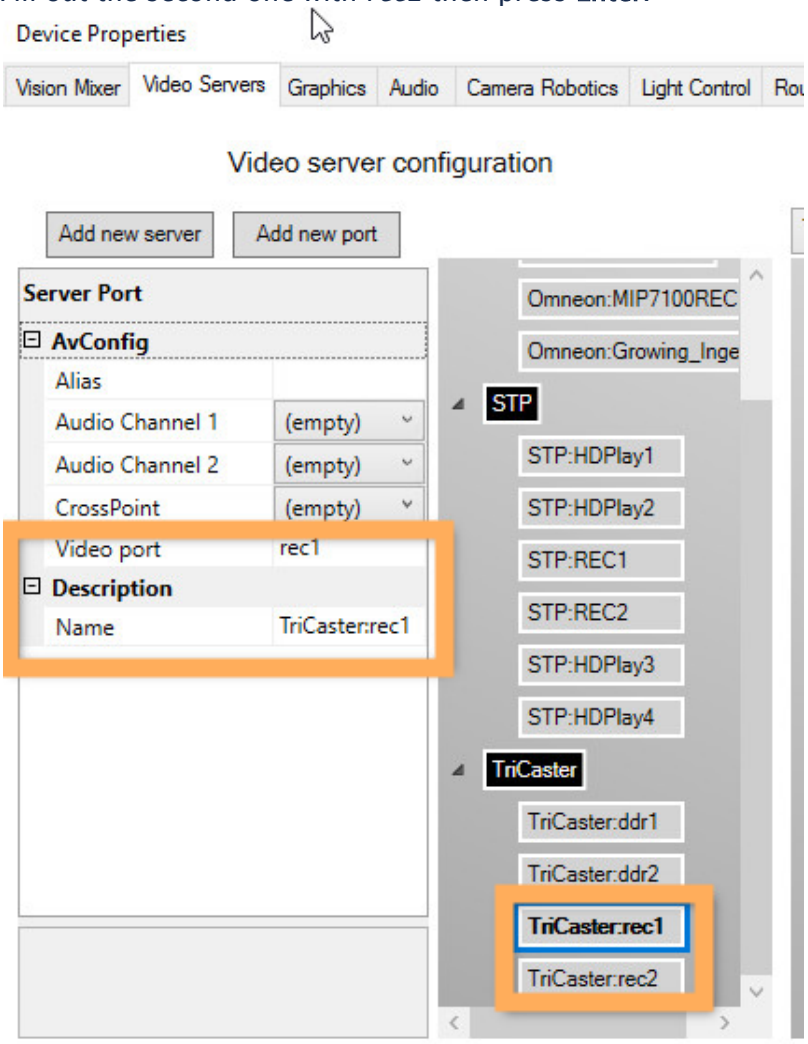
10. Click on the first **Video server configuration** (physical) port on the left half and drag-and-drop it to the first virtual one on the right under Mosart port configuration.



11. Repeat with the second port, by repeating steps (**9**) to (**12**).
12. On the *TriCaster* black box on the left half, add two more ports
    a. Fill out the first one with *rec1* then press **Enter**.

Copyright © 2024 Vizrt

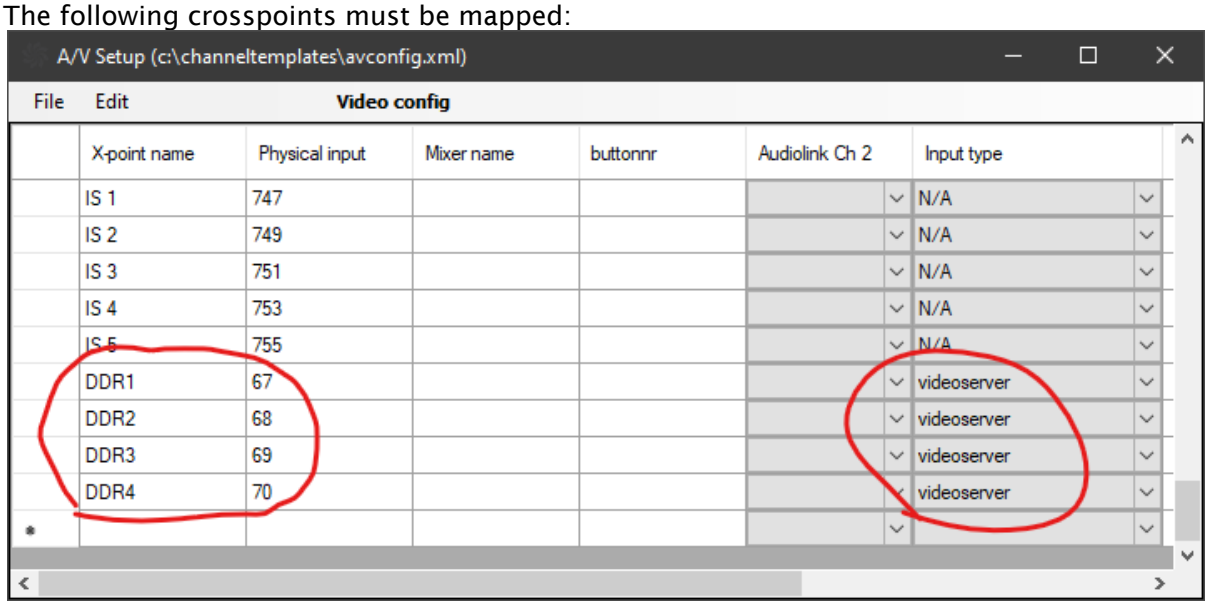
b. Fill out the second one with *rec2* then press **Enter**.



13. Similarly, fill out the second one with *rec2*.
14. Now set
    a. **Crosspoint**: *DDR1* for first port and *DDR2* for the second port.
    b. **Video port**: *ddr1* for first port and *ddr2* for the second port.
    c. **Name**: Any meaningful name.
15. This adds a virtual ripple group with 2 virtual ports.
16. Click **Apply**, then **Save & Close**.
17. Navigate to **AV Automation > Devices > AV Setup > Edit > Video config**.

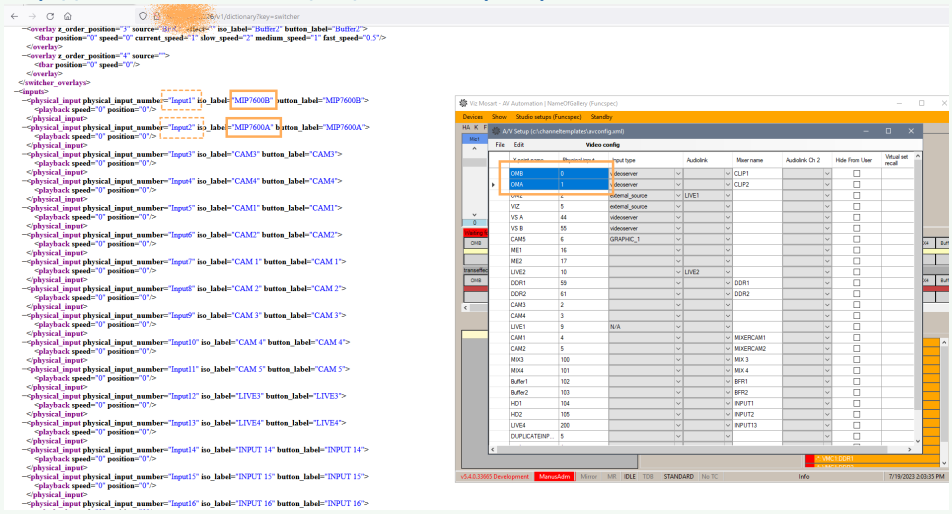
Copyright © 2024 Vizrt

18. Add the crosspoint mapping video server ports as configured on the TriCaster.
The following crosspoints must be mapped:



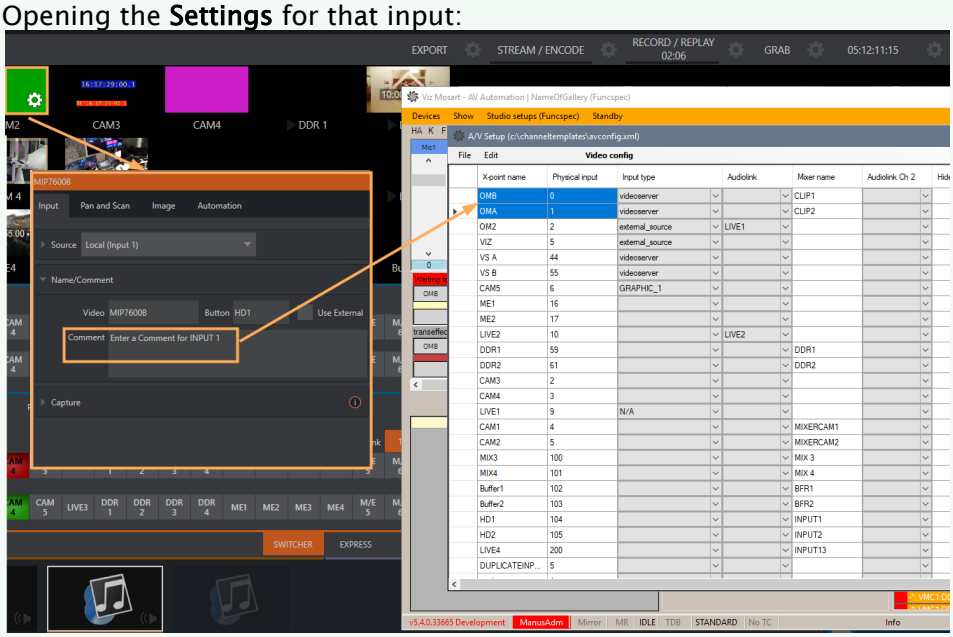
Tip:

- All (physical) inputs are found by accessing the TriCaster over its REST API:
  - http://<TriCaster_host>/v1/dictionary?key=switcher



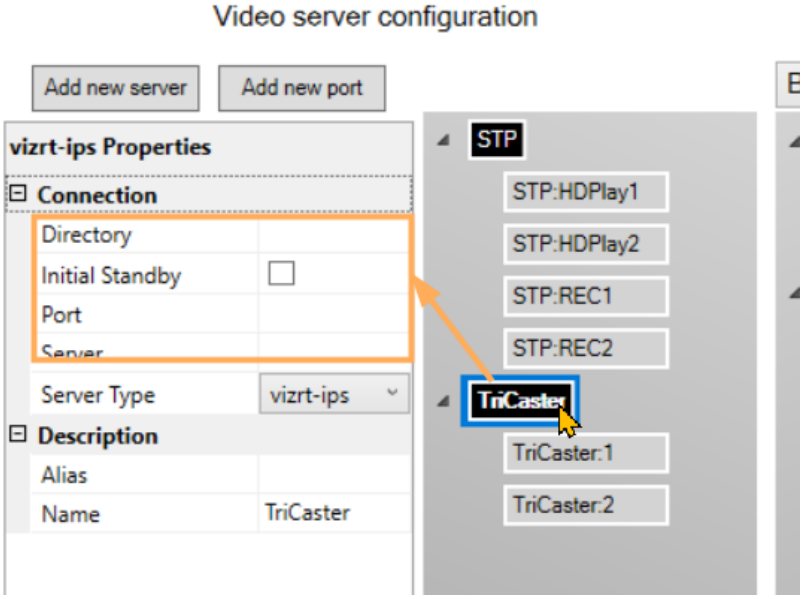
  - http://<TriCaster_host>/v1/dictionary?key=tally
- Alternatively, you can access the Physical number by:

i. Opening the **Settings** for that input:



ii. Check the number attached to the *Input#n* text in the **Comment**.
iii. Set the value in Mosart to *n-1*.

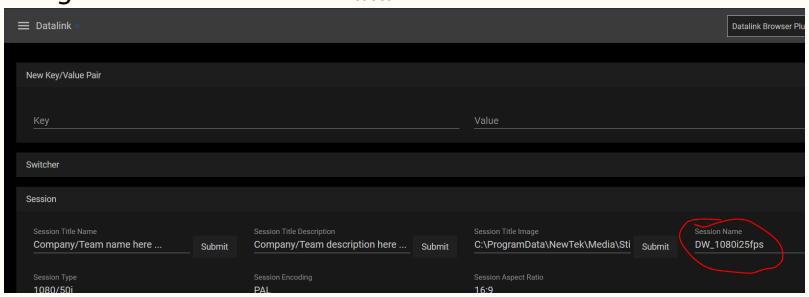
19. Click on the black box (see below) and configure the following properties for the server:



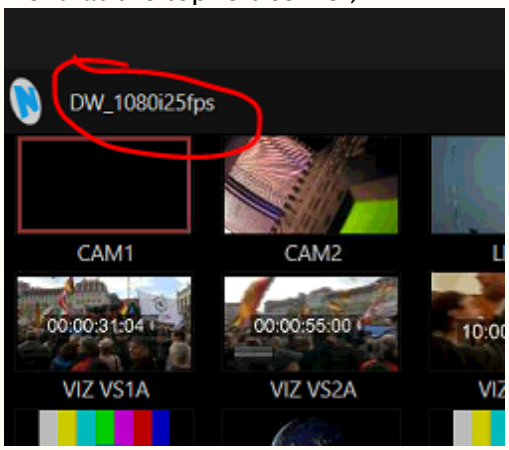
a. **Server**: http://<TriCaster_host> as IP number.
b. **Directory**: {volume}:/Sessions/{session name}/Clips/Import

> ⚠ **Note**: To find *session name*:

- Navigate to the TriCaster **DataLink** menu:



- The session name is also displayed on the TriCaster home page (hidden menu at the top left corner):



c. **Extension**: let it empty (when Media Service is used, no extension is needed)

> ⚠ **Note**: The clips id can be given from NRCS without any extension.

d. **Name**: Any meaningful name.

> ⚠ Make sure **Name** in the connection string is the same as the one configured in Media Administrator (see Video Servers below).

20. Click **Save & Close**.
21. Restart AV Automation if needed.

4.4.3 Graphics

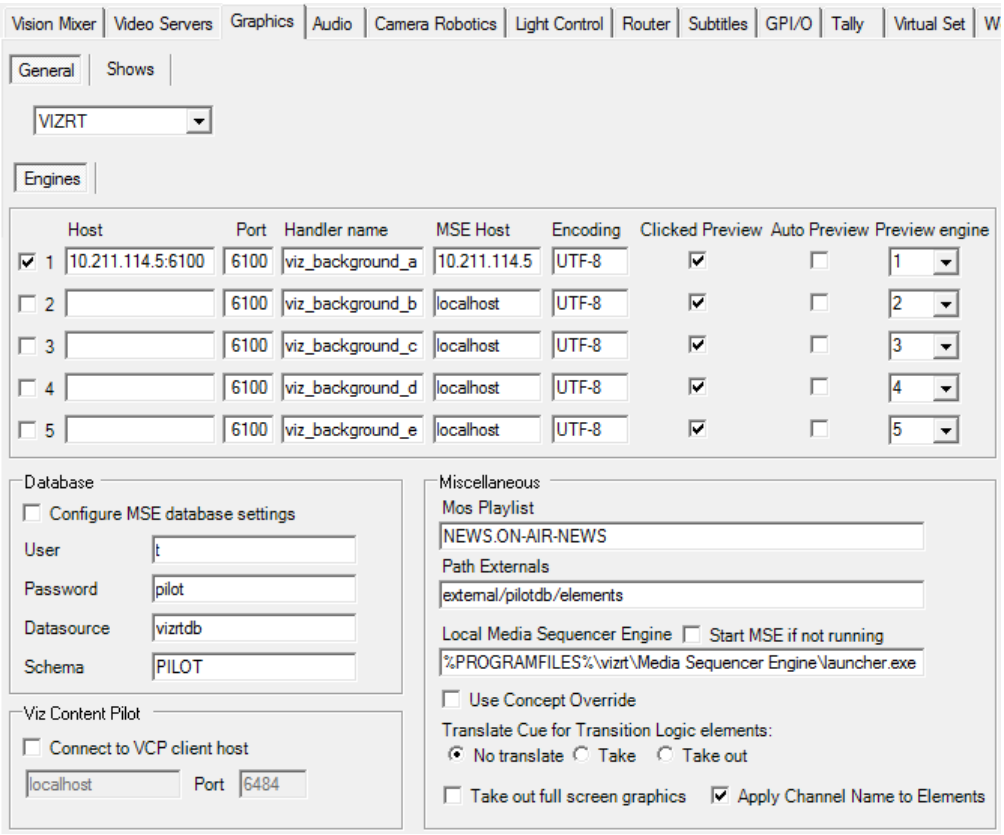
Viz Mosart is capable of connecting to a variety of graphic engines available from many different manufacturers.

AV Automation handles full frame graphics playout, from a primary template type in the rundown.

The properties available depend on the system to be used:

- Vizrt Settings
  - General Tab
  - Show Tab
- Deko Settings
- XPression Settings
- Orad Settings
- Pixel Power Settings
- Chyron Settings

Vizrt Settings



Copyright © 2024 Vizrt

General Tab

- **Engines:** Enables connection to each Viz Engine. Viz Mosart supports five graphics engine connections when checked.
  - **Host:** Hostname or IP address to Viz Engine.

    > ⚠ **Note:** A port other than the default (6100) should be added as an appendix to the host. For example: 127.0.0.1:**6101** and *not* to the **Port** field.

    Mirroring
    You can add multiple Vizrt Viz Engines to any of the five graphics connections.

    - Separate the engine name with a **comma** (**,**).
    For example: *engine1:port1, engine2:port2, engine3:port3*

> ⚠
> - All mirrored engines will share the same MSE.
> - You can add up to 9 engines.

  - **Port:** PDB only: Port to communicate with the Viz Engine. Default: `6100`.
  - **Handler name:** Media Sequencer only: Internal Media Sequencer handler name.
  - **MSE Host:** Media Sequencer only: Hostname or IP address to the machine running the Media Sequencer.

    > ⚠ **Note:** For Media Sequencer, *both* **Handler name** and **MSE Host** must be specified.

    > ⚠ **Creating a backup MSE**
    >
    > You can define a backup MSE here by adding a second IP after a ";"

  - **Encoding:** Font encoding for the Viz Engine. Default: UTF-8.
  - **Clicked Preview:** Enables the “click on full screen graphic element” to be taken on the preview engine.
  - **Auto Preview:** Enable to automatically take all full screen graphic elements in the preview engine.
  - **Preview engine:** The number of the Viz Engine that is selected to be the preview engine.
- **Database**
  - **Configure MSE database settings:** Enable to update the database settings in the Media Sequencer with the details below.
  - **User:** Username on the Oracle database. Default: pilot.
  - **Password:** Password on the Oracle database. Default: pilot.
  - **Datasource:** TNS name or connection string of the Oracle database. Default: vizrtdb.
  - **Schema:** Oracle database schema for the Viz Pilot connection. Default: PILOT.
- **Viz Content Pilot**

Copyright © 2024 Vizrt

- **Connect to VCP client host** (check-box):
  When checked, enables playout of Viz Pilot elements directly through a Viz Pilot client. Use of Viz Pilot must be assigned for each template, and the special Viz Pilot macros `mosart_load`, `mosart_start` and `mosart_continue` must exist in Viz Pilot.

  > **Note:** The "Display name" of both configurations must be exactly the same

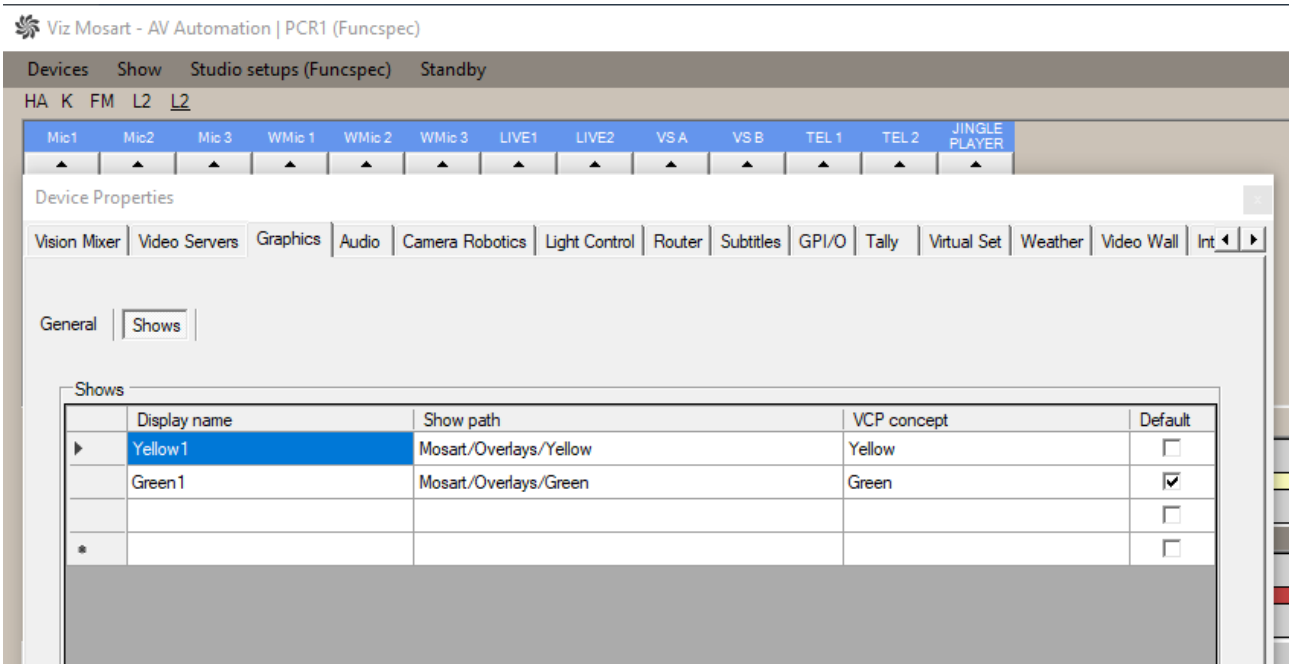
  - **Host:** Hostname or IP address to the Viz Pilot client.
  - **Port:** Port to connect to the Viz Pilot macro port.
- **Miscellaneous**
  - **Mos Playlist:** Playlist in VCP that is populated through the Vizrt MOS gateway. Default: NEWS.ON-AIR-NEWS

    > **Info:** The playlist name used here cannot be the same as that used in Trio Interface > Engines Setup Tab > Playlist > Playlist name.

  - **Path Externals:** Internal Media Sequencer path to the location for handling database elements. Default: external/pilotdb/elements
  - **Local Media Sequencer Engine:** Path to the launcher.exe in the Media Sequencer program files folder (only if Media Sequencer is running as a console application on the same machine).
  - **Start MSE if not running:** If checked and the Media Sequencer is not running, then AV Automation will try to execute it from the location given below.
  - **Use Concept Override:** If the full screen graphics concept should change when a new Graphics Profile is selected, this is configured here. Check to enable the Vizrt Concept Override if your scenes are prepared for this. Requires Viz Pilot 5.2 or later.
  - **Translate Cue for Transition Logic elements:** Only applicable when using Transition Logic full screen graphics as the Media Sequnecer ignores a normal cue command for these items.
    - No translate: Send cue command to the Media Sequencer (on Media Sequencer versions below 1.20 no cue will be performed).
    - Take: Use a take command when cuing.
    - Take out: Use a take out command when cuing.
  - **Take out full screen graphics:** Check to take out full screen graphics. By default, Viz Mosart will not do a takeout for full screen graphics.
  - **Apply Channel Name to Elements:** This will send the Channel name to be displayed with the graphic elements in the GUI.

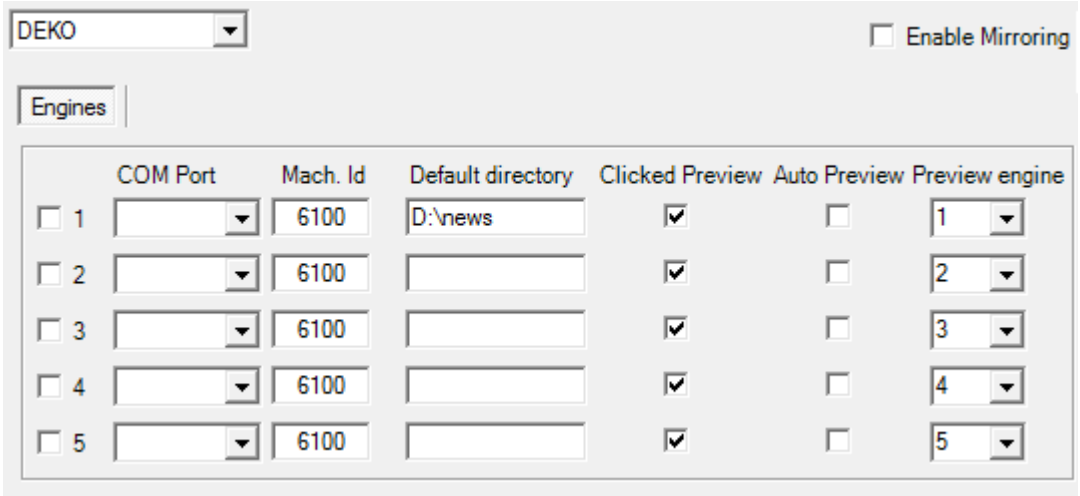
Copyright © 2024 Vizrt

Show Tab



⚠ It is mandatory to have an exclusive (non-null) value for **Show path** configured!

Deko Settings

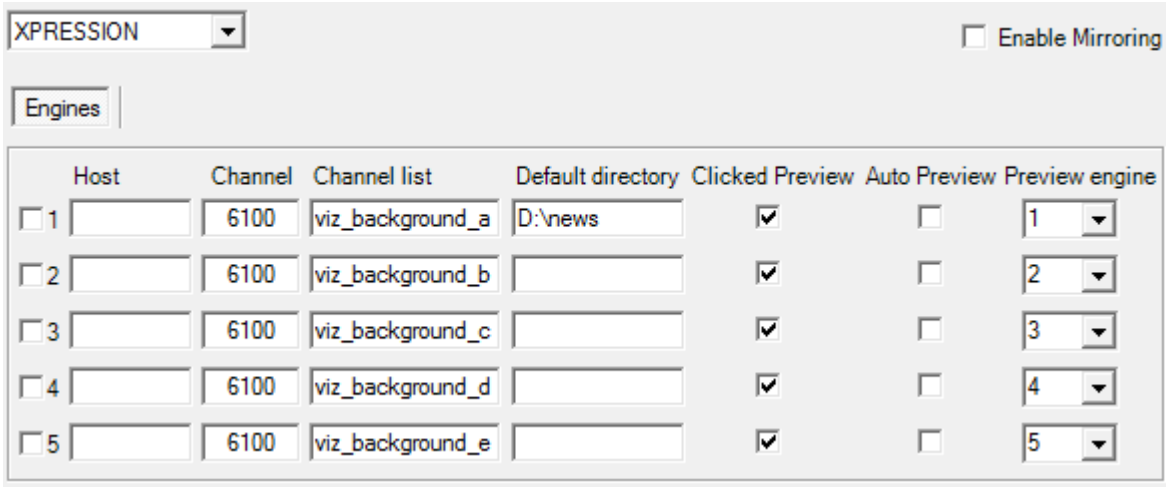


- **Engines:** Enables the connections to the Deko engines. Viz Mosart supports five graphics engine connections when checked.
- **COM port:** Serial ports connected to the Deko engines.
- **Mach. Id:** Machine IDs of the Deko engines. Valid range is `0` to `9`. Leave blank to send global commands.

Copyright © 2024 Vizrt

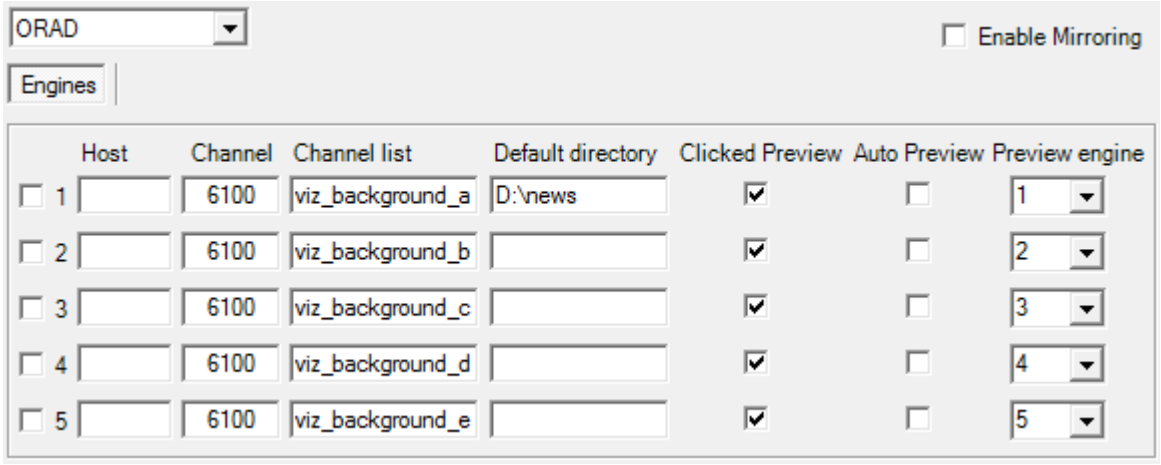
- **Default Directory:** The folder on the Deko engine that contains the graphics templates.
- **Clicked Preview:** Click on full-screen graphic element to send to the Preview engine.
- **Auto Preview:** Automatically send full-screen graphic element to the Preview engine.
- **Preview engine:** Defines the Deko engine number.

XPression Settings



- **Engines:** Check to enable the connections to the XPression engines. Viz Mosart supports five graphics engine connections.
- **Host:** Hostname or IP address to the XPression engine.
- **Channel:** Output channel from the XPression engine. Use `0` for default channel.
- **Channel list:** Output channel(s) to be cleared when Channel is set to 0.
- **Default directory:** The folder on the XPression engine that contains the graphics templates.
- **Clicked Preview:** Click on full-screen graphic element to send to the Preview engine.
- **Auto Preview:** Automatically send full-screen graphic element to the Preview engine.
- **Preview engine:** Defines the XPression engine number.

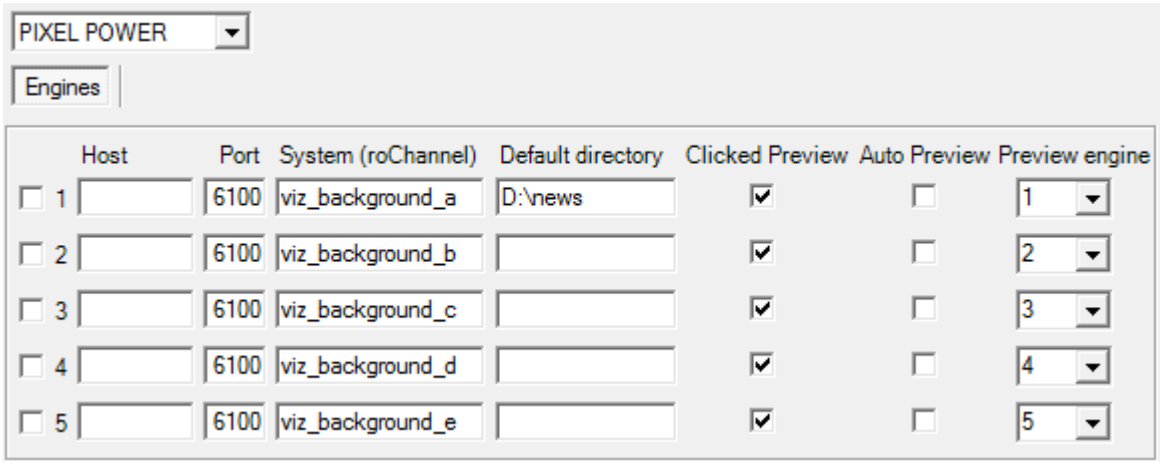
Orad Settings



Copyright © 2024 Vizrt

- **Engines:** Check to enable the connections to the Orad engines. Viz Mosart supports five graphics engine connections.
- **Host:** Hostname or IP address to the Orad engine. Include the IP port number, for example 172.20.51.55:10001.
- **Channel:** Output channel from the Orad engine. Use `0` for default channel.
- **Channel list:** Output channel(s) to be cleared when Channel is set to 0.
- **Default directory:** The folder on the Orad engine that contains the graphics templates.
- **Clicked Preview:** Click on full-screen graphic element to send to the Preview engine.
- **Auto Preview:** Automatically send full-screen graphic element to the Preview engine.
- **Preview engine:** Defines the Orad engine number.

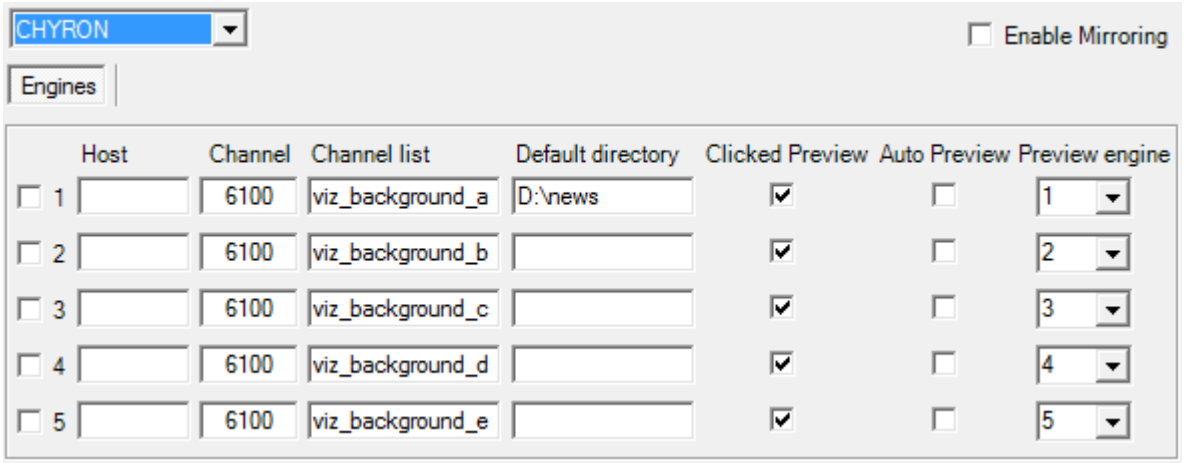
Pixel Power Settings



- **Engines:** Check to enable the connections to the Pixel Power engines. Viz Mosart supports five graphics engine connections.
- **Host:** Hostname or IP address to the Pixel Power engine.
- **Port:** Port to communicate with the Pixel Power engine.
- **Default directory:** The folder on the Pixel Power engine that contains the graphics templates.
- **Clicked Preview:** Click on full-screen graphic element to send to the Preview engine.
- **Auto Preview:** Automatically send full-screen graphic element to the Preview engine.
- **Preview engine:** Defines the Pixel Power engine number.

Copyright © 2024 Vizrt

Chyron Settings



- **Engines:** Check to enable the connections to the Chyron engines. Viz Mosart supports five graphics engine connections.
- **Host:** Hostname or IP address to the Chyron engine.
- **Channel:** Output channel from the Chyron engine.
- **Channel list:** Output channel(s) to be cleared when Channel is set to 0.
- **Default directory:** The folder on the Chyron engine that contains the graphics templates.
- **Clicked Preview:** Click on full-screen graphic element to send to the Preview engine.
- **Auto Preview:** Automatically send full-screen graphic element to the Preview engine.
- **Preview engine:** Defines the Chyron engine number.

Copyright © 2024 Vizrt

4.4.4 Audio

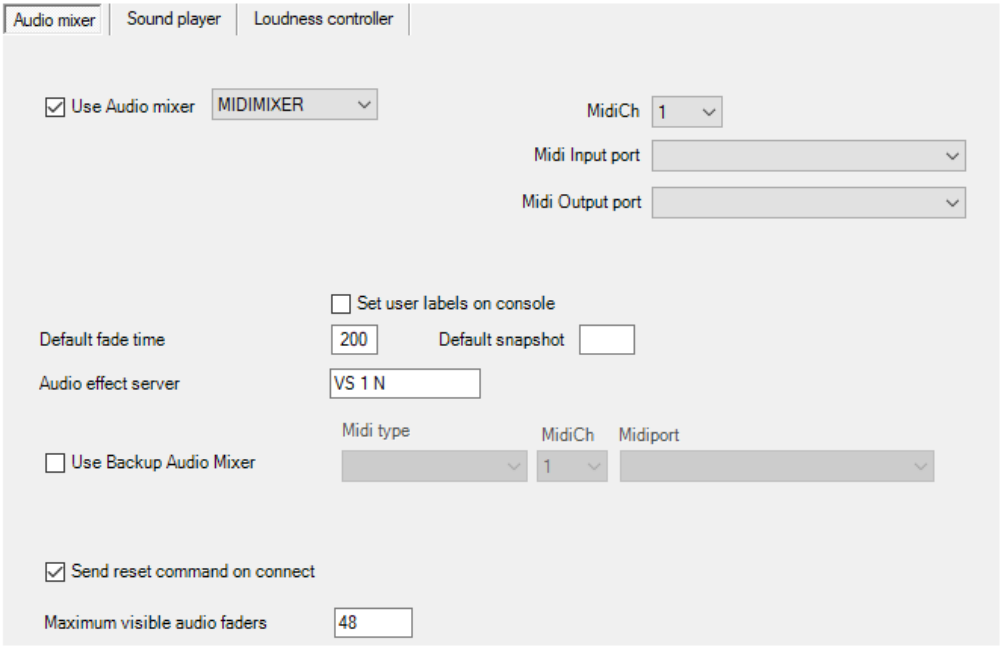
Viz Mosart connects to a variety of audio devices, from various manufacturers.

Select the required audio device from **AV Automation > Devices** > **Preferences** > **Audio**.

- Audio Mixer
- Audio Player
- Loudness Control

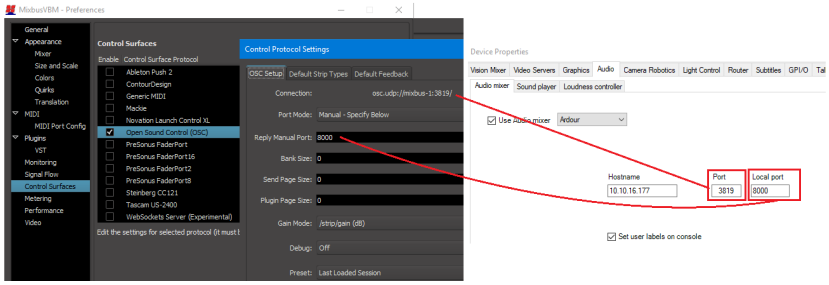
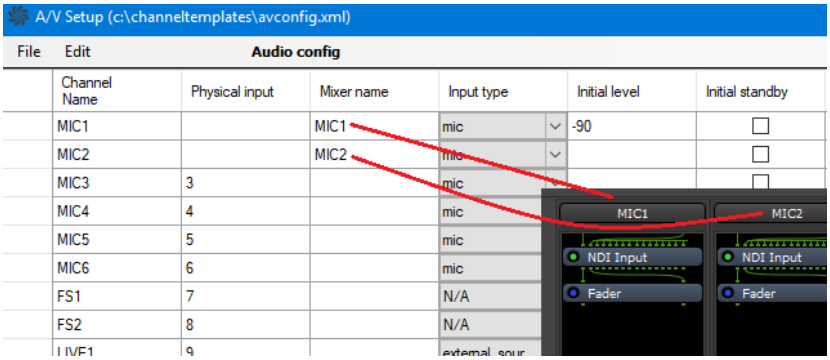
Audio Mixer

This tab enables setting-up of the main mixer and optionally, a backup.



- **Use Audio mixer:** Enables the audio mixer functionality.

Audio Mixer Types

Audio mixer	Protocol	Supported hardware / Comments	Interface
Ardour / Harrison Mixbus VBM	OSC	Harrison Mixbus VBM · Ensure *OSC* is enabled in the **Control Surface Protocol** selector and that in the **OSC Setup** tab, Viz Mosart Audio mixer properties match:  In the example above, where the port number (here *3819*) match correctly. · The input field **Port Mode** must be: *Manual - Specify Below*. · The input field **Reply Manual Port** can be any value, including empty. · The **Mixer name** must match the input name, **Physical Input** is blank:  ⚠ **Note**: Harrison MixBus 32C is *not* supported.	UDP
CALREC	RAP		RS-422, TCP/IP (Artemis)

Audio mixer	Protocol	Supported hardware / Comments	Interface
DHD	DHD RM4200-D v2.02		TCP/IP
EUPHONIX	RAP		COM
LAWO RMNOPL	Remote MNOPL	Lawo mc²	TCP/IP
LAWO ZIRKON	RAS, Lawo Zirkon subset		RS-422 or TCP/IP
LAWO EMBER+	Ember+	Lawo audio mixers with A__UHD Core. This requires that the Lawo audio mixer has correct firmware and settings to allow Ember+ control (contact Lawo for details about which specific audio mixer models support this functionality).	TCP/IP
MIDIMIXER	Sanford. Multimedia		MIDI
SSL	Light Broadcast Automation protocol, SSL Broadcast Automation Control Specification – v1.0		RS-422 or TCP/IP

Copyright © 2024 Vizrt

Audio mixer	Protocol	Supported hardware / Comments	Interface
STAGETEC	Stagetec Diamond RAS protocol, Monitora RAS		COM
STUDER_3000	Monitora + Ember, Ember 1.0	Studer OnAir 3000	RS-232 or TCP/IP
STUDER_VISTA	Ember	Studer Vista	TCP/IP
VIZENGINE	n/a	Viz Opus only	TCP/IP
WHEATSTONE	Wheatstone Mixer Automation Protocol v.1.4		TCP/IP
YAMAHA_O2	MIDI	Yamaha 01V96, DM1000, DM2000, DME64N, O2V96, LS9, CL5	MIDI
vizrt-ips	NewTek	Vizrt Group/NewTek IPseries mixers. TriCaster, VMC, Viz Verdi, Viz Vectar	TCP/IP

- **Serial (Calrec, Lawo, Studer, etc.):** Serial port connected to the audio mixer, when **Serial** radio button is selected.
- **TCP/IP:** Hostname or IP address and port to communicate with the audio mixer, when **TCP/IP** radio button is selected. Note that for *Calrec* it is possible to configure a main and backup audio mixer by specifying two comma separated hostname values and port values.
- **Hostname:** Hostname or IP address of the audio mixer.
- **Port:** Port on the audio mixer.
- **Local port:** (Ardour only) The mixer will send messages to this port on the Mosart server. Use 8000 unless there is a good reason for choosing another port.

Copyright © 2024 Vizrt

- **Backup hostname and port (LAWO RMNOPL):** Hostname (or IP address) and port to the backup audio mixer console.
- **Set user labels on console:** When selected, user labels on the (non-EMBER+) Lawo or Vista console are overwritten.
- **MidiCh (MIDIMIXER and YAMAHA_O2):** MIDI channel assigned to the audio mixer. Depending on mixer model, this setting may have to correspond to settings on the mixer.
- **Midi Input port (MIDIMIXER and YAMAHA_O2):** System MIDI input port to communicate with the audio mixer.
- **Midi Output port (MIDIMIXER and YAMAHA_O2):** System MIDI output port to communicate with the audio mixer.
- **Default fade time:** Time in frames for fading open faders with the manual fade function (**CTRL+F**).
- **Default snapshot:** Snapshot to recall when starting the automation (MIDI parameter change).
- **Audio effect server:** Name for the Audio effect fader. This fader will open if effect transitions are used.
- **Send reset command on connect:** Used for STUDER3000 SW 2.2, or lower. Recalls a default setup for the Mixer panel.
- **Visible audio faders:** Maximum visible faders in AV Automation.

AV Automation Audio Mixer Availability Table

	Serial	TCP/IP	MidiCh	Midiport	Hostname	Port	Backup hostname	Backup Port	Faderinput offset	Useember protocol	Comport audio mixer
Ardour					X	X					
CALREC	x	x			x	x					x
DHD		x			x	x					
EUPHONIX	x										x
LAWO RMNOPL		x			x	x	x	x			
LAWO ZIRKON	x	x			x	x					x

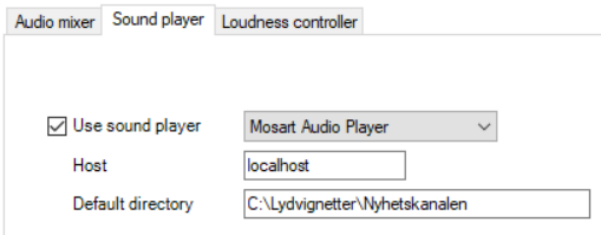
Copyright © 2024 Vizrt

	Serial	TCP/IP	Midi Ch	Midiport	Hostname	Port	Backup hostname	Backup Port	Faderinput offset	Useember protocol	Comport audio mixer
LAWO EMBER+					x	x					
MIDI MIXER			x	x							
SSL	x								x		x
STAGE TEC	x	x			x	x					x
STUDER 3000	x	x			x	x				x	x
STUDER VISTA		x			x	x					
VIZ ENGINE		x			x	x					
WHEATSTONE		x			x	x					
YAMAHA O2			x	x							
Vizrt ips		x			x	x					

Audio Player

A basic audio player is always included as standard in a Viz Mosart.

Copyright © 2024 Vizrt



Requirements

Refer to general section Prerequisites (section Viz Mosart Audio Player).

Setting Up an Audio Player

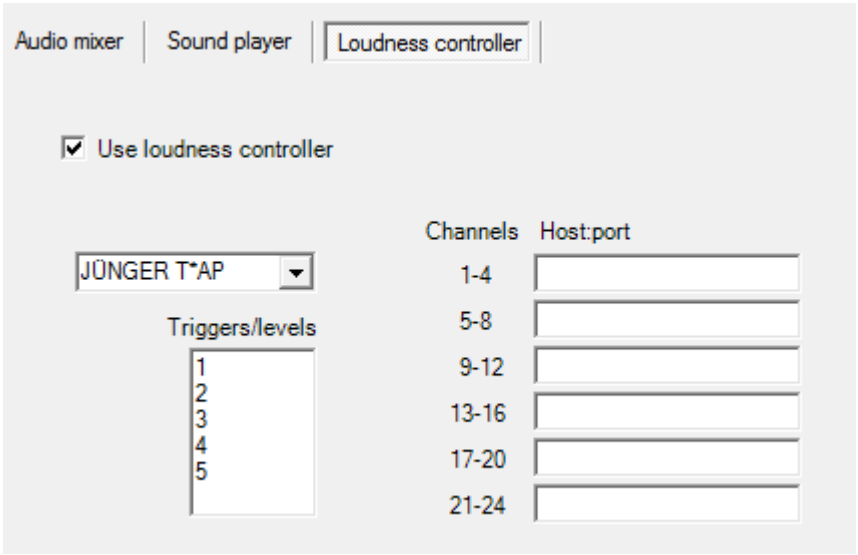
Refer to section Audio Players.

Operating an Audio Player

Refer to section *Playing Audio* in the Viz Mosart User Guide.

Loudness Control

A loudness control from Junger.



Copyright © 2024 Vizrt

- **Use loudness controller:** Check the box to enable loudness control.
- **Loudness controller type:** JUNGER T*AP.
- **Triggers/levels:** Name of triggers or loudness levels, one for each line. These will appear in a tool-tip menu in the Template editor, and must correspond to the configuration in the loudness controller(s).
- **Channels:** Fixed values. For Junger T*AP there are four stereo channels per host. Connections to the audio faders are configured in A/V Setup/Edit/Audio config/Loudness channel.
- **Host:port:** Hostname or IP address and optional port number for each controller.

> ⚠ **Note:** If the root node name is different than **T*AP** the name must be set in the AVA configuration property *JungerRoot* .

See Also

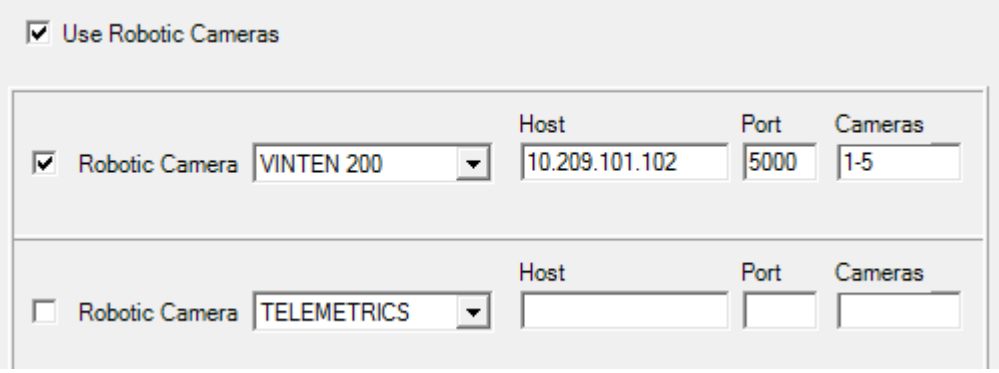
- Audio Player
- Stagetec Driver Configuration
- Audio Mixer Configuration Files

Copyright © 2024 Vizrt

4.4.5 Camera Robotics

Configuring a Robotic Camera

From **AV Automation > Devices > Properties > Camera Robotics tab**, you can configure up to 4 robotic camera controllers.



- **Use Robotic Cameras:** Check to enable the robotic camera control functionality.
- **Robotic Camera:** CAMBOTICS, CAMEROBOT, CINNEO, ELECTRIC FRIENDS, FX-MOTION, PANASONIC, PANASONICNEW, PANASONIC TCP, SHOTOKU TR-T, SHOTOKU TR-T IP, TECHNODOLLY, TELEMETRICS, VINTEN 200, .
- **Host:** Hostname or IP address of the *RCC* (Robotic Camera Controller) server.
- **Hosts (FX-Motion and Technodolly):** A comma separated list of names or IP addresses for the camera robots starting from Camera 1. Missing robots are indicated by extra commas.
- **Port:** Port to communicate with the RCC server.

  > ⚠ **Note:** When using PANASONIC TCP, you may or may not specify a port here. If no port is specified, the default port 80 is used.

- **Com port:** For serial communication with the RCC server.

  > ⚠ **Note:** When using PANASONIC or PANASONICNEW you *must* select the Com port to use. Only one controller can be connected to each Com port.

- **Cameras:** List of robotic cameras for individual stand-by control and for matching against the Camera Number in Robotic Camera Control in Template Device Functions. List members are separated by commas, or by hyphens indicating ranges.

  > ⚠ **Note:** When using PANASONIC, PANASONICNEW, or PANASONIC TCP, the connection goes through a controller. The *Cameras* value is used to set the controller, so only enter one number here.

Copyright © 2024 Vizrt

Camerobot Camera Systems

Viz Mosart can control anchor dependent shots for the Camerobot camera system. For details, please refer to Control Commands in Templates, section *Command Values and Parameters,* DEVICE_PROPERTY, using the value *CAMERA CONTROL*.

.

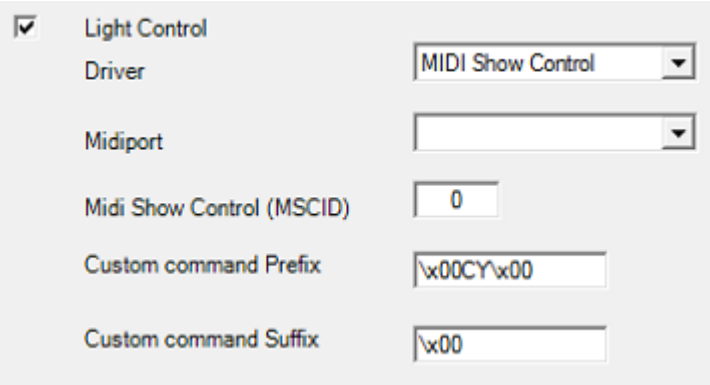
Robotic Camera Configuration Files

For both general and device-specific settings, see section Robotic Camera Configuration Files.

Copyright © 2024 Vizrt

4.4.6 Light Control

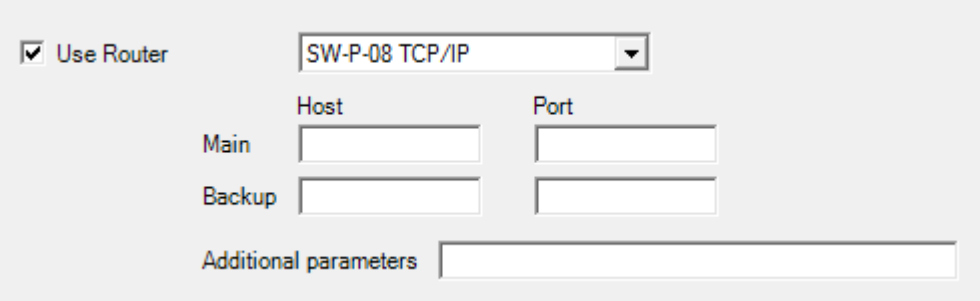
MIDI Show Control



- **Light Control:** Enables the Light control through MIDI Show Control.
- **Midiport:** System MIDI port to communicate with the Light board.
- **Midi Show Control (MSCID):** The MIDI Show Control ID that is used for communicating with the Light board. Default: `0`.
- **Custom command Prefix:** This is the prefix test sent with the MIDI Show Control command. Characters can be escaped using the x00 where 00 is replaced by the numeric value of the character. Default: `\x00CY\x00`.
- **Custom command Suffix:** This is the suffix sent with the MIDI Show Control command. Characters can be escaped using the x00 where 00 is replaced by the numeric value of the character. Default: `\x00`.

Copyright © 2024 Vizrt

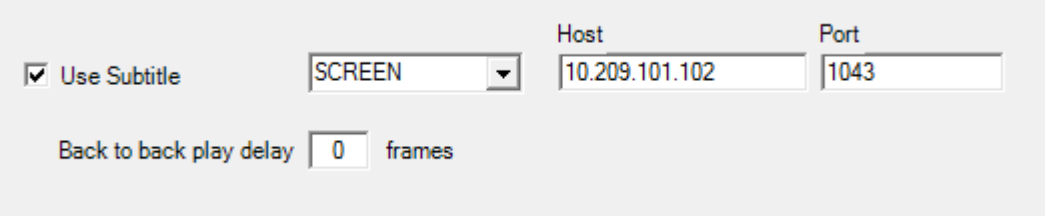
4.4.7 Router



- **Use Router:** Check to enable router control and select protocol from the list.
- **Router type:** EVERTZ QUARTZ, GVG G-SMS7000, SW-P-08 TCP/IP, SW-P-08 SERIAL, VENUS, VIKINX SERIAL or VIKINX TCP/IP.
- **Port:** Serial port (GVG G-SMS7000, Venus, SW-P-08 serial, and Vikinx serial).
- **Bit rate:** Selectable serial bit rate (GVG G-SMS7000).
- **Parity:** Parity bit, default None (GVG G-SMS7000).
- **Main Host:** Hostname or IP address of the main router (EVERTZ QUARTZ, SW-P-08 TCP/IP and Vikinx TCP/IP).
- **Main Port:** IP port of the main router (EVERTZ QUARTZ, SW-P-08 TCP/IP and Vikinx TCP/IP).
- **Backup Host:** Hostname or IP address of the backup router (SW-P-08 TCP/IP).
- **Backup Port:** IP port of the backup router (SW-P-08 TCP/IP).
- **Additional parameters:** For future use.

Copyright © 2024 Vizrt

4.4.8 Subtitles



- **Use Subtitle:** Check to enable Screen Polystream subtitling.
- **Subtitling type:** SCREEN, SVT
- **Host:** Hostname or IP address of the subtitling system.
- **Port:** Defines the IP communication port.
- **Back to back play delay:** Delay in frames on the play command to the subtitling system when playing subtitle files back to back.

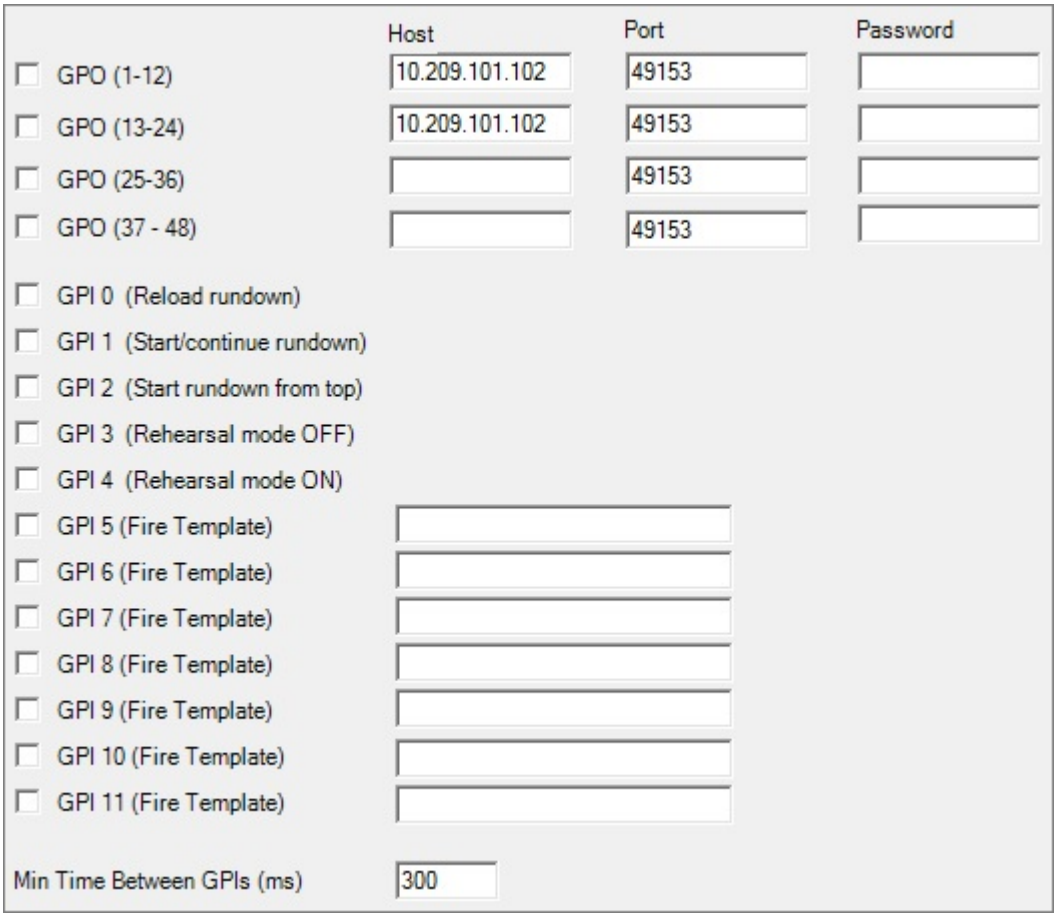
Copyright © 2024 Vizrt

4.4.9 GPIO

GPI functionality supports triggering any template from the currently active template set and includes a selection of optional control commands.

- GPO
- GPI 0-4
- GPI 5-11
  - Triggering a Template
  - GPI Control Commands
- Troubleshooting
  - Verifying the Connection
  - Configuration Issues

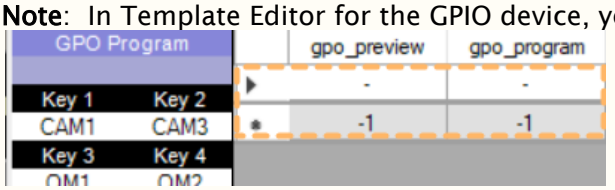
Adding GPI/O control to a device is performed in Viz Mosart AV Automation (**Devices > Properties > GPI/O tab**).



Copyright © 2024 Vizrt

GPO

Note: In Template Editor for the GPIO device, you enter:



- 0 for GPO 1
- 11 for GPO 12 etc.,

in the same way as GPI.

- **GPO (0-11):** Enables a WebIO box for the first 12 GPO when checked. Note that this has to be configured for GPI 0-11 to be enabled.
- **GPO (12-23):** Enables a WebIO box for the second 12 GPO when checked.
- **GPO (24-35):** Enables a WebIO box for the third 12 GPO when checked.
- **GPO (36-47):** Enables a WebIO box for the fourth 12 GPO when checked.
- **Host:** Hostname or IP address to the WebIO GPI/O box.
- **Port**: Port on the WebIO GPI/O box.
  If configured as *80*, Mosart will use the HTTP WebIO protocol, while for other Port values Mosart will use the Binary WebIO protocol.
- **Password:** Defines the password, if needed.

GPI 0-4

The first 5 GPI signals are connected to fixed Viz Mosart commands for backward compatibility.

- **GPI 0 (check box):** When the external pulse is received, reload rundown is triggered: The timeline is stopped and the current rundown is reloaded from the NRCS cache.
- **GPI 1 (check box):** Start/continue rundown. Will trigger the TAKE-NEXT event and with that start the timeline or advance to the next timeline element.
- **GPI 2 (check box):** Start rundown from top. Will start the rundown from the first story..
- **GPI 3 (check box):** Disable rehearsal mode.
- **GPI 4 (check box):** Enable rehearsal mode.
- **GPI 5-11 (check boxes/custom):** Enables firing of the template or command given in the text box.
- **Min Time Between GPIs (ms)**: Threshold time before a new trigger command will be processed.

GPI 5-11

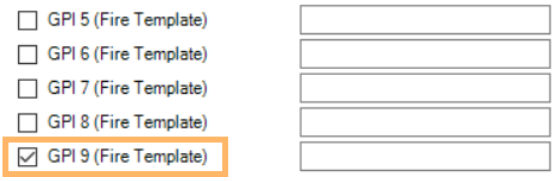
Each of the GPI signals 5-11 may be assigned to either trigger a Viz Mosart control command, trigger an arbitrary template from the active template set or trigger a direct take, as follows.

Note: A GPI/GPO device must first be configured. The configuration of GPI/GPO devices is done in AV Automation Devices - GPIO.

Copyright © 2024 Vizrt

Triggering a Template

Taking a template using one of GPIs 5 to 11 (Fire Template) adds the corresponding template to the timeline.



GPI Control Commands

As template to the timeline (see GPIO above) is not always wanted behavior, you can instead insert a *control command*.

To Trigger Mosart Control Commands

- To trigger a Mosart Control Command enter a valid command in the text field associated with the GPI signal.
  Supported commands are:
  - **DIRECTTAKE**: Trigger the numbered direct take.
  - **CLEARFOREGRUONDGRAPHICS**: Removes all CGs
  - **SKIPNEXTSTORY**: Skips next story **item**
  - **UNSKIPNEXTSTORY**: Unskips next story **item**
  - **TAKELASTGRAPHICSOUT**: Removes last CG
  - **TAKEMENUALGRAPHICSOUT**: Removes manual CGs
  - **PLAYCUEDVIDEOSERVER**: Plays out the last cued clip on a video server
  - **PAUSEPLAYINGVIDEOSERVER**: Pauses the currently playing clip on a video server
  - **STOPPLAYINGVIDEOSERVER**: Stops the currently playing clip on a video server
  - **RECUECUEDVIDEOSERVER**: Recues the last cued clip on a video server

Direct Takes

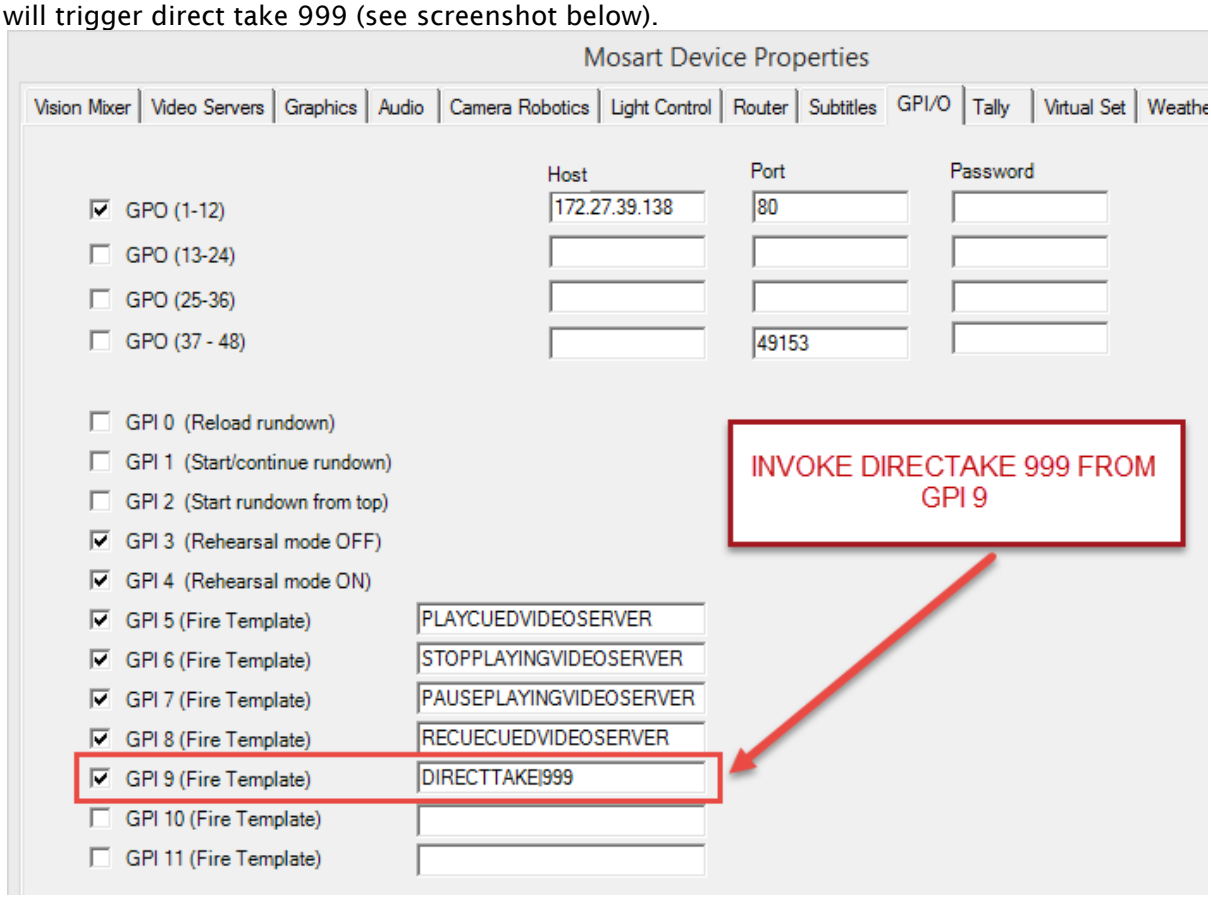
A direct Take is an example of an available control command.

Syntax

DIRECTTAKE|<the Recall number of the direct take>
Example: DIRECTTAKE|999

Copyright © 2024 Vizrt

will trigger direct take 999 (see screenshot below).



To Trigger A Mosart Template

This can be used to trigger a Mosart template from the active template set:

Syntax: `[TemplateType]|[Variant]`

- Example: Camera|1

Troubleshooting

Verifying the Connection

Verifying

1. Assign a direct take to a GPI between 5 and 11.
2. Restart AV Automation.
3. Trigger this GPI from the external device connected to this GPI (see GPIO below).
4. Verify on the Mosart server that the corresponding GPI is executed.
   Direct takes are shown in the AV Automation Log window (lower left) at Take.

Copyright © 2024 Vizrt

To Test GPI Signals Without Using GPI

There is an simple method to test the GPI configuration.

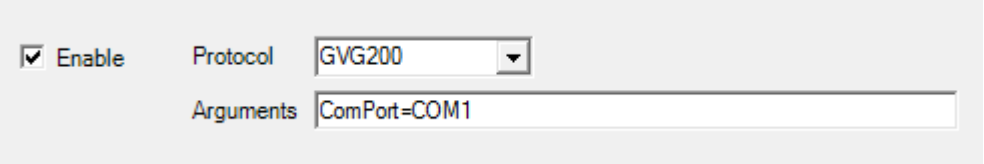
- If a GPI/GPO device is configured, as shown above, *GPI signals 0-9* can be simulated inside AV Automation using keyboard sequence **CTRL+SHIFT+<number>** to trigger a corresponding GPI, where **<number>** is the GPI number *+1* (modulo 10).
  For example, **CTRL+SHIFT+6** triggers GPI **5**.

Configuration Issues

If the GPIO will not import the configuration file directly, a workaround is to save the current configuration and using Notepad++ or similar, to compare the difference between the new configuration and the existing configuration.
Amend the configuration on the existing configuration to match the new settings.

Copyright © 2024 Vizrt

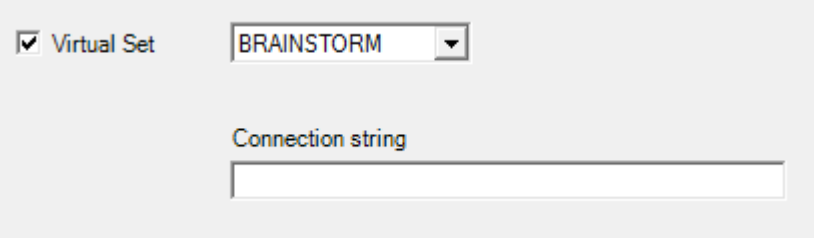
4.4.10 Tally



- **Enable:** Check to enable tally functionality.
- **Protocol:** GVG200 or SONY MVS
- **Arguments:** ComPort=COMx

Copyright © 2024 Vizrt

4.4.11 Virtual Set



- **Virtual Set:** Check to enable virtual set control.
- **Virtual Set:** BRAINSTORM, Viz Virtual Studio.
- **Connection string:** Server=hostname; [port=portnumber;] [mode=mixmode;].

Connection String Parameters for Viz Virtual Studio

- **Server**: IP address or network name of machine where Viz Tracking Hub is running. Required.
- **Port**: TCP/IP port Viz Tracking Hub accepts connections. Optional.
  Default value is 20000.
- **LocalIP**: IP address of local network interface. Optional. Default value is first Ethernet interface.
- **Router**: Router name configured in Viz Tracking Hub. Optional.
  Default value is first router in Viz Tracking Hub.

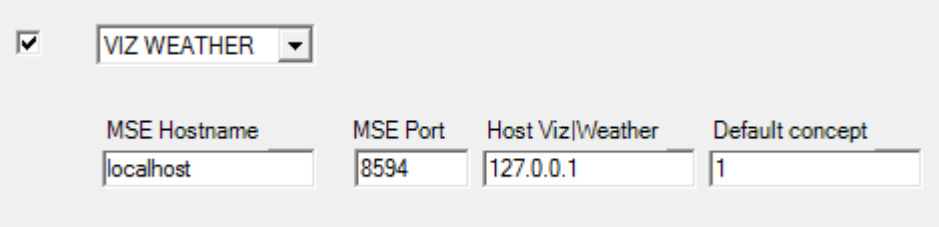
Sample Connection string: *Server=10.45.0.15;Port=20000*

Camera set index enumeration starts from 1. This means that if a router has two camera sets configured, their indices will be 1 and 2.

⚠ **Known Issue**: Driver cannot detect when hub server is going offline. For example, if you kill the server, the driver keeps sending messages to server without an error being raised.

Copyright © 2024 Vizrt

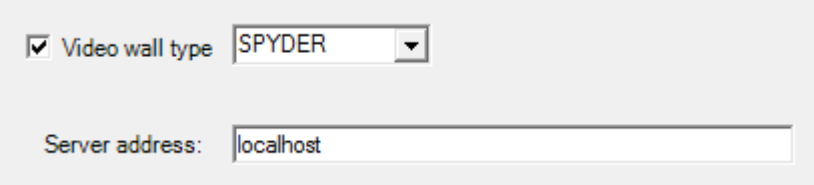
4.4.12 Weather



- **Use Weather:** Check to enable control of a weather control.
- **Weather controls:** VIZ WEATHER or WEATHER ONE
- **MSE Hostname (Viz Weather):** Hostname or IP address to the Media Sequencer controlling the Weather Control engine.
- **Hostname (Weather One):** Hostname or IP address of the Borealis Weather One engine.
- **MSE Port (Viz Weather):** Port to communicate with the Media Sequencer.
- **Port (Weather One):** Port to communicate with the Borealis Weather One engine.
- **Host Viz Weather (Viz Weather):** Hostname or IP address to the Viz Weather engine.
- **Default concept:** The weather concept to be used as default.

Copyright © 2024 Vizrt

4.4.13 Video Wall



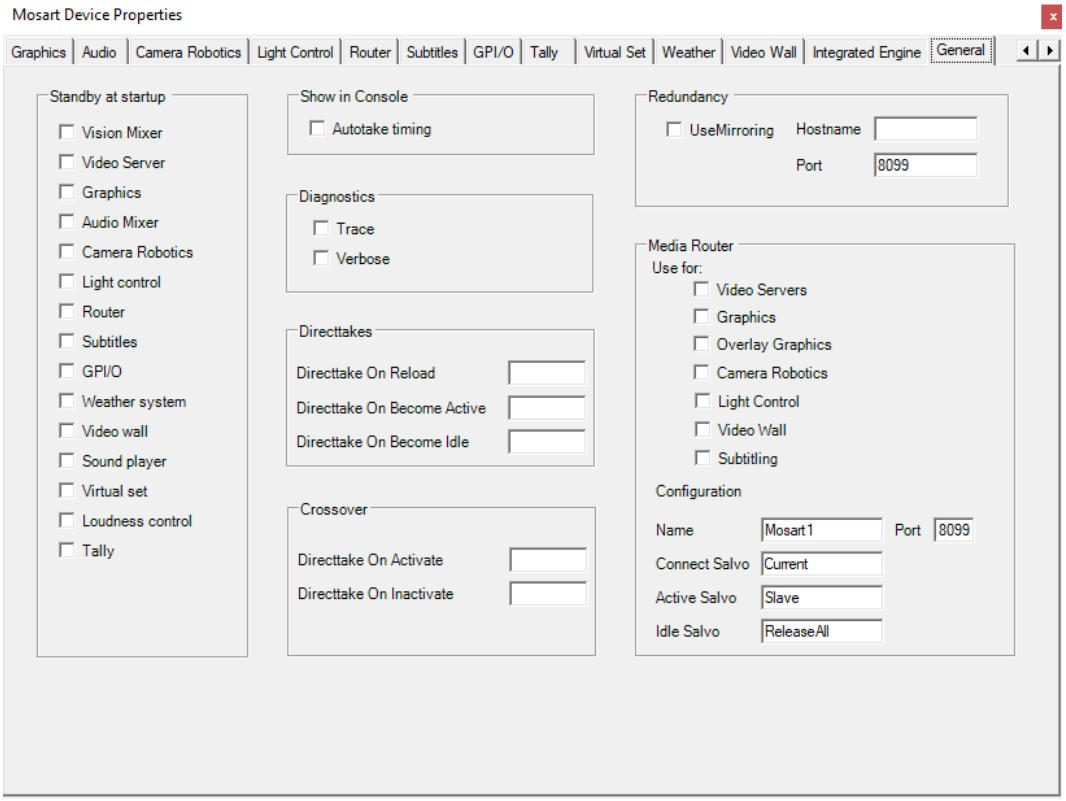
- **Video wall type:** Check to enable video wall control.
- **Video wall controls:** ENCORE, PANDORA, SPYDER, WATCHOUT (Version 5.2, DISPLAY CLUSTER PROTOCOL).
- **Port (Encore):** Serial port.
- **Bit rate (Encore):** Serial bit rate.
- **Server address (Pandora, Spyder, Watchout):** Hostname or IP address of the video wall server.
  For Pandora, the server is the PB Widget Designer.
  For WATCHOUT, full connection string. For details see Watchout Connection String.
- **Backup (Pandora):** Hostname or IP address of the backup PB Widget Designer.

See also

- Jupiter Systems PixelNet Video Wall Control

Copyright © 2024 Vizrt

4.4.14 General



- **Standby at startup:** Selector for drivers that will start in standby mode. These selections must then be manually reset from standby mode, either using the Viz Mosart GUI or from AV Automation.

  > ⚠ **Note:** Whenever a Viz Mosart GUI is operational or if a backup Viz Mosart server is operational, they will *override* any standby settings made here. Standby settings applied in this menu only take affect at complete restart of all Viz Mosart applications. See the *Standby* description under **Main Menu** of the *Viz Mosart User Guide* for further details..

- Show in Console
  - **Autotake timing:** Adds timing to the log window. Used for debugging purposes.
- **Diagnostics**
  - **Trace:** Adds logging information to console used for dbugview.
  - **Verbose:** Increases logging information from AV Automation to the log file.
- **Directtakes**
  - **Directtake On Reload:** The Recall Nr (see Template Properties) of the direct take template to be fired upon re-loading the rundown.
  - **Directtake On Become Active:** The Recall Nr of the direct take template to be fired when the server becomes active from idle, when switching between main and backup Viz Mosart servers.

Copyright © 2024 Vizrt

    - **Directtake On Become Idle:** The Recall Nr of the direct take template to be fired when the server becomes idle from active, when switching between main and backup Viz Mosart servers.
- **Crossover**
    - **Directtake on Activate:** The Recall Nr of the direct take template to be taken when this server goes active in crossover switch.
    - **Directtake on Inactivate:** The Recall Nr of the direct take template to be taken when this server goes inactive in crossover switch.
- **Redundancy**
    - **UseMirroring:** Check to enable mirroring. Upon enabling mirroring on both main and backup server, the settings are synchronized.
    - **Hostname:** Hostname or IP address of the Backup Server if using Main Server, and vice versa.
    - **Port:** IP port number of the Backup Server if using Main Server, and vice versa. **NB**: For redundancy to work, the same port number must be configured as **Media Router** Port. Recommended value: 8099.
- **Media Router**
    - **Use for:** Select which controls should be used by the Media Router. (Also see the MMR Tab for Overlay Graphics Interface configuration).
    - **Name:** Media router's hostname or IP address.
    - **Port:** Media router's connection port number. Must be the same as **Redundancy** Port. Recommended value: 8099.
    - **Connect Salvo:** The salvo name to be taken when this server connects after restart. Default: Current.
    - **Active Salvo:** The salvo name to be taken when this server becomes the active server from being idle. Default: Slave.
    - **Idle Salvo:** The salvo name to be taken when this server becomes idle after being active. Default: ReleaseAll.

If Redundancy is to be used, the **Media Router** Port has to also be set, and then to the same value as the **Redundancy** Port.

Copyright © 2024 Vizrt

4.4.15 Vizrt Graphics

For Vizrt graphics, the workflow in Viz Mosart is continually being improved.

- The Viz Mosart integration towards other Vizrt products is now handled by a single driver that communicates with Vizrt's Media Sequencer over a REST API exposed by Media Sequencer.
- In future versions, a expanding range of both existing and new Viz playout content will be supported.

To configure and use the new Vizrt Graphics driver

- Please refer to section Vizrt Graphics.

Copyright © 2024 Vizrt

4.5 Audio And Video Setup

4.5.1 Working with Audio and Video Setup

In the **Audio and Video Setup** menu, you can add the relevant audio and video channels and configure their attributes.

Viz Mosart relies on predefined *video crosspoints* entered into AV Automation at initial installation and setup of Viz Mosart.

- XML Editor Utility
- Video Configurations
- Audio Configurations
- Template-driven Effects
- Effect Number and Studio Setup
- Router Source Configurations
- Router Destination Configurations

XML Editor Utility

A simple XML editor utility (avconfig) is used to manage and configure the settings in AV Automation.

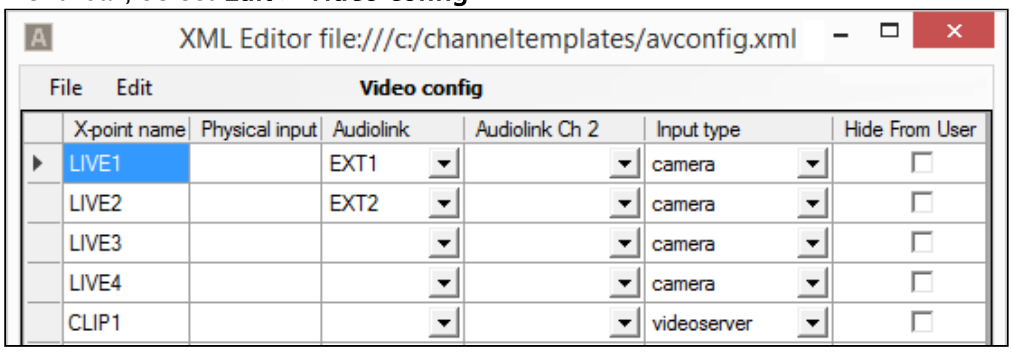
To open the avconfig utility

- From AV Automation, select **Devices > A/V Setup**.

This presents *all* configuration values. You can filter a specific set of configurations, for example *video* configurations, as explained below.

Video Configurations

- To work with just video crosspoint configurations, Audio and Video Setup and from the menu bar, select **Edit > Video config**.



- As a minimum, you should enter the crosspoint details: **X-point name**, **Physical input** and **Input type** (where applicable).

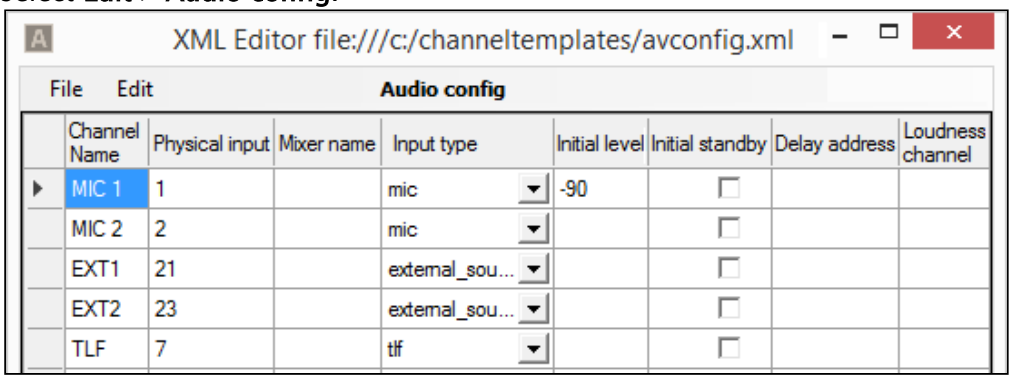
Settings

- **X-point name:** Name to use for the video crosspoint in the Viz Mosart system. This name will be displayed in the **Crosspoints** drop-down menu in all **Templates** and in the **ActiveX** application.
- **Physical input:** Input on the video switcher. These are the physical inputs of the video switcher.
- **Audiolink:** Audio fader to link to the video source when the video source is changed in the Newsroom Control System or Viz Mosart GUI, enabling the audio faders to follow the video crosspoints.
  The drop-down menu presents Audiolink-values corresponding to the audio sources that may be selected for a fader, alongside the selected input source, for example a camera.
- **Audiolink Ch 2:** Second audio fader link for the video source. The fader name that will be used when enabling videoserver Ch2 in the Template Editor. When the video source is changed in the Newsroom Control System or Viz Mosart GUI, the audio faders will follow video crosspoints.
  Should be used for audio mixers with mono-faders which cannot be coupled in the mixer.
- **Input type:**
  - **camera:** Select to assign this crosspoint as a camera type. This is used by the **Audio Panel** application.
  - **external_source:** Select to assign this crosspoint as an external source. This is used by the **Audio Panel** application.
  - **videoserver:** Select to assign this crosspoint as a server channel.
  - **GRAPHIC_1-5:** Select to assign this crosspoint as a graphics engine.
  - **ROUTER_RIPPLE_A:** Select to use as router source *A*.
  - **ROUTER_RIPPLE_B:** Select to use as router source *B*.
- **Hide from user:** Do not show crosspoint in AV Automation. This prevents the crosspoint from displaying in the **Crosspoint** drop-down menu in **Templates** and the **ActiveX** application. Default: *Unchecked.*
- **Comments:** Only for useful setup notes, in this menu. Not visible or used anywhere else in the application.

Copyright © 2024 Vizrt

Audio Configurations

- To work with just audio mixer configuration, Audio and Video Setup and from the menu bar, select **Edit > Audio config**.



- As a minimum, you should enter **Channel Name** and **Physical input**.

Settings

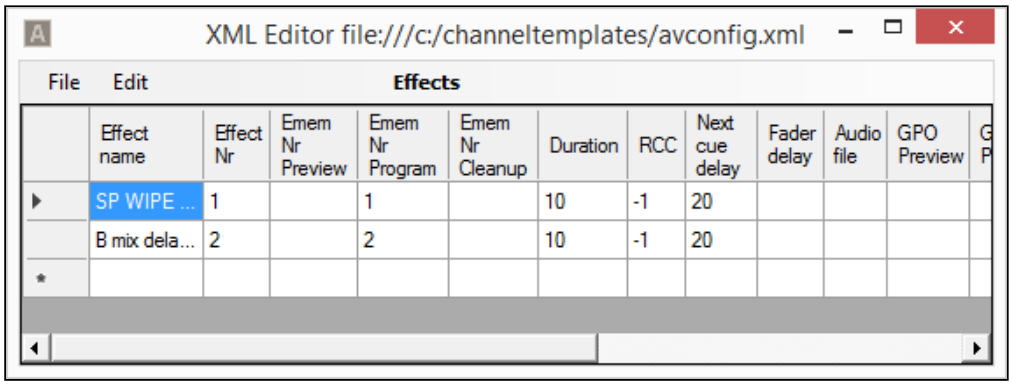
- **Channel Name:** Name to use for the audio fader in the Viz Mosart system (for referencing the audio channel in Viz Mosart templates).
- **Physical input:** For audio mixers that use fader numbers or MIDI assigned faders, this number must correspond with the fader number on the audio desk.
- **Mixer name:** A unique fader identifier, for audio desks that use fader names instead of numbers. (LAWO EMBER+: These types of faders are controllable and may be configured: auxes, groups, inputs, sums. Use the value of the 'startup' label, found in the Ember+ tree/ viewer, located under mixers/primary/channels for the respective fader types. For a *stereo* fader, use the *left*. *Do not* configure the *right*.)
- **Input type:** For the internal Viz Mosart audio logic, the input type of the fader must be set. This can also be done in the **Template Editor**.
- **Initial level:** Desired level for the audio fader to obtain after a reload of the Viz Mosart rundown.
- **Initial standby:** Selects the faders that should *not* be reset at the start-up of the **AV Automation** application. Default: *Unchecked.*
- **Delay address:** (Lawo non-Ember+ audio mixer boards only) Gives the signal address for the delay function.
- **Loudness channel:** Defines the channel number for loudness level control.

Template-driven Effects

You can define effects in Viz Mosart that can then be recalled by a template. This avoids having to recreate the effect in each template that refers to it.
Use the **Effects** table, which is then referenced during the execution of a template that has specified an **Effect name** or **Effect Nr**.

Copyright © 2024 Vizrt

- To edit vision mixer effects configurations, Audio and Video Setup and from the menu bar, select **Edit > Effects**.



- An *effect* may perform various actions on the vision mixer, the RCC, the Audio player, the GPO, and possibly other devices.

> **Note:** When using an entry in the predefined Viz Mosart **Effect** table, the EMEM *must include the transition* between bus A and B, since transition is not performed by Viz Mosart.

Settings

- **Effect name:** Name or description of the effect. *Mandatory*.
- **Effect Nr:** Value of the effect assigned from the newsroom system. Used in conjunction with **Studio Setup** to uniquely identify effects. *Mandatory*.
- **EMEM Nr Preview:** Video switcher preset to recall when the template is taken in the preview. *Mandatory*.
- **EMEM Nr Program:** Video mixer preset to recall when the template is taken on the program. *Mandatory*.
- **EMEM Nr Cleanup:** This register will be recalled when cueing the next template. Use only if the register used for the effect demands another "cleanup" register recall to prepare the mixer for the next events.
- **Duration:** Duration of the effect in frames. Will hold the triggering of the template following the effect, for this amount of time.
- **RCC:** Recall a **Robotic Camera Control** shot.
- **Next cue delay:** Used to hold the cueing in the preview of the second template/element after the effect. The number of frames entered will be in addition to what is already set as Min. cue delay in the **Video Switcher** setup.

  > Note: **Next cue delay** is ignored if **EMEM Nr Program** has a negative value!

- **Fader delay:** Delay opening the audio effect fader (the **Audio Player** fader). Value is given in frames. Ignored if this value is greater than **Next cue delay**.
- **Audio file:** Viz Mosart can play an audio file on **Audio Player** synchronous with the mixer effect. Specify the filename for the audio file here.
- **GPO Preview:** Send GPO # when entering Preview mode.

Copyright © 2024 Vizrt

- **GPO Program:** Send GPO # when entering Program mode.
- **Macro Preview:** Refer to the **Text** field in **Recall preview** in the Template Device Functions - Macro Recall section.
- **Directtake:** Number of Direct Take template triggered when performing an effect transition. Often used to allow control of additional devices.
  For example, this could be a direct take triggering a named overlay graphic or a direct take with additional commands required by the vision mixer for this effect.
- **Studio Setup:** Specifies a specific studio setup for which the effect will be run. This attribute is optional and is used in conjunction with **Effect Nr** to uniquely identify effects.
- **Macro Preview Action:** Refer to the **Drop-down** description in **Recall preview** in the Template Device Functions - Macro Recall section.
- **Macro Program Action:** Refer to the **Drop-down** description in **Recall program** in the Template Device Functions - Macro Recall section**.**

  > ⚠ **Macro Preview Action**, **Macro Program Action:** *Load Project* could affect the performance of the switcher and should thus be used with caution during a running show.

- **Macro Program:** Refer to the **Text field** details in **Recall program** in the Template Device Functions - Macro Recall section.

Effect Number and Studio Setup

Normally the effects are shared between all the studios/control rooms. But in some cases, a given effect is required to function differently in one studio setup than another. This is achieved by adding an extra entry for that effect, but with the **Studio Setup** that this effect is reserved for. The rules for how this works are as follows:

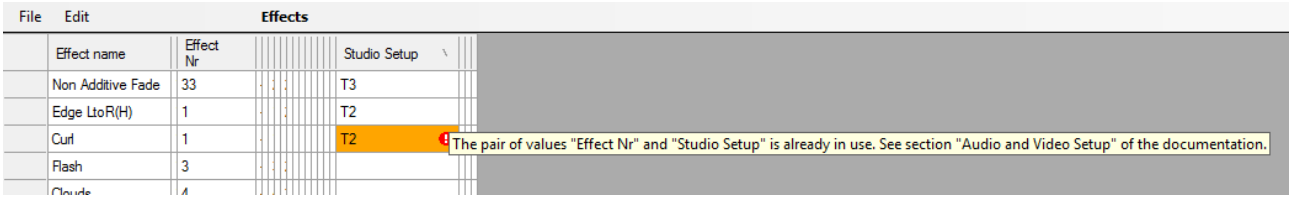
- The combination of **Effect Nr** and **Studio Setup** must be unique within the **Effects** table. Multiple effects can have the same **Effect Nr**, but these must have *different* values in **Studio Setup**.
- An effect which has **Studio Setup** specified will only be executed for the corresponding studio setup.
- An effect without **Studio Setup** specified will be executed for *all* studio setups that do not have any dedicated effect specified.

Viz Mosart processes an effect in the following order:

- Search for an effect with given **Effect Nr** and with **Studio Setup** that matches the *active* Studio Setup.
- If no match is found, search for an effect with given **Effect Nr** and *empty***Studio Setup**.

If there are duplicate values of **Effect Nr** and **Studio Setup** the cell will be highlighted and an error message will indicate the reason for the warning.

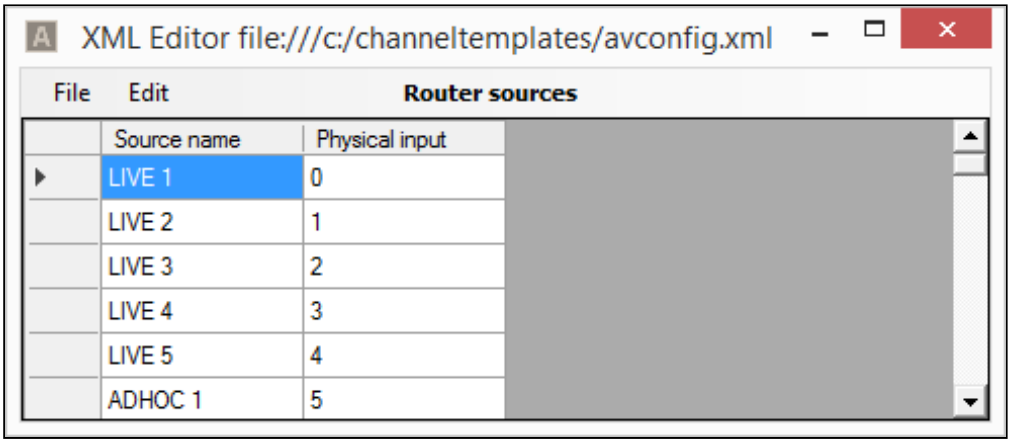
Copyright © 2024 Vizrt



Router Source Configurations

Router source crosspoints can act as an emergency switcher if the vision mixer fails to switch. Alternatively, a router switch can change a background wall source in the studio, or anything in your production that requires switching.

To edit Router source configurations, Audio and Video Setup and from the menu bar, select **Edit > Router sources**.



Router switches are predefined as part of a Viz Mosart template. Keyboard shortcuts can also be programmed for a router switch to take place.

Settings

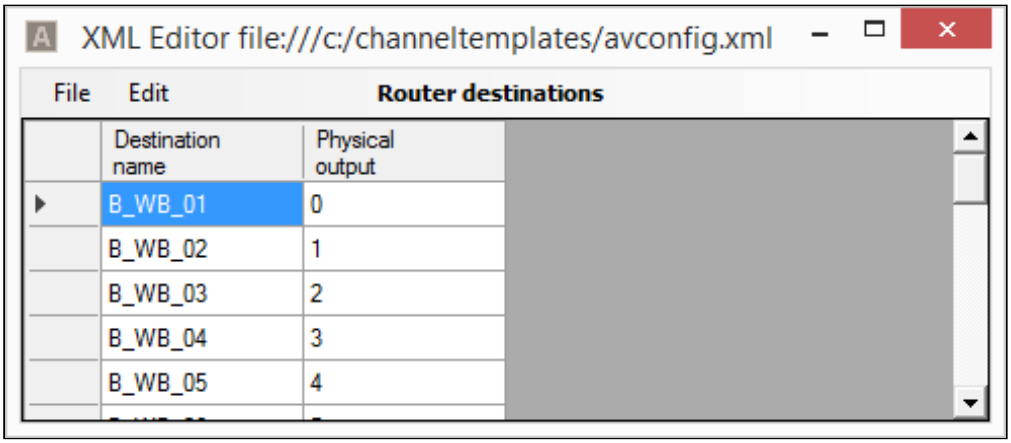
- **Source name:** Name to use for the router source in the Viz Mosart system.
- **Physical input:** Physical or virtual input to the router.

Router Destination Configurations

Router destination crosspoints can act as an emergency studio output if the vision mixer fails to switch. Alternatively, a router destination could be a monitor in the studio, a transmission output, or anything that needs switching as part of your production.

To edit Router destination configurations, Audio and Video Setup and from the menu bar, select **Edit > Router destinations**.

Copyright © 2024 Vizrt



Settings

- **Destination name:** Name to use for the router destination in the Viz Mosart system. This name will be displayed in the router crosspoints drop-down menu in all templates and in the ActiveX application.
- **Physical output:** Physical or virtual output on the router.
- **Audiolink**: Linked device

Copyright © 2024 Vizrt

4.6 Device Connection Strings

This section contains an overview of Device Configuration Strings for connection to equipment in your broadcast environment.

A *device* in this context is a physical unit, such as an Viz Engine or a video server. The *device driver* is the code that provides Viz Mosart the ability to handle and control the actual device. The *Device Connection string* is a textual description that identify the type, name and various other parameters required for Viz Mosart control the device. In some cases, you need to provide additional parameters for the device, typically specified in an XML file, in addition to the Connection String.

This section contains the following topics:

- Video Server, MAM and Database Connection Strings
- Video Wall Connection Strings
- Device Configuration Files

Copyright © 2024 Vizrt

4.6.1 Video Server, MAM and Database Connection Strings

A connection string contains parameters needed for connecting a studio device to Viz Mosart. The string always comprises two parts:

- Standard Connection String Properties
  For example *Name*, *Type*, *Server*, *Config* (Config is the name of the associated file that contains custom settings).
- Device-specific Connection String Parameters
  Driver-specific configurations. For example, what to log in error situations, manipulation of standard commands, or timeout behavior.

Connection strings are used both in the Media Administrator configuration and, in some cases, AV Automation.

In AV Automation:

- Connection strings get autogenerated when configured in the video server GUI.
- You can enter additional device parameters to match specific control room parameters.

This section describes the following connection string types:

- System-level Location of Device Configuration Files
- Standard Connection String Properties
  - Viz Mosart Video Server Connection String
  - General Configuration File Properties
  - General Search Configuration File Properties
- Device-specific Connection String Parameters
  - AirSpeed Classic Connection String
  - AirSpeed Multi Stream / AirSpeed 5000 Connection Strings
  - AirSpeed Multi Stream Search
  - Amadeus Connection String
  - EVS LinX Connection String
  - EVS Xedio Connection String
  - Grass Valley K2 Connection String
  - Harris Nexio Connection String
  - JupiterWebService Connection String
  - MVCP Connection String
  - Omneon Connection String
  - Omnibus OPUS Connection String
  - Orad OCIP Connection String
  - Quantel Connection String
  - ScreenLL Connection String
  - SimpleTestPlayer Connection String
  - SQL Database Connection String
  - VDCP Connection String
  - Viz Engine Connection String
  - Viz One and Media Service Connection String

Copyright © 2024 Vizrt

System-level Location of Device Configuration Files

In a connection string where no path is provided as a value to the Config parameter, Viz Mosart searches for the file in the following order of folders:

- `%ALLUSERSPROFILE%\Mosart Medialab\ConfigurationFiles\`
- `C:\ChannelTemplates\`
- `{Application path}\`
- `{Application path}\ConfigurationFiles`
- `{Current folder}\ConfigurationFiles`

Standard Connection String Properties

Irrespective of device type, several values are standard parameters:

- Viz Mosart Video Server Connection String
- General Configuration File Properties
- General Search Configuration File Properties.

Viz Mosart Video Server Connection String

Parameters not recognized by the driver in question will simply be ignored.

```
Name=myServerName; Type=myServerType; Server=myServerAddress; Config=myConfigFile.xml
```

- **Name:** A logical name consistently associated with the device, allowing the device to be uniquely recognized throughout the system setup.
  All Media Administrator connection strings contain a *name*, which forms a distinction for Media Administrator, when handling multiple clip metadata sources and video servers.
  If the same video server also is to be controlled by AV Automation, *Name* in AV Automation must be the same as *Name* used here.

> ⚠ **Note**: Naming plays a critical role in the powerful feature *Dynamic Video Server Configuration*. Please see the topic Dynamic Video Server Configuration.

- **Type:** Specifies a driver to be used for connecting to the server. Potential values are:
  - AirSpace: AirSpace driver.
  - AirSpeed: AirSpeed driver.
  - AirSpeed Search: As AirSpeed driver, but contains search functionality only.
  - AirSpeedMultiStream: AirSpeed Multi Stream driver, also Airspeed 5000.
  - AirSpeedMultiStreamSearch: As AirSpeed Multi Stream driver, but only search functionality.
  - Amadeus: Amadeus MAM driver.
  - EVS LinX: EVS LinX driver.
  - EVSXedio: EVS Xedio driver, used by AV Automation.

Copyright © 2024 Vizrt

  - EVSXedioSearch: EVS Xedio driver for search functionality, used by Media Administration.
  - GrassValleyK2: Grass Valley K2 driver.
  - HarrisNexio: Harris Nexio driver.
  - Jupiter Web Service: Jupiter web service driver.
  - MVCP: MVCP driver.
  - OPUS: Omnibus MAM driver using OPUS interchange.
  - Omneon: Omneon (Harmonic) driver.
  - OmneonSearch: As Omneon driver, but contains search functionality only. Used by Media Administrator to obtain clip status from an Omneon (Harmonic) video server.
  - Orad OCIP: Orad OCIP and Avid FastServe driver.
  - Quantel: Quantel driver.
  - ScreenLL: Screen LinguaLib search driver.
  - SimpleTestPlayer: SimpleTestPlayer driver
  - SqlMediaSearch: SQL database driver.
  - VDCP: VDCP driver over serial port.
  - VDCPtcp: VDCP driver over TCP/IP port.
  - VizEngine: Viz Engine driver.
  - VizOne:Viz One and Media Service connection string.
- **Server:** Specifies the server name or the IP address of the video server.
- **Config:** Specifies a configuration file containing custom configuration for a particular video server driver. Editing these files is typically done during installation.
- **Static**: (Effective for *Media Administrator* only) If *True*, Media Administrator always connects to the MAM system (or video server), also when **Enable dynamic configuration** (see the topic Dynamic Video Server Configuration) is *True*, and this MAM system (or video server) is *not* part of the salvo selected in AV Automation. (see also section *Virtual Server Ports* in Automation Devices Video Servers).

  > ⚠ Normally, a MAM system will not be part of any salvo. This setting is mostly intended for MAM systems which should always be connected, regardless of the selected salvo. Default: *False*.

General Configuration File Properties

- **VerifyClipDelay** (integer): Delay in frames to wait before verifying new clips from server.
- **KeepCache** (boolean): When set to *true*, the internal Media Administrator clip cache is kept on reconnect (Default: *false*).
- **RefreshOnUpdate** (boolean): If set to *true*, any clip update message received from the video server results in a clip refresh operation (Default: *false*).
- **ReleasePortOnDispose** (boolean): If *true*, the assigned video port is released when server connection is closed (Default: *true*).
- **SupportSearchUsingObjSlug** (boolean): Activates searching using *objSlug*. ObjSlug is the clip slug as shown in the GUI Media pool Search tab. (Default: *true*).
- **SupportSearchUsingRefID** (boolean): Activates searching using *RefID*. RefID is a specific clip property that has been defined as a reference (Default: *false*).

Copyright © 2024 Vizrt

- **VerifyModifiedClips** (boolean): If *true*, clip modified/update events for all clips trigger a complete verification (for example treated as the clip was created). If *false*, only clips currently in the cache are verified. (Default: *false*).
- **CueTimeout** (integer): Specifies the maximum time in milliseconds for AV Automation to wait for a cue operation to complete. (Default: *1000 msec*).
- **PlayTimeout** (integer): Specifies the maximum time in milliseconds for AV Automation to wait for a play operation to complete. (Default: *0*).
- **SetLoopDelay** (integer): Specifies the time when to set the clip to play in loop after a successful cue operation, in milliseconds. (Default: *1000 msec*).
- **NextPingDelay** (integer): Specifies the interval (in milliseconds) at which AV Automation issues a 'Ping' operation (verifying that the connection is still OK) against the server. The nature of the 'Ping' operation depends on the driver chosen. (To modify the ping interval for Media Administrator see Media Administrator Properties Editor > **Ping Delay**.)
- **NextServerAttemptDelay** (integer): In case of failure, specifies the time between trying to reconnecting to the video server, in milliseconds. (Default: *5000 msec*).
- **MirrorActivePortTimeout** (integer): For mirroring setups. Specifies the time to wait for the active port to complete before waiting for any of the ports, in milliseconds. (Default: *100 msecs*).
- **IgnoreCase (boolean):** If *true*, searching for clips is done case insensitive. Must be used in combination with <MatchExpression ignoreCase="true" /> from General Search Configuration File Properties.
- **AssetToMediaMap**: Choose from two ways of mapping Asset to MediaObject. *UseTitleAsSlug* or *UseMamIdAsSlug*.
- **MaxClipNameLength**: Naming standard for recordings.
- **ClipNamePattern:** When recording a file, the system can be configured to use a predefined pattern for filenames. The pattern shall be defined in the associated config file for the video server. The following variables are available for injecting into the clip name:

Variable name	Description
clipname	The value given in the recording dialogues, or from custom template commands.
gallery	The name of the gallery.
templateset	The name of the currently selected template set.
timestamp	The current time. This value can be formatted using the patterns from MSDN.

Example

```
Gallery = MyGallery
Current template set = MyTemplateSet
Clipname = MyClipName
Current Time = 21 September 2016 12:00:00
```

Config File For Video Server

```
<DeviceConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="
http://www.w3.org/2001/XMLSchema">
  <Properties>
    <item name="ClipNamePattern" value="{gallery}_{templateset}_{clipname}
_{timestamp:yymmdd}" />
  </Properties>
</DeviceConfig>
```

The file name sent to recording is *MyGallery_MyTemplateSet_MyClipName_160921*.

General Search Configuration File Properties

- **HitSortOrder** (enum): Specifies how to sort a list of multiple hits for a single search. Used in combination with the BestMatchMethod property.
  Options:
  - **Ascending:** The list of hits are sorted ascending alphabetically.
  - **Descending:** The list of hits are sorted descending alphabetically.
  - **Unspecified** (Default): No sorting is applied to the resulting hits.
    Example: <HitSortOrder>Descending</HitSortOrder>
- **BestMatchMethod** (enum): Method used to determine single search result if a search results in multiple hits. Options:
  - **First:** The hits are sorted alphabetically and the first hit is returned.
  - **Last:** The hits are sorted alphabetically and the last hit is returned.
  - **Filter** (Default): Returns the first hit evaluated *true* from a MatchExpression filter.
- **MatchExpressions** (list): List of filters to be used for narrowing searches that may result in multiple hits.
  Each MatchExpression filter is constructed from three attributes:
  - **ignoreCase:** (boolean) *True* if the filter is case insensitive. Must be used in combination with property "IgnoreCase" from General Configuration File Properties.
  - **expression:** (string, regex) Expression used to create a regular expression to be applied to the given clip name. Use the pattern attribute to extract parts of the original clip name to search for. (Default: Empty = *exact search*).
  - **pattern:** (string) Regular expression used to extract information from the clip name. (Default: *The original clip name*).
    An empty MatchExpressions list results in an exact case sensitive search.

The following example sets up a regular expression filter searching:

Copyright © 2024 Vizrt

MatchExpressions

```
<MatchExpressions>
    <!-- Returns true for all hits with clip names starting with the given clip name
-->
    <MatchExpression expression="^$1.*$"/>


    <!-- Returns true for all hits with identical clip name but case insensitive -->
    <MatchExpression ignoreCase=true/>


    <!-- Returns true for all hits with clip names containing leading or trailing
spaces -->
    <MatchExpression pattern = "^\s*(.*)\s*$" expression = "$1"/>
</MatchExpressions>
```

Device-specific Connection String Parameters

AirSpeed Classic Connection String

Applies To Models

- AirSpeed
- AirSpace

Connection String Properties

- **Type:** AirSpeed and AirSpace.
- **Name:** See Name above. Defines the ID of this connection (Default: *AirSpeed*).
- **Server:** Defines the hostname or IP-address of the server.
- **Config:** Defines the name of the configuration file.

Examples

AV Automation:

```
Name=HAR; Type=AirSpeed; Server=192.168.1.1; Config=ClipServerAirspeed.xml
```

Configuration File Properties

Configuration file properties can be overridden by connection string properties.

- **KeepCache** (boolean): How often (in seconds) to look for clip changes. value=*1* is the default value if not configured here (Default: *false*).

Copyright © 2024 Vizrt

- **RefreshOnUpdate** (boolean): When set to *true*, the cache is updated on each update clip message from the server (Default: *true*).
- **VerifyClipDelay** (integer): Delay in frames to wait before verifying new clips from server.
- **ReceiveTimeout** (integer): Time in seconds to wait for response from AirSpeed (Default: *10*).
- **ChangeCheckInterval** (integer): Time in seconds how often to poll AirSpeed for clip changes (Default: *2*).
- **HeartbeatInterval** (integer): Time in seconds to send out a heartbeat in case of no communication detected from AirSpeed (Default: *30*).
- **GetMinimumList** (boolean): Set to *true* to retrieve only clip names when initializing database (Default: *false*).
- **DelayGetClipData** (boolean): Delay getting detailed clip information. Information is retrieved for clips in rundown only (Default: *true*).
- **DelayEventsOnDatabaseInitialization** (boolean): If *true*, all clip handling during database load is postponed until last clip is received (Default: fa*lse*).
- **SignalDatabaseReady** (boolean): Refreshes all clips after database has been built instead of refreshing for every item in the database (Default: *true*).
- **VerifyModifiedClips** (boolean): If *true*, all modified clip events for clip not in cache are verified, as if they were created (Default: *true*).
- **ReceiveBufferSize** (integer): Set buffer size used for receiving date from AMS. Default `16384` = two times default buffer size of Net.Socket (Default: *16384*).
- **Encoding** (string): Set to encoding used for send/receive from AMS (Default: *UTF-8*).
- **SupportSearchUsingObjSlug:** Ignores search for objSlug if set to *false* (Default: *true*).
- **SupportSearchUsingRefID:** Ignores search for refID if set to *false* (Default: *false*).

Use for debugging purposes only:

- **VerboseLogging** (boolean): Set to *true* to enable logging of received messages from AMS (Default: *false*).
- **DatabaseCache** (boolean): If set to a valid filename, initial load database stores the content in this file (Default: *false*).
- **DatabaseCacheUpdate** (boolean): Determines when the debug database cache is updated (either [*Never,Always,Initial*]).
- **LoadDatabase:** Specifies whether the AirSpeed clip database should be initially loaded and maintained by the Media Administrator. Setting this attribute to *true* is required if any search should be possible using an AirSpeed video server.

AirSpeed Multi Stream / AirSpeed 5000 Connection Strings

The AirSpeed Multi Stream drivers supports up to six multiple player ports per connection.

Applies To Models

- AirSpeed Multi Stream 1.8
- AirSpeed 5000

Copyright © 2024 Vizrt

Connection String Properties

- **Type:**
  - AV Automation: AirSpeedMultiStream
  - Media Administrator: AirSpeedMultiStreamSearch
- **Name:** See **Name** in Standard Connection String Properties above.
- **Server:** See Standard Connection String Properties above.
- **Config:** See Standard Connection String Properties above. The installed xml config file is ClipServerMultiStream.xml. Unless changed, the value of **Config** should be *ClipServerMultiStream.xml*. See Examples below.

Optional

- **BasePort*:** Specifies the number of the lowest-numbered port on the server. (Default: *59451*). For AV Automation, six consecutive player ports, starting at this one, are available for playout. (They should be referred to by numbers 1 to 6 in AV Automation.) For Media Administrator, refer to Player and the note below for how the actual port number is calculated.
- **AV Automation only**
  - **LoopPostRoll:** (only used when *NewLooping* is *true*) No. of frames before clip-end to restart loop. Should probably be set to a positive value to obtain smooth looping. (Default: *4*).
  - **NewLooping:** Whether to use an alternative looping mechanism. Must be *true* if MPEG-2 HD clips are involved. (Default: *false)*.
  - **OpenSomePlayerAfterConnect**: By default, server connection is considered successful ('green') without connecting to any players. If this setting is enabled, also connection to some configured player will be necessary. (Default: *false)*.
    Set to *true* if connection problems are experienced, e.g.
    i. Green server line
    ii. Unsuccessful port connect (red port line)
    iii. Reconnect
    iv. ... and so on ...
  - **OpenPlayerAfterConnectTimeout**: (only used when *OpenSomePlayerAfterConnect* is *true*) Timeout (in milliseconds) for the above *OpenSomePlayerAfterConnect* functionality (Default: *2000*).
  - **RecordEjectFirst**: If *true*, initiating a recording on a channel/port will first eject any existing clips (Default: *false*).
  - **RecordOverwrite**: If *true*, any recording initiated on an existing clip will delete the existing clip before creating the new (Default: *true*).
  - **RecordUseNameAsClipId**: If *true*, the clip name is also used as clip id when preparing a recording. Otherwise the airspeed will automatically assign clip ids (Default: t*rue*).

⚠ **Note:** The asterisk (*) above denotes the TCP port to connect to becomes BasePort + Player - 1.

Copyright © 2024 Vizrt

Examples

AV Automation:

```
Name=AMS; Type=AirSpeedMultiStream; Server=192.168.1.1;
Config=ClipServerMultiStream.xml
```

Configuration File Properties

With a few exceptions (Type, Name, Config), the connection string properties described above may alternatively be specified in the configuration file. (A connection string setting overrides a corresponding file setting.)

In addition the following properties may be specified in the configuration file:

- **Media Administrator only**
  - **DelayEventsOnDatabaseInitialization** (boolean): If *true*, all clip handling during database load is postponed until last clip is received (Default: *false*).
  - **DelayGetClipData** (boolean): Delay getting detailed clip information. Information is retrieved for clips in rundown only (Default: *true*).
  - **Encoding** (string): Set to encoding used for send/receive from AMS (Default: *UTF-8*).
  - **GetMinimumList** (boolean): Set to *true* to retrieve only clip names when initializing database (Default: *false*).
  - **IgnoreCase:** Whether clip search is case-insensitive (Default: *false*).
  - **NumClips** (integer):Number of clips to receive for each server query (Default: *50*).
  - **ReceiveBufferSize** (integer): Set buffer size used for receiving date from AMS. Default *16384* = two times default buffer size of Net.Socket (Default: *16384*).
  - **SignalDatabaseReady** (boolean): Refreshes all clips after database has been built instead of refreshing for every item in the database (Default: *true*).
  - Use for troubleshooting only:
    - **DatabaseCache** (boolean): If set to a valid filename, initial load database stores the content in this file (Default: *false*).
    - **DatabaseCacheUpdate :** Determines when the debug database cache is updated (either [*Never,Always,Initial*]).
    - **VerboseLogging** (boolean): Set to *true* to enable logging of received messages from AMS (Default: *false*).
      Effective only if Verbose logging is turned on in Media Administrator.
- **AV Automation only**
  - **ReadTimeout:** Time in seconds to wait for response from AirSpeed (Default: *5*).

AirSpeed Multi Stream Search

This driver is similar to the AirSpeed Multi Stream driver, but contains search functionality only. It is used by Media Administrator for obtaining clip status from an AirSpeed Multi Stream or 5000 video server.

- **BasePort*:** The lowest port number to connect to (Default: *59451*).

Copyright © 2024 Vizrt

- **Player*:** The player number (1-6). Used for setting the TCP port (Default: *1*).
- **LoadDatabase** (boolean): Set to true to build internal database of clips present on server and enable clip monitoring (Default: *false*).
- **ChangeCheckInterval:** Time in seconds how often to poll AirSpeed for clip changes (Default: *1*).
- **ReceiveTimeout:** Time in seconds to wait for response from AirSpeed (Default: *10*).
- **HeartbeatInterval:** Time in seconds to send out a heartbeat in case of no communication detected from AirSpeed (Default: *30*).

> ⚠ **Note:** The asterisk (*) above denotes the TCP port to connect to becomes BasePort + Player - 1.

Example

```
Name=AMS; Type=AirSpeedMultiStreamSearch; Server=192.168.1.1; BasePort=59451; Player=2;
 ChangeCheckInterval=5; ReceiveTimeout=5; HeartbeatInterval=20;
Config=ClipServerMultiStream.xml
```

Amadeus Connection String

```
Name=HTC-DB; Type=Amadeus; Port=[port]; Config=ClipServerAmadeus.xml
```

Connection String Properties

- **Name:** Amadeus ID for database. See Name in *Standard Connection String Properties* above.
- **Type:** Defines the Amadeus driver for clip handling.
- **Port:** Set to port to listen for an Amadeus connection.

Optional

- **Port:** Defines the connection port (Default: *2202*).
- **ResponseReadTimeout:** Defines the wait time until timeout (default: *5000*).
- **PollIntervalClips:** Defines the time between polling (Default: *15*). Set to *0* to disable.
- **RundownPrefix:** Defines the prefix to add for the rundown name (Default: "").
- **GetAllCGs:** If *true*, METAREQ_ALLCGS message type is used when getting clip info (Default: ""). For more details, see the Amadeus documentation.
- **LowerCaseRundowns:** If *true*, then the rundown name is always generated with lower case (Default: *false*).
- **IgnoreSTEG:** If *true*, all messages from Amadeus starting with STEG are ignored (Default: *true*). For more details, see the Amadeus documentation.
- **StegCommandIfSubtitled:** Sends the specified STEG command if clip is subtitled (Default: *KLOCKA INVOKEOFF*). For more details, see the Amadeus documentation.
- **SubtitleStegOutOn:** Sets the graphics_out_on xml value in the STEG command set to Amadeus (Default: *BACKGROUNDEND*).

Copyright © 2024 Vizrt

- **PurgeCacheInterval:** Use hh:mm:ss format (Default: *00:00:00*).
- **PurgeCacheAge:** Use hh:mm:ss format (Default: *00:05:00*).

EVS LinX Connection String

This connection string is used for EVS video servers controlled by the LinX protocol (as opposed to EVS Xedio systems, see EVS Xedio Connection String).

```
Type=EVS LinX; Name=[name]; Server=[IPaddress]; LoadDatabase=true;
Config=ClipServerEvsLinX.xml
```

Example

```
Type=EVS LinX; Name=EvsLinXName; Server=192.168.62.6; LoadDatabase=true;
Config=ClipServerEvsLinX.xml
```

Connection String Parameters

- **Type**: EVS LinX or EVS (*EVS* is retained purely for backward compatibility).
- **Name**: See Name in *Standard Connection String Properties* above.
- **Server**: Defines the IP address of the server.
- **LoadDatabase**: If *true*, loads all clip metadata into local cache (Default: f*alse*). Use *true* for Media Administrator.
- **Config**: Configuration file containing further configuration settings.

Optional

- **LogOutput**: The LinX log file created for the LogFlag option 0x00000003. The files for the other options are in the same folder, and with names derived from LogOutput (see **LogFlag** below for details.) The folder should exist in advance, and the Viz Mosart user should have permission to create and write files.
  AV Automation and Media Administrator should log to different files, so LogOutput should be specified (with different folders) in the respective connection strings rather than in this file.
- **VeryVerbose**: If *true* (and the Viz Mosart application issues verbose logging), then some extra log messages are issued (Default: *false*).

Config File Parameters

- **SupportSearchUsingObjSlug**: Search by 'slug' (clip name) is supported (Default: *true*).
- **SupportSearchUsingRefID**: Search by a configurable clip property (given by the SearchPropertyName parameter; see below) is supported (Default: *false*).
- **HeartbeatTimeout**: The time (in milliseconds) after which the server connection may close if there is no activity (Default: *12000*).
- **KeepAliveIntervalDivisor**: In case of connection problems, multiply both HeartbeatTimeout and KeepAliveIntervalDivisor by the same factor (for example, two) (Default: *8*).

Copyright © 2024 Vizrt

- **LocalMtpcIpAddress**: Necessary only if the Viz Mosart server has several network adapters, e.g. a virtual adapter (or several virtual adapters) in addition to the physical adapter, e.g. when VMWare (or similar) is installed on a demo PC, or when connecting through VPN. In this case, use the IP address (in the usual format of four dot(.)-separated groups of one to three decimal digits) of the network adapter connected to the EVS servers.
- **LogFlag**: Bitmask specifying LinX log level. Available options are listed in the table below. (In addition, *linxFunctions.log* is produced.) The options may be combined. For example, 0xFFFFFFFF produces *all* log files.

Bitmask	Description	Log file, given LogOutput=...\linx.log
0x00000003	Function entry and exit points, with parameters and return code	linx.log
0x00000010	Connections	linxConnect.log
0x00000200	Keeping alive	linxKeepAlive.log
0x00000400	Databases	linx_Archive.log
0x00008000	Debug	linxDebug.log
0x01000000	Tools	linxTool.log

- **MaxTries**: Obsolete. Instead, use SharedNbMaxConnectRetry and SharedConnectTimeout, described below. (The maximum number of times to try to synchronize the database (Default: *10*).
- **NbMaxLinxConnection**: The maximum number of connections. The default is 18, to accommodate two EVS servers. Add *9* for each additional server.
- **OsdDefine00 - OsdDefine11**: Defines the 12 OSD lines.
- **OsdShowHideError**: Defines whether or not to show OSD.
- **OsdShowHideErrorSignals**: Defines whether or not to show OSD. (For further details of these and other OSD settings, please see the default configuration file.)
- **SearchPropertyName**: Clip property that may be used for clip search. The default is *Varid* (a LinX specific clip ID that may be used to identify clips in the NRCS). If some other property is needed, please ask Viz Mosart personnel for a list of available values.
- **SharedNbMaxConnectRetry**: Maximum number of retries in case of connect error on a shared connection. Increase if the error code 'Too many retries' is encountered, e.g. when synchronizing the database. (Default: *16*).
- **SharedConnectTimeout**: Maximum time in milliseconds to execute connection request on a shared connection. Consider increasing if the error code 'Too many retries' is encountered, e.g. when synchronizing the database. (Default: *500*).

Copyright © 2024 Vizrt

Applies To Models

- XS family (6U, 4U)
- XT family (XT2+, XT3, not nano)

Device Prerequisites

- Multicam 10.03, or later
- LinX license code
- Channels to be controlled by Viz Mosart must have Main CTRL LinX.
- The network, including any firewalls, connecting the Viz Mosart server(s) and the EVS server(s) must support and allow TCP (ports `50000` - `50002`) and UDP (unicast and multicast, ports `50100` - `50108`).

EVS Xedio Connection String

These connection strings are used for EVS Xedio video servers (as opposed to EVS video servers controlled by the LinX protocol, see EVS LinX Connection String).

From a Viz Mosart perspective, a Xedio system has three parts:

- One or more playout servers
- A control center
- A DB connected to through ODBC.

AV Automation controls the playout servers, whereas Media Administrator connects to the control center and the database. Consequently, the connection strings for AV Automation and Media Administrator are somewhat different.

AV Automation Connection String

```
Type=EvsXedio; Name=[name]; Config=ClipServerEvsXedio.xml
```

Example:

```
Type=EvsXedio; Name=EvsXedioName; Config=ClipServerEvsXedio.xml
```

AV Automation Connection String Parameters

- **Type:** EvsXedio.
- **Name:** See Name in *Standard Connection String Properties* above.
- **Config:** Defines the configuration file containing further details.

> ⚠ **Note:** The parameters mentioned so far are insufficient to connect to a playout server. The IP address and port number are also required.
> These are specified (separated with :) in the Video port field(s) in AV Automation Device Properties, for example *192.168.76.11:4021*.

Copyright © 2024 Vizrt

Media Administrator Connection String

```
Type=EvsXedioSearch; Name=[name]; Server=[IPaddress]; Config=ClipServerEvsXedio.xml
```

Example:

```
Type=EvsXedioSearch; Name=EvsXedioSearchName; Server=192.168.76.10;
Config=ClipServerEvsXedio.xml
```

Media Administrator Connection String Parameters

- **Type**: EvsXedioSearch.
- **Name**: See **Name** in *Standard Connection String Properties* above.
- **Server**: Defines the IP address for the Control center.
- **Config**: Defines the configuration file containing further details.

> ⚠ Parameters mentioned so far are insufficient to connect to the database. Additional parameters DataSourceName, User, and Password are required.
> The defaults are probably sufficient. If not, these are more conveniently set as configuration file parameters, described below.

Configuration File Parameters

- **PrefixTypeEdit**: MOS objIDs with this prefix are treated as 'edits' (Default: *EVSE*).
- **PrefixTypeMedia**: MOS objIDs with this prefix are treated as 'media' (Default: *EVSM*).

Relevant For AV Automation Only

- **TimeOut**: Defines the timeout in milliseconds (Default: *1*).

Relevant For Media Administrator Only

- **AcceptHighResolutionOnly**: Whether only high resolution clips are accepted (Default: *false*).
- **DataSourceName**: ODBC System DSN entry for the CleanEdit DB (Default: *CleanEditDB*).
- **MonitorClipInterval**: The interval (in seconds) at which found clips are monitored (Default: *60*). Used only if MonitorFoundClips is *true*, see below.
- **MonitorFoundClips**: Whether found clips are monitored (Default: f*alse*).
- **Password**: Password for the CleanEdit DB.
- **PrefixCheckItemReturnAllItemHiResPresent**: The prefix added to the Description of a hi-res clip (Default: *HIRES-*).
- **PrefixCheckItemReturnError**: The prefix added to the Description of an error clip (Default: *ERROR-*).
- **PrefixCheckItemReturnNotPlayable**: The prefix added to the Description of non-playable clip (Default: N*OTPLAYABLE-*).

Copyright © 2024 Vizrt

- **PrefixCheckItemReturnOnlyItemLowResPresent**: The prefix added to the Description of a lo-res clip (Default: *LORES-*).
- **SkipGetDuration**: Whether to skip getting clip duration. Note that this operation may be slow and MAY be rendered unnecessary by future Media Administrator dev (Default: *false*).
- **User**: User for the CleanEdit DB.

Debug Parameters, Relevant For Media Administrator Only

- **LogFile**: CleanEditOcx log file (Default=*C:\EvsLogs\CleanEditOcx\CleanEditOcx.log*).

Grass Valley K2 Connection String

Applies To Models

- K2
- Summit

Connection String Properties

- **Type:** GrassValleyK2 (selects the Grass Valley K2 driver for clip handling).
- **Name:** See **Name** in *Standard Connection String Properties* above.
- **Config**: See Standard Connection String Properties above. The installed xml config file is ClipServerGrassValleyK2.xml. Unless changed, the value of **Config** should be ClipServerGrassValleyK2.xml, cp. the examples below.
- **Bin:** When *LoadDatabase* (see below) is *true*, a 'cache'/'database' of server clips is built and maintained. Unless *IgnoreBin* (see below) is *true*, this 'cache'/'database' comprises the clips in the bin designated by the two settings *Volume* (see below) and *Bin*. (The *Bin* setting must designate a bin in the volume designated by the *Volume* setting.) (The default value is the string default. Yes, there is normally a bin named default.)
- **Domain**: The domain as part of the user credentials.
- **DontUsePath**: (Media Administrator only) For each clip and server, Media Administrator keeps track of the 'id' of that clip on that server. By default (**DontUsePath** *false*), that *id* is the full path and name of the clip on the server. When ***DontUsePath*** is *true* , the *id* is the clip name only. The default *false* may be used in most cases. For a Stratus system, changing to *true* may be considered. (However, this should not be combined with setting *IgnoreBin* (see below) to *true*.)
- **EnsureUniqueObjSlug**: (default *false*) This may be left at the default *false* in most circumstances. However, if the *clip_hirespath* from NCS is the full clip path, e.g. *edl/cmf//BGO-LAB-K2SUMMI/C:/default/Clip*, this should be set to *true*. (But if *IgnoreBin* (see below) is *true*, it is not necessary to also set EnsureUniqueObjSlug.)
- **FilterGetXmlChanges:** (Media Administrator only) If *true*, only clip changes involving the bin selected by the properties **Volume** and **Bin** will be handled. Otherwise (*false*), *all* changes are handled. The default *false* may be used in most cases. For a Stratus system, changing to *true* may be considered. (However, this should not be combined with setting *IgnoreBin* (see below) to *true*.)

Copyright © 2024 Vizrt

- **IgnoreBin**: (default *false*) If *true*, the *Bin* setting (see above) is ignored, and the 'cache'/'database' of server clips (see *LoadDatabase* above) will comprise all clips in the server volume designated by the *Volume* setting. (See below.) In this case, the *clip_hirespath* from NCS *must* be the full clip path, e.g. *edl/cmf//BGO-LAB-K2SUMMI/C:/default/Clip*. (Clips in different bins may have the same name, so the path is necessary to select a particular clip.)
- **LoadDatabase**: Whether to maintain a 'cache'/'database' of server clips. Default *false*. For AV Automation, keep the default *false*. For Media Administrator, use *true*. By default, this 'cache'/'database' comprises the clips in the bin designated by the two settings *Volume* (see below) and *Bin* (see above). However, if the setting *IgnoreBin* (see below) is *true*, the 'cache'/'database' will comprise all clips in the server volume designated by the *Volume* setting.
- **ModificationsInterval** : (Media Administrator only; valid only when **LoadDatabase** is *true)* Interval (in ms) for checking for server change.
- **Pass:** The password as part of the user credentials.
- **Server:** The IP address/hostname of the Grass Valley K2 server.
- **Suite:** This value may be chosen arbitrarily. However, these guidelines should be followed:
  - For each K2 server, two, or even three, different Suite values should be used: First, the values for Media Administrator and AV Automation should be different.
  - Second, in a redundancy setup with Main and Backup Mosart server the two values for Media Administrator should be different.
  - (Whereas both Main and Backup AV Automation should use the *same* value, different from the two Media Administrator values.)
  - For further info wrt. the redundancy setup, refer to Failover with K2 Servers.
- **User:** The username as part of the user credentials.
- **Volume:** When *LoadDatabase* (see above) is *true*, a 'cache'/'database' of server clips is built and maintained. This 'cache'/'database' is limited to clips in the volume designated by the *Volume* setting. If *IgnoreBin* (see above) is *true*, this 'cache'/'database' comprises *all* clips in that volume. If *IgnoreBin* is false, the 'cache'/'database' comprises the clips in the bin designated by the *Bin* setting (see above) only. (The *Bin* setting must designate a bin in the volume designated by the *Volume* setting.) (The default value is *V:*)

Examples

AV Automation:

```
Name=Summit; Type=GrassValleyK2; Server=K2SERVER1;
Config=ClipServerGrassValleyK2.xml; Bin=default; User=GVAdmin; Pass=GVPass;
Volume=V:; Suite=AVA; Domain=K2SERVER1
```

Media Administrator:

You also need to add the connection string in Mosart Media Administrator. This is the same connection string as in AVAutomation, except that you need to

- add *LoadDatabase=true* at the end
- use another value for the *Suite* property.

Copyright © 2024 Vizrt

```
Name=Summit; Type=GrassValleyK2; Server=K2SERVER1;
Config=ClipServerGrassValleyK2.xml; Bin=default; User=GVAdmin; Pass=GVPass;
Volume=V:; Suite=GVAdmin; Domain=K2SERVER1;LoadDatabase=true;
```

Configuration File Properties

- **KeepCache**: See General Configuration File Properties above.
- **RefreshOnUpdate**: See General Configuration File Properties above.
- **VerifyClipDelay**: See General Configuration File Properties above.
- **VerifyModifiedClips**: See General Configuration File Properties above.
- **BypassRetrieveServerConfig** (Boolean, default: *false*): (AV Automation only) Whether to *bypass* retrieval of channel configuration from the server. Set to *true* if there are problems connecting to one or more channels (red line).
- **CloseChannels** (Boolean, default *true*): (AV Automation only) Whether channels are closed at disconnect. Use *false* in a redundancy setup with Main and Backup Mosart servers to prevent clip ejection on failover. Otherwise, use *true*. (See Failover with K2 Servers below.)
- **MaxCueRetries**: Maximum number of cue/load attempts. Default: *1*.
- **RecordingBin** (string): (AV Automation only) The server bin where clips are to be recorded. (This bin must be in the *volume* specified by **RecordingVolume**.) Both **RecordingBin** and **RecordingVolume** must be specified if clips are to be recorded in a specific bin. Otherwise, the recording location is undefined.
- **RecordingVolume** (string): (AV Automation only) The server *volume* where clips are to be recorded. Both **RecordingBin** and **RecordingVolume** must be specified if clips are to be recorded in a specific bin. Otherwise, the recording location is undefined.
- **SuspendConnection** (Boolean, default *false*): (AV Automation only) See Failover with K2 Servers below. Use *true* in a redundancy setup with Main and Backup Mosart servers to prevent clip ejection on failover. Otherwise, use *false*.
- **UseExtendedLogging** (Boolean, default *false*): Adds some more logging. Valid only if Verbose logging is turned on. Should normally be kept at the default *false*.
- **UseMovieAddedNotice** (Boolean, default *false*): Whether to utilize the **MovieAddedNotice** element when processing a **MovieAdded** notice.

  > ⚠ **Note:** This setting was introduced to handle added clips at an installation with rather old GV firmware. In the default *ClipServerGrassValleyK2.xml*, this setting is *true*. (But if the setting is *removed* altogether, the actual *default* value is *false*.) Change to *false* (or remove setting) if added clips are not handled correctly (for example, if a clip added on the GV server does not turn solid blue in the Viz Mosart GUI).

- **UseVideoFormatFromServer** (Boolean, default *true*): Set to *false* if clip durations (for example, as shown in GUI) are incorrect.

Failover With K2 Servers

When GVG K2 servers are connected to both main and backup Mosart servers, any cued or playing clips would be ejected from the K2 ports during failover.
This is not desirable, so a further layer of configuration file properties have been provided for avoiding this:

Copyright © 2024 Vizrt

- **CloseChannels**: (Default: *true*)
- **SuspendConnection**: (Default: *false*).

For a redundancy setup with Main and Backup Mosart servers, use instead these settings:

```
<item name="CloseChannels" value="false" />
<item name="SuspendConnection" value="true" />
```

With these settings, the outgoing Mosart server (e.g. main) will disconnect in a way that allows the incoming Mosart server (e.g. backup) to retake the same connections and channels.

This ensures that no clips are ejected in the process.

The prerequisite for this to work is that the Connection String Properties on main and backup are set up with *identical* **Suite** names (for AV Automation).

Stratus

The two connection string properties

- **DontUsePath**
- **FilterGetXmlChanges**

were introduced at installations with a Stratus system to ensure that deleted clips appeared chequered in the Mosart GUI.

The default value of both settings is *false*, and that has, as far as we know, worked for all other installations. We recommend the following:

1. Start using the default value *false* for both settings.
2. If problems of the type indicated (deleted clips stay solid blue in GUI) are encountered, change both settings to *true*.

Other combinations (than both *false* and both *true*) have not been tested.

Harris Nexio Connection String

The Harris Nexio driver supports Binary-Coded Decimal TCP/IP transport. The driver can be configured to be case sensitive on searches and use Unicode encoding. The Harris Nexio driver supports recording.

- Control via RS-422 can be done through VDCP.
- Maximum five video server units.

Connection String Properties

- **Type:** Harris Nexio
- **Name*:** See Name in *Standard Connection String Properties* above. Defines the ID of this connection.
- **Server*:** Defines the hostname or IP-address of the server.
- **Config:** Defines the name of the configuration file.

Optional

- **Port*:** TCP port to connect to (Default: *557*).

Copyright © 2024 Vizrt

- **Encoding:*** Set to *U* or *Unicode* if it should be Unicode.
- **Case*:** Set to *B* to be case sensitive on clip name searches.
- **ClipNames:*** Property for using exact clip names. If set to *EXACT*, clip names are case-sensitive (Default: *EXACT*).
- **ChangeCheckInterval:** Defines the number of seconds between checks for clip changes.
- **LoadDatabase** (boolean):* Set to *true* to build internal database of clips present on server and enable clip monitoring (Default: *false*).

> ⚠ **Note:** The asterisk (*) above denotes the configuration file set property can be overridden by a connection string parameter.

Examples

AV Automation:

```
Name=Harris1; Type=HarrisNexio; Server=192.168.1.1; Port=557;
Config=ClipServerHarris.xml
```

Media Administrator:

```
Name=Harris1; Type=HarrisNexio; Server=192.168.1.1; Port=557; LoadDatabase=true;
Encoding=UTF-8; ClipNames=UNEXACT; ChangeCheckInterval=15;
Config=ClipServerHarris.xml
```

Configuration File Properties

- **ChangeCheckInterval** (integer): Time in seconds how often to poll HarrisNexio for clip changes (Default: *5*).
- **Port** (integer): Defines the default connection port (Default: *557*).
- **Encoding** (string): (Default: *ANSI*).
- **Case** (string): (Default: *UPPER*).
- **ClipNames** (string): Property for using exact clip names. If set to *EXACT*, clip names are case-sensitive (Default: *EXACT*).
- **KeepCache** (boolean): When set to *false*, the cache is cleared on reconnects (Default: *false*).
- **RefreshOnUpdate** (boolean): When set to *true*, the cache is updated on each update clip message from the server (Default: *true*).
- **VerifyClipDelay** (integer): Delay in frames to wait before verifying new clips from server (Default: *0*).
- **SupportSearchUsingObjSlug:** Ignores search for objSlug if set to *false* (Default: *true*).
- **SupportSearchUsingRefID:** Ignores search for refID if set to *false* (Default: *false*).
- **MaxCueTime** (integer): Maximum milliseconds to wait for cue (Default: *5000*).
- **ClipNamePattern:** See General Configuration File Properties.

Copyright © 2024 Vizrt

JupiterWebService Connection String

JupiterWebService is an asset management system.

```
Name=myServerName; Type=Jupiter; Url=myConfigFile
```

Example:

```
Name=Jupiter; Type=Jupiter; Url=ClipServerJupiter.xml
```

This server type does not support any additional parameters in the connection string. The following settings must be added in the configuration file.

Configuration File Properties

- **Server** (string): Defines the hostname or IP-address of the server (Default=localhost).
- **Port** (integer): Defines the TCP port to connect to (Default= `8102`).
- **ServiceName** (string): Default=JupiterService.svc.
- **Service** (string): Default=http://\{Server}:{Port}/{ServiceName}.
- **GetStatusCommand** (string): Default={Service}/clipstatus/{0}.
- **GetKeyFramesCommand** (string): Default={Service}/keyframes/{0}.
- **PollIntervalStatus** (string): Default={Service}/keyframes/{0}.

MVCP Connection String

MVCP (Multiport Video Computer Protocol) is a simple request/response protocol, which is implemented over a TCP byte-stream connection (for example, a stream socket).

Applies To Models

- Vizrt Xlator.
- SVT F2C file name conversion service.

Connection String Properties

- **Type:** MVCP
- **Name:** See Name in *Standard Connection String Properties* above. Defines the ID of this connection (Default: *MVCP*).
- **Server:** Defines the hostname or IP-address of the server (Default: *localhost*).
- **Port:** Defines the TCP port to connect to (Default: *5250*).

Configuration File Properties

- **CustomSearch*:** (Example) Set to a unique string like *SVT* to enable a custom search named *SVT* on *objSlug*.

Copyright © 2024 Vizrt

- **SetNameAsRefId*:** Set clip name as ref ID in the media object (Default: *false*)for re-use in other clip servers (Default: *false*).
- **WriteTimeOut:** Sets the amount of time in milliseconds that a write operation blocks waiting for data from the server (Default: *5000*).
- **ReadTimeOut:** Sets the amount of time in milliseconds that a read operation blocks waiting for data to the server (Default: *5000*).
- **AppendExtension*:** A file extension used when sending messages to the MVCP interface (Default: "").
- **RemoveExtension*:** Used to strip the extension of a clip name from the MVCP interface (Default: "").
- **ClipNamePattern:** See General Configuration File Properties.

> ⚠ **Note:** The asterisk (*) above denotes a custom search called *SVT*, where in this example SVT is set up to process file names with 36 characters. As Nexio only supports 32 characters, *SVT will* expose a mapping proxy with an MVCP interface and a custom command F2C which translates a long name to a shorter one.

Examples

AV Automation:

```
Name=MVCP2; Type=MVCP; Server=192.165.4.1; Port=5250
```

Media Administrator:

```
Name=MVCP2; Type=MVCPSearch; Server=192.165.4.1; Port=5251
```

Omneon Connection String

Applies To Models

- Models with Omneon Spectrum

> ⚠ **Note:** Viz Mosart is incompatible with Omneon Spectrum versions earlier than 4.6. Additionally, in some cases, clip length is not updated when using Omneon Spectrum 6.4.3 or earlier.
> It is recommended to upgrade to Omneon firmware version 7.9.x or later.

Connection String Properties

- **Type:** Connection type: Omneon or OmneonSearch.
- **Name:** See Name in *Standard Connection String Properties* above. ID of this connection.
- **Server:** Hostname or IP-address of the server.

Copyright © 2024 Vizrt

- **Player:** Performs searches associated with directory for the given player if set. If not set, the Directory option should be used to identify the search directory (optional).
- **Directory:** Specifies the search directory (optional).
- **ExtList:** File extensions to be used. List of case sensitive extensions separated by periods.
- **ClipDir:** Clip directory to monitor for files.
- **LoadDatabase** (boolean): Set to true to build internal database of clips present on server and enable clip monitoring (Default: *false*).
- **Config:** Name of the configuration file (*ClipServerOmneon.xml*).
- **UseFullPath:** If *true*, the Media Administrator returns full clip paths. The clip ID contains both the Omneon server directory, as well as the clip name. Setting this property to *true* is mandatory if multiple directories are used on a single Omneon server. (Default: *false*).

Examples

AV Automation:

```
Type=Omneon; Name=Omneon; Server=10.211.114.104; ExtList=.mov.MOV.mxf.MXF;
UseFullPath=true; Config=ClipServerOmneon.xml
```

Media Administrator:

```
Type=OmneonSearch; Name=Omneon; Server=10.211.114.104; Player=Play1; clipdir=/fs0/
media; ExtList=.mov.mxf; LoadDatabase=True; Config=ClipServerOmneon.xml
```

Configuration File Properties

- **KeepCache** (boolean): When set to *false*, the cache is cleared on reconnects (Default: *false*).
- **RefreshOnUpdate** (boolean): When set to *true*, the cache is updated on each update clip message from the server (Default: *true*).
- **VerifyClipDelay** (integer): Delay in frames to wait before verifying new clips from server (Default: *0*).
- **DescriptionPattern** (string): Makes it possible to extract a description from the object slug (objSlug) based on this Regex pattern (Default: *[^_]*_(.*)*).
- **DescriptionReplacement** (string): Regex expression referring to first matched group. Used when replacing object slug (objSlug). (Default: *$1*)
- **InvokeServerOnGetClipInfo** (boolean): If *true* (non-zero), a request is sent to the video server when doing clip search. If *true*, all modified clip events for clip not in cache are verified (as if they were created) (Default: *false*).
- **VerifyModifiedClips** (boolean): If *true*, all modified clip events for clip not in cache are verified (as if they were created) (Default: *false*).
- **IgnoreCase:** If *true*, searching for clips is done case insensitive. Must be used in combination with <MatchExpression ignoreCase=*true* /> from General Search Configuration File Properties .
- **GoToTimeCodeTimeOut** (integer): Maximum time to wait (in milliseconds) for successful execution of command to play clip from a given time code (Default: *6000*).

Copyright © 2024 Vizrt

- **UseNativeGoToTimeCode** (boolean): If *true*, it uses GoToTimeCode method available in Omneon API. This may take a longer time to cue a clip at a specified position (Default: *false*).
- **UseStopOnCue:** If *true*, the timeline is stopped, not paused (play with speed 0) when cueing a new clip. This causes cueing to take longer than usual.
- **ClipNamePattern:** see General Configuration File Properties.

  Search Properties: See General Search Configuration File Properties.

Omnibus OPUS Connection String

```
Name=myServerName; Type=OPUS; Url=myUrl; Config=myConfigFile.xml
```

- **Url:** Specifies the URL of the OPUS Interchange web service.

Orad OCIP Connection String

This connection string is used for video servers controlled by OCIP (Open Control Interface Protocol). This includes Orad VJ (obsolete) and Avid FastServe video servers.

```
Type=Orad OCIP; Name=[Name]; Server=[hostname/IP address]; Port=[Port];
LoadDatabase=true; Config=ClipServerOradOcip.xml
```

Example:

```
Type=Orad OCIP; Name=OradVJ; Server=192.168.15.1; Port=10001; LoadDatabase=true;
Config=ClipServerOradOcip.xml
```

Connection String Parameters

- **Type:** Orad OCIP.
- **Server:** Defines the server hostname.
- **Name:** See Name in *Standard Connection String Properties* above.
- **Port:** Defines the server port. By default, the four ports `10001`, `10002`, `10003`, and `10004` are configured on an OCIP-controlled Orad server. We recommend using different ports for AV Automation and Media Administrator.

Copyright © 2024 Vizrt

> **Example:** *10002* for AV Automation and *10003* for Media Administrator.

- **LoadDatabase:** If *true*, the server database is loaded at start-up (Default: *false*). Use *true* for Media Administrator.
- **Config:** Configuration file containing further configuration settings.

Configuration File Parameters

- **AllowSetPosition:** Whether to enable 'scrubbing'/'scrolling' the in-point of a cued clip in the GUI. Default: *false*. Effective only for servers supporting OCIP 3.0.0 or higher. (Only tested with OCIP 4.0.0.)
- **CopyObjSlug:** Whether the value of the clip 'slug' (the slug is the value of the OCIP item parameter set in ObjSlugParameter) shall be copied 'upwards' (to Media Admin). Default: *false*.

  > The value for **CopyObjSlug** in the installed ClipServerOradOcip.xml file, is initially set to *true*. This will be the value for new installations.

  If problems are experienced when a clip is first deleted and then re-created with the same SourceId (assuming that is the value of the ObjSlugParameter setting), but a different Id, make sure this setting is *true*.
- **IgnoreGrowingItems:** Whether clips with g*rowing* parameter set to *true* will be ignored. When set, this implies that growing clips are processed as non-existent and are unavailable. Available for Avid FastServe running FSP 2018.8 or higher. The RequestGrowing setting must be *true* for this to have effect.
- **LoopPostRoll:** Number of frames before clip-end to restart loop. Set to a positive value to obtain smooth looping.
- **ObjSlugParameter:** The OCIP item parameter to use for clip 'slug'. The *values* of this parameter should match the *values* of the 'clip_hirespath' newsroom tag.
  Default 'Name'. This default value has worked for the video server Orad VJ. Other workflows, especially using Avid FastServe, require using the value 'SourceId'. Other values are not verified as supported.
- **RequestGrowing:** Whether to request the Growing parameter from the server. Default: *false*. Set to *true* if the IgnoreGrowingItems feature is to be used.
- **SetInAndOutPoint:** How a clip's in-point and out-point are set. Available values:
  - **FromServer:** (Default) From the item's In and Out parameters. This may yield incorrect results if the In parameter is different from *0*.
  - **ZeroAndDuration:** (Value in the installed ClipServerOradOcip.xml file, so this will be the value for *new* installations.) As *0* and the item's Duration parameter.
  - **Dont:** For future/internal Vizrt Support use. Do *not* use this value.
- **SkipAbortInLoadClip:** If *true*, skip the ChannelAbort command at cue (Default: *false*).
- **TimeoutDefault:** Timeout in milliseconds for commands with no specific timeout set (Default: *1000* which is 1 second).
- **TimeoutChannelAbort:** Timeout in milliseconds for the ChannelAbort command performed at cue.

Copyright © 2024 Vizrt

- **TimeoutChannelJog:** Timeout in milliseconds for the Jog command performed at 'scrubbing'/'scrolling' (See AllowSetPosition).
- **TimeoutChannelPause:** Timeout in milliseconds for the ChannelPause command.
- **TimeoutChannelPlay:** Timeout in milliseconds for the ChannelPlay command performed at cue.
- **TimeoutChannelPlayInit:** Timeout in milliseconds for the ChannelPlayInit command performed at cue (Default: *4000* (4 seconds)).
- **TimeoutChannelSetSpeed:** Timeout in milliseconds for the ChannelSetSpeed command performed at play.
- **TimeoutChannelStatus:** Timeout in milliseconds for the ChannelStatus command.
- **TimeoutCreateCallbackItem:** Timeout in milliseconds for the CreateCallbackItem command at start-up (Default: *2000* (2 seconds)).
- **TimeoutItemCheck:** Timeout in milliseconds for the ItemCheck command occasionally performed at cue.
- **TimeoutItemGet:** Timeout in milliseconds for the ItemGet command occasionally performed at cue, and also for getting status.
- **TimeoutListChannel:** Timeout in milliseconds for the ListChannel command at start-up.
- **TimeoutListItem:** Timeout in milliseconds for the ListItem command at start-up (Default: *2000* (2 seconds)).
- **UpdateInAndOutPoint:** Whether clip in-point and out-point should be copied 'upwards' (to MediaAdmin) for clips with growing duration.
  Default: *false*. Change to *true* if such clips are not shown with correct duration in GUI.
- **UseHashKey:** If *true*, the item Id will be used as clip 'id'. If *false*, clip 'slug' will be used. (See settings ObjSlugParameter and CopyObjSlug above.)
  Default: *false*. However, the value in the installed ClipServerOradOcip.xml file is *true*, so this will be the value for *new* installations.
  If problems are experienced when a clip is first deleted and then re-created with the same SourceId (assuming that is the value of the ObjSlugParameter setting), but a different Id, make sure this setting is *true*.
- **UsePingOcipOperation (boolean):** If *true*, use an OCIP operation for Ping instead of the default TcpIpClient ping (Default: *false*).

Debug Parameters

- **KeepSpeed:** For Vizrt Support use. Default: *false*.
  Do not change!
- **WSACancelBlockingCallExceptionTraceEventType:** The TraceEventType for the WSACancelBlockingCall exception.
  Options: Critical, Error, Warning, Information, Verbose (Default: *Error*).

Applies To Models

- Orad VJ
- Avid FastServe

Copyright © 2024 Vizrt

Device Prerequisites

- OCIP 1.0.4. Some features require OCIP 3.0.0.
- Avid FastServe. Some features require FSP 2018.8.

Quantel Connection String

Applies To Models

- Connects to Quantel using Quantel CORBA interface.

Device Prerequisites

- Requires Quantel CORBA services to run.
- Serial numbers of playout servers, normally obtained via Quantel Web interface.

Quantel Video Server Connection String

```
Name=myServerName; Type=Quantel; Mode=Player; SerialNo=mySerialNo; IOR=myQuantelIOR;
Slave=mySlaveAddress; Timeout=10; Config=myConfigFile.xml
```

Connection String Properties

- **Name:** See Name in *Standard Connection String Properties* above. A logical name to be associated with the video server. This is used to identify the server across AV Automation and Media Administrator configurations.
- **Type:** Specifies which driver to be used for connecting to the server. This should be set to *Quantel* for both AV Automation and Media Administrator.
- **Mode:** Specifies whether the Quantel connection is used for playing clips or searching (Default: *Search*):
  - **Mode=Player:** The Quantel connection is used for playing clips (AV Automation).
  - **Mode=Search:** The Quantel connection is used for searching (Media Administrator).
- **SerialNo:** Specifies the serial number of the Quantel video server. This number is used to recognize the particular video server within a Quantel zone portal. Serial numbers may be obtained via the Quantel Web interface.
- **IOR:** Specifies the Corba IOR used to establish connection with a Quantel server. This IOR is normally obtained from the system administrator.
- **Slave:** Specifies an optional slave address. Used when Quantel is configured in a redundancy setup.
- **Timeout:** Specifies the timeout in seconds when initially connecting to the Quantel zone portal (ISA manager). Default is *10* seconds.
- **Config:** Specifies a configuration file containing custom configuration for the Quantel server driver. Normally set to *ClipServerQuantel.xml*, which contains the default configuration.

Copyright © 2024 Vizrt

Examples

AV Automation:

```
Name=sQ7; Type=Quantel; SerialNo=19343; Mode=Player; IOR=http://
quantel:@192.168.60.33/ZoneManager.ior; Config=ClipServerQuantel.xml
```

Media Administrator:

```
Name=sQ7; Type=Quantel; SerialNo=19343; Mode=Search; IOR=http://
quantel:@192.168.60.33/ZoneManager.ior; Config=ClipServerQuantel.xm
```

Quantel Configuration File Properties

- **DefaultTakenPortAction** (enum): Action when not able to take a port when owned by someone else (Default: *Steal*).
  Options:
  - **Steal:** Takes control over the port even if in use by someone else.
  - **Share:** Shares control with the other user. Not recommended.
  - **Cancel:** Cancels the operation. Port is still in control by other user.
- **IssueRequestForTakenPorts** (boolean): If *true*, the user is prompted to confirm taking a port in use by someone else (Default: *true*). A dialog box appears in AV Automation, warning about taking a port that is currently in use. Only applicable when DefaultTakenPortAction=*Steal*.
- **ReleasePortsWithNoAssignedChannels** (boolean): If *true*, any unassigned ports (with no assigned channels) are released on player initiation (Default: *true*).
- **ServerPollFrequency** (integer): Polling frequency for server, in seconds. This is used as heartbeat between Viz Mosart and Quantel. Note that heartbeats are not issued if data has been received from Quantel since last heartbeat. (Default: *30*).
- **ClipNameIsValidNumber** (boolean): If *true* then it is possible to search from Viz Mosart GUI using valid Quantel ClipIDs directly (Default: *false*).
- **QuantelSlugColumn** (string): Specifies the Quantel database column that is used for the clip slug (Default: *Title*).
- **QuantelClipIDColumn** (string): Specifies the Quantel database column that is used for the clip ID (Default: *ClipID*).
- **QuantelSearchColumn** (string): Specifies the Quantel database column that is used for the clip searches (Default: *Title*).
- **MaxPingAttempts** (integer): Maximum ping attempts before a lost connection is detected (Default: *1*). When Viz Mosart detects a lost connection, the current connection is closed followed by attempts to reconnect. For playout servers only (AV Automation).
- **QuantelRefIDColumn** (string): Specifies the Quantel database column that is used for the clip RefID. Used if SupportSearchUsingRefID is set to *true*. (Default: *Title*).
- **SearchUsingServerIdOnly** (boolean): Forces AV Automation to only search for clips using Quantel ClipId's obtained from Media Administrator if *true*.
- **ClipNamePattern:** see General Configuration File Properties.

Copyright © 2024 Vizrt

Quantel Failure Handling Properties

- **QuantelStatusInterval** (integer): Interval in seconds for the Quantel server to send regular status messages. These messages are only sent when the Quantel server is idle. (Default: *2* seconds).
- **QuantelCommErrorNumRetries** (integer): The number of retries in case of a Quantel CORBA operation results in a CORBA_COMM_FAILURE. (Default: *1*).
- **QuantelCommErrorTimeout** (integer): The timeout in milliseconds between retries in case of a Quantel CORBA operation results in a CORBA_COMM_FAILURE. (Default: *100* msec).

Quantel Search Filters

It is possible to define a set of dedicated filters used to either exclude or include hits returned from the Quantel ISA manager. Note that the Quantel ISA manager returns hits for all attached pools. It is necessary to filter out the clips not available on the pools used by the corresponding video servers.

The Quantel search filters are defined in the Quantel configuration file as a list of Filter elements within a SearchFilter element. The following XML structure defines the default filters:

```
<SearchFilters>
    <Filter property="Category" pattern="TEMPORARY CLONE" include="false" />
    <Filter property="PoolID" pattern="[PoolID]" include="true" />
    <Filter property="IgnoreCase" pattern="true" />
</SearchFilters>
```

Each filter has the following attributes:

- property (string): Name of a Quantel database column. A special property=`IgnoreCase` is available to specify case insensitive or case sensitive searches.
- pattern (string): Regular expression applied on the resulting property.
- include (boolean): Action to take if the regular expression results in a match:
  - include=*false*, the hit is excluded from the search.
  - include=*true*, the hit is returned as part of the search.
    The default example given above shall be interpreted as follows:
- All clips stored in the Quantel database with Category=*TEMPORARY CLONE* are ignored.
- Only clips in the Quantel database with a PoolID that matches the pool identity of the corresponding video servers are returned.
- All searches are case insensitive.

ScreenLL Connection String

Screen LinguaLib Connection String, used by Media Administrator.

Copyright © 2024 Vizrt

```
Name=myServerName; Type=ScreenLL; Server=myMainServerLocator;
BackupServer=myBackupServerLocator; ServerUsage=myServerUsageMode;
Config=myConfigFile.xml
```

- **Name:** See Name in *Standard Connection String Properties* above.
- **Type=ScreenLL:** Defines the Screen Lingua Lib search driver for clip handling.
- **Server=path:** Defines where subtitle clips are located on the main subtitle server.
- **BackupServer=path:** Defines where subtitle clips are located on the backup subtitle server, if applicable.
- **ServerUsage=mode:** Specifies how availability status is updated based on (main) server and backup server clip status:
  - 0: available if clip is present on main server.
  - 1: available if clip is present on backup server.
  - 2: available if clip is present on either main or backup server.
  - 3: available if clip is present on both main and backup server.
- **Config=ClipServerScreenLL.xml**

SimpleTestPlayer Connection String

The Simple Test Player can be used in Media Administrator for Viz Engine, see the example in section Viz Engine connection string, below.

> ⚠ **Note**: With the exception of testing purposes, this string provides *no* functionality with AV Automation.

The Clip File can contain Clip Element Attributes, often used for development of Viz Mosart features.

The SimpleTestPlayer may be used in either of two modes:

- **File mode**: A specified file contains clip metadata. The file is read at start-up and is monitored for changes.
- **Folder mode**: A specified folder contains actual clip files. The folder is read at start-up and is monitored for changes.

File mode example:

```
Name=STP;Type=SimpleTestPlayer;File=D:
\DemoClip\DemoClip.xml;Config=ClipServerVizEngine.xml
```

Folder mode example:

```
Name=STP;Type=SimpleTestPlayer;Folder=D:\;Config=ClipServerVizEngine.xml
```

Copyright © 2024 Vizrt

Connection String Properties

Type: SimpleTestPlayer
File: The clip metadata file (used in File mode). See below.
Folder: The clip folder (used in Folder mode)

Clip File

The file must be in this format:

```
<ClipDB>
  <clips>
    <clip slug="..." status="0" dur="..."/>
    ...
  </clips>
</ClipDB>
```

Clip Element Attributes

Attribute	Mandatory	Value	Comment
slug	Yes	String	
status	Yes	0	If missing or other value, the clip is considered missing
dur	Yes	Positive integer	Duration in frames. Should be greater than the value of the MinimumClipLength setting.
objid	No	String	
description	No	String	
mark_in	No	String	
mark_out	No	String	
pdur	No	String	
statusdescription	No	String	
rights	No	String	
refid	No	String	

Copyright © 2024 Vizrt

Attribute	Mandatory	Value	Comment
timestamp	No	String	
type	No	String	
MamId	No	String	

Additionally, the ClipDB element *may* have a *serverDown* attribute. This is for testing purposes only, and should *not* be used.

Configuration File Properties (Folder Mode Only)

- **InternalBufferSize**: The size (in bytes) of the internal buffer. For details, see Microsoft's article: InternalBufferSize.
- **IncludeSubdirectories**: Whether subdirectories should be monitored
- **MediaInfoMethod**: May have one of these two values:
  - *MediaInfo* (default): Use MediaInfo to retrieve clip file data
  - *VDCPProxy*: In this case, the 'clip files' are regarded as VDCP proxy files. For further details, contact Viz Mosart Support, quoting ~~MOSART-6143~~.
- **SkipExtension**: Whether the file extension is *not* to be part of the clip slug
- **AlwaysGetDuration**: Whether the actual clip duration is retrieved from the clip file
- **UseObjSlugAsServerId**: Whether to use the slug as the server ID. (Otherwise, the path is used.)
- **UseDurationAsMarkOut**: Whether to set InPoint (=0) and OutPoint (=Duration) from Duration

SQL Database Connection String

This connection string (in Media Administrator) is for the SqlMediaSearch driver, for reading clip metadata from an RDBMS (Relational DataBase Management System) using SQL, and then computing clip properties based on:

- SQL query results.
- Existing clip properties (referred to as properties of 'the clip sent down to the SqlMediaSearch driver').

It is assumed that

- All metadata resides in a single database table (the *clip table*).
- The metadata of a single clip is found in a single row (the *clip row*) in this table.

The driver is designed to be used in conjunction with one or more video servers, to provide additional metadata about clips residing on the server(s). The presence of clip metadata in the database is *not* considered proof of clip existence. Only clips that exist on a video server appear as blue in the viz Mosart GUI.

Place this connection string *above* (for example, in Video clip Server 1) that/those of the video server(s).

Copyright © 2024 Vizrt

Applicable Models

The driver has been designed to interact with any RDBMS (Relational Database Management System) using SQL. However, it has only been verified with following versions of Microsoft SQL Server:

- Microsoft SQL Server 2012 (SP1) - 11.0.3000.0 (X64)
- Microsoft SQL Server Express (64-bit) - 11.0.2218.0

Device Prerequisites

Create a DB user with:

- Read access to the table containing clip data.
- No write or admin access. (As no attempts to prevent SQL injection have been done, this is necessary to prevent unauthorized changes to the DB).

General Format

```
Name=DB;Type=SqlMediaSearch;ProviderName=System.Data.SqlClient;Config=ClipServerSql.xml;DbConnectionString="..."
```

Connection String Parameters

- **Name:** See **Name** in *Standard Connection String Properties* above.
- **Type:** SqlMediaSearch.
- **Config:** See Viz Mosart Video Server Connection String above. A configuration file ClipServerSql.xml is installed. Use the name of this file (*ClipServerSql.xml* unless changed).
- **ProviderName:** The 'invariant name' of a .NET Framework data provider. At least the following providers are installed with the .NET Framework:
  - System.Data.SqlClient
  - System.Data.Odbc
  - System.Data.OleDb
  - System.Data.OracleClient.

  Others *may* have been installed. Only System.Data.SqlClient has been verified.
- **DbConnectionString:** The connection string (containing connection settings) sent to the chosen provider. The format depends on the chosen ProviderName.
  For System.Data.SqlClient (the only verified provider) the format is documented in https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnection.connectionstring(v=vs.110).aspx.
  Quotes in the value have not been tested and may not work as expected.
- **FileClipProperties:** Same as the ClipServerSql.xml parameter. As this is set in *ClipServerSql.xml*, *see* below.
- **FileSelectStatements:** Same as the ClipServerSql.xml parameter. As this is set in *ClipServerSql.xml*, *see* below.
- **LogResult:** Same as the ClipServerSql.xml parameter. As this is set in *ClipServerSql.xml*, *see* below.

Copyright © 2024 Vizrt

- **DefaultTableName:** Same as the ClipServerSql.xml parameter. As this is set in *ClipServerSql.xml, see* below.
- **MonitorFoundClips:** Same as the ClipServerSql.xml parameter. As this is set in *ClipServerSql.xml, see* below.
- **MonitorClipInterval:** Same as the ClipServerSql.xml parameter. As this is set in *ClipServerSql.xml, see* below.

Example

```
Name=DB;Type=SqlMediaSearch;ProviderName=System.Data.SqlClient;Config=ClipServerSql.x
ml;DbConnectionString="Persist Security Info=True;Initial Catalog=HarrisDBTest;Data
Source=BGO-OFRENGINE\SQLEXPRESS;Failover Partner=aut-db-srv03.fs-
pn.vizrtnet.int\harris;User ID=test;Password=test;"
```

Configuration Files

There are *three* configuration files:

- ClipServerSql.xml
- SelectStatements.xml
- ClipProperties.xml

When the Mosart Server is installed, these files are placed in the `ConfigurationFiles` sub-folder of the installation folder. If any file shall be be changed, follow the standard procedure:

- Copy the file to `c:\channeltemplates`, and *change the copy* .

  If you choose to change the name, you must also change (accordingly):
  - If the name of *ClipServerSql.xml* is changed: The Config connection string parameter.
  - If the name of *SelectStatements.xml* is changed: The FileSelectStatements parameter in *ClipServerSql.xml* (below).
  - If the name of *ClipProperties.xml* is changed: The FileClipProperties parameter in *ClipServerSql.xml (below).*

ClipServerSql.xml

This is the main configuration file for the driver. These configuration parameters are available:

- **FileClipProperties:** The name of the file containing clip properties to be pulled from DB. Default: *ClipProperties.xml*, as described below.
- **FileSelectStatements:** The name of the file containing the SELECT statements to be used. Default: *SelectStatements.xml*, as described below.
- **LogResult:** Whether query results should be logged (in Media Administrator console and in Viz Mosart log). Default *true*. Set to *false* if everything works fine, and the messages are superfluous.
- **DefaultTableName:** The name of the outer XML element shown when using LogResult. The default (*dummy*) is usually OK.
- **MonitorFoundClips:** Whether found clips should be monitored. Default *true*. If set to *false*, missing or changed clips are *not* detected.

Copyright © 2024 Vizrt

- **MonitorClipInterval:** If the value is X (and if MonitorFoundClips is *true*), each found clip is re-queried every X seconds. Increase if DB traffic is too high.
- **DbConnectionString:** Same as the connection string parameter; As this is set in the connection string parameters, see above.
- **ProviderName:** Same as the connection string parameter; see above. Should be set in the connection string.

In addition, these 'common'/'standard' parameters are available:

- **SupportSearchUsingObjSlug:** See General Configuration File Properties above.
- **SupportSearchUsingRefID:** See General Configuration File Properties above.

SelectStatements.xml

If the name of this file is changed, the configuration parameter FileSelectStatements in ClipServerSql.xml *must* be changed accordingly.

To further understand this configuration, please study the examples provided in the installed file.

This file defines the SQL SELECT statements that are executed when:

- The DB is queried for data on a single clip.
- The DB is searched for clips satisfying a given search criterion.

Each of these two SQL SELECT statements is defined by a SelectStatement element. For technical reasons, the two SelectStatement elements are enclosed in a Statements element, which in turn is enclosed in a SelectStatements element.

A **SelectStatement** element has these attributes:

Attribute	Value(s)	Description
type	Single/Multiple	Distinguishing between the two types of SQL SELECT statements.
database		The DB name.
schema		The schema name.
table		The table name.

A **SelectStatement** element has these sub-elements:

Sub-element	Description
SelectColumns	The columns in the SELECT clause of the SQL statement.

Copyright © 2024 Vizrt

Sub-element	Description
WhereColumn	The WHERE clause column to be searched for the (variable) search criterion.
AdditionalWhereColumns	Additional WHERE clause columns being searched for *constant* values.

The **SelectColumns** element has one or more SelectColumn sub-elements, each with these attributes (and no sub-elements):

Attribute	Value(s)	Description
name		The DB column name
type	String/Int	The DB column data type. For Microsoft SQL Server, · String is used for char and varchar columns. · Int is used for int columns.
format	Default/ HhMmSsFfPackedBCD	How a column value is to be interpreted: · Default: The value is passed as is. (And this *is* default, so it is used if no format attribute is present.) · HhMmSsFfPackedBCD: The (four-byte) int value is interpreted as the packed BCD representation of a time code hh:mm:ss:ff. This is converted to the total number of frames. E.g., the int (decimal) 404310278 is hex 18194906, corresponding to the time code 18:19:49:06, in total 1649731 frames.

The **AdditionalWhereColumns** element has one or more **WhereColumn** sub-elements.

A WhereColumn element (either as a direct sub-element of SelectStatement, or as a sub-element of AdditionalWhereColumns) has these attributes:

Attribute	Value(s)	Description	Comment
name		The DB column name	

Copyright © 2024 Vizrt

Attribute	Value(s)	Description	Comment
type	String/Int	The DB column data type. For Microsoft SQL Server, · String is used for char and varchar columns. · Int is used for int columns.	Only *String* has been tested.
value		The value to search for	Used only for WhereColumn elements which are sub-elements of AdditionalWhereColumns. For the WhereColumn sub-element of SelectStatement, the value to search for is given by context.
searchMethod	BeginsWith/ IsEqualTo/EndsWith/ Contains	How the column value (c) is compared to the value to search for (s): · IsEqualTo: c must be exactly equal to s · BeginsWith: c must begin with s · EndsWith: c must end with s · Contains: c must contain s	Only IsEqualTo makes sense for Int. EndsWith and Contains have *not* been tested. IsEqualTo has not been tested for Int.

ClipProperties.xml

If the name of this file is changed, the configuration parameter FileClipProperties in ClipServerSql.xml *must* be changed accordingly.

To further understand this configuration, please study the examples provided in the installed file.

This file defines the clip properties computed by the driver. Each of these properties is defined by a Property element. For technical reasons, the Property elements are enclosed in a Properties element, which in turn is enclosed in a ClipProperties element.

A **Property** element has one attribute:

Copyright © 2024 Vizrt

Attribute	Description	Comment
name	The property name	Only these have been implemented: · Description (the clip description as shown in the GUI (if selected by the setting Video clips should be presented with there)) · InPoint (the clip in point as shown in the GUI Preview window) · ObjSlug (the clip slug as shown in the GUI Media pool Search tab) · OutPoint (the clip out point as shown in the GUI Preview window)

A **Property** element has one sub-element, an *expression*. The principal feature of an expression is that it may be evaluated to yield a value. There are four kinds of expression elements (so a Property element has one of these as a sub-element):

Sub-element	Description	Comment
ClipExpression	The value of a ClipExpression depends only on a property of the clip as sent down to the SqlMediaSearch driver.	The clip properties available depend on the video server driver(s) used together with SqlMediaSearch. Only the BasePoint property (start timecode in frames, available for the HarrisNexio driver) has been tested.
DbExpression	The value of a DbExpression depends only on the value of a column in the clip row (the DB row pertaining to the clip).	The column has to be defined by a SelectColumn element as described in the **SelectStatements.xml** section above.
ConstantExpression	The value of a ConstantExpression is (as strongly indicated by the name) constant , it depends neither on the clip as sent down to the SqlMediaSearch driver, nor on the DB.	ConstantExpressions were introduced for internal purposes, and have not been real-life tested.

Copyright © 2024 Vizrt

Sub-element	Description	Comment
OperationExpression	An OperationExpression is used to combine other (simpler) expressions by performing an operation, which means applying an *operator* to the values of its *operands* (sub-expressions). Each operand/sub-expression may be of any of the four kinds of expression listed in this table. In particular, OperationExpressions may be nested to any depth.	

A **ConstantExpression** element has one attribute:

Attribute	Description
value	The (constant) value of the ConstantExpression

Both a **ClipExpression** and a **DbExpression** has two attributes:

Attribute	Description
name	See table below
default	The value to be used if evaluating the expression as described in the table below doesn't make sense in the given context

The **name** attribute and expression evaluation is described in more detail for each of the two kinds of expressions:

Expression kind	Description of the name attribute	Expression value	Comment
ClipExpression	The name of a clip property	The value of the property for the given clip	Only the BasePoint property (start timecode in frames, available for the HarrisNexio driver) has been tested.
DbExpression	The name of a column in the DB clip table	The value of the column for the clip row	The column has to be defined by a SelectColumn element as described in the **SelectStatements.xml** section above.

Copyright © 2024 Vizrt

An **OperationExpression** element has one attribute:

Attribute	Value(s)	Description	Comment
operator	Minus/Plus	The operator to be applied to the values of the operands. Details are given in the table below.	Both operators make sense for numerical arguments only, and have been implemented for integer arguments only.

An **OperationExpression** element has one sub-element: Operands. An Operands element has no attributes, but any number of expression sub-elements, meaning any number of ClipExpression, DbExpression, ConstantExpression, and OperationExpression sub-elements.

These sub-elements are the operands / sub-expressions of the OperationExpression.

Some further details on each operator:

Operator	Expression value
Minus	If there is one operand, the negation of the value of that operand. If there are two (or more) operands, the difference between the values of the first operand (minuend) and the second operand (subtrahend).
Plus	For any number of operands, the sum of the values of the operands. (As a special case, the 'sum' of no operands is zero. However, the Operands sub-element itself must be present. so <ConstantExpression value= `0` /> is convenient.)

VDCP Connection String

The VDCP driver supports both serial and TCP as transport. Video Disk Communications Protocol (VDCP) is also known as Louth protocol. The driver can be configured to run in eight character or variable character mode. (See the UseLongClipNames setting below.)

> **Note**: Standard VDCP does not support *looping* of clips. For a very limited number of servers, a customized solution has been introduced to support looping over VDCP, if described in the server model's configurations.

Applies To Models

- Dalet BRIO
- Dayang VIPS-2H-4XHD
- EVS XT-VIA
- Ross Blackstorm
- Vector MultiPlay VServer

Copyright © 2024 Vizrt

Device Prerequisites - VDCP

- Dalet BRIO
  - Single VDCP connection configuration per port - does not support multiple control ports per connection.
  - Only clips in folders added to the BRIO VDCP configuration are visible for Viz Mosart to monitor and play-out.
  - Duration of video files is only available if the VDCP connection is configured to use a physical play-out port.
- Dayang VIPS-2H-4XHD
  - For Media Administrator to function correctly, the Dayang server must be configured so that a dedicated signal port is available for media search. Viz Mosart Support does not have details about this procedure and recommends consulting Dayang.

Connection String Properties - VDCP

- **Type:** VDCP (serial) or *VDCPtcp* (Ethernet).
  For Dayang, only VDCP (serial) has been tested.
- **Name:** See **Name** in *Standard Connection String Properties* above.
- **Server:** Hostname or IP-address of the server (for Type=*VDCPtcp* only).
- **Port:**
  - Serial port (Type=VDCP, Default: *COM1*)
  - Server side TCP port to connect to (Type=VDCPtcp, Default: *5205*).
- **LoadDatabase:** If *true*, builds internal database of clips present on server and enables clip monitoring (Default: *false*).
- **Config:** Name of the configuration file. At installation the file `ClipServerVDCP.xml` is copied to the ConfigurationFiles subfolder of the installation folder. Unless you *rename* this file, use Config=*ClipServerVDCP.xml*. (You may need to copy and rename if you have several VDCP servers which should be configured differently.) This file may be taken as a starting point for custom configuration, See Configuration File Properties - VDCP below.
- **DisableHeartbeat (boolean):** Whether the 'Ping' operation (see NextPingDelay above) is just a simple connection check. If *false*, the Ping operation sends a command to the server. For some servers, this is necessary to prevent the server from closing the connection. This applies to at least the Ross Blackstorm server.
- **NoOfPlayers**: The number of signal ports on the server to be made available to the program (AV Automation or Media Administrator). (The ports are numbered consecutively from 1 upwards.) Default 5. (So then ports 1-5 are available.) Use a larger value if you need to control a signal port with a number greater than 5.
- (MediaAdministrator and Dayang only) **PlayerToOpen**: The number of the dedicated signal port available for media search. If this number is greater than 5, the NoOfPlayers must be set to at least this number.

> ⚠ **Note**: There have been known issues with this vendor's implementation of the VDCP protocol. Vizrt recommends that you consult Dayang if there are issues where the video server seems to require an open port to respond to ID List command.

Copyright © 2024 Vizrt

Examples

AV Automation: **VDCP (serial)**

```
Type=VDCP; Name=Server1; Port=COM1; Config=ClipServerVDCP.xml
```

AV Automation: **VDCPtcp (Ethernet)**

```
Type=VDCPtcp; Name=Server1; Server=192.168.42.42; Port=10001;
Config=ClipServerVDCP.xml
```

AV Automation: **Dayang**

```
Name=DY1;Type=VDCP;Port=COM4
```

Please see the Note below.

Media Administrator: **VDCP (serial)**

```
Type=VDCP; Name=Server1; Port=COM2; LoadDatabase=true; Config=ClipServerVDCP.xml
```

Media Administrator: **VDCPtcp (Ethernet)**

```
Type=VDCPtcp; Name=Server1; Server=192.168.42.42; Port=10002; LoadDatabase=true;
Config=ClipServerVDCP.xml
```

Media Administrator: **Dayang**

```
Name=DY1;Type=VDCP;Port=COM3;Config=ClipServerVDCP.xml;LoadDatabase=True;NoOfPlayers=
101;PlayerToOpen=101
```

⚠ **Note**: For Dayang, different COM ports should be used for AV Automation and Media Administrator

Configuration File Properties – VDCP

- **DefaultRecordingLength (integer):** Default duration of recorded video files in frames *(*Default:*1000).*
- **DelayGetClipDuration (Boolean):** Enable to build the initial playlist without getting the file duration (Default: f*alse*).
- **IgnoreClipExistence (Boolean):** Enable to ignore requesting clip presence on the server. This should be set to *false* in the configuration file used by AV Automation (Default: *true*).

Copyright © 2024 Vizrt

- **LogMessages (Boolean):** If *true*, log any messages that are sent and received. Verbose logging must be activated for this to have effect (Default: *false*).
- **MaxClipList (integer):** The maximum amount of clips to hold in cache. When this limit is reached, there are no more clips added to the cache. Setting this to a low value reduces load time, but might cause invalid clip status if non-cached clips are in the rundown.
- **ClipListChangesInterval:** The interval, in seconds, to poll for clip list changes (added/deleted clips) on the Media Admin server (Default=*30*). See also UseAddedClipsCommand.
  Requires LoadDatabase=*true* (set in connection string) and UseAddedClipsCommand=*true*.
- **DelayUpdateClipsMsec:** Delay, in milliseconds when to trigger a clip update after retrieval of clips. Used when obtaining multiple clips when loading and updating the database (Default=*5000*).
  Requires LoadDatabase=*true* (set in connection string) and DelayGetClipDuration=*false*.
- **DelayLoadDatabaseMsec:** Delay in milliseconds for delaying loading the database after connecting to the server (Default=*5000*).
  Requires LoadDatabase=true (set in connection string).
- **RequestPortStatus123 (Boolean):** Whether to request more status from the VDCP server. May safely be left at *false* for normal production. Change to *true* for debugging purposes if advised to by Vizrt Support.
- **SendStillWhenCueDone (Boolean):** When the server has cued a clip, whether to send a command making sure that the first frame is shown on the server output. Change to *true* if supposedly cued clips are not shown on the server output.
- **SendStopBeforeCueing:** Whether a Stop command is always sent before cueing a clip. Change to *true* if cueing problems are encountered, in particular when cueing/playing adlibs back-to-back (Default=*false*).
- **StopShouldCheckForIdle (Boolean):** Whether stopping a clip should make sure that the port is idle. Change to *true* if cueing problems are encountered.
- **TimeoutPlay (integer)**: The time (in ms) for the server to acknowledge that it is playing the cued clip. Used only if VerifyPlay is *true* (or if EnableLoopingDaletBrio is *true*, and the cued clip should loop).
- **UseAddedClipsCommand (Boolean):** Enable to retrieve a list of new files added to the video server.
- **UseFixedPortStatusBitMap (Boolean):** Whether to force Mosart to use a fixed bit map when requesting port status. Change to *true* for new installations. In general, CHANGE the value if problems are encountered, in particular problems related to cueing.
- **UseInternalPlayerState (Boolean):** Enable to ignore requesting presence of the clip on the play-out device using the ID REQUEST message.
- **UseLongClipNames (Boolean):** If *true*, use messages supporting variable length file names (maximum length 80 characters). If *false*, use fixed eight character files names padded with spaces if less than eight characters.
- **VerifyPlay (Boolean):** Whether to verify that the server is actually playing.
- **WaitTimeoutPlay (Boolean):** The time (in ms) to wait between each server playing status polling. Used only if VerifyPlay is *true* (or if EnableLoopingDaletBrio is *true*, and the cued clip should loop).

Copyright © 2024 Vizrt

Configuration File Properties – Dalet BRIO Only

The following properties are used for the looping functionality of the Dalet BRIO:

- **EnableLoopingDaletBrio (Boolean):** Whether to enable the looping mechanism for Dalet Brio. Set to *true* only if this file is for a Dalet BRIO, and you want looping (Default: *false)*.
- **TimeoutDaletBrioLoop (Boolean):** The time (in ms) to wait for the server to acknowledge that looping has been turned on/off.
- **WaitTimeoutDaletBrioLoop (Boolean):** The time (in ms) to wait between each server looping status polling.

Viz Engine Connection String

Viz Engine Connection String, used by Media Administrator and AV Automation.

Connection String Properties

- **Type:**
  For AV Automation: *VizEngine*.
  For Media Administrator: *SimpleTestPlayer*.
- **Name:** See Name in *Standard Connection String Properties* above. Defines the ID of this connection.
- **Server:** Defines the hostname or IP-address of the server (for Type=*VizEngine* only).
- **Port:** Defines the port.
- **Config:** Defines the name of the configuration file.
- **Folder:** Location of the Viz Mosart configuration files. Either in `%ProgramFiles(x86)%\Mosart Medialab\Mosart Server\ConfigurationFiles` or with the video configurations in `C:\channeltemplates`.

Examples

Connection string for Integrated Engine (Viz Engine connection in AV Automation):

`Type=VizEngine;Server=localhost;Config=VizEngineSwitcherConfig.xml (Port=6100 is optional)`

Connection string for Video Server (Viz Engine connection in AV Automation):

`Name=Vizrt;Type=VizEngine;Server=localhost;Port=6100;Config=ClipServerVizEngine.xml`

Connection string for Media Administrator:

`Name=VizEngine;Type=SimpleTestPlayer;Folder=D:\;SkipExtension=true;Config=ClipServerVizEngine.xml`

Configuration File Properties

- **AlwaysGetDuration** (Default: *false*):

Copyright © 2024 Vizrt

- When *false*, only files in the active Viz Mosart rundown get the actual video duration.
- When *true*, get the video duration from all files in the monitored folder. Note that setting this value to *true* impacts the startup time of the SimpleTestPlayer driver in Media Administrator for folders with a large number of video files.

Viz One and Media Service Connection String

This connection string is used by Media Administrator for Viz One and Media Service.

Viz Mosart can search for clips in Viz One and initiate transfer of Viz One clips to a specified playout storage. The driver uses the REST-based Viz One Third party API to execute tasks.

Device Prerequisites (Viz One)

For further details, please consult the *Viz One Administrator Guide*.

- At least one Viz One *user* at least one *publishing point* must be configured.

Connection String Properties

Naming	Examples	Description
Name	VizOne VizEngine MediaSearch	Name of connection. Must be same as Name of the video server in AV Automation. See Name in *Standard Connection String Properties* above.
Type	VizOne	Type of connection. Must be *VizOne* to use the Viz One Clip Administrators.
UserName	vizmse	Viz One only Username for user that connects to third party API. Please look at separate section in this document on how to create a dedicated user for Viz Mosart to access Viz One. *admin* as username and password may be used for testing, but should *not* be set in production.
Password	vizmse	Viz One only Password for user that connects to third party API
Server	10.211.112.213 bgoqavizone	IP address or host name of Viz One server

Copyright © 2024 Vizrt

Naming	Examples	Description
PublishingPoints	vve2-1 vizengine-1 MediaService	A publishing point is a playout storage defined in Viz One. This can be accessed through Administration menu on the Web interface. Comma separated list of publishing point identifiers where the clip administrator should ensure assets are present. If *MediaService* is specified for PublishingPoints, Media Service and not Viz One is used to search for clips.
ValidDuration	26	Viz One only Duration for which an asset should be present on the publishing point from time of transfer. **Value is in hours**. Default: 24
AutoTransfer	true false	Viz One only *true*: Any assets in the playlist that is present in Viz One, but not on the publishing points will be transferred. *false*: Assets are marked as not present. *true* is default value.
PublishingPointAsObjId	true false	*true*: An ObjectId will be set for each publishing point as well as Name from connection string (an additional server ID is set for each publishing point). *false*: Use Name from connection string as ObjId (only one server ID is set: for Name). *false* is default and most common.
Port	21099	Media Service only MediaService requires port to be specified. Default port is *21099* for MediaService. See also *Configure Media Service* section in the Media Service guide. (Viz One does not require this field).
Config	ClipServerVizOne.xml	Name of configuration file located in ChannelTemplates folder. See also Standard Video Server Connection String.

Copyright © 2024 Vizrt

Example (Media Service)

```
Name=VizEngine;Type=VizOne;Server=localhost;Port=21099;PublishingPoints=MEDIASERVICE;Conf
ig=ClipServerVizOne.xml
```

Example (Viz One)

```
Name=VizOne;Type=VizOne;UserName=username;Password=password;Server=10.211.112.21
3;PublishingPoints=vizengine-1;ValidDuration=26;AutoTransfer=true;PublishingPoin
tAsObjId=false;Config=ClipServerVizOne.xml
```

Note: The differences between ClipServerVizEngine.xml and ClipServerVizOne.xml is that the first one handles video server in configuration Av Automation and second is only for establishing clip searches in Media Administrator.

Copyright © 2024 Vizrt

4.6.2 Video Wall Connection Strings

Video wall connection strings are used by AV Automation. This section contains Connection String types.

> **Note:** For general information on connection strings, see Video Server, MAM and Database Connection Strings.

Watchout Connection String

This connection string is used for WATCHOUT video walls.

Modes of Operation

The driver may operate in any of three different *modes*. (For the sake of simplicity, assume that at least two commands are to be performed in one operation).

- **default:** A driver operation sends the commands one after the other. It waits for WATCHOUT to respond finally (or time-out) before sending the next command. If a command fails, the rest is skipped. The advantage is that when the driver operation finishes, the entire operation is complete (or has failed). However, as driver calls may arrive in different threads, and therefore simultaneously or overlapping, problems may occur. (The driver is made for performing (sending and feedback-waiting) one command at a time.)
- **SequenceRecallAndMixes:** A driver operation *queues* the commands to be performed as a unit (without interference from other such units), possibly at a later time. The problem mentioned above is solved, however at the cost of the driver operation possibly finishing before the commands are complete.
- **SequenceIndividualCommands:** A driver operation *queues* the *first* command (to be performed without interference from other commands). If successful, the *next* command is queued, and so on. This *may* be useful *if* WATCHOUT is able to keep several shows loaded at the same time (and is able to use the run timeline argument to choose between them).

Examples

AV Automation Connection String:

```
AuthenticateTimeout=[authenticateTimeout];BufferSize=[bufferSize];BusyNonCommandTimeo
ut=[busyNonCommandTimeout];BusyTimeout=[busyTimeout];GetStatusThreshold=[getStatusThr
eshold];LoadTimeout=[loadTimeout];PingInterval=[pingInterval];PingTimeout=[pingTimeou
t];Port=[port];ReconnectInterval=[reconnectInterval];ResetAnyway=[resetAnyway];ResetT
imeout=[resetTimeout];RunTimeout=[runTimeout];SequenceIndividualCommands=[sequenceInd
ividualCommands];SequenceRecallAndMixes=[sequenceRecallAndMixes];Server=[server];UseG
etStatus=[useGetStatus]
```

Example:

Copyright © 2024 Vizrt

```
Server=192.168.97.129;AuthenticateTimeout=200;RunTimeout=40
```

Connection String Properties - Watchout

String	Type	Description	Default
AuthenticateTimeout	integer>0	The time (in ms) allowed for WATCHOUT to process an authenticate command and respond with Ready.	`2000`
BufferSize	integer>0	The maximum number of bytes to receive from WATCHOUT at a time. Should be set at least as large as the longest conceivable WATCHOUT response.	`1024`
BusyNonCommandTimeout	integer>0	The time (in ms) allowed for WATCHOUT to return to normal operation after a non-command Busy.	`10000`
BusyTimeout	integer>0	The time (in ms) allowed for WATCHOUT to increase the % of otherwise equivalent Busy responses.	`60000`
GetStatusThreshold	integer	The highest acceptable value of the third parameter of the Reply feedback to the getStatus command General health status of the cluster: · 0: OK · 1: Suboptimal · 2: Problems · 3: Dead	`1` (Suboptimal)
LoadTimeout	integer>0	The time (in ms) allowed for WATCHOUT to process a load command and respond with Ready.	`10000`
PingInterval	integer>0	The time (in ms) between ping commands sent to WATCHOUT.	`10000`

Copyright © 2024 Vizrt

String	Type	Description	Default
PingTimeout	integer>0	The time (in ms) allowed for WATCHOUT to process a ping command and respond with Ready.	`10000`
Port	integer>0	The port to connect to.	`3039`
ResetAnyway	true/false	Whether reset commands are always sent before run commands even if not present in the RecallAndMix argument.	false
ResetTimeout	integer>0	The time (in ms) allowed for WATCHOUT to process a reset command and respond with Ready.	`10000`
RunTimeout	integer>0	The time (in ms) allowed for WATCHOUT to process a run command and respond with Ready.	`10000`
ReconnectInterval	integer>0	The amount of time (in ms) to wait to reconnect after a failed connect or authenticate.	`10000`
SequenceIndividualCommands	true/false	Whether the driver should operate in that mode. Has no effect when SequenceRecallAndMixes=true.	false
SequenceRecallAndMixes	true/false	Whether the driver should operate in that mode.	false
Server	string	The DNS name of WATCHOUT.	localhost
UseGetStatus	true/false	Whether the Ping operation should use the getStatus command instead of the ping command.	false

Copyright © 2024 Vizrt

4.6.3 Device Configuration Files

The device configuration files are located in the following directory: *%ProgramFiles(x86)%\Mosart Medialab\Mosart Server\ConfigurationFiles*.

> **Note:** Do not modify any files at this location, as they are overwritten by the next installation. Copy any files that must be modified to the "`%ProgramData%\Mosart Medialab\ConfigurationFiles`" directory, and modify them there. This ensures that configurations remain valid after system upgrades or downgrades. Note that historically, `C:\channeltemplates` has been the folder to copy to, but with a few exceptions, Viz Mosart applications will look in `%ProgramData%` before checking `C:\channeltemplates` for configuration files.

> **Name of the configuration file(s)**
>
> In the following sections, the configuration files are shown with their *default* names. For some device types it is possible to specify a different configuration file name in the device connection string.
> This makes it possible to use different configurations for various devices of the same type, for example when you have two slightly different settings for two video servers of the same type and brand.
> There **must** be a 1:1 match between the name of the *configuration file* and the name referred to in the device *connection string*.

This section contains the following topics:

- Robotic Camera Configuration Files
- Graphics Configuration Files
- Subtitling Configuration Files
- Video Router Configuration Files
- Audio Mixer Configuration Files
- Vision Mixer Configuration Files
- Engine Switcher Configuration

Information about *Video Server Configuration Files* is in section, Video Server, MAM and Database Connection Strings.

Copyright © 2024 Vizrt

Robotic Camera Configuration Files

This section contains the Robotic Camera types:

- Shotoku
- FxMotion
- Vinten 200 Radamec
- Cambotics Configuration File
- Cinneo Configuration File
- Technodolly Configuration File
- Panasonic (serial) Configuration File
- Panasonic TCP Configuration File
- Electric Friends Configuration File

All configuration files are by default stored at `C:\Program Files (x86)\Mosart Medialab\Mosart Server\ConfigurationFiles\`. Please refer to the Note in Device Configuration Files for information on copying the files before changing.

Shotoku

Specific Shotoku parameters are found in `RoboticCameraConfiguration.xml`.

You can control behavior of the connection to Shotoku when switching Viz Mosart from main to backup, or at an AVA restart. Ensure the following properties are managed in the configuration file:

- **ShotokuForwardStandbyCommands** (boolean) - If set to true, standby state will be forwared to Shotoku controller. Default *false*.
- **ShotokuDisableHeartbeat** (boolean) - If set to true, the heartbeat mechanism is disabled. Default *true*.
- **ShotokuDisconnectInStandby** (boolean) - If set to true, disconnect controller when in standby. Default *false*.

The above settings govern the two situations:

1. Whether the sending of a disable command (when AVA restarts or when the active Mosart server is switched) to the Shotoku controller will block all automation commands until an enable command is received.
2. Whether instead of timing-out, Mosart shall stay connect to the Shotoku controller when in standby/idle.

In addition, the `HeartbeatInterval` property also applies to Shotoku. Refer to the configuration for file for information.

FxMotion

Specific FxMotion parameters are found in `RoboticCameraConfiguration.xml`. Refer to the configuration for file for information.

- HeartbeatInterval

Copyright © 2024 Vizrt

- FxMotionReceiveTimeout

Vinten 200 Radamec

Specific Vinten 200 Radamec parameters are found in `RoboticCameraConfiguration.xml`. Refer to the configuration for file for information.

- HeartbeatInterval
- Vinten200ShowFilterValue
- Vinten200DisconnectInStandby
- Vinten200MinSendInterval

Cambotics Configuration File

The integration between Viz Mosart and Cambotics Camera Robotics Control Systems supports moving to a stored position. Camera Robotics control is handled by AV Automation.

For details on the Cambotics (Ross CamBot) Protocol, refer to the product documentation written by Ross Video (www.rossvideo.com).

The Cambotics configuration file is located in the program-folder under `\Mosart Medialab\Mosart server\ConfigurationFiles\CamboticsConfiguration.xml`.

The following configurations are available:

- **Heartbeat:** (keep connection alive) interval: Default `2000` ms.
- **Connection timeout:** Default `1500` ms.
- **ConnectionAttempts:** Default `5` tries.
- **ConnectionDelay:** Default `60000` ms.

> ⚠ **Note:** To integrate with Cambotics (Ross Cambot) Camera Robotics, you must also configure it in Camera Robotics. Select the router protocol CAMBOTICS, set the IP address of the camera controller, and the Port number (default is set to `2050`).

> ⚠ **Note:** For more information on setting up templates for Cambotics, see the section on *Robotic Camera Control* in the *Viz Mosart User Guide*.

Cinneo Configuration File

Cinneo configuration is contained within the general device configuration file *DeviceConfig.xml*.

```
<item name="cinneoSceneFileTimeout" value="30000" />
```

- **cinneoSceneFileTimeout:** The value is given in milliseconds.

Copyright © 2024 Vizrt

Technodolly Configuration File

The Technodolly configuration file is named *TechnodollyConfig.xml*.

```
<DeviceConfig name="TechnodollyConfiguration" connectionString="localhost">
   <Properties>
      <!-- Connection parameters, may be overridden -->
      <item name="Port" value="15243" />
      <item name="HeartbeatInterval" value="20" />
      <item name="MaxQueueLength" value="2" />
      <item name="ProtectInPreview" value="cut,move" />
      <item name="ProtectInProgram" value="-" />
   </Properties>
</DeviceConfig>
```

- **Port:** Contains the default value, it is not used.
- **HeartbeatInterval:** Defines how often Viz Mosart should check if the connection to Technodolly is OK. The value is given in seconds.
- **MaxQueueLength:** Restricts queuing of commands to Technodolly. A value of `1` means only one prepare or one run. A value of `2` means one prepare and/or one run. A value greater than `2` means no queuing restrictions.
- **ProtectInPreview:** Defines which commands should not be sent to Technodolly in preview when a mixer cross-point is protected. Valid values are "-" for none, "cut" for prepare, "move" for run, and "cut,move" for both prepare and run.
- **ProtectInProgram:** Defines which commands should not be sent to Technodolly in program when a mixer cross-point is protected. The values have the same meaning as for *ProtectInPreview*.

Panasonic (serial) Configuration File

The Panasonic configuration file (used for PANASONIC and PANASONICNEW) is named *PanasonicConfiguration.xml*.

```
<?xml version="1.0" encoding="utf-8" ?>
<DeviceConfig name="PanasonicConfiguration">
    <Properties>
        <!-- Connection parameters, may be overridden -->
        <item name="HeartbeatInterval" value="2000" />
        <item name="DisableHeartbeat" value="false" />
    </Properties>
</DeviceConfig>
```

- **HeartbeatInterval:** Time between heartbeats, in milliseconds. Integer >= 0. Default = `2`.
- **DisableHeartbeat:** If false, heartbeats are sent. This checks the connection and sends a keep alive message to the controller. If true, no heartbeats are sent. Default = False.

Copyright © 2024 Vizrt

Panasonic TCP Configuration File

The Panasonic TCP configuration file is named *PanasonicTcpConfiguration.xml*.

```
<?xml version="1.0" encoding="utf-8" ?>
<DeviceConfig name="PanasonicConfiguration" connectionString="localhost">
  <Properties>
    <!-- Connection parameters, may be overridden -->

    <!--Whether to disable pings. If true, pings are still sent if no response is
received for a request.-->
    <item name="DisablePing" value="false" />

    <!--The path component of the Request-URI. Do not change.-->
    <item name="Path" value="cgi-bin/aw_cam" />

    <!--Time between pings in ms-->
    <item name="PingInterval" value="2000" />

    <!--In case of connection loss, a ping will be sent after this many ms-->
    <item name="ReconnectInterval" value="1000" />

    <!--The number of ms to wait for a response for a Ping command-->
    <item name="TimeoutPing" value="500"/>

    <!--The number of ms to wait for a response for a SelectPort command-->
    <item name="TimeoutSelectPort" value="10"/>

    <!--The number of ms to wait for a response for a PlayPreset command-->
    <item name="TimeoutPlayPreset" value="10"/>
  </Properties>
</DeviceConfig>
```

- **DisablePing**: Whether to disable pings. If true, pings are still sent if no response is received for a request.
- **Path**: The path component of the Request-URI. Do not change.
- **PingInterval**: Time between pings in ms.
- **ReconnectInterval**: In case of connection loss, a ping will be sent after this many ms.
- **TimeoutPing**: The number of ms to wait for a response for a Ping command.
- **TimeoutSelectPort**: The number of ms to wait for a response for a SelectPort command.
- **TimeoutPlayPreset**: The number of ms to wait for a response for a PlayPreset command.

Electric Friends Configuration File

The configuration file for Electric Friends should be saved as *ElectricFriendsConfiguration.xml* normally stored in *C:\ChannelTemplates*

Copyright © 2024 Vizrt

Note: For more information on setting up templates in AV Automation, see the section on *Robotic Camera Control* in the *Viz Mosart User Guide*.

Example configuration:

```
{{
<?xml version="1.0" encoding="UTF\-8"?>}}{{
<DeviceConfig name="ElectricFriendsConfiguration" connectionString="localhost">}} {{
    <Properties>}} {{
        <!\-- Connection parameters, may be overridden \-->}} {{
        <item name="HeartbeatInterval" value="2000" />}} {{
        <!\-- Time between heartbeats in ms \-->}} {{
        <item name="PollDataInterval" value="10000" />}} {{
        <!\-- Time between renewing list of shows, cameras and shots, in ms \-->}} {{
        <item name="UseJsonWorkaround" value="true" />}} {{
        <!\-- Early development. Timed move command needs a workaround for correct
response to server. Please let the value remain true \-->}} {{
        <item name="PollRunningStatus" value="200" />}} {{
        <!\-- Time between polls in ms. This will check to see if camera movement is
finished before starting a new move \-->}} {{
        <item name="PollTimeout" value="5000" />}} {{
        <!\-- Time the poll is allowed to run before ignoring the play command, in ms
\-->}} {{
        <item name="DefaultPort" value="9000" />}} {{
        <!\-- If no port is set, use this value. The ElectricFriends default port is
9000 \-->}} {{
    </Properties>}}{{
</DeviceConfig>}}
```

Copyright © 2024 Vizrt

Graphics Configuration Files

This section contains the graphics type:

- Pixel Power Control Center (PPCC) Configuration File
- Pixel Power (Clarity) Configuration File
- Vizrt Graphics Configuration File
- Vizrt Media Sequencer VDom Logic Macros
- Chyron Configuration File
- Deko, Orad, Xpression Configuration File

Pixel Power Control Center (PPCC) Configuration File

The PPCC configuration file is called *PPCCConfiguration.xml*

```
<?xml version="1.0" encoding="utf-8" ?>
<DeviceConfig name="PixelPowerConfiguration" connectionString="localhost:0">
  <Properties>
    <!-- Default channel nubmer to use -->
    <item name="SystemNumber" value="1"/>
    <item name="ClientApplication" value="Overlay Graphics"/>
    <item name="ClientName" value="Vizrt Mosart"/>
    <item name="DefaultOverlaysTriggerCount" value="2"/>
    <item name="EnableOutputBlackOnTakeOut" value="true"/>
    <item name="PreventPreloadOverlayIfItemsAreOnAir" value="true"/>
    <item name="RundownPrefix" value=""/>
    <item name="ConvertStoryIdToDecimal" value="false"/>
    <item name="ConnectedWhenIdle" value="false" />
    <item name="ConnectedWhenInStandBy" value="false" />
  </Properties>
</DeviceConfig>
```

Properties

- **DefaultOverlaysTriggerCount:** Number of triggers used for take-in and take-out from PPCC in overlays.
- **EnableOutputBlackOnTakeOut:** True if OutputBlack should be sent when continue points still exist on item.
- **PreventPreloadOverlayIfItemsAreOnAir:** Will prevent selecte page of next overlay if the engine currently has items on air (only overlays).
- **RundownPrefix:** The prefix to be used in the roId of the commands. (The same as set in the mosconfig.xml file in the Avid MOS Gateway.
- **ConvertStoryIdToDecimal:** Specifies whether the story id should be converted from Hex to Decimal value.
- **ConnectedWhenIdle:** This value determines if the connection to the graphic devices are maintained while Viz Mosart Server is in Idle mode.
  True = Maintain connection to graphic devices while in Idle mode. This setting will also

Copyright © 2024 Vizrt

initialize a connection to devices if Viz Mosart Server is started in Idle.
False = Do not start/maintain connection while in Idle mode.

- **ConnectedWhenInStandBy:** This value determines if the connection to the graphic devices are maintained while Mosart is in Standby mode.
  True = Maintain connection to graphic devices while in Standby mode. This setting will also initialize a connection to devices if Mosart is started in *Standby*.
  False = Do not start/maintain connection while in Standby mode.

Note:

- You can specify Channel on the PPCC for each PPCC engine in AV Automation and Overlay Graphics.
- This channel will only be used for OutputBlack messages (clearing an engine).
- It's possible to define *multiple* channels using both comma and semicolon as separator.
- Be aware that the channel will be cleared on the PPCC regardless of the connection information.
  For example, clearing channel 1 clears *all* graphics on this channel.

Pixel Power (Clarity) Configuration File

The Pixel Power (Clarity) configuration file is named *PixelPowerConfiguration.xml*.

```
<?xml version="1.0" encoding="utf-8" ?>
<DeviceConfig name="PixelPowerConfiguration" connectionString="localhost:0">
    <Properties>
        <!-- Connection parameters, may be overridden -->
        <item name="Server" value="localhost" /> <!-- Not in use -->
        <item name="Port" value="10220" /> <!-- Not in use -->
        <item name="DefaultTakeOutPage" value=""/> <!-- Not in use -->
        <item name="OutputBlack" value="false"/> <!-- Not in use -->
        <item name="JobPath" value="c:\Pixel_Power\Jobs\"/> <!-- Not in use -->
        <item name="Extension" value=".pjz"/>
        <item name="LowPageRange" value="1-999"/> <!-- Not in use -->
        <item name="FullScreenPageRange" value="1000-1999"/>
        <item name="CgPageRange" value="2000-9999"/>
        <item name="ClarityProtocolVersion" value="1.0"/>
        <item name="UseAsUpdateJob" value="false"/>
        <item name="CreatePageMode" value="true"/>
        <item name="WaitHandle" value="2000"/>
        <item name="HeartbeatInterval" value="10"/>
        <item name="PathCriteria" value=":"/>
        <item name="DisableAVAutomationLoadJob" value="true"/>
        <item name="DefaultOverlayPage" value="999" />
        <item name="UseDefaultPageChannel" value="true" />
        <item name="WaitForStories" value="3000"/>
        <item name="AskPageInfo" value="false"/>
        <item name="AskPageImage" value="false"/>
        <item name="AskFieldInfo" value="false"/>
```

Copyright © 2024 Vizrt

```
        <item name="ConnectedWhenIdle" value="false" />
        <item name="ConnectedWhenInStandBy" value="false" />
    </Properties>
</DeviceConfig>
```

Note: Items that are no longer supported for Viz Mosart's Pixel Power implementation are noted with `<!-- Not in use -->`.

Properties

- **Extension:** The file extension for the Pixel Power job to be loaded.
- **FullScreenPageRange:** Defines the range of Pixel Power page numbers used by Viz Mosart for full screen pages.
- **CgPageRange:** Defines the range of Pixel Power page numbers used by Viz Mosart for CG (lower third) pages.
- **UseAsUpdateJob:** Set if Pixel Power is to treat the loaded job as an update job.
- **CreatePageMode:** Informs Pixel Power to create the page from Viz Mosart or not.
- **WaitHandle:** Defines how long Viz Mosart should wait before removing a layer from the cued page after sending a trigger animation char command to Pixel Power.
- **HeartbeatInterval:** Defines how often Viz Mosart should check if the connection to Pixel Power is valid. The value is given in seconds.
- **PathCriteria:** Defines criteria that must be met (if any) for a job path to be valid.
- **DisableAVAutomationLoadJob:** If this value is set to true, only Overlay Graphics will load jobs on Pixel Power. Note that the current implementation will not handle other job/page logic from AV Automation (such as delete pages). This value is expected to be true.
- **DefaultOverlayPage:** Defines the overlay page to use as default on which to add layers to. This is the CG page that will be transferred on taking overlay graphics.
- **UseDefaultPageChannel:** Lets the pages created use the default page channel defined in Pixel Power. If set to false, the pages created will use the channel defined in engine settings.
- **WaitForStories:** Defines how many seconds Viz Mosart should wait for a story before deleting old/non-existing pages (handles shortcut keys from Viz Mosart GUI).
- **ConnectedWhenIdle:** This value determines if the connection to the graphic devices are maintained while Mosart is in Idle mode.
  True = Maintain connection to graphic devices while in Idle mode. This setting will also initialize a connection to devices if Mosart is started in Idle.
  False = Do not start/maintain connection while in Idle mode.
- **ConnectedWhenInStandBy:** This value determines if the connection to the graphic devices are maintained while Viz Mosart is in Standby mode.
  True = Maintain connection to graphic devices while in Standby mode. This setting will also initialize a connection to devices if Viz Mosart is started in *Standby*.
  False = Do not start/maintain connection while in Standby mode.
- **AskPageInfo:** Defines whether Viz Mosart requests Pixel Power for page info when creating a page.
- **AskPageImage:** Defines whether Viz Mosart requests Pixel Power for page image when creating a page.

Copyright © 2024 Vizrt

- **AskFieldInfo:** Defines whether Viz Mosart requests Pixel Power for a pages field info when creating a page.

  ⚠ **Note:** It is not necessary to ask for the last three items when creating a page.

Vizrt Graphics Configuration File

The Vizrt Graphics Configuration file is named *VizrtGraphicsConfiguration.xml*.

```
<?xml version="1.0" encoding="utf-8" ?>
<DeviceConfig name="VizrtGraphicsConfiguration">
    <Properties>
        <item name="default_effect_dsk" value=""/>
        <item name="default_effect_wall" value=""/>
        <item name="default_effect_full1" value=""/>
        <item name="allowUpdateOfOnAirItems" value="true"/>
        <item name="RemoveUnusedChannelsOutputs" value="true"/>
        <item name="SetConceptOnOutputChannel" value="true"/>
        <item name="TakeInCommand" value="" />
        <item name="TakeOutCommand" value=""/>
    </Properties>
</DeviceConfig>
```

Properties

⚠ **Note**: For a complete list of available properties, refer to the comments in *VizrtGraphicsConfiguration.xml*. This file is usually located under folder C:\channeltemplates.

- **default_effect_dsk:** Default effect to be used on the dsk handler (must be lower case)
- **default_effect_wall:** Default effect to be used on the wall handler (must be lower case)
- **default_effect_full1:** Default effect to be used on the fullscreen graphics engine 1 (must be lower case)
- **allowUpdateOfOnAirItems:** Allows an item to be updated even if it is onair (default = true)
- **RemoveUnusedChannelsOutputs:** When enabled, MSE Mosart Profile will synchronize to the Overlay Graphics Configuration, i.e. all outputs that are not used in Viz Mosart will be deleted. (default = true)
- **SetConceptOnOutputChannel:** Current Concept Override functionality sets a context environment variable named "alternative_concept" when scheduling an operation to the MSE. This variable will override any Concepts defined for an Output in the VCP/Trio profile editor. In a setup where there are multiple outputs assigned to a Channel where each output has Concepts assigned except the first, all outputs will run with the main concept. The following setting will set the Concept of all outputs under Mosart control in the Profile to the currently selected Concept and this will be the default behavior. The old way of sending the value in the environment for ScheduleElement can still be used by setting the property SetConceptOnOutputChannel to false. (default = true)

Copyright © 2024 Vizrt

The following **Viz Trio properties** are used when connecting to Viz Trio. They were originally stored in *VizTrioConfig.xml*. Viz Trio connections must use port 6200.

- **TakeInCommand:** Used to override take in commands. Use this if you require something other than `page:take`
- **TakeOutCommand:** Used to override take out commands if you require something other than `page:takeout`

Vizrt Media Sequencer VDom Logic Macros

It is possible to send custom commands to the Media Sequencer. This can be done by VDom logic which can run on the Media Sequencer as macros from Viz Mosart.

An optional `VizrtUserMacros.xml` file should be created and placed in the config files folder with the following format:

Macro file format:

```
<macros>
  <macro name="myMacroName">
    <!-- Place a single VDom element here, for instance <env /> -->
  </macro>
  <macro name="StartScene">
    <env>
      <viz>RENDERER*STAGE START</viz>
    </env>
  </macro>
</macros>
```

The macros are executed with the following variables:

Variable	Value	Comment
profile	/config/profiles/MOSART	
viz	The viz handler for the engine targeted in the MACRO call	
channel	The profile channel name for the engine targeted in the MACRO call	Only send if channel names are applied to elements

Chyron Configuration File

The Chyron configuration file is named *ChyronConfig.xml*.

The configurable parameters are documented with self-explanatory values in the code below.

```
<DeviceConfig name="DeviceConfig">
  <Properties>
```

Copyright © 2024 Vizrt

```
    <!-- Macro for taking out graphics on Chyron channel x -->
    <item name="TakeOutMacro1" value="BREAKOUT_A" />
    <item name="TakeOutMacro2" value="BREAKOUT_B" />

    <!-- Configure settings for taking out with transition graphics on Chyron channel
x for Intelligent Interface Rules Engine (IIRE).
    Specify the transition to be used to effect out the graphic. In the examples
below, 2 take out commands have been defined for port 1 and 2
    with tranistion "animate". {GraphicID} is a placeholder and must be given in the
specified format. -->
    <item name="IIRETakeOutMacro1" value="L\{GraphicID}\animate" />
    <item name="IIRETakeOutMacro2" value="L\{GraphicID}\animate" />

    <!--
         Encoding: Default encoding utf-16
         Page containing valid .Net encodings:
         http://blogs.msdn.com/b/shawnste/archive/2009/08/18/alternate-encoding-
names-recognized-by-net-ie.aspx
    -->
    <item name="Encoding" value="utf-16"/>
    <!-- If true the connection to the graphics devices are maintained while Mosart
is in idle mode -->
    <item name="ConnectedWhenIdle" value="false" />
    <!-- If true the connection to the graphics devices are maintained while Mosart
is in standby or idle modes -->
    <item name="ConnectedWhenInStandBy" value="false" />

    <!--Whether to send the NEW Check Existence command. (Requires Lyric build 2021
 or higher. Use Help / About Lyric to find out.)-->
    <item name="ShouldCheckExistence" value="false"/>

    <!--Timeout for the NEW Check Existence command-->
    <item name="TimeoutCheckExistence" value="1000"/>

    <!--
      Enable the PreviewOnOutput to keep the old behavior of cueing and taking an
item for preview.
      Leave as false for previewing on the local preview output of the Chyron. Note
that when the below
      setting is enabled, preview of CGs will be taken directly to air and is a
misconfiguration of the system.
    -->
    <item name="PreviewOnOutput" value="false"/>
  </Properties>
</DeviceConfig>
```

Deko, Orad, Xpression Configuration File

Several graphics drivers use variants of the common config file `IntelligentInterfaceConfiguration.xml`. These include

- Deko
- Orad
- Xpression

Their respective parameters are documented with self-explanatory values in the code below.

```
<?xml version="1.0" encoding="utf-8" ?>
<DeviceConfig name="IntelligentInterfaceConfiguration">
  <Properties>
    <!-- Config file for all Intelligent Interface Graphics Devices -->
    <item name="IgnoreCreateCommandInStandby" value="false" />
    <item name="LoadInBackground" value="false"/>
    <!-- Encoding to be used for communication. Typical values: "utf-16", "utf-8" -->
    <item name="Encoding" value="utf-16"/>
    <!-- The name of the Xml attribute containing the graphics id used when
communicating with the graphics devices -->
    <item name="GraphicsIdAttribute" value="graphics_id" />
    <!-- If true the connection to the graphics devices are maintained while Mosart
is in idle mode -->
    <item name="ConnectedWhenIdle" value="false" />
    <!-- If true the connection to the graphics devices are maintained while Mosart
is in idle mode -->
    <item name="ConnectedWhenInStandBy" value="false" />

    <!-- DEKO -->
    <item name="UseEffectsAttachedToTemplates" value="true"/>
    <item name="TakeInWithCueAndTransferKeyCodes" value="false"/>
    <item name="IgnoreKeyCodesOnTakeIn" value="false"/>
    <item name="ClearScene" value=""/>
    <!--<item name="TakeInPrefixFileName" value="\xF7"/>
    <item name="TakeInPostfixFileName" value="\xF8\xF7"/>-->

    <!-- XPRESSION -->
    <item name="XpressionResumeBeforeTakeOffline" value="false" />
    <!--true/false-->
    <item name="XpressionTakeOfflineDelay" value="2000" />
    <!--milliseconds (req. XpressionResumeBeforeTakeOffline to be true)-->
    <item name="XpressionSuppressAlfanumericTemplates" value="false" />
    <!--true/false-->
    <item name="XpressionRetryIndexGenerator" value="10" />
    <!--true/false-->
    <item name="XpressionOverlayIndexLimit" value="100000" />
    <!--true/false-->

    <!-- GRAPHICS : ORAD -->
    <!-- Activates preload in preview for given destinations.
         Syntax: comma separeted list of destinations: Example "DSK,WALL" -->
    <item name="OradPreloadDestinations" value="" />
    <!-- Either [1,2]. The number of response message expected after a preload
command V\5\13. Default 1-->
    <item name="OradPreloadNumResponses" value="1" />
    <!-- Orad Heartbeat interval in seconds. Set to 0 to disable -->
```

```
    <item name="OradPollInterval" value="60" />
    <!--seconds-->
    <!-- Minimum time to wait after take command before sending new commands to Orad
-->
    <item name="OradTakeBlocks" value="10" />
    <!--milliseconds-->
    <!-- Minimum time to wait after cue command before sending new commands to Orad
-->
    <item name="OradMaxWaitTillCued" value="6000" />
    <!--milliseconds-->
    <!-- Obsolete: Use general Encoding property instead -->
    <!--<item name="OradEncoding" value="unicode" />-->
    <item name="OradSkipRecue" value="never" />
    <!-- Either [last,loaded,never]. Controls when to skip recue-->
    <!-- If set, responses from last preview operation will be stored. Corresponding
play will not take place if cue is not verified -->
    <item name="OradEnableCueVerification" value="false" />
    <!-- If set, a Select Output command is sent as part of preview. Complient with
Orad/CII protocol, section 4.13-->
    <item name="OradEnableSelectOutput" value="false" />
  </Properties>
</DeviceConfig>
```

Copyright © 2024 Vizrt

Subtitling Configuration Files

ScreenLL Configuration File

The ScreenLL configuration file is named *ClipServerScreenLL.xml*.

```
<?xml version="1.0" encoding="utf-8" ?>
 <MediaServerConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="
http://www.w3.org/2001/XMLSchema">
   <Properties>
     <!-- Language code to look for to decide clip availability -->
     <item name="AvailabilityLanguageCode" value="OPN" />
     <!-- First Incue Time code that indicates invalidated clip -->
     <item name="InvalidatedFirstIncueTime" value="--:--:--:--" />
   </Properties>
 </MediaServerConfig>
```

Properties

- **AvailabilityLanguageCode:** Specifies which language code must be present in order for a clip to be marked as available. The language code is a three letter text string, which relates to a list of language codes found in the file *PoliSTLServerLangs.lst*, which must be present on the ScreenLL subtitle file server.
- **InvalidatedFirstIncueTime:** Specifies the First Incue Time value that will indicate that a subtitle clip has been invalidated (is not available even if it matches the *AvailabilityLanguageCode)*.

Other configuration items are customer-specific or for Vizrt internal use only.

Copyright © 2024 Vizrt

Video Router Configuration Files

- Miranda NVision Configuration File
- EvertzQuartz

Miranda NVision Configuration File

The integration between Viz Mosart and Miranda NVision video routers supports setting cross points on the router.

- Viz Mosart will interface directly with the NVISION 5128 Router.
- Router Control is handled by AV Automaton.
- Testing can be done using the Viz Mosart TestRouterControl application.

The config file is called VideoRouterMirandaNV9000.xml, located at \\Mosart Medialab\Mosart Server\ConfigurationFiles

```
<?xml version="1.0" encoding="utf-8" ?>
<DeviceConfig name="MirandaNV9000Config">
  <Properties>
    <!--The amount of time (in ms) to wait before reconnection after unsuccessful
connect or lost connection-->
    <item name="ReconnectInterval" value="10000"/>

    <!--The Take mode parameter to the Take commands. Refer to the NV9000 spec pp. 8-9
 for details. The default value is Automatic take mode.-->
    <item name="TakeMode" value="0x00000001"/>

    <!--Not used-->
    <item name="TimeoutSetCrossPoint" value="1000"/>

    <!--The User ID. Available in the NV9000 DB. Alternatively the client IP address
as a network byte order (big-endian) 32 bit word. E.g., the (hex) IP address
FE.DC.BA.98 should be written
    0xFEDCBA98.-->
    <item name="UserID" value="0xFEDCBA98"/>

    <!--Whether (True/False) to use the Take Source To Destination command. If False,
use the Take Input To Output command instead. Default True.-->
    <item name="UseTakeSourceToDestination" value="True"/>
  </Properties>
</DeviceConfig>
```

Protocol Details

For details on the NV9000 protocol, refer to the GrassValley product documentation (https://community.grassvalley.com/support/s/).

Copyright © 2024 Vizrt

File Location

The Miranda Nvision configuration file is located in the program-folder under `\Mosart Medialab\Mosart server\ConfigurationFiles\VideoRouterMirandaNV9000.xml`.

Properties

- **ReconnectInterval**: The amount of time (in milliseconds) to wait before reconnection after unsuccessful connect or lost connection. Default: `10000`.
- **TakeMode**: The Take mode parameter to the Take commands. Default: `0x00000001` (Automatic take mode).
- **UserID**: The User ID. This can be fetched from the NV9000 database, or can be created by converting the client IP address to a network byte order (big-endian) 32 bit word. Default: `0xFEDCBA98`.
  For example, the (hex) IP address FE.DC.BA.98 should be written 0xFEDCBA98.
- **UseTakeSourceToDestination**: This value determines which protocol commands to use for setting a crosspoint:
  - True = 0x0000 3000 *Take Source To Destination* (default).
  - False = 0x0000 3001 *Take Input To Output.*

> **Note:**
> - To integrate with Miranda NVision video routers, you must also configure it in Router.
>   - Select the router protocol MIRANDA NV9000.
>   - Set the IP address of the router, and the Port number (use `9193`).

EvertzQuartz

The config file is called VideoRouterEvertzQuartz.xml, located at \\Mosart Medialab\Mosart Server\ConfigurationFiles

Copyright © 2024 Vizrt

Audio Mixer Configuration Files

These config files are initially located in the Viz Mosart Server installation folder in a sub folder, *Configuration Files*.
For operations, you copy the file to "%ProgramData%/

- Calrec TCP/IP Configuration File
- SSL Configuration File
- Lawo Ember+ Configuration File
- Ardour Configuration File
- MidiMixer Configuration File
- General Audio Mixer Configuration File

- See also Audio.

Calrec TCP/IP Configuration File

The Calrec TCP/IP configuration file is named *AudioMixerCalrec.xml*.

SSL Configuration File

The SSL configuration file is the same as used for Calrec TCP/IP Configuration File.

Lawo Ember+ Configuration File

The Lawo Ember+ configuration file is named *AudioMixerLawoEmberPlus.xml*.

Please see the file for the available settings and their documentation.

Ardour Configuration File

The Ardour configuration file is named *AudioMixerArdour.xml*.

Please see the file for the available settings and their documentation.

MidiMixer Configuration File

The MidiMixer configuration file is named *AudioMixerMidiMixer.xml*.

Please see the file for the available settings and their documentation.

General Audio Mixer Configuration File

For the other audio mixers, those not specifically mentioned above, there is a general configuration file named *AudioMixerConfiguration.xml*.

Please see the file for the available settings and their documentation.

Note that in the default version of this file, there are some settings specific for the Lawo RMNOPL audio mixer driver.

Copyright © 2024 Vizrt

Vision Mixer Configuration Files

The configuration files for the following vision mixers (switchers) are listed here for convenience. See the inline commenting for parameter explanation.

> ⚠ **Note**: Only changes to commented values, where indicated as variables, are supported!

- ClipServerVizEngine.xml
- GvgVideoSwitcherConfig.xml
- GrassValleyCPLVideoSwitcherConfig.xml
- RossVideoSwitcherConfig.xml
- SonyVideoSwitcherConfig.xml
- VizEngineSwitcherConfig.xml
- VizrtIpSystemsVideoSwitcherConfig.xml

ClipServerVizEngine.xml

Settings for the video server configuration file and in addition, for Viz Engine as a video server. Only commented values may be changed.

```
<DeviceConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http:/
/www.w3.org/2001/XMLSchema"> <Properties>
    <item name="KeepCache" value="False" />
    <item name="RefreshOnUpdate" value="True" />
    <!-- The layer a switching scene is located in on. Only used for video servers,
for Engine Switcher (Viz Engine) this value is always "MIDDLE". Default: Back -->
    <item name="ControlSceneLayer" value="MIDDLE" />
    <!-- Shall always be true, do not change. Default: true -->
    <item name="UseScriptCommands" value="true" />
    <!-- The path prefix of the clips. Default: <clip_root> -->
    <item name="ClipNamePrefix" value="" />
    <!-- Clip extension (if not set, all types can be played, but needs to be
specified in the NRCS). Default: empty -->
    <item name="ClipNamePostfix" value="" />
    <item name="AssetToMediaMap" value="UseTitleAsSlug" />
    <!-- The version of the Switcher Scene. Used by Mosart to check the channel
status when connecting to a port. For versions after 1, it does not need Mosart to
check the status. It is dynamic. Default: 1 -->
    <item name="SwitcherSceneVersion" value="2" />
 </Properties>
```

> ⚠ **Note**: For Engine Switcher Mode configurations in the Viz Engine, see Engine Switcher Configuration.

Copyright © 2024 Vizrt

GvgVideoSwitcherConfig.xml

```
<DeviceConfig name="GvgVideoSwitcherConfig">
  <Properties>
    <item name="SendBreak" value="true" />
    <item name="EnableHeartbeat" value="true" />
  </Properties>
</DeviceConfig>
```

- **SendBreak**: All GVG 4000 protocol switchers requires a break before connecting (but there are switchers which do not need this).
- **EnableHeartbeat**: When no commands are issued, a heartbeat is sent to the vision mixer with an interval equal to the configured heartbeat interval.

GrassValleyCPLVideoSwitcherConfig.xml

```
<DeviceConfig name="GrassValleyVideSwithcerConfig">
  <Properties>
    <!-- Interval in seconds when to test for a valid connection when not connected
-->
    <item name="ConnectionCheckIntervalInSecondsNotConnected" value="5" />
    <!-- Interval in seconds when to test for a valid connection when connected -->
    <item name="ConnectionCheckIntervalInSecondsConnected" value="10" />
  </Properties>
</DeviceConfig>
```

RossVideoSwitcherConfig.xml

⚠ **Note**: This config replaces CarboniteVideoSwitcherConfig.xml.

```
<DeviceConfig name="CarboniteVideoSwitcherConfig">
  <Properties>
    <!-- The properties below is the virtual cross point mapping for internal Ross
Carbonite sources. Alias can be anything. Value is the virtual assigned crosspoint to
be set in AV Config XML. Name is the RossTalk command for the item.
    -->
    <item alias="MediaStore1" value="1001" name="MS:1" />
    <item alias="MediaStore2" value="1002" name="MS:2" />
    <item alias="MediaStore3" value="1003" name="MS:3" />
    <item alias="MediaStore4" value="1004" name="MS:4" />
    <item alias="Black" value="1008" name="BK" />
    <item alias="MatteColor" value="1009" name="BG" />
    <item alias="ME1" value="1011" name="ME:1" />
    <item alias="ME2" value="1012" name="ME:2" />
```

Copyright © 2024 Vizrt

```
    <item alias="MiniME1" value="1021" name="MME:1" />
    <item alias="MiniME2" value="1022" name="MME:2" />
    <item alias="MiniME3" value="1023" name="MME:3" />
    <item alias="MiniME4" value="1024" name="MME:4" />
    <item alias="MuliScreen1" value="1031" name="MSC:1" />
    <item alias="MuliScreen2" value="1032" name="MSC:2" />
  </Properties>
</DeviceConfig>
```

SonyVideoSwitcherConfig.xml

```
<DeviceConfig name="SonySwitcherConfig">
  <Properties>
    <!-- Truns the heartbeat mechanism on/off. -->
    <!-- item name="EnableHeartbeat" value="false" / -->

    <!-- The number of milliseconds before a time-out occurs when a read operation
does not finish.
         If set less than zero the time-out will be infinite -->
    <!-- item name="ReadTimeout" value="100" / -->

    <!-- ComPort2 is used for the SonySerialTallyDriver in split mode, using the
second serialport for the SCU connection -->
    <!-- <item name="ComPort2" value="COM23" /> -->

    <!-- The properties below routes the different commands sets to either the
standard or the extra COM port
         None: always com port 1,
         Partial: com port 1 if the crosspoint range or mix effect range is supported
by com port 2, otherwise com port 2
         Full: always com port 2
    -->
    <!--
    <item name="AutoTransComPort2" value="None" />
    <item name="SetCrossPointComPort2" value="Full" />
    <item name="SetCrossPointComPort2" value="Full" />
    <item name="SetAuxCrossPointComPort2" value="Full" />
    <item name="SetKeyFillCrossPointComPort2" value="Full" />
    <item name="SetKeyerComPort2" value="Full" />
    <item name="SetDskComPort2" value="Partial" />
    <item name="SetTransTypeComPort2" value="Full" />
    <item name="RecallRegionComPort2" value="None" />
    -->
  </Properties>
</DeviceConfig>
```

VizEngineSwitcherConfig.xml

This config is employed when controlling video switching directly from a Vizrt Viz Engine, as presented in section Viz Engine as a Switcher.

Copyright © 2024 Vizrt

```
<DeviceConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http:/
/www.w3.org/2001/XMLSchema">
  <Properties>
    <item name="ControlScenePath" value="Vizrt/VizMosart/ControlScenes/switcher" />
    <item name="NumberOfKeyBuses" value="8"/> <!-- Number of key buses in template
editor (default value is 4, only other valid option is 8) -->
  </Properties>
</DeviceConfig>
```

Note: Configurations are also required on the Viz Engine server. See Engine Switcher Configuration.

VizrtIpSystemsVideoSwitcherConfig.xml

```
<DeviceConfig name="VizrtIpSystemsVideoSwitcherConfig">
  <Properties>
    <!-- The path where EMEMs are stored -->
    <item name="PathToEmem" value="D:\Sessions\SessionName\emem\" />

    <!-- Set this to true if the crosspoint names should be sent in the command to
the driver instead of the physical inputs for the NDI Outputs. Default: true.-->
    <item name="UseOutputCrosspointNames" value="true" />

  </Properties>
</DeviceConfig>
```

Copyright © 2024 Vizrt

Engine Switcher Configuration

For setup and configurations that use these settings, please refer to section Viz Engine as a Switcher.

- switcher_config.cfg
- ClipServerVizEngine.xml

switcher_config.cfg

> **Note**: These configurations are performed on the Viz Engine (not Viz Mosart) machine.

File Location

The Switcher Scene can be configured from its configuration file, `switcher_config.cfg` located in C:\ProgramData\vizrt\VizEngine.

Configurable Parameters

#scene path for preview engine
<switcher_scene>Vizrt\Engine-Switcher\switcher</switcher_scene>

#scene path for openers if static_paths = 0 or 2
<opener_path>Vizrt\Engine-Switcher\openers\</opener_path>

##scene path for effects if static_paths = 0 or 2
<effect_path>Vizrt\Engine-Switcher\effects\</effect_path>

#type of paths: 0 = absolute path i.e. ALL/MyPath/openers, 1 = openers/effects underneath switcher scene, 2 = relative path underneath switcher scene i.e. myeffects
<static_paths>2</static_paths>

#amount of effect scenes for preloading
<effects>10</effects>

#content transition effect number to begin with
<content_effect_start>90</content_effect_start>

#external transition effect number to begin with, end at content_effect_start
<ext_effect_start>50</ext_effect_start>

#external opener/trans within GFX channels
<gfx_channel_support>1</gfx_channel_support>

#activating external transition effect engine - not recommended due to possible network delays
<ext_trans_active>0</ext_trans_active>

#transition engine ip
<transition_host>127.0.0.1</transition_host>

#transition engine port
<transition_port>6100</transition_port>

#activating external top layer engine
<ext_top_scene_active>0</ext_top_scene_active>

Copyright © 2024 Vizrt

```
#external top layer engine ip
<ext_top_scene_host>127.0.0.1</ext_top_scene_host>

#external top layer engine port
<ext_top_scene_port>6100</ext_top_scene_port>

#activating external preview
<ext_prev_active>0</ext_prev_active>

#external preview hostname
<ext_prev_host>127.0.0.1</ext_prev_host>

#external preview port
<ext_prev_port>6100</ext_prev_port>

#show commands (log level)
<show_commands>2</show_commands>

#delaying the actual take in fields coming via external gfx scenes - this is used for having more time to load ip sources in preview
<external_take_delay>0</external_take_delay>

#i am peview
<i_am_preview>0</i_am_preview>

#activating dynamic ip workflow (Probel source request)
<dynamic_ip_workflow>0</dynamic_ip_workflow>

#shared memory map name used for status signaling
<dynamic_ip_status>pgmstatus_prv</dynamic_ip_status>

#shared memory map name used for Probel status callbacks - mandatory for using Probel communication!
<dynamic_ip_callback>probelstatus-machinename</dynamic_ip_callback>

#shared memory map name used for indicating a toggle (might be used in other gfx scenes)
<toggle_status>toggle_mapname</toggle_status>

#time waiting for the release of "old" sources after the toggle of new ones
<release_delay>10</release_delay>

#Source Prefix
<source_prefix>LIVE</source_prefix>

#size of the maximum number of active preview sources
<preview_queue_size>5</preview_queue_size>

#viz one in use (needed for setting clip prefix)
<viz_one>0</viz_one>

#clip prefix
<clip_prefix>V:\</clip_prefix>

#defines how the status overlay scene gets loaded - 1 = direct load to frontlayer - 2 = dynamic scene container in fl scene
<status_overlay_mode>1</status_overlay_mode>

#path to status overlay scene in GH
<status_overlay>Default\statusoverlay_new</status_overlay>

#start playing clip on preview machines
<play_clip_on_preview>0</play_clip_on_preview>
```

Copyright © 2024 Vizrt

#trigger tally lights
<tally_active>0</tally_active>

#tally server name
<tally_name>tally_pcr_main</tally_name>

#tally backup server name
<tally_name_backup>tally_pcr_backup</tally_name_backup>

#custom content transition effects
<content_transitions>80:wipe,81:roll,82:twist</content_transitions>

#vs sources input table (not used)
<vs_input_sources>11:960,12:961,13:962,14:963,15:964,16:965,17:966,18:967,21:960,22:961,23:962,24:963,25:964,26:965,27:966,28:967,31:960,32:961,33:962,34:963,35:964,36:965,37:966,38:967</vs_sources>

#vs sources output table (not used)
<vs_output_sources>11:42,12:43,21:44,22:45,31:46,32:47</vs_sources>

#tracking hub hosts (not used)
<tracking_hub_A>10.29.33.54:20000</tracking_hub_A>
<tracking_hub_B>10.29.33.54:20000</tracking_hub_B>
<tracking_hub_C>10.29.33.54:20000</tracking_hub_C>

#delay added for fullscreen toggle - this helps in adjusting proper timing with VS switches
<full_toggle_delay>4</full_toggle_delay>

#delay for VS switch when toggling from a fullscreen to a window with a cam switch happening
<full_win_delay>4</full_win_delay>

#time waiting for toggling superchannels when an external wipe animation is triggered
<wipe_delay>20</wipe_delay>

#time waiting for toggling superchannels containing clips which need to match a wall clip
<wall_clip_delay>10</wall_clip_delay>

#gpi sharedmemory prefix name
<gpi_shm_name>Sealevel::0::I</gpi_shm_name>

#tracking hub user
<th_user>user1</th_user>

#CUE Clip Workflow
<cue_clip_on_load>1</cue_clip_on_load>

#replacing clip path when file is not found
<substitute_clip>V:\CLIP_NOT_READY.mxf</substitute_clip>

#shared memory map name used for storing the upcoming label names
<label_names>MIXER_LABELS</label_names>

ClipServerVizEngine.xml

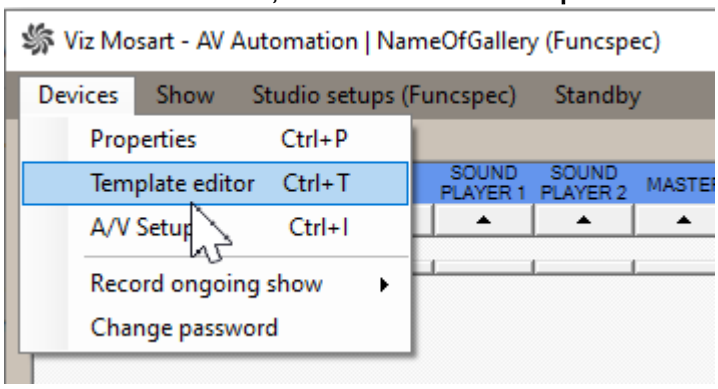
This configuration is employed on the Viz mosart side, when controlling video switching directly from a Vizrt Viz Engine, as presented in section Viz Engine as a Switcher.
For available configuration options, refer to section Vision Mixer Configuration Files.

Copyright © 2024 Vizrt

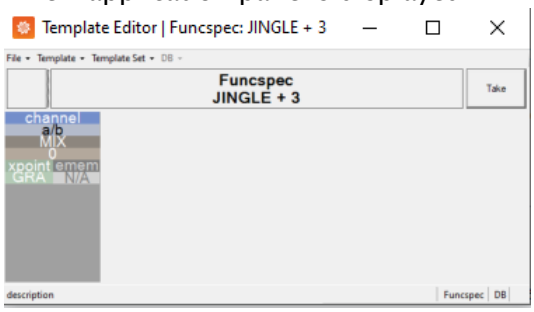
4.7 Template Editor

You create templates as part of your show design. Use the **Template Editor** to configure the Viz Mosart templates.

- From **AV Automation**, select **Devices** > **Template editor**.



A new application pane is displayed:

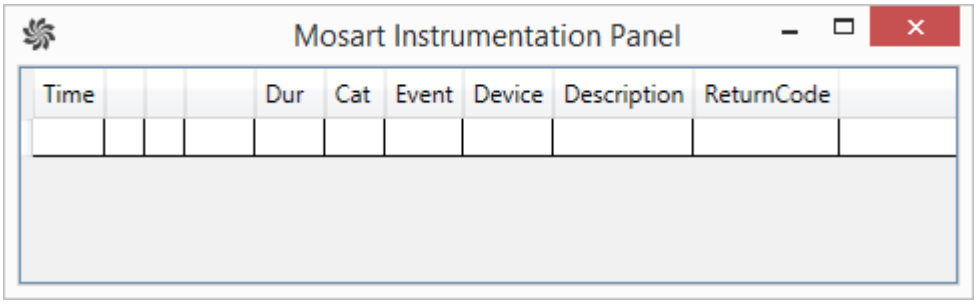


- From this view you can select the required template/template variant (with **Template > Select**) and work with the device settings as described in Template Device Functions.

Show design and detailed template topics are presented in section Show Design and Template Editor.

Copyright © 2024 Vizrt

4.8 Mosart Instrumentation Panel



The **Mosart Instrumentation Panel** displays all device commands in real time, that are being sent to the various connected devices. Each command is time stamped, revealing when a device command was issued, in relation to the taking of a Mosart template.

Copyright © 2024 Vizrt

4.9 Recording

Viz Mosart can instruct a connected video server to initiate recording.

> **Note**: The **Story Recorder** feature of Viz Mosart offers more versatile record-pause-retake features, beyond the recording functionality described in this topic.
> Please refer to section Story Recorder Mode.

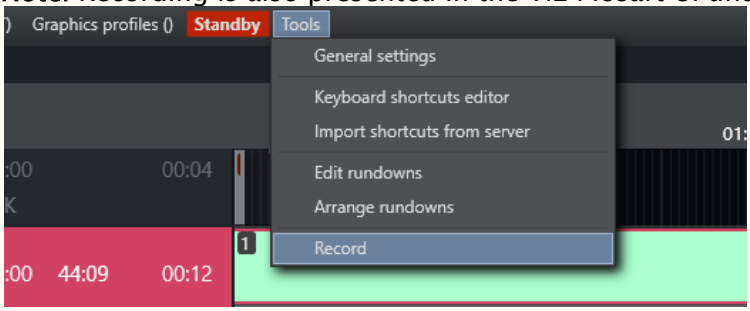
4.9.1 Working with Recordings

- Prerequisites
- Managing Recordings

Prerequisites

The video server must first be configured, as explained in the procedure **To add a recording port** of Video Servers.

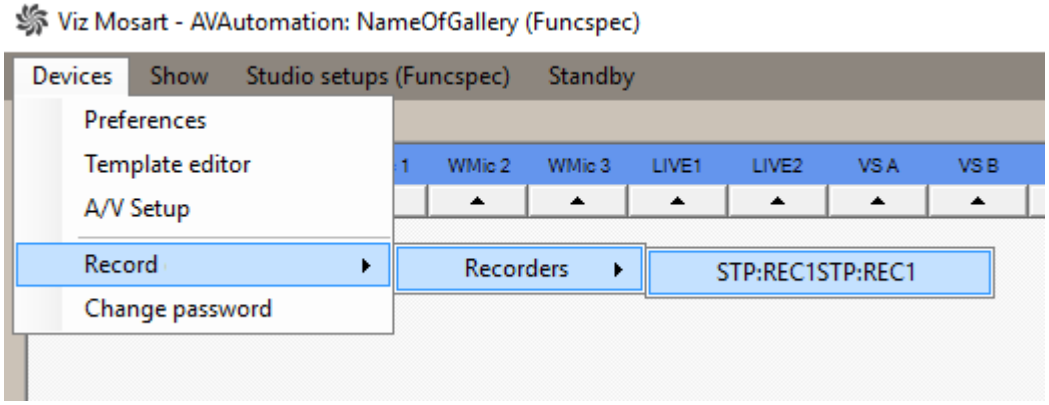
- The configured video servers are displayed in AV Automation under **Devices > Record**.

> **Note:** Recording is also presented in the Viz Mosart UI under **Tools > Record**
>
> 
>

Managing Recordings

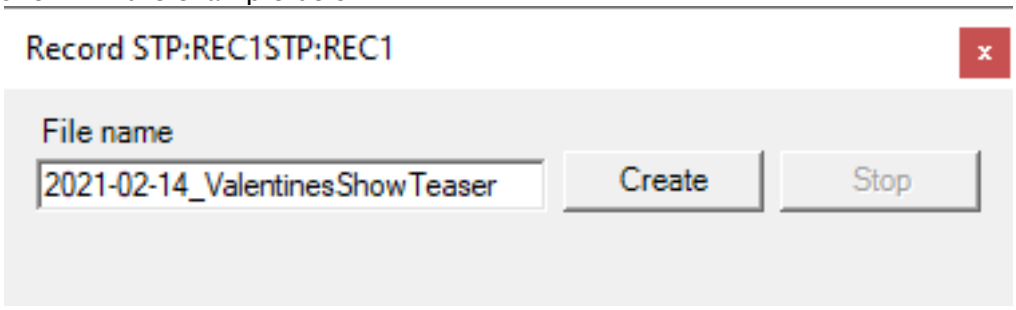
To Start a Recording

1. In AV Automation, select **Devices > Record > Recorders > [Name of recording port]**

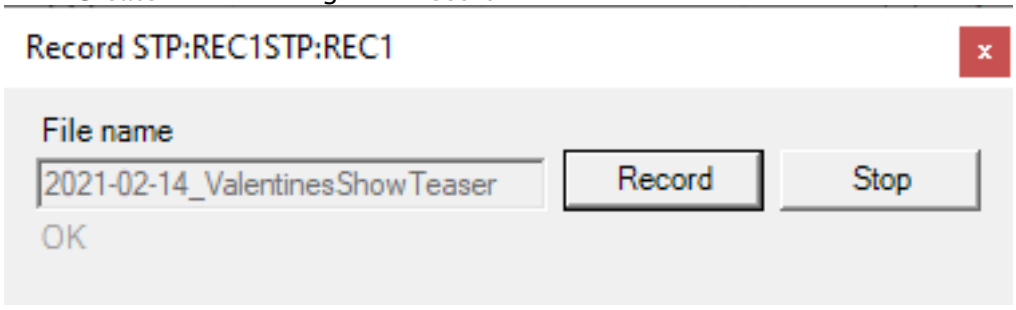


> **Note:** If several recording ports have been configured, the Recorders menu will have several menu items.

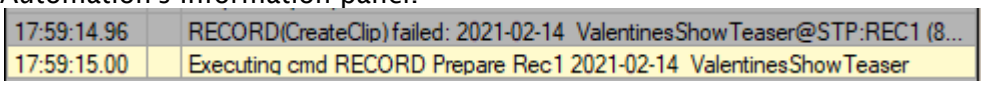
2. In the dialog box, enter a file name that will make it simpler to identify your recording, as shown in the example below:



3. Click **Create**.
The appearance of the dialog box changes.
- The **Create** button changes to **Record**.

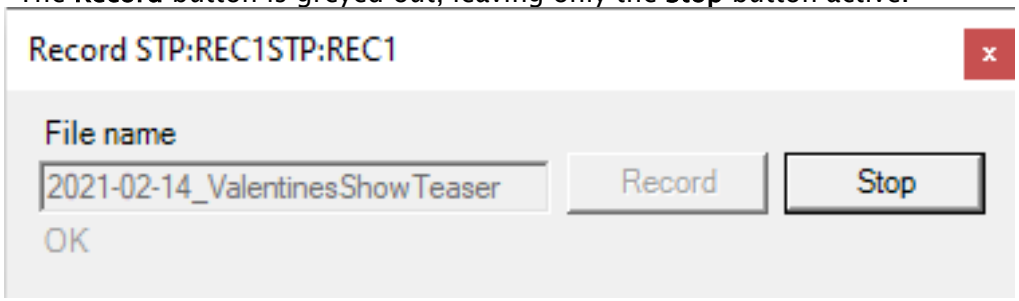


- The commands that are sent in the background to the video server, are presented in AV Automation's Information panel.

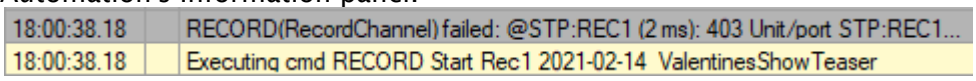


Copyright © 2024 Vizrt

4. Click **Record**.
   The appearance of the dialog box changes once again.
   - The **Record** button is greyed-out, leaving only the **Stop** button active.



   - The commands that are sent in the background to the video server, are presented in AV Automation's Information panel.



Recordings are stored on the video server's system. Exactly where, is dependent on the video server type.

To Stop a Recording

To stop an ongoing recording,

- Click **Stop**.
  The commands that are sent in the background to the video server, are presented in AV Automation's Information panel.

4.9.2 Supported Video Servers

Viz Mosart is continually improved to extend the range of video server drivers that are supported for recording.

Driver	Supported since
Omneon	2010-01
Harris Nexio	2011-09
VDCP	2014-07
Quantel	2016-08
MVCP	2016-08
Grass Vally K2	2016-09

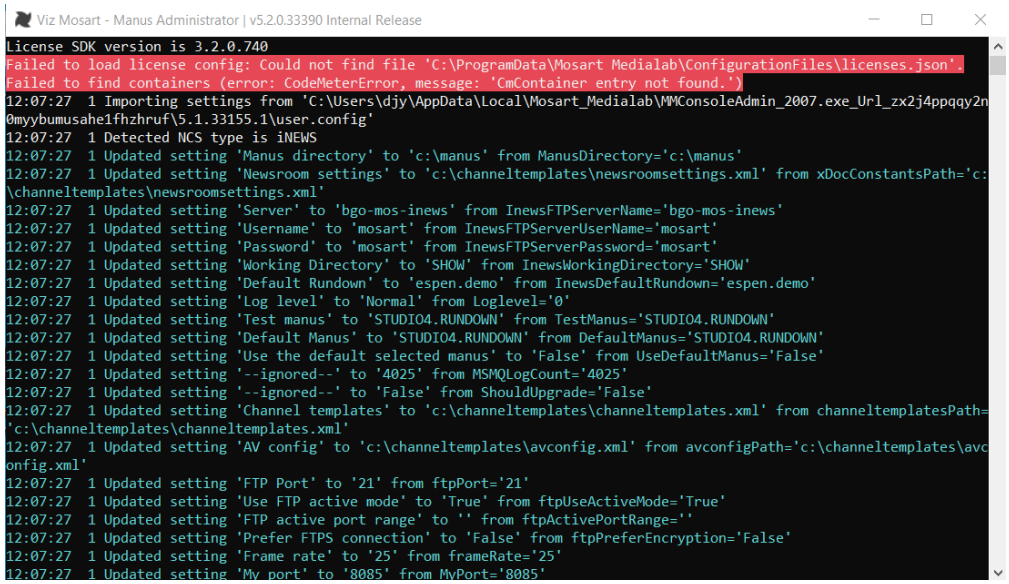
Copyright © 2024 Vizrt

Driver	Supported since
AirSpeed MultiStream	2018-05

Copyright © 2024 Vizrt

5 Manus Administrator

Manus Administrator is the intelligent element of Viz Mosart, monitoring the rundown in the connected NRCS, parsing the content into Viz Mosart functionality that is presented in the Viz Mosart GUI.



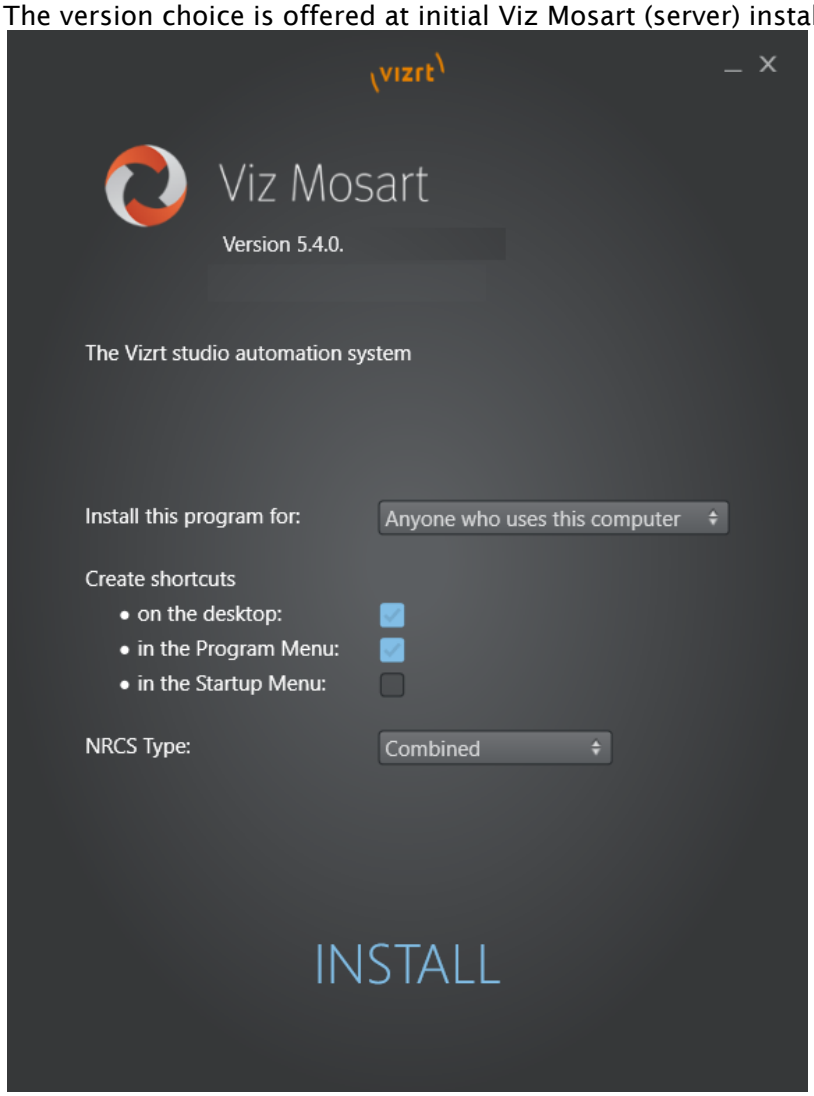
Manus Administrator is a console application and must be kept running at all times on your Mosart server.

5.1 NRCS Software Compatibility

There are three versions of the Manus Administrator, corresponding to the flavor of newsroom system that it connects to.

Copyright © 2024 Vizrt

The version choice is offered at initial Viz Mosart (server) installation:



- (Default) **Vizrt.Mosart.ManusAdministrator**.exe. A combined iNEWS/MOS program.
- **MMConsoleAdmin_2007**.exe. Only for iNEWS (with ftp) workflow.
- **MMConsoleAdmin_MOS**.exe. Only for newsroom systems with *MOS* workflow.
  Use for all systems except iNEWS with ftp workflow, including iNEWS with MOS workflow.

> **Note**: All three (MOS, iNEWS and Combined) Viz Mosart Manus Administrator entities are always installed. The selection made with the **NRCS Type** drop-down decides which of these three, the Manus Administrator shortcut shall point to.

Only *one* Manus Administrator can be active at any given time.

> **Tip:** Type **help** in the Manus Administrator console window to see which commands you can use. See Manus Administrator Commands.

See also Avid iNews Web Service for Status Feedback in the appendix.

Copyright © 2024 Vizrt

5.2 Configuration Utilities

5.2.1 Centralized Settings Menu

A significant Viz Mosart ease-of-use improvement was introduced with the combined Manus, from version 5.3.

- All settings, including NRCS connectivity, are now presented in a single Settings menu. See section Viz Mosart Settings and Configurations.

To open the Settings menu

From the Manus Administrator console you can open a configuration editor in a new menu:

- Type `settings` in the Manus Administrator console, then ENTER.

For details see:

- Viz Mosart Settings and Configurations (combined)
- Settings Editor iNews or
- Settings Editor MOS
  according to your Manus selection at initial installation.

5.2.2 Newsroom Settings

- To open the **Newsroom Settings XML Editor**, type `ns` in the console window followed by **ENTER**. For details see:
  - Newsroom Settings Editor.

> ⚠ **Note:** The Newsroom Settings editor opens in a new window. If you make changes to the configuration, restart Manus Administrator for the changes to be applied.

The rest of this section covers these topics:

- Settings Editor iNews
- Settings Editor MOS
- Configuring the MSE Playlist Panel
- Newsroom Settings Editor
- Field Mapping
- Manus Administrator Commands

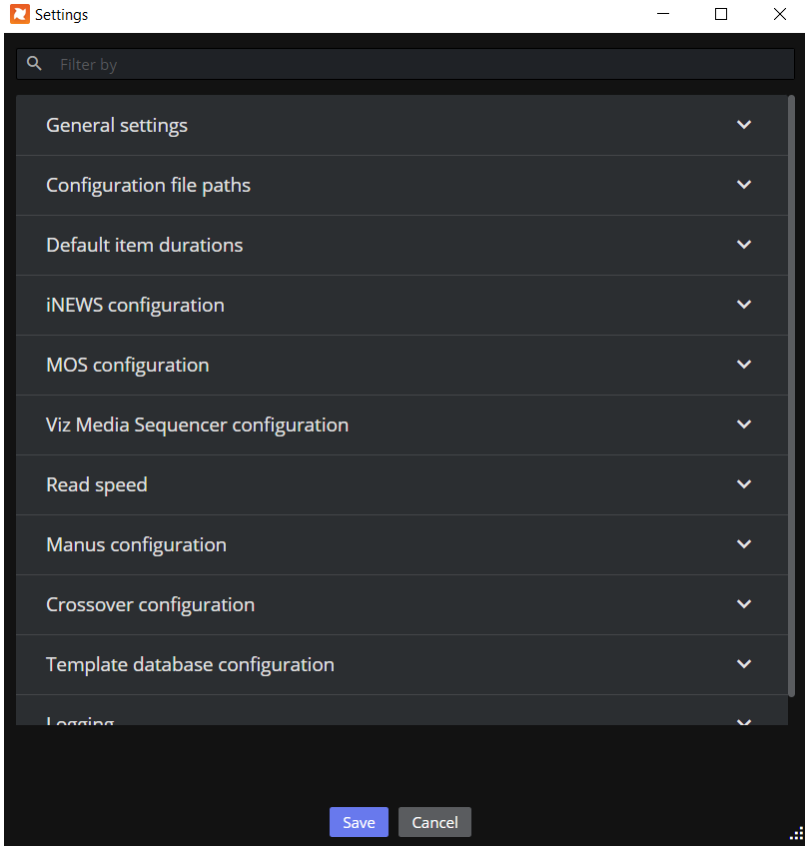
Copyright © 2024 Vizrt

5.3 Settings Editor INews

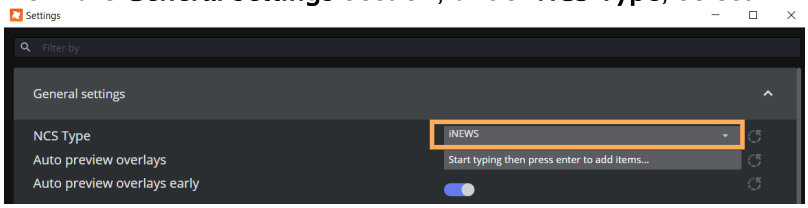
5.3.1 Working with iNEWS Settings

To simplify setting up the connection to iNEWS, the Manus Admin console includes a Settings utility.

1. Open the Settings utility by typing `settings` followed by **ENTER**.

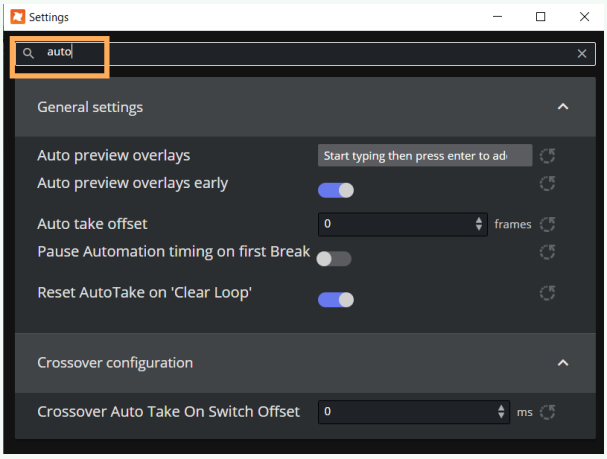


2. From the **General settings** section, under **NCS Type**, select *iNEWS*.



Tip:

- You can search for a specific value from the Search box.



The editor comprises several sections, each dedicated to a collection of settings:

- General
- File Paths
- Default Item Durations
- iNEWS
- Viz Media Sequencer
- Read Speed
- Manus
- Crossover
- Template Database
- Logging
- Vizrt Graphics

General

- **NCS Type:** Selecting *iNEWS* presents the configurable values listed here.
- **Auto preview overlays:** List of graphic destinations that support preview of overlay graphics (For example, *DSK* enables preview of all overlay graphics with *handler=DSK).* Default: *Empty* = no preview.
- **Auto preview overlays early**: Enable preview of overlay graphics as early as possible. When *not* enabled, the preview will only be displayed when the graphic is the next item to be taken.
- **Autotake offset:** Sets offset for the autotake function. Default: *0* frames.
- **Default handler name:** For all graphics that have no handler name, Mosart can default to a specific handler for these. You can assign this handler here. Default: *DSK.*
- **Default Lower Third out behaviour:** Default method for taking out a lower third. Default: *TIMECODE*.
- **Force clip editorial time:** Shows Editorial time in GUI. Default: *Off*.
- **Frame rate:** Determines the frame rate of the system. Valid rates *25, 29.97, 50* and *60*.

Copyright © 2024 Vizrt

- **Ignore updates if nothing changes:** Ignores story update from the NRCS if the updated story is assumed to be equal to the current version of the story when set. Default: *Off* (NRCS story updates are *not* ignored).
- **Keep story status on updates from NRCS:** Keeps played-out stories gray if the story is updated in the NRCS. Default: *Not enabled*, do not keep story status.
- **Max number of dynamically inserted items**: The number of story items that will be inserted and stored dynamically on a Mosart story row. Items can, for example, be inserted with a keyboard shortcut. Default: *0* (all items will be stored).

  > ⚠ **Note**: The greater the number of dynamic items that are kept in a story, the slower the system will become. You will need to verify what performance is acceptable for you, depending on your story properties.
  > If you experience slowness in the Viz Mosart UI while dynamically taking items into preview or program on a running story, you can try initially changing this value to *5* and then verifying on-air performance.
  > Default value for this setting is *0*, which means that behavior of your system will be unchanged, keeping all manual items.
  >
  > - For *adlibs* (floats) this solution introduces a potentially undesirable side effect. If the adlib is removed from the list of manual items, Viz Mosart will lose the paused position, so that when the adlib is taken again, it will start playing from the *beginning* instead of from the paused position.

- **Metadata creator priority:** Overrides priority timing information from the video file's source. Select between:
  - *Newsroom editorial time*
  - *MAM duration*
  - *Full video clip length*.

Separate IDs with commas. Use the name of the Media Administrator entry. Special names are *NCS* (for information from the NRCS) and *GLOBAL* (global values from media object).
If *Empty*, Viz Mosart inserts *NCS* as the first and *GLOBAL* as last entries.
Default: *Empty*.

- **Minimum clip length**: Sets the minimum visible length of an offline clip in a story in the GUI. When Viz Mosart receives clip information the visible length is updated. Default: *5* (ss:ff).
- **My port:** The port used for other Viz Mosart applications to connect to the Manus Administrator. Default: *8085*.
- **Network exclude:** Manus Administrator automatically enters idle mode on network failure. This property is a CSV list of network names to ignore when monitoring networks. Default: *Empty* (monitor all networks).
- **Network include:** Manus Administrator automatically enters idle mode on network failure. This property is a CSV list of network names to monitor for network failure. Default: *Empty* (monitor all networks).
- **Offset secondary events with mix delay:** Secondary events follow primary events mix delay. For example, a lower third element is delayed with the same mix delay as the package it is to be keyed on. Default: *False*

Copyright © 2024 Vizrt

- **Pause Automation timing on first Break:** When enabled, any first story with a single break template will not start the automation server rundown timing (Elapsed rundown duration). The timing will start when the next story is taken. This does not affect the NRCS timing.
- **Preload Accessory Cue Delay**: Defines the delay between end of cue the next item and when a pending preload/pretake accessory shall be executed.
  Time is in frames. Default = *6* frames, so the accessory preload/pretake functions take place after the cue operation is done in AV Automation, by this number of frames.
- **Reset AutoTake on 'Clear Loop':** Disables the autotake mode automatically when using the clear loop function from the GUI when enabled. Default: *Enabled*.
- **Server description:** User-defined descriptive name of the server that is displayed in the **Timing Display**.
- **Should upgrade:** For *Vizrt Support* use only. Do not change. Default: *False*
- **Startup in idle mode:** Determines whether or not the **Manus Administrator** is idle at start-up. For normal Viz Mosart operation, this setting shall always be *True*, so that the Viz Mosart server starts up in Idle mode. Switching between *Idle* and *Active* mode shall in normal operation, be done from the GUI. Default: *True*
- **Story compare ignore attributes:** Used for debugging purposes only. Default: *Empty*
- **Templates allowing graphic pretake:** Lists template types that allow pretake of overlay graphics elements. Available template types:

- *ADLIBPIX*
- *BREAK*
- *CAMERA*
- *DVE*
- *FULLSCREENGRAPHICS*
- *LIVE*
- *PACKAGE*
- *TELEPHONEINTERVIEW*
- *VOICEOVER*

Default *: PACKAGE* and *VOICEOVER.*
Only the first graphic found on the next item is taken and this is taken immediately regardless of its start time.

> ⚠ **Note:** For graphic pretake to work, make sure the Pretake overlay on handler setting is also configured in the Newsroom Settings Editor.

- **Transitions on Accessories:** Allows transition effects on accessory templates when enabled. If disabled, any transition effects are removed from accessory templates when translating newsroom XML into Mosart XML. This is how it has been, historically.
- **Use TakeOut logic:** Enables the CG takeout functionality. When two CG graphic objects with the same graphic IDs are next to each other, the first one is not taken out, only **take in** on the second object is executed.
  When *True*, enables the takeout functionality for overlay graphics. Default: *False.*
- **Advanced Clip Status Display:** When selected, this option activates advanced visualization within the Mosart UI, providing indicators for different clip states. These include, for example,

Copyright © 2024 Vizrt

- clips present only on mirror servers
- clips available on the main server but missing from the mirror
- clips missing on the specified playout server in the template.

When this option is *unselected*, the system defaults to the original visualization method, where a solid blue color signifies clip availability on at least one of the configured video servers in Media Administrator, and a blue and red checkerboard pattern denotes unavailability.
This feature is accessible exclusively in the new combined Manus Administrator settings.
Default: *Selected*.

- **Ignore Send Cue Status to NRCS for offline clips:** When selected, the NRCS will not receive CUED and READY status for offline video clips. Default: *Unselected,* d*o not ignore*.
- **Refresh media on NRCS update:** When selected, triggers a clip refresh for all clips belonging to a story that is being updated. This usually happens through the NRCS story update mechanism. Ensures that all clip information is synchronized. Default: *Do not trigger refresh*.
- **Time in ms to delay initializing devices while waiting for NRCS to add story:** Set to 1200 to avoid traffic overload (blinking whilst loading). Set to 500 to avoid the first story being set till the end of the rundown. Default: *500 ms*.

File Paths

- **AV config:** The path to the XML-file containing the audio and video mappings defined in the AV Automation application. Default: `c:\channeltemplates\avconfig.xml`.
- **Channel templates:** The path to the file containing the Viz Mosart templates built in the AV Automation Template Editor. Default: `c:\channeltemplates\channeltemplates.xml`.
- **Manus directory:** The path to the folder containing copies of the internal Viz Mosart rundown. Default: `c:\manus`.
- **Newsroom settings:** The path to the XML-file containing the mappings from newsroom system commands to Viz Mosart templates. Default: `c:\channeltemplates\newsroomsettings.xml`.

Default Item Durations

- **Break:** The minimum length given to the **BREAK** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *100* (frames).
- **Item:** The minimum length given to any template type in the rundown. If no time is given in the NCS, then this time is given to the element.
  Default: *100* (frames).
- **Camera:** The minimum length given to the **CAMERA** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *125* (frames).

Copyright © 2024 Vizrt

- **DVE:** The minimum length given to the **DVE** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *125* (frames).
- **Fullscreen graphic:** The minimum length given to the **FULLSCREEN GRAPHIC** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *125* (frames).
- **Live:** The minimum length given to the **LIVE** template type in the rundown. If no time is given in the NCS, then this time is given to the element.
  Default: *125* (frames).
- **Lower third:** The minimum length given to the Lower-third template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *100* (frames).
- **Telephone interview:** The minimum length given to the **PHONO** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *125* (frames).
- **Video clip:** The minimum length given to the **CLIP** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *125* (frames).

iNEWS

- **Server:** Hostname or IP address where the iNews FTP server is running.
- **User name:** Username of a valid iNews FTP account. Normally the name of a valid iNews user with access rights to the iNews FTP server.
- **Password:** Password of a valid iNews FTP account. Normally password to a valid iNews user with access rights to the iNews FTP server.
- **FTP Port:** iNews FTP port.
- **Use FTP active mode:**
  Selected: Use FTP Active mode, where Manus Administrator establishes the command channel but the iNEWS server is responsible for establishing the data channel.
  Unselected: Use FTP Passive mode, where Manus Administrator establishes both channels.

  > ⚠ **Note**: In *active* mode, the client starts listening for incoming data connections from the server on port M. It sends the FTP command PORT M to inform the server on which port it is listening. The server then initiates a data channel to the client from its port 20, the FTP server data port.

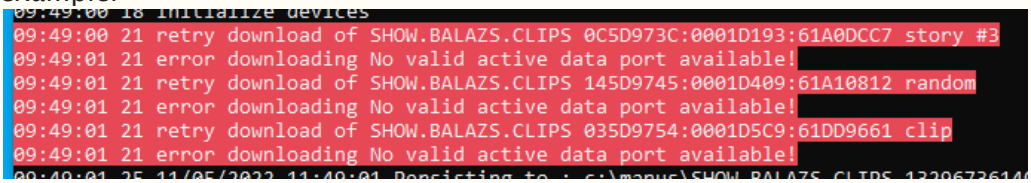
  False: Passive mode where Manus Administrator establishes both channels.

  > ⚠ **Note**: Where the client is behind a firewall and unable to accept incoming TCP connections, *passive* mode may be used. The client then uses the control connection to send a PASV command to the server and then receives a server IP address and server port number from the server, which the client then uses to open a data connection from an arbitrary client port to the server IP address and server port number received

Copyright © 2024 Vizrt

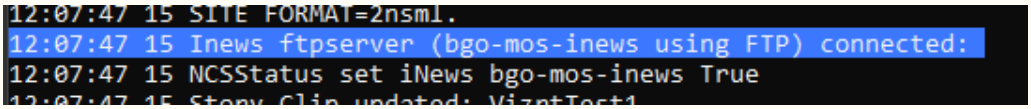
- **FTP active port range:** Ports used for Active mode. The format is '*<low_port>..<high_port>*', random ports are used when left empty or the format is wrong. Useful when the current machine has certain ports that are blocked or used for other purposes.

  ⚠ Only used when active mode is selected. Similar to the iNEWS environment variable RXDATAPORTRANGE. Because a new connection is opened for each data transfer (listing directories, downloading stories) the port range must be sufficiently wide (~100 wide range works in most cases).
  - When the port range is left empty or the format is invalid, random ports are used.
  - When the port range is not wide enough, the rundown won't fully load.
  - These situations are indicated in the Manus Admin console and in the logs. For example:

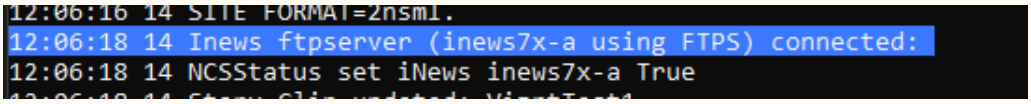


- **Prefer FTPS connection:**
  True: FTPS encryption is used if supported by the iNews server. Otherwise it falls back to plaintext FTP communication.
  False: Using plain text FTP communication.

  ⚠ Because FTP uses a dynamic secondary port (for data channels), many firewalls were designed to snoop FTP protocol control messages in order to determine which secondary data connections they need to allow. However, if the FTP control connection is encrypted using TLS/SSL, the firewall cannot determine the TCP port number of a data connection negotiated between the client and FTP server. Therefore, in many firewalled networks, an FTPS deployment will fail when an unencrypted FTP deployment will work. This problem can be solved with the use of a *limited range of ports* for data and configuring the firewall to *open* these ports.
  FTP connection type and status are reported in Manus Admin:
  - With FTP:



  - With FTPS:



- **Working directory:** Initial directory in iNews. Viz Mosart provides access to all rundowns stored hierarchically within this directory.
  Default: *SHOW*
- **Encoding:** Character encoding of the iNews story items. Should match the encoding used in iNews.

Copyright © 2024 Vizrt

Default: Encoding for the operating system's current ANSI code page.
Other values: *UTF7, UTF8, UTF32, UNICODE, BIGENDIANUNICODE, ASCII*.

- **Web service connection:**
  Use the provided connection string in the format 'param1=value1;param2=value2;...' to establish a connection with Avid iNEWS Web Services.
  If you prefer not to connect to any web service, leave this field blank.
  - **Mandatory parameters**:
    WebServiceServer=host:port, where you can provide only the host if using the default port (8080).
  - **Optionally parameters**: You can include 'iNewsServer,' 'iNewsUsername,' 'iNewsPassword,' and 'Type,' which are sourced from your iNEWS configurations, if available.
    For 'Type,' valid values are 'iNEWS' and 'MosiNEWS.'
  - **Additional configurations**: These can be made in the AvidInewsWebServiceConfig.xml file, located in the Mosart installation folder under Mosart Server/ConfigurationFiles. Any settings configured here will override values found in the configuration file.

⚠ **Note**: For complete setup and configuration of iNews Web Service, please refer to section Avid iNews Web Service for Status Feedback.

Viz Media Sequencer

NCS server: When Showmaker 1.0 is used as NRCS for Viz Mosart, this is where you configure the host address to the Media Sequencer (MSE) that Showmaker is connected to.

MSE Playlist Panel Host: The host address of the Media Sequencer (MSE) used by the **MSE Playlist Panel** in the Viz Mosart UI
Configure the host address of existing MSE host to activate the **MSE Playlist Panel**. The **Playlist Panel** offers users active (touch or click-sensitive) thumbnails of graphics held on an MSE.
The Viz Mosart UI needs this MSE connection defined in order for the user to see the **Playlist Panel** (**View > Floating windows > Playlist Panel**). See the *Viz Mosart User Guide*, section *Viz Mosart user Interface > MSE Playlist Panel*.

Read Speed

- **Read speed:** Number of words read per minute by the anchor. Applied to the prompter text to determine the duration of an announcement, which is then the length of the green camera bar and the green part of the voice-over bar in the **Viz Mosart** GUI.
  Default: *145* (words per minute)
  Words are usually taken to be the parts of text separated by space characters. See the setting Use character for read speed word below.

- **Use character for read speed word:** If *True*, each *character* counts as a word for the purpose of calculating the spoken duration of prompter text, see Read speed above.
  This option can be useful for languages like Chinese and Thai, where words are not

Copyright © 2024 Vizrt

separated by using the space character. If *False* (default), words are separated by space characters.

Manus

- **Default Manus:** Selects the default rundown to use. If set to *True,* the system loads this rundown on startup.
- **Manus expiration time:** Number of days to keep Manus Administrator files. Older files (than that number of days) are deleted when Manus Administrator is started.
  Default: *60* (files are deleted after 60 days).
  Legal values: *-1* (never deleted), *1 and above* (number of days until deleted from `manus` folder).
- **Manus keep file pattern:** If Manus expiration time is given, this property allows a list of file search patterns for files to keep in the Manus Administrator directory.
  Files with names satisfying any of these patterns will be kept even if they are 'too old' according to the **Manus expiration time**
  For example, "*TEST**;DEMO??.xml" keeps all files starting with "*TEST*" and all *DEMO??.xml* files where '?' denotes a wild character.
  Default: *Empty* (no files to keep).

  > ⚠ **Note**: ';' is used to separate the file search patterns.
- **Test manus:** Rundown to be used for maintenance purposes.
- **Use the default selection of Manus:** Initializes the rundown given in the Default Manus setting automatically when enabled. Default: *True*.

Crossover

- **Connection string:** The connection string of the crossover, for example: **controller**=*IP address*, **client**=*crossover*. Default: ""
- **Crossover autotake on switch offset:** Offset in milliseconds to pre-take an autotake when the next story item is after a crossover switch.
  A positive value triggers the autotake before the given duration of the story item.
  A negative value triggers the autotake after the given duration of the story item.
- **Crossover SetNext on switch delay:** Delay of sending set next story from the server running the show when taking control. Default: *500* (milliseconds).

Template Database

- **Use template database:** When true, enables the Template DB functionality, so template type aliases can be exported from DB to `newsroomsettings.xml` at startup, and exporting `newsroomsettings.xml` to DB after saving `newsroomsettings.xml`.
  Default: False.

Copyright © 2024 Vizrt

⚠ **Note:** Enter the **Connection string** before enabling **Use template database**.

- **Name of DB provider:** The provider name for the **Template DB**. For example, *MySql.Data.MySqlClient* for MySQL.
  Default: *MySql.Data.MySqlClient.*
- **Connection string:** The connection string for the **Template DB**. For example, for MySQL: server=<hostname>; User Id=<user>;Password=<password>; database=mosarttemplatedb.
  Default: *server=localhost;User Id=root;database=mosarttemplatedb.*
- **Default template DB inserter:** The name to be used for the **insertedby** and **updatedby** columns in the Mosart Template Database.
  Default: *inserter.*

Logging

- **Log level**: Sets the detail level of logging to the log file. *N*ormal, *W*arnings, *E*rrors, Info, *D*etailed.
  Default: *0*.
- **Log MOS communication:** Enable logging of the communication to and from the NRCS.
  Default: *False.*
- **In use:** Enables or disables the logging of events from the Manus Administrator to the log file.
  Default: *True*.
- **WTC Level:** Sets the detail level of logging to the console. *Error, Message, External message, Details*.
  Default: *Details*.
- **Trace internally:** Enables or disables internal tracing to console, for debugging only.
  Default: *False*.
- **Use verbose logging:** Enables or disables verbose logging. If **In use** is set, using verbose the details sent to the log.
  Default: *False*.
- **Ignore verbose events filter:** Semicolon based list of events to ignore in the log when using verbose logging.
  Default: *Empty* (log all events)
- **Pass verbose events filter:** Semicolon based list of events to log when using verbose logging.
  Default: *Empty* (log all events)
- **MSMQ Log limit:** Value to identify when the application should dump the log queue to file.
  Default: *4023*
- **Path for MMLog:** The path where the **Viz Mosart Log** is stored.
  Default: *MosartLog*.

Vizrt Graphics

- **Enable**: Allow use of Vizrt graphics
- **MSE Host**: The hostname and port number (default *8580*) of the MSE used for playout.

Copyright © 2024 Vizrt

- **Playlist name**: The name of the MSE playlist where Vizrt content from a Mosart show will be added and made ready for playout.
- **MSE Profile name**: Do not change this value.

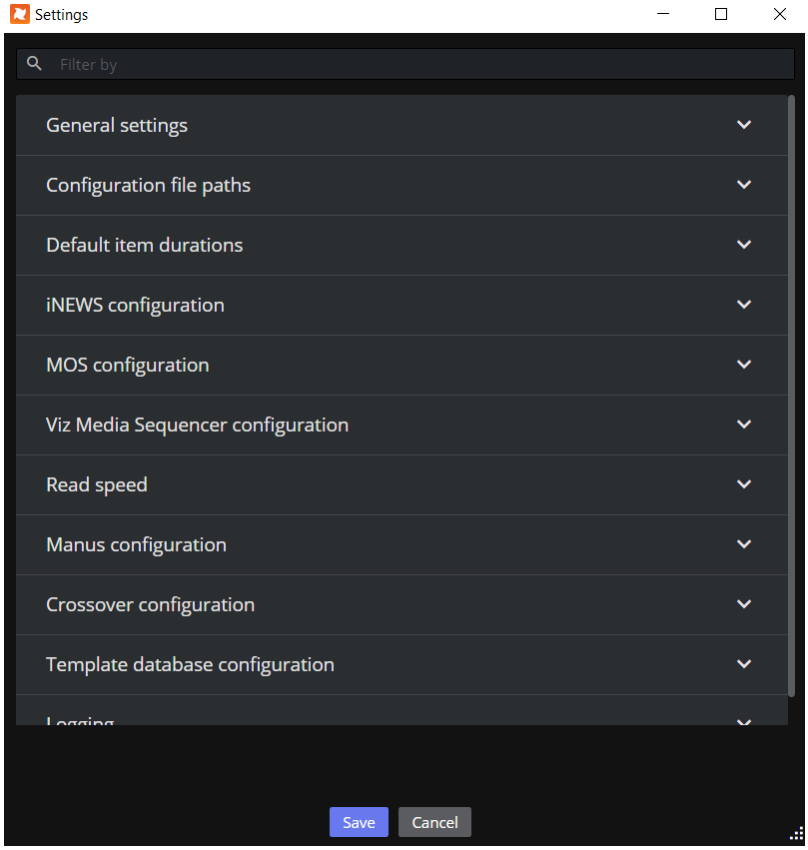
Copyright © 2024 Vizrt

5.4 Settings Editor MOS

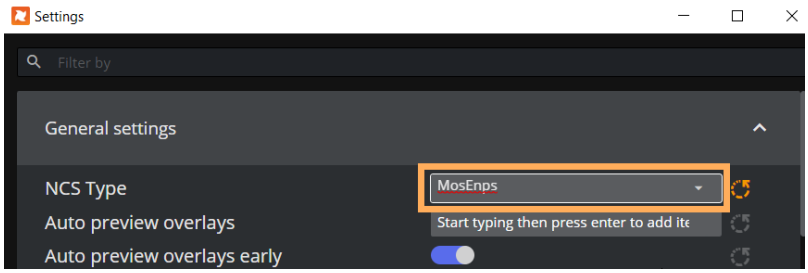
5.4.1 Working with MOS Settings

To simplify setting up the connection to a MOS-based NRCS, the Manus Admin console includes a Settings utility.

1. Open the Settings utility by typing `settings` followed by **ENTER**.

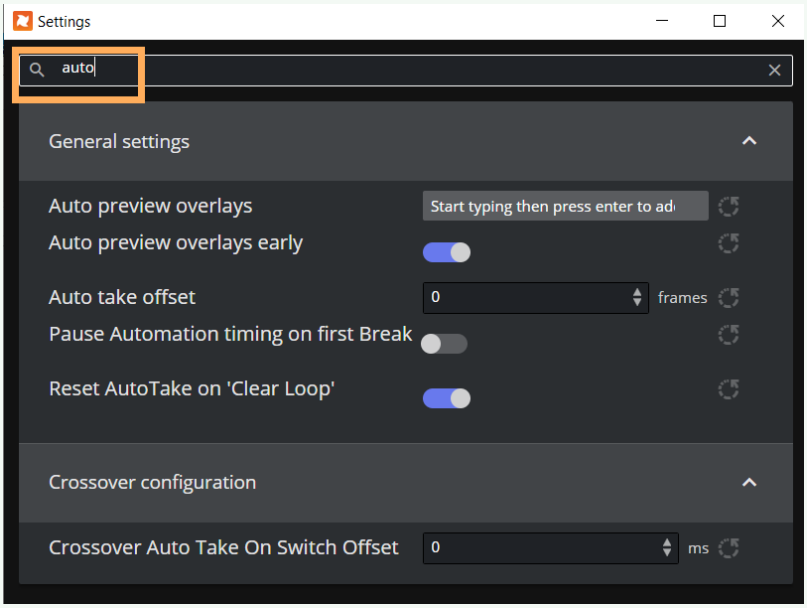


2. From the **General settings** section, under **NCS Type**, select the flavor of MOS that the NRCS uses.



Tip:

- You can search for a specific value from the Search box.



The editor comprises several sections, each dedicated to a collection of settings:

- General
- File Paths
- Default Item Durations
- MOS
- Read Speed
- Viz Media Sequencer
- Manus
- Crossover
- Template Database
- Logging
- Vizrt Graphics

General

- **NRCS Type:** Generic: No special handling of native NRCS commands. Can be: *Generic*, *DaletPlus*, *ENPS*, *NcPower*, *Octopus*, *Open Media* and *MOSInews*.
  Default: *Generic.*
- **Auto preview overlays:** Lists graphics destinations that support preview of overlay graphics (i.e. DSK enables preview of all overlay graphics with *handler=DSK).*
  Default: *Empty* = no preview.
- **Auto preview overlays early**: Enable preview of overlay graphics as early as possible. When set to *False*, the preview will be displayed only when the graphics is the next item to be taken.
- **Autotake offset:** Sets offset for the autotake function.
  Default: *0* (milliseconds).

Copyright © 2024 Vizrt

- **Default handler name:** Defines a default handler name for all graphics that have no handler name.
  Default: *DSK.*
- **Default Lower Third out behaviour:**
  Default: *TIMECODE.*
- **Force clip editorial time:** Shows Editorial time in GUI.
  Default: *False.*
- **Frame rate:** Determines the frame rate of the system. Valid rates *25*, *29.97*, *50* and *60*.
  Default: *25.*
- **Ignore updates if nothing changes:** If *True* , a story update from the NRCS is ignored if the updated story is assumed to be equal to the current version of that story.
  Default: *False* (no NRCS story updates are ignored).
- **Keep story status on updates from NRCS:** When enabled, keeps played out stories gray if the story is updated in the NRCS.
  Default: *False.*
- **Max number of dynamically inserted items**: The number of story items that will be inserted and stored dynamically on a Mosart story row. Items can, for example, be inserted with a keyboard shortcut. Default: *0* (all items will be stored).

  > ⚠ **Note**: The greater the number of dynamic items that are kept in a story, the slower the system will become. You will need to verify what performance is acceptable for you, depending on your story properties.
  > If you experience slowness in the Viz Mosart UI while dynamically taking items into preview or program on a running story, you can try initially changing this value to *5* and then verifying onair performance.
  > Default value for this setting is 0, which means that behavior of your system will be unchanged, keeping all manual items.
  >
  > - For *adlibs* (floats) this solution introduces a potentially undesirable side effect. If the adlib is removed from the list of manual items, Viz Mosart will lose the paused position, so that when the adlib is taken again, it will start playing from the *beginning* instead of from the paused position.

- **Metadata creator priority:** Select between
  - *Newsroom editorial time*
  - *MAM duration*
  - *Full video clip length*.

Separate IDs with commas. Use the name of the Media Administrator entry. Special names are *NCS* (for information from the NRCS) and *GLOBAL* (global values from media object).
If *Empty*, Viz Mosart inserts *NCS* as the first and *GLOBAL* as last entries.
Default: *Empty.*

- **Minimum clip length:** This value sets the minimum visible length of an offline clip in a story in the GUI. When Viz Mosart receives clip information, the visible length is updated.
  Default: *5* (ss:ff).
- **Network exclude:** Manus Administrator automatically enters idle mode on network failure. This property is a CSV list of network names to ignore when monitoring networks.
  Default: *Empty* (monitor all networks).

Copyright © 2024 Vizrt

- **Network include:** Manus Administrator automatically enters idle mode on network failure. This property is a CSV list of network names to monitor for network failure.
  Default: *Empty* (monitor all networks)
- **Offset secondary events with mix delay:** Secondary events follow primary event mix delay. For example, a lower-third element has the same mix delay as the package on which it is to be keyed.
  Default: *False.*
- **Pause Automation timing on first Break:** When enabled, any first story with a single break template will not start the Automation server rundown timing (Elapsed rundown duration). The timing will start when the next story is taken. This does not affect the NRCS timing.
  Default: *False.*
- **Preload Accessory cue delay**: Defines the delay between end of cue the next item and execution of a pending preload/pretake accessory. The duration is displayed in frames.
  Default: *6* frames, i.e. the accessory preload/pretake functions shall take place after the cue operation is done in AV Automation with an additional and configurable delay of **PreloadAccessoryCueDelay** frames.
- **Reset AutoTake on 'Clear Loop':** Enable this to automatically disable the autotake mode when using the clear loop function from the GUI.
  Default: *True.*
- **Server description:** Used for display only, in **Timing Display**.
  Default: *ControlRoom.*
- **Startup in idle mode:** Determines whether or not Manus Administrator is idle at start-up. For standard Viz Mosart operation, this setting shall always be *True*, so that the Viz Mosart server starts up in *Idle* mode. Switching between Idle and Active mode shall in normal operation be done from the GUI.
  Default: *True.*
- **Story compare ignore attributes:** Used for debugging purposes only.
  Default: *Empty.*
- **Templates allowing graphic pretake:** Lists template types that allow pretake of overlay graphics elements. Available template types: *ADLIBPIX, BREAK, CAMERA, DVE, FULLSCREENGRAPHICS, LIVE, PACKAGE, TELEPHONEINTERVIEW, VOICEOVER*. Default: *PACKAGE, VOICEOVER.*
  Only the first graphic found on the next item is taken. This graphic is taken *immediately* regardless of its start time.

  > **Note:** To enable graphic pretake, the **Pretake overlay on handler** setting must be configured in the Newsroom Settings Editor.

- **Use item status to NRCS:** Sends online offline CUE/PLAY/STOP status to the NRCS when enabled. Default: *False*.
- **Advanced Clip Status Display:** When selected, this option activates advanced visualization within the Mosart UI, providing indicators for different clip states. These include, for example,
  - clips present only on mirror servers
  - clips available on the main server but missing from the mirror
  - clips missing on the specified playout server in the template.

Copyright © 2024 Vizrt

When this option is *unselected*, the system defaults to the original visualization method, where a solid blue color signifies clip availability on at least one of the configured video servers in Media Administrator, and a blue and red checkerboard pattern denotes unavailability.
This feature is accessible exclusively in the new combined Manus Administrator settings.
Default: *Selected*.

- **Ignore Send Cue Status to NRCS for offline clips:** When selected, the NRCS will not receive CUED and READY status for offline video clips. Default: *Unselected,* d*o not ignore*.
- **Refresh media on NRCS update:** When selected, triggers a clip refresh for all clips belonging to a story that is being updated. This usually happens through the NRCS story update mechanism. Ensures that all clip information is synchronized. Default: *Do not trigger refresh*.
- **Time in ms to delay initializing devices while waiting for NRCS to add story:** Set to 1200 to avoid traffic overload (blinking whilst loading). Set to 500 to avoid the first story being set till the end of the rundown. Default: *500 ms*.

File Paths

- **AV config**: The path to the XML-file containing the audio and video mappings defined in the AV Automation application.
  Default: *c:\channeltemplates\avconfig.xml*.
- **Channel templates:** The path to the file containing the Viz Mosart templates built in the AV Automation Template Editor.
  Default: *c:\channeltemplates\channeltemplates.xml*.
- **Manus directory**: The path to the folder containing copies of the internal Viz Mosart rundown.
  Default: *c:\manus*
- **Newsroom settings:** The path to the XML file containing the mappings from newsroom system commands to Viz Mosart templates.
  Default: *c:\channeltemplates\newsroomsettings.xml.*

Default Item Durations

- **Break:** The minimum length given to the **BREAK** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *100* (frames).
- **Item:** The minimum length given to any template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *100* (frames).
- **Camera:** The minimum length given to the **CAMERA** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *125* (frames).
- **DVE:** The minimum length given to the **DVE** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *125* (frames).

Copyright © 2024 Vizrt

- **Fullscreen graphic:** The minimum length given to the **FULLSCREEN GRAPHIC** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *125* (frames).
- **Live:** The minimum length given to the **LIVE** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *125* (frames).
- **Lower third:** The minimum length given to the Lower-third template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *100* (frames).
- **Telephone interview:** The minimum length given to the **PHONO** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *125* (frames).
- **Video clip:** The minimum length given to the **CLIP** template type in the rundown. If no time is given in the NRCS, then this time is given to the element.
  Default: *125* (frames).
- m length given to the CLIP template type in the rundown. If no time is given in the NRCS, this time is given to the element.
- Default: *125* (frames).

MOS

These settings are used when sending Mosart templates as MOS objects to the NRCS. MOS objects can then, for example be used to create items in a rundown (as an alternative to the legacy Mosart ActiveX implementation).
You can also send a MOS object containing lower-third information.

- **MOS server:** IP address or hostname of the main newsroom system's MOS gateway. See Notes below.
  Default: *NCSSERVER.*
- **MOS server backup:** IP address or hostname of the backup newsroom system's MOS gateway. See Notes below.
  Default: *Empty.*
- **NRCS ID:** Same form as mosID, but this value is the ID for the newsroom system. See Notes below.
  Default: *NCSSERVER.*
- **NRCS backup ID:** Same form as mosID, but this value is the ID for the backup newsroom system. See Notes below.
  Default: *Empty.*
- **MOS ID:** MOS identification of this instance of the Manus Administrator. Generic value is mosart.<galleryID>.<stationID>.mos.
  Default: *mosart.mos.*
- **Use backup server:** Enables an NRCS backup configuration. See Notes below.
  Default: *False.*
- **Force MOS protocol version:** Protocol version to use for NRCS communication. Default (blank) setting detects the version from the NRCS, otherwise, use a discrete value, for

Copyright © 2024 Vizrt

example, *2.8.3*.
Default: *Empty*.

- **MOS lower port:** The MOS protocol communicates on three ports: lower, upper and top. MosUpperPort is MosLowerPort + 1 and MosTopPort is MosLowerPort + 2.
  Default: *10540*
- **Mos server timeout:** Timeout information for MOS ports. Default: *0.0.0.0*.
  - **1. NRCS to Mosart lower port:** NRCS heartbeat sent to Viz Mosart. If the time between received heartbeats exceeds this value, the connection is displayed as *lost*.
    Value=*0* means no timeout.
    Default: *0* (seconds).
  - **2. NRCS to Mosart upper port**: NRCS heartbeat sent to Viz Mosart. If the time between received heartbeats exceeds this value, the connection is displayed as *lost*.
    Value=*0* means no timeout.
    Default: *0* (seconds).
  - **3. NRCS to Mosart high port:** NRCS heartbeat sent to Viz Mosart. If the time between received heartbeats exceeds this value, the connection is displayed as *lost*.
    Value=*0* means no timeout.
    Default: *0* (seconds).
  - **4. Mosart to NRCS:** Timeout value for an established connection between Viz Mosart and the NRCS. If the time from heartbeat is sent to response is received exceeds this value, the connection is displayed as *lost*. Value=*0* means no timeout.
    Default: *0* (seconds).
- **MOS server time zone:** Used when the time zone is not indicated by the NCS Server. Valid values are *Z* (meaning UTC) and *+/-hh[:mm]*, where *hh*and *mm* are double digit hours and optional minutes.
  Default: *Z* (UTC) unless another value is specified.
- **MOS schema:** A string used to identify Viz Mosart items in the rundown. Default: http://www.mosartmedialab.no/schema/mositem.dtd.
- **Default ready to air:** If the value is true, all MOS active rundowns are assumed to be ready to air.
  Default: *False*.
- **Templates to MOS objects:**
  - **Enabled:** Enables sending templates to the NRCS through MOS communication.
    Default: *Enabled*.
  - **GenerateUniqueObjId:** Create a unique identifying name when creating a new MOS Object. The NRCS uses this information to search for and reference the MOS Object.
  - **GroupedByType:** Enable to send all template types as one MOS object with the variants embedded in the objects. Disable to send all template variants as separate MOS objects.
  - **MergedClips:** Determines whether **PACKAGE** and **VOICEOVER** templates are merged as **CLIP** templates.
  - **SendAllTemplateSets:** Sends templates from all template sets to the NRCS when enabled. Disable to send only the default template set.
  - **TemplateChangeWanted:** Set to **true**, the NRCS will receive an update whenever a template is changed or deleted.

Copyright © 2024 Vizrt

- **Story Scope:** Displays either:
  - *Single* = A single story uses one row in the Viz Mosart GUI.
  - *Grouped* = A single story may span over multiple rows in the Viz Mosart GUI. Currently only supported for ENPS stories in *Grouped* mode that have the same story name but different story segments; these are then treated as the same story in Viz Mosart.
    Default: *Empty* (Single).
- **MOS reply with Connection MOS ID:** If set, any MOS message sent to the NRCS uses the connection MosId instead of the current mos object ID.
  Default: *False.*
- **Min. time between story update:** If no update is received from the NRCS within the minimum time stipulated, Viz Mosart forces a story update.
  Default: *1000* (milliseconds).
- **Item ID generator:** Default: *Empty*.
  Mosart generates internal IDs using a combination of rundown, story and item IDs. Some newsroom systems reuse item IDs between story updates. If a new object is inserted before an old object, Viz Mosart will therefore generate the same ID for the new object that the old object had, and the old object will be given a new ID.
  A custom ID generator can be assigned in **Item ID generator**.

Value	Description
(Empty)	This is the default value and uses the default ID generator
ItemAndObjectID	This ID generator appends the object ID to the default ID generator. This setting should be used for Dalet Newswire.

- **Ignore initial synchronization:** Ignores all roReq on startup when synchronizing with the NRCS.
  Default: *False.*
- **Ignore Send Item Status filter:** Lists regular expressions used to prevent sending MOS status back to the NRCS.
  The regular expressions are matched against MosIds of MOS objects. Set to '*' to disable the sending status for all MOS objects.
  Default: *Empty* (send status for all MOS objects).
- **Newsroom tag keep keywords**: A comma (or semicolon) separated list of CAPITALIZED words.
  For example:
  KEEP
  KEEP,INHERIT
  If a newsroom tag value is equal to any of these words, the newsroom tag will be assigned the value of the last Newsroom tag with the same name in the same story.
  Default: *KEEP*.

Copyright © 2024 Vizrt

See **Keep Newsroom tag (with the same name) within a story** in *section* Newsroom Tags . If "," (only a comma) is used all newsroomtags will be kept.

- **Item status unavailable values:** Lists status values used to identify NRCS content as invalid. Content as graphics or clip objects.
  Default: *NOT_READY, ERROR.*
- **Video server goto delay on retake**: The delay, measured in milliseconds, determining the time taken to position the video server timeline after cueing a pre-roll template for previewing recorded material while Viz Mosart is in Story Recorder mode.
  The configuration of this delay takes into account several factors, including clip length, video server response, and network connection, all of which can influence the cueing time.
  For further details, see section Story Recorder.

Read Speed

- **Read speed:** Number of words read per minute by the anchor. By default, words are defined as the parts of text separated by space characters. See **Use character for read speed word** below. This is applied to prompter text to determine the duration of an announcement and thus the length of the green camera bar and the green part of the voice-over bar in the Viz Mosart GUI.
  Default: *145* (words per minute).

> ⚠ **Note:** This is a *fallback read rate*, only used when the NRCS System fails to provide a **Read speed** of its own. This function can be tested by changing any **Read speed**="xx" to **Read speed**="" on an item in a local Manus file.

- **Use character for read speed word:** If set to *True*, each *character* counts as a word when calculating the spoken duration of prompter text, see Read speed above. This can be useful in languages like Chinese and Thai, where the space character is not used to separate words. If *False* (default), words are separated by space characters.
  Default: *False.*

Viz Media Sequencer

- **NCS server:** When Showmaker 1.0 is used as NRCS for Viz Mosart, this is where you configure the host address to the Media Sequencer Engine (MSE) that Showmaker is connected to.
- **MSE Playlist Panel Host:** The host address of the Media Sequencer Engine (MSE) used by the **MSE Playlist Panel** in the Viz Mosart UI
  Configure the host address of existing MSE host to activate the **MSE Playlist Panel**. The **Playlist Panel** offers users active (touch or click-sensitive) thumbnails of graphics held on an MSE.
  The Viz Mosart UI needs this MSE connection defined in order for the user to see the **Playlist Panel** (**View > Floating windows > Playlist Panel**). See *Viz Mosart User Guide*, section *Viz Mosart user Interface > MSE Playlist Panel*.

Copyright © 2024 Vizrt

Manus

- **Default Manus:** This value selects the default rundown to use. The **READYTOAIR** value selects and uses all rundowns indicated by the newsroom system as ready to air.
  Default: **READYTOAIR**
- **Manus expiration time:** Number of days to keep Manus Administrator files. Older files (than that number of days) are deleted when Manus Administrator is started.
  Default: 60 (files are deleted after 60 days).
  Legal values: *-1* (never deleted), *1 and above* (number of days until deleted from the manus folder).
- **Manus keep file pattern:** If Manus expiration time is given, this property allows a list of file search patterns for files to keep in the Manus Administrator directory.
  Files with names satisfying any of these patterns will be kept even if they are 'too old' according to the **Manus expiration time**
  For example, "*TEST**;DEMO??.xml" keeps all files starting with "*TEST*" and all *DEMO??.xml* files where '?' denotes a wild character.
  Default: *Empty* (no files to keep).
- **Test manus:** Rundown to be used for maintenance purposes.
- **Use the default selection of Manus:** Initializes the rundown given in the Settings Editor MOS setting automatically when enabled.
  Default: *True*

Crossover

- **Connection string:** The connection string of the crossover. For example: *controller=IP address, client=crossover*.
  Default: *Empty*.
- **Crossover autotake on switch offset:** Offset in milliseconds to pre-take an autotake when the next story item follows a crossover switch. A positive value triggers the autotake before the given duration of the story item. A negative value triggers the autotake after the given duration of the story item.
  Default: *0*.
- **Crossover SetNext on switch delay:** Delay of sending set next story from the server running the show when taking control.
  Default: *500* (milliseconds).

Template Database

- **Use template database:** When true, enables the Template DB functionality, so template type aliases can be exported from DB to `newsroomsettings.xml` at startup, and exporting `newsroomsettings.xml` to DB after saving `newsroomsettings.xml`.
  Default: False.

> ⚠ **Note:** Enter the **Connection string** before enabling **Use template database**.

Copyright © 2024 Vizrt

- **Name of DB provider:** The provider name for the **Template DB**. For example, *MySql.Data.MySqlClient* for MySQL.
  Default: *MySql.Data.MySqlClient.*
- **Connection string:** The connection string for the **Template DB**. For example, for MySQL: server=<hostname>; User Id=<user>;Password=<password>; database=mosarttemplatedb.
  Default: *server=localhost;User Id=root;database=mosarttemplatedb.*
- **Default template DB inserter:** The name to be used for the **insertedby** and **updatedby** columns in the Mosart Template Database.
  Default: *inserter.*

Logging

- **Log level**: Sets the detail level of logging to the log file. *N*ormal, *W*arnings, *E*rrors, Info, *D*etailed.
  Default: *0*.
- **Log MOS communication:** Enable logging of the communication to and from the NRCS.
  Default: *False.*
- **In use:** Enables or disables the logging of events from the Manus Administrator to the log file.
  Default: *True*.
- **WTC Level:** Sets the detail level of logging to the console. *Error, Message, External message, Details*.
  Default: *Details*.
- **Trace internally:** Enables or disables internal tracing to console, for debugging only.
  Default: *False*.
- **Use verbose logging:** Enables or disables verbose logging. If **In use** is set, using verbose the details sent to the log.
  Default: *False*.
- **Ignore verbose events filter:** Semicolon based list of events to ignore in the log when using verbose logging.
  Default: *Empty* (log all events)
- **Pass verbose events filter:** Semicolon based list of events to log when using verbose logging.
  Default: *Empty* (log all events)
- **MSMQ Log limit:** Value to identify when the application should dump the log queue to file.
  Default: *4023*
- **Path for MMLog:** The path where the **Viz Mosart Log** is stored.
  Default: *MosartLog*.

Vizrt Graphics

- **Enable**: Allow use of Vizrt graphics
- **MSE Host**: The hostname and port number (default *8580*) of the MSE used for playout.
- **Playlist name**: The name of the MSE playlist where Vizrt content from a Mosart show will be added and made ready for playout.
- **MSE Profile name**: Do not change this value.

Copyright © 2024 Vizrt

5.4.2 Notes

- If there is more than one main newsroom system, the setting parameters **MOS server**, **MOS server backup**, **NRCS ID**, and **NRCS backup ID** should contain space-separated lists of values, and the values must be in the same order.
- If some of the newsroom systems do not have a backup system, put them last in the list and leave the corresponding values blank for **MOS server backup** and **NRCS backup ID**.
- If two newsroom systems are backup for each other, both must be listed as both *main* and *backup*. If one system is backup for more than one *main* system, it must be listed as *backup* for each system.
- If different newsroom systems need to give Viz Mosart different MOS IDs, the settings parameter **MOS ID** should also be a space-separated list of values, in the same order as for **NRCS ID**.
- Use only a single space between list values. All values are case sensitive.
- Story Recorder status feedback to the NRCS is supported.

Copyright © 2024 Vizrt

5.5 Configuring The MSE Playlist Panel

The **MSE Playlist** panel is presented on the Viz Mosart UI, offering users active (touch or click-sensitive) thumbnails of graphics held on an MSE (Media Sequencer).

> ⚠ **Note**: For end user actions, please refer to the Viz Mosart User Guide, section **Viz Mosart User Interface > MSE Playlist Panel**.

- Setting User Access to MSE Playlist Panel
- Adding the Host Address of the Media Sequencer in the Viz Mosart UI

5.5.1 Managing the MSE Playlist Panel

Setting User Access to MSE Playlist Panel

To configure MSE Playlist Panel access using Manus Administrator

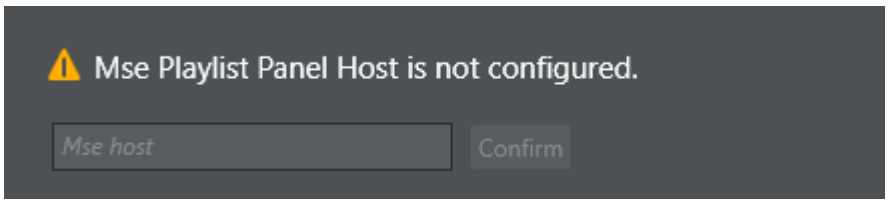
The Viz Mosart UI needs a connection to an active MSE. To activate the **MSE Playlist Panel**, you must configure which MSE to use.

1. Open the Viz Mosart Manus Administrator (console) window.
2. Type `settings`
3. Navigate to the **Viz Media Sequencer configuration**.
4. Modify the value of **Mse Playlist Panel Host** (either host name or IP address can be used).
5. Restart the console.

Adding the Host Address of the Media Sequencer in the Viz Mosart UI

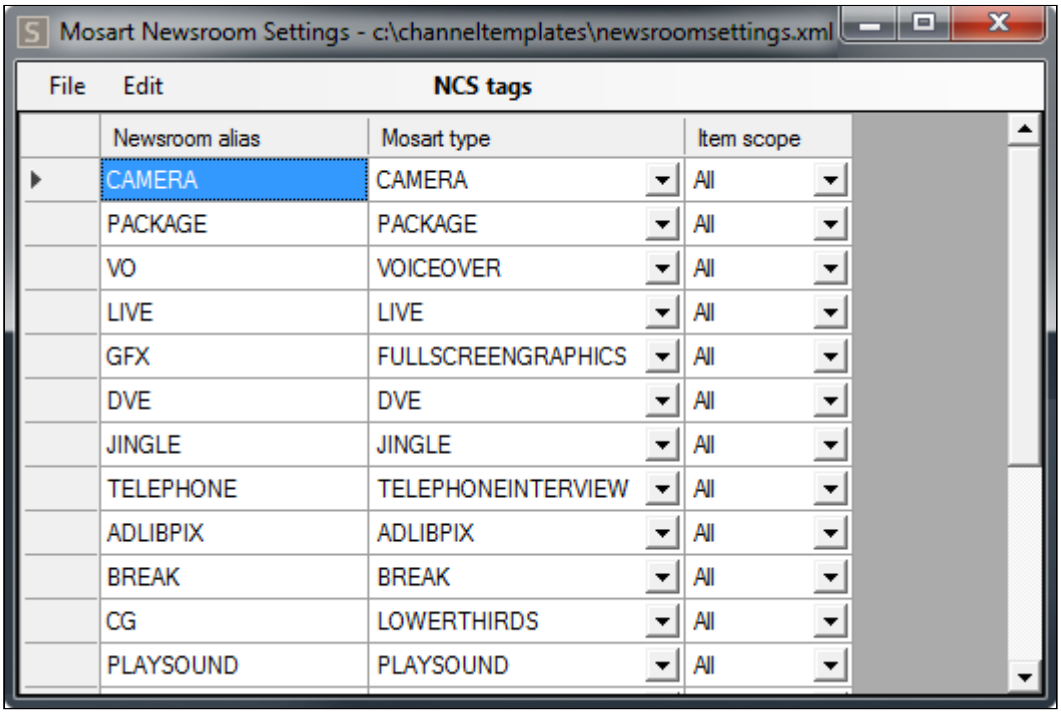
To configure MSE access using the Viz Mosart client

- If the **MSE Playlist** panel is visible in the Viz Mosart UI, you can add the host address of the Media Sequencer directly.
  Either host name or IP address can be used.



Copyright © 2024 Vizrt

5.6 Newsroom Settings Editor



This section contains the following topics:

- Working with the Newsroom settings
  - Configuring NRCS Tags
  - Newsroom Settings Editor - Edit Menu
  - Story External Metadata
  - Graphic Destination Letters
  - Properties
- Viz Mosart Object Structure
  - Vizrt MOS Item With Payload Data Nodes
  - Vizrt MOS Item Without Payload Data Nodes
- Viz Mosart and Viz Content Pilot (VCP) MOS Item Integration
- MOS ID Mapping
- Additional Newsroom Settings Examples
  - Defining Default Values for Lower Thirds
  - iNEWS: Overriding Default Field Names when Parsing Stories
  - iNEWS: Presenter Instructions

Copyright © 2024 Vizrt

5.6.1 Working with the Newsroom settings

Configuring NRCS Tags

To open the Newsroom Settings XML editor

1. Open the **Manus Administrator** console
2. Type `ns` followed by **ENTER**.
3. From the menu bar **Edit > NCS Tags**.

To save changes

Any changes made to the Newsroom Settings configuration must be saved:

1. Either
   a. Go to **File > Save**, or,
   b. Close the **Newsroom Settings Editor** and select **Save**.
2. Restart **Manus Administrator** for changes to apply.

Newsroom Settings Editor – Edit Menu

The **Edit** menu lists the configuration categories that are available.

- **NCS tags:** Menu for assigning the NRCS command aliases from the newsroom system to the corresponding Viz Mosart type. Each type can have multiple aliases. For example, *Camera* type can have aliases **CAM, CAMERA, KAM, KAMERA**, and so on.
- **NCS omit stories:** List of stories that should be ignored, and not displayed in the Viz Mosart GUI.
- **NCS omit tags:** (iNews only) List of iNews grommet commands that are ignored in Viz Mosart.
- **NCS breakline stories:** List of story names that are automatically converted to a Viz Mosart break line story when the story is tagged as *break*.
- **NCS accessory stories:** Viz Mosart interprets a story with this title as a story with accessories only.
- **NCS lowerthird mapping:** List for converting lower third template names from what is written in the NRCS (origin) to the required template (translated) in the graphics system.
  Transtype can be *begins, exact* or *contains.*
- **Timecode character:** Settings for textual analysis of timecode character, start character and split character.
- **Keywords:** Translation of **CUT** , **MIX** , **WIPE** and **EFFECT** transition keywords.
- **Lower third keep while background:** Lower third variants that are kept on air for the duration of the parent Viz Mosart template.
- **Lower third keep while story:** Lower third variants that are kept on air for the duration of the parent story.

Copyright © 2024 Vizrt

- **Lower third none auto out:** Lower third variants that are kept on air until replaced by another lower third of the same variant. Can be taken out with a keyboard shortcut (for example **HOME**).
- **Lower third keep until manual taken out:** Deprecated.
- **Pretake overlay on handler:** Name of the overlay handlers that automatically take the first lower third element in the next story on item types configured in Manus Administrator Settings, **Template allowing graphic pretake** (same configurations for MOS Settings Editor MOS or iNews*).*
  The graphic is taken immediately regardless of its start time.
  In the following example, only the graphics with destination **DSK**, **WALL** or **WALL_2** are pretaken.

  Example

  ```
  <pretakeoverlayhandlername>
        <handlername name="DSK" />
        <handlername name="WALL" />
        <handlername name="WALL_2" />
  </pretakeoverlayhandlername>
  ```

- **NCS device shortening:** (iNews only) A list of aliases for **clipname** or **clip_hirespath** values that can be used as newsroom tags in iNews grommets.
- **Parenthesis:** Start and end parenthesis. Content within these parentheses is translated to Viz Mosart commands. A value of (*) translates all commands in the form (** COMMAND ***) as the * (asterisk) value is interpreted as one or more asterisks.
- **Story External Metadata:** The NRCS can send external story values in a container called **mosExternalMetadata** (MEM). These can be translated to Viz Mosart content, as described in section Newsroom Settings Editor below.
- **Graphic Destination Letters:** Used to identify the output and behavior of graphic elements. See Newsroom Settings Editor below.

  > ⚠ **Note:** Only applies Vizrt VCP MOS items and the use of the Viz Mosart extended data element description.

- **Lower thirds type translation:** (ENPS only) CG commands. Use this to translate a CG item to a Viz Mosart primary story element
  - **templatetype**: CG template ID to translate
  - **type**: Viz Mosart primary story element
  - **variant**: Viz Mosart variant of the primary story element
  - **transtype**: begins, contains or exact. Describes the usage of the templatetype value when searching the CG item.
- **Newsroom tag to lowerthird channel map:** Translate the handler name of a graphic item into the value defined by this mapping in the newsroomsettings.
  For example, with the following configurations in newsroomsettings.xml:

Copyright © 2024 Vizrt

```
<inewstags>
<tags>
   <tag name="CAMERA" type="CAMERA" scope="All" />
   <tag name="PACKAGE" type="PACKAGE" scope="All" />
   <tag name="VOICEOVER" type="VOICEOVER" scope="All" />
   <tag name="VOICE OVER CLIPS" type="VOICEOVER" scope="All" />
   <tag name="CG" type="LOWERTHIRDS" scope="All" />
   <tag name="FLOAT" type="ADLIBPIX" scope="All" />
</tags>

<newsroomtypehandlernamemap>
   <tag name="CG" value="DSK" />
</newsroomtypehandlernamemap>

</inewstags>
```

when a graphic MOS object is parsed in Viz Mosart and found to be a CG command as defined in the <tag> node, Viz Mosart converts it to a Viz Mosart graphic item.
It then sets the **handler_name** attribute to *DSK*, based on the mapping defined in `<newsroomtypehandlernamemap>` node.
In this case, the graphic is redirected on the DSK channel of the configured graphic engine.

- **Omit text:** (iNews only) Omit text from iNEWS presenters area.
- **Properties:** (iNews only) iNEWS custom settings, see Overriding Default Field Names When Parsing iNEWS Stories below.
- **Ignore parenthesis:** (MOS-systems only) List of start and end parenthesis that encloses text ignored in Viz Mosart.
- **MOS Id mapping:** Enables Viz Mosart to recognize MOS elements from the NRCS and trigger specific Viz Mosart actions when encountered. Must not be changed by the user.
  See Newsroom Settings Editor below for a description of mapping MOS ID fields in the newsroomsettings.xml file.
- **MediaObject property mapping:** keywords specified during Viz Mosart installation for installation-specific behavior of Viz Mosart. Should not be changed by the user.

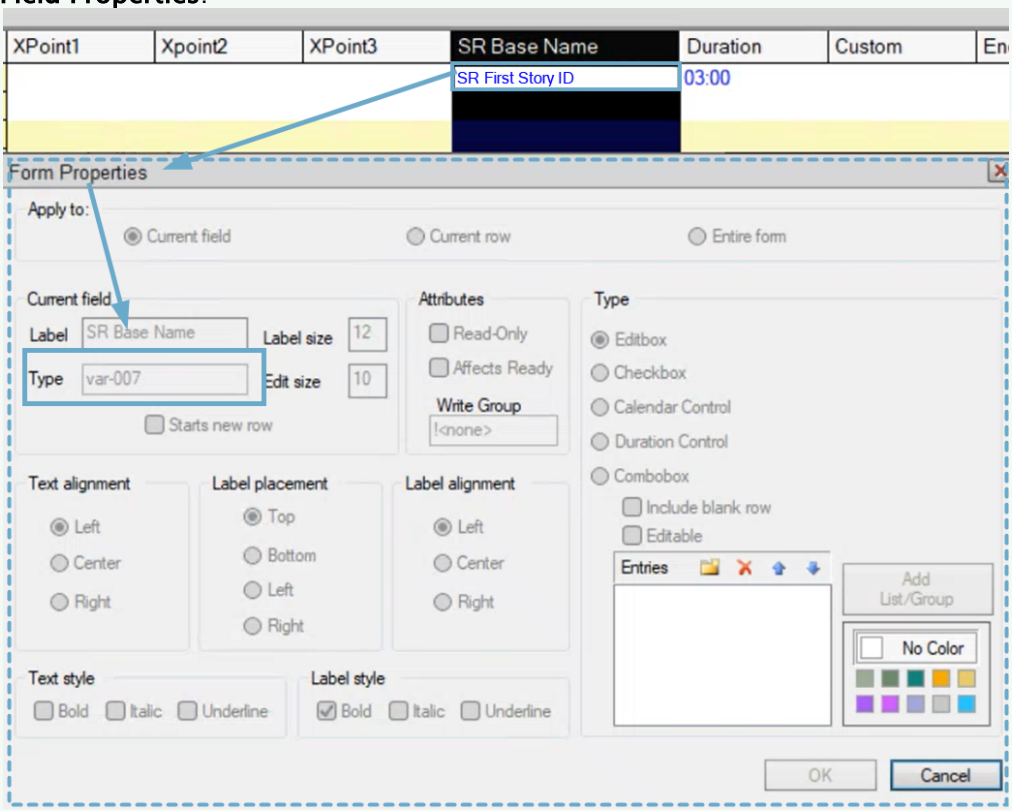
Story External Metadata

The NRCS can share story values in a container called **MosExternalMetadata** (MEM)**.** These values can then be translated to Viz Mosart content. The values are:

- **mostagname:** An XML element, often originating from a column name in the NRCS.

Copyright © 2024 Vizrt

You can derive this NRCS-assigned name for example in *iNews*, by **Right-clicking > Field Properties**:



In this example, the user friendly *SR Base Name* is stored internally in iNews as *var-007*.

- **mos_value:** This is the text that appears in the XML element **mostagname** (set in the NRCS).
- **Mosart Action:**
  - **template_type:** Use a value from the NRCS to create a new Mosart item in the rundown story. The value from the NRCS can be a newsroom tag of TEXT item scope. Newsroom tags are configured in `newsroomsettings.xml`, (for example, *CAM*, *KAM*, *KAMERA*, *PKG*).
    You can also use the Viz Mosart type number, where for example, *0* is for Camera, *1* for Package. If the value from NRCS is *empty*, no item is created.
    Example:

    ```
    <mostag mostagname="var-1" mosart_action="template_type" format_type="Stri
    ng" />
    ```

    A new Viz Mosart item of type given in the NRCS column **var-1** is created.
  - **item_variable:** Uses the value from NRCS to set a field on the *first* primary item found in the story. Non-empty values overwrite any existing fields.
    Example:

```
<mostag mostagname="var-2" mosart_action="item_variable" action_value="clip_hirespath" format_type="String" />
<mostag mostagname="var-3" mosart_action="item_variable" action_value="output_preview" format_type="String" />
```

The fields **clip_hirespath** and **output_preview** are set on the first primary item found in the story, with the values taken from the columns **var-2** and **var-3** configured in NRCS.

- **item_duration:** Use the value from the NRCS column as the duration for an item or several items in the story.
  - If **action_value** is empty,
    - the duration is applied to the *first primary* item in the story.
  - If **action_value** contains the template type of a Viz Mosart item (given as a newsroomtag defined in the NRCS tags from Newsroomsettings or as a number (for example *CAM* or *0*) or a list of template types separated by comma,
    - the duration applies to *all* items of that type.
    For example: **action_value**=*CAM* or **action_value=***PKG, JINGLE* or **action_value=***0* or **action_value=***1,JINGLE*.
  - If **action_value** contains pairs `<type><separator><variant>` or a list of such pairs separated by a comma,
    - duration applies to those specific items in the story.
    For example: **action_value=CAM=***1* or **action_value=***CAM|1,PKG*).
    The `<separator>` can be "**=**", "**|**", "**:**" , "**-**" or "**;**".
    Use Format type to select if the value is given as seconds (integer), frames (integer) or as a time code.
- **template_variant:** Use the value from the NRCS to add or change the variant of the first primary item found in the story.
- **template_transition:** Use the value from the NRCS to add effect transitions to the first primary item found in the story.
  This also provides a good overview in the newsroom about which effects are used between the stories.
  For example:

```
<mostag mostagname="TranFromNcs" mos_value="SLIDE" format_type="String" mosart_action="template_transition" action_value="MIX,24" />
```

Use the value from the NRCS to add effect transitions to the *first primary* item found in the story.
The effect transition is given in the format `<effect_name><separator><input>`, where `<separator>` can be white space, bar (**|**), comma(**,**), semicolon(**;**) or colon (**:**).
The `<effect_name>` can be *EFFECT, MIX, WIPE* or their translations as configured in the **Keywords** section in Newsroomsettings.
For example, if *EFFECT* is mapped as *EFFEKT* in the **Keywords** section, we can add the following effect to the Viz Mosart item: *Effekt 2*.

Copyright © 2024 Vizrt

- **story_variable:** Use the value from the NRCS to set a field on all items (excluding prompters and some secondaries with type > 300) found in the story. Non-empty values do *not* overwrite existing fields.
- **story_duration:** Use the value from the NRCS as the planned duration for this story. Use **Format** type to select if the value is given as seconds (integer) or as a time code.
- **directtake_pre_story:** Executes the direct take template given in the **action_value** at the start of the story (in the switch from the previous to current story).
- **directtake_post_story:** Executes the direct take template given in the **action_value** at the end of the story (in the switch from the current to the next story).
- **guimarker_X:** Translates the set of **mostagname** and **mos_value** to a marker in the GUI, valid **action_values** are *blank , 0* (green), *1* (orange) and *2* (red).
- **device_property:** Sends a key-value pair to the device driver. For example, for camera robots this can be used to adjust speed and camera positions.
- **story_endphrase:** Use the value from the NRCS to override the **endphrase** attribute for the *last* primary item in a Viz Mosart story.
- **segment_duration:** Use the value from the NRCS to set the duration of a group of stories.
- **words_per_minute** *:* Sets the read rate for a story.
- **back_time:** Sets the **backtime** attribute for a story (the exact time when a story must start in order for the show to remain on schedule.). This is used to set a next break time of the show. The **break** attribute must be set to *true.*
- **cume_time:** Sets the **cumetime** attribute for a story - the amount of airtime required from the beginning of the show up to a certain point in the show in order for the show to remain on-schedule. This is used to set a next break duration of the show. The **break** attribute must be set to *true.*
- **category:** Sets the **category** attribute for a story used to identify a story category such as sports, news, business etc.
- **field_value_X** *:* Overrides the field value of a Viz Mosart item with the value stipulated in the NRCS**.** This applies only to the first primary item in a story and the field must already exist for that item. For example, if the first primary item in a story is a Package with the following fields:

```
<fields>
    <field name="clip_description" value="" default="" fieldtype="TEXT"
keylist="" />
    <field name="clip_hirespath" value="" default="" fieldtype="TEXT"
keylist="" />
    <field name="metadata_lookuppath" value="" default="" fieldtype="TEXT"
 keylist="" />
</fields>
```

and the following mappings are given in newsroom settings:

```
<mostag mostagname="ClipDescrCol" mosart_action="field_value_1"
action_value="clip_description" format_type="String" />
```

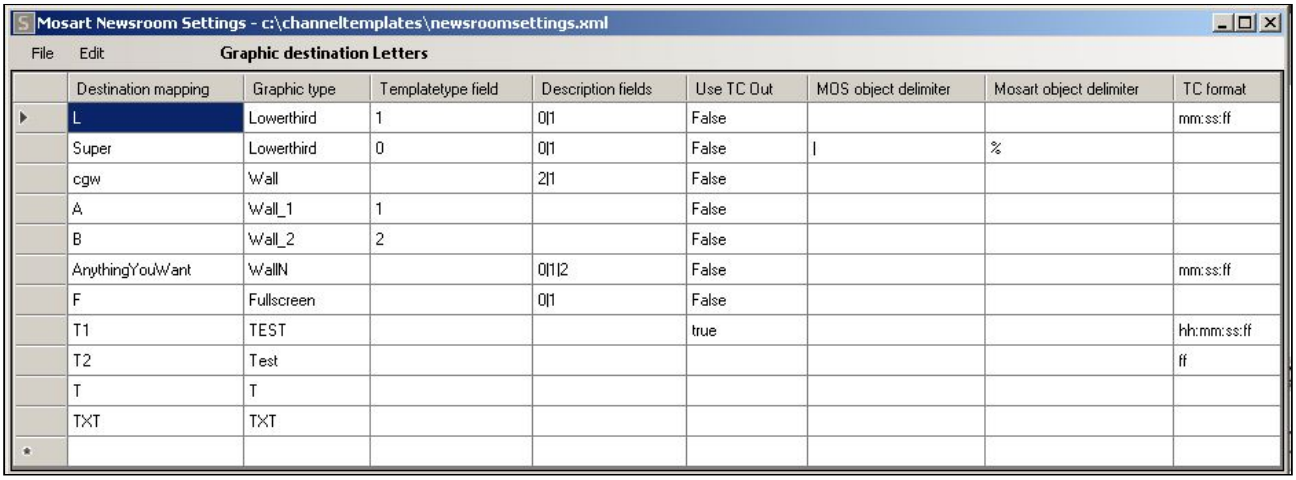
```
<mostag mostagname="ClipIdCol" mosart_action="field_value_2"  action_value="
clip_hirespath=MyClip" mos_value="test" format_type="String" />
```

then the field is set to the value given in the NRCS in the column with id **ClipDescrCol** and the field **clip_hirespath** are set to the value **MyClip** if the value from column **ClipIdCol** in the NRCS is *test*.

> **Note:** The attribute **action_value** for **<mostag>** node in newsroom setting can contain the *field name* or pairs of *field name=field value* separated by comma, colon, semi-colon, dash, equal or bar.

- **asruninfo:** Sets the attribute **asruninfo** for all items in a Viz Mosart story. This is used to store specific information that is stored in the AsRunLog.

Graphic Destination Letters



Destination mapping	Graphic type	Templatetype field	Description fields	Use TC Out	MOS object delimiter	Mosart object delimiter	TC format
L	Lowerthird	1	0|1	False			mm:ss:ff
Super	Lowerthird	0	0|1	False	|	%	
cgw	Wall		2|1	False			
A	Wall_1	1		False			
B	Wall_2	2		False			
AnythingYouWant	WallN		0|1|2	False			mm:ss:ff
F	Fullscreen		0|1	False			
T1	TEST			true			hh:mm:ss:ff
T2	Test						ff
T	T						
TXT	TXT						

Graphic destination letters identify output and behavior for graphics elements. Use them to parse a Vizrt MOS object and obtain a Viz Mosart object.

Properties

Each row contains the following columns:

- **Destination mapping:** The mapping defined in the Viz Mosart command from Viz super.
- **Graphic type:** Used to identify the output for the graphics elements, either on predefined channels:
  - **Fullscreen**: Full screen graphics for FULL output (channel)
  - **Lowerthird**: Overlays for DSK output
  - **Wall**: Overlays for WALL output
  - **WallN**: Overlays for outputs WALL_1, WALL_2, WALL_3 etc.
  - **Ticker**: Ticker for TICKER output

  or on custom channels (so any string can be set as graphic type).
- **Templatetypefield:** Index of the description field which is set as the TemplateType. (the value of the description field found at the specified index). The index is 0-based. Default:

Copyright © 2024 Vizrt

empty.
For example: If the **MosAbstract** is: `<mosAbstract>field1/field2/field3/field4/Mosart=S|00:04|00:10</mosAbstract>`, and you set `Templatetypefield=1`, then the `templatetype` attribute from the Viz Mosart object is set to `field2`.

- **Descriptionfields:** A list of indexes of the fields which should be shown in the description of a Mosart story item. These fields are shown in the Viz Mosart GUI separated by the character given in the **outputtxtdelimiter** setting. The indexes are 0-based and are separated using a pipe '|' character.
  For example: **descriptionfields=**"*0|1*". Default: *empty*.
- **Use TC Out:** Specifies if the duration should be used as **tc_out**.
- **Inputtxtdelimiter:** Vizrt MOS object delimiter. Default: '**/**' (slash)
- **Outputtxtdelimiter:** Viz Mosart object delimiter. Default: '**/**' (slash)
- **Mask:** Specifies the timecode format of the graphics starting time, end time or duration. Default: **mm:ss**. Example of formats: *hh:mm*, *hh:mm:ss*, *mm:ss:ff*, *mm:ss*, *ss:ff*, *ff*.

Mapping example:

```
<graphicdestinationletters>
  <graphicdestinationletter destinationmapping="L" type="Lowerthird"
templatetypefield="1" descriptionfields="0|1" use_tc_out="False" inputtxtdelimiter=""
 outputtxtdelimiter="" mask="mm:ss:ff"/>
  <graphicdestinationletter destinationmapping="Super" type="Lowerthird"
templatetypefield="0" descriptionfields="0|1" use_tc_out="False" inputtxtdelimiter="|
" outputtxtdelimiter="%" mask=""/>
  <graphicdestinationletter destinationmapping="cgw" type="Wall" templatetypefield=""
 descriptionfields="2|1" use_tc_out="False" inputtxtdelimiter="" outputtxtdelimiter="
"/>
  <graphicdestinationletter destinationmapping="A" type="Wall_1" templatetypefield="1
" use_tc_out="False" inputtxtdelimiter="" outputtxtdelimiter=""/>
  <graphicdestinationletter destinationmapping="B" type="Wall_2" templatetypefield="2
" use_tc_out="False" inputtxtdelimiter="" outputtxtdelimiter=""/>
  <graphicdestinationletter destinationmapping="AnythingYouWant" type="WallN"
templatetypefield="" descriptionfields="0|1|2" use_tc_out="False" inputtxtdelimiter="
" outputtxtdelimiter="" mask="mm:ss:ff"/>
  <graphicdestinationletter destinationmapping="F" type="Fullscreen"
templatetypefield="" descriptionfields="0|1" use_tc_out="False" inputtxtdelimiter=""
outputtxtdelimiter=""/>
  <graphicdestinationletter destinationmapping="T1" type="TEST" use_tc_out="true"
mask="hh:mm:ss:ff"/>
  <graphicdestinationletter destinationmapping="T2" type="Test" mask="ff"/>
  <graphicdestinationletter destinationmapping="T" type="T"/>
  <graphicdestinationletter destinationmapping="TXT" type="TXT"/>
</graphicdestinationletters>
```

Walls
There are two ways to support a configurable number of walls:

1. By having a general rule in the graphics destination mappings where the graphics type must be *WallN.* The **destionationmapping** attribute can be set to any string of alphanumeric characters.

```
<graphicdestinationletter destinationmapping="Walls" type="WallN"
templatetypefield="" descriptionfields="-1" use_tc_out="False"
inputtxtdelimiter="" outputtxtdelimiter="" mask="mm:ss"/>
```

   In this case, the Vizrt graphic objects must contain a Viz Mosart string command ending in a number (this provides the wall number).
   For example:
   `Mosart=cgw1|00:00|00:05`
   or
   `Mosart=something2|M|S`
   or
   `Mosart=wall3|00:00|B`
   or `Mosart=W4|00:04|00:10`
   and so on.
   Given these examples, the destination for the first graphic is **WALL_1**, followed by **WALL_2**, **WALL_3** and **WALL_4**, respectively.
2. By not ending the Viz Mosart string destination mapping with a number.
   For example:
   `Mosart=B|00:00|00:05, Mosart=Z|00:00|00:05, Mosart=Wall|00:00|00:05, Mosart=Something|00:00|00:05`
   and so on.
   In this case, the configuration in **newsroomsettings** must specify the wall number in the graphics type. For the Mosart string examples given above, the configuration can therefore be:

```
<graphicdestinationletter destinationmapping="B" type="Wall_2"
templatetypefield="" descriptionfields="-1" use_tc_out="False"
inputtxtdelimiter="" outputtxtdelimiter=""  mask="mm:ss"/>
<graphicdestinationletter destinationmapping="Z" type="Wall_3"
templatetypefield="" descriptionfields="-1" use_tc_out="False"
inputtxtdelimiter="" outputtxtdelimiter="" mask="mm:ss" />
<graphicdestinationletter destinationmapping="Wall" type="Wall_4"
templatetypefield="" descriptionfields="-1" use_tc_out="False"
inputtxtdelimiter="" outputtxtdelimiter="" mask="mm:ss"/>
<graphicdestinationletter destinationmapping="Something" type="Wall_5"templatet
ypefield="" descriptionfields="-1" use_tc_out="False" inputtxtdelimiter=""
outputtxtdelimiter=""   mask="mm:ss"/>
```

   Overlays with
   `Mosart=B|00:00|00:05`
   are then redirected to **WALL_2**, those with `Mosart=Z|00:00|00:05` to **WALL_3** and so on.

Copyright © 2024 Vizrt

As well as redirecting graphics to predefined destinations like *DSK, WALL, WALL_i, FULL*, the graphic destination letters rules can be used for any desired destination.
For example:

```
<graphicdestinationletter destinationmapping="T" type="TEST" templatetypefield=""
descriptionfields="-1" use_tc_out="False" inputtxtdelimiter="" outputtxtdelimiter=""
mask="mm:ss"/>
```

and a Vizrt graphic with Mosart string `T|00:00|00:06` sends the graphic for playout on the channel *TEST*.

5.6.2 Viz Mosart Object Structure

When building a graphic Viz Mosart object from a Vizrt MOS object, an XML structure is created:

```
<item type="100" slug="L3-Anna Smith|Vizrt reporter|Mosart=L|00:00|00:06" source="1"
index="100___2_5" idref="5" templatetype="TIMECODE-DSK" status="0" error="0" in="0"
dur="150" pin="0" pdur="150" rdur="0" externaleffect="" intimeline="true" date_0=""
accessory="False" static="false" endfrase="" mosid="PILOT" objid="121" ismoselement="
true" use_graphics_id="true" graphics_id="121" handler_name="DSK" graphics_out_on="TI
MECODE" description="(DSK) - L3-Anna Smith|Vizrt reporter|Mosart=L|00:00|00:06"
auto_continue="false">
  <fields>
    <field name="graphics_description" fieldtype="TEXT" value="L3-Anna Smith|Vizrt
reporter|Mosart=L|00:00|00:06"/>
    <field name="graphics_id" fieldtype="TEXT" value="121"/>
    <field name="tc_dur" fieldtype="TIMECODE" inputmask="mm:ss" default="00:00" value="
00:06"/>
    <field name="continuecount" value="-1" fieldtype="TEXT"/>
    <field name="payloaduri" value="http://bgoemo:8177/dataelements/121/payload"
fieldtype="TEXT"/>
    <field name="thumbnailuri" value="http://bgoemo:8177/dataelements/121/thumb"
fieldtype="TEXT"/>
    <field name="tc_in" fieldtype="TIMECODE" inputmask="mm:ss" default="00:00" value="
00:00"/>
  </fields>
</item>
```

The graphic *slug* is used to visualize the graphics in the Viz Mosart GUI **Assets** window or in the **Overlay Graphics** interface, for example. The slug is taken from **MosAbstract** found in the Vizrt MOS object and the fields delimited by the inputtxtdelimiter are replaced with outputtxtdelimiter.

Viz Mosart can receive several types of Vizrt MOS graphic objects from the NRCS, including a

- Vizrt MOS Item With Payload Data Nodes and a
- Vizrt MOS Item Without Payload Data Nodes.

Copyright © 2024 Vizrt

Vizrt MOS Item With Payload Data Nodes

If a Vizrt MOS item contains `<mosPayload>` with data entries, then the **graphics_description** is built from these data nodes delimited by **outputtxtdelimiter**.

The property **inputtxtdelimiter** is used only to build the *slug*, not the description.

If **descriptionfields** is empty, then the **graphics_description** attribute from the Viz Mosart item (created from the Vizrt MOS item) is set to the *slug* value (created from **MosAbstract**, see above).

If the field is populated, the slug is set to the description created from data nodes entries where only the data nodes from the positions specified in the **descriptionfields** are added and separated by **outputtxtdelimiter**.

Example 1:

Using a Vizrt MOS item containing the following values:

```
<mosExternalMetadata>
  <mosScope>OBJECT</mosScope>
  <mosSchema>http://www.vizrt.com/mosObj/data</mosSchema>
  <mosPayload>
    <data>
      <entry name="data">
        <entry name="">
          <entry name="" type="widestring">Navn/Titel/Sted</entry>
        </entry>
        <entry name="1" description="Name">
          <entry name="1" description="Name" type="richtext" upper="true" singleline="
true" location="2/3/1/1">L3-Anna Smith</entry>
        </entry>
        <entry name="2" description="Title">
          <entry name="2" description="Title" type="richtext" upper="true" singleline="
true" location="2/2/2/1">Vizrt reporter</entry>
        </entry>
        <entry name="Mosart">
          <entry name="Mosart" type="widestring">Mosart=L|00:00|00:06 </entry>
        </entry>
      </entry>
    </data>
  </mosPayload>
</mosExternalMetadata>
```

apply the following mapping:

```
<graphicdestinationletters> <graphicdestinationletter destinationmapping="L" type="Lo
werthird" templatetypefield="" descriptionfields="" inputtxtdelimiter=""
outputtxtdelimiter="|" /> </graphicdestinationletters>
```

Note that **inputtxtdelimiter** is empty, so the default ‘ `/` ’ is used.

Copyright © 2024 Vizrt

The result is the following Mosart item. Note the `slug` and `description` attributes and `graphics_description` field:

```
<item type="100" slug="L3-Anna Smith|Vizrt reporter|Mosart=L|00:00|00:06" source="1"
index="100___2_5" idref="5" templatetype="TIMECODE-DSK" status="0" error="0" in="0"
dur="150" pin="0" pdur="150" rdur="0" externaleffect="" intimeline="true" date_0=""
accessory="False" static="false" endfrase="" mosid="PILOT" objid="121" ismoselement="
true" use_graphics_id="true" graphics_id="121" handler_name="DSK" graphics_out_on="TI
MECODE" description="(DSK) - L3-Anna Smith|Vizrt reporter|Mosart=L|00:00|00:06"
auto_continue="false">
  <fields>
    <field name="graphics_description" fieldtype="TEXT" value="L3-Anna Smith|Vizrt
reporter|Mosart=L|00:00|00:06"/>
    <field name="graphics_id" fieldtype="TEXT" value="121"/>
    <field name="tc_dur" fieldtype="TIMECODE" inputmask="mm:ss" default="00:00" value="
00:06"/>
    <field name="continuecount" value="-1" fieldtype="TEXT"/>
    <field name="payloaduri" value="http://bgoemo:8177/dataelements/121/payload"
fieldtype="TEXT"/>
    <field name="thumbnailuri" value="http://bgoemo:8177/dataelements/121/thumb"
fieldtype="TEXT"/>
    <field name="tc_in" fieldtype="TIMECODE" inputmask="mm:ss" default="00:00" value="
00:00"/>
  </fields>
</item>
```

Example 2:

Using the same Vizrt MOS item as above, but with the following mapping:

```
<graphicdestinationletter destinationmapping="L" type="Lowerthird" templatetypefield="
1" descriptionfields="0|1" outputtxtdelimiter="%" />
```

In the result below you can see that the delimiters are now '**%**' instead of '|'. Note that **templatetype** is set to the value found at index 1.

```
<item type="100" slug="L3-Anna Smith%Vizrt reporter" source="1" index="100___2_5"
idref="5" templatetype="Vizrt reporter" status="0" error="0" in="0" dur="150" pin="0"
 pdur="150" rdur="0" externaleffect="" intimeline="true" date_0="" accessory="False"
static="false" endfrase="" mosid="PILOT" objid="121" ismoselement="true"
use_graphics_id="true" graphics_id="121" handler_name="DSK" graphics_out_on="TIMECODE
" description="(DSK) - banner%Anna Smith" auto_continue="false">
  <fields>
    <field name="graphics_description" fieldtype="TEXT" value="L3-Anna Smith%Vizrt
reporter" />
    ...
  </fields>
</item>
```

Copyright © 2024 Vizrt

Vizrt MOS Item Without Payload Data Nodes

MosAbstract

If the Vizrt MOS item does not contain data nodes, but has **MosAbstract** set, then the slug and the description are built from **MosAbstract.**

In this case, **inputtxtdelimiter** is taken into consideration when building both the slug and description.

Similarly, if **descriptionfields** is empty, the **graphics_description** attribute from the Mosart item (created from the Vizrt MOS item) is set to the slug value (created from **MosAbstract**, as explained above).

Otherwise, the slug is set to the description built as explained above from MosAbstract where only the fields from the positions specified in **descriptionfields** are added and separated by **outputtxtdelimiter.**

Example 3:

Using the following Vizrt MOS Item:

```
<mosAbstract>00:00 | Super Tema + info | 26 | 1:Tema | 2:This is what the item is all
about |Mosart=S|M|00:04</mosAbstract>
```

apply the following mapping:

```
<graphicdestinationletters> <graphicdestinationletter destinationmapping="S" type="Lo
werthird" templatetypefield="" descriptionfields="1|3" inputtxtdelimiter="|"
outputtxtdelimiter="%" /> </graphicdestinationletters>"
```

The result is:

```
<item type="100" slug="Super Tema + info%1:Tema" source="1" index="100___2_2" idref="
2" templatetype="TIMECODE-DSK" status="1" error="0" in="0" dur="100" pin="0" pdur="10
0" rdur="0" externaleffect="" intimeline="true" date_0="" accessory="False" static="f
alse" endfrase="" mosid="VIZ.NPRO.MOS" objid="1529718" ismoselement="true"
use_graphics_id="true" graphics_id="1529718" handler_name="DSK" graphics_out_on="TIME
CODE" description="(DSK) - Super Tema + info%1:Tema" auto_continue="false">
  <fields>
    <field name="graphics_description" fieldtype="TEXT" value="Super Tema +
info%1:Tema"/>
    <field name="graphics_id" fieldtype="TEXT" value="1529718"/>
    <field name="tc_dur" fieldtype="TIMECODE" inputmask="mm:ss" default="00:00" value="
00:04"/>
    <field name="continuecount" value="-1" fieldtype="TEXT"/>
    <field name="payloaduri" value="n/a" fieldtype="TEXT"/>
    <field name="thumbnailuri" value="n/a" fieldtype="TEXT"/>
  </fields>
```

Copyright © 2024 Vizrt

```
</item>
```

5.6.3 Viz Mosart and Viz Content Pilot (VCP) MOS Item Integration

Vizrt MOS items do not contain any information about playout channel destination. By adding an optional **Mosart-part** in the **Vizrt data description**, properties for channel destination, automatic or manual play-out and timing information can be extracted from the MOS item.
This section presents the Vizrt MOS item metadata required to allow Viz Mosart to separate play-out channel and extract timing information.

Basic requirements

Viz Mosart requires details on whether the graphics element is a lower third, a graphics played out on a video wall engine, or a full screen graphics. Any graphic requires the following properties:

- **Destination**: lower third, video wall or full-screen graphics
- **For lower thirds/video walls**:
  - Play-out: manual or automatic
  - For automatic play-out: in time
  - Duration/out time or the special durations background end, story end or open end
- **Mosart item** (optional): Stored at the end of the description of the data element or as a specified field in the **mosExternalMetadata** section of the MOS item and is formatted as follows:
  Mosart={destination}|{in behaviour}|{out behaviour}|{X}|{F}
  where
  - {destination} is an alphanumeric string specifying the destination of the graphic. Suggested values are **L**=lower third, **W**=wall, **F**=full screen graphics
  - {in behaviour} is
    - either **M** for manual play-out
    - or a time code formatted as **mm:ss** (minutes, seconds) as in time
  - {out behaviour} is
    - either a time code formatted as **mm:ss** (minutes, seconds) as duration
    - or a single character field with values **B**, **S** or **O** where **B**=background end, **S**=story end and **O**=open end
  - X is used only for lower thirds to ignore “Replace TakeOut Logic” meaning that when taking and taking out the item, the replace logic is ignored if the value is set on the item
  - F is used only for lower thirds to force “Replace TakeOut Logic” for graphics coming from different templates with same layers (when used, it practically ignores the “O” state for a layer). Not to be confused with destination letter “F” often used for full screen graphics.

The rules which define the output and behavior for graphics elements can be configured in Newsroom Settings Editor.

Field Example:

The following table contains examples of the **Mosart= part** of the Vizrt data element description in the value **cg_gordon_brown/prime minister/Mosart=L|00:02|00:05**. The data element is played out as a lower third, automatically in at two seconds with five seconds duration.

L|00:02|00:05	Lower third, automatically in at 2 seconds with 5 seconds duration
L|00:02|S	Lower third, automatically in at 2 seconds with duration equal to the length of the story (taken out at the switch from one story to the next)
L|00:00|O	Lower third, automatically in at 0 seconds, never taken out (except for when replaced by another item or manually from the operator)
L|M|00:05	Lower third, manual play-out, duration 5 seconds
L|00:00|00:10|X	Lower third, automatically in at 0 seconds with 10 seconds duration and Replace TakeOut Logic is ignored
W|00:00|B	Video wall, automatically in at 0 seconds, out when switching from one story element to another
F	Full screen graphics (note that this letter is usually used for full screen graphics, but any combination of alphanumeric characters can be used to identify a fullscreengraphic)

If no fields are given, the element defaults to a full screen graphics element.

VCP template requirements

The Mosart field is supported when embedded as a part of the VCP data element description or as a dedicated field in the **mosExternalMetadata** section.

Storing the Mosart field in the VCP data element description

When saving the VCP data element, the **Mosart** field shall be appended to the stored database description (the name of the data element).

Storing the Mosart field in the mosExternalMetadata section of the VCP data element

This method is only supported if the **mosExternalMetadata** data section is enabled for the VCP Template Filler ActiveX. See Appendix A for configuring the ActiveX to include this section in the MOS object.

The scene needs a **ControlObject.** The Mosart field should be stored in a hidden text field in the template, and the text field shall use **Mosart** in the **ControlObjectName** property. The Mosart field

Copyright © 2024 Vizrt

can either be included in the scene or added manually after importing the scene to the Viz Template Wizard.

VB script Example

The following VB script can be used in a Vizrt Template Wizard template and returns a properly formatted Mosart field string:

```
'' Method that returns a properly formatted Mosart description
Function GetMosartDescriptionPart(AType, AVariant, AOperation, ATCIn, ATCOut)
  'Template variant
  Dim FType, FVariant, FDefaultVariant, FDefaultOperation
  Select Case AType
    Case "LOWER"
       FType = "L" 'LOWER (lower third, over shoulder gfx)
       If AVariant = "" Then
         AVariant = "AUTOOUT"
       End If
    Case "WALL"
       FType = "W" 'LOWER (lower third, over shoulder gfx)
       If AVariant = "" Then
         AVariant = "OPENEND"
       End If
    Case Else
       'FType = "F" 'FULL (fullscreen)
       GetMosartDescriptionPart = "Mosart=F"
       Exit Function
  End Select

  'Take out logic (Only applies for lower thirds)
  FVariant = TranslateVariant(AVariant)

  'Operation selection
  Select Case AOperation
    Case "MANUAL"
      FOperation = "M" 'FULL
      FTCIn = FOperation
      If (FVariant="A") Then
        FTCOut = ATCOut
      Else
        FTCOut = FVariant
      End If
    case else 'case "AUTO+"
      FOperation = "A" 'AUTO+
      FTCIn = ATCIn
      If (FVariant="A") Then
        FTCOut = ATCOut
      Else
        FTCOut = FVariant
      End If
```

Copyright © 2024 Vizrt

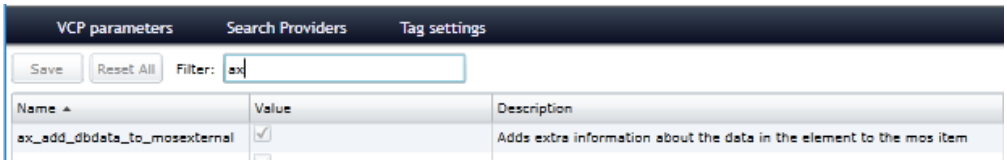
```
  end select

  GetMosartDescriptionPart = ToMosartProperty(FType, FTCIn, FTCOut)
End Function

Function ToMosartProperty(AType, ATCIn, ATCOut)
  splitChar = "|"
  Dim vals
  'vals = vbNewLine
  If Not AType = "" Then
    vals = vals & "Mosart="
    vals = vals & AType
    If Not ATCIn = "" Then
      vals = vals & splitChar & ATCIn
      If Not ATCOut = "" Then
        vals = vals & splitChar & ATCOut
      End If
    End If
  End If
  ToMosartProperty = vals
End Function

Function TranslateVariant(AVariant)
  select case AVariant
    case "OPENEND"
      TranslateVariant = "O" 'OPENEND
    case "STORYEND"
      TranslateVariant = "S" 'STORYEND
    case "BACKGROUNDEND"
      TranslateVariant = "B" 'BACKGROUNDEND
    case else 'case "AUTOOUT"
      TranslateVariant = "A" 'AUTOOUT
  end select
End Function
```

Viz Pilot plugin Prerequisites

To enable the **mosExternalMetadata** section in the Viz Pilot News (Active X) plugin, the system administrator must also enable a field in the VCP parameters list in the Viz Pilot database.

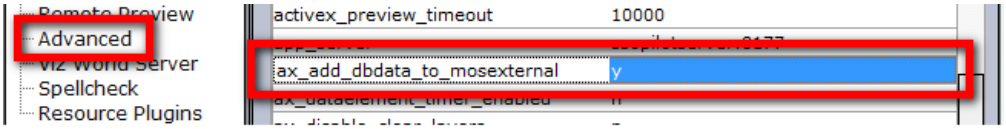
If you Viz Pilot 5.7 or later, this setting is available in the Viz Pilot Data Server settings page found in:

`http://[hostname of PDS]:8177/settings#Params`



In the **Value** column, the checkbox should be checked (set to true).

If you have a Viz Pilot version older than 5.7 this is easily achieved by opening the Preferences editor in the **Viz Content Pilot** (**Options** → **Preferences** → **Advanced**), by adding or editing the field called **ax_add_dbdata_to_mosexternal** setting the value to **y**.



5.6.4 MOS ID Mapping

A **MOS Id** mapping can be performed in the `newsroomsettings.xml` file, as follows:

```
<mosids>
    <tag name="" value="" fieldMapping="" appendContent="" embeddedMosartItem=""
 keywords="" />
</mosids>
```

Where:

- **name:** The MOS ID that identifies the MOS elements coming from the NRCS**.**
- **value:** The mapping ID used to identify a mapping for MOS element fields to be parsed. The MOS element's fields to be parsed are defined using `<mosmaps> (/mmConstants/xmltemplates/mosmaps/mosmap/).`
- **fieldMapping:** Set to *true* if field mapping should apply.
- **appendContent:** Set to *true* if the mapping should be appended to the Viz Mosart element item.
- **embeddedMosartItem:** Set to *true* if the MOS element has a Viz Mosart command embedded.
- **keywords:** Currently only the MOS parsing of Vizrt objects uses this column. It contains a comma-separated list of device-specific values.
  - `novideo` : Adding this excludes video clip information from Vizrt Video MOS objects, and only uses the lower thirds of the MOS item. The video ID can then be given from any other MOS object, or a story field. For example the video id column in iNEWS**.**
  - `nodefaultitem:` Adding this excludes Vizrt full-frame graphics if no default item is found (when missing Viz Mosart template).
  - `IgnoreTemplateSet` (or `IgnoreTemplateSet=true`): When added to a MOS ID named `"MOSART"` , the **templateset** attribute from a MOS object coming from the NRCS is ignored.
    For example:
    `<tag name="MOSART" value="" appendContent="false" fieldMapping="false" keywords="IgnoreTemplateSet=true"/>`

Copyright © 2024 Vizrt

5.6.5 Additional Newsroom Settings Examples

Defining Default Values for Lower Thirds

You can set default values for parsing of lower third objects. Usually these values are set from the objects received from the NRCS. Altering these values is usually part of a customization task.

- In `newsroomsettings.xml` add the default values to the *mmConstants/inewstags/properties* element:

```
<mmConstants><inewstags>
    <properties>
        <item name = "DefaultLowerThirdDuration" value = "125"/>
         <item name = "DefaultLowerThirdHandlerName" value = "DSK"/>
         <item name = "DefaultLowerThirdOutBehaviour" value = "TIMECODE"/>
         <item name = "DefaultLowerThirdOwnerMedia" value = "MEDIA"/>
    </properties>
</inewstags></mmConstants>
```

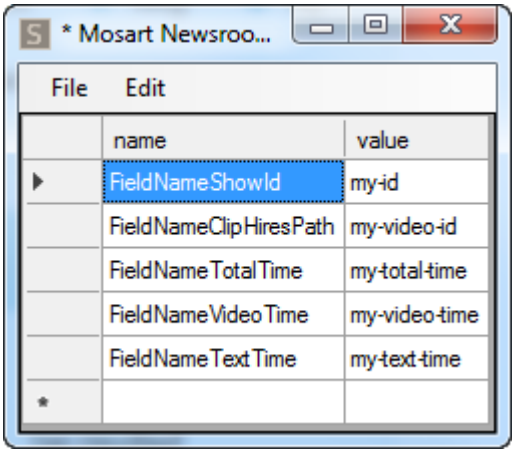
Lower Third Properties

Property	Description	Default value
DefaultLowerThirdDuration	The default duration of a TIMECODE lower third, in frames	125
DefaultLowerThirdHandlerName	The default graphics destination / handler	DSK
DefaultLowerThirdOutBehaviour	The default out behaviour	TIMECODE
DefaultLowerThirdOwnerMedia	The default owner attribute for secondary NRCS objects	MEDIA

iNEWS: Overriding Default Field Names when Parsing Stories

When parsing the iNEWS NSML, the system extracts various field values using default field names. These defaults can be overridden by setting new values in **Newsroom Settings > Edit > Properties**.

Copyright © 2024 Vizrt



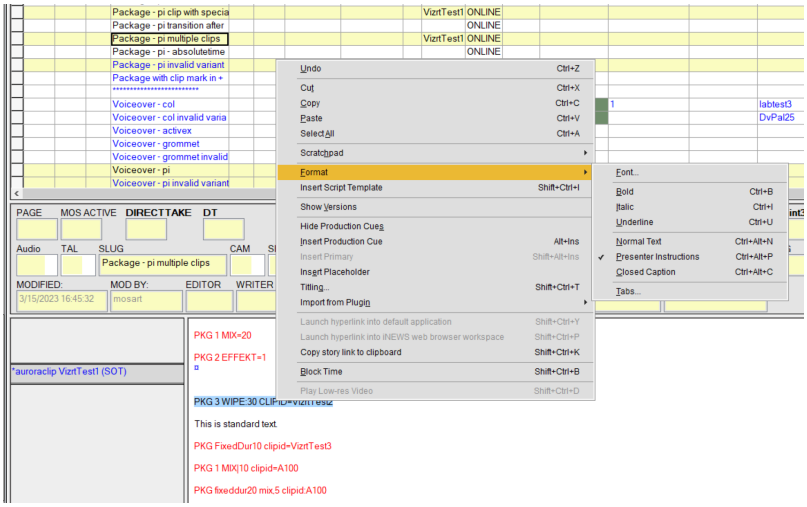
Field name	iNEWS default field name	Description
FieldNameShowId	id	The house ID of the show. Used for as-run logging.
FieldNameClipHiresPath	video-id	The video ID column. The field is appended to the first primary element in the story.
FieldNameTotalTime	total-time	The total planned duration for the story.
FieldNameVideoTime	tape-time	The video time of the story.
FieldNameTextTime	text-time	The iNEWS-calculated text time of the story. If the total time is not available, the value is calculated using this value is added to video time.

iNEWS: Presenter Instructions

You can use Mosart commands as presenter instructions in iNEWS for item types CAMERA, PACKAGE, VOICEOVER, ENDFRASE, ATTACHA, CLIPSTART, FULLSCREENGRAPHICS, DVE, TELEPHONEINTERVIEW, BREAK, JINGLE, LIVE, ADLIBPIX, CLIPHIRESPATH and ACCSESORIES.

A presenter instruction (*red text*) command in iNews is a text command written in the story script. The journalist designates text as a presenter instruction by selecting the text, right-clicking and selecting **Format > Presenter Instruction** (or shortcut **Ctrl+Alt+P**). The selected text appears in red indicating it is not standard story text.

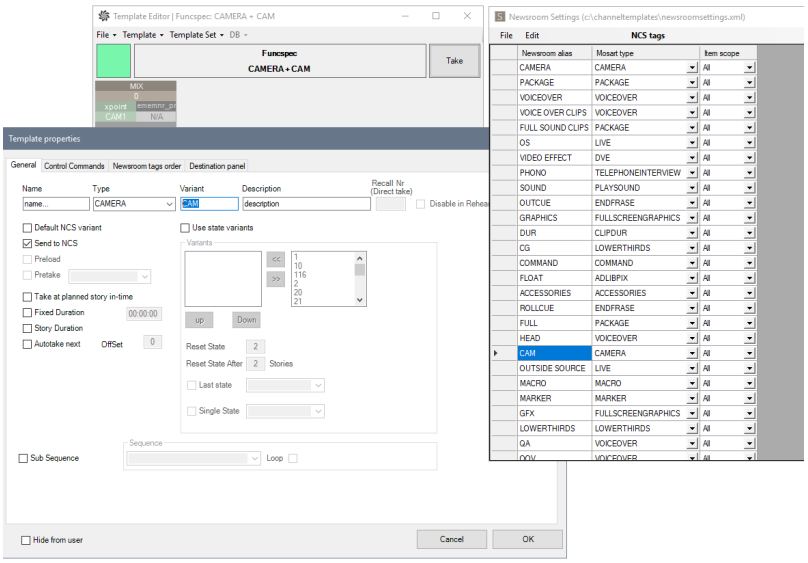
Copyright © 2024 Vizrt



To set up presenter text

Before working with presenter text in iNEWS, so one-off setups must first be made in Viz Mosart.

1. In Template Editor make sure you have template types defined for *all* (except for *Cliphirespath* and *Endfrase* which are not template types) the item types listed above.
   a. In Viz Mosart's AV Automation console, from the menu bar select **Devices > Template editor**.
   b. You will use presenter commands without specifying a template variant.
   You need to create a template variant with the same name as the newsroom alias, as shown below.



2. If not already defined, you must first add newsroom aliases for each of the following item types:
   Camera, Package, Voiceover, Live, Dve, Telephoneinterview, Break, Jingle, Adlibpix, Accessories, Cliphirespath, Endfrase.

Copyright © 2024 Vizrt

> **Note**: There can be several newsroom aliases for the same item type.
> You can use any of them for commands in your iNEWS story for defining a Mosart story item in a Mosart template.

a. In Viz Mosart's Manus Administrator console, type *ns*.
b. From the menu bar, select **Edit > NCS tags**.
c. Create mappings for any of the item types above that do not yet have an alias.

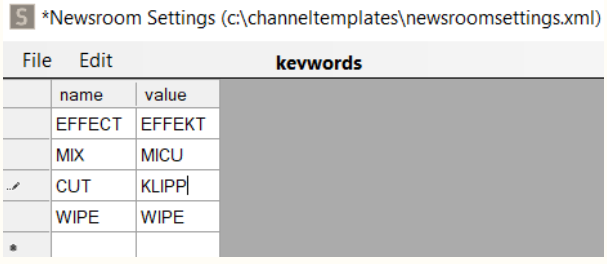
To work with presenter text

The following general commands can be used for CAMERA, LIVE, FULLSCREENGRAPHICS, DVE, TELEPHONEINTERVIEW, BREAK, JINGLE:

```
<ncs-alias> <variant> <transition><sep><value>
```

where

- **<ncs-alias>** is a Newsroom alias for the supported Mosart template types as defined in **Mosart Manus Administrator > Edit > NCS tags**:
  For example, the NcsAlias could be *cam* of type CAMERA.
- **<variant>** is a template variant defined in the Template Editor (step 1 above) for the given type.
  If no variant is provided in the command, **<ncs-alias>** is used as template variant, which is why you need to have a template defined in Template Editor with variant **<ncs-alias>**.
- **<transition>** is the video transition when switching from one story item to the next in a show.
  Can be by default **mix**, **wipe** or **effect**. The value for *mix* and *wipe* is the rate of the transition (in frames), while for the *effect* this is the effect number *(Physical Input)* configured in **AV Automation > Devices > A/V setup**.

> **Note**: For **<transition>**
> - Names are not case sensitive.
> - You can use a separator equal (=), bar(|) or colon (:).
> - Default Viz Mosart naming can be overridden with preferred names by navigating to the Mosart Manus Administrator console, typing `ns` followed by **ENTER**. Then **Edit > keywords** and mapping a new *value* to the default *name*.
>
> 
>
>
> This could, for example, be a local language text.

To work with clips

You can specify the clip id for clip types (PACKAGE, VOICEOVER or ADLIBPIX), enabling multiple clips to be added to a single story.

The command format is:

```
<ncs-alias> <variant> <transition><sep><value> clipid=<clipname>
```

where:

- **<ncs-alias>** is a Newsroom alias for the supported clip Mosart template types (PACKAGE, VOICEOVER, ADLIBPIX) in **Mosart Manus Administrator > Edit > NCS tags**.
  For example, the NcsAlias could be *pkg* of type PACKAGE.
- **clipid** assigns a clip to the Mosart item created with this command.

  > ⚠ **Note**: For ***<clipname>***
  > - This is the name of a clip residing on the MAM system controlled by Viz Mosart.
  > - clip name must not contain spaces.
  > - This parameter is useful when defining multiple clips in the same story. The first clip can get the clip id from the VID-ID column, the other clips using this option on the command line.

  If clipid is given for the first clip in the story, this overrides the value given in the VID-ID column.
- **<transition>**: same as above.
- **@** character. The absolute time field will be added to the element indicating that the element will be prepared and taken on at absolute time from midnight.

To attach a clip to last Viz Mosart item

The type CLIPHIRESPATH can also specify a clip id that will be attached to the last Mosart item.

The full command is

```
<ncs-alias> <clipname>
```

where

- **<ncs-alias>** is a Newsroom alias of type CLIPHIRESPATH as defined in **Mosart Manus Administrator > Edit > NCS tags**.

To work with accessories

The type ACCESSORIES can be used with manual or time-controlled takes.

The command format is

Copyright © 2024 Vizrt

- For auto accessory with timecode in and/or timecode out relative to the start of the story item:

```
<ncs-alias> <variant> tc-in=mm:ss tc-out=mm:ss
```

- For manual accessory which can be taken manually or executed at the absolute time specified by @:

```
<ncs-alias> <variant>  @hh:mm:ss|@hh:mm
```

where:

- **<ncs-alias>** is a Newsroom alias of type ACCESSORIES as defined in **Mosart Manus Administrator > Edit > NCS tags**.
- **tc-in** is the time when the accessory is taken in, relative to a story item start.

  > **Note**:
  > - If no **tc_in** is given, the accessory is considered *manual*.
  > - Other keywords for **tc-in** are **tcin**, **tc_in** or simply **in**

- **tc-out** is given to calculate the duration for an accessory item.

  > **Note**:
  > - Other keywords for **tc-out** are **tcout**, **tc_out** or simply **out**

- **@** character. Same as above.

To work with an end phrase

This command attaches the text given in the command to the story item preceding this command.

The command format is

```
<ncs-alias>=some text
```

- The text is displayed in the **Program** area in Viz Mosart UI if *Show end phrase* is enabled for this template type, from the menu
  Viz Mosart UI > Tools > General settings > Preview/Program window > End phrase bar > Show end phrase <type>.

Copyright © 2024 Vizrt

5.7 Field Mapping

The custom MOS object mapping is a setup which extracts field values from foreign MOS objects and translates these into Viz Mosart fields. This is done in the `newsroomsettings.xml` configuration file. This simplifies integration, as no new code needs to be deployed.

- The Tool Set
- What to Configure
- Target Fields
- Mapping Source Fields to Target Fields
- Examples
  - Video Server Item
  - Graphics Item
  - Harris Video Objects in Octopus

5.7.1 The Tool Set

- Default configurations are available as part of newsroomsettings.xml file.
- Select content from XML using simple XPath 1.0 expressions.
- Concatenate the XPath result using a separator.
- Replace parts of the result using static or dynamic variables.
- Apply conversion rules for the specific field type.

5.7.2 What to Configure

- Reuse or modify existing MOS field mapping in newsroomsettings.xml.
- Add MOS Id map entry using newsroom settings editor.
  - Newsroom alias: MOS ID of foreign item.
  - value: Field mapping entry name.
  - appendContent: appends the complete MOS object to the internal Viz Mosart <item /> as a _<content /> _sub child.
  - embeddedMosartItem: signal if the item has a Viz Mosart item embedded.

5.7.3 Target Fields

Target fields is a list of the fields used when controlling a video server or graphics elements.

Video Server

- **clip_hirespath:** The name or ID of the video file.
- **clip_description:** A human-readable description of the video file (optional).
- **clip_dur:** The duration of the video file.
- **clip_mark_in:** The start frame of the video file (optional).

Graphics Systems

Copyright © 2024 Vizrt

- **graphics_id:** The ID of the graphics item.
- **graphics_description:** A human-readable description of the graphics item.
- **tc_in:** Planned in time for the item relative to the parent item.
- **tc_dur:** Planned duration for the item.
- **graphics_profile:** The graphics concept context this item should be forced to play in. This value is forwarded to graphics systems supporting concepts for graphics when cueing and playing an item.
- **graphics_category:** Used in CG Take Out Logic to identify graphics that should be kept at story item transitions. If not set, **graphics_id** is used for identification.
- **channel:** The channel to play the item, i.e. **DSK**, **WALL** etc.
- **take**: **MANUAL** or **AUTO**. If **MANUAL** then **tc_in** is **0** and only **tc_dur** is used.
- **outBehaviour:**
  - **TIMECODE** = the element is taken out relative to the current primary element using **tc_dur**.
  - **BACKGROUNDEND** = the element is taken out when switching primary story element.
  - **STORYEND** = the element is taken out when switching from one story to the next.
  - **OPENEND** = the element is taken out when reloading the rundown or by manually taking the element out.
- **auto_continue:** true/false (only for Orad, to send PlayWithoutPause command).

5.7.4 Mapping Source Fields to Target Fields

The following mapping `<mosmap id=”VENDOR” elementtype=”CLIP”> </mosmap>` contains the attribute descriptions:

- **Id:** Value used in the MOS map section of the newsroomsettings.xml.
- **Elementtype:** CLIP for video servers, GRAPHICS for graphics systems.

And `<fieldmapping />` contains the attribute descriptions:

- **fieldname:** Viz Mosart target field name.
- **mospath:** A simple XPath 1.0 expression to the location of the source value.
- **separator:** Value used for joining multiple returned values for the mospath.
- **fieldtype:**
  - **FIELDS** – the source value is in video fields.
  - **FRAMES** – the source value is in video fields.
  - **TIMECODE** – the source value is in time code format. Use mask attribute to identify format.
  - **STATIC** – the value for the fieldname has no mospaths and should be as defined in the value attribute.
  - **REPLACE**
  - **ADJUST**
- **value:** The value to use if fieldtype is **STATIC**.
- **valuetype:** **TEXT**, **TIMECODE**.
- **mask:** Value on time code format for, i.e. **hh:mm:ss** or **hh:mm:ss:ff**.

Copyright © 2024 Vizrt

- **mustexist:** Set to **true** if the Viz Mosart field should only be added if a non-empty value exists.
- **regex:** Regular expression used on the value. Use this to extract parts of the value from the MOS object.
- **matchindex:** Group index of the result from the regex.
- **overwrite:** Overwrites if a field already exists if set to **true**. Default is **false**.

5.7.5 Examples

Video Server Item

The example below is of a Quantel video item:

Source MOS XML (Quantel MOS item)

```
<mos>
    <itemID>3</itemID>
    <itemSlug>New Row 3 CVD 1-3</itemSlug>
    <objID>15160::559</objID>
    <mosID>QUANTEL</mosID>
    <mosAbstract>Flug über die Alpen fertig2 1:00</mosAbstract>
    <abstract>Flug über die Alpen fertig2 1:00</abstract>
    <objDur>1500</objDur>
    <objTB>25</objTB>
    <objSlug>Flug über die Alpen fertig2</objSlug>
</mos>
```

⚠ **Note:** The objID is constructed from two numbers where the first is the Quantel clip id. The second number is the Quantel zone.

Viz Mosart MOS mapping

```
<mosmap id = "QUANTEL" elementtype = "CLIP" idasindex = "false">
    <fieldmapping fieldname = "clip_hirespath" mospath = "//objID" regex = "^\d+"/>
    <fieldmapping fieldname = "clip_description" mospath = "//objSlug"/>
    <fieldmapping fieldname = "clip_mark_in" fieldtype = "static" value =
"00:00:00:00" valuetype = "TIMECODE"/>
    <fieldmapping fieldname = "clip_dur" fieldtype = "FIELDS" mospath = "//objDur"
valuetype = "TEXT"/>
</mosmap>
```

Copyright © 2024 Vizrt

Target Viz Mosart XML

```
<fields>
    <field name = "clip_hirespath" value = "15160"/>
    <field name = "clip_description" value = "Flug über die Alpen fertig2 1:00"/>
    <field name = "clip_mark_in" value = "00:00:00:00"/>
    <field name = "clip_dur" value = "750"/>
</fields>
```

Graphics Item

The example below is of an XPression graphics item:

Source MOS XML (XPression MOS item (modified and stripped to fit screen))

```
<mos>
    <itemID>11</itemID>
    <objID>{01E4E214-2414-4C71-956F-6587E3FD1E4B}</objID>
    <mosID>XPRESSION</mosID>
    <mosAbstract>MAIN KEY Name (B2B): This is a test | Test</mosAbstract>
    <itemChannel>1</itemChannel>
    <itemEdStart>0</itemEdStart>
    <itemEdDur>0</itemEdDur>
    <itemTrigger>CHAINED</itemTrigger>
    <macroOut>NONE</macroOut>
    <mosExternalMetadata>
        <mosPayload>
            <gfxtype>CG</gfxtype>
            <itcTimeIn>500</itcTimeIn>
            <itcTimeDur>250</itcTimeDur>
        </mosPayload>
    </mosExternalMetadata>
</mos>
```

Viz Mosart MOS mapping

```
<mosmap id = "XPRESSION" elementtype = "GRAPHICS">
    <fieldmapping fieldname = "graphics_id" mospath = "(//itemID | //objID)"
separator = "-" fieldtype = "REPLACE" value = "{STORYID}-{MOSPATH}"/>
    <fieldmapping fieldname = "graphics_description" mospath = "//mosAbstract"/>
    <fieldmapping fieldname = "tc_in" fieldtype = "FRAMES" mospath = "//itcTimeIn"
valuetype = "TEXT"/>
    <fieldmapping fieldname = "tc_dur" fieldtype = "FRAMES" mospath = "//
itcTimeDur[text()!='0']" valuetype = "TEXT"/>
    <fieldmapping fieldname = "outBehaviour" mospath = "//macroOut"/>
```

Copyright © 2024 Vizrt

```
    <fieldmapping fieldname = "channel" fieldtype = "STATIC" mospath = "//
gfxtype[text()='CG']" value = "DSK" valuetype = "TEXT" mustexist = "true"/>
    <fieldmapping fieldname = "channel" fieldtype = "STATIC" mospath = "//
gfxtype[text()='OTS']" value = "WALL" valuetype = "TEXT" mustexist = "true"/>
    <fieldmapping fieldname = "channel" fieldtype = "STATIC" mospath = "//
gfxtype[text()='FS']" value = "FULL" valuetype = "TEXT" mustexist = "true"/>
    <fieldmapping fieldname = "channel" fieldtype = "STATIC" mospath = "//
gfxtype[text()='OTHER']" value = "FULL" valuetype = "TEXT" mustexist = "true"/>
</mosmap>
```

Target Viz Mosart XML

```
<item>
    <fields>
        <field name = "graphics_id" value =
"VER7ENPS1;P_VANLAB\W\R_F3144D55-78F2-494B-863ED060A74D92F7;D02D7434-58BB-43D6-
B03601E7B7BBC1E7-4-{1ECA795F-101D-4301-A559-2A1392F3AA2D}" fieldtype = "TEXT"/>
        <field name = "graphics_description" value = "BANNER:Chan1: | Manual |
Timed(---):TIMED" fieldtype = "TEXT"/>
        <field name = "tc_in" value = "" fieldtype = "TEXT" inputmask = "ff" default
 = "00:00"/>
        <field name = "tc_dur" value = "125" fieldtype = "TEXT" inputmask = "ff"
default = "00:00"/>
        <field name = "outBehaviour" value = "TIMED" fieldtype = "TEXT"/>
        <field name = "channel" value = "DSK" fieldtype = "TEXT"/>
    </fields>
</item>
```

Harris Video Objects in Octopus

How to configure MOS mapping of a Harris video object in Octopus:

Viz Mosart MOS mapping

```
<storyItem>
    <itemID>2-1</itemID>
    <itemSlug>0406 ZN MUNDE 5 WIN 3 PM NEW</itemSlug>
    <octext_elemLabel>VO</octext_elemLabel>
    <objID>0406 ZN MUNDE 5 WIN 3 PM NEW</objID>
    <mosID>ZEE.HARRIS.MOS</mosID>
    <itemTrigger>MANUAL</itemTrigger>
    <octext_elemType>vo</octext_elemType>
</storyItem>
```

Copyright © 2024 Vizrt

- Update the `newsroomsettings.xml` file by opening it in a text editor and adding the following XML entry in the `<mosmaps />` section (as a child to the `<mosmaps />` element node)

Target Viz Mosart XML

```
<mosmap id="HARRIS" elementtype="CLIP" idasindex="false">
        <fieldmapping fieldname="clip_hirespath" mospath="//objID" />
        <fieldmapping fieldname="clip_description" mospath="//itemSlug" />
        <fieldmapping fieldname="clip_mark_in" fieldtype="static" value="00:00:00:00"
 valuetype="TIMECODE" />
<!--
        <fieldmapping fieldname="clip_dur" fieldtype="FIELDS" mospath="//itemEdDur"
valuetype="TEXT" />
-->
</mosmap>
```

- Open the newsroomsettings.xml file in Mosart Server application using the **ns** command. Open **Edit -> MOS Id mapping** from the menu and add an entry where the value in Newsroom alias is the MOS ID of the Harris item, (in the above example ZEE.HARRIS.MOS), and the **Value** column is set to **HARRIS** (the id of the mosmap above). If the columns **appendContent** and **embeddedMosartItem** are shown, they should both have the value **false** .

Copyright © 2024 Vizrt

5.8 Manus Administrator Commands

These console commands are only intended for advanced users, and principally for troubleshooting. They shall never be used on a live system.

> **Warning:** Use of these commands may adversely affect system performance. Unauthorized use of these commands may invalidate a contracted Vizrt Support agreement.

tmpPrint: Print the manus

pcol: Print col1

setrundown, **SR**, **CD [name]**: Set a new rundown

TESTUPDATE: Start test timer

PRINTRUNDOWNS: Print newsroom rundowns

setManusDirectory: Set the directory in which manus are stored

VERSION Prints the current version

START: Start or stop the timeline
-FILEWATCHER -FW Starts the application in file watcher Modus

R: Release background

INIFILEPATH: Set ini file

INEWSFTPSERVER [servername] [username] [password]: Set the ftp connection to iNEWS FTP server, you must restart

GTS, **G**: Go to next story

NONEAUTOS: List none auto lower thirds

S: Skip next story

WQ: Write queue list

INEWSWORKINGDIRECTORY: Set the working directory

INEWSDEFAULTRUNDOWN: Set the default Showpath

LOG: Collect logs for all connections

SETTINGS: Show the Settings dialog

NEWSROOMSETTINGS, **NS**: Show Settings dialog for newsroomsettings

LOG: Write current log to file

SHOWLOG: Show logging elements

HIDELOG: Hide logging elements

PARSENATIVESTORY, **PNS** - Parse a native NRCS file to Mosart XML

PS [0] [0]: Print story or story item

PST: Print states

READSPEED [0]: Set read speed in words per minute

Copyright © 2024 Vizrt

RELOADMANUS, **RL**, **RELOAD**: Reload current manus

RELOADTEMPLATES, **RT**: Reload channeltemplates

RESYNC: Resync rundown with NRCS

LM, **LOADMANUS [filename]**: Load manus from file

LISTCLIPS: List all items in media cache

TRACE: Activate/deactivate the Trace listener

SAVE [path]: Save the rundown as a file.

STEST Y or **N** millisec

SS, **STUDIOSETUP**: Set the current studio setup

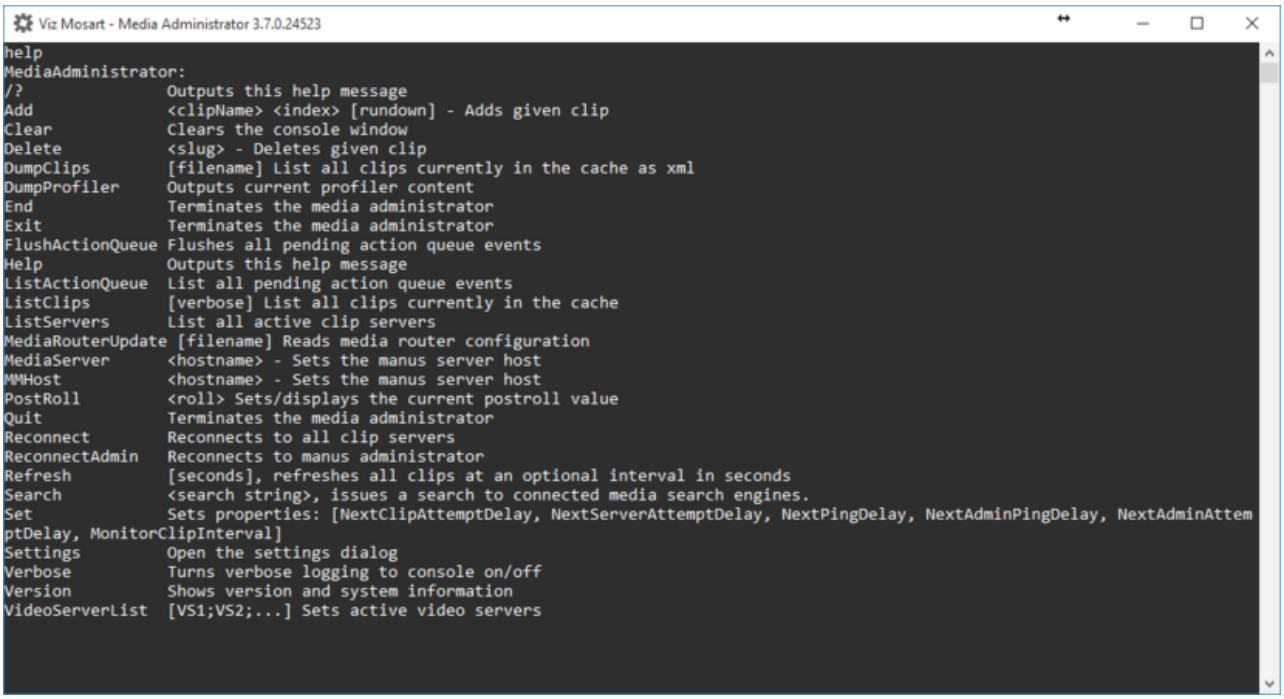
SAVEASSEQUENCE [sequencename as string]: Save the rundown as a sequence

DUMPNCSSTORY, **DNS [storyname] /p[destination filepath]**: Dump a story in native NSML Format

DUMPNCSRUNDOWN DNR /p[destination filepath]: Dump the current rundown in the native NSML Format

Copyright © 2024 Vizrt

6 Media Administrator



The **Media Administrator** handles the database connection to your video servers. It is responsible for reporting the clip status for all the video server elements in the current rundown, and for searching the video server for clips within the Viz Mosart GUI. The **Media Administrator** runs as a console application and should be kept running at all times as long as its services are required.

The **Media Administrator** receives the video server configuration (i.e. which servers are active) in the following circumstances:

- when the **Media Administrator** is started.
- when the **Manus Administrator** is activated (i.e. from **idle** to **active transition**).
- when the **Av Automation** changes the current setup.

To open the **Settings XML Editor**, type `settings` in **Media Administrator** followed by **ENTER**.

> ⚠ **Note:** The application must be restarted for any change to the settings to be applied.

This section contains the following topics:

- Media Administrator Commands
- Media Administrator Properties Editor

Copyright © 2024 Vizrt

6.1 Media Administrator Commands

Media Administrator is a console application, where you can run the following commands:

- **add <clipname> <index>:** Adds a clip manually to the list of clips currently monitored. Typically used to test the installation.
- **delete <index>:** Deletes a monitored clip with the given index.
- **dumpclips <filename>:** Outputs a list of all monitored clips in an XML format. Outputs to console if no filename is given.
- **dumpprofiiler:** Shows timing information. Typically used to test the installation.
- **exit:** Exits the Media Administrator.
- **flushactionqueue:** Flushes the internal action queue of all pending actions.
- **help:** Outputs all available commands.
- **listactionqueue:** Outputs the pending list of actions to be executed.
- **listclips:** Outputs a list of all monitored clips.
- **listservers:** Outputs a list of all configured media servers.
- **mmhost <hostname>:** Sets the hostname of the Manus Administrator. Default: **localhost**.
- **postroll <roll>:** Sets/displays the current value of the post roll.
- **reconnect:** Reconnects to all configured media servers. Typically only one video server is connected.
- **reconnectadmin:** Reconnects to the Manus Administrator.
- **refresh <n>:** Refreshes the status for all monitored clips <n> times (by making a request io the server for all clips). Typically used to test the installation. Default **<n> =** `1`.
- **search <regex>:** Issues a clip search to the video server. The regex is a regular expression for clip names. Typically used to test the installation.
- **set <property>=<value>:** Sets a property value. If no property is given, a list of all supported properties is displayed. The following properties are available:
  - **NextClipAttemptDelay:** Delay in seconds when to automatically verify presence of non-existing clips. Note that most video servers support asynchronous notifications making this functionality not necessary.
  - **NextServerAttemptDelay:** Delay in seconds when to try to reconnect to a video server. When the connection to a video server is not successful, this delay determines the time to wait before a new connection attempt.
  - **NextPingDelay:** Delay in seconds when to ping the connected video servers, to detect whether the servers are valid.
  - **MonitorClipInterval:** Interval for monitoring clips. A clip is polled for changes as long as any of its attributes changes.
- **settings:** Opens the Media Administrator Properties Editor.
- **verbose:** Toggles verbose output. In verbose mode, more information is logged to the console.
- **version:** Outputs Media Administrator version and some system information.

Copyright © 2024 Vizrt

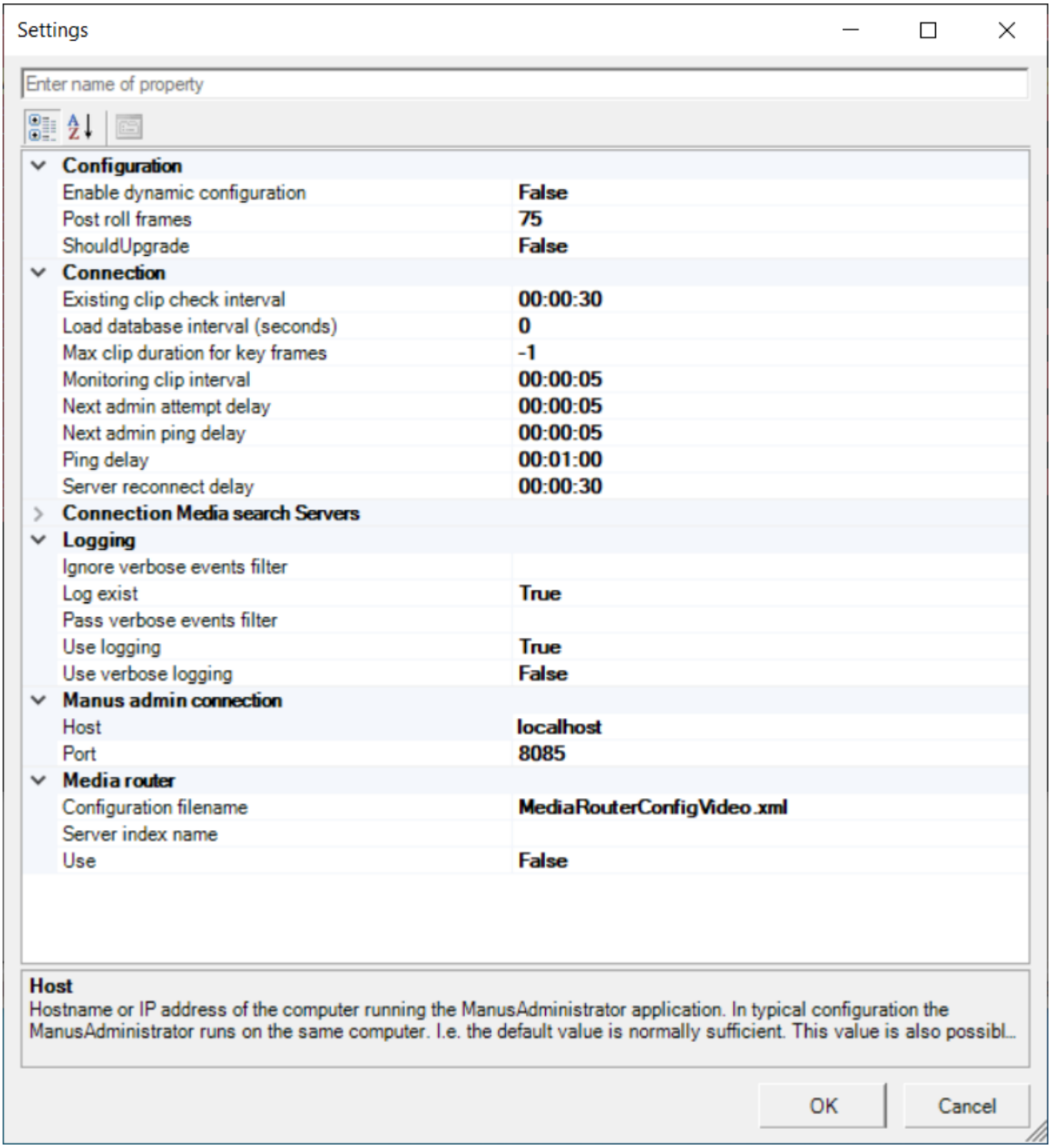
6.2 Media Administrator Properties Editor

Media Administrator is configured through the **Properties XML Editor**, which is opened by typing `settings` in the console window.

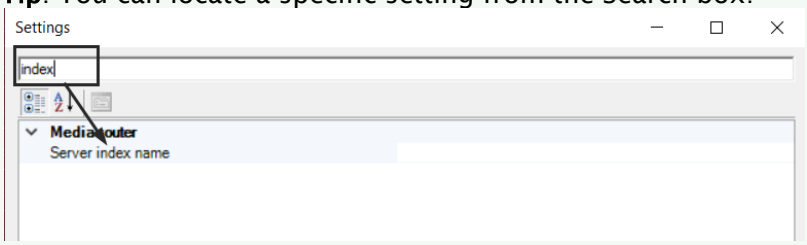
> ⚠ **Note:** Once you have made any changes, you must restart the Media Administrator for changes to apply.

- Sections of the Mosart Media Administrator Properties File
  - Configuration
  - Connection
  - Connection Media Search Servers
  - Logging
  - Manus Admin Connection
  - Media Router
- Dynamic Video Server Configuration
  - Enabling Dynamic Configuration

Copyright © 2024 Vizrt



Tip: You can locate a specific setting from the Search box:



Copyright © 2024 Vizrt

6.2.1 Sections of the Mosart Media Administrator Properties File

Configuration

- **Enable dynamic configuration** (EnableDynamicConfiguration):
  - When *False* (default), *Media Administrator* connects to all servers configured (see Connection Media Search Servers, **Video clip Server <n>** below).
  - When True, it connects only to
    - Servers that are part of the salvo selected in *AV Automation* (see Video Servers > Virtual Server Ports > Switch between salvos)
      and
    - Servers which have Static=*True* in the connection string (see Connection Media Search Servers, **Video clip Server <n>** below).

> **Note**: For further details, see section Dynamic Video Server Configuration below.

- **Post roll frames** (PostRollFrames): Subtracts value from the actual clip length when sending clip info back to the **Manus Administrator**. Default: `75` *(frames).*
- **ShouldUpgrade**: For Vizrt Support use only, must be set to *False*. Default: *False.*

Connection

- **Existing clip check interval** (ExistingClipCheckInterval): Delay in seconds when to automatically verify the presence of non-existing clips. Note that most video servers support asynchronous notifications, making this functionality unnecessary. Default: *00:00:30.*
- **Max clip duration for key frames** (MaxClipDurationForKeyFrames): Disables showing keyframe markers for video files greater than this value in frames. Default: *-1* .
- Monitoring clip interval (MonitoringClipInterval): Default interval for monitoring clips. A clip is polled for changes as long as any of its attributes changes. Default: *00:00:05.*
- **Next admin attempt delay** (NextAdminAttemptDelay): Interval for retrying reconnecting to the Manus Administrator. Default: *00:00:05* (hh:mm:ss).
- Next admin ping delay: Interval for sending heartbeat to the Manus Administrator. Default: *00:00:05 (hh:mm:ss).*
- **Ping delay** (PingDelay): Interval between command requests (heartbeats) that Media Administrator sends to the video server. Pinging is used to detect whether the connected video servers are valid. Default:
  00:01:00.
  For some video servers, ping can be disabled using the DisableHeartbeat property inside the connection string. See, for example, Configuration File Properties - VDCP > DisableHeartbeat.
- **Server reconnect delay** (ServerReconnectDelay): Delay in seconds when to try to reconnect to a video server. For example, if connecting to a video server is unsuccessful, this delay determines the time to wait before a new connection attempt is made. Default: *00:00:30.*

Copyright © 2024 Vizrt

Connection Media Search Servers

- **Video clip Server <n>** (VideoClipServer <n>): Connection string to a media server (a video server or MAM system) <n>.
  A maximum of five media servers may be connected. In most cases more than one server will be connected.
  These connection strings all share a common format of a semicolon-separated list of pairs Property=*Value*.
  Although the selection of properties and values may vary according to the type of media server, some properties are common to all.
  For example:
  - *Name*.
  - Any connection string may have the property (and value) *Static=True*.
    For details, see Video Server, MAM and Database Connection Strings.

Viz Mosart in general requires *two* connections to each video server, one from *Media Administrator* (as defined here) and one from *AV Automation* (see Video Servers, section *Working with Video Server Configuration*).

AV Automation also has a *Name* property, and the two connections to the same server should have the same, unique *Name*, This name cannot be used for any other connection.
By default, *Media Administrator* connects to *all* servers defined here.

However, if the setting **Enable dynamic configuration** is *True*, it connects only to

- The servers (identified by the *Name* property) that are part of the salvo selected in *AV Automation* (see Video Servers, Virtual Server Ports, Switch between salvos), and
- The servers which have *Static=True* in the connection string.

> **Note:** When Media Administrator searches through all servers for information on a media clip, the search sequence starts with the last (highest numbered) server configured. This may impact the result of the media search if the media clip is present on more than one of the servers, and some property of the clip is not identical on these servers. In this case, the property is set from the *last* server checked (the lowest numbered server configured) from where the clip was available.

- **Video clip Server <n> Backup:** Deprecated.

Logging

- **Ignore verbose events filter** (IgnoreVerboseEventsFilter): Semicolon based list of events to ignore when using verbose logging. Default: *<none>.*
- **Log exist** (LogExist): Verifies the presence of the Viz Mosart Log Service at startup. Default: *True.*
- **Pass verbose events filter** (PassVerboseEventsFilter): Semicolon based list of events to log when using verbose logging. Default: *<none>* .
- **Use logging** (UseLogging): Passes all logging to the Log Service if true. Default: *True.*
- **Use verbose logging** (UseVerboseLogging): Enables verbose logging if true. Default: *False.*

Copyright © 2024 Vizrt

Manus Admin Connection

- **Host:** Sets the hostname or IP address of the computer running the Manus Administrator application. In a typical configuration, the Manus Administrator runs on the same computer (i.e. the default value is normally sufficient). This value is also possible to set using the **mmhost** command. Default: **localhost.**
- **Port:** Determines TCP/IP (.NET Remoting) port to establish the connection to the Manus Administrator application. Default: `8085`.

Media Router

- **Configuration filename** (ConfigurationFilename): Determines the configuration file to use for **Media Router**. Default: *MediaRouterConfigVideo.xml.*
- **Server index name**: Deprecated.
- **Use**: Uses the Media Router if true. Default: *False.*

6.2.2 Dynamic Video Server Configuration

Enabling dynamic configuration ensures that only the video servers that are currently in use by AV Automation are searched by Media Administrator.
The key setting is **Enable dynamic configuration**, described in Configuration Settings above.

Enabling Dynamic Configuration

The settings **Enable dynamic configuration** (EnableDynamicConfiguration) affects

1. The *Name* properties as used in connection strings in AV Automation and Media Administrator must match. Only video servers with a name that corresponds to the video server naming in AV Automation get used.
   Any other configured connections are ignored.
   The *Name* parameter is Media Administrator's method of distinguishing between the various clip metadata sources. It is also Viz Mosart's means of matching clip metadata sources and video servers.
2. Any Media Administrator connections that do not relate to a particular video server used by AV Automation. These shall be marked with Static=*True* in the connection string. A typical example is a connection to a MAM system, for clip information.

Example 1: Connection string to a video server (Quantel) in Media Administrator:

When dynamic configuration is enabled a corresponding video server (with name=Q1) has to be configured and used by AV Automation

```
Name=Q1;Type=Quantel;SerialNo=21234;IOR=http://quantel:@10.0.0.131/
ZoneManager.ior;Slave=10.0.0.132;Config=ClipServerQuantel.xml
```

Copyright © 2024 Vizrt

Example 2: Static connection with a MAM system

When dynamic configuration is enabled, this connection shall always be established, regardless of the current AV Automation configuration.

```
Name=VizOne;Type=VizOne;Static=True;Server=10.0.0.136;Port=8070;Config=ClipServerJupiter.xml
```

Addit

7 Overlay Graphics

- Overlay Graphics Interface
- Menu Bar
- Overlay Graphics Configuration and Use

The Overlay Graphics interface provides control and monitoring of overlay graphics for all Viz Mosart-approved graphic engines.

- For a list of supported graphics devices, see Overlay Graphics Types.
- Configuration is done in the Overlay Graphics Configuration Properties menu.

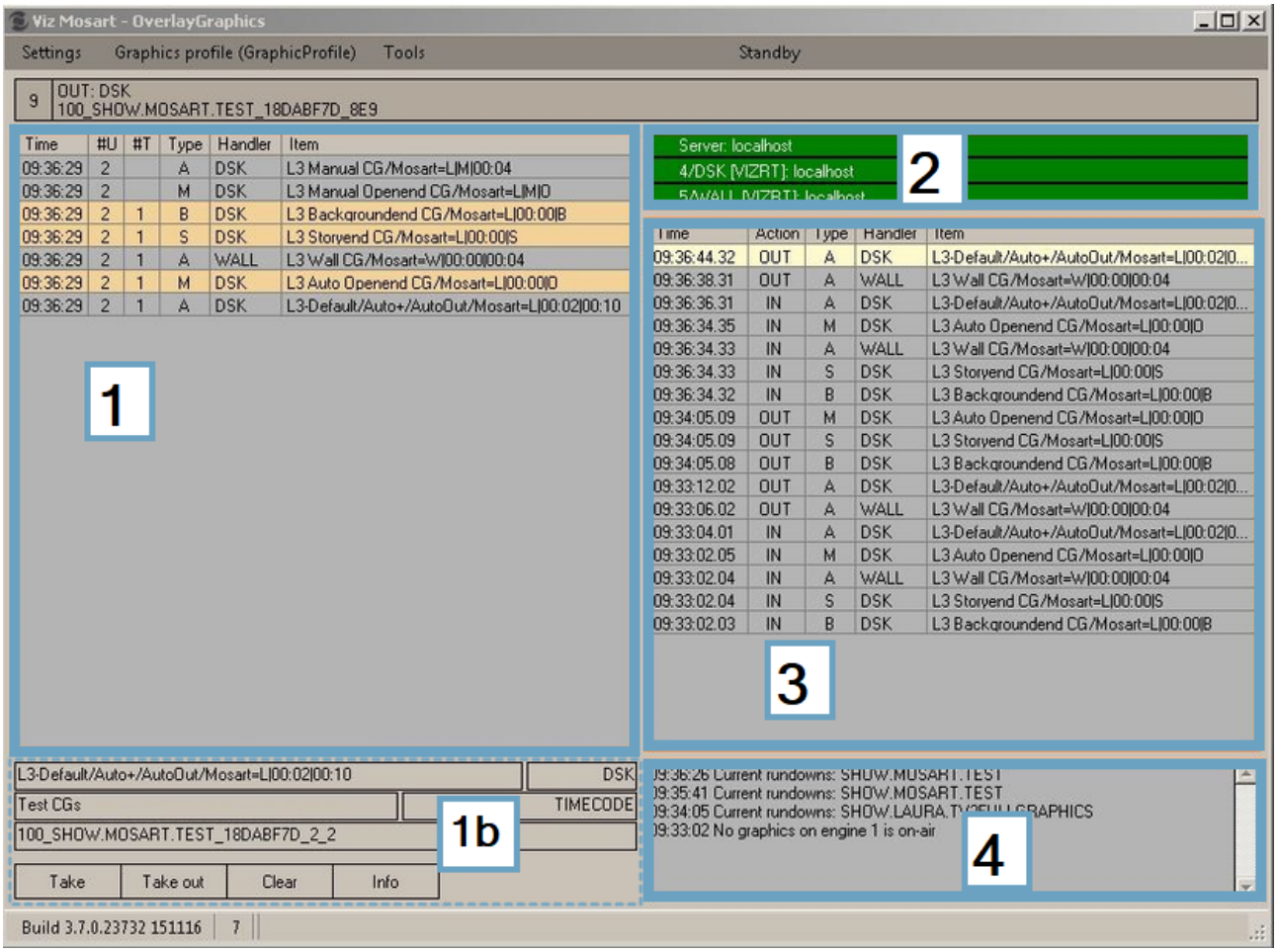
Start **Mosart Graphics Interface** by double-clicking the application icon:

or by navigating to the installation directory (by default `C:\Program Files (x86)\Mosart Medialab\Mosart Server`) executable `MMOverLayGraphicsInterface.exe`.

> ⚠ The Overlay Graphics Interface (and application icon) is only available if Overlay Graphics was selected when installing the Viz Mosart Server.
> The Overlay Graphics interface is *only* used for overlay graphics. *Full-frame* graphics are controlled through the AV Automation application.

A menu bar at the top offers **Settings**, **Graphics profile**, **Tools** and **Standby**.

Copyright © 2024 Vizrt



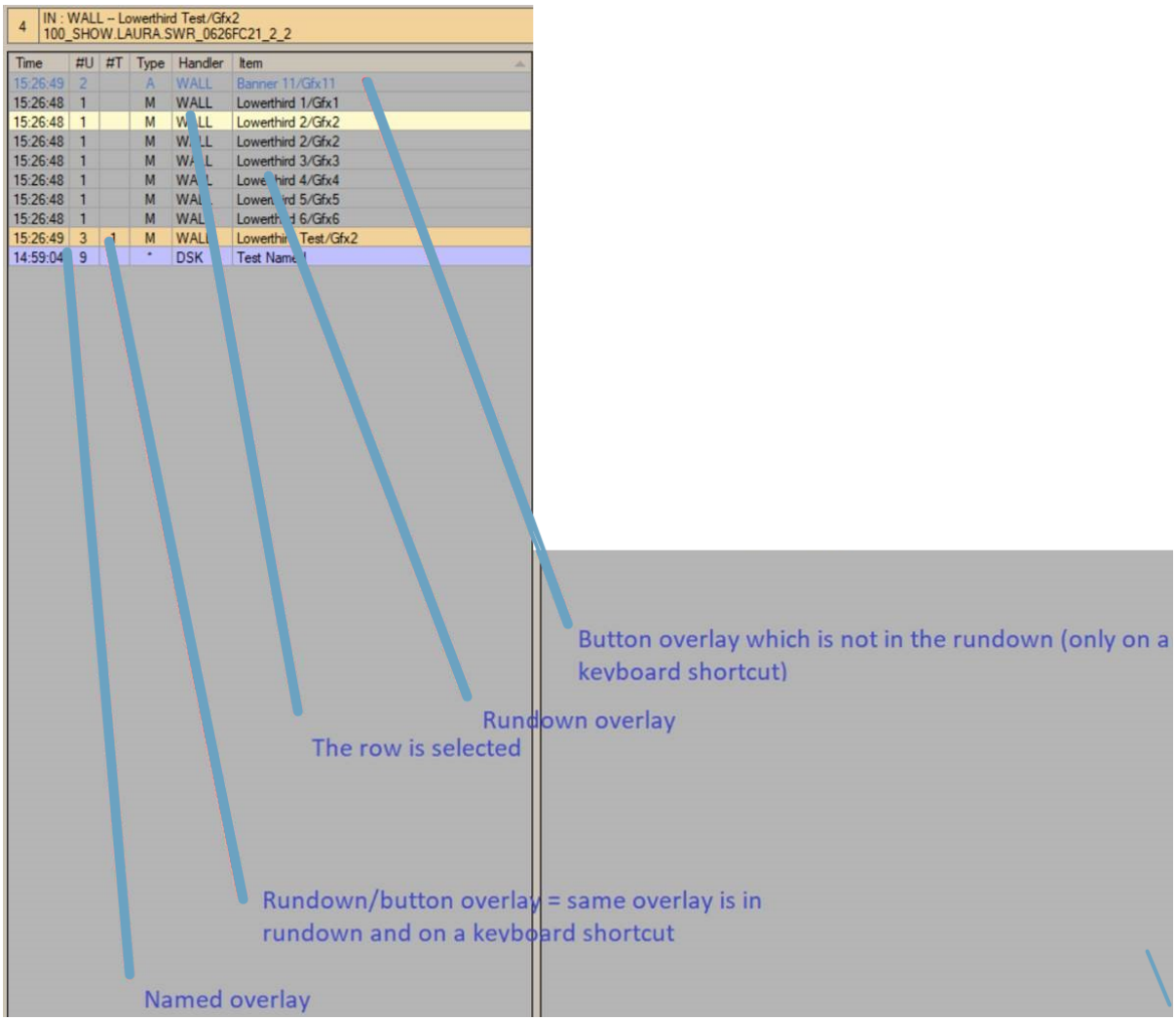
7.1 Overlay Graphics Interface

The Overlay Graphics interface consists of four main areas (**[1]** to **[4]** in the illustration).

A status bar at the foot of the UI provides information on product version number and the total number of graphics in Viz Mosart (rundown overlays + named overlays + button overlays. *7* in the example above).

Copyright © 2024 Vizrt

7.1.1 [1] List of Overlay Graphics in the Rundown



The main column labels are:

- **Time:** The time when the graphic was added in Overlay Graphics (usually at rundown import).
- **#U:** The number of times this graphic has been updated.
- **#T:** The number of times this graphic has been taken.
- **Type:** The graphic type. Can be:

Type	Description
A	Auto Out (a Manual CG can also be Auto Out)
B	Background End

Copyright © 2024 Vizrt

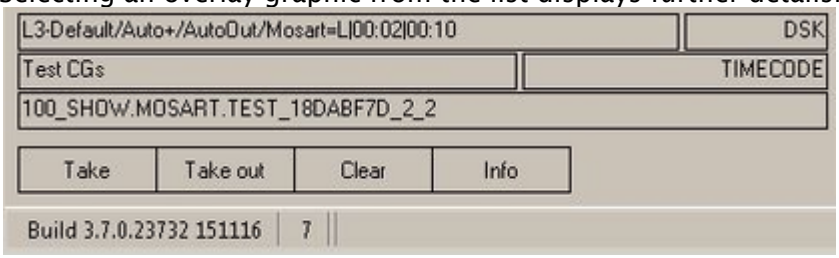
Type	Description
S	Story End
M	Manual (Open End)
L	Last Out
-	Stay Always
>	if template is **STORYSTART**
*******	None of the above

If the graphic is a *locator*, then the type is suffixed with an "**L**".

> **Note: Locator** is a special graphics item that is **linked** to a video server crosspoint. When taking the crosspoint from a Viz Mosart template, either as a switcher crosspoint or keyed crosspoint (in a DVE box), the graphics are also (re-)taken.

[1b] Details of a Selected Graphic from the List of Overlay Graphics

Selecting an overlay graphic from the list displays further details:



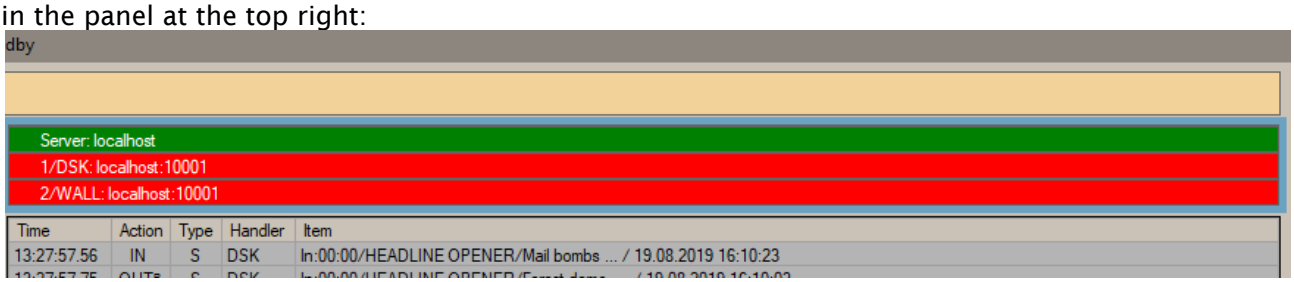
Explanation of Details

Element	Description
	Graphic title
	Handler name (the destination or the channel for this graphic)
	Graphic description

Element	Description
	Graphic ID ⚠ This is the *Viz Mosart* Graphic ID, not the graphic ID from graphic database (Graphic Hub/Oracle)
	Action Buttons **Take:** Takes graphic on air **Take out:** Takes graphic off air **Clear:** Takes out all taken graphics **Info:** Opens an editor window where the graphic MOS item can be changed and updated

7.1.2 [2] Server Status

Viz Mosart server and configured graphic engines statuses are displayed (Green: OK, Red: Failure) in the panel at the top right:



7.1.3 [3] General Log

The center right contains a log of all actions performed on the overlay graphics either triggered by Viz Mosart or by clicking **Overlay Graphics** buttons.

To clear this area, double-click inside it.

7.1.4 [4] Event Log

The bottom right contains an event log with messages, warnings and errors from the Overlay Graphics application.

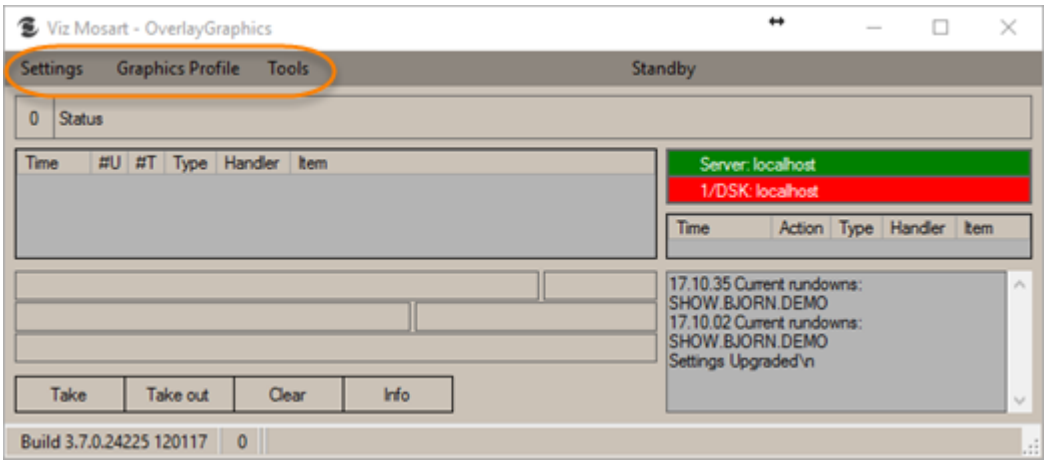
To clear this area, double-click inside it.

Copyright © 2024 Vizrt

7.1.5 System Information

The information line at the bottom displays application version, number of graphics in the table and whether the Mosart Media Router (**MMR**) is in use:

7.2 Menu Bar



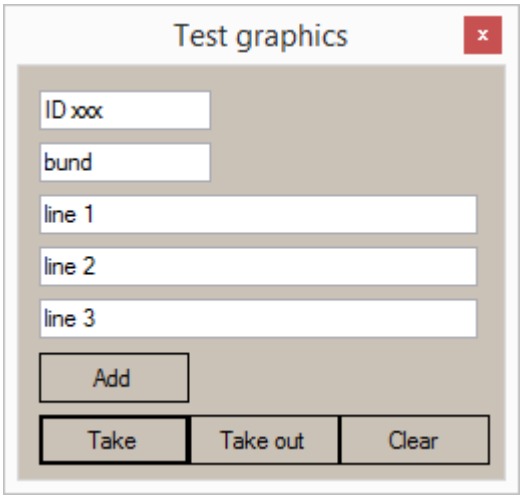
Settings

Properties: Opens Overlay Graphics Configuration.

Graphics Profile: Select the required graphics profile.
See Graphics Profiles Tab.

Tools
When things don't go as expected, there is a debugging option.

Editor: Opens a **Test graphics** window.



Instrumentation Panel: Displays all device commands in real time, as sent to the various connected devices.
See Mosart Instrumentation Panel.

Standby: Toggles the graphics engines in or out of standby.

Take any controller into standby. All the engines linked to the controller are set /unset to standby. The same action can also be performed from the Viz Mosart UI.

Take any engine individually into standby (only in OGI) while the other engines in the same controller are active.

7.3 Overlay Graphics Configuration And Use

See the respective sections:

- Managing Overlay Graphics Controllers and Engines
- Overlay Graphics Configuration Property Tabs
- Mosart Graphics Destinations and Takeout Logic
- Overlay Graphics Types
- Sending Lower Third Graphics MOS Object to an NRCS

Copyright © 2024 Vizrt

7.4 Managing Overlay Graphics Controllers And Engines

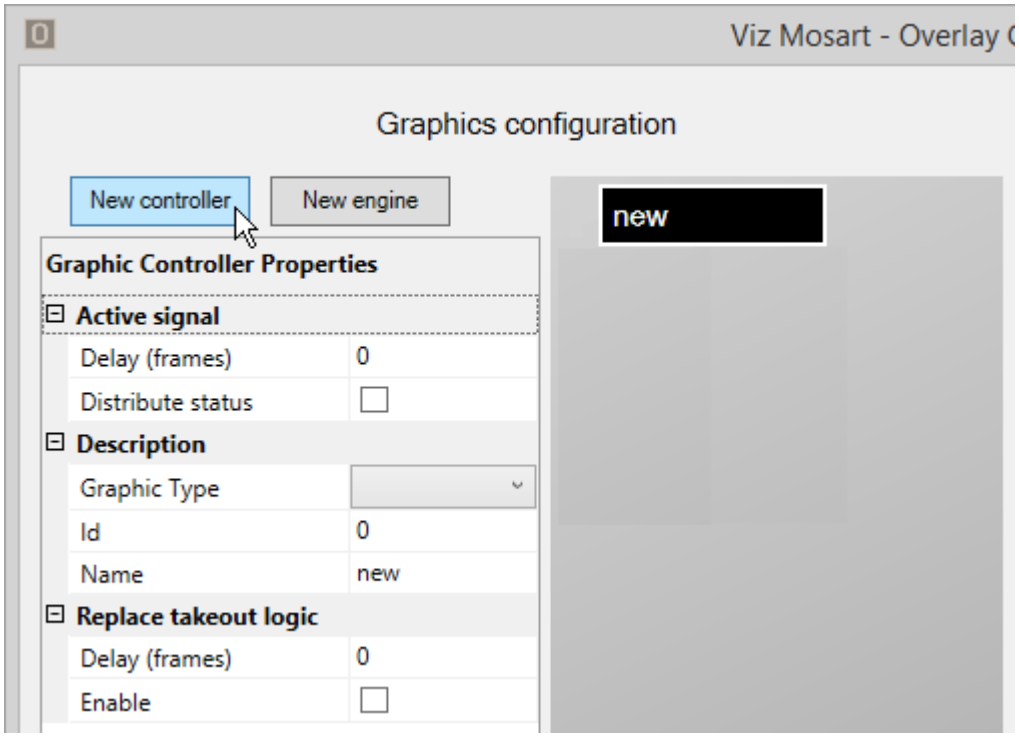
7.4.1 Overlay Graphics Controllers and Engines

- Working with Controllers
- Working with Engines
- Common Graphic Controller Properties
- Common Graphic Engine Properties

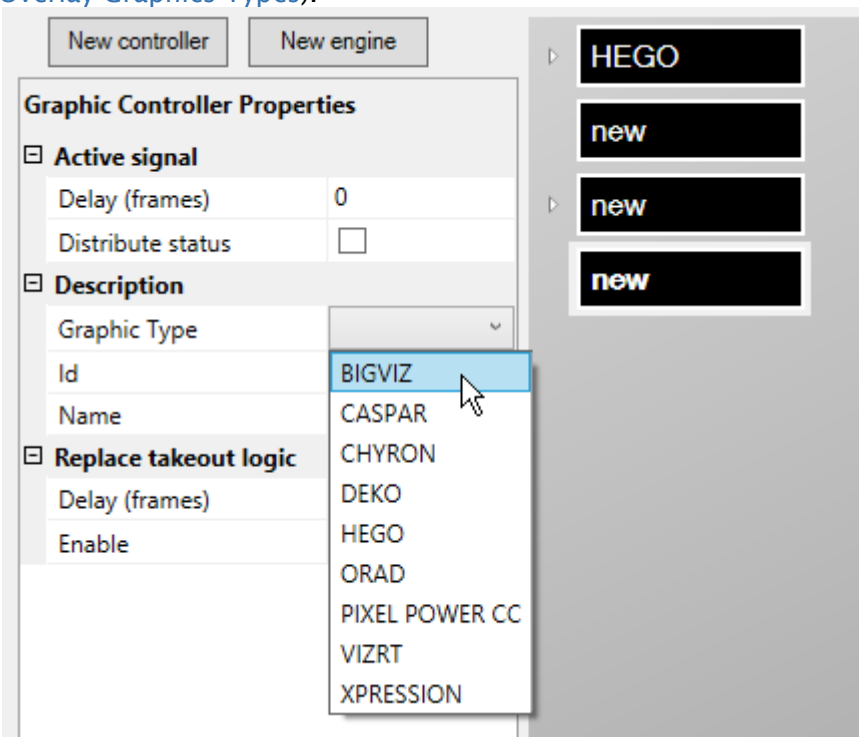
Working with Controllers

To add a new Controller

1. Open the Overlay Graphics Configuration window by going to **Settings > Properties**.
2. Click **New controller**.



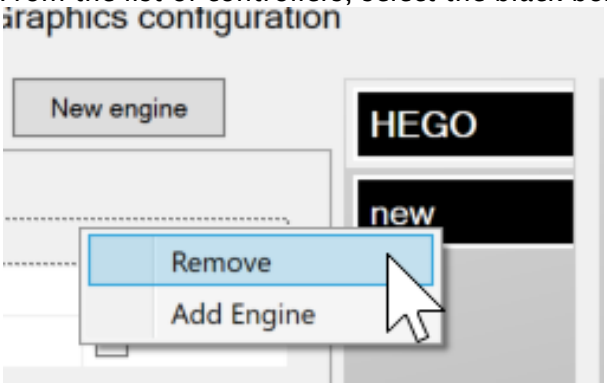
3. In the **Graphic Type** drop-down, select the type of graphics system you are using (see Overlay Graphics Types).



4. Set additional controller properties according to the selected **Graphic Type**.
5. Optionally, add graphics profiles via the Graphics Profiles Tab. Only applicable for some graphics systems.
6. To add engines to the controller, see To Add a New Engine below.

To remove a Controller

1. Open the Overlay Graphics Configuration window by navigating to **Settings > Properties**.
2. From the list of controllers, select the black box of the controller you want to remove.

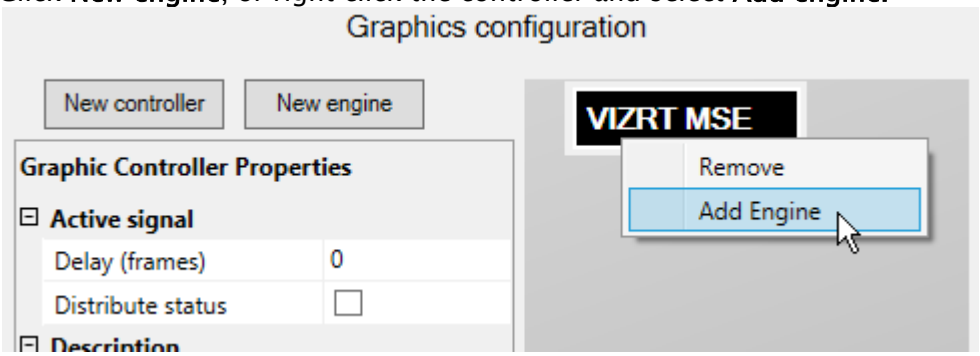


3. Right-click and select **Remove**.

Working with Engines

To add a new Engine

1. Open the Overlay Graphics Configuration window by navigating to **Settings > Properties**.
2. From the list of controllers, select the black box of the Controller you want to use.
3. Click **New engine**, or right-click the controller and select **Add engine.**



4. Set the engine properties according to the selected graphic type (See Overlay Graphics Types).
5. Optionally, set actions for selected system events using the Actions Tab (Event and Action Rules).

To remove an Engine

1. Open the Overlay Graphics Configuration window by navigating to **Settings > Properties**.
2. From the list of engines, select the black box of the engine you want to remove.
3. Right-click and select **Remove**.

Common Graphic Controller Properties

The Graphic Controller properties that are available depend on the graphic type of the controller. The controller properties common to most graphics types are described below.

> ⚠ **Note**: Additional properties specific to each graphic type are described in section Overlay Graphics Types.

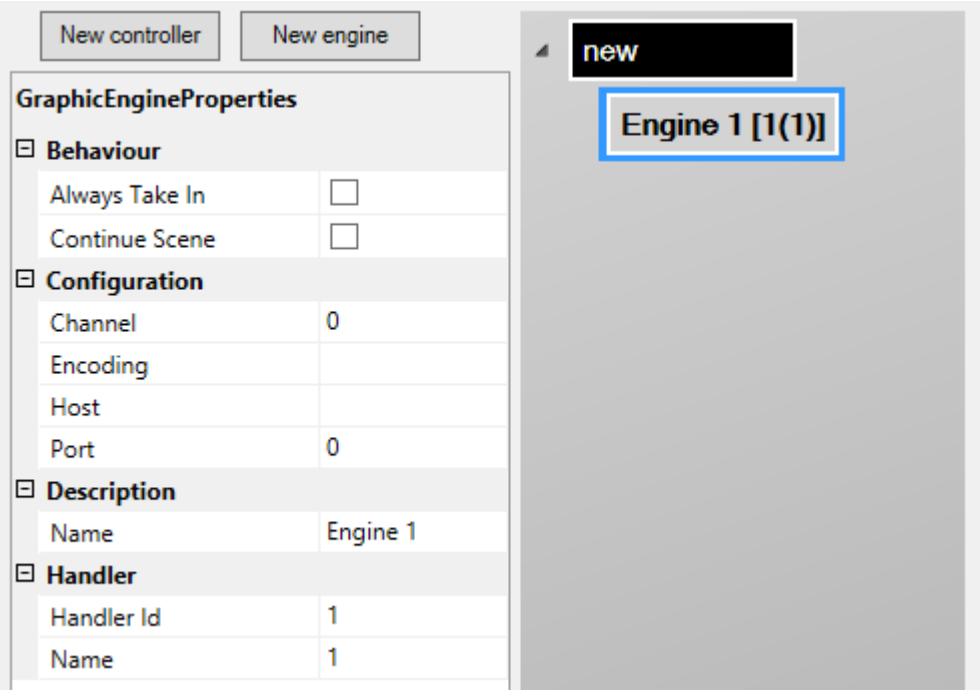
- **Active signal:** The active status signal is sent to AV Automation containing on air status of a graphics engine.
  - **Distribute status:** When set to t*rue*, reports the on air/usage status of an engine to AV Automation. This status message is sent to the AV Automation log area, the Manus Administrator console and the Overlay Graphics log area.
  - **Delay (frames):** Sets the delay of the graphics active status changed event signal. Using a negative value for the delay disables the functionality (**Distribute status** is *false*).
    Upon receiving the signal, AV Automation performs the following actions:
    - Sets the DSK if this is enabled (Control DSK from OverlayInterface is checked on the **Switcher preferences** page and the switcher can set *DSK*).

Copyright © 2024 Vizrt

        - Executes direct take templates configured on the **Switcher preferences** page in **Overlay DirectTake control** (see section Vision Mixer).
        - Cuing the graphics in certain conditions (the rundown story item is in preview, the channel template used has a graphics device setup, the engine is not in use, the channel is not on air).
- **Description**
    - **Graphic Type:** Selects the brand and type of graphics.
    - **Id:** Identifies the graphics controller (unique number ID). Shall be unique among all defined controllers.
    - **Name:** Defines the name of the server (optional).
- **Replace takeout logic:** Reads out the transition logic context of elements to either suppress take out or re-take items. Take outs are suppressed if another item with the same context was taken before the scheduled for take out action. Take outs are replaced with re-takes if the item scheduled for take out replaced an item with the same context. Only for Vizrt graphics with transition logic.
    - **Delay (frames):** Delay when checking for duplicate 'take in' commands in the next element.
    - **Enable:** Use transition logic aware take in/out commands.

> **Note:** See also Graphics Profiles Tab for other controller settings.

Common Graphic Engine Properties



The **Graphic Engine Properties** that are available depend on the graphic type of the controller. The engine properties common to most graphics types are described below.

The additional properties for each Graphic Type are described in Overlay Graphics Types.

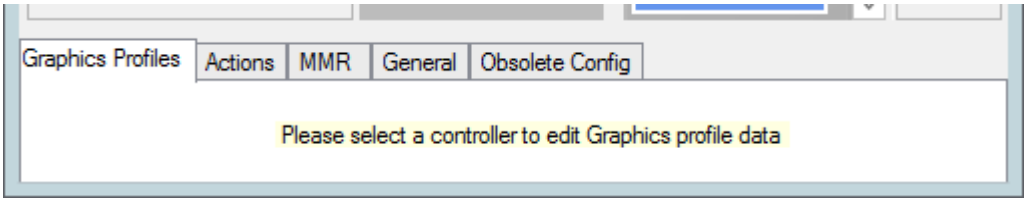
Copyright © 2024 Vizrt

- **Behavior**
  - **Always Take In:** Turns off the logic that does not take in a graphic element that is already taken and no other graphic element has been taken since.
  - **Continue Scene:** Use continue for take out of scene based graphics.
- **Configuration**
  - **Channel:** Defines the output channel from the graphics system. Set to `0` when not used for takes.
  - **Encoding:** Encoding to use on text. **UTF-8** or **UTF-16**. Default: **UTF-8**.
  - **Host:** Defines the hostname or IP address of the graphics engine. See also descriptions for the various graphics types.
  - **Port:** Defines the TCP port to use when connecting to the graphics system.
- **Description**
  - **Name:** The name of the engine.
- **Handler:** ID and name of the handler. Note that if handler ID is *changed* and the engine is assigned to a destination, then the link must be removed and engine re-assigned.
  - **Handler Id:** Auto-generated ID.
  - **Name:** The name of the handle (auto-generated).

> ⚠ **Note:** See also:
>
> - Section Graphics under *AV Automation Device Properties*.

Copyright © 2024 Vizrt

7.5 Overlay Graphics Configuration Property Tabs



The tabs functions are:

7.5.1 Working with Overlay Graphics Configurations

- To change a Graphics profile
- To set up event-action rules
- To configure the Mosart Media Router
- To configure general settings
- To convert an incompatible overlay graphics configuration

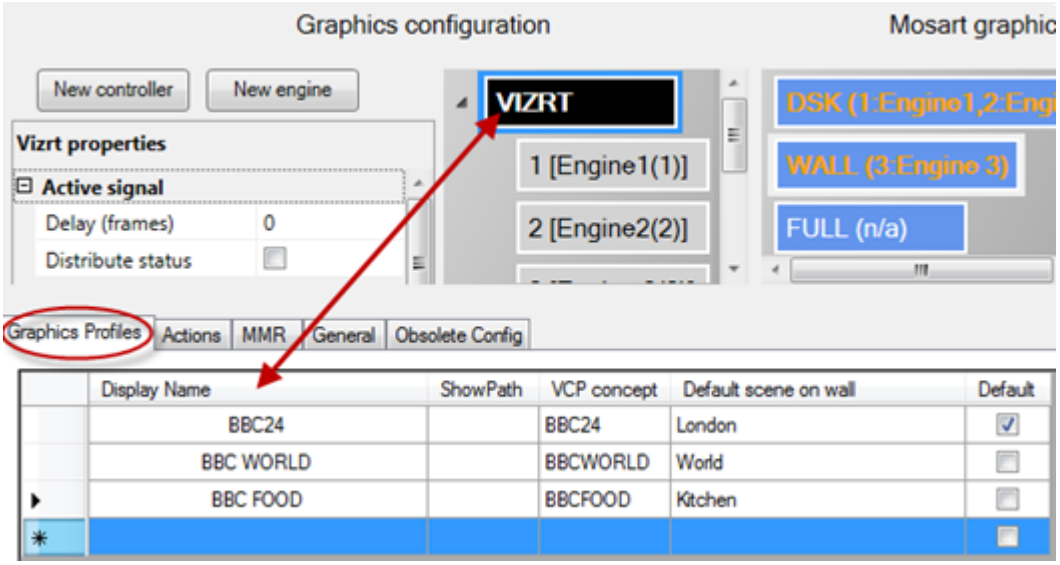
Graphics Profiles Tab

The selection of graphics profile affects the look of all graphics used for the specific show. Viz Mosart handles which concept the graphics devices use (on Vizrt graphics systems this technique is called **concept override**, on other graphics systems Viz Mosart uses its own concept override technique).

To change a Graphics profile

1. Navigate to **Settings > Properties > Graphics Profiles** tab.
2. Select the controller (black) that you want to configure.
   The **Graphics Profiles** list changes dynamically, depending on which controller is selected.
3. When the graphics profile is changed in the Overlay Graphics Interface (or the Viz Mosart GUI), all the controllers are scanned.
   If a controller has the selected graphics profile, the show path, VCP concept (Viz only) and default scene for wall values, are all changed.
4. Remember to restart the Overlay Graphics Interface application for any changes to take effect.

Copyright © 2024 Vizrt



- **Display Name:** Name of the Graphics Profile that is displayed to the user. The Graphics Profiles are displayed as menu options both in the Mosart GUI and in Overlay Graphics.
- **ShowPath:** Path to the files the graphics engine uses. Absolute path where graphics are located within the graphics engines. Dependent upon graphics type.
- **VCP concept:** Name of the Viz Pilot concept that should be used when working with the selected profile (Vizrt only).
- **Default scene on wall** Scene name to use as default for wall graphics. Dependent upon Graphics Type.
- **Default:** Selects the graphics profile to use as default. The default graphics profile is used initially before any profile is set from the user.

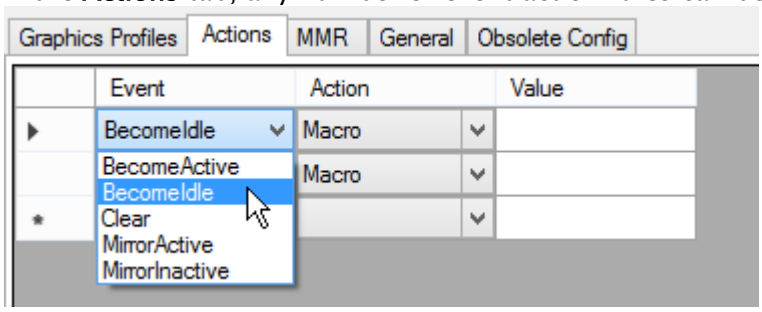
Click on the Graphics Profile tab to switch between profiles. The default profile is displayed in bold, and the active profile is marked with a check mark, and is displayed in parenthesis on the menu itself.

Actions Tab

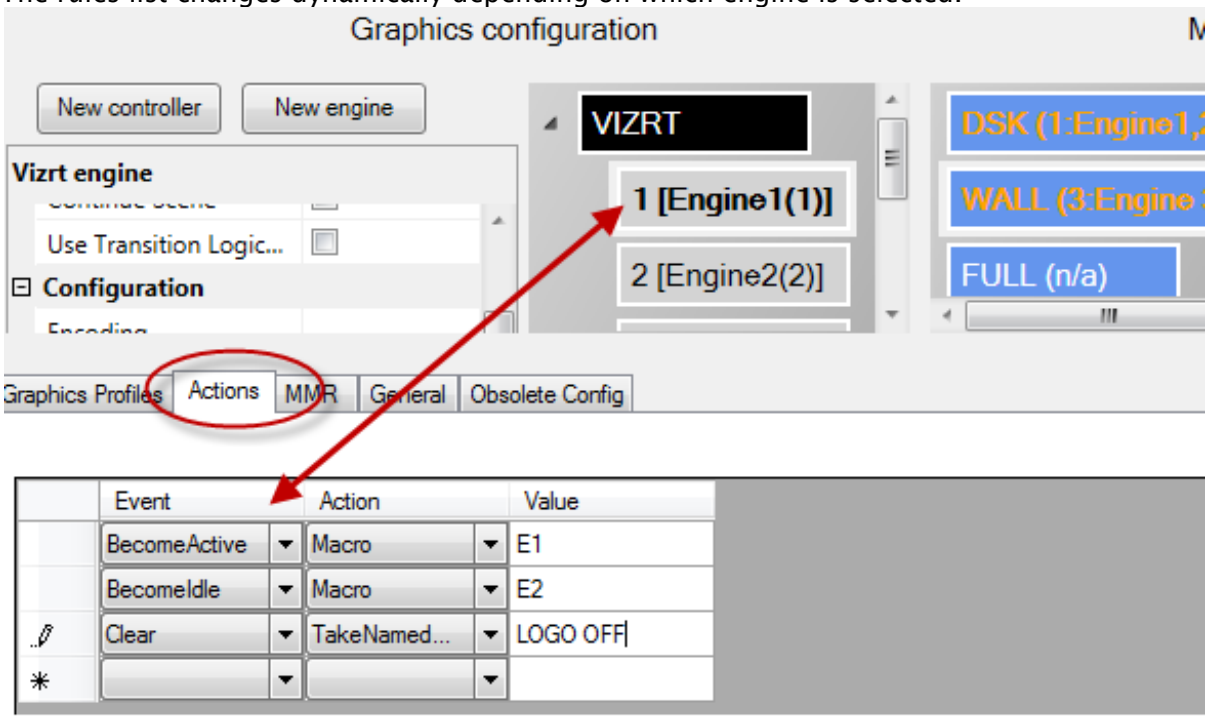
Configurations on this tab control *Event* and *Action* rules.

To set up event-action rules

1. Navigate to **Settings > Properties > Actions** tab.
   In the **Actions** tab, any number of event-action rules can be set up.



2. Select the engine that you want to configure these actions on.
   The rules list changes dynamically depending on which engine is selected.



3. Set required event-action rule by selecting parameters.
   You must define three parameters to set up an event-action rule:

- **Event:** Specifies the event that will trigger an action. The following events are available:
  - **BecomeActive:** Triggers an **Action** whenever the Overlay Graphics Interface state changes from **Idle** to **Active**.
  - **BecomeIdle:** Triggers an **Action** whenever the Overlay Graphics Interface state changes from **Active** to **Idle**.
  - **Clear:** Triggers and **Action** whenever a **Clear** event is received to clear all overlay graphics. A clear event is typically received when a rundown is reloaded or deleted.
  - **MirrorActive:** Triggers an **Action** when the mirroring graphics engine becomes active (for mirroring). Can only be used only when using Mirrored Graphics Playout.

Copyright © 2024 Vizrt

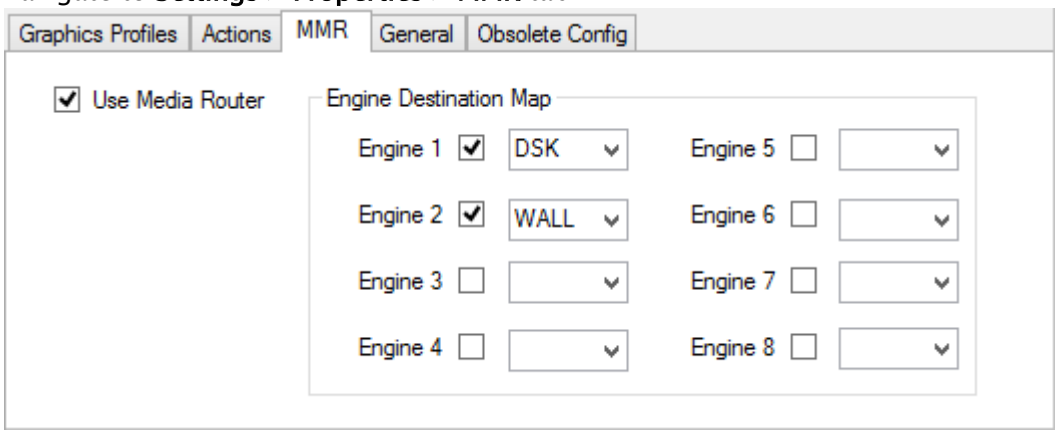
    - **MirrorInActive:** Triggers an **Action** when the mirroring graphics engine becomes inactive (for mirroring). Can only be used only when using Mirrored Graphics Playout.
- **Action:** Specifies which action to trigger for the given event. Two actions are available:
    - **Macro:** Invokes a macro on the given engine. The macro to take is specified in the **Value** parameter.
    - **TakeNamedOverlay:** Invokes a named (or constant) CG on the given engine. The named CG to take is specified in the **Value** parameter.
- **Value:** Optional value depending on selected **Action**. As shown above, there may be multiple event-action rules for a controller, for example:
    - **Add** a new event-action rule by setting values in the row at the bottom, marked with a star '*'.
    - **Selecting** the corresponding row is done by clicking in the column left of the **Event** column.
    - **Delete** existing event-action rules by selecting the corresponding row, and then press **DELETE** on the keyboard.

MMR Tab

Configurations on this tab affect behavior of the Mosart Media Router.

To configure the Mosart Media Router

- Navigate to **Settings > Properties > MMR** tab.



Explanation of configurable fields

- **Use Media Router:** To use the Media Router (MMR), this must be enabled (checked). The use of Media Router by Overlay Graphics Interface must also be enabled in General **> Media Router**.
- **Engine Destination Map:** Maps between Engine number and graphics destination. The Engine number is part of the Media Router configuration and is given as a *DeviceName* property. For more information see the Viz Mosart Media Router Guide. Use the check boxes to select a maximum of eight engines. In the example above the **MMR** configuration specifies *two* overlay graphics engines with numbers *1* and *2* respectively.

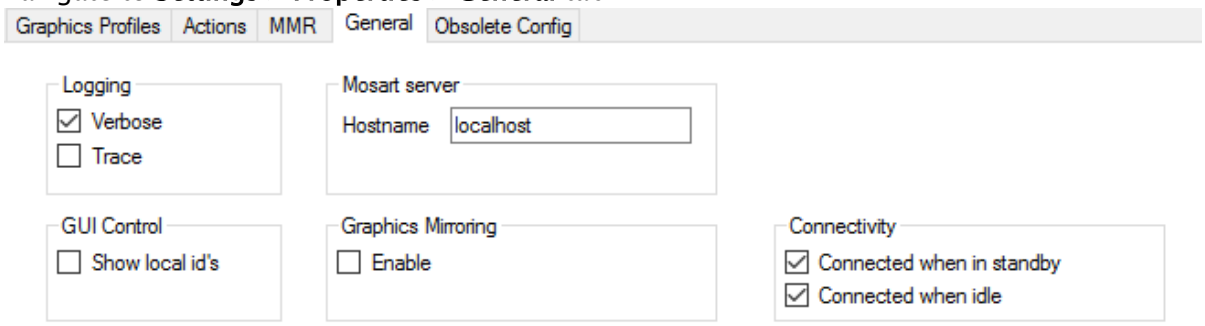
Copyright © 2024 Vizrt

General Tab

On this tab are several configurations that affect Viz Mosart Overlay Graphics operation and behavior.

To configure general settings

- Navigate to **Settings > Properties > General** tab.



Explanation of configurable fields

- **Logging**
  - **Verbose:** Increases the detail level of information sent to the Viz Mosart log if enabled.
  - **Trace:** Activates internal tracing (more details are sent to the debug view) if enabled. Used for debugging purposes only.
- **GUI Control**
  - **Show local id's:** Enables display of graphics ID in the OverlayGraphics GUI. Shown in dedicated columns in both repository and log tables.
- **Mosart server**
  - **Hostname:** Defines the hostname or IP number of the Manus Administrator. Default: *localhost*.
- **Graphics mirroring**
  - **Enable:** Allows simultaneous control of two different graphics engines driven in mirroring by Viz Mosart. When enabled, both engines will receive "take" commands at the same time. For more details, see Mirrored Graphics Playout
  - **Connectivity**
    - **Connected when in standby:** Maintains connection to graphics devices when in standby or idle modes if enabled.
    - **Connected when idle:** Maintains connection to graphics devices when put on idle, for example when switching from main to backup if enabled (Default = *disabled*).

> ⚠ **Note:** A similar parameter is available for full screen graphics, see **_GraphicsConnectedWhenIdle _** in AV Automation Settings (see Graphics).

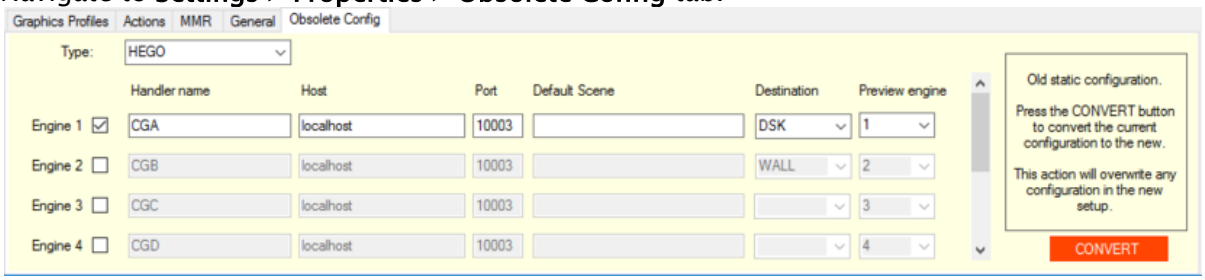
Copyright © 2024 Vizrt

Obsolete Config Tab

This tab is used to convert overlay graphics configuration from Viz Mosart 3.4 or earlier versions.

To convert an incompatible overlay graphics configuration

1. Navigate to **Settings > Properties > Obsolete Config** tab.



> **IMPORTANT!** If you have configuration based on Viz Mosart 3.5 or later, then *DO NOT* use the **Obsolete Config** tab.
> Only if you have a configuration based on Viz Mosart *3.4 or earlier*, is it strongly recommended to convert your configuration to the new standard.

   This tab displays the configuration parameters as they were in Viz Mosart 3.4 and earlier, and is used to update old configurations to the new standard.
2. Click the **CONVERT** button.
   This overwrites any existing standard configuration.

The fields in this tab are:

- **Type:** The type of graphics used
- **Engine 1-8 (check box):** Check the graphics engines to be active. When converting to new configuration, only active engines are converted.
- **Handler Name:** Default CGA, CGB etc.
- **Host:** Hostname or IP-address of graphics engine.
- **Port:** TCP/IP port number of graphics engine.
- **Default scene:** Specifies the graphics destination to be associated with the graphics engine. The graphics destination is assigned to graphics received from NCS.
- **Preview engine:** The number of the engine to use as a preview engine.

Copyright © 2024 Vizrt

7.6 Mosart Graphics Destinations And Takeout Logic

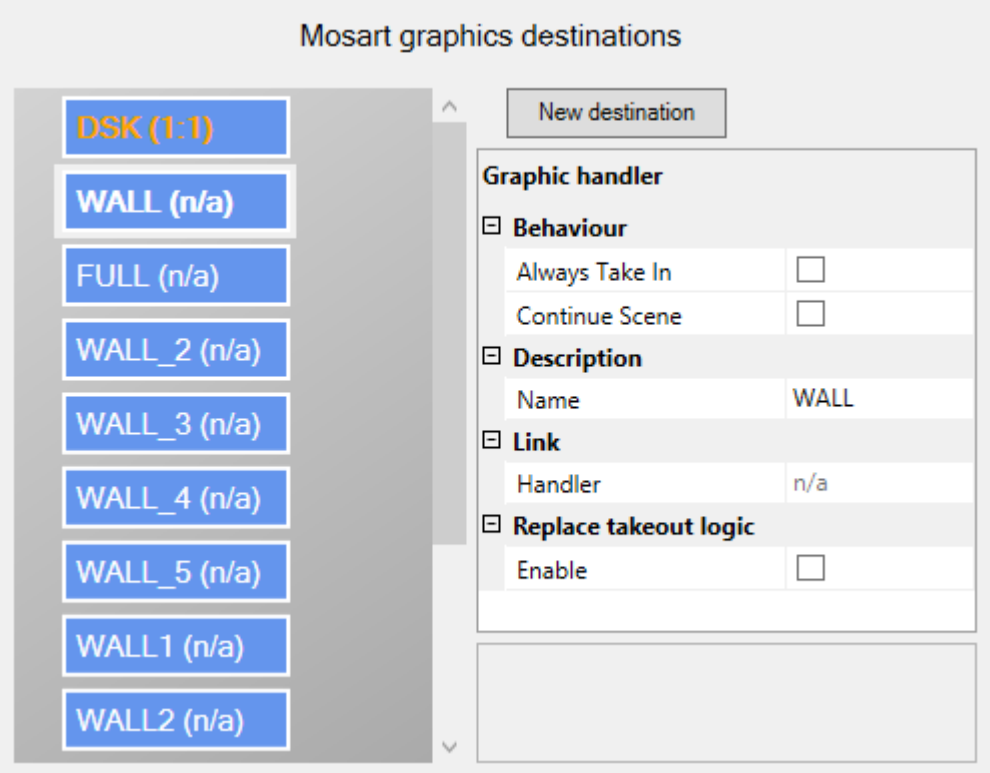
This section presents configuration of overlay graphics for various system setups and the use of takeout logic.

- Mosart Graphics Destinations
  - Overlay Graphics Interface
  - Graphic Handler Properties
  - Mirrored Graphics
- Takeout Logic
  - Introduction
  - Enabling Take Out Logic
  - Prerequisites for Take Out Logic

7.6.1 Mosart Graphics Destinations

Overlay Graphics Interface

The Overlay Graphics Interface provides definitions to a set of standard Mosart graphics destinations. These include WALL handlers (WALL, WALL_2 to WALL_5), FULL, and TABLE. It is recommended that you keep each definition as-is.



To Add a New Mosart Graphics Destination

- Click **New destination**, then fill in the **Graphic handler** properties.

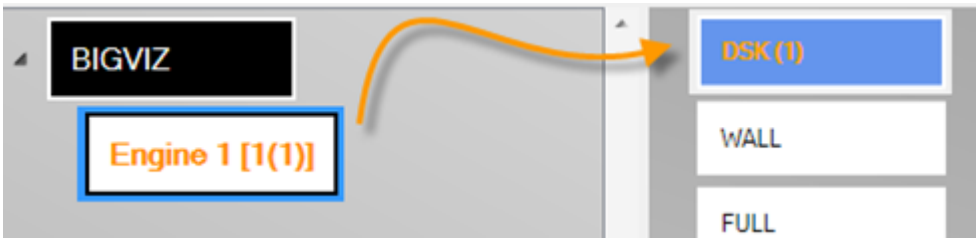
Graphic Handler Properties

The properties displayed depend on the graphic handler used. For some handlers (e.g. Viz Engine) the full set of or possible properties will be shown. For others, a simplified set of properties will show: **Description** and **Link**.

- **Behavior**:
  - **Always Take In**: Turns off the logic that does not take in a graphic element that is already taken and no other graphic element has been taken since.
  - **Continue Scene**: Use continue for take out of scene based graphics.
- **Configuration**:
  - **Preview Engine**: Handler name used for preview.
- **Description**:
  - **Name**: Name of the graphics destination.
- **Link**:
  - **Handler**: Shows the engine that is linked to this destination. The engine is shown as Handler ID: Handler Name.
- **Replace takeout logic**:
  - **Enable**: Reads out the Transition Logic context of elements and use this to either suppress take out or re-take items. Take outs will be suppressed if another item with the same context was taken before the scheduled for take out action. Take outs will be replaced with re-takes if the item scheduled for take out replaced an item with the same context.

To Link an Engine to a Destination

- To create a link from an engine to a destination, drag an engine node to the preferred destination.
  The destination label and Handler property will be updated to show the Handler ID and Name of the engine.



To Remove Engine Links from a Destination

1. Mark an engine.
2. **Right-click** to display the context menu context menu and select **Remove**.

Copyright © 2024 Vizrt

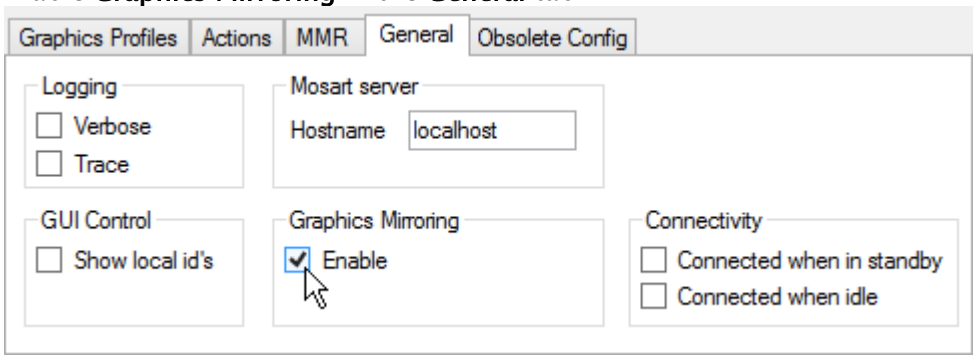
Mirrored Graphics

Mirrored Graphics Playout

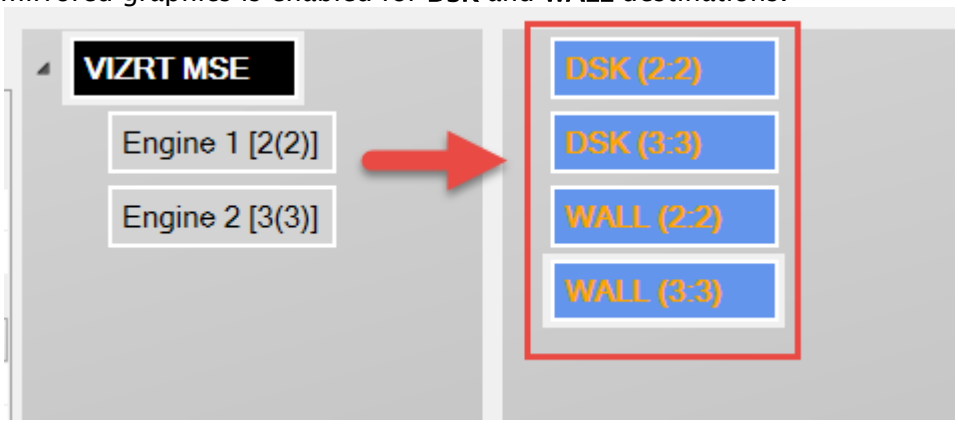
The feature of playing out the same graphics through multiple graphics engines is called Mirrored Graphics Playout. This functionality is supported both for overlay graphics and full-screen graphics (AV Automation). Here we describe the approach used in Overlay Graphics Interface.

> ⚠ For using Mosart Media Router (MMR) in conjunction with Mirrored Graphics, see Mosart Graphics Destinations and Takeout Logic below.

1. Enable **Graphics Mirroring** in the **General** tab.



2. Set up multiple graphics engines with the same destination.
   Example: To configure mirrored graphics for a destination, create two destinations with the same name and assign a different engine to each destination. In the example below, mirrored graphics is enabled for **DSK** and **WALL** destinations.



3. (Optional): Configure actions to be triggered when changing the active mirrored graphics engine.
   Do this from the Actions tab.

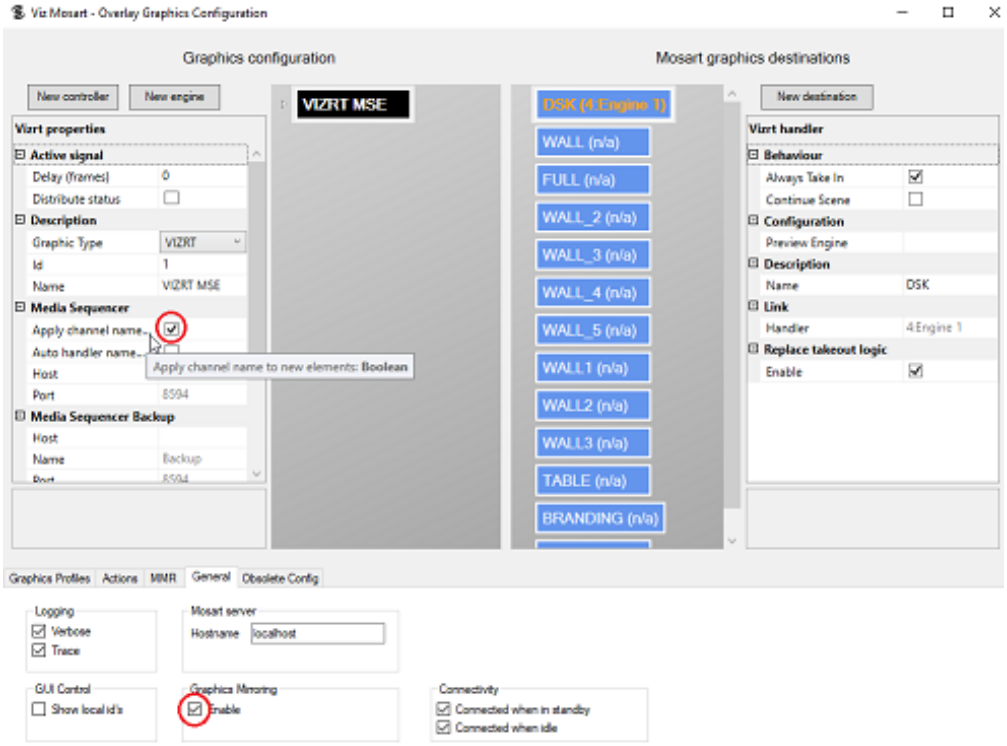
Mosart Media Router (MMR) and Mirrored Graphics

If your setup employs MMR for mirrored graphics, the integration requires additional configuration to ensure that all new graphics elements are tagged with their corresponding destinations in the Mosart playlists.

Copyright © 2024 Vizrt

- Ensure the **ApplyChannelNameToNewElements** property located in DefaultGraphicsConfiguration.xml, is set.

This is also configurable from the UI:



This is described in detail in the Viz Mosart Media Router Guide, in the section **MMR Configuration of Vizrt Graphics**.

7.6.2 Takeout Logic

- Mosart Graphics Destinations
- Takeout Logic

Introduction

For overlay graphics, Viz Mosart supports four different methods to take out any graphics that are currently on air:

1. **Automatic**: Timed, where graphics are taken out automatically after a fixed duration, provided in a timecode.
2. **Background End**: The graphic is taken out automatically, when the next story item is taken.
3. **Story End**: The graphic is taken out automatically when the first story item of the next story is taken
4. **Manual**: Open ended, the graphic is taken out manually. Also called a *non-auto out*.

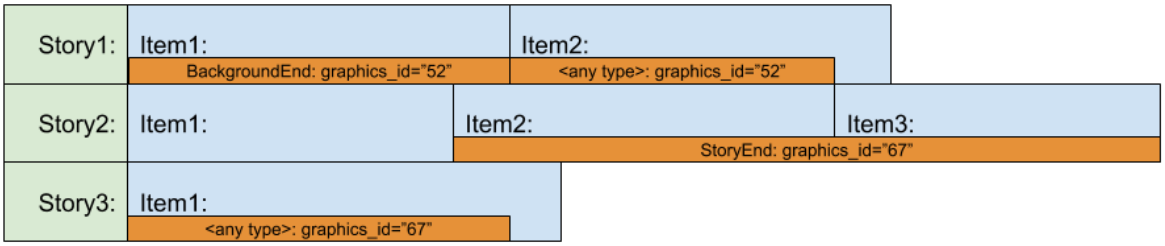
For the *Background end* and *Story end* cases, there is a potential issue if the graphic to be taken out reappears in the next story item. When taking the next story item, the standard behavior of such graphics is:

Copyright © 2024 Vizrt

1. Take out the graphics due to the end of scope (either *Background end* or *Story end*)
2. (Re)take in the graphics since the graphics is also present in the story item to be taken.

This leads to a visual artifact on air, since the graphics will be take out and retaken within a short time interval.

This is illustrated in the following figure:



Example 1: Background End

Story1 consists of two story items, *Item1* and *Item2*.

- In *Item1* there is an overlay graphics of type *BackgroundEnd.*
- In *Item2* the same graphics is reused. The type is not relevant here.
- When the user takes *Item2* on air, the normal behavior is to take out the graphics (out of *Item1*) and then (re)take it (also part of *Item2*).
- If Take Out Logic is enabled, the graphics will remain on air during the transition from *Item1* to *Item2.*

Example 2: Story End

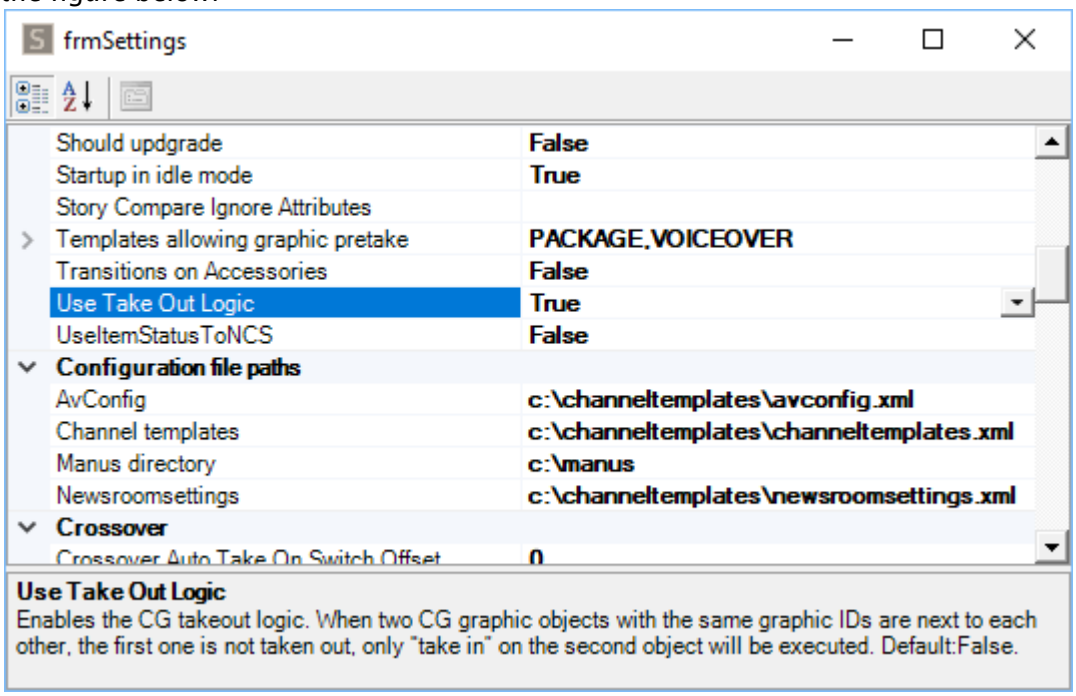
Story2 consists of three story items, *Item1*, *Item2* and *Item3* and the next story, *Story3* consists of one story item, *Item1*

- In *Story2.Item2* there is an overlay graphic of type *StoryEnd*. I.e. this graphic is scheduled to be taken out at the end of the story.
- In *Story3.Item1* the same graphics is reused. The type is not relevant here.
- When the user takes *Story3* on air, normal behavior is to take out the graphics (out of *Story2.Item3*) and then (re)take it (also part of *Story3.Item1*).
- If take out logic is enabled the graphics will stay on air during the transition from *Story2* to *Story3.*

Copyright © 2024 Vizrt

Enabling Take Out Logic

- Take out logic is enabled via setting **Use Take Out Logic** in Manus Administrator as shown in the figure below:



Note: Enabling take out logic applies to non-Vizrt graphics only.

Prerequisites for Take Out Logic

For take out logic to work, there must be a unique way of identifying the graphics.

There are two ways to do this:

1. Use the *graphics_id* graphics property. The *graphics_id* property is used by most graphics drivers to do actions on the graphics. Therefore it can normally be used as a unique identifier for the graphics. For most graphics systems used by Viz Mosart, there is a defined way of setting this property when receiving graphics from NRCS.
2. Use the *graphics_category* graphics field property. This is a special property for the take out logic. Use this if graphics with different *graphics_id* shall be treated as the same graphics and handled as part of the take out logic. This normally requires custom field mapping in configuration file *newsroomsettings.xml*.

Copyright © 2024 Vizrt

7.7 Overlay Graphics Types

Each graphic provider is stored as a **Graphics Type.** Several Graphics Types are currently supported. These include:

- VIZRT
- VIZ FLOWICS
- CHYRON
- DEKO
- ORAD
- PIXEL POWER CC
- XPRESSION
- BIGVIZ

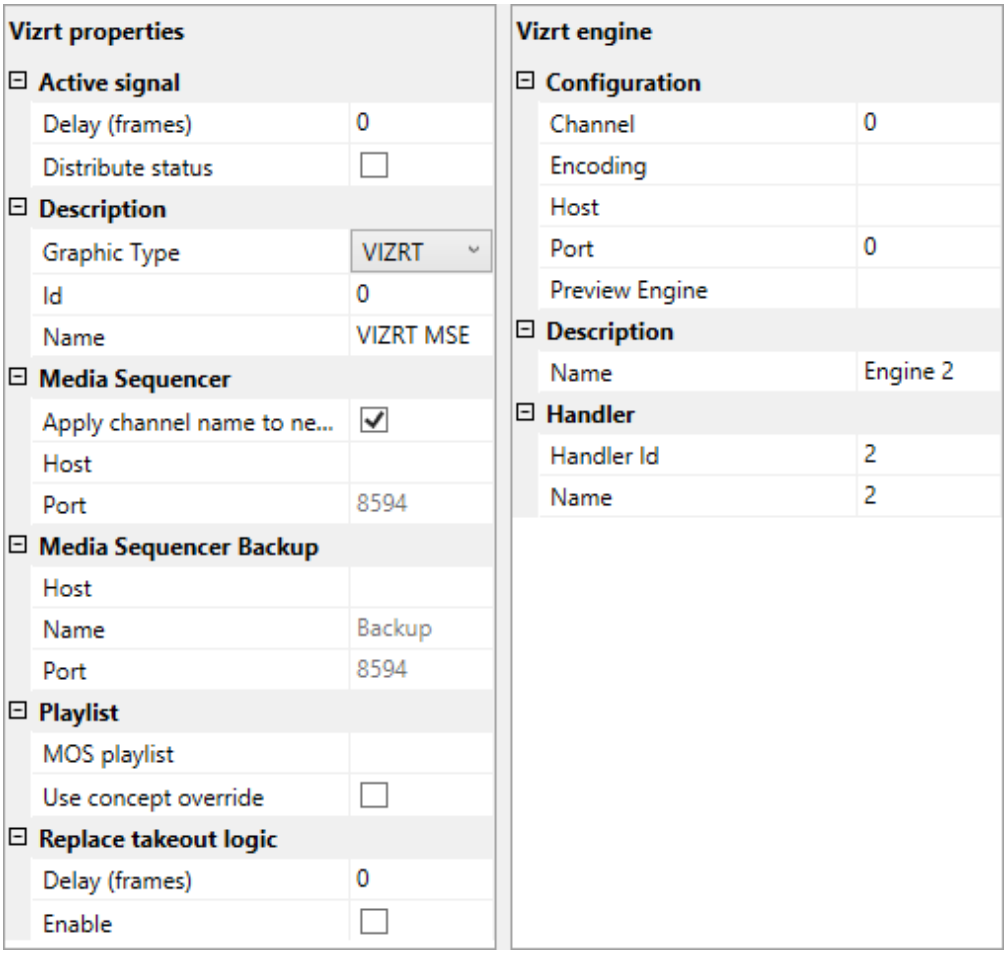
All operations are managed from the **Overlay Graphics Configuration** menu.

- Open the **Overlay Graphics Configuration** menu by opening the console application **Viz Mosart Overlay Graphics Interface > Settings > Properties**.

7.7.1 VIZRT

Graphic Type = Vizrt (Vizrt via *Media Sequencer* [shortname: MSE])

Copyright © 2024 Vizrt



- **Vizrt properties:** Properties in addition to Common Graphic Controller Properties:
  - **Media Sequencer**
    - **Apply channel name to new elements:** Assigns a channel name automatically to new graphics elements.
    - **Host:** Hostname or IP address to the Media Sequencer.
    - **Port:** IP-port to the Media Sequencer. Normally `8594` (Media Sequencer TreeTalk protocol) or `8580` (REST protocol).
  - **Media Sequencer Backup:** For an optional backup MSE. Same properties as for Media Sequencer. Note that the **Apply channel name to new elements** property is common to both main and backup Media Sequencers.
  - **Playlist**
    - **MOS playlist:** Name of playlist in MSE where graphics elements are stored.
    - **Use concept override:**If concept override is enabled, the currently selected graphics concept will override any concept defined within the graphics element. Use the **Graphics Profile** tab to define the various graphics concepts.
  - **Vizrt engine**
    - **Configuration**
    - **Host:** Hostname or IP address of the **Vizrt Engine**. The input can be specified as just a Hostname, just an IP Address, or a combination of Hostname:Port or

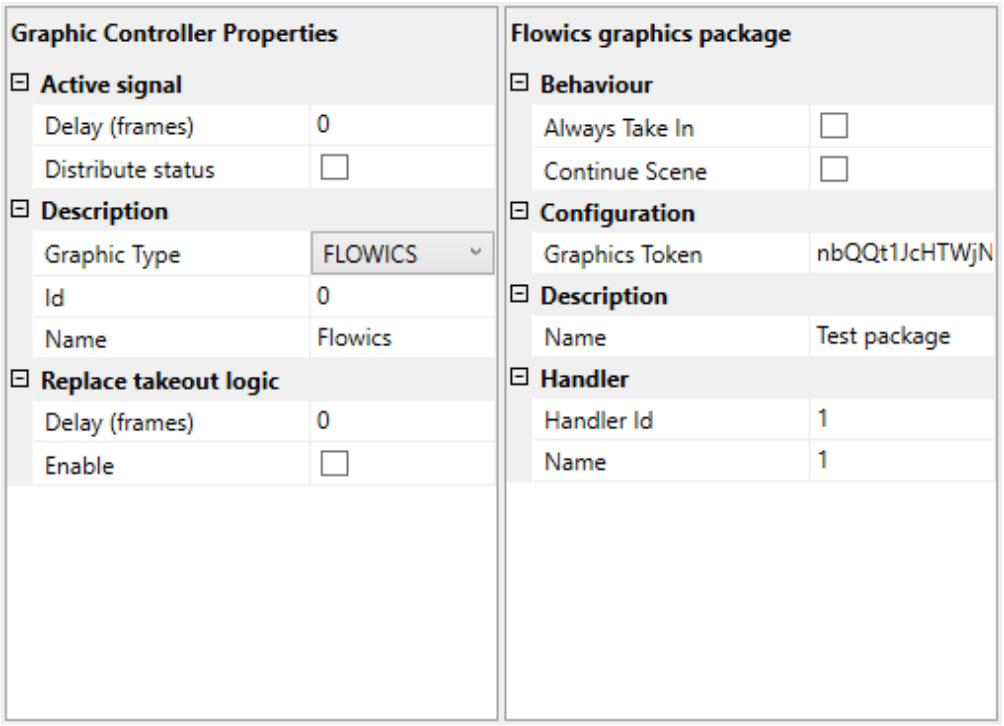
IPAddress:Port. If no port value has been specified as part of the **Host** field, the port from the **Port** field will be used. However, if no port value has been specified in either of the **Host** or **Port** fields, the default port value will be: `6100`.

> ⚠ **NOTE:** If a port value is specified in the **Host** field, this value will override whatever port value has been specified within the **Port** field.

- **Port:** TCP/IP port of the **Viz Engine**.

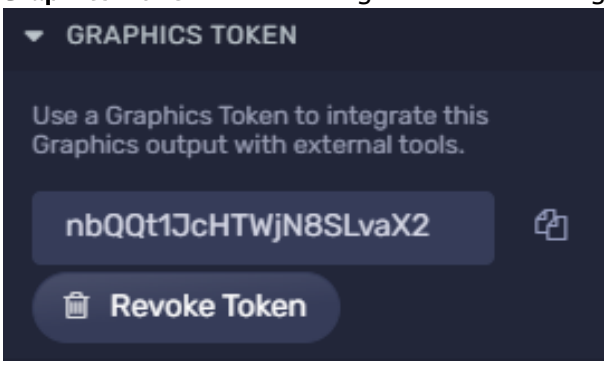
7.7.2 VIZ FLOWICS

Graphic Type = Flowics (Viz Flowics)



- **Graphics Controller properties:** The properties are the same as described in Common Graphic Controller Properties.
- **Flowics graphics package:** Contains the following properties in addition to Common Graphic Engine Properties:
  - **Configuration:**

- **Graphics Token:** The token generated for the given Graphics package

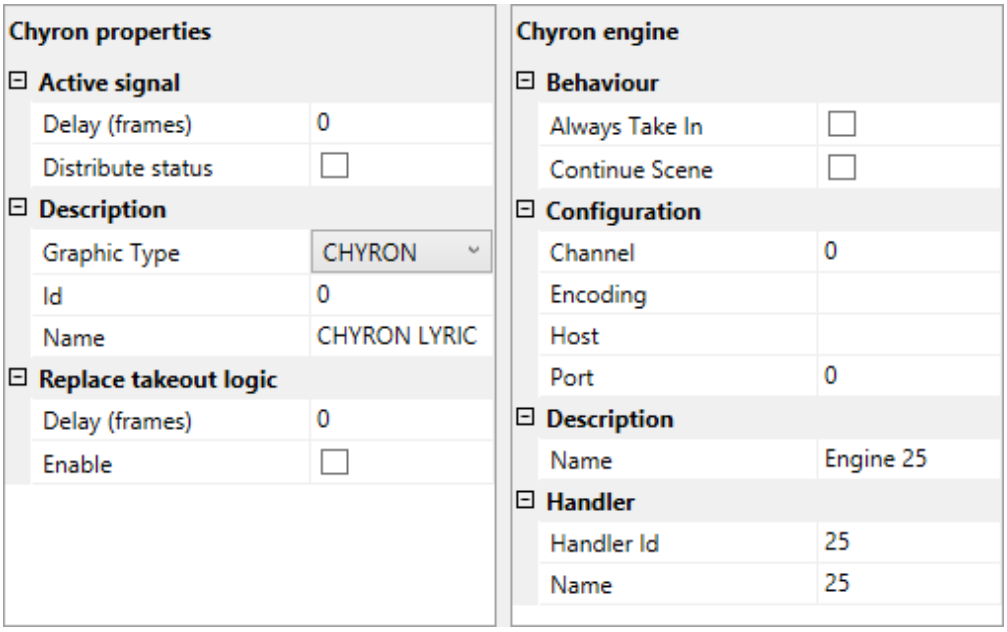


- **Description:**
  - **Name:** The name of the Graphics package

> ⚠ You need to install and configure the Viz Flowics MOS Gateway for this integration to operate.

7.7.3 CHYRON

Graphic Type = Chyron (Chyron Lyric)

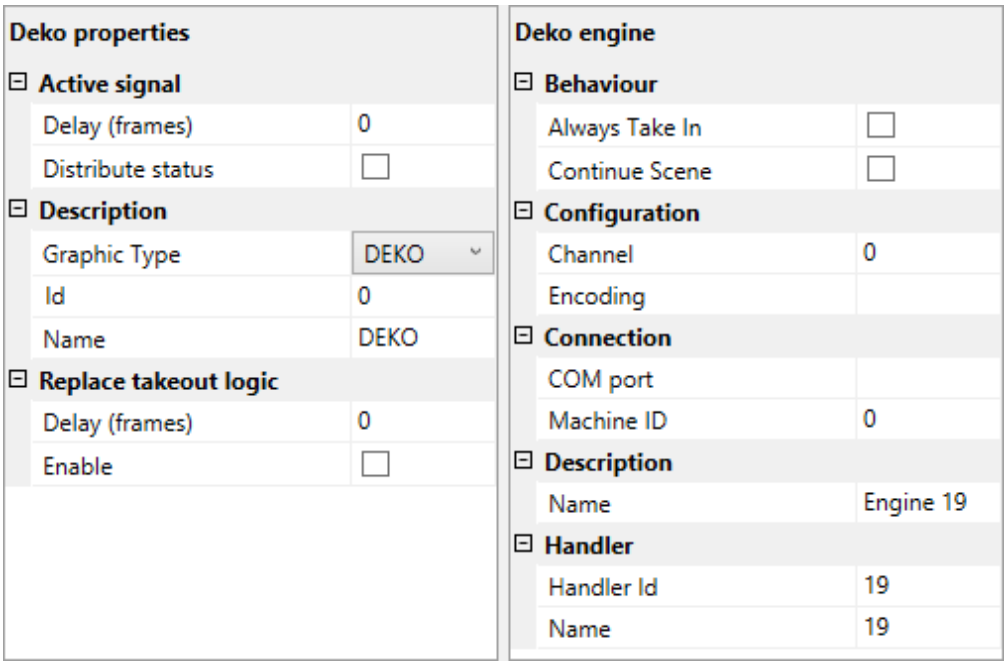


- **Chyron properties:** The properties are the same as described in Common Graphic Controller Properties.
- **Chyron engine:** The properties are the same as described in Common Graphic Engine Properties.

Copyright © 2024 Vizrt

7.7.4 DEKO

Graphic Type = Deko



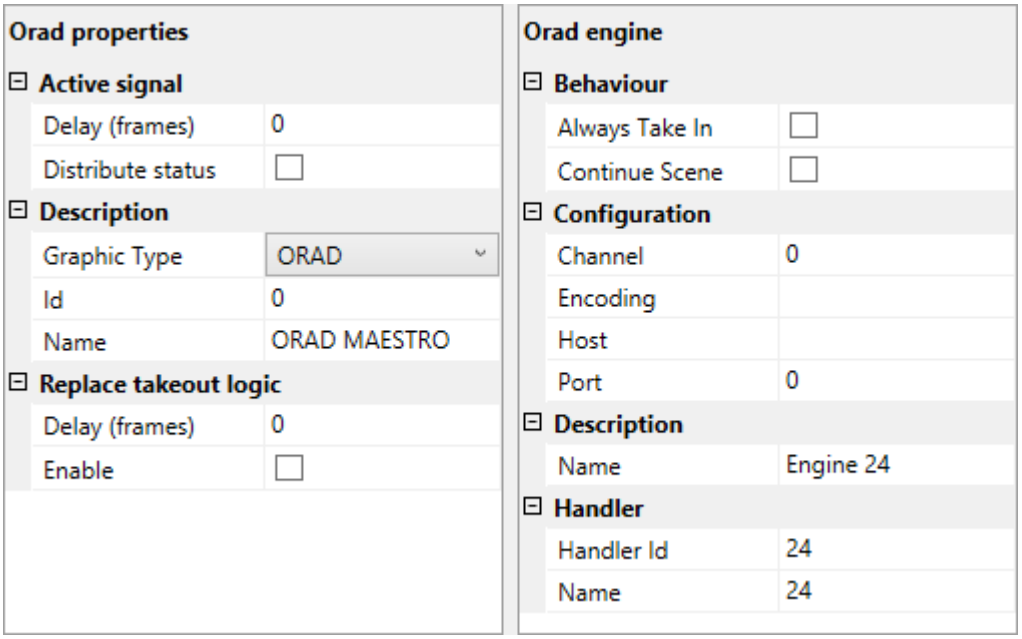
- **Deko properties:** The properties are the same as described in Common Graphic Controller Properties.
- **Deko engine:** Contains the following properties in addition to Common Graphic Engine Properties:
  - **Connection:**
    - **COM port:** Defines the serial port connected to the Deko engine.
    - **Machine ID:** Digit 0-9. May be used for multi-drop. Multi-drop lets you specify different fields or layers (TypeDeko) when more than one TypeDeko is connected to the Deko host. Set equal to `1` if there is only one.

> ⚠ **Note:** For using a default take out scene in Deko, the setting *<item name="ClearScene" value=""/>* needs to be added to the IntelligentInterfaceConfiguration.xml file instead of the clear command. The value should be set to any scene that should be used for take out.

7.7.5 ORAD

Graphic Type = Orad (Orad Maestro)

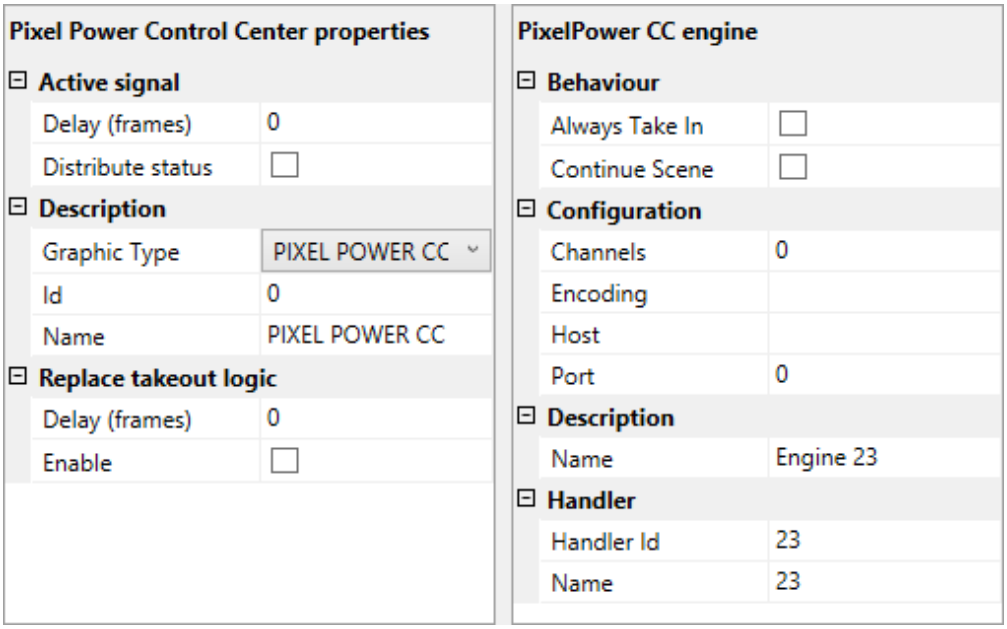
Copyright © 2024 Vizrt



- **Orad properties:** The properties are the same as described in Common Graphic Controller Properties.
- **Orad engine:** The properties are the same as described in Common Graphic Engine Properties.

7.7.6 PIXEL POWER CC

Graphic Type = Pixel Power CC (Pixel Power Control Center)



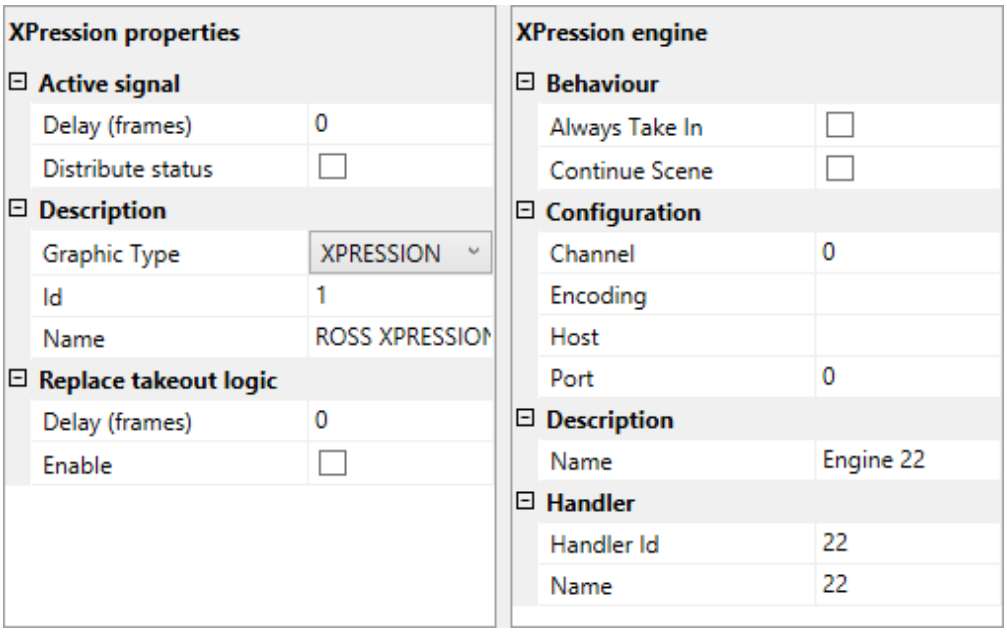
- **Pixel Power Control Center properties:** The properties are mostly the same as described in Common Graphic Controller Properties.

Copyright © 2024 Vizrt

- **Pixel Power CC engine:** The properties are the same as described in Common Graphic Engine Properties.
  - **Configuration**
    - **Channels:** Specifies the Pixel Power channels associated with the Pixel Power engine. Specify as a comma-separated list of integers.

7.7.7 XPRESSION

Graphic Type = Xpression (Ross Xpression)

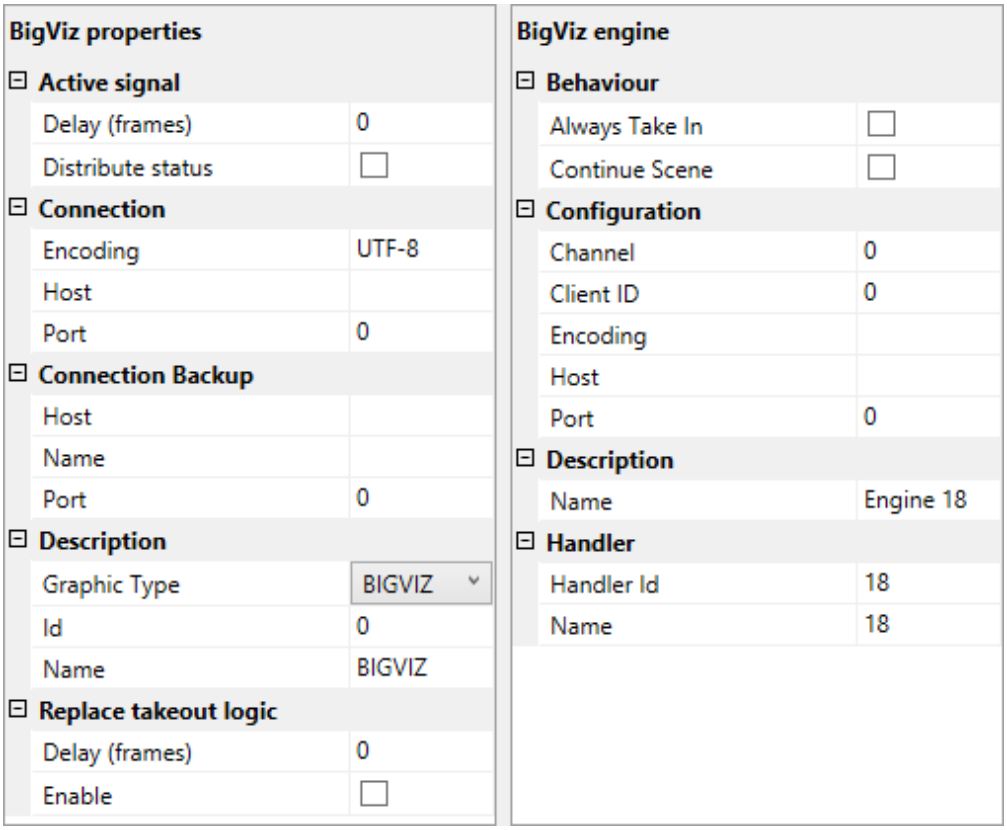


- **XPression properties:** The properties are the same as described in Common Graphic Controller Properties.
- **XPression engine:** The properties are the same as described in Common Graphic Engine Properties.

7.7.8 BIGVIZ

Graphic Type BigViz is a BBC developed system that integrates with ENPS, Vizrt graphics, IBIS Playout, and Autoscript.

Copyright © 2024 Vizrt



- **BigViz properties:** Controller connection properties are not used. All connectivity to BigViz is configured as part of the Viz Engine configuration.
- **BigViz engine:** The following properties are used only (all other properties related to Viz Graphics configuration are not used by BigViz):
  - **Host:** Hostname or IP address of BigViz service.
  - **Port:** TCP/IP port of remote BigViz service. Default port is `8165`.
  - **Client ID:** Identifies the Mosart server. Must match the corresponding BigViz configuration. Default: **mosart**.

Copyright © 2024 Vizrt

7.8 Sending Lower Third Graphics MOS Object To An NRCS

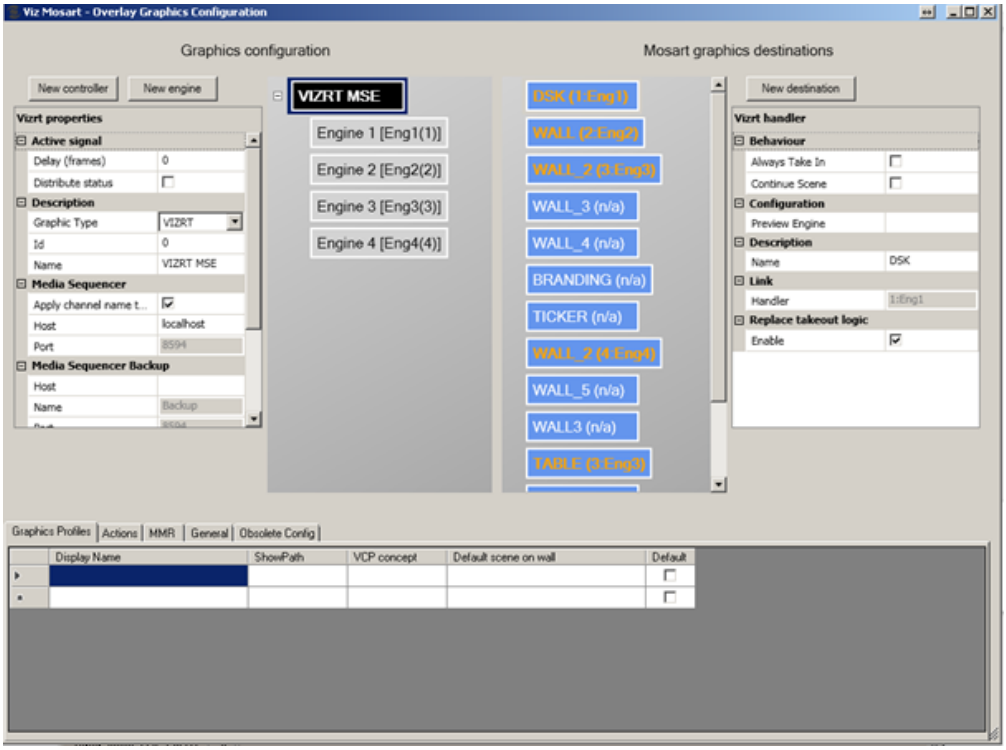
You can send details to a MOS-based NRCS as MOS objects.

For all template details as MOS objects in the NRCS, see section Mosart Templates Feedback to the NRCS.

7.8.1 Managing Destinations

Active Destinations

The list of channels contains the active destinations is configured in the Overlay Graphics Interface. *Active destinations* are destinations linked to a graphic engine.



In the configurations above, the following engine-destination mapping is performed:

Destination	Engine number
DSK	1
WALL	2
WALL_2	3

Destination	Engine number
WALL_2	4
TABLE	3

MOS Format

The list of channels in the LOWERTHIRDS MOS object will be:

```
<keys name="channels">
<k>WALL_2</k>
<k>WALL</k>
<k>TABLE</k>
<k>DSK</k>
</keys>
```

The active channels from Overlay Graphics are sent to the NRCS only if the Overlay Graphics Interface is started.
This is the recommended workflow to correctly send the channels configured in Overlay Graphics Interface to NRCS.
The steps in square brackets are optional:

1. Start Mosart Administrator Mos application
2. Start Overlay Graphics
3. [Do any configuration changes if necessary in OverlayGraphics Settings]
4. [Restart OverlayGraphics or do Ctrl+Shift+G in case any configurations have been changed]
5. [Verify that %localappdata%\Temp\Mosart Medialab\MOS State\ MosTemplateCache.xml contains expected data]
6. Execute mosReqObjList or mosReqAll from NCS

7.8.2 Detailed Description of MOS Objects

See the Appendix for code-level descriptions of MOS Object with Lowerthird Information.

Copyright © 2024 Vizrt

8 Show Design And Template Editor

Central to reliable and error-free productions is the Viz Mosart concept of *templates*. A template is pre-programed before the production during *show design*, with help of the Viz Mosart utility, *Template Editor*.

Each template performs a device-specific operation, like studio camera selection or playing a specific video clip from a video server.

This section introduces some examples of how complex studio device operations can be programed onto a template, and walks you through show design, where the template operations are programmatically chained together in a Viz Mosart *rundown*.

- Overview
- Template Examples
- Working with Templates
- Other Template Functionality
- Working with Template Sets
- Newsroom Tags
- Template Device Functions

Copyright © 2024 Vizrt

8.1 Overview

Behind each operation, as displayed on the timeline in Viz Mosart's Rundown view, is a Viz Mosart template.

A Viz Mosart template is the key element of controlling external devices and events handled by Viz Mosart. A template can send simultaneous control commands to multiple studio devices.

Typically, the actions defined on a template are either:

- **Automatic**
  Either invoked from another template or at a given time as part of the rundown
- **Preconfigured**
  Configured in the newsroom control system (NRCS or NCS) as part of a rundown, and then manually executed by the Viz Mosart operator
- **Manual**
  The operator initiates the template-driven operation, usually from a keyboard shortcut or touch panel.

Some essential template concepts are introduced below:

- Template Sets
- Working with Templates
- Template Terms and Definitions

8.1.1 Template Sets

Templates are organized in *template sets*. Typically, there will be a template set for each show/program. As Viz Mosart runs, one template set will be *current*, meaning templates that will be executed during the show are drawn from this set.
An alternative set can be selected as the new current set, see Overview below. The templates that comprise a show can be arranged in two ways:

- Overview
- Overview

Standalone

Traditionally, Viz Mosart template sets are treated individually, with no connection to each other.

For example, a template set is created for each show or program. This is usually combined with automatically selecting the template set through the associated NRCS rundown. In addition, startup commands for preparing the studio (lights, sound etc.) for running the show are included in the template set.

When creating a template set for a *new* program, it is based upon a copy of an existing template set, with small adjustments according to the functionality needed by the program.

When this new template set is created (by copying an existing set), a copy of *all* templates is made, and no references are made between the original and copied templates. Each template set is classified as ‘standalone’.

Copyright © 2024 Vizrt

Disadvantages with this approach:

- Many duplicates are created. Normally only a few of the templates in the set need to be adjusted for meeting the needs of a new program.
- With an ever-growing bank of templates being created for each new show, the amount of stored templates in Viz Mosart can exceed system norms. This results in long response times when saving templates and increases the memory footprint of both the Mosart Server and Viz Mosart client workstations.
- Maintenance of templates becomes cumbersome and error prone. When a change is made in a template that is common for all template sets, the same change has to be applied individually to every template set. This is a time consuming task across numerous template sets.

Hierarchical

An alternative approach to Standalone is *hierarchical* template sets. This method also enables a new template set to be created, based on the logic of an existing template set. However, to begin with, *no* templates are copied from the existing template set.
Only when a change is made to an existing template, or when a new template created, is a template created and stored in the new template set.
The hierarchical approach can be adopted with all existing (standalone) template sets, to optimize machine capacity and introduce a more efficient workflow.

Referring to the background discussion in Overview above, when you copy a template set, it is more efficient to use a hierarchical approach. These operations are usually performed by a system administrator, as outlined in Overview below.

Direct Takes

In addition to standard show-related template sets, there is a special template set called *Directtakes*.

Unlike standard templates, templates in this set are *not* intended to be executed as part of the rundown.
Instead, they are usually manually executed during a running show *immediately* in an ad-hoc fashion.

Often a unique numeric value is assigned to each direct take. The operator enters executes the direct take template by entering its *number* on a numeric keypad.

8.1.2 Working with Templates

You create and edit individual templates and template sets (standalone and hierarchic) in the AV Automation application, **Template Editor**.

Creating a Hierarchic Template Set

- Templates are created and customized during the 'show design' phase of a Viz Mosart system setup.

Copyright © 2024 Vizrt

This is a specialist operation, described in section **AV Automation > Template Editor** of the *Viz Mosart Administrator Guide* (can be browsed from Viz Mosart UI menu bar **Help > Viz Mosart Administrator Guide**).

Templates in the AV Automation Application

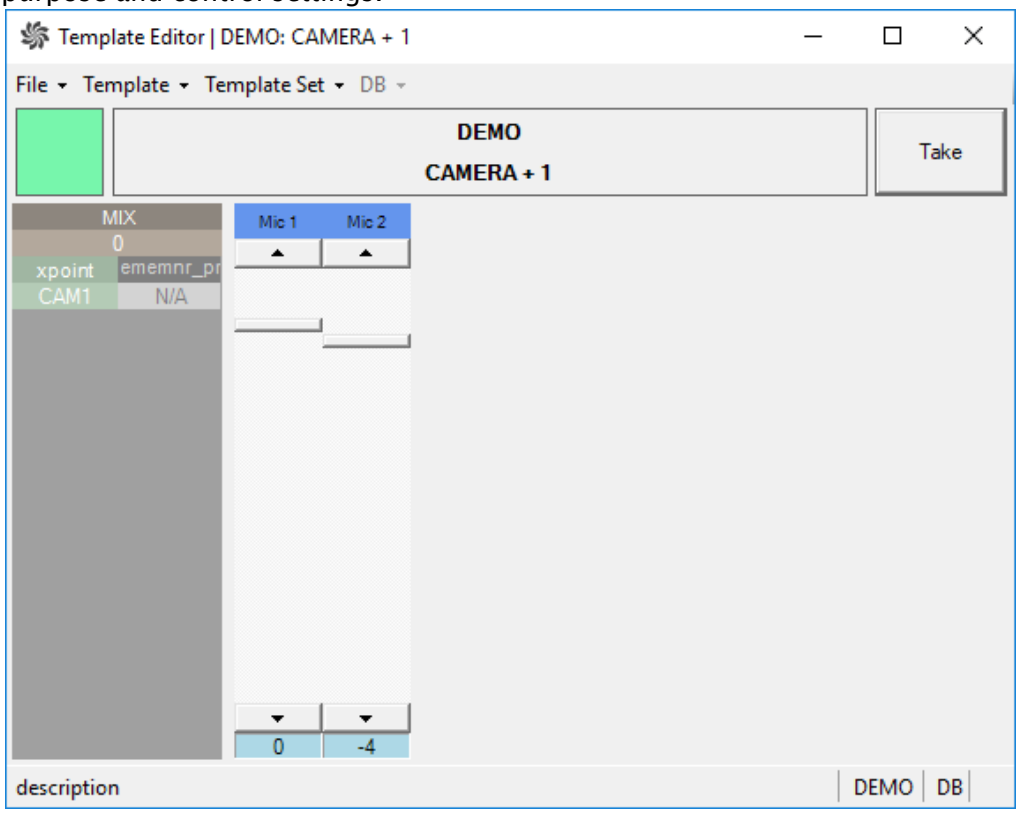
To create and modify show templates

- With a combination of mouse-over and contextual menus, an administrator can create and modify the templates that comprise a Viz Mosart-driven show.

This is presented in more detail later in this section.

To browse your show templates

- From the AV Automation console program (running on the Viz Mosart server), navigate to **Main menu > Devices > Template editor**.
  Templates displayed in Template Editor provide detailed information about their origin, purpose and control settings.



8.1.3 Template Terms and Definitions

Some common terminology used in template design and execution:

- **Base template set:** The start point for a show design.
  For example, if a template set *TS2* is based upon template set *TS1* then *TS1* is defined as the *base template set of TS2*. Programmatically, this relation can be written in short notation as *TS2 > TS1*
- **Current template set:** The active template set in Viz Mosart. Changing the current template set is normally done from either:
  - The Viz Mosart client, from the **Studio setups** menu.
  - The NRCS (for example, ENPS or iNEWS) using the Viz Mosart control command [STUDIO SETUP]
  - The Viz Mosart server application Av Automation, from the the **Studio setups** menu.
- **Directtakes:** A special template set, where templates in this set are intended to be manually executed in an ad-hoc fashion.
- **Empty template:** An undefined, non-operative template. An empty template does nothing when executed. Note that empty templates are *not* displayed in the Viz Mosart client.
- **Gallery:** Corresponds to a Viz Mosart installation. Typically a gallery has a dedicated Viz Mosart server, with configured utilities and a Viz Mosart client.
- **Inherited template set:** A template set based on another template set.
  For example, if *TS2 > TS1* then *TS2* is classified as an inherited template set.
- **Missing template:** A template that does not exist in the current template set. The operator is alerted to a missing template in the Viz Mosart client, where it is displayed with r*ed* text.
- **Shared template set:** The template set shared among all Viz Mosart installations that are connected to the same Viz Mosart template database.
- **Show design:** The background task of creating a set of templates, in a logical and programmatically valid order, for controlling a scheduled, rundown-driven, studio transmission.
- **Stand alone template set:** A template set not based on any other template set. All template sets in a current installation of Viz Mosart are classified as *standalone*. Programmatically, this relation can be written in short notation as *TS1 > ()*
- **Template description:** The template properties common for all Viz Mosart installations in a shared template set. This description uniquely identifies and describes the template.
- **Template implementation:** Contains the behavior or device control aspects of the template. The template implementation is related to the Viz Mosart installation. For a *shared* template set, each Viz Mosart installation will implement the template set differently, according to local studio setup.
- **Template set:** Viz Mosart templates are always contained in a template set. There may exist multiple *versions* of the same template but in *different* template sets. A template is uniquely defined by its *type* and *variant name* within its corresponding template set.
- **Template type:** Template type or category. Viz Mosart has a predefined set of template types. Each are displayed with a distinct visual appearance in Viz Mosart.
  Examples of template types are CAMERA, PACKAGE, VOICEOVER, LIVE, GRAPHIC and DVE.
- **Template variant:** Unique identifier of a template for given template type within a template set.
  For example, *camera1* and *camera2* are two variants of a camera template.

Copyright © 2024 Vizrt

8.2 Template Examples

Each major control room operation is driven by a unique template *type*. These operations are allocated to a *primary* template.

Examples of primary templates are *Camera*, *DVE*, *Package*, *Voiceover*, *Live*.

This section contains the following examples:

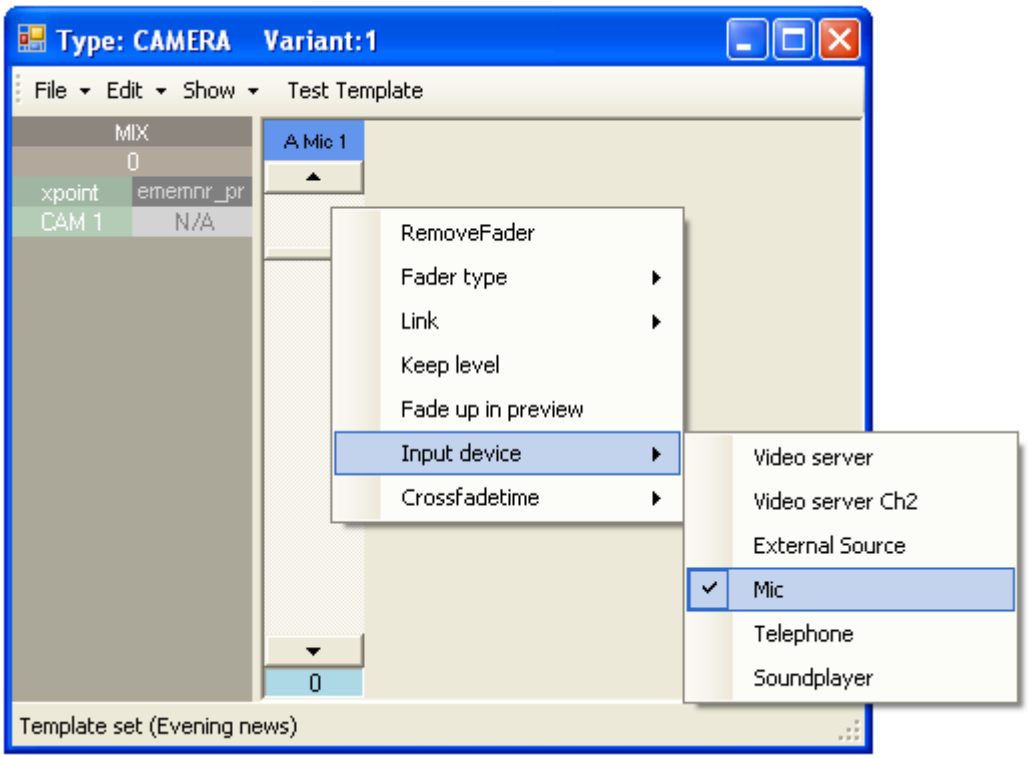
- Studio Camera
- Package: Video Clip with Full Sound
- Voice-over
- Live External Source
- DVE
- Graphic: Full Screen Graphics
- Accessory Templates
- Switcher Plugin

Full details of the operations displayed in the examples below are presented later in this section

8.2.1 Studio Camera

The template for a typical studio camera is based on the CAMERA type. The example below defines a standard camera 1 variant. An entry named CAM 1 in the video setup is selected at the PP bus cross point. Transition duration is `0` frames (hard cut) and E-mem recall is disabled. There is only one studio microphone, which fader is called "A Mic 1" in this template. Audio level is set to `0` dB, and the input device is a microphone. The input device type is important for the handling of audio in an ad lib sequence of full camera, DVE and full external source.

Copyright © 2024 Vizrt

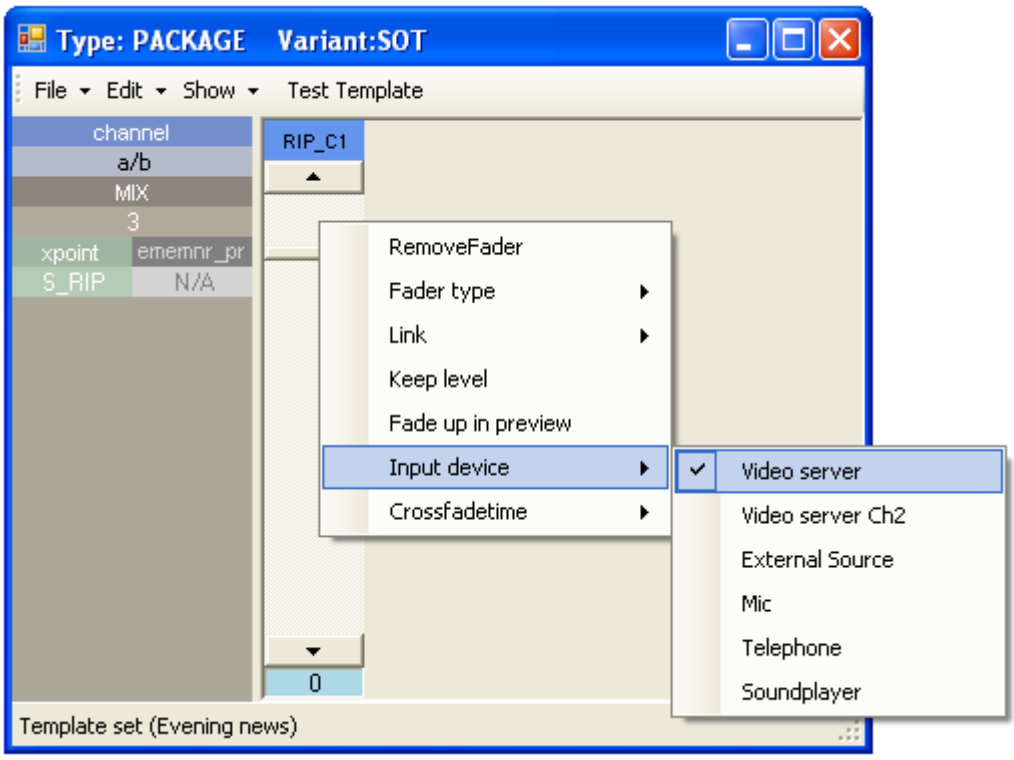


For our standard camera 2 variant, the only difference in the above setup is the PP bus cross point which is called CAM 2.

8.2.2 Package: Video Clip with Full Sound

As crosspoint on the vision mixer, we use a source called S_RIP, which means Viz Mosart uses the A/B roll and roll between the sources defined in the AV setup as *videoserverA* and *videoserverB*. We chose RIP_C1 as the audio source, but as long as A/B roll is enabled, any source defined as Video server follows the video source. The default transition is a three frame mix. The video channel control is also enabled and controls the clip. The server channel is automatically assigned within the automation.

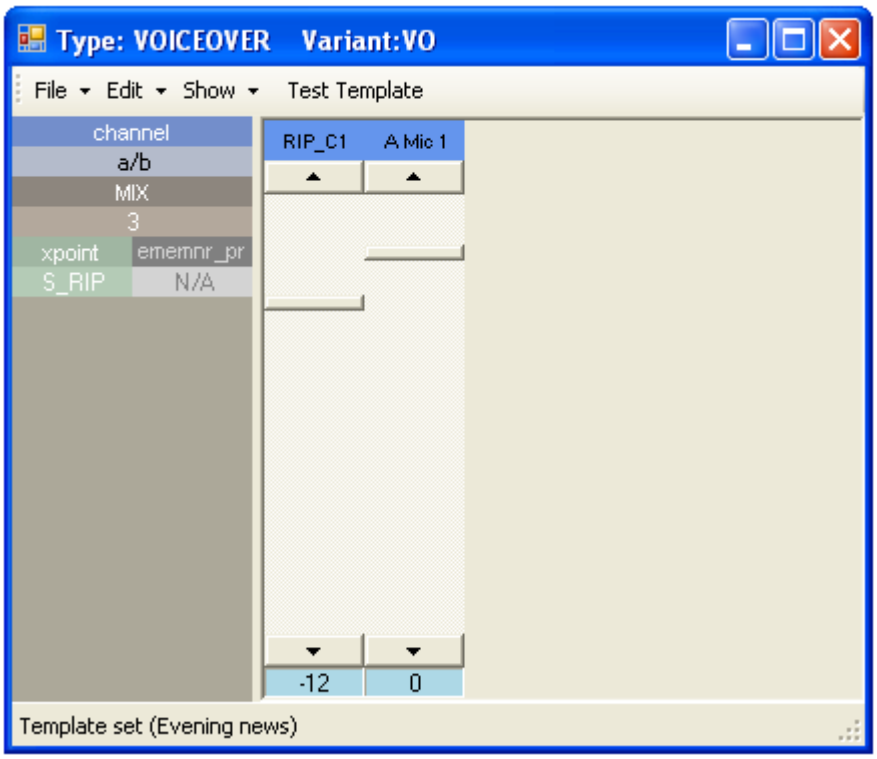
Copyright © 2024 Vizrt



8.2.3 Voice-over

In this example, the variant VO is defined. Notice the difference from the two last examples and the two fader setups. A Mic 1 is microphone input type and RIP_C1 is video server type.

Copyright © 2024 Vizrt

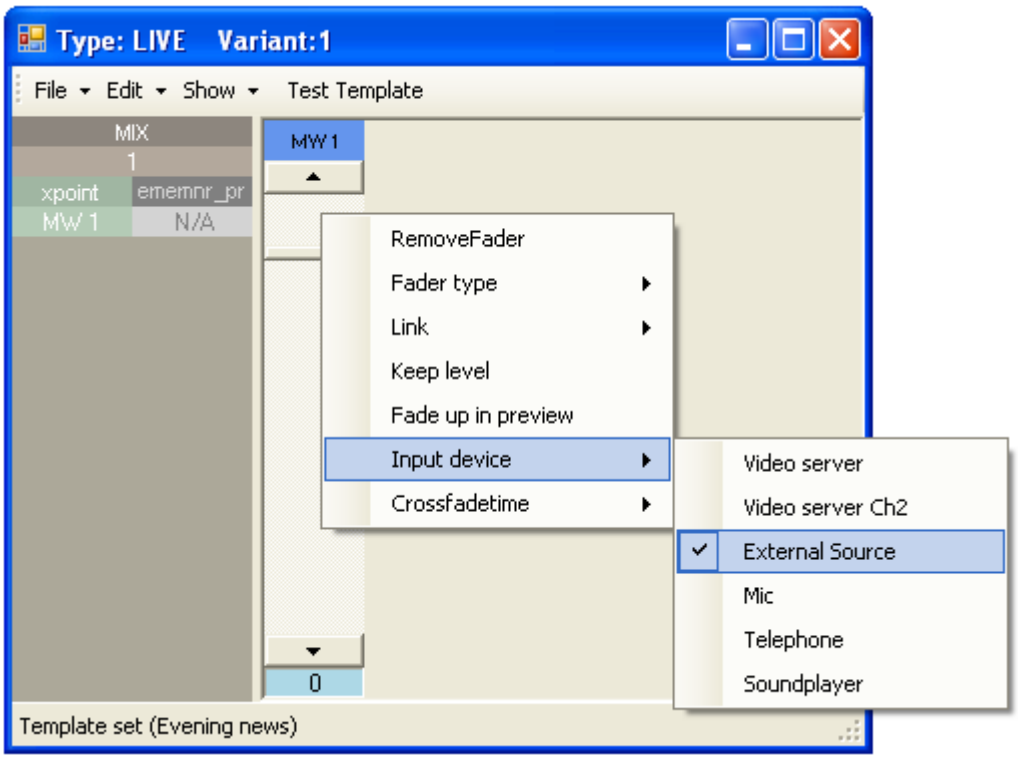


A second sound level can be set for each fader by using the **SHIFT** key over the fader to set the second level. This second sound level for all faders is used as the start level for the voice over sound bite type. The same special variant logic for video clip with full sound applies for the voice-over type.

8.2.4 Live External Source

Live external source variants are typically defined by their corresponding vision and audio mixer inputs. In the example below, we define variant 1 as source MW 1 on both PP bus cross point and audio fader.

Copyright © 2024 Vizrt



Additional variants 2 to 9 would typically use MW 2 to MW 9 as sources for both video and audio.

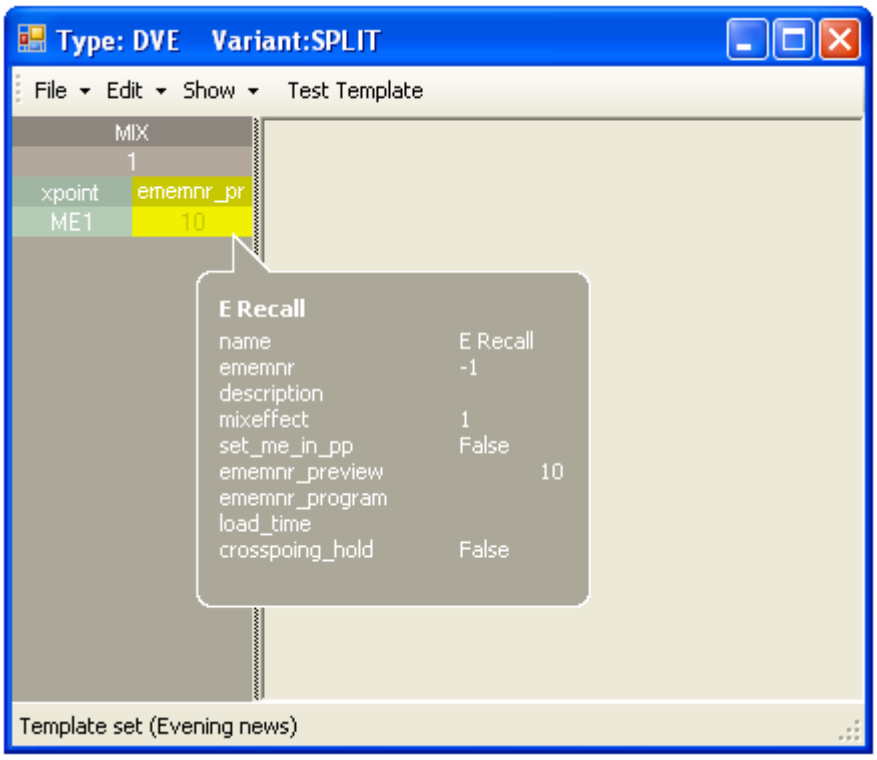
8.2.5 DVE

The example below describes four setup steps necessary to create a template that recalls a predefined E-mem (emem number *10*) defining a split screen DVE on M/E 1 on the vision mixer.

1. E-mem Recall Definition

The vision mixer effect uses key 1 and key 2 on M/E 1 as left and right split windows.

Copyright © 2024 Vizrt

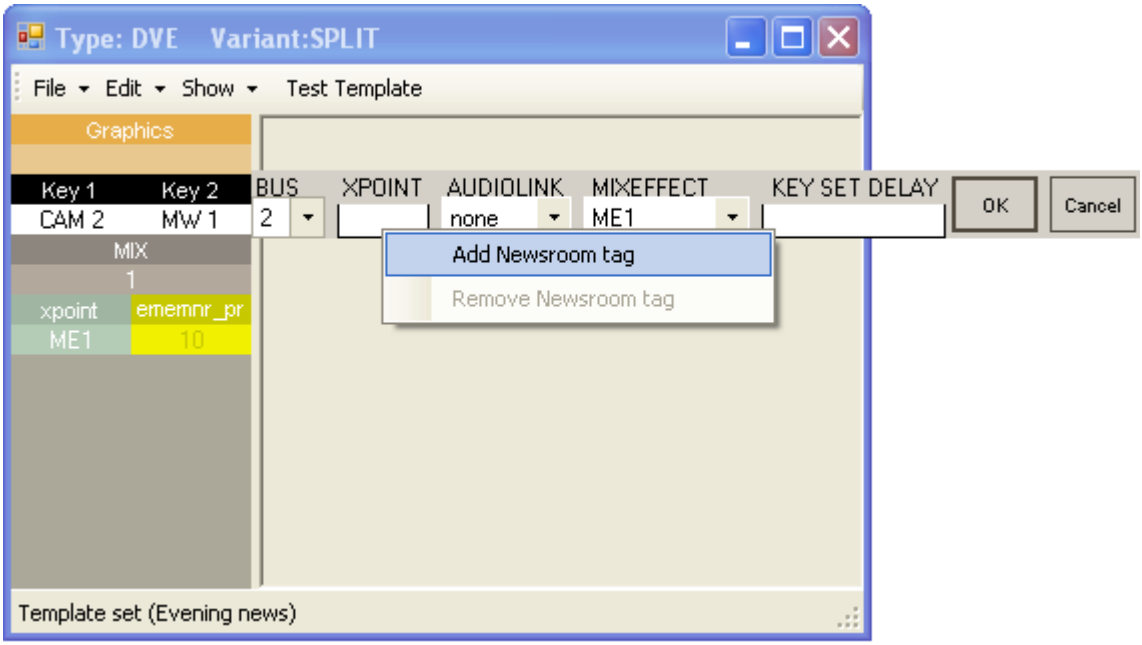


The PP bus cross point is set to ME1 (program output of mixer effect). The E-mem recall control is defined to recall emem 10 when the template is taken to preview.

2. Key bus Delegation Setup

Default crosspoints are chosen (*CAM 2* and *MW 1*) and since a newsroom tag has been added for both keys, the crosspoints can be set from the newsroom system.
Right click over the XPOINT window to add a newsroom tag. In addition, the template defines a recall of a graphics background

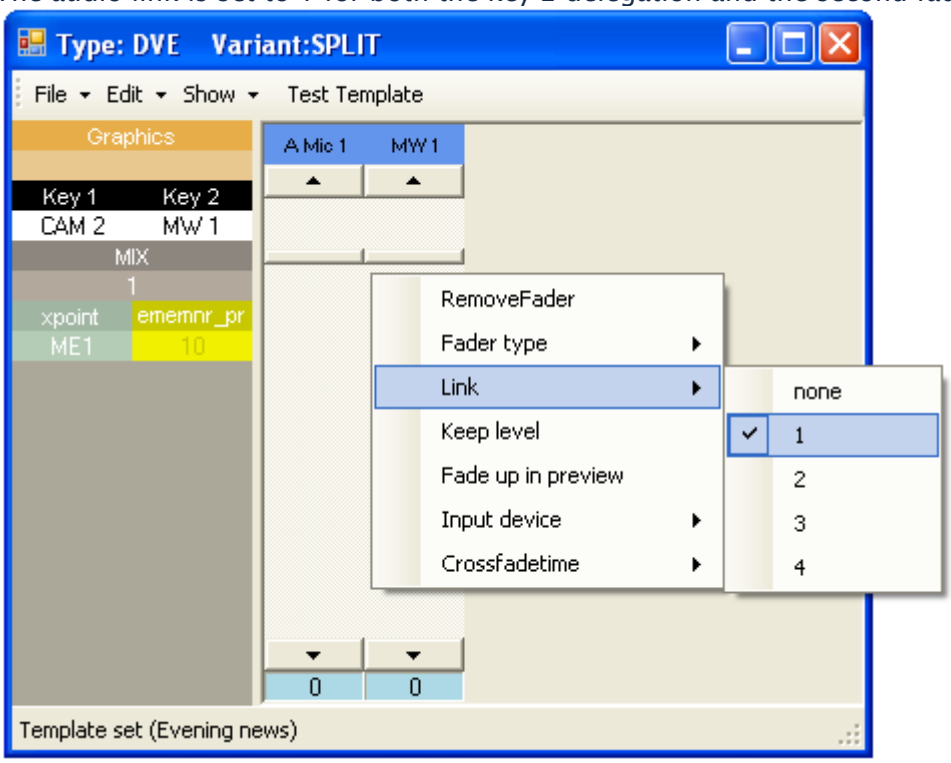
Copyright © 2024 Vizrt



The second key bus delegation is shown. As a default we assign CAM 2 to key bus 1 and MW 1 to key bus 2 on the ME1 mixer effect bus.
(This assignment can be overridden from the NRCS story element by using the newsroom tag).

3. Audio Linking

The audio link is set to *1* for both the key 2 delegation and the second fader (MW 1).

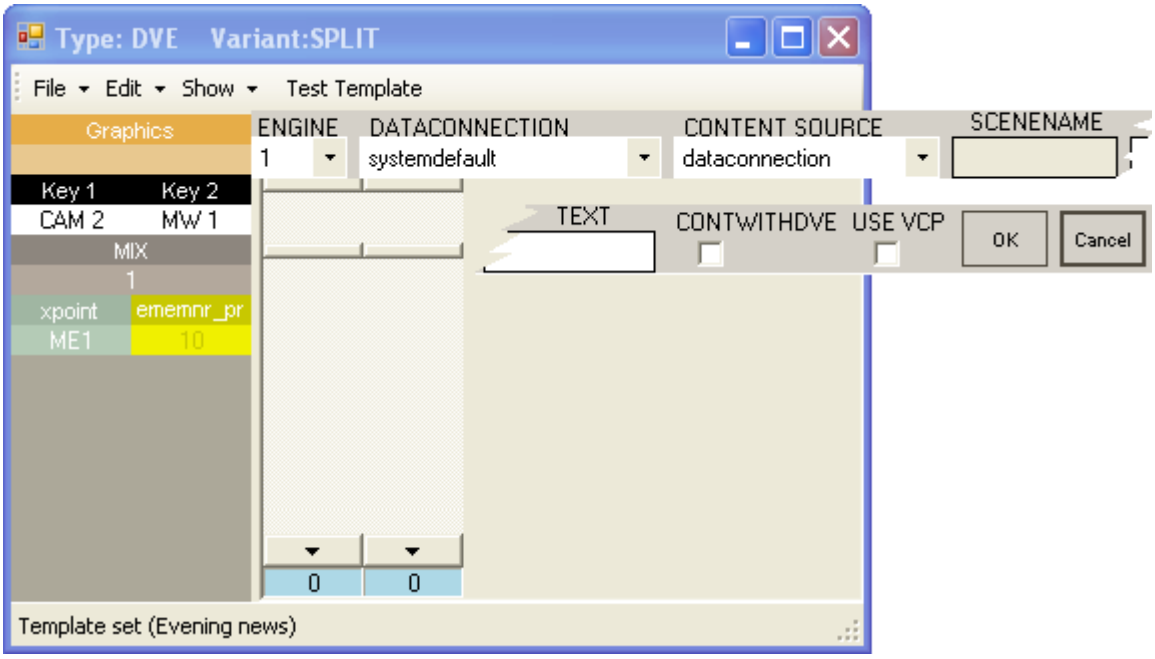


When using the newsroom tag to set key 2 to a different source (for example *MW 3*), both video

Copyright © 2024 Vizrt

and audio delegation will follow:
This leads to the key 2 source being set to *MW 3* and the second audio fader being set to *MW 3*.

4. Graphics Engine Data Element Setup



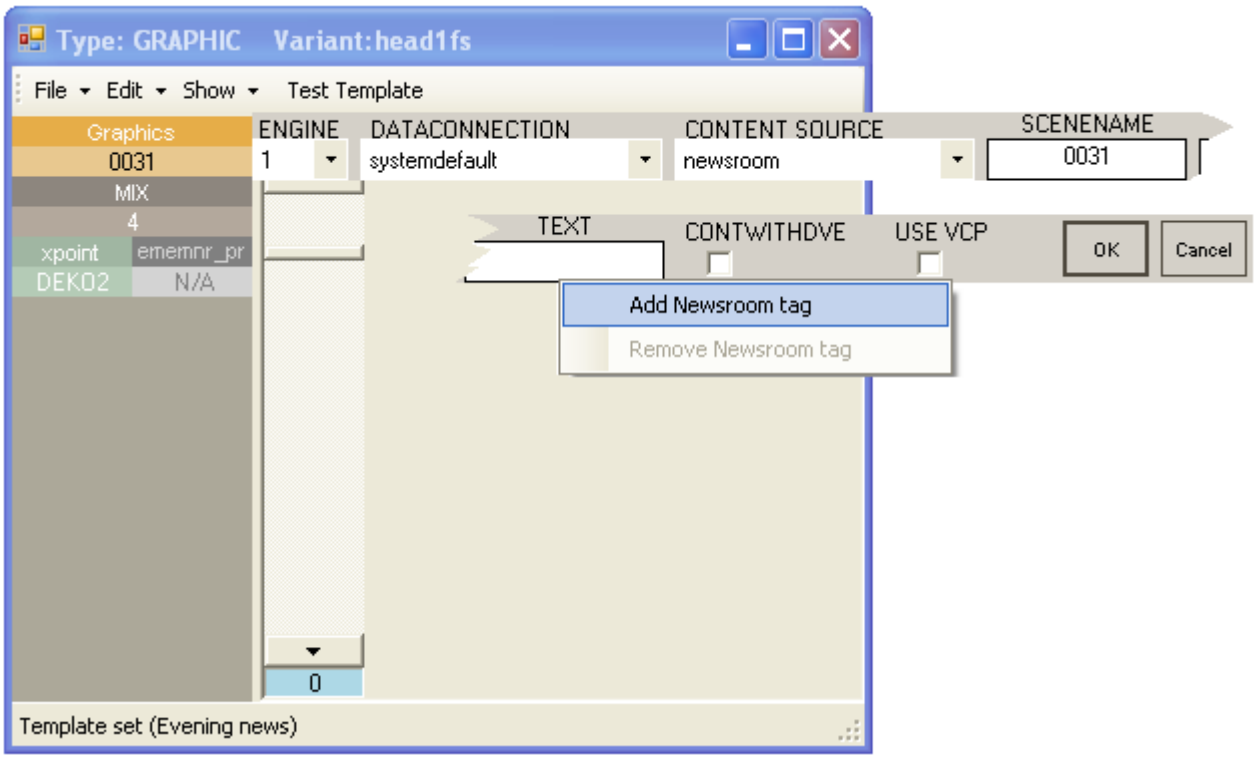
Setup of the *graphics controller* is described in the next section.

8.2.6 Graphic: Full Screen Graphics

The graphics controller in the **Template Editor** is vendor-specific, due to the lack of a generic control API. As a general feature, it can control multiple graphic engines. A typical controlled cluster could for example be, one for full screen graphics, one for studio wall graphics and additionally, an overlay CG engine. For DVE effects, this can be a fixed element, while for telephone graphics or maps the element name and data are provided in the NCS story.

The controller's task is to cue graphics when taken to preview and run them when taken to program.

Copyright © 2024 Vizrt



8.2.7 Accessory Templates

In this example you create an accessory template for driving content to a video wall driven by the Template Router feature in Viz Mosart. this feature merges new template-based instructions onto an existing one, enabling, as in this example, an asset to be diverted to the video wall. You need a separate accessory template for each wall that will be controlled. The accessory template can contain settings for cross points, video server port, graphics engine and aux.

Creating the Wall Accessory Template

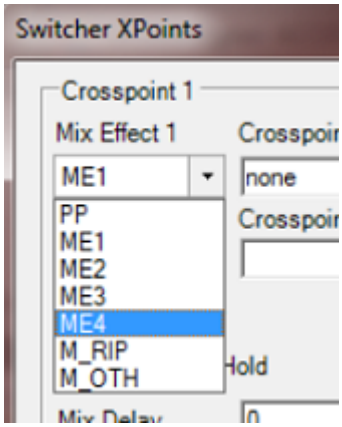
The wall accessory template is created as a standard template.

- You can create a new accessory template with a variant name of for instance "wall2".

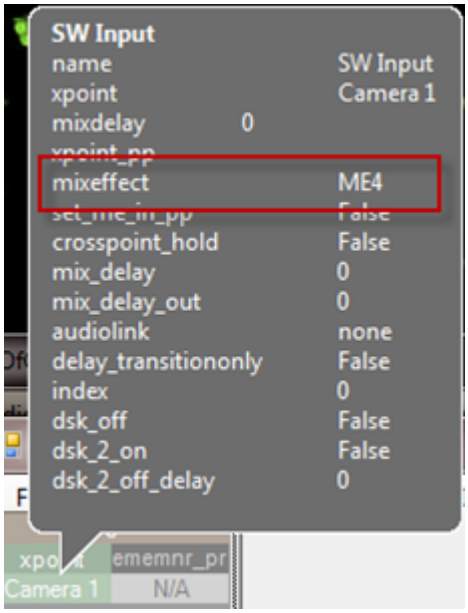
Switcher Cross Point

- To change switcher crosspoints enable **Switcher crosspoint** in the Template Editor.
  The Switcher XPoints appear.
  Here you can set the ME-step you want for the wall taken template.
- When the template is set to the wall item it will use this ME step instead of the one specified in the template.

Copyright © 2024 Vizrt



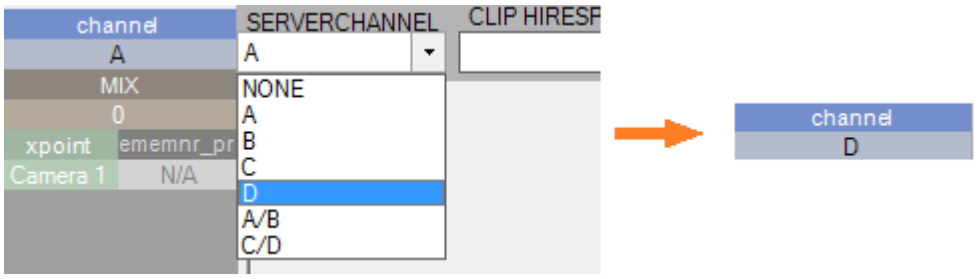
The ME-step will show as a mix effect in the Template Editor.



Video Server Port

If a video clip is wall taken; another video server should be used.

- This can be changed in the accessory.



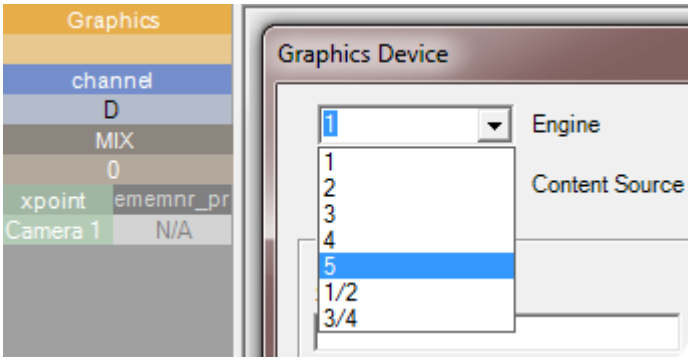
Make sure the video clips are available through this port.

AUX

- The AUX is controlled by the template but should be added to the accessory template. The setting will be replaced by the wall taken template.

Graphics

- To avoid conflict with other graphic elements, you may want the wall to run on its own engine. This can be changed in the accessory.
  A wall taken graphic element will then use this engine.

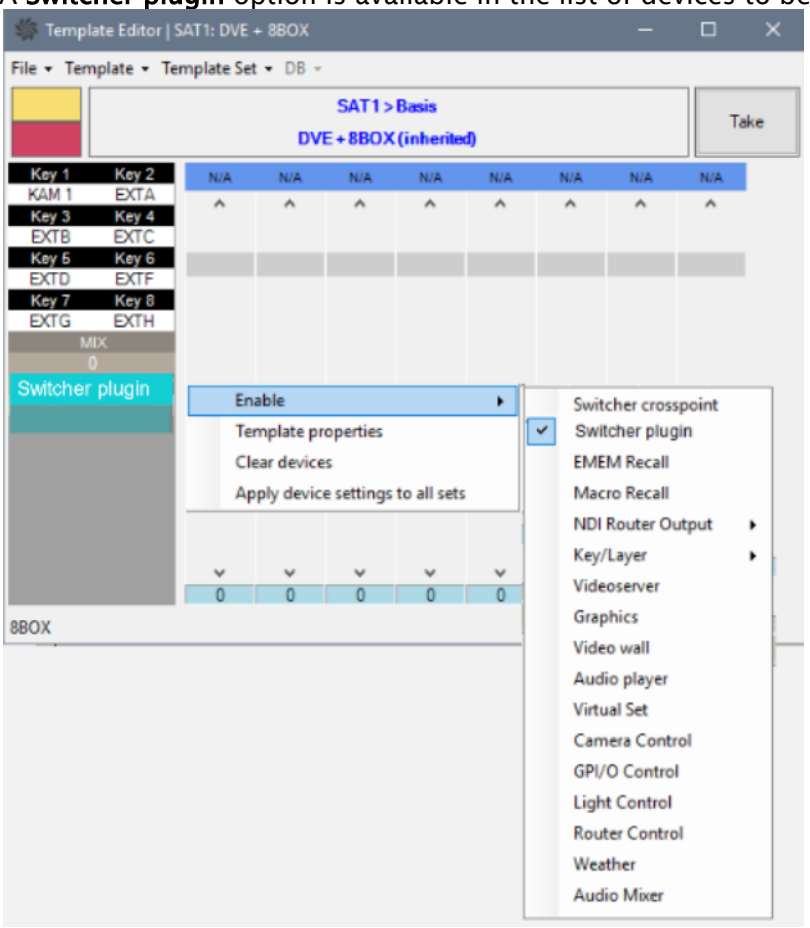


8.2.8 Switcher Plugin

These are advanced configurations to a standard switcher where functionality is extended. The example in this section refers to extending *Viz Engine* functionality.

Copyright © 2024 Vizrt

A **Switcher plugin** option is available in the list of devices to be enabled on the template.



The **Switcher plugin** view introduces two new configurable features, Template Examples and Template Examples.

Static values can be specified directly on the template. Newsroom tags can also be configured, enabling control from within the NRCS.

Top Scenes

Note: To work with top scenes, you must first perform the required setup, as described in the *Configuration* section of Viz Engine as a Switcher.

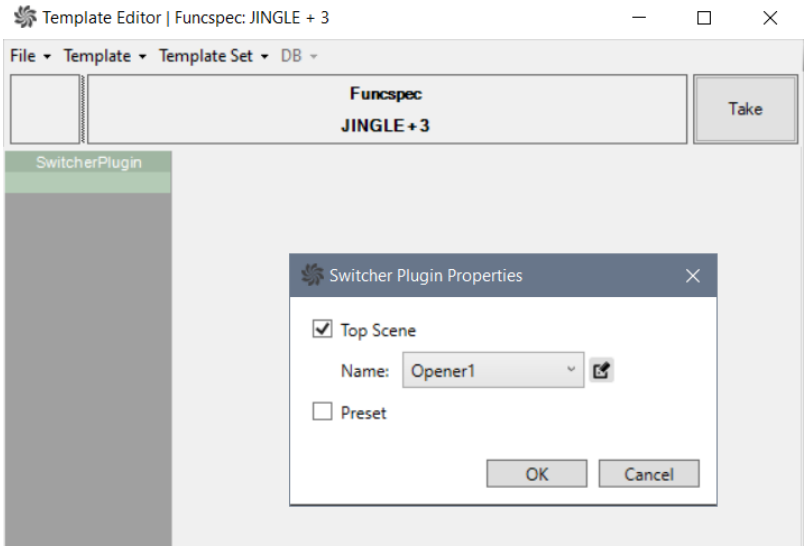
A typical use of top scenes is as an *opener*. For example, the top scene is played to introduce the 8 o'clock news show with many visual effects that fade to the following camera scene, and a news host taking over.

Tip: For an in-depth explanation of Top Scenes and their time-related configurations, refer to the section Viz Engine as a Switcher (Top Scenes).

Copyright © 2024 Vizrt

To create a template that plays a top scene

1. Create (typically) a Jingle template.
2. Enable **Switcher plugin** for the device.
3. Specify the top scene name from the list with the following options:
    a. None (empty).
    b. A list of top scene pre-configured in the A/V Setup.
    Select the required top scene (this is the name of the scene in the associated Graphics Hub folder).

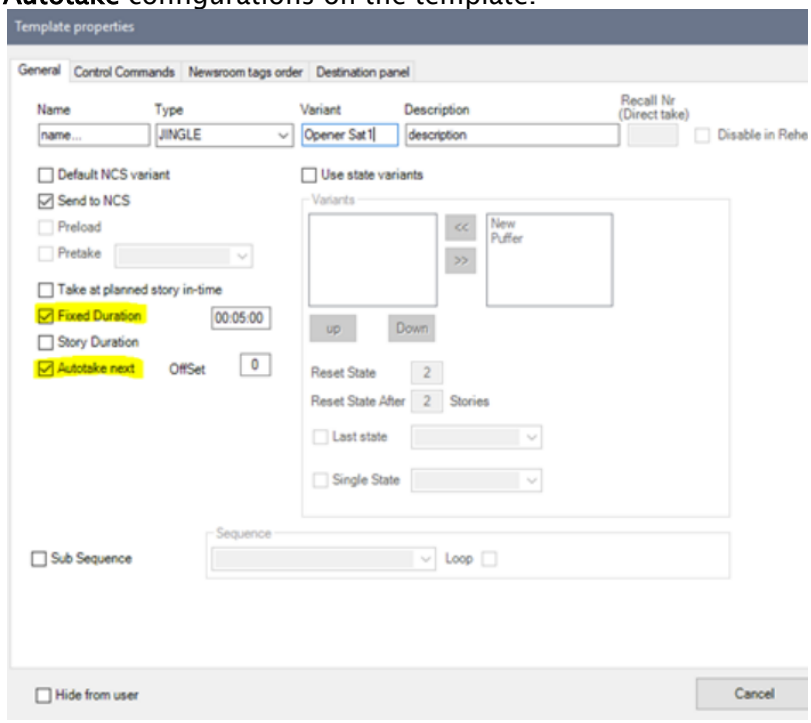


There are two time-related aspects when creating new templates for top scenes:

1. The time it takes for the scene to load. This is handled automatically by Viz Mosart when cueing the story item.

Copyright © 2024 Vizrt

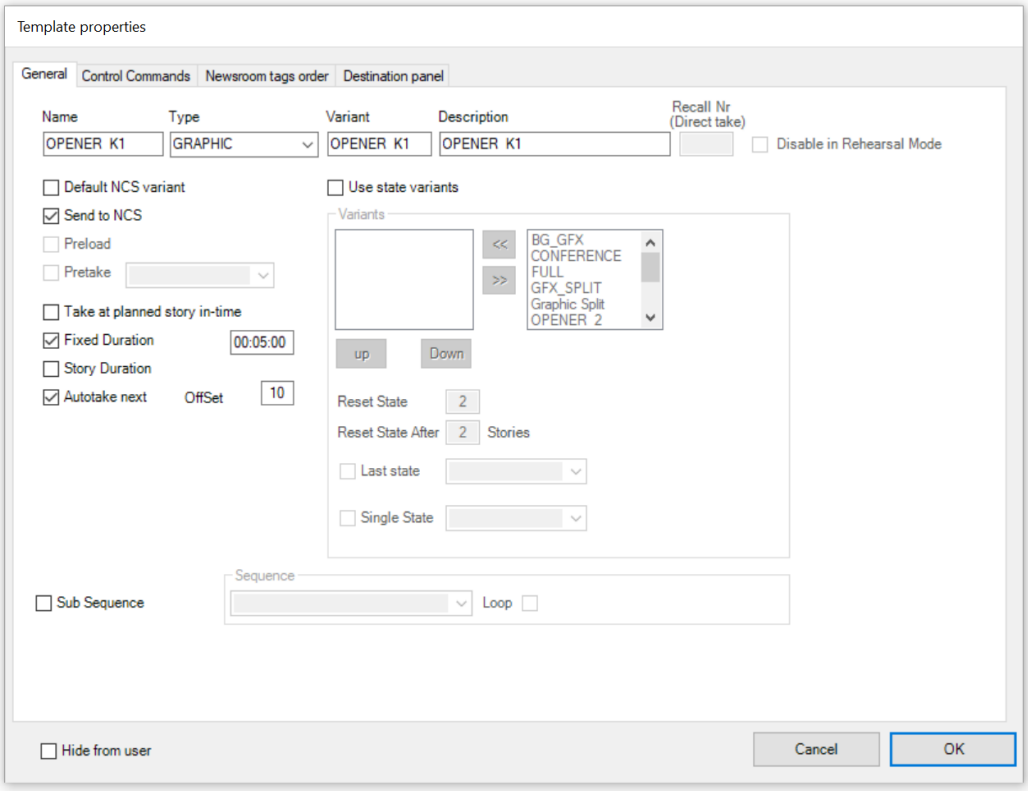
2. The time it takes for the scene to run. This is controlled with the **Fixed Duration** and **Autotake** configurations on the template.



Copyright © 2024 Vizrt

Autotake

- You can add an *Auto Take* to the template to jump to next story item automatically (switch underneath the Top Scene)

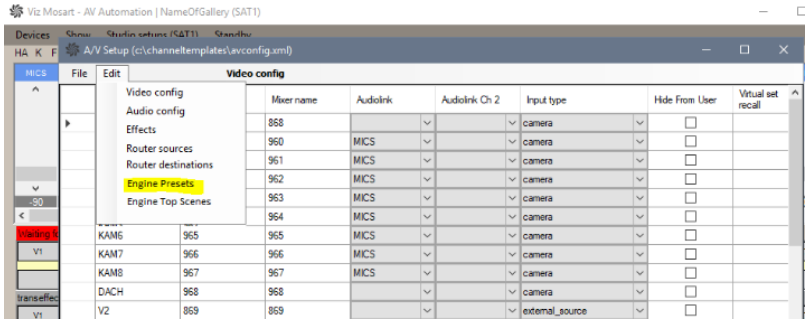


Presets

Select *Engine Presets* from **Edit** tab.

Use one of the following options:

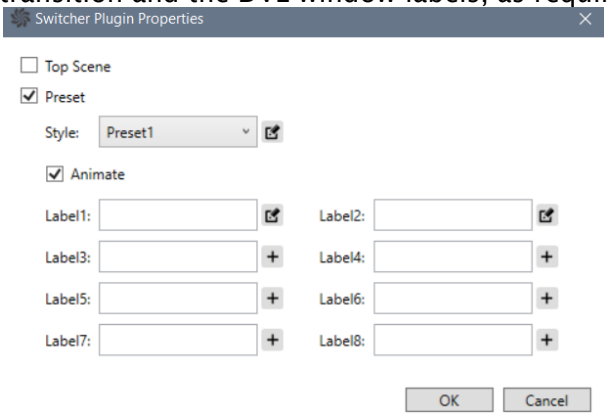
- None (empty).
- A list of preset names pre-configured in A/V Setup (AvConfig.xml) file.



Transition Style

The transition style (hard-cut or animation), as well as several label values can be specified.

For example, on the DVE template above, where **Switcher plugin** for the device is available, the user can select the layout and looks of the DVE effect by specifying a preset style, the preset transition and the DVE window labels, as required:

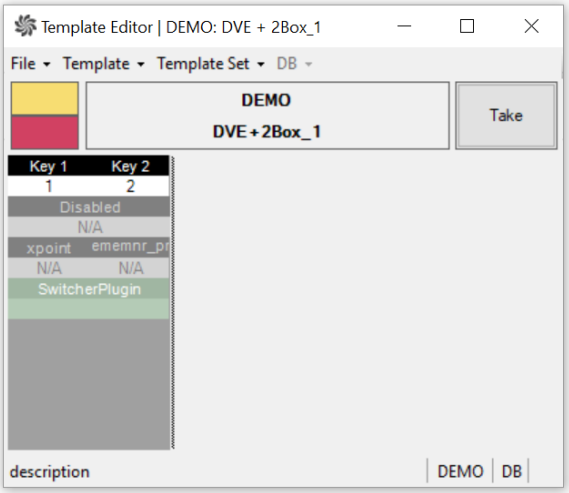


Switcher Plugin Additional Features

DVE Boxes (in a Preset)

Simple 2-Box For Live Sources

- Keyfill 1
- Keyfill 2
- *No* XPOINT !
- Switcherplugin

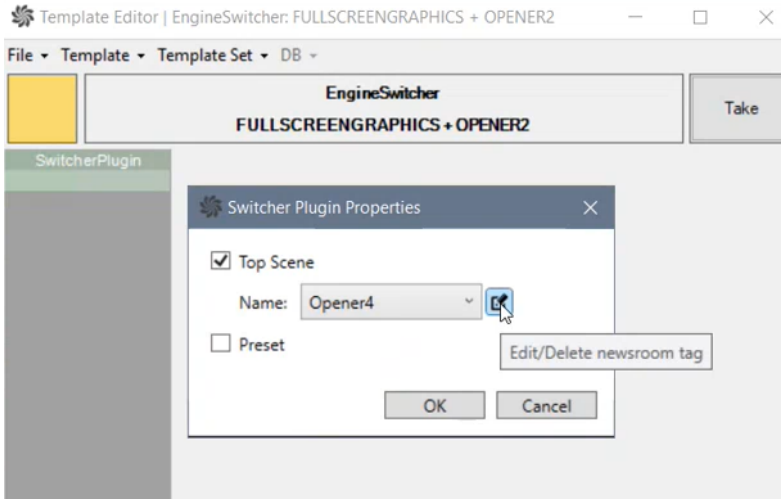


Switcher Plugin Settings

- Select the required Top Scene.

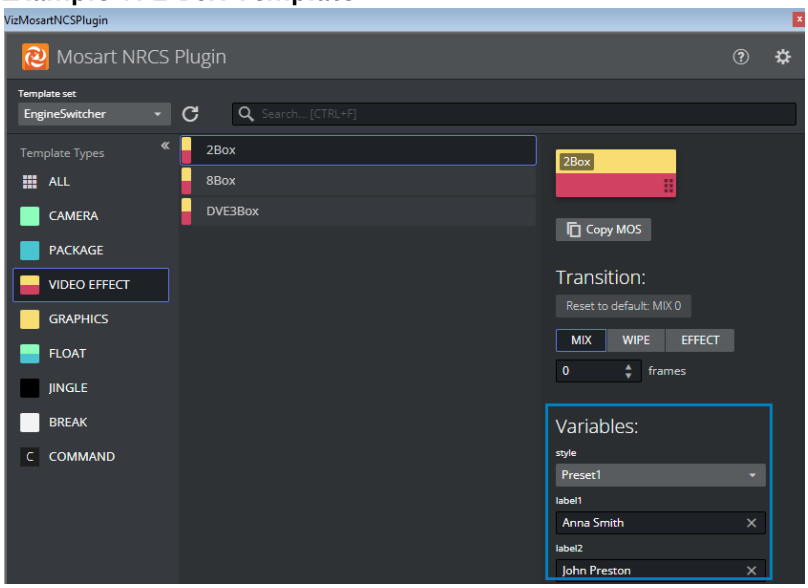
Copyright © 2024 Vizrt

- Fill in any required labels.



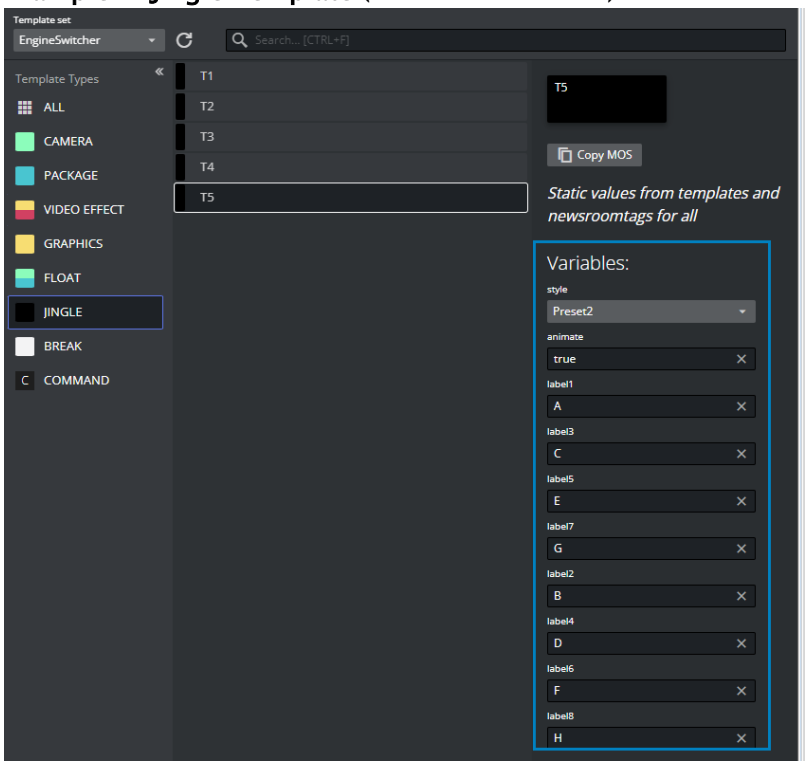
- Labels can be linked to an NRCS with a newsroom tag.
  Clicking the **Edit/Delete newsroom tag** icon, opens a mappings menu:

Example 1: 2-Box Template



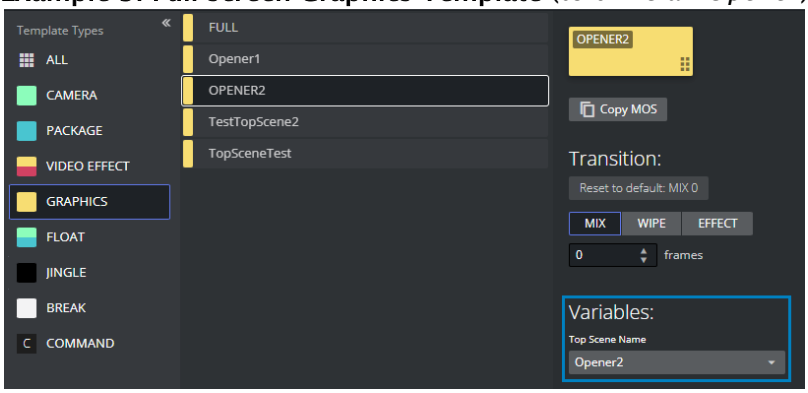
Copyright © 2024 Vizrt

Example 2: Jingle Template (to control a *Preset*)



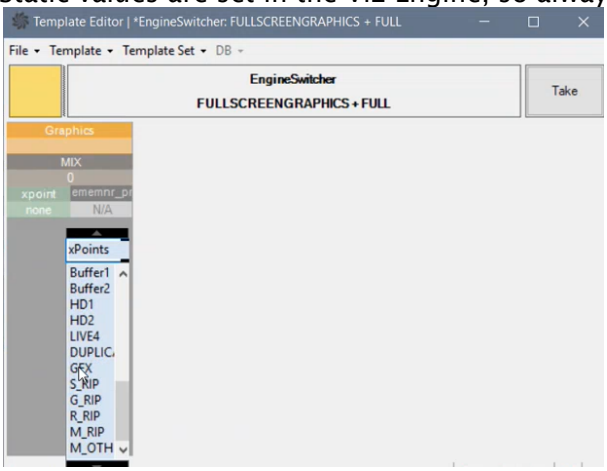
> **Note**: To use the *animate* newsroom tag, set the **animate** dropdown option to *true*.

Example 3: Full screen Graphics Template (to drive an *Opener*)



Graphics

Static values are set in the Viz Engine, so always choose GFX for the switcher crosspoint.



See also: Viz Engine as a Switcher, section Template Design.

8.3 Working With Templates

This section contains the following topics:

- Template Operations
- Template Properties
- Control Commands in Templates
- Destination Panel
- Mosart Templates Feedback to the NRCS
- Controlling a REST-based Driver

See also: Linking Device Properties and Newsroom Tags.

Copyright © 2024 Vizrt

8.3.1 Template Operations

To Add a Template

You can add a new template to the currently selected template set.

1. Select **File > New template**.
2. Enter the new Template Properties.

To Edit a Template

- Select **Edit > Template properties**.

> **Note:** Template properties can also be accessed by right-clicking in the device's Function area. See the table Template Properties.

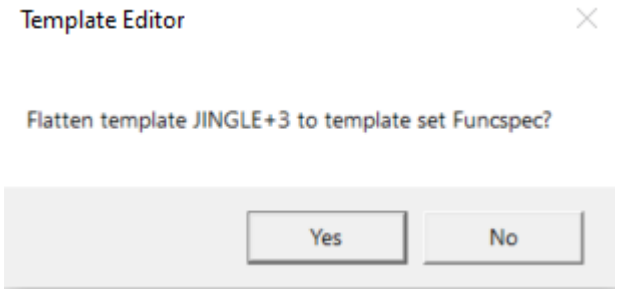
To Remove a Template

1. Select **File > Remove template**.
2. Confirm delete in the dialog box.

To Flatten a Template

You can push a new template, created in another template set (that was created using the hierarchical approach), back to the originating template set.

1. Select **Template > Flatten**.
   A dialog box displays which template set the selected template will be flattened to.



2. Click **Yes**.

> **Note:** For a description of Hierarchical templates, please refer to Working with Template Sets.

To Copy a Template

You can copy a template from one template set to another template set.

1. Open the template you want to copy.
2. Select **Edit > Copy template**.

Copyright © 2024 Vizrt

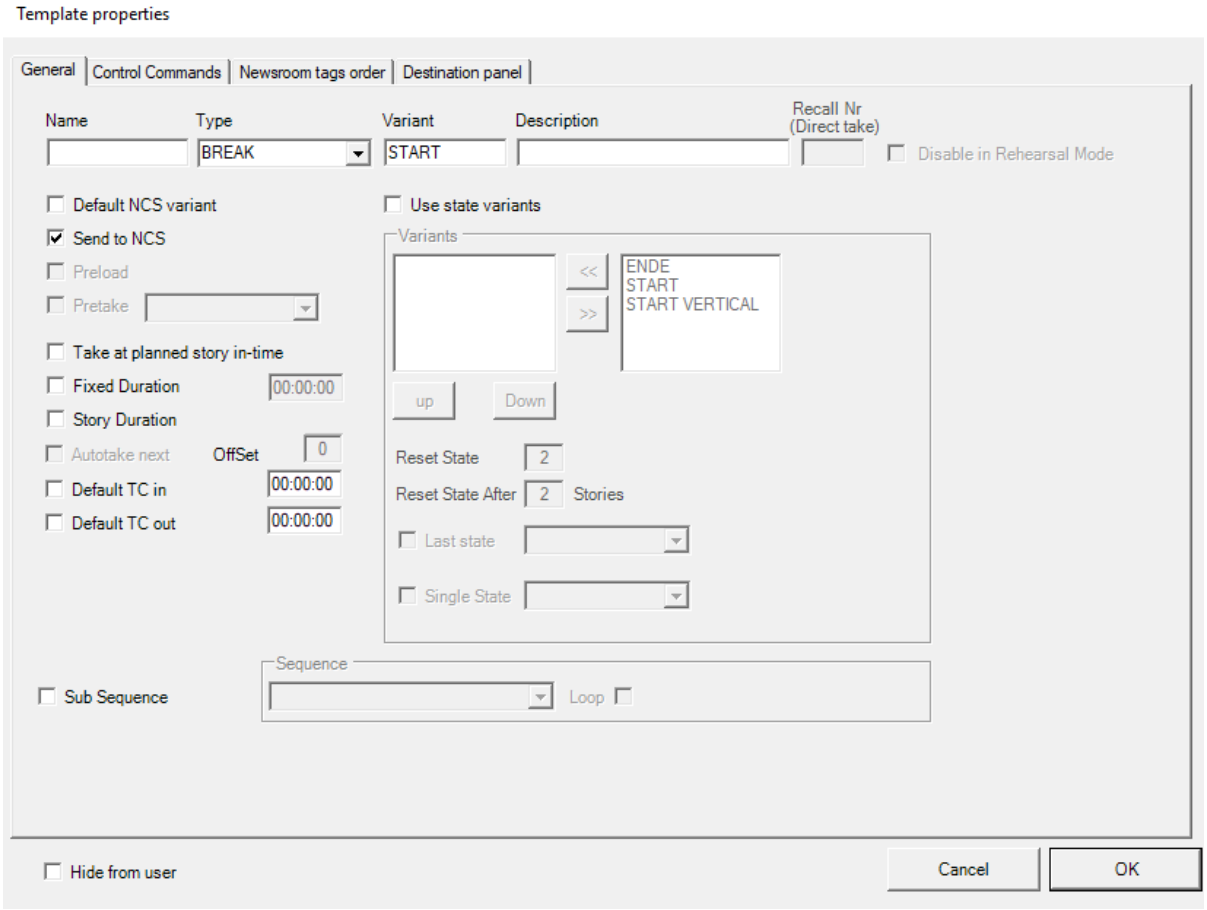
3. Switch to the other template set, and then apply **Edit > Paste template**.
4. Select **Edit > Paste to all** to copy the template to all template sets.

Copyright © 2024 Vizrt

8.3.2 Template Properties

This section explains the configurable fields on the **General** tab of the **Template Properties** menu, and also clarifies the function of a *Sequence*.

- Configuration Descriptions
- AutoTake Timings
  - Autotake next Templates
  - Duration
- Sequences
  - Recalling a Sequence from Template
  - Recall a Sequence from a Direct Take Template
  - Controlling a Sequence



Configuration Descriptions

Item	Description
Name	The name of the template. This value will only show in log files. English language is preferred, for making it easier to understand log-files.
Type	The different primary and secondary templates types are described in the *Viz Mosart User Guide,* section *Viz Mosart User Interface* under *Rundown Window*.
Variant	The variant is a unique name for the selected type, and should match the value entered in the newsroom system.
Description	Description of the template (optional). English language is recommended, for future maintenance purposes.
Recall Nr (Direct take)	(Applies only for templates in the Directtakes template set). One or more decimal digits. If this direct take template is to be recalled from the GUI using the numeric keypad, a maximum of four digits is allowed. (If more than four digits are used, the template may still be executed using e.g. a control command.) The number must be *unique* among all direct take templates. **Note:** Avoid starting numbers with zero (0). For example, use 56 instead of 0056.
Disable in Rehearsal mode	(Applies only for templates in the Directtakes template set). The direct take will not be executed when in Rehearsal Mode
Default TC in	
Default TC out	
Default NCS variant	This sets the template as the default variant for the selected type.
Send to NCS	When selected, allow this template to show in the NRCS ActiveX.
Preload	When selected, enable the preview functionality for the Accessory template. The Accessory template must have time code 00:00 for this to function. *Only valid for Accessory templates.*

Copyright © 2024 Vizrt

Item	Description
Take at planned story in-time	If you want a story to start at a specific time of day, you can add a template with this option checked, to a story that has a set "hit time" in the NRCS. The "hit time" must be included in the story data sent to Viz Mosart, and the field-name must be mapped to **back_time** in Newsroomsettings. See Story External Metadata.
Fixed Duration	Make the template always have the same length (overriding the time from the NCS) in the Viz Mosart rundown. Select this option and enter the time in mm:ss:ff (minutes:seconds:frames). This can also be combined with Autotake to make a frame accurate continue to the next story. Remember to write the whole number each time, you cannot edit a single character at a time.
Story duration	When selected, the duration of the template will be set equal to "story editorial duration", coming from the NRCS. The "story editorial duration" must be included in the story data sent to Viz Mosart, and the field-name must be mapped to **story_duration** in Newsroomsettings. See Story External Metadata. If more templates are added to the story, their duration will be in addition to the "Story duration" template, so calculated total duration of templates will be greater than "story editorial duration".
Autotake next	(Available for *primary* templates only, cf. **Type** above.) When selected, automatically perform a **Take Next** (**F12**) at the end of the template's duration, with an offset as described in **Offset** below
Offset	Set the offset when Autotake is enabled to adjust the Viz Mosart continue action at the end of the event. The value is in frames. A positive value will make the continue action happen later. A negative value will make it happen earlier.
Sub Sequence	See separate explanation below this table.
Sequence	See separate explanation below this table.
Loop	See separate explanation below this table.
Hide from user	When selected, this template is removed from the template list in the **Quick Editor**.
Use state variants	Please refer to Other Template Functionality for explanation of state variants.

Copyright © 2024 Vizrt

Item	Description
Control Commands (tab)	If the template should have *control commands* attached, configure here. Please refer to Control Commands in Templates for further instructions.

AutoTake Timings

An item has a *duration*. The method of determining this duration depends on the template type. For example, for a *Package*, it is the duration of the *video clip* that determines duration.

Duration can also be set from the NCS.

Autotake next Templates

An **Autotake next** template enforces that the *next* item will be taken automatically when *this* item is 'finished'.

For most Autotake next items, the *next* item will be taken when the *duration*, as defined below, has elapsed. An exception is an item with *clips, where Mix Delay* will also have to elapse before the next item is taken

Duration

- For an item based on an **Autotake next** template, the Offset (*positive, 0* , or *negative*) is *added* to the duration.
- If the underlying template has an *effect* transition, the *duration* of the effect (as defined in the Effects table in A/V Setup) is *added* to the duration.

Sequences

A sequence is a saved story (including primary and secondary items with relative timing information).

Sequences are created in Viz Mosart GUI.
This is described in the *Viz Mosart User Guide*, under Operation, in section Creating Sequences.

Recalling a Sequence from Template

To Recall A Sequence From A Template

1. Select **Sub Sequence.**
2. Select the Sequence to recall. (The available options are the sequence names saved to the template set).
   For a direct take template, *all* sequence names, regardless of template set, are eligible, see Recall a sequence from a direct take template.
3. If required, select **Loop**.

Copyright © 2024 Vizrt

When a sequence is recalled from a template, actually only the *secondary* items of the *first* primary item are recalled (but still with the same timing information that it had in the saved story sequence).

Recall a Sequence from a Direct Take Template

To Recall A Sequence From A Direct Take Template

- Sequences may be recalled from direct take templates just as from other templates. However, when taking a direct take template, a sequence will be recalled only if the *current* template set (*at that time*) has a sequence with the same name chosen. (And then that sequence will be recalled.)
- Any take out logic might not work as expected if the direct take is not fired at the beginning of the current main template.
- When using direct takes with sequences it is recommended to recall them *from a primary template* and *not* use them with keyboard shortcuts.

Controlling a Sequence

When a sequence has been started it can be controlled using keyboard shortcuts with Control Command - SEQUENCE.

See table Command Values and Parameters , COMMAND SEQUENCE, Value LOOP, STOP_LOOP, STOP, START.

Copyright © 2024 Vizrt

8.3.3 Control Commands in Templates

A Mosart *control command* can be triggered from a template, executed when the template is taken (Command On Take), at continue points (Continue Points) or when the template is taken out (Command On Take Out).

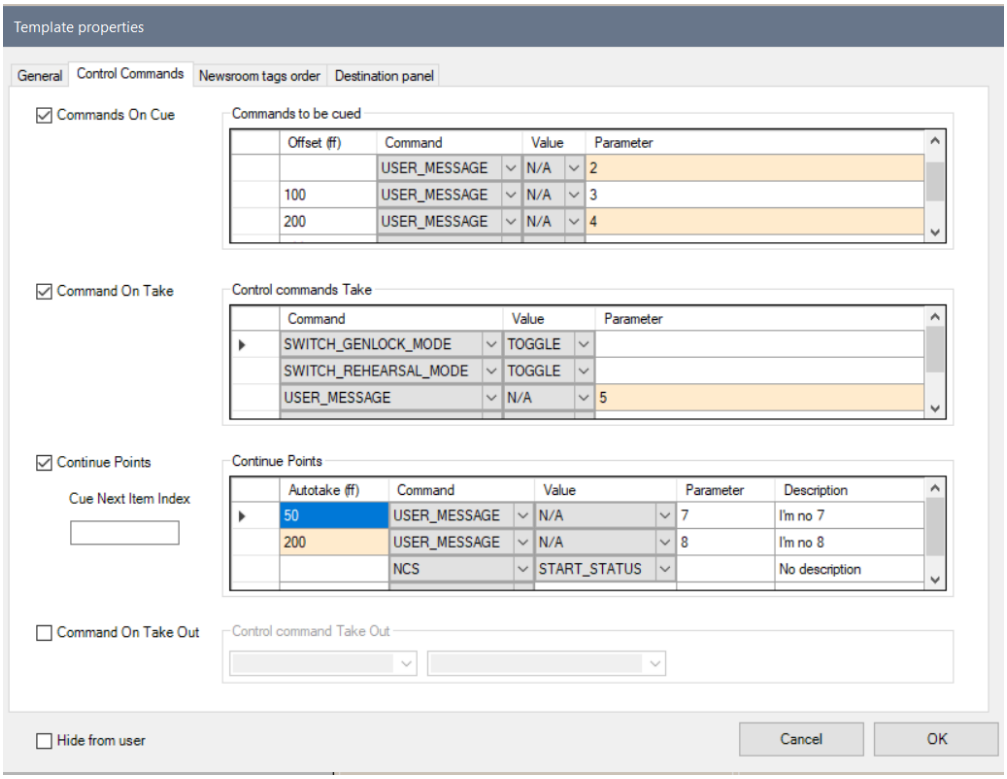
- Working with Control Commands
  - Adding a Control Command
  - Commands on Cue
  - Command On Take
  - Continue Points
  - Command on Take Out
- Command Values and Parameters
  - Parameters with Placeholders
- VIDEO_PORT Control Commands in AV Automation Template Properties
- RECORD Control Commands in AV Automation Template Properties

Note:
- A Mosart template can trigger *multiple* control commands.
- A (not-templated) control command can trigger a Mosart template, often used for a *DirectTake* template.

Copyright © 2024 Vizrt

Working with Control Commands

This is done through **AV Automation > Devices > Template Editor > Template > Properties > Control Commands** tab.



The figure above shows a *single* (video) control command to be executed when the template is taken. Multiple control commands can also be assigned here.

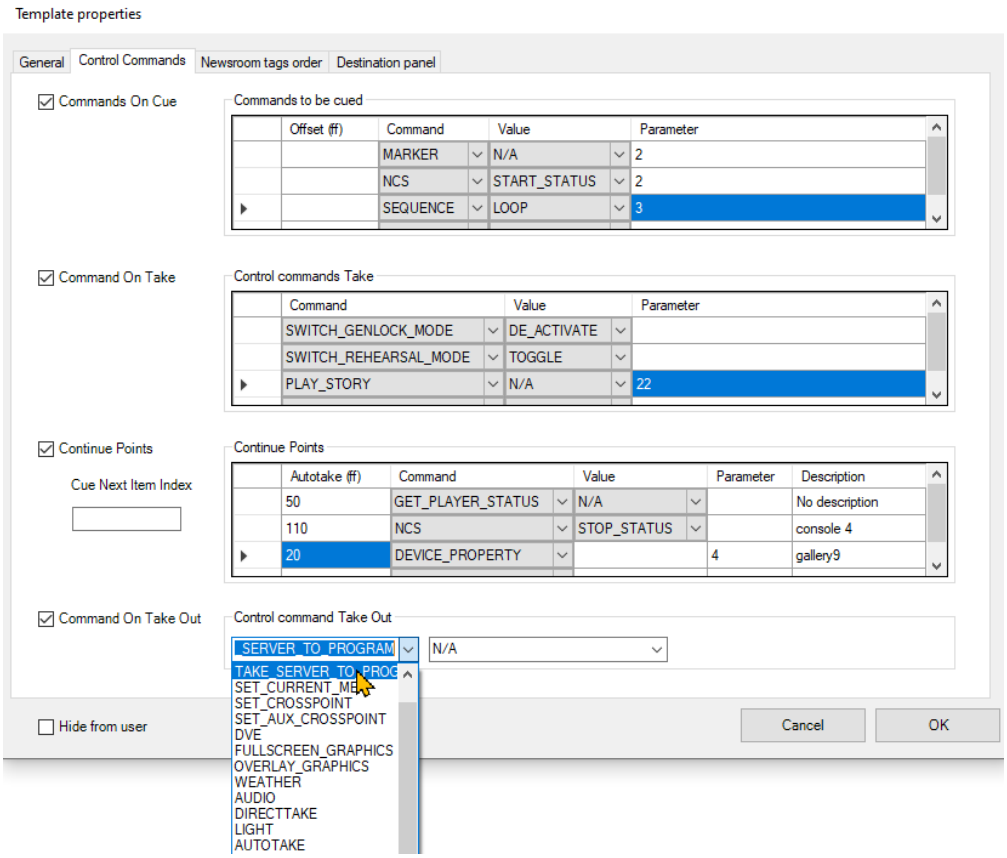
It is also possible to assign control commands to be executed at continue points and when the template is taken out.

The latter has limitation since the *Parameter* setting is not available. In this context it may be useful to trigger a direct take at this place.

Adding a Control Command

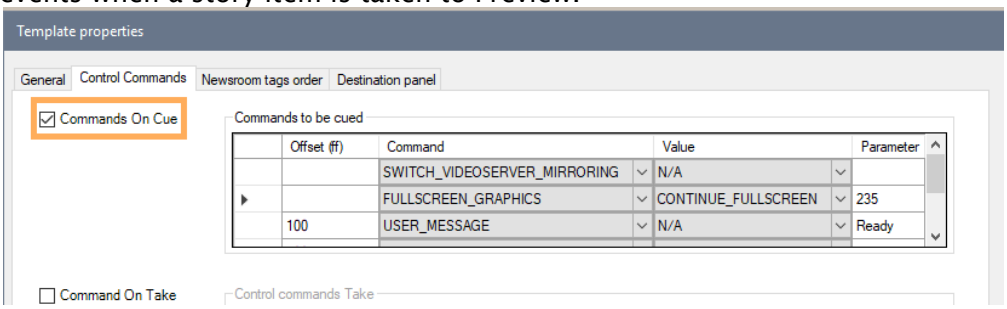
To Add A Control Command

- Click the **Control Commands** tab in the **Template properties** window.
  The three categories of control command are Commands on Cue, Command On Take, Continue Points and Command on Take Out.
- Refer to the table Command Values and Parameters for command details.



Commands on Cue

You can recall control commands in Preview, when the template is cued. This enables triggering of events when a story item is taken to Preview.



As illustrated above, you can give the command a *Value* and a *Parameter* in the same way as for Command On Take, Command on Take Out.
Also, in a similar way to Continue Points you can set an offset for when the command is to be executed.

> **Note**: For *Cue* commands, all offsets are relative to the *start of cueing*. This differs from *Continue Points*, where each offset is relative to the *previous* continue point.
> In the illustration above, the User Message (*Ready*) will be sent 100 frames after cueing.

Copyright © 2024 Vizrt

By defining a *Newsroom* tag as a parameter for an *On Cue* command, the user can set the parameter from the NRCS.

- In the *Parameter* column right-click and select either Add/Remove/Change a newsroom tag.

Command On Take

When selected, the commands inserted in **Control commands Take** are executed when the template is taken to Program or On Air.

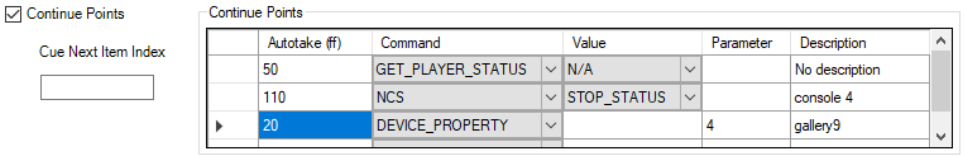
By defining a *Newsroom* tag as a parameter for an *On Take* command, the user can set the parameter from the NRCS.

- In the *Parameter* column, right-click and select either Add/Remove/Change a newsroom tag.

> **Tip**: This selection then appears in the Mosart Web App NRCS Plugin (or the legacy ActiveX Plugin) for adding the template to a rundown.

Continue Points

When selected, the commands inserted in **Continue Points** will be taken when pressing TAKE NEXT (default: **F12**). Viz Mosart executes this continue point from within the primary template in which it is inserted. Each line represents a new Continue Point.



- **Cue Next Item Index**: Enter the line number (for example, *0, 1, 2, 3*) of the continue point that executes a cue for the next/following Primary object. If empty, Viz Mosart cues when entering the last continue point in the list.
- **Autotake**: Insert the autotake time, in frames. The command is then automatically taken after that time.

> **Note**: For *Continue Points*, all offsets are relative to the last Continue Point. This differs from Commands on Cue, where each offset is relative to the *start of cueing*. In the illustration above, the Device Property is fetched 20 frames after being taken on-air, Player Status is fetched at 50 frames later and from the NRCS, the Stop Status at 110 frames after the previous (Get Player Status) Continue Point was processed.

By defining a *Newsroom* tag as a parameter for a *Continue Point*, the user can set the parameter from the NRCS.

- In the *Autotake* column right-click and select either Add/Remove/Change a newsroom tag.

> **Tip**: This selection then appears in the Mosart Web App NRCS Plugin (or the legacy ActiveX Plugin) for adding the template to a rundown.

Copyright © 2024 Vizrt

Command on Take Out

The commands inserted here are executed when the template is taken from the program, taken Off Air, or at the end of the template. Only a single command can be used in **Command On Take Out**, and it does accept any parameters.

Command Values and Parameters

> **Note:** Parameters marked with (*) do not work for templates.

> **Note:** It is not possible to enter a *Parameter* in the **Control command Take Out** section.

COMMAND	Value	Parameter	Description
ACCESSORIES	TAKE_NEXT	(*)	
AUDIO	FADE_MANUAL		Toggle fade manual.
	FADE_OUT_KEEPS	Fader rate	Faders that are set as "keep level" in the template On Air are faded out. Will also work when new templates have been taken in between the 'keep' action and the FADE_OUT_KEEPS.
	FADE_DOWN SECONDARY_AUDIO	Fader rate	Viz Mosart fades down relevant audio sources not connected to the video currently On Air.
	FADE_UP SECONDARY_AUDIO	Fader rate	Viz Mosart fades up relevant audio sources not connected to the video currently On Air, when cutting between sources.

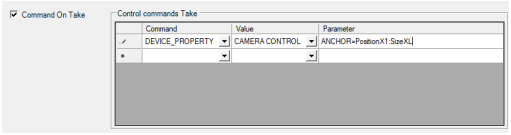
Copyright © 2024 Vizrt

COMMAND	Value	Parameter	Description
	SET_LEVEL_2_ONAIR	TRUE FALSE <empty>	These settings apply to fader second levels in the template that is *On Air*. TRUE: The second level for faders are set to *ON*. FALSE: The second level for faders are set to *OFF*. <empty>: (no value entered) The second level for faders are set to *TOGGLE their current state* (default). **Note**: A blank command parameter is a toggle function, and is therefore NOT suited for use from a template.
	SET_LEVEL_2_PREVIEW	<empty>	This setting applies to fader second levels in the template that is in *Preview*. The operation will be performed at next transition. <empty>: (no value entered) The second level for faders in the template in Preview are set to *TOGGLE their current state*. **Note**: A blank command parameter is a toggle function, and is therefore NOT suited for use from a template.
	FREEZE_AUDIO	TRUE FALSE <empty>	TRUE: All sound faders are frozen, and do not respond to commands from templates. FALSE: All sound faders respond to commands from templates. <empty>: (no value entered) All faders are set to *TOGGLE their current response state* (default). **Note**: A blank command parameter is a toggle function, and is therefore NOT suited for use from a template.

Copyright © 2024 Vizrt

COMMAND	Value	Parameter	Description
AUTOTAKE	N/A	(*)	
AUTOTRANS	PP, ME1, ME2, ME3, ME4	MixEffect (MIX OR WIPE) + Transitionrate	For example: MIX 33, WIPE 10. ⚠ **Note**: The *Effect* cannot be used.
	AUX, Default	(*)	
CROSSOVERCOMMAND	N/A	(*)	

Copyright © 2024 Vizrt

COMMAND	Value	Parameter	Description
DEVICE_PROPERTY	CAMERA CONTROL	See next column.	**Note**: This is for *camera robotics*. When the DEVICE_PROPERTY control command is used with the value 'CAMERA CONTROL', only setting speed (Camerobot) and presenter (Camerobot/FxMotion) are controllable from a template. Details for Camerobot **Presenter**: ANCHOR=pos1:presenter1[,pos2:presenter2]... The position and presenter is given as a pair divided by ':'. Multiple position-presenter pairs are possible, separated by ','.  **Note:** The position is a *string* and does not have to be a number. Details for Camerobot **Time**: If Time =1 (>0), Viz Mosart sends MATRIX_MOVE_MPOS command to Camerobot, while if Time = 0 Viz Mosart sends MATRIX_CUT_MPOS command Camerobot.

Copyright © 2024 Vizrt

COMMAND	Value	Parameter	Description
	AUDIO	demands <fields> structure as parameter with parameter1=AUDIO, parmeter2=key, parameter3=value OR connectionstring style	⚠ **Note**: This is for an *audio mixer*.
DEVICE_STANDBY	AUDIO_MIXER, GPI, LIGHTS, LOUDNESS, ROUTER, SOUND_PLAYER, SUBTITLING, VIDEO_SERVER, VIDEO_SWITCHER, VIDEO_WALL, VIRTUAL_SET, WEATHER	ON OFF TOGGLE	ON: The device chosen (in Value) is put in Standby. OFF: The device chosen (in Value) is taken out of Standby. TOGGLE: The Standby status (ON/OFF) of the device chosen (in Value) is toggled. (If there are several devices of the chosen kind (e.g. video servers), *all* of them are affected.)
DEVICE_STANDBY	GRAPHICS (full-screen)	As above (ON/OFF/TOGGLE) OR engineno:[ON|OFF|TOGGLE] (Example: 2:TOGGLE)	ON/OFF/TOGGLE: As above, applied to all graphics engines. If an engine number is given, only the designated engine is affected.

Copyright © 2024 Vizrt

COMMAND	Value	Parameter	Description
DEVICE_STANDBY	ROBOTIC_CAMERA	As above (ON/OFF/TOGGLE) OR controller:[ON|OFF|TOGGLE] (Example: 1:ON) OR controller:camera:[ON|OFF|TOGGLE] (Example: 1:2:ON)	ON/OFF/TOGGLE: As above, applied to all cameras on all controllers. If a controller (but no camera) is given, only the cameras on the designated controller are affected. If both a controller and a camera is given, only the designated camera is affected.
DEVICE_STANDBY	OVERLAY_GRAPHICS	As above (ON/OFF/TOGGLE) OR Engine No:[ON|OFF|TOGGLE] For example: *2:TOGGLE*: Toggles Engine 2 Standby status)	ON/OFF/TOGGLE: Applied to all overlay graphics (all controllers and corresponding engines). If an engine number is given, only the designated engine is affected. **Note**: Only *engine number* is supported. Engine numbers are shown in the list of active engines in Overlay Graphics Interface and followed by '/' and destination name of the engine. (*Engine Handler*, *Destinations*, *Controller* have *no effect*).
DIRECTTAKE	The Recall Nr (see Template Properties) of the direct take template to execute	-	Executes the direct take template given by Value.
DVE	FORWARD REVERSE	Recall forward/reverse EMEM	n/a
ENABLE_GRAPHICS_MIRRORING	n/a	(*)	

Copyright © 2024 Vizrt

COMMAND	Value	Parameter	Description
FULLSCREEN_GRAPHICS	CONTINUE_FULLSCREEN	AUTO	Continue Command Only: The AUTO parameter will replace the continue command with the number of continue points in the graphics item. The number of continue points is taken from the graphics_continuecount field. Examples: *AUTO+1*: The number of continue counts from the graphics + 1 *AUTO-2*: The number of continue counts from the graphics - 2
FULLSCREEN_GRAPHICS	CONTINUE_FULLSCREEN	Engine no	The target output fullscreen engine number to execute the action on. (e.g. 1,2,3)
	MACRO	Engine no : macro	The target output fullscreen engine number to execute the action on. (e.g. 1,2,3) and Macro name. The engine number must be same as the number defined in AV Automation
GRAPHICSPROFILE	N/A	-	
HOLD_AUDIO_TRANSITION	N/A	-	
HOLD_VIDEO_TRANSITION	N/A	-	
LIGHT	N/A	(*)	
MARKER	N/A	(*)	
NCS	START_STATUS, STOP_STATUS	RUNDOWN STORY ITEM	Specific for Open Media

Copyright © 2024 Vizrt

COMMAND	Value	Parameter	Description
NEXT_CUE_DELAY	EXTEND	A number of frames	If there is a pending cue (when this control command is executed), this cue is *postponed* by the number of frames given. Otherwise (no pending cue), the next cue operation will be *additionally delayed* by the number of frames given. (I.e., the value will be added to the sum of the settings · Min. Cue delay in AV Automation Devices Vision Mixer · **Next cue delay** in Audio and Video Setup, Vision Mixer Effects Configurations, Settings (if an effect is involved)).
OVERLAY_GRAPHICS	CLEAR	Destination or Render (Engine no.)	Destination or Render (Engine no.) to clear the graphics from. Numerical value: Render (Engine no.) Non-numerical value: Destination.
	CONTINUE	Engine no	
	TAKE_MANUAL_OUT	Destination or Empty	Destination to take out the graphics from. Non-numerical value: Destination Empty or Numerical value: Takes out all the manual graphics **Note** For versions before Viz Mosart 5.5.0, the TAKE MANUAL OUT command would also take out overlay graphics set to wait for background end or story end. If this functionality is needed, use the below TAKE ALL OUT command introduced in Viz Mosart 5.5.0.

Copyright © 2024 Vizrt

COMMAND	Value	Parameter	Description
	TAKE_ALL_OUT	Destination or Empty	Destination to take out the graphics from. Non-numerical value: Destination Empty or Numerical value: Takes out all the graphics
	MACRO	Engine no : macro	For example: 4:macrohere
	PRETAKE_NEXT_OVERLAY	Render (engine no)	
	TAKE_NEXT_OVERLAY, TAKE_NAMED_OVERLAY	(*)	
OVERLAY_TO_MANUAL	Parameter 1: ONAIR (default), PREVIEW.	Parameter 2: Comma-separated list of handler names Parameter 3: AUTOMATIC (default) / MANUAL Parameter 2 and Parameter 3 must be separated by a semicolon. (E.g. WALL,DSK;MANUAL.)	⚠ **Note :** If there is no semicolon after Parameter 2; nothing to the right of the semicolon; or Parameter 3 has any value other than 'MANUAL', then Parameter 3 gets the default value 'AUTOMATIC'. · For user operations, see the Viz Mosart User Guide, section *Viz Mosart User Interface > Keyboard Shortcuts > Control Command Keys > OVERLAY_TO_MANUAL*.
PLAY_STORY	N/A	(*)	

Copyright © 2024 Vizrt

COMMAND	Value	Parameter	Description
RECORD	PREPARE, START, STOP	Split parameters using, - first: clipname, - second: recorder (optional), - third: port name (default *Rec*), - fourth: group name (default *Rec*)	For example: clip33,recorder1,Rec,Rec • For more configuration details, see the section RECORD Control Commands in AV Automation Template Properties below.
RELEASE_BACKGROUND	N/A	(*)	
RUNDOWN_NCS_RESYNC	N/A	(*)	
SEQUENCE	LOOP	-	Sets a running sequence to start looping
	STOP_LOOP	-	Stops running a looped sequence in loop
	STOP	-	Stops a running sequence
	START	-	Restarts a previously stopped sequence
SET_AUX_CROSSPOINT	N/A	(*)	

Copyright © 2024 Vizrt

COMMAND	Value	Parameter	Description
SET_CROSSPOINT	PP, ME1, ME2, ME3, ME4 AUX Default	bus (A,B,C,D) : xpoint	If layers are supported by the driver (e.g. vizrt-ips) it is possible to also specify the layer (A,B,C,D) in addition to the crosspoint. Two arguments are supported separated by semicolon. First parameter is crosspoint, the second is layer. If no semicolon is used the value is interpreted as crosspoint only and bus A is used. Example: B:CAM2
SET_CURRENT_ME	PP, ME1, ME2, ME3, ME4 AUX Default	(*)	
SET_VIDEOSERVER_SALVO	Select a configured video server from the list	-	These same settings are available as keyboard shortcuts, as presented in the *Viz Mosart User Guide*, under section *Viz Mosart User Interface*.
STUDIOSETTUP	#	-	
SWITCH_GRAPHICS_MIRRORING	TOGGLE, ACTIVATE, DE-ACTIVATE	-	For standard and Story Recorder (SR) shows.
SWITCH_GENLOCK_MODE	TOGGLE, ACTIVATE, DE-ACTIVATE		Story Recorder (SR) specific: As long as some standard show elements are not supported with SR, use this command to control the use of Genlock mode. SR can only run in Genlock mode.

Copyright © 2024 Vizrt

COMMAND	Value	Parameter	Description
SWITCH_REHEARSAL_MODE	TOGGLE, ACTIVATE, DE-ACTIVATE		Story Recorder (SR) specific: SR can only run with Rehearsal mode deactivated, use this command to ensure Rehearsal mode is not accidentally active.
SWITCH_VIDEOSERVER_ MIRRORING	N/A	-	For standard and Story Recorder (SR) shows. This command has no parameters, it simply acts as a toggle.
TAKE_CONTINUE_POINT	N/A	(*)	
TAKE_SERVER_TO_PROGRAM	N/A	transRate (integer) - parameter. (*)	Takes a video server port to program on a selected ME. For example: a video clip is running on a video wall. The shortcut can then be used to take the last used/active video server to program.
TRANSITION_TYPE	CUT, MIX	Rate	
	EFFECT	Effect no	
	TOGGLE	(*)	
UNKNOWN	N/A		Default fallback. Does nothing.
VIDEO_PORT	N/A	Must be a comma separated list in the format: command,port,parameter	For example: PLAY_PAUSE,A,PLAY. · For more configuration details, see the section VIDEO_PORT Control Commands in AV Automation Template Properties below. · For user operations, see Viz Mosart User Guide, section *Viz Mosart User Interface > Keyboard Shortcuts > Video Port Control Commands >Video Port Keyboard Shortcuts*.

Copyright © 2024 Vizrt

COMMAND	Value	Parameter	Description
VIDEO_SERVER_GOTO	N/A	(*)	
VIDEOWALLMODE	N/A	(*)	
WEATHER	N/A	(*)	

Parameters with Placeholders

Control command parameters can include *placeholders* which are then populated with values found among the fields of a currently On Air, Viz Mosart item. A placeholder is defined as a string of characters, within curly brackets {placeholder}.
For example, a placeholder can be useful for transmitting values from the NRCS, for use as control commands that will be executed when the template is taken On Air.

Example: Providing the clip name to be recorded from NRCS.

In the NRCS, a column can be created for giving the name of the clip to be recorded with Viz Mosart. Then, in Viz Mosart Newsroom settings, map the column in *Story External Metadata* as follows:

Story External Metadata

```
<mostag mostagname="NCScolumn" mosart_action="item_variable" action_value="NCS_filename" format_type="String" />
```

In Manus file, the Mosart item will have the following fields (the value *Clip1* is given from the NRCS):

Mosart Item

```
<fields>
     <field name="NCS_filename" value="Clip1" />
</fields>
```

Then, in **Template Editor**, the control command for the respective template is configured as follows:

Control Command

```
RECORD PREPARE {NCS_filename}_Test,Recorder,RecPort
```

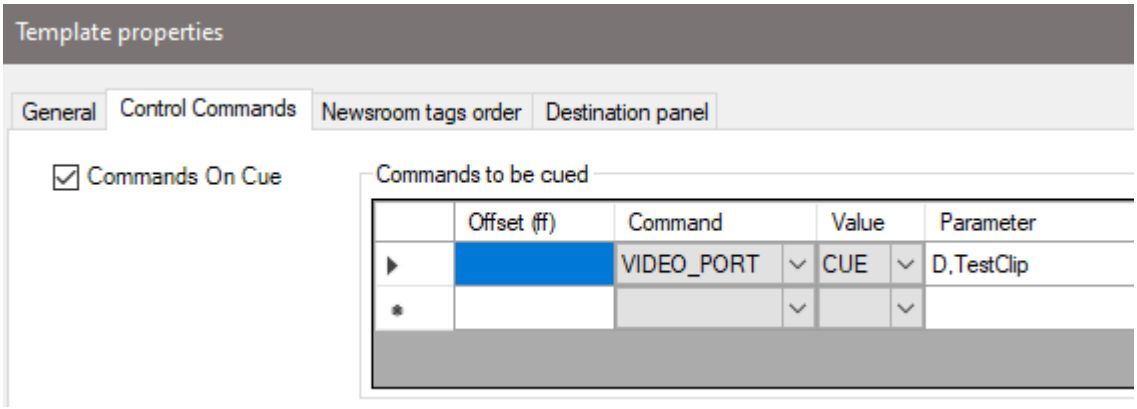
When the template is executed, a clip with name *Clip1_Test* is prepared for recording.

Copyright © 2024 Vizrt

VIDEO_PORT Control Commands in AV Automation Template Properties

- Examples
- VIDEO_PORT: Syntax
- VIDEO_PORT: Commands
- VIDEO_PORT VideoPort Syntax

The VIDEO_PORT control commands can also be used in templates. The syntax order of the parameter is important.



Examples

Examples using port A:

```
PLAY_PAUSE,A,PLAY PLAY_PAUSE,A,PAUSE PLAY_PAUSE,A
    STOP,A STOP,A,CUE
    RECUE,A,PLAY RECUA,A
    SET_LOOP,A SET_LOOP,A,OFF SET_LOOP,A,12-444
    CUE_TAIL,A,-15 CUE_TAIL,A,15
    PLAY_TAIL,A,-15 PLAY_TAIL,A,15
```

VIDEO_PORT: Syntax

Value=Command, Parameter=[@Delay,]VideoPort,Properties

VIDEO_PORT: Commands

- PLAY_PAUSE: Viz Mosart will begin to play the cued content. If the content is already playing, it will pause the content
  - No Properties: PAUSE if playing and PLAY if paused.
  - Properties=PLAY: Always try to PLAY channel :
  - Properties=PAUSE: Always try to PAUSE channel
- STOP: Viz Mosart will stop the cued content.
  - Properties: CUE. Cue the channel after the stop
- CUE: Viz Mosart will initially CUE the port.

Copyright © 2024 Vizrt

    - Properties: ClipName
- RECUE: Viz Mosart will RECUE the port
    - Properties: PLAY. Viz Mosart will play the clip after the recue
- SET_LOOP: Viz Mosart will set the already cued clip to looping
    - Properties: OFF, RESET. Viz Mosart will turn the looping off
    - Properties: 12-444 (configurable). Viz Mosart will set the looping from frame 12 to frame 44
- CUE_TAIL: Viz Mosart will recue the cued clip at a specific time (in seconds)
    - Properties: -15 (negative value) - configurable. Viz Mosart will cue at the start of the last 15 seconds of the clip (the clip will have 15 seconds left to play plus Post roll)
    - Properties: 15 (positive value) - configurable. Viz Mosart will cue 15 seconds into the clip
- PLAY_TAIL: Viz Mosart will recue and play the cued clip at a specific time (in seconds)
    - Properties: -15 (negative value) - configurable. Viz Mosart will cue at the start of the last 15 seconds of the clip
    - Properties: 15 (positive value) - configurable. Viz Mosart will cue 15 seconds into the clip.

All commands may be prefixed with an optional delay given in frames. If a delay is specified then the properties section shall start with an '@' character.

VIDEO_PORT VideoPort Syntax

The VideoPort may be given either explicit or using one of the following identifiers:

- ONAIR: The current onair video port used by a primary story item.
- PREVIEW: The current video port in preview from a primary story item
- VideoPortGroup,CURRENT: The current video port for the given video port group.

VIDEO_PORT Examples:

Value	Parameter / Properties	Description
PLAY_PAUSE	D	Alternatively plays / pauses the clip cued in Mosart port D
PLAY_PAUSE	D,PLAY	Always do a play for the clip cued in port D
PLAY_PAUSE	D,PAUSE	Always do a pause for the clip cued in port D
STOP	D	Stops any running clip in port D
STOP	D,CUE	Stops any running clip in port D followed by a recue

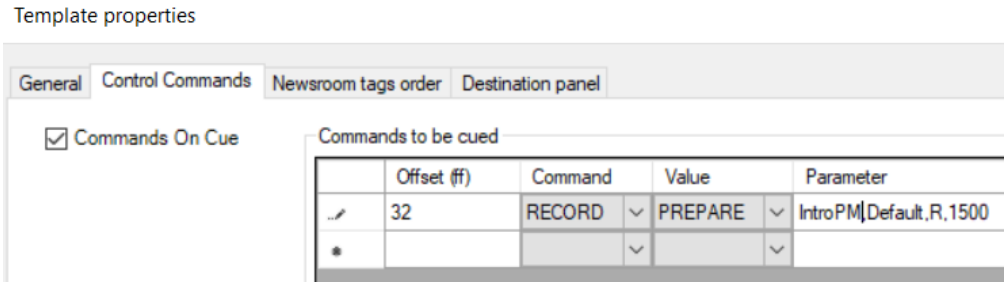
Copyright © 2024 Vizrt

Value	Parameter / Properties	Description
CUE	D,TestClip	Cues the clip TestClip in port D.
CUE	@25,D,TestClip	Cues the clip TestClip in port D after a delay of 25 frames.
RECUE	D	Recurs the clip in port D
RECUE	D,PLAY	Recues the clip in port D and plays it.

RECORD Control Commands in AV Automation Template Properties

- Examples
- RECORD: Syntax

The RECORD control commands can also be used in templates.



Examples

Value	Parameter	Description
PREPARE	TestClip,Default,R	Prepare a clip with name *TestClip* on the *Default* recorder. The *Default* recorder will be using video port *R*. For Quantel this clip will be initiated using the default duration.
PREPARE	TestClip,Default,R,1500	Prepare a clip with name *TestClip* on the *Default* recorder. The *Default* recorder will be using video port *R*. For Quantel this clip will have an initial duration of 1500 frames = 1 minute (25 frames/sec)

Copyright © 2024 Vizrt

Value	Parameter	Description
PREPARE	TestClip;Category=Totem ,Default,R,1500	Same as above but the clip will be assigned to the “Totem” category in a Quantel video server system. **Note**: There is a semicolon between *TestClip* and *Category*.
START	,Default	Starts the recording on the *Default* recorder. **Note**: There is a leading ‘,’ in the *Parameter*
STOP	,Default	Stops the recording on the *Default* recorder **Note**: There is a leading ‘,’ in the *Parameter*

The options in the *Value* dropdown are

- PREPARE
- START
- STOP
- STOP_AND_DETACH
- DELETE.

The *Parameter* should be a comma-separate list of up to four items:

- *ClipName*
- *Recorder*
- *PortName*
- *Duration*.

If all four items are present, the Parameter will be

<ClipName>,<Recorder>,<PortName>,<Duration>

RECORD: Syntax

If not all items are present, any commas to the left of present items will still have to be present. For example, if only *Recorder* is present, the syntax is:

,*<Recorder>*

Copyright © 2024 Vizrt

In some cases, the *ClipName* item may have sub-items. In that case, the sub-items are separated by semicolons. (An example is given below.)

Here the options in the *Value* dropdown are explained in more detail:

- PREPARE: Viz Mosart will prepare a clip for recording.
  - *Parameter: <ClipName>,<Recorder>,<PortName>,<Duration>*
  - *ClipName*: The name of the clip to be created
  - *PortName*: The name of the video port where the recording shall take place. Note that this port has to be part of the '*Rec*' virtual video port group.

    > **Note:** For the setup of recording ports, please refer to *The Viz Mosart Administrator Guide*. See also **Recording** in the *Viz Mosart User Guide*.

  - *Recorder*: A user-defined name to be used for the recording, tied to *ClipName* and *PortName*. Used by START and STOP.
  - *Duration*: Estimated duration of the recording in frames.
- START: Viz Mosart will start a recording previously PREPAREd
  - *Parameter*: ,*<Recorder>*

    > **Note:** Note the leading comma as explained above.

  - *Recorder*: Name of recording session as created by PREPARE
- STOP: Viz Mosart will stop a recording previously STARTed.
  - *Parameter*: ,*<Recorder>*

    > **Note:** Note the leading comma as explained above.

  - *Recorder*: Name of recording session as created by PREPARE

Copyright © 2024 Vizrt

8.3.4 Destination Panel

In the **Destination panel** tab, you can create rectangles that are then available in the Viz Mosart UI for assigning sources to newsroom tag destinations, while the template is in preview or program.

- Press **Add** to add the selected target area.
  The target areas are displayed in the Viz Mosart UI as drop down boxes. They become clickable if a source or shortcut is selected by the user.

See also section *Program Window* in the *Viz Mosart User Guide*.

Copyright © 2024 Vizrt

8.3.5 Mosart Templates Feedback to the NRCS

You can send Mosart templates as MOS objects to an NRCS. The MOS object can then be used by an NRCS operator to insert the detailed Viz Mosart item (rundown event) as a program cue into their rundown (some users call this a “brick system”).

This is an alternative to the Mosart ActiveX, and is the basis for the Viz Mosart web-based application, **NRCS Plugin**.

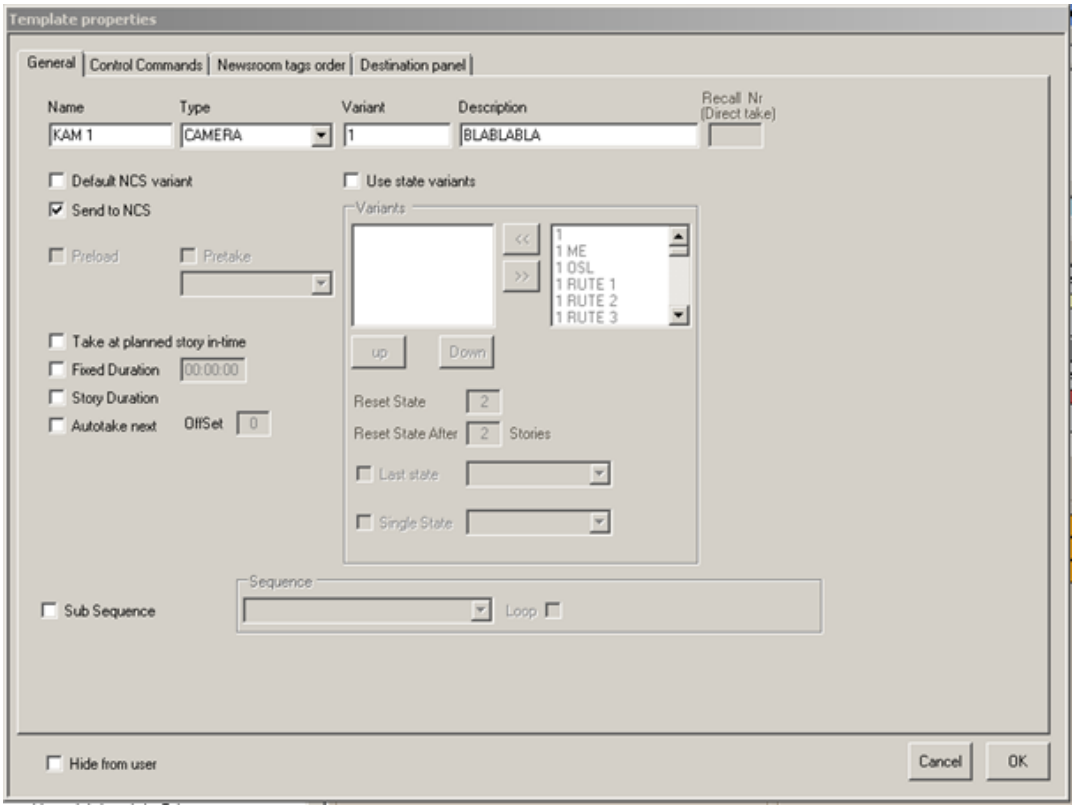
In addition to MOS objects that contain Viz Mosart primary template details, a special MOS object can be sent, containing *lower third* information.

- Template Properties Menu
- Working with Templates as MOS Objects
- Sending Viz Mosart Template Content to the NRCS
- MOS Object Containing Lower third Information
- Usage Notes
- Hierarchical Template Support
- Detailed Description of MOS Objects

Template Properties Menu

You can view and modify the properties of a selected template.

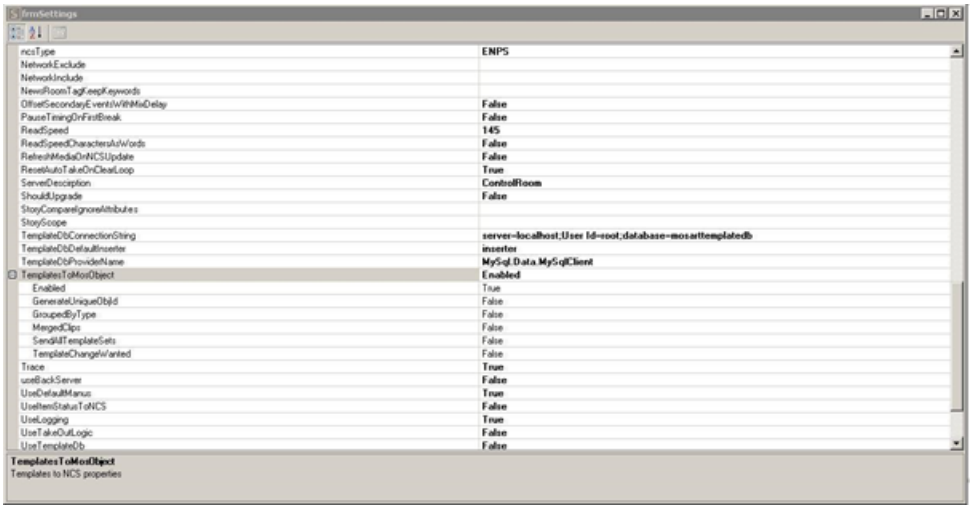
In AV Automation, select **Devices > Template Editor > Template > Properties**.



Working with Templates as MOS Objects

Enabling the Template-to-MOS Object Feature

You enable the features described in this section from the **Viz Mosart Administrator** (MOS version) **Settings** menu.



Settings Parameters

Value	Description
TemplateToMosObject	Used to send templates to NCS through MOS communication. *Enabled*: Enables sending templates to NRCS through MOS communication Default: *Enbaled*
GenerateUniqueObjectId	Some NRCSs (for example, OpenMedia) use **objId** to identify a template. *True:* The MOS object **objId** is sent to the NRCS as the channel template ID , defined as **GUID**. False: When **GenerateUniqueObjectId** is false and - **GroupedByType** (see below) is *false*, **objId** is set to the value of the *TemplateSet-Type-Variant*. - **GroupedByType** is *true*, **objId** is set to the value of the template *type*. **Note**: **objId** is unique among the templates from the *same* gallery. If the same channeltemplates file is used from two different galleries, MOS **objId**s are no longer unique.

Value	Description
GroupedByType	Use to send all template types as *one* MOS object, with the variants embedded in the objects from the default template set. · The **SendAllTemplateSets** property must be set to *false*. · The default template set must first be defined in · AV Automation: **Devices > Template Editor > Template Set > Set As Default**. · If the **SendAllTemplateSets** property is set to *true*, a MOS object for each template type (with the variants embedded in the object) is sent for each template set. · If no default template set is set, then *no* template will be sent to the NRCS. *True*: **objId** is set to the type of the template. *False*: Send all template variants as *separate* MOS objects. Default: *True*
MergedClips	Use this option to merge only the clips from the default template set, regardless of the value of **SendAllTemplateSets**. *True*: Send all PACKAGE and VOICEOVER templates as a collection of type "CLIP" to the NCS using the NEWT or UPDT commands in the Manus Admin console. This property has to be used together with **GroupedByType** (see above). · The default template set must first be defined in AV Automation: **Devices > Template Editor > Template Set > Set As Default**. · If no default template set is set, then *no* template will be sent to the NRCS. Default: *true* See Appendix section Structure of MOS objects sent to NRCS for details about the MOS object structure when merging clips.

Copyright © 2024 Vizrt

Value	Description
SendAllTemplateSets	Use to send templates from all template sets to the NRCS. · The default template set must first be defined in AV Automation: **Devices > Template Editor > Template Set > Set As Default**. *False*: Send only the default template set. Default: *False*
TemplateChangeWanted	Use to decide whether templates changed are updated in the NRCS. · When a template is changed or deleted on the Viz Mosart side, the NRCS will only receive an update if - **mosReqObjList/mosReqAll** (from the NRCS) or - **NEWT** or **UPDT** (from the Viz Mosart Manus console) has been executed at least once since Viz Mosart Manus console was started. · Only a template checked in the field **Send to NCS** (in the Template properties menu), is sent to the NRCS. *True*: Always receive updates.

Sending Viz Mosart Template Content to the NRCS

There are two ways to send Mosart templates to the NRCS:

1. Requesting from the NRCS using commands like
   a. mosReqObjList
   b. mosReqAll
   c. mosReqObj <objId>
2. From the Viz Mosart Manus console, using commands like:
   a. NEWT: Send the Viz Mosart templates to the NRCS as MOS objects with status *NEW*.
   b. UPDT: Send the Viz Mosart templates to the NRCS as MOS objects with status *UPDATED*.
   c. DELT: Send the Viz Mosart templates to the NRCS as MOS objects with status *DELETED* (meaning that the templates will be deleted).

· To see samples of the corresponding MOS XML code, see section Detailed Description of MOS Objects below.

⚠ **Note**: You can also use the new Viz Mosart NRCS Plugin to quickly drag and drop the underlying MOS object details of a selected Viz Mosart template, directly into an NRCS rundown.

Copyright © 2024 Vizrt

Caching MOS Objects for Sending to the NRCS

All the MOS objects corresponding to the templates sent to NRCS are saved in a cache file

- MosTemplateCache.xml under %localappdata%\temp\Mosart Medialab\MOS State.

This file is created or updated when Viz Mosart Administrator (MOS version) is started.

Troubleshooting

This file is useful when troubleshooting, enabling you to verify exactly what MOS objects have been sent to the NRCS.

You can delete it, then perform a fresh request of the templates, using a **NEWT** command.

Ignoring the templateset Attribute for a Viz Mosart Item

When sending *all* Mosart templates to the NRCS (**SendAllTemplateSets** is *true*), an attribute **templateset** is set inside the MOS object.
This attribute is used internally to identify which Viz Mosart template set the type and template type (variant) shall be used.

When a rundown is later loaded in Viz Mosart, the NRCS returns the MOS object with this **templateset** attribute set.

Example

1. You change the studio setup in the Viz Mosart UI by selecting another Template Set (the *active* template set) which contains some of the variants used in the previous template set, but with different properties (e.g same CAMERA 1, but with different crosspoints)
2. The item from the template set referred to by the attribute **templateset** will be executed, instead of the item from the new, *active* template set.
3. To get round this, you can instruct Viz Mosart to ignore the **templateset** attribute (by setting it to an empty value) from a MOS object coming from NRCS.
   Now the Viz Mosart item from the *active* template set will be executed.
   a. Do this by adding a **mosid** tag with value *MOSART* in newsroomsettings.xml with a keyword set to *IgnoreTemplateSet*:
   <tag name="some.mosid" value="MOSART" appendContent="false" fieldMapping="false" keywords="IgnoreTemplateSet=true"/>
   or
   <tag name="some.mosid" value="MOSART" appendContent="false" fieldMapping="false" keywords="IgnoreTemplateSet"/>

MOS Object Containing Lower third Information

- For *graphics*, please refer to section Sending Lower Third Graphics MOS Object to an NRCS.
- The corresponding MOS XML code is presented in Detailed Description of MOS Objects.

Copyright © 2024 Vizrt

Usage Notes

Automatic Sending of Viz Mosart Templates to the NRCS

Templates are automatically sent to the NRCS whenever Manus Admin on the Viz Mosart server is started/restarted if **Template feedback to NCS** is enabled in the Manus Admin settings and **TemplateChangeWanted** is set to *true*.
In this case, Viz Mosart will send:

- All templates from all sets if **SendAllTemplates** is *true*.
- The templates from default template set, if **SendAllTemplates** is *false* and a template set has been defined as default.
- No templates, if **SendAllTemplates** is *false* and no template has been defined as default.

See the Settings Parameters table above.

Creating and modifying Templates

- When a new template is created or an existing template is changed
  - The template set attribute is defined in the MOS object
  - <objGroup> is set to the template set that the template belongs to.
  - On the NRCS side, the template is stored in a folder named after the template set.
- When a template is changed or deleted, the NRCS will only receive this update when either the commands
  - mosReqObjList/mosReqAll (from the NRCS) or
  - NEWT or UPDT (from Viz Mosart Manus console)
  are executed at least once since Viz Mosart Manus Admin was started.
- To always receive these updates, set **TemplateChangeWanted** to *true*.

  ⚠ **Note**: When **GroupedByType** is set to *true*, deletion of templates is not handled.

  - Execute UPDT in the Viz Mosart Administrator console or request the templates from the NRCS for update

Hierarchical Template Support

The template feedback to NRCS functionality partially integrates with Hierarchical Templates, supporting the two operations

1. Requesting a single template set
2. Requesting *all* template sets

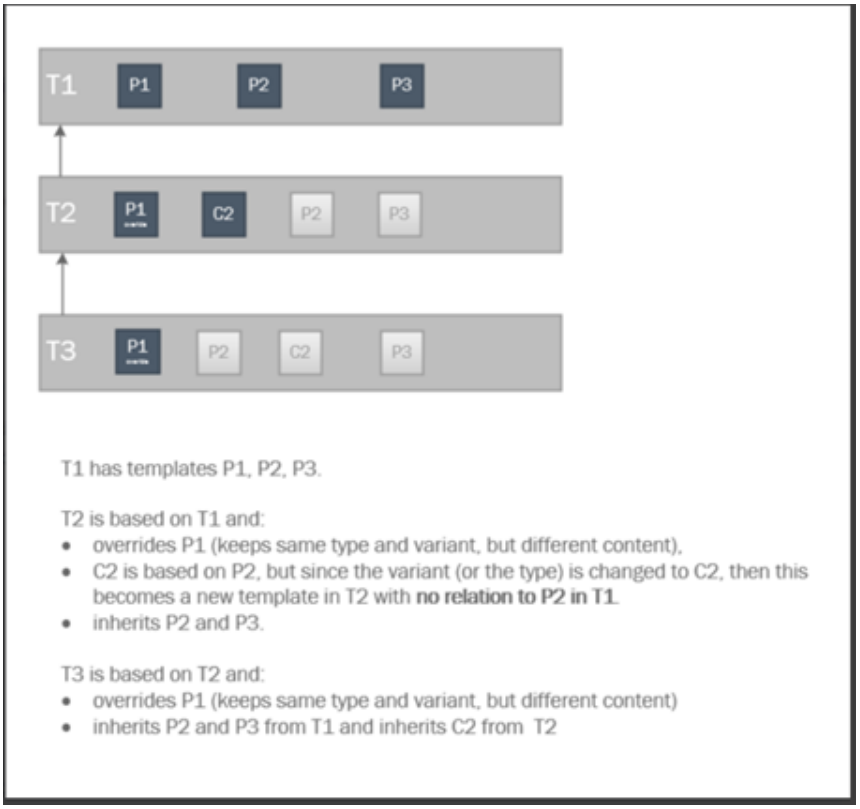
from a gallery with hierarchical templates.

⚠ **Note**: The management of a template set (NEW, UPDATE, DELETE a template) is *not* supported.

Copyright © 2024 Vizrt

Example

Assume the following template sets:



- If template set T1 is requested, then P1, P2 and P3 are sent.
- If template set T2 is requested, then P1 and C2 from T2 and P2 and P3 from T1 are sent.
- If template set T3 is requested, then P1 from T3, C2 from T2 and P2 and P3 from T1 are sent.

Detailed Description of MOS Objects

See the Appendix for code-level descriptions of

- MOS Objects with Mosart Templates Grouped by Type
- MOS Object with Mosart Templates not Grouped by Type
- MOS Object with Clip Templates Merged in a Mosart Template
- MOS Object with Lowerthird Information
- Octopus

Copyright © 2024 Vizrt

8.3.6 Controlling a REST-based Driver

- Template Commands
- Command Management
- Troubleshooting

Working with a REST-based Driver

Viz Mosart offers an easy to create, generic driver which can send REST commands to an external device. The initial driver construction is explained in section Generic REST Drivers.

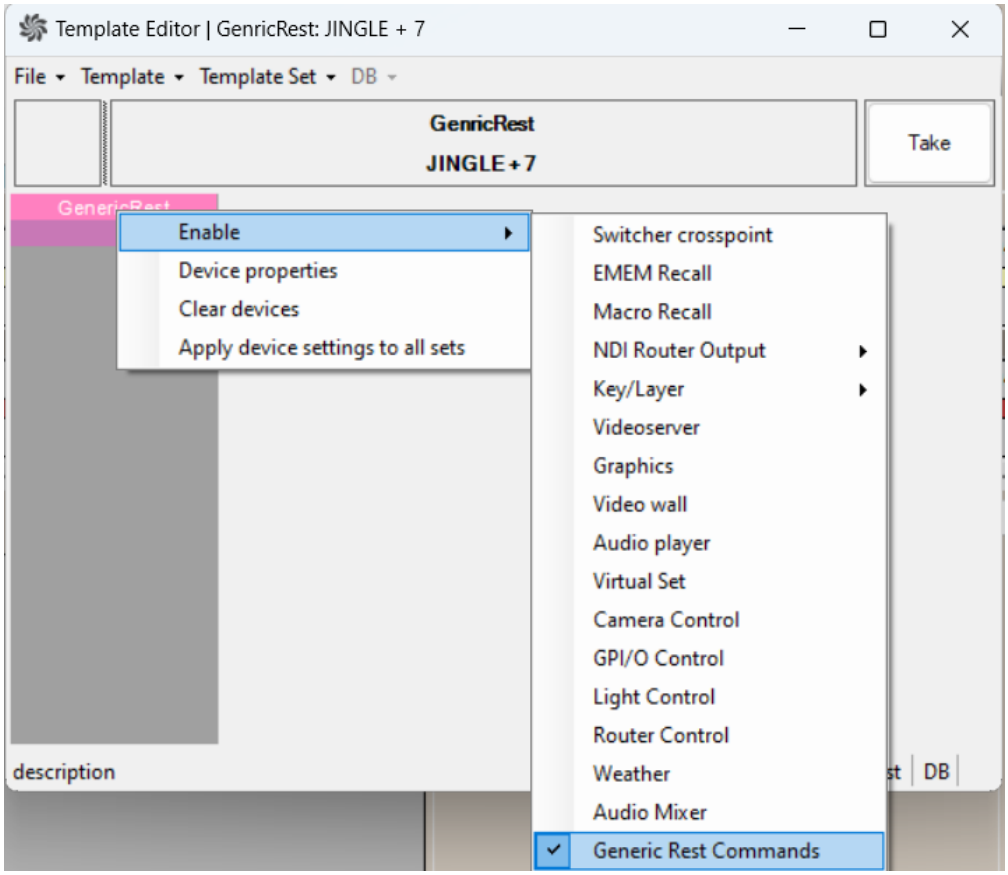
This section explains how to communicate with a REST-based driver from a template.

Template Commands

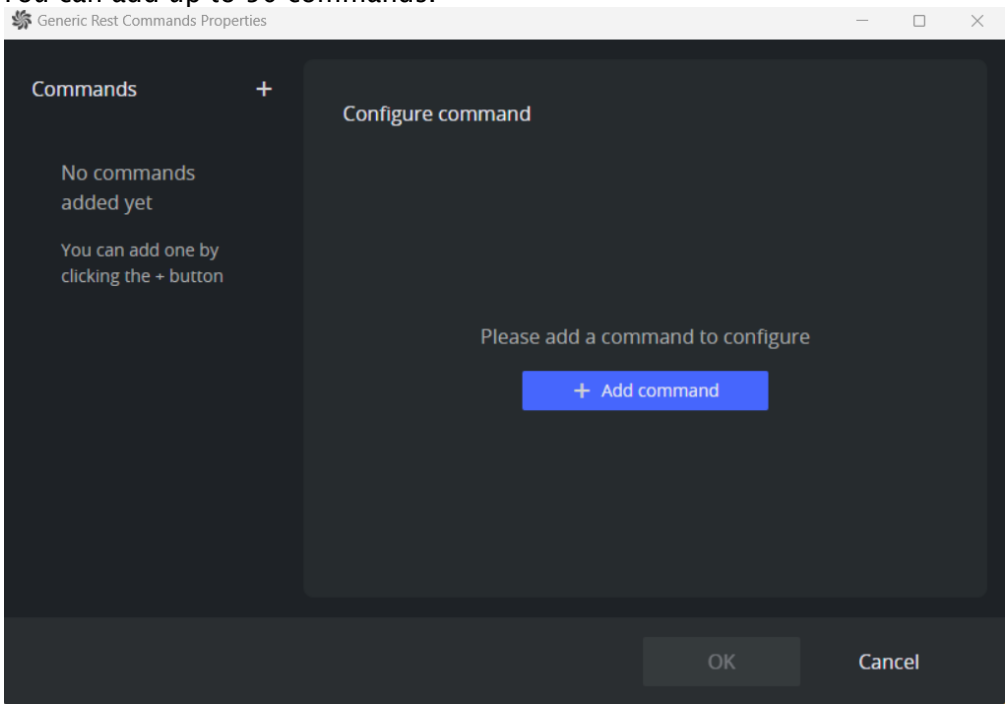
To Configure REST Commands

Begin by globally enabling Generic Rest Commands:

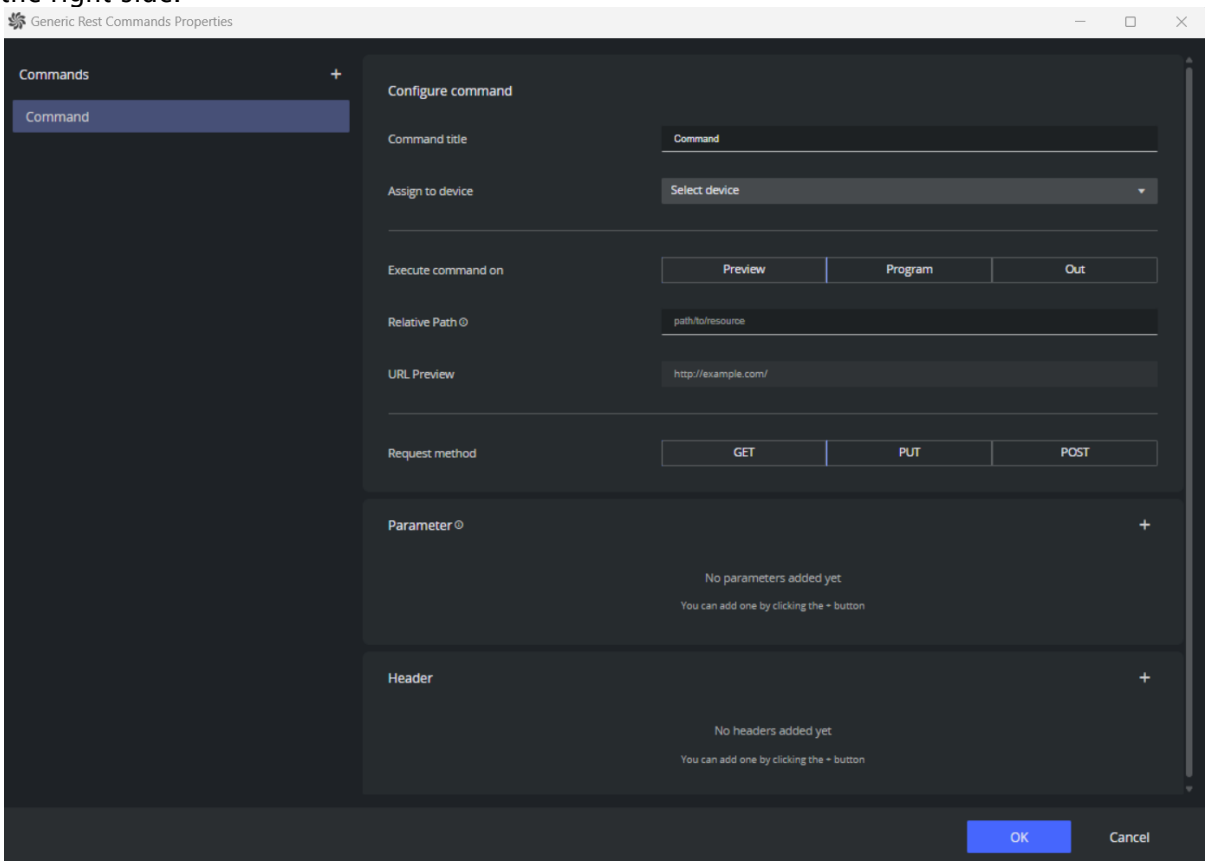
1. Navigate to **AV Automation menu > Devices > Template editor**.
2. Select or create a new template.
3. In the device's Function area (black row), right-click and select **Enable > Generic REST Commands**.



4. In the **Generic REST Commands Properties**, click the Plus (+) sign or **Add command** to add a new REST command.
   You can add up to 50 commands.



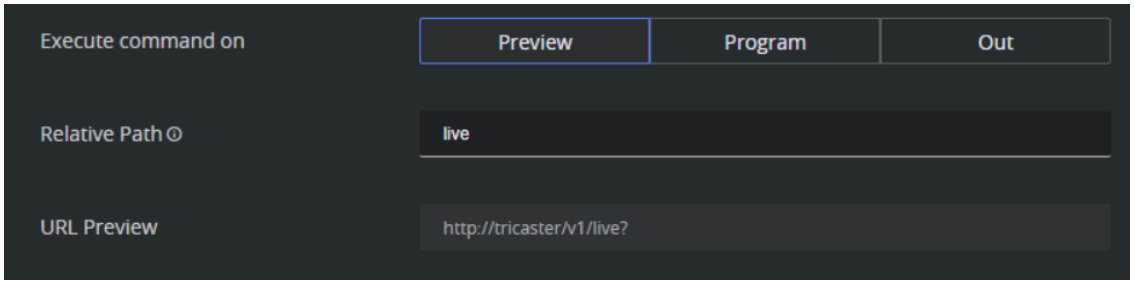
5. A new command appears in the **Commands** list (left side) with its properties configurable on the right side.

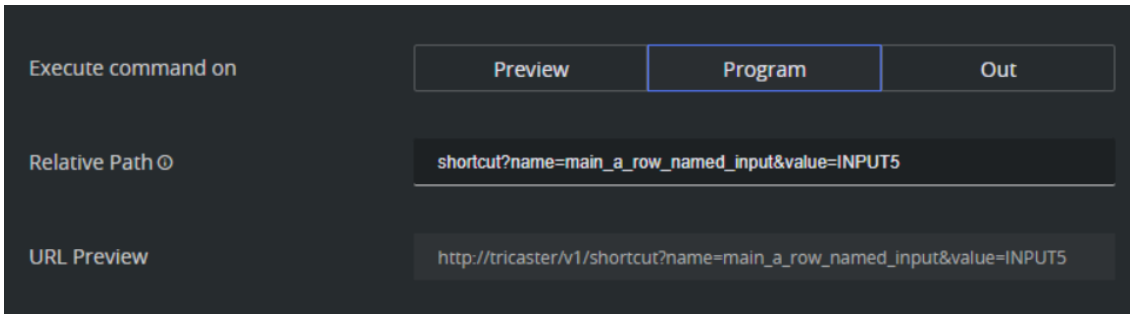


6. Configure the properties for a Generic REST command as follow:

   - **Command title**: The name that identifies the generic REST command. Default: *Command*.
   - **Assign to device**: Select the Generic REST device configured in Mosart Administrator Settings to execute the command. If no device is selected, the command will not be executed. Default: *empty string*.
   - **Execute command on**: Specifies when Mosart should execute the command in relation to the template lifecycle. Options include when the template is *cued*, *taken* on-air, or when the template is *taken out*. Default: *Preview*.
   - **Relative Path**: The resource path appended to the Base URL configured for the selected device. This path should not start with a scheme (e.g., http://) or a leading slash. It can optionally include query string parameters. For instance, if a parameter value does not require a newsroom tag, it can be added directly to the Relative Path. URL parameters contain a key and value, separated by an equal sign (=). A URL can contain multiple parameters, which are then each separated with an ampersand (&).

Copyright © 2024 Vizrt

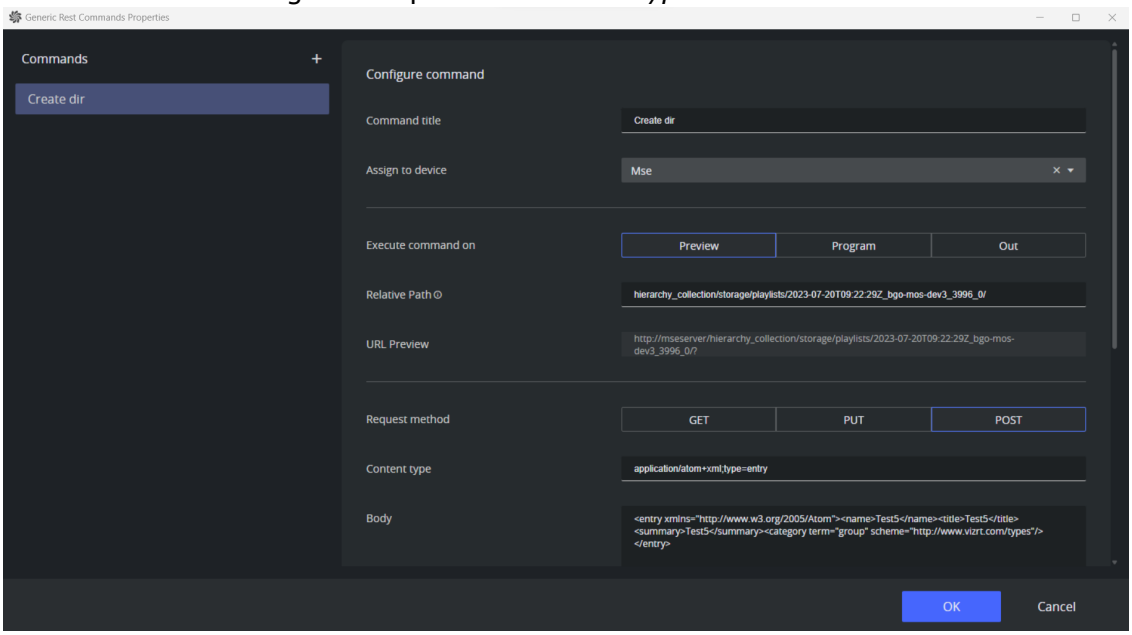




- **URL Preview:** This field displays a preview of the complete URL, composed of the *Base URL*, *Relative Path*, and any extra parameters specified in the *Parameters* section.
- **Request method**: The HTTP method used for the request. Available options are *GET*, *PUT*, and *POST*. For PUT or POST methods, the *Content Type* and *Body* of the request can be configured.
- **Content Type**: Specifies the value for the "Content-Type" HTTP header, which indicates the media type of the resource or data in the request body.
  Required for PUT and POST methods.
  Common Content-Type values include *text/plain*, *text/xml*, *application/xml*, *application/json*.
  The *charset* parameter can be used to set the character encoding for the text being sent. Default: *text/plain; charset=utf-8*.

Copyright © 2024 Vizrt

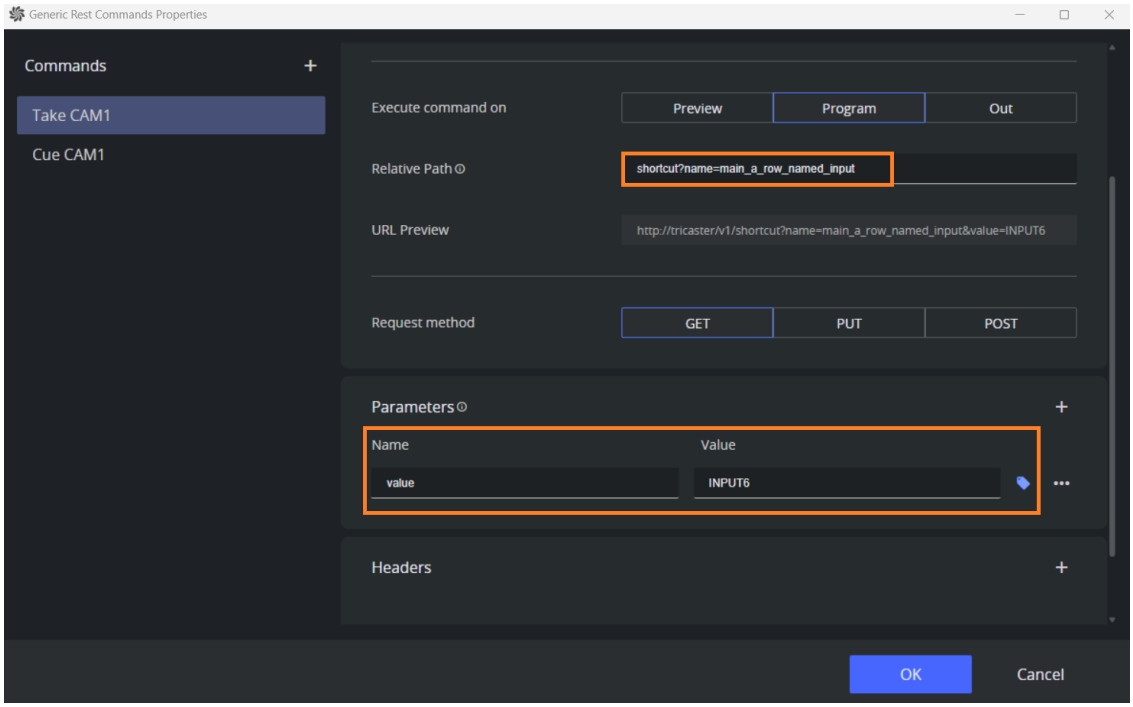
- **Body**: The data sent in the request body. Required for PUT and POST methods. It must be formatted according to the specified *Content Type*.



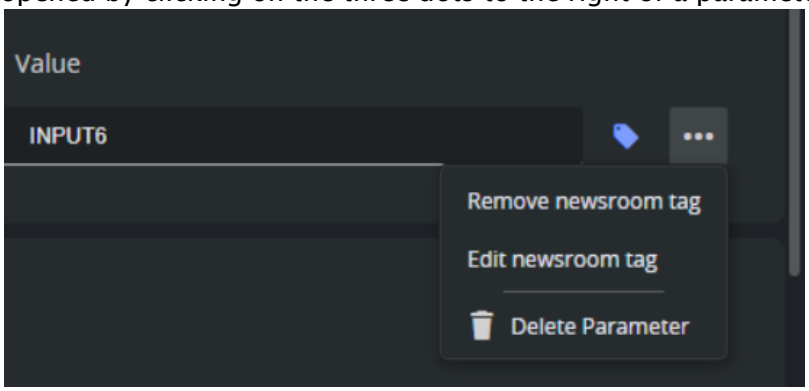
- **Parameter**: Additional parameters can be included in the command request by adding them in this list using the Plus (+) sign. This gives the possibility of providing *newsroom tags* for the parameters' values, enabling dynamic control from the Newsroom system (NRCS).

  In the following example, two parameters are added to the request command.
  - Parameter **name** is attached to the *Relative Path* with a static value *main_a_named_input*.
  - Parameter **value** is added in the *Parameters* section with static value *INPUT6* and a newsroom tag attached. This enables the value to be overwritten from the NRCS when planning the show design.

Copyright © 2024 Vizrt



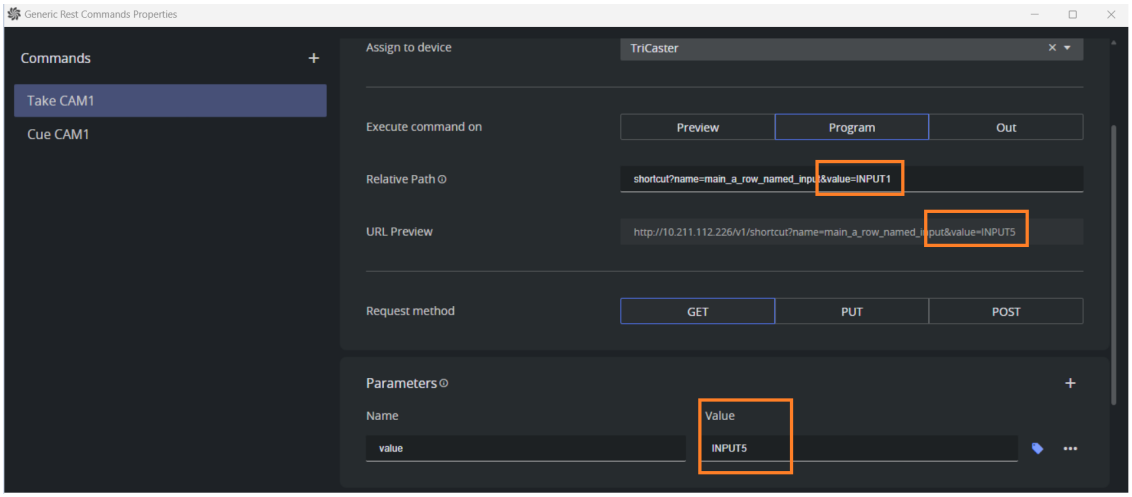
A newsroom tag can be added/removed/changed using the options in the menu opened by clicking on the three dots to the right of a parameter:



If a parameter is included in both the *Relative Path* and the *Parameters* list, the one in the *Parameters* list will take precedence and overwrite the value provided in the

Copyright © 2024 Vizrt

Relative Path.

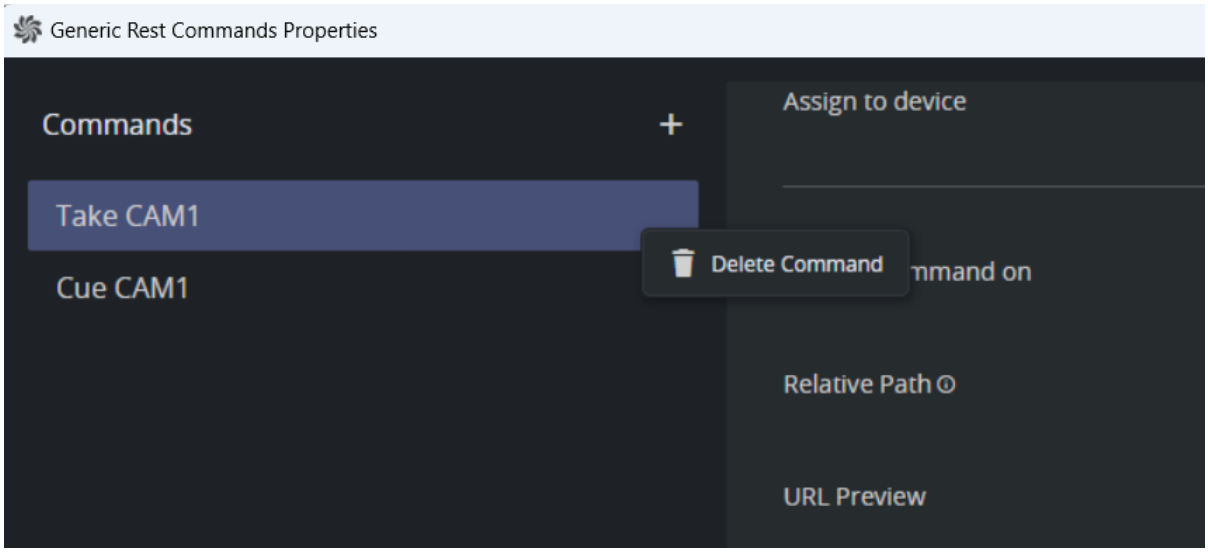


- **Header**: Additional HTTP headers can be included in the request for setting authentication tokens, cache control, or other custom headers required by the REST server.
  Headers are added similarly to parameters, with the option of using newsroom tags for header values.

Command Management

To Remove A Configured Command

- Click on the three dots to the right of an item in the list of commands and select **Remove command**.

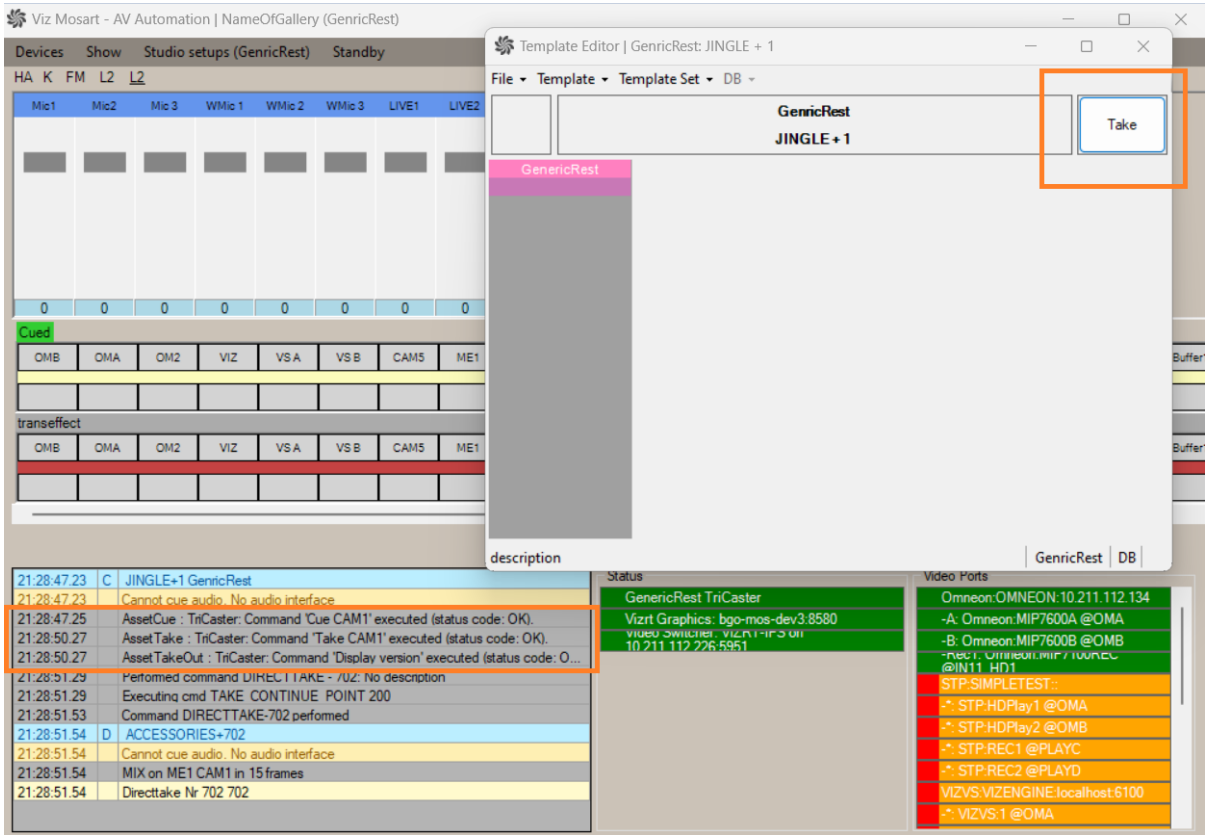


Troubleshooting

Here are some steps to take if results are not as expected.

To Verify The Execution Of A REST Command

1. To verify the execution of REST commands configured in a template, run the template using the **Take** button.
2. Check the lower-left Log area of AV Automation for execution confirmation from the Mosart server.



3. Confirm the expected outcome on the target device.

If the command failed to execute, verify Mosart logs for more information.

To Verify A REST Command Is Correctly Configured

- Copy the URL from the field **URL Preview** and execute the command using a browser or a REST client application like *Postman*.

Copyright © 2024 Vizrt

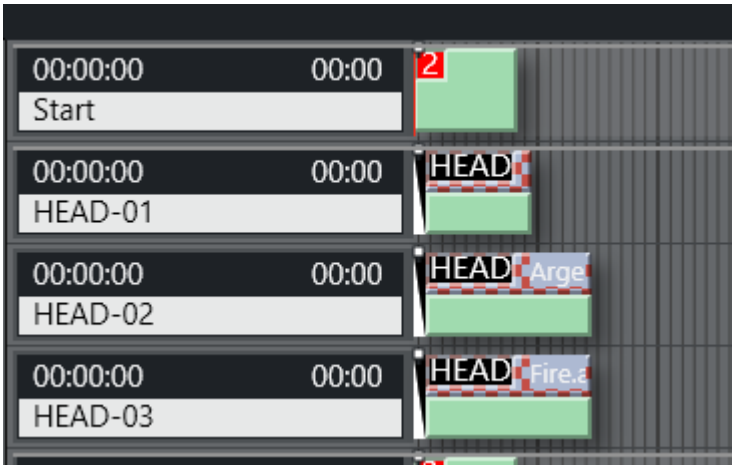
8.4 Other Template Functionality

- State Variants
- Dynamic ME Allocation
- Template Editor Password

8.4.1 Additional Template Features

State Variants

A story sequence comprises items based on templates. With the help of state-based variants, a single template can be used for *all* stories. For example, the figure below shows, after the first story, a sequence of three stories where each story contains a single story item, and each story is using the *same* Viz Mosart template (VOICEOVER+HEAD). State variance enables a different template variant to be used depending upon the story's position in the story sequence.



For example, the first story in a sequence can behave differently to the remaining stories. Another example is when combining a sequence with music, so that each story in the sequence has a specific length, synchronized with the music.

In both examples, is it necessary to invoke different templates depending upon the position of the story in the sequence. You can make it simple for the user to add their stories by just referring to the master template (HEAD in this example).

> **Note:** Template variance only works on *Story* level, meaning that only *one* story item / template should be assigned per story.

Copyright © 2024 Vizrt

Example of Using State Variants

Continuing with the headlines example presented above, the templates can be placed into a sequence.

1. The template called HEAD is the *master template*, and the first story with a HEAD variant uses this template.
2. The second story with a HEAD variant uses the template called HEAD2,
3. HEAD3 is used for the third story.
   In this way, the only command (template variant) used in the newsroom system is HEAD, regardless of whether it is the first, second or third.
4. You can now change the order of the headlines without any modification to the script, making it very quick to do last minute changes.
   In the same way, you can also define which template to use if there is only one HEAD, or if the last HEAD template should act in a specific way.

To Configure a Template State Variant

Configuring template state variance is done in the template editor using the **Template Properties** dialog.

1. Select any template as the *master template* which will be used as the first template in the sequence.
2. Add additional variants in the sequence using the **<<** and **>>** arrow buttons.
   For a particular template type, it is only possible to select variants of the same template type. The same template variant may be used multiple times.
3. Use the **Up** and **Down** buttons to shuffle the variant order.

The following properties provide additional control over which template variant will be used:

- **Reset State**: Index number of the template variant that will continue, if the number of stories in a sequence is larger than the specified number of variants. The index *0* refers to the *first* variant (i.e. the master template itself).
- **Reset State After**: The number of stories not belong to the sequence, that can be run before the new sequence is started. This allows the variance sequence to be kept, even when there

Copyright © 2024 Vizrt

are stories within the sequence that are not part of the sequence itself. See the example below.

- **Last state**: Special variant to be used for the *last* story in the story sequence.
- **Single State**: To be used as the last state when there is only one story in the sequence.

> ⚠ **Note:** When using **Single State**, the property **Last state** must be defined.

Example - Reset State After

This example continues with the story sequence presented above with the three stories all using the HEAD template.
The HEAD template variant has a variance sequence of *three* item: HEAD, HEAD-2 and HEAD-3.
Executing the story sequence then invokes the template variants "HEAD", "HEAD-2" and "HEAD-3" for the three stories respectively.

If you now add another story sequence, for example HEAD, HEAD, *OTHER-STORY*, HEAD, with **Reset State After** defined, then executing this story sequence invokes template variants depending upon the value of **Reset State After**:

- **Reset State After <= 1**: HEAD, HEAD-2, OTHER-STORY, HEAD
  The sequence is restarted since **Reset State After** is less than or equal to 1 (story).
- **Reset State After > 1**: HEAD, HEAD-2, OTHER-STORY, HEAD-3
  The sequence is continued since the **Reset State After** is larger than 1 (story).

Dynamic ME Allocation

- To be able to use dynamic allocation, the vision mixer must have at least two mix effect banks in addition to your program/preset bank.
- Viz Mosart uses ME1 and ME2 for the dynamic allocation.

> ⚠ **Note:** Programming of the emem registers on the vision mixer vary between the various models on the market. Contact Viz Mosart support for instructions on how to prepare your switcher to work with Viz Mosart's dynamic ME allocation.

- When ME's are included in a template with crosspoints or an emem recall, it's possible to do a *dynamic selection* of the ME, by choosing M_RIP or M_OTH.
  Dynamic allocation can be useful in situations where you want an effect to load on an ME not in use, or to set a crosspoint on the ME currently in use.
- To allocate a "new" ME in your template, then choose M_RIP (ME ripple). This loads the effect or set crosspoint on the ME that has the status of "not in use".
  When this template goes On Air, the status of the ME will change to "in use". The next template that uses M_RIP will then load on the other ME.
- Conversely, use M_OTH (ME other). Using this in a template loads the effect or set crosspoint on the ME with the status "in use", and does *not* cause a change of status between the two ME's.
  This means that if the next template also uses M_OTH, it addresses the same ME as the previous.

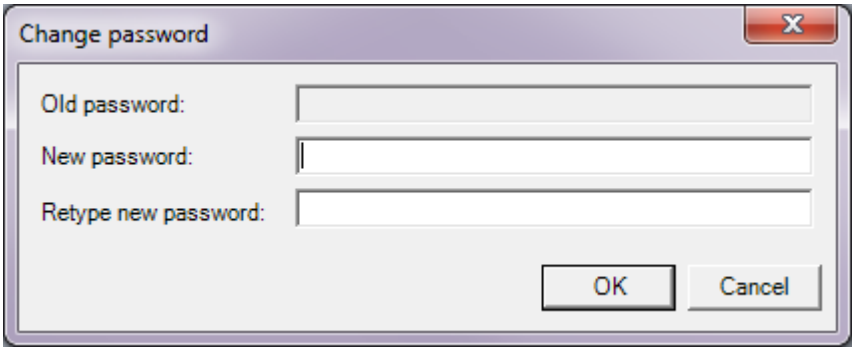
Copyright © 2024 Vizrt

Template Editor Password

Password protection can be enabled for the Template Editor. When enabled, the user is prompted for the password when opening the Template Editor.
By default, *no* protection is enabled, and the user is not asked for a password before opening the Template Editor.

To Enable or Change a Template Editor Password

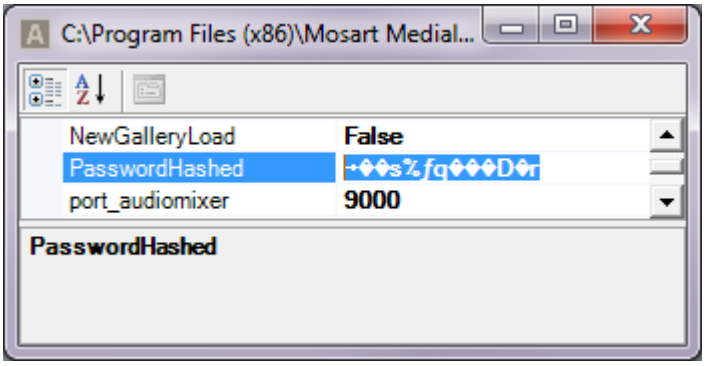
1. Set focus on AV Automation.
2. Select **Devices > Change Password**.
3. In the **Change password** dialog box, enter a new password or change an old one.



To Reset a Template Editor Password

If the password is lost, it can be reset.

1. Set focus on AV Automation.
2. Opening AV Automation Settings with **CTRL+SHIFT+S**.
3. Find the PasswordHashed entry, and remove the value.



Copyright © 2024 Vizrt

8.5 Working With Template Sets

This section contains the following topics:

- Template Set Types
  - Standalone
  - Hierarchical
- Template Set Operations
- Template Set Properties

8.5.1 Template Set Types

A Viz Mosart template and its variants are arranged in a *set*. There are two approaches to working with template sets.

- Working with Template Sets
- Working with Template Sets

Standalone

Traditionally, Viz Mosart template sets are treated individually, with no connection to each other.

For example, a template set is created for each show or program. This is usually combined with automatically selecting the template set through the associated NCS rundown. In addition, startup commands for preparing the studio (lights, sound etc) for running the show are included in the template set.

The copy contains the same templates as the original, and each template set created in this manner can then be changed without affecting the other. When working with the template set for a *new* program, (based on a copy of an existing program's template set), you make small adjustments according to the functionality needed by the program.

When this new template set is created (by copying an existing set), a copy of *all* templates and any nested sets is made, and no references are made between the original and copied templates. Each template set is classified as ‘standalone’.

Disadvantages with this approach

- Many duplicates are created. Normally only a few of the templates in the set need to be adjusted for meeting the needs of a new program.
- With an ever-growing bank of templates being created for each new show, the amount of stored templates in Viz Mosart can exceed system norms. This results in long response times when saving templates and increases the memory footprint of both the Mosart Server and Viz Mosart client workstations.
- Maintenance of templates becomes cumbersome and error prone. When a change is made in a template that is common for all template sets, the same change has to be applied individually to every template set. This is a time consuming task across numerous template sets.

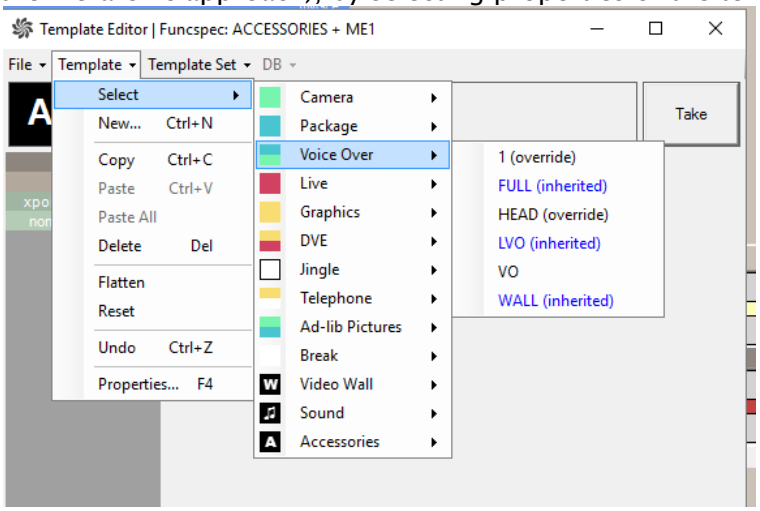
Copyright © 2024 Vizrt

Hierarchical

An alternative approach to standalone is *hierarchical* template sets. This method also enables a new template set to be created, based on the logic of an existing template set. However, to begin with, *no* templates are copied from the existing template set.
Only when a change is made to an existing template, or when a new template created, is a new template created and stored in the new template set.
The hierarchical approach can be adopted with all existing (standalone) template sets, to optimize machine capacity and introduce a more efficient workflow.

Guidelines for this approach

You can see the relationship between a templates inherited from one set to another (created using the hierarchic approach), by selecting properties of the template from the newer set.



- **Black letters without any parenthesis**: Operation is defined in the current template set only
- **Black letters with (override)**: Operation is defined in the template set but also exist in the base template set. The current template set definition is used and overrides the implementation in the base
- **Blue letters with (inherited)** : Operation is defined in a base template set and has no implementation in the current template set
- **Red letters**: An empty template, this operation is not defined.

- You can break the relation (inheritance) between two template sets
- You can base a new template set on a template set that was itself created using the hierarchic approach
- You can 'flatten' an inherited template set back to its base (original) template set. This will push any new templates back into the originating template set. After this operation, the new (inheriting) template set is deleted
- When renaming the originating template set, the relationship to any inheriting (based on) template sets still works.

Copyright © 2024 Vizrt

8.5.2 Template Set Operations

To work with a template set, from the AV Automation utility, select **Devices > Template editor**, to open the editor.
From here you have several operations:

To Add a Template Set

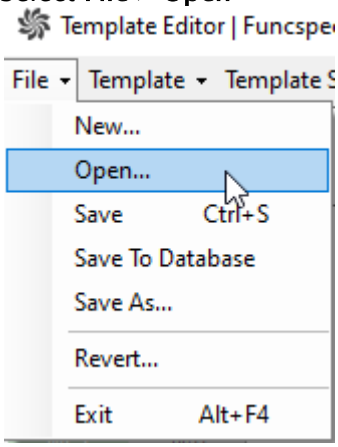
1. Select **Template Set > New**.
2. In the **Name** field, enter the name of the new set.
3. Click **OK**.

File-based Template Sets

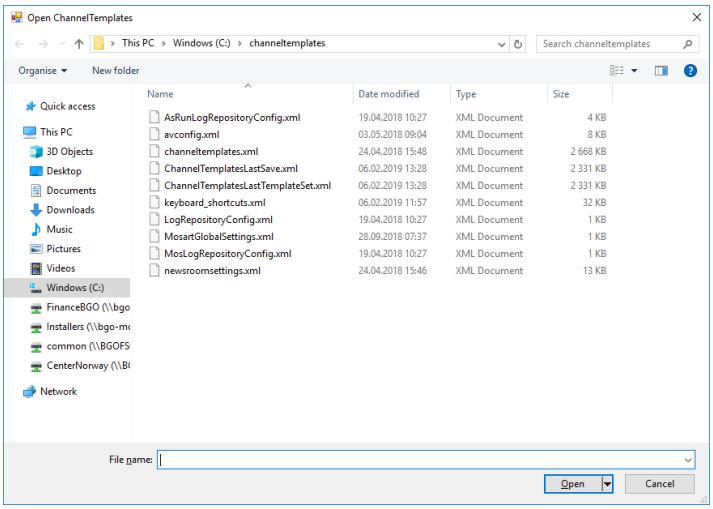
This method can be used *without* the Template Database, for example at installation / show design when trying out several solutions/designs.

You can select a ready-made template set, stored as an XML-file. This presents a convenient way of switching between sets.

1. Select **File > Open**



2. Select the preferred XML-file from the list, and click **Open**.



Copyright © 2024 Vizrt

To Open a Template Set

1. Select an existing template set with **Template set > Select**.
2. The name of the currently opened set is shown in the title of the **Template Editor**.

To Rename a Template Set

1. Select **Template set > Rename** .
2. In the **Name** field, enter a new name for the set.
3. Click **OK**.

To Delete a Template Set

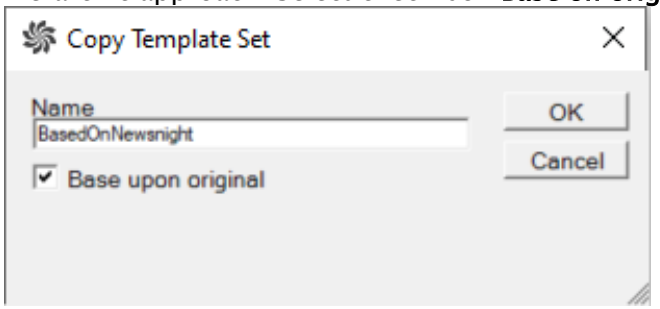
1. Select **Template set > Delete** .
2. In the dialog box, click **Yes** to confirm delete.

> **Note:** You cannot delete a template set that is the origin of any other (hierarchic) template sets. In other words, if a copy of your template set has been made using the hierarchic approach (see Working with Template Sets below), you cannot delete the originating (base) template set

To Copy a Template Set

Referring to the background discussion in Working with Template Sets above, when you copy a template set, it is more efficient to use a hierarchical approach.

1. Select **Template set > Copy**.
2. Enter the name of the new set in the text box.
3. Hierarchic approach: Select check-box **Base on original**.



Copyright © 2024 Vizrt

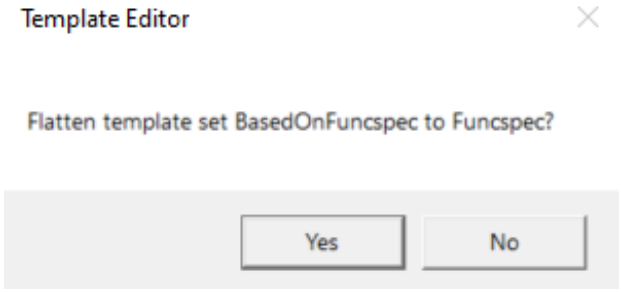
> ⚠ **Note:** If the **Base on original** check-box is *not* selected a physical copy of the entire template set including all nested templates, is created.

4. Click **OK**.

To Flatten a Template Set

You can push all templates in a new template set (created using the hierarchical approach), back to the originating template set.

1. Select **Template set > Flatten** .
2. A dialog box displays which template set the selected template set will be flattened to.



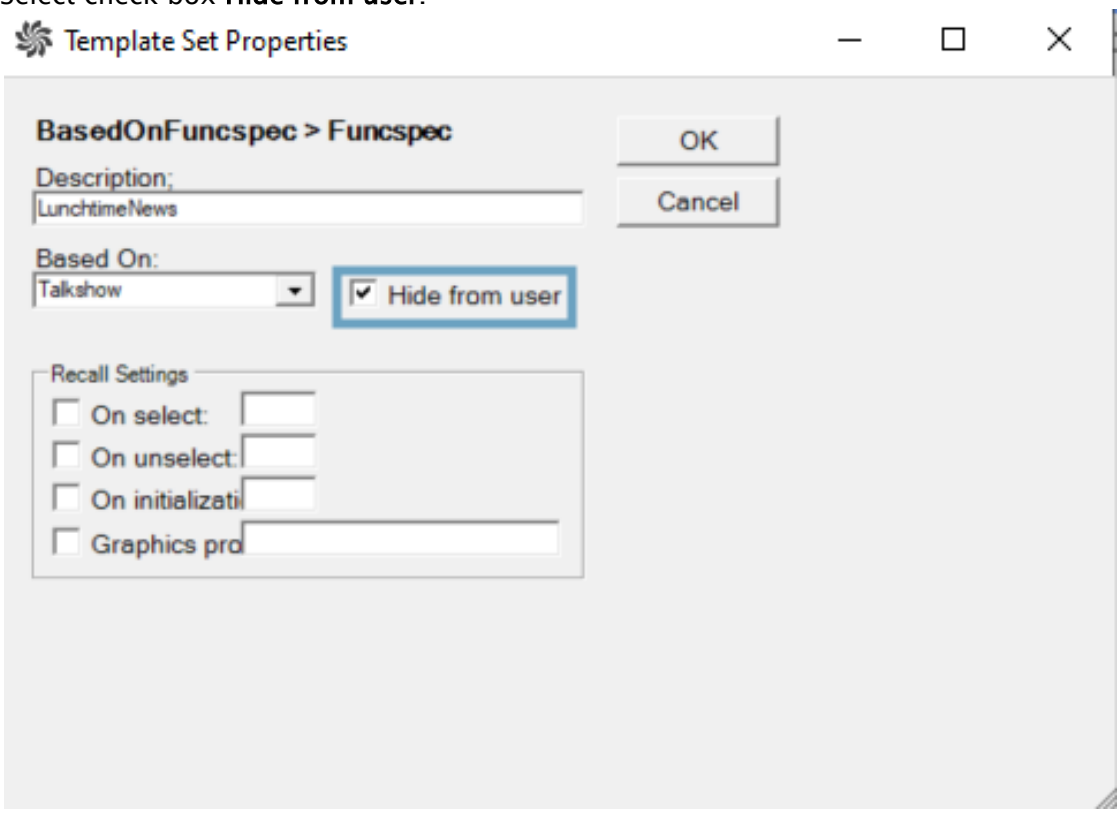
3. Click **OK**.
   All templates in the selected set are now inherited from the base set.

To Hide a Template Set

When a template set is hidden it is unavailable to users of the Viz Mosart GUI and any NCS components (ActiveX) that normally provide user access to select template sets.

1. Select **Template set > Properties** .

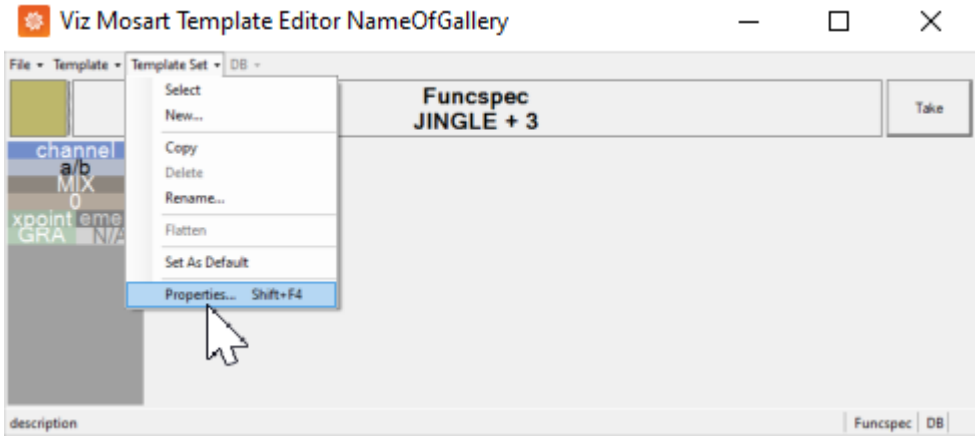
2. Select check-box **Hide from user**.



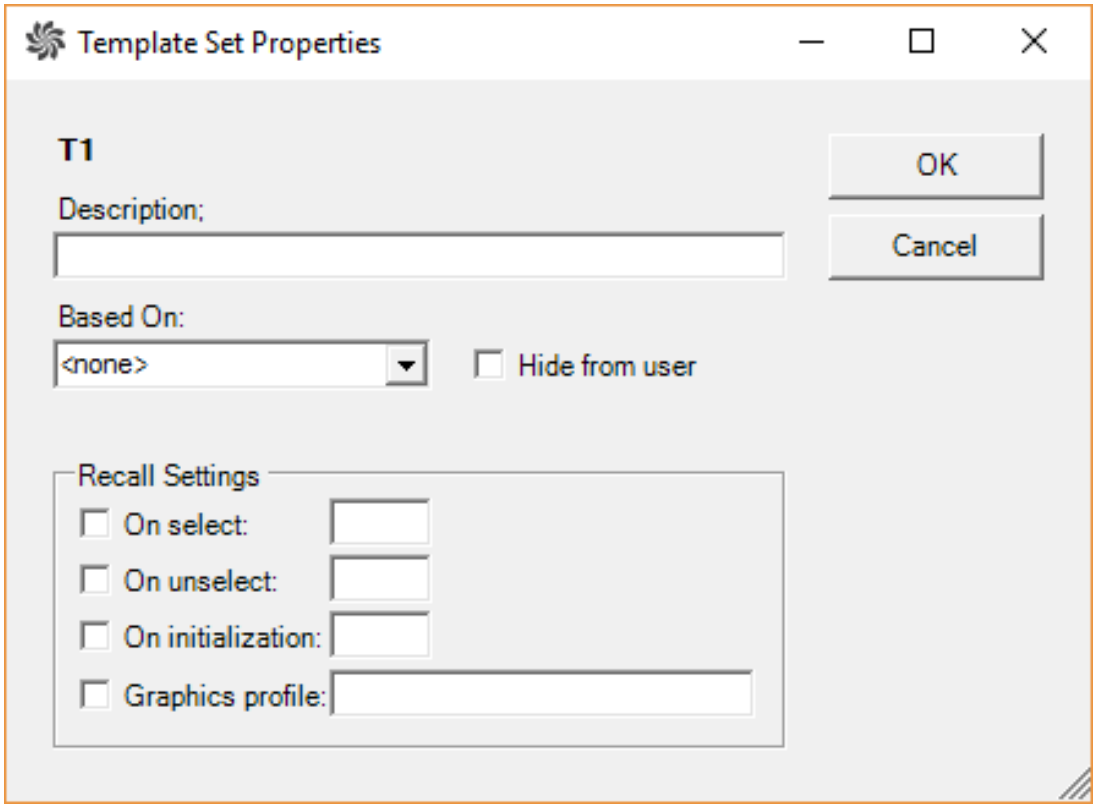
3. Click **OK**.

8.5.3 Template Set Properties

- You can manage some properties of a template set with **Template Set > Properties**.



Which displays the configurable parameters:



Item	Description
Description	The name of the template set.
Based On	Original template set, upon which this template set is a unique variant.
Hide from user	Prevent template set from appearing in user selection menus.
On select	Behavior of recall under this condition.
On unselect	Behavior of recall under this condition.
On initialization	Behavior of recall under this condition.

Copyright © 2024 Vizrt

Item	Description
Graphics profile	(When used) Profile that will be applied to the new template set.

Copyright © 2024 Vizrt

8.6 Newsroom Tags

Newsroom tags is the mechanism for *configurably* transferring data values from the NRCS to Viz Mosart. For example, a template for playing a clip can have a Newsroom tag for the *clip name*. The journalist specifies the clip name in their NRCS story item. This clip name is then automatically transferred to Viz Mosart, where the correct clip is played-out when the story item is taken.

A Newsroom tag has

- A *name*
- Optionally a *value*
- Optionally a *list* of permissible values. (If a list is given, the tag value, if any, will be in the list.) This list will be one of those defined in AV Automation, section Audio and Video Setup:
  - Video Config
  - Audio Config
  - Effects
  - Router sources
  - Router destinations.

Note: invalid characters in XML 1.0 are also invalid characters when used with ActiveX and newsroom tags. For more information, see the XML 1.0 documentation:
https://web.archive.org/web/20140228010526/http://validchar.com/d/xml10/xml10_namestart
https://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references
https://en.wikipedia.org/wiki/Valid_characters_in_XML

The basic workflow for *one* tag is:

1. In the Template editor, a Newsroom tag is added to a template field. (See Adding a Newsroom tag to your Viz Mosart Template below). The tag *name* is specified.
   If the template field has an associated list of values (as listed above), that list will be the tag list, too. If a value is given for the template field, that value will become the tag value, too.
2. In the ActiveX in the NRCS, when inserting the template into a story item in a rundown, the tag value may be changed or set (depending on whether a value was given in the previous step).
3. When the rundown is loaded into Viz Mosart, the Newsroom tag value, if any, is *automatically* assigned to the template field, overriding any value set in the Template editor.

Note: Step [1] may be repeated, adding *several* tags to a single template. Then step [**2**] may be repeated in the ActiveX, and step [**3**] will automatically be repeated, too.
If in step [**1**], the *same* tag name is specified for several template fields (*in the same template*), only *one* newsroom tag will be added, so only one value need be supplied (for this tag; there may be others) in step [**2**].
In step [**3**], this single value will be assigned to all template fields (*in that same template*) having the newsroom tag with this name.

Copyright © 2024 Vizrt

- (Optional) If the newsroom tag shows up in the GUI Program window or Preview window (see the explanations for **Program** window and **Preview** window (Key Select sub-sections) in the *Viz Mosart User Guide*, section *Main User Interface*), the value may be modified there.

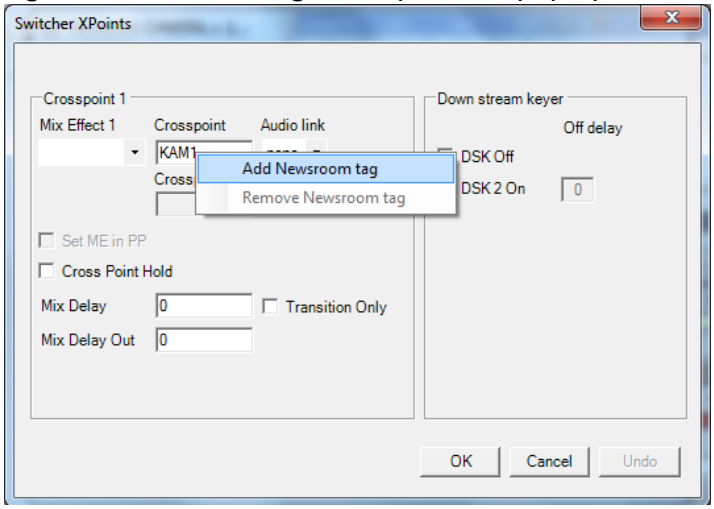
The remainder of this section deals with:

- Adding a Newsroom tag to your Viz Mosart Template
  - To Add a Newsroom tag
- Newsroom Tags Order
  - To Set newsroom tag order
- Advanced Features for Specific NRCSs
  - To Use a Column in ENPS as a Newsroom Tag
  - Inheriting Newsroom Tags (applies to MOS-based NRCS s only)
    - Keep Newsroom tag (with the same name) within a story
    - Inherit from a Newsroom tag with another name
    - Inherit with default value
    - Adding inheriting values to the lists defined in AV Automation's Audio and Video Setup
    - Examples
      - DVE Template with inheritance
      - Timeline Example

8.6.1 Adding a Newsroom tag to your Viz Mosart Template

To Add a Newsroom tag

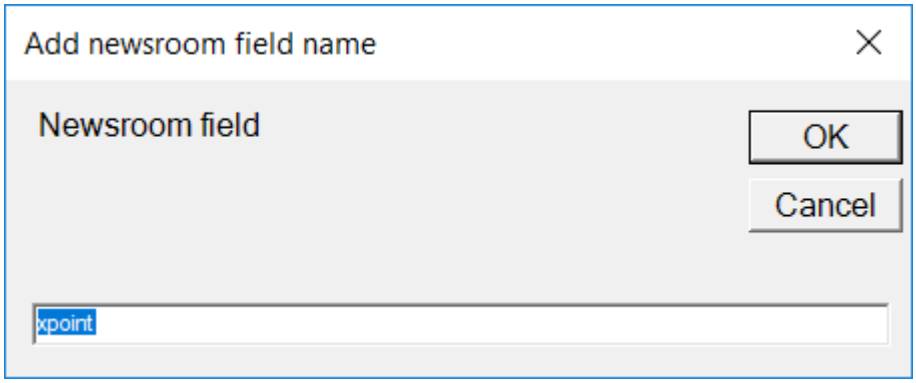
- Refer to section Working with Device Functions, which lists the properties of the device where you want to add the Newsroom tag (e.g. the 'Switcher XPoints' as shown below).
- Right-click the field (e.g. Crosspoint). A pop-up menu appears:



(If the field already has a Newsroom tag, there will instead be a menu option **Change Newsroom tag**, and the **Remove** option will be enabled. Please note that not all template

fields support Newsroom tags. When right-clicking a non-supporting field, a pop-up menu may or may not appear. If it does, it will not contain an **Add Newsroom tag** option).

- Select **Add Newsroom tag**.
  A dialog menu appears:



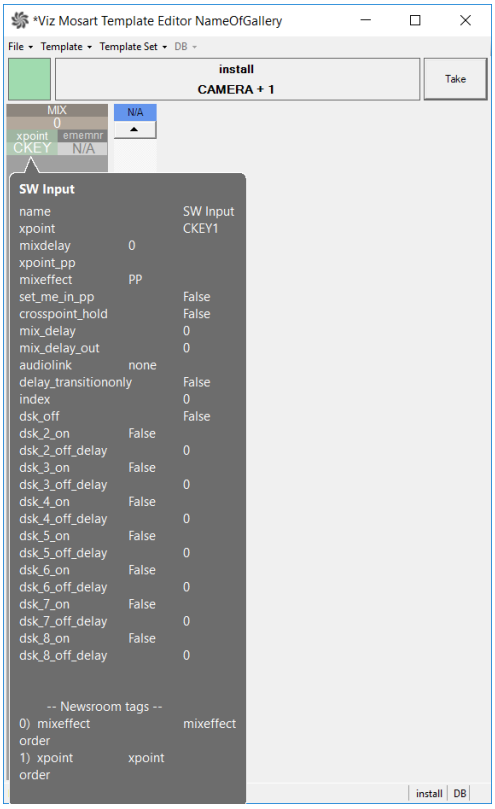
  Please note that a default name may be given.

- Enter or change the Newsroom tag name, and click **OK**.

> ⚠ **Note:** Avoid using any *special characters* in Newsroom Tag naming

- Close the device properties window (e.g. the 'Switcher XPoints' as shown above).
- To verify, hover-over the device details. The Newsroom tag(s) are presented at the bottom of the panel.

Copyright © 2024 Vizrt

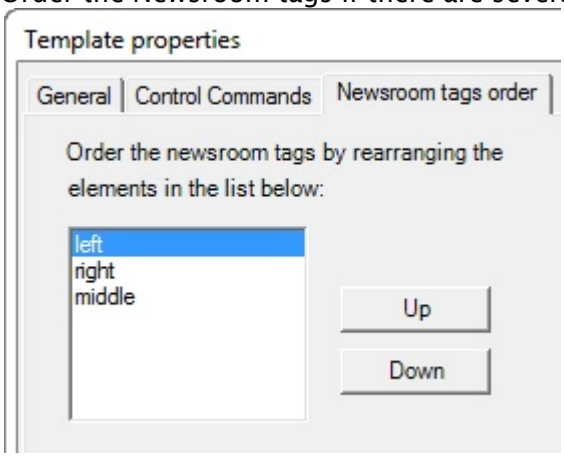


8.6.2 Newsroom Tags Order

If a template has more than one newsroom tag, the order in which the tags are shown in Viz Mosart's ActiveX, may be changed in the Template editor.

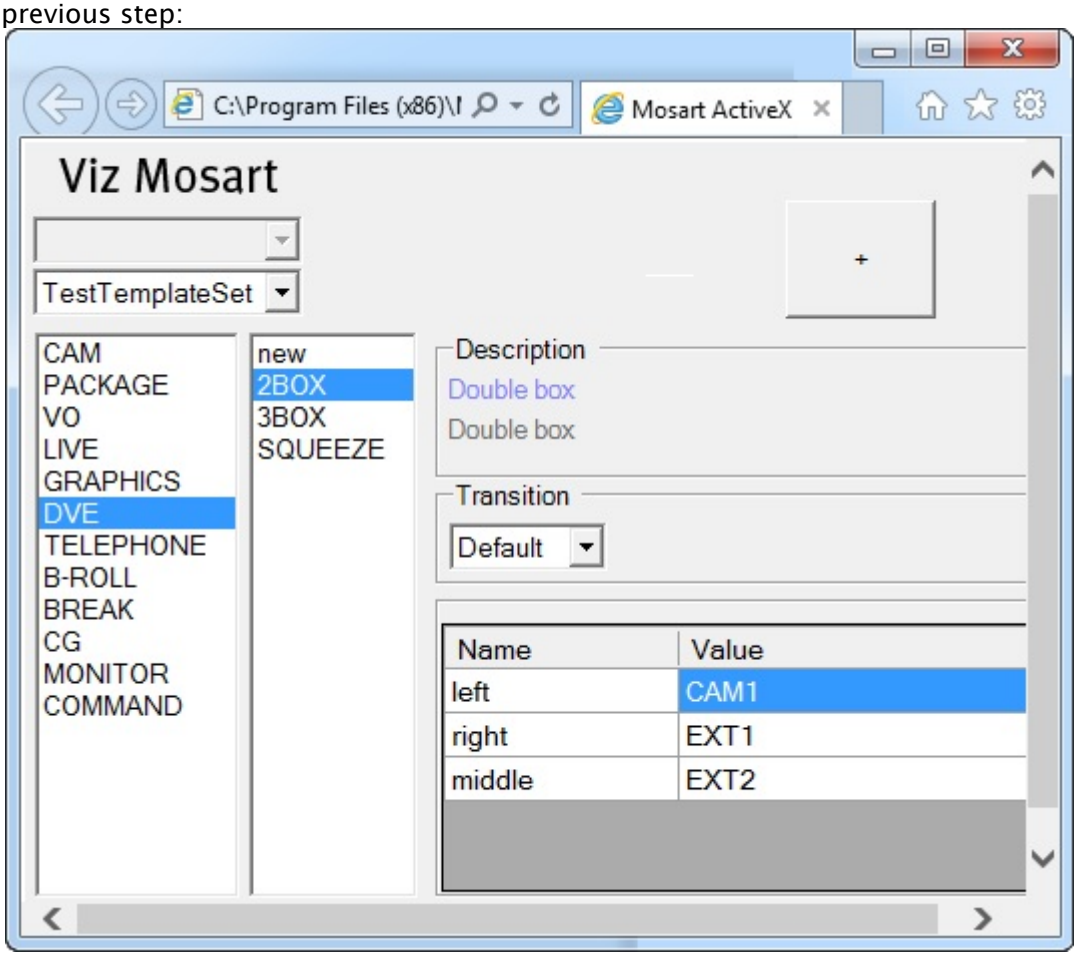
To Set newsroom tag order

1. In the **Template Editor**, open the target template
2. Open **Template Properties** and select the **Newsroom tags order** tab
3. Order the Newsroom tags if there are several. (Select a tag, and then click **Up** or **Down**).



Copyright © 2024 Vizrt

4. To verify, open the Activex in the NRCS, and observe that the tag order is as set in the previous step:



8.6.3 Advanced Features for Specific NRCSs

To Use a Column in ENPS as a Newsroom Tag

You can use a column value in ENPS as a Newsroom tag in a Viz Mosart Template.

1. Add the column in ENPS, by using the TV station-specific routine to add a column in ENPS. All the current fields are visible in **System fields->ENPS fields**.

> **Usage Notes**
>
> The id of the field and the Newsroom tag in Viz Mosart must be named *identically*.
> The naming is case-sensitive.
> Make sure that **Mos Send** is checked.
> **Type** must be *Script+RO Column*

2. Restart ENPS client to be able to add your new column in the layout.

Copyright © 2024 Vizrt

3. Add the column as a story variable in Viz Mosart config:

File Edit **Story External Metadata**

	mostagname	Mosart Action	Format Type	mos_value	action_value
	inp1	item_variable	String		inp1
	inp2	item_variable	String		inp2
	inp3	item_variable	String		inp3
	inp4	item_variable	integer		inp4

4. In Viz Mosart **External Metadata**
   a. Add the new ENPS field as, for example, an item_variable.
   b. There is also a story_valuable available, but using this action value will affect all items in the story (the Newsroom tag binds to all primaries in that story).
   c. The target here should be the *first primary item* in a story.
5. Restart all Viz Mosart server applications to apply settings.
6. Add the Newsroom tag to required template, a standard operation as described in Adding a Newsroom tag to your Viz Mosart Template.

Inheriting Newsroom Tags (applies to MOS-based NRCS s only)

For MOS-based NRCS, Viz Mosart offers a method where a Newsroom tag need only to be set once within a story.

> ⚠ **Note:** If, for ENPS, the Story scope is set to Grouped, a story in this context means the contiguous collection of GUI rows with the same story name. See Settings Editor MOS, section *Story scope*.

Story items that follow can be programmed to follow/inherit the given Newsroom tag (i.e., inheriting its value), with an optional default value.

Keep Newsroom tag (with the same name) within a story

If a Newsroom tag has the special value 'KEEP', this value will act as a placeholder in the sense that the tag will get the value of the last Newsroom tag in the story with the same name.

> ⚠ **Note:** The special value KEEP is configurable . See definition for **Newsroomtag keep keywords** in Settings Editor MOS, section *NCS Configuration*.

Inherit from a Newsroom tag with another name

A Newsroom tag may *inherit* from another tag *in the same story*. Then the inheriting tag will 'inherit' the value of the tag referred to, i.e., get the same value.

This is done in the following way: The value of one tag (the *inheriting tag*) may be given as '{*name*}', where *name* is the name of another tag (the *tag inherited from*).

Example: If there is a tag with name 'left', another tag (with some other name) may inherit the tag *left* by having this value: '{left}'.

Copyright © 2024 Vizrt

Inherit with default value

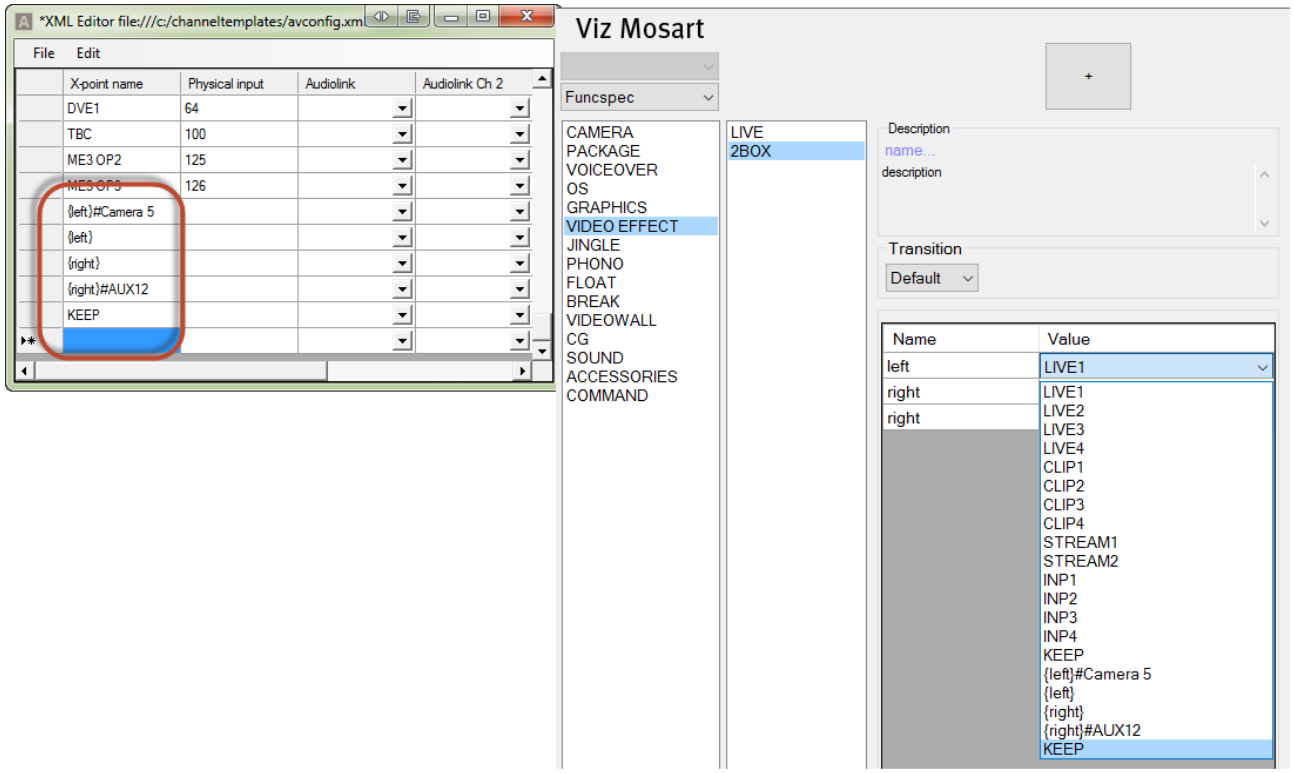
Optionally, a default value may be given, to be used when the tag inherited from has no value (in the story), or does not exist at all. The general syntax (for the value of the inheriting tag) is '{*name*} *#default*', where *name* is the tag inherited from (as above), and *default* is the default value.

Example: Continuing the example above, the inheriting tag may be given the value '{left}#OS 2'. Then, if the tag *left* has no value (or does not exist at all), the value of the inheriting tag will be 'OS 2'.

Adding inheriting values to the lists defined in AV Automation's Audio and Video Setup

As mentioned in the introduction above, a Newsroom tag may have a list being one of those defined in AV Automation's Audio and Video Setup. Special values as defined above ('KEEP'', {*name*}', and '{*name*}*#default*') may actually be added to these lists, making the available for selection both in the template editor and in the ActiveX.

The following figures five such values added to the Video Config list, and how these values appear in the ActiveX:



Explanation:

- {left}#Camera 5 – Use the value of the last Newsroom tag {left} or “Camera 5” if no such Newsroom tag exists.
- {left} - Use the value of the last Newsroom tag {left}. Nothing happens if no such Newsroom tag exists

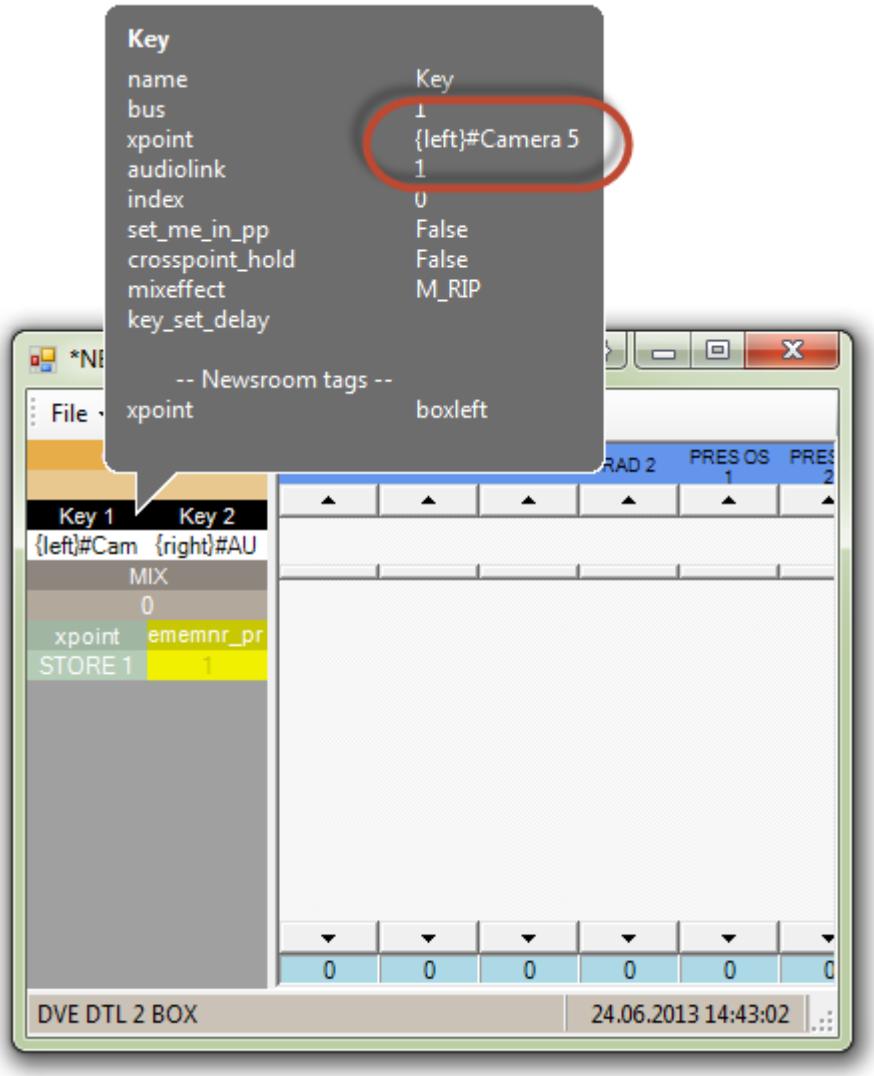
Copyright © 2024 Vizrt

- KEEP – Special configurable word that means keep existing value of the corresponding Newsroom tag. (In the ActiveX example "left")

Examples

DVE Template With Inheritance

A DVE 2BOX template where the left and right input may be inherited from sources within the same story



In this template the values for Key 1 and Key 2 are:
Key 1 = {left}#Camera 5
Key 2 = {right}#AUX 12
This means that Key 1 will be either the current value of a Newsroom tag named left within the same story or “Camera 5” if no such Newsroom tag exists.

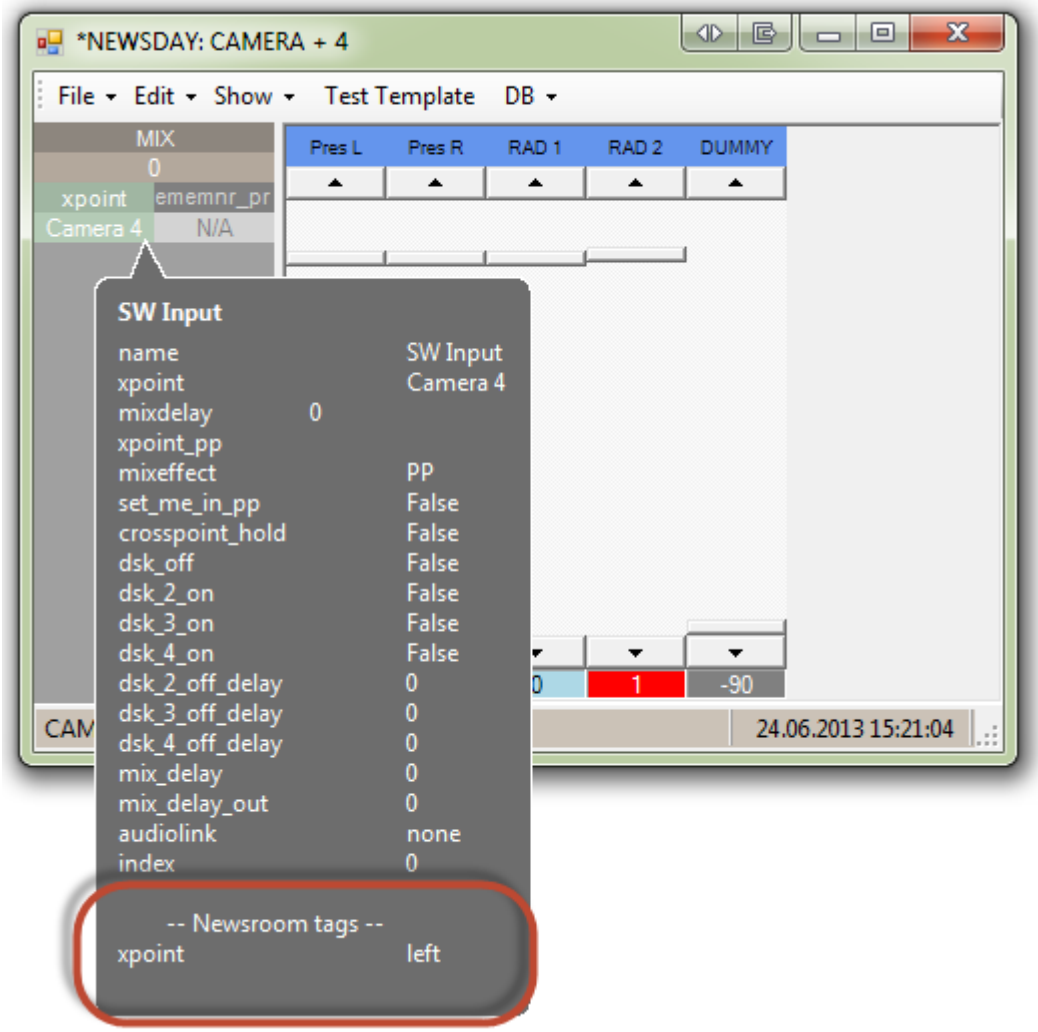
Copyright © 2024 Vizrt

Timeline Example

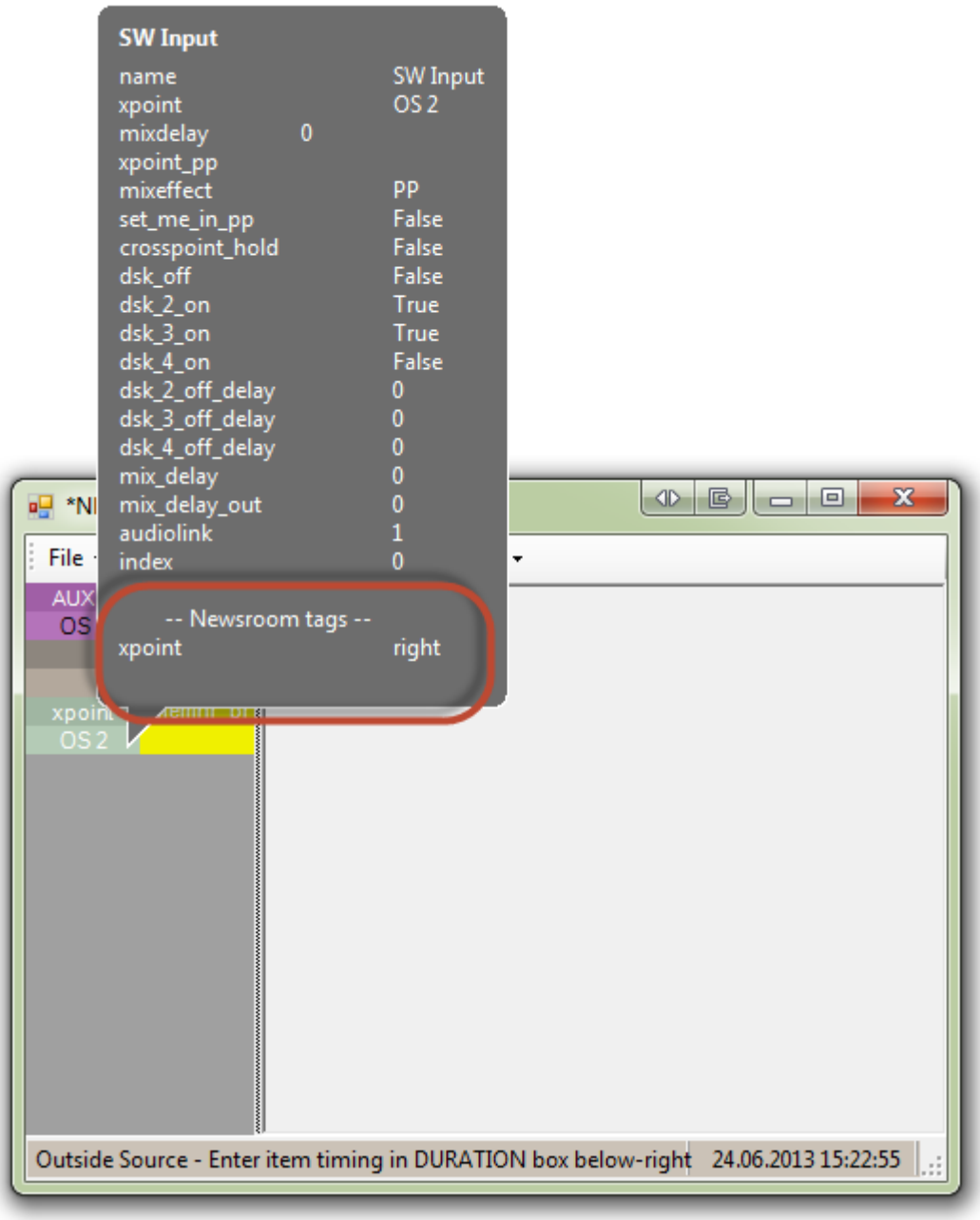
Camera+4: left = Camera 4	**LIVE+2:** right = OS 2	**DVE+DTL2BOX:** left = {left} right={right}

In the figure above three templates are placed in a timeline:

1. Camera+4: Where xpoint is set to “Camera 4” with Newsroom tag = “left”



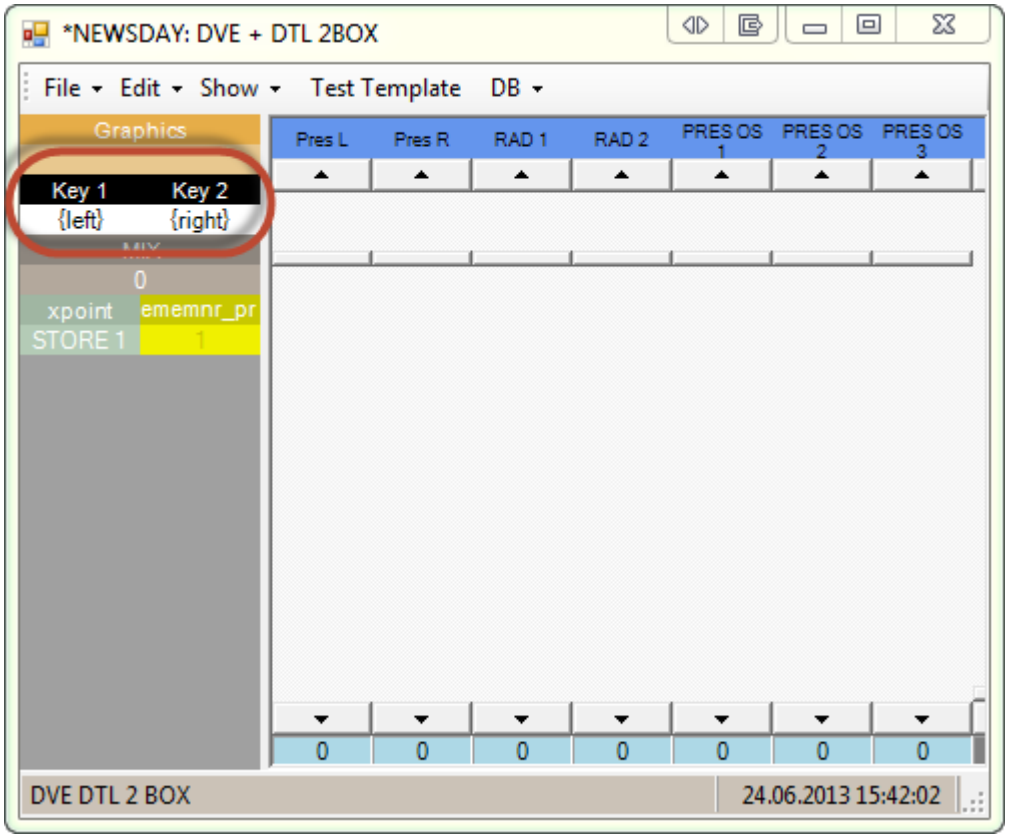
2. Live+2: Where xpoint is assigned to OS 2 with Newsroom tag = “right”



> **Note:** The Newsroom tag here is entered to make it possible for later templates in the story to inherit the value. Adding a Newsroom tag here will also expose this value to the ActiveX

3. DVE+DTL2BOX where left and right inputs are set to inherit Newsroom tag {left} and {right} respectively.

Copyright © 2024 Vizrt



In this example, the left and right inputs are inherited from previous templates in the same story exposing Newsroom tags with the same names.

> **Note:** Here, *no* default values are given., so nothing will happen if no values have been assigned to the left and right Newsroom tags

Copyright © 2024 Vizrt

8.7 Template Device Functions

This section lists a selection of device properties:

- Working with Device Functions
- Testing a Template
- Video Switcher Crosspoint
- Video Switcher Transition
- Video Switcher Register/Timeline Recall
- Macro Recall
- Video Switcher Auxiliary Bus Delegation
- Keyfill: Video Switcher Key Bus Delegation
- Video Server
- Graphics
- Video Wall Register Recall
- Audio Player
- Virtual Set
- Robotic Camera Control
- GPO
- Light Control
- Router Control
- Audio Faders
- Generic REST commands

8.7.1 Working with Device Functions

Right-click inside the device function area to open a menu containing the following options:

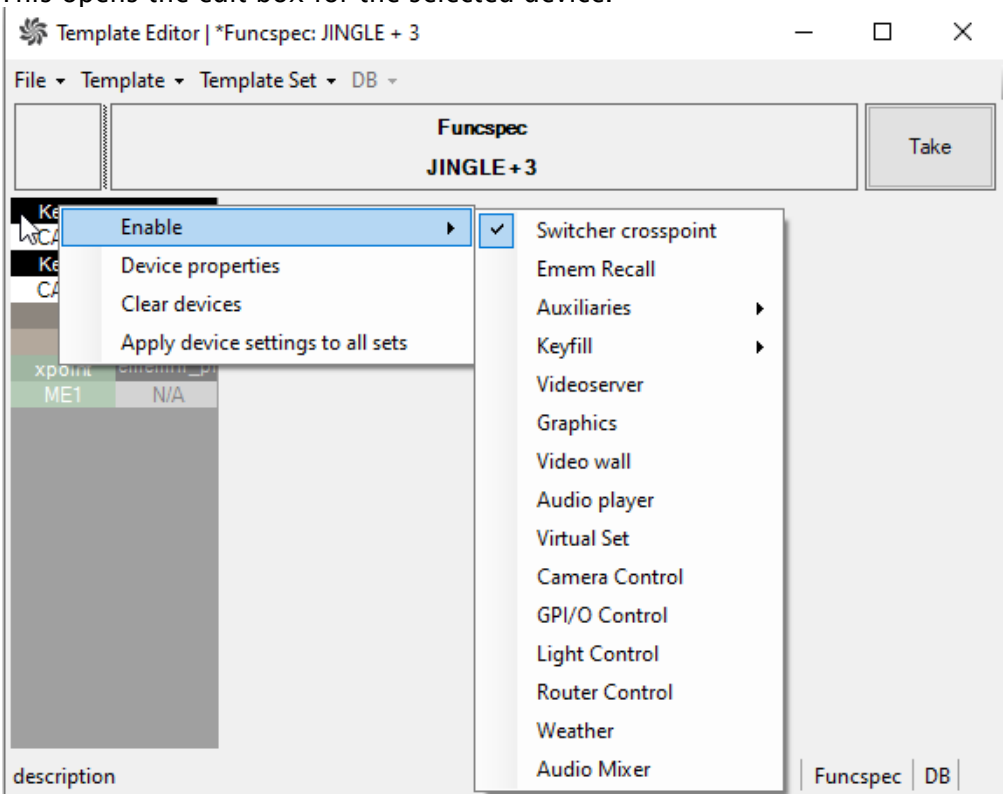
- **Enable**: These commands are described in more detail below.
- **Template properties:** Opens the current template's properties page, see Template Properties in section *Working with Templates*.
- **Clear devices:** Removes all devices from the device function menu, after confirming in a pop-up message window.
- **Apply device settings to all sets:** Applies all the device settings to all templates of same type and variant throughout all template sets.

To Enable Device Functions

1. Right-click in the device's Function area (black row).

Copyright © 2024 Vizrt

2. Select the device to configure from the **Enable** menu.
   This opens the edit box for the selected device.



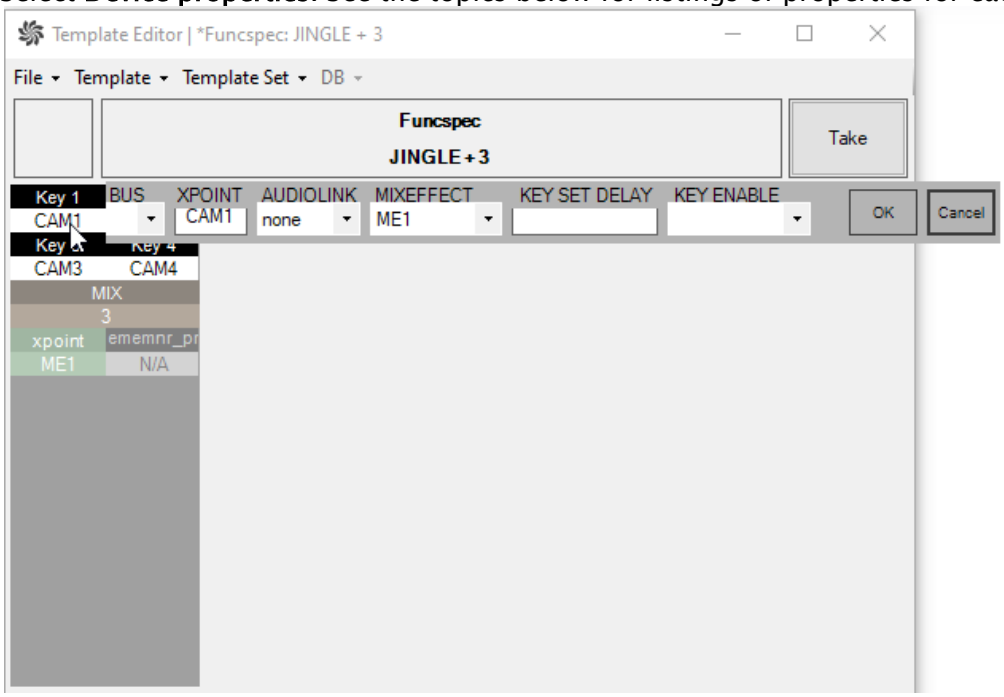
> **Note:** Depending on the protocol chosen, the caption **EMEM Recall** may alternatively be:
>
> - **Emem Recall**
> - **Dmem Recall**
> - **Snapshot Recall**
> - **Memory Recall**

To Modify Device Functions

1. Right-click in the device's Function area (black row).

Copyright © 2024 Vizrt

2. Select **Device properties**. See the topics below for listings of properties for each device.



Note: Depending on the Protocol chosen, the caption **Macro Recall** may alternatively be:
- **Custom Control**

To Remove Device Functions

1. Right click in the Device function area
2. De-select it from the **Enable** menu.

Linking Device Properties and Newsroom Tags

To Make a Device Property Editable from the NRCS

- Right-click over **Property** and select **Add newsroom tag**.
  The property is then reassigned to the value supplied from the newsroom system.

If several device properties are given to a newsroom tag with the *same* name in a template, all devices use the value given in the NRCS.

See also: Newsroom Tags.

8.7.2 Testing a Template

You can test the behavior of your template by simulating a Preview (cue) followed by a Take.

Copyright © 2024 Vizrt

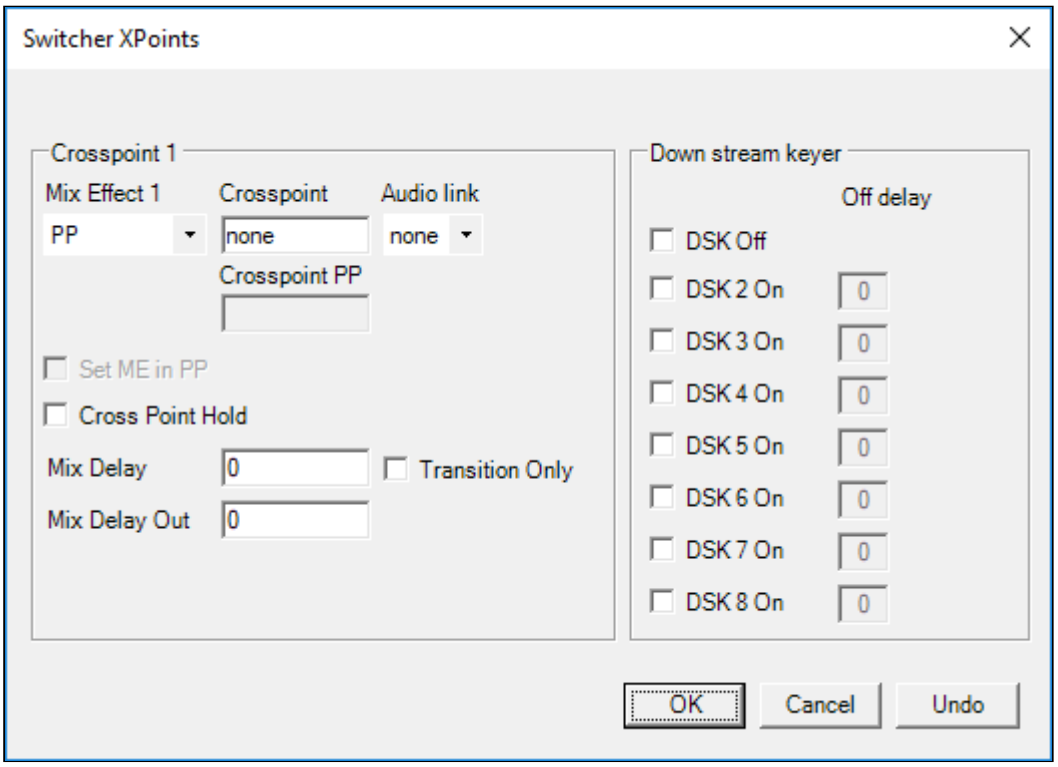
To Test a Template

- Press the **Take** button to the right of the template heading.

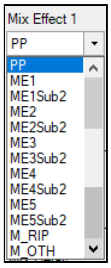
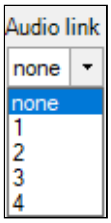
This cues the template in Preview, and then five seconds later takes it to Program with the selected transition.

8.7.3 Video Switcher Crosspoint

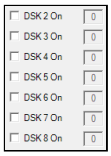
The video switcher crosspoint enables switching on the video switcher. The A or B bus assignment of the mixer effect (PP/MEx) is automatically handled by the automation.



Guidelines for Video Switcher Crosspoint Configuration

Item	Description
Mix Effect	Selects which Mix Effect (ME) bank on your vision mixer that you are currently handling. **Note** : Newsroom tag available. 
Crosspoint	Crosspoint on the video switcher. Default value: none · Enter a text value (shall be same as in the list of available x-points for Video configurations). ⚠ **Note:** Newsroom tag available.
Audio link	Default value: *none* 
Crosspoint PP	If another Mix Effect (ME) than *PP* is selected, then you can select a crosspoint for program by writing the crosspoint name in the field.
Set ME in PP	If another Mix Effect (ME) than *PP* is selected, check this box to make that ME the crosspoint for PP. (Checking this box disables XPOINT PP).
Cross Point Hold	Keeps the crosspoint on the selected **Mix Effect**.
Mix Delay	Offset, in frames, of when to perform the transition.
Mix Delay Out	Delay, in frames, of when to perform the next transition.

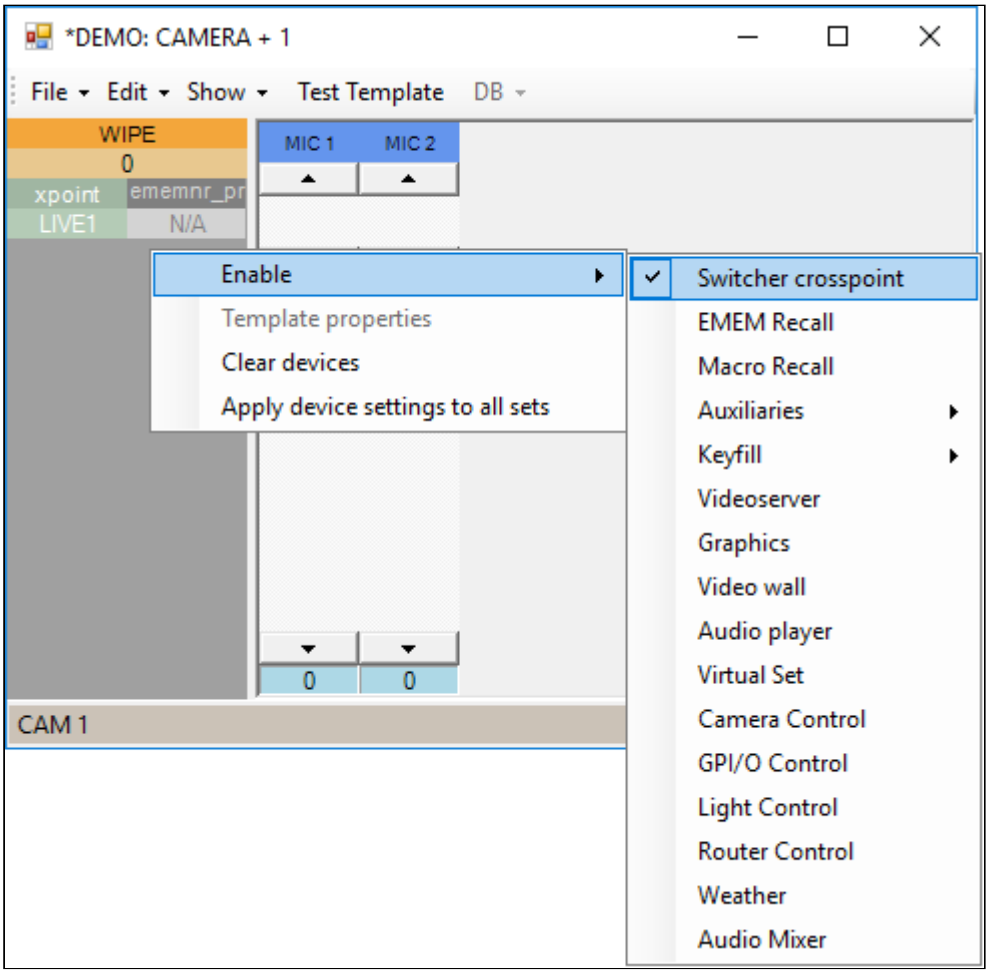
Copyright © 2024 Vizrt

Item	Description
Transition Only	Only delay the switcher transition (sub items will perform).
DSK Off	Turns off the Downstream Keyer (DSK) on the video switcher when the template is active in program. The keyer selected as DSK in the switcher device property **(Devices** > **Properties** > **Vision Mixer >** ***selected DSKNr***), will be used.
DSK 2 On, ..., **DSK 8 On**	Downstream Keyer (DSK) 2-8 on. Turns on the DSK 2-8 (PP key 2-8) on the video switcher when the template is active in program. 
Off delay	Delay in frames to set the DSK 2-8 off.

8.7.4 Video Switcher Transition

The transition device is only available when video **Switcher crosspoint** is enabled.

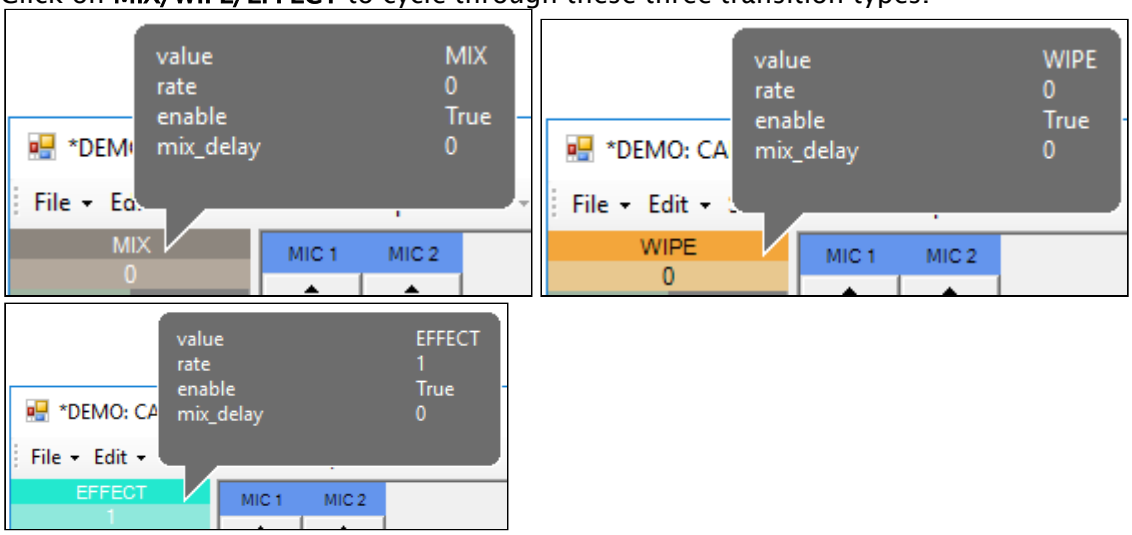
Copyright © 2024 Vizrt



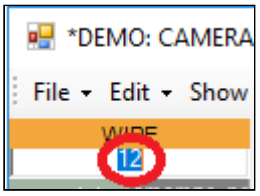
To Define Video Switcher Transitions

This procedure is based on the example in the screenshot.

1. Click on **MIX/WIPE/EFFECT** to cycle through these three transition types.



2. Click directly on the selected value (in the screenshot above, this is either *0* or *1*).
3. Enter
    a. The default duration (in frames) for the Mix/Wipe transition, or
    b. The effect number for the effect transition.



The transition is performed with the auto-transition functionality in the video switcher.

Usage and Tips

- To do a *cut*, use a MIX transition with `0` frames duration.
- To add a mix *delay*, right-click on the device (**MIX/WIPE/EFFECT**), select the number (below **Disable**), and enter another number (in frames).
  Adding a mix delay delays the start of the transition, compared to other device commands, when the template goes On Air.
- To *disable* the transition, right-click over the device and select **Disable**.

8.7.5 Video Switcher Register/Timeline Recall

Video Switcher Register/Timeline Recall (**EMEM Recall/Emem Recall/Dmem Recall/Snapshot Recall/ Memory Recall)** will recall a register and/or timeline in the video switcher.

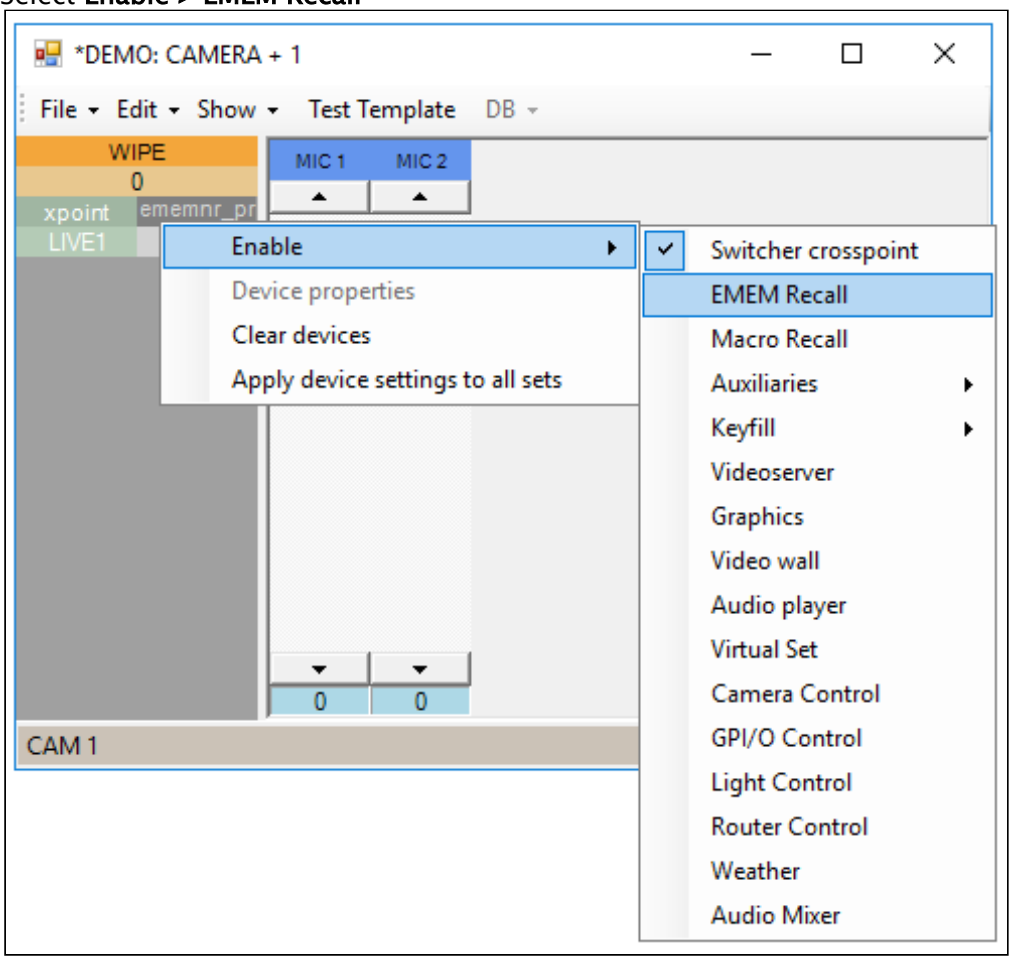
To Open the Video Switcher Register/Timeline Recall

This procedure is based on the example in the screenshot.

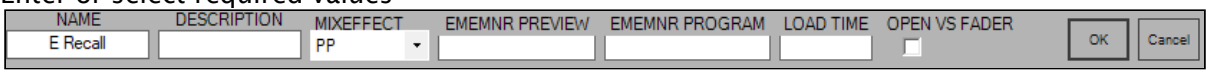
1. Open **Template Editor**.
2. Right-click over **ememnr_pr**

Copyright © 2024 Vizrt

3. Select **Enable > EMEM Recall**



4. Enter or select required values

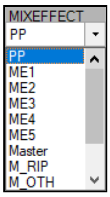


Note: If a Register recall is paired with the video switcher Template Device Functions and/or Template Device Functions, the *key/aux bus delegation* should be *disabled in the Video Switcher Register* to prevent a conflict between the stored delegation and the Viz Mosart-assigned delegation.

Guidelines for Video Switcher Register/Timeline Recall Configuration

Item	Description
NAME	Name of Register/Timeline.
DESCRIPTION	Description of Register/Timeline.

Copyright © 2024 Vizrt

Item	Description
MIXEFFECT	Select the **MIXEFFECT** (ME) from where the Register/Timeline should be recalled. Drop-down menu values are *PP* (Program), *ME1*, *ME2*, *ME3*, *ME4*, and *Master*. Default value: *Master*. (Depending on the Protocol chosen, there may be additional values).  To dynamically choose *ME1* or *ME2*, values *M_RIP* and *M_OTH* are available. ⚠ **Note:** Newsroom tag available
EMEMNR PREVIEW	Register/Timeline on the video switcher that will be recalled when the template is cued in Preview.
EMEMNR PROGRAM	Register/Timeline on the video switcher that will be recalled when the template is taken to Program.
LOAD TIME	Delay time (in frames) between when the mixer register is recalled (in Program or Preview) and when all other template instructions are executed.
OPEN VS FADER	When selected, Viz Mosart opens the fader specified in **Audio effect server**, defined in the **Audio** tab of **Device Properties** in AV Automation.

8.7.6 Macro Recall

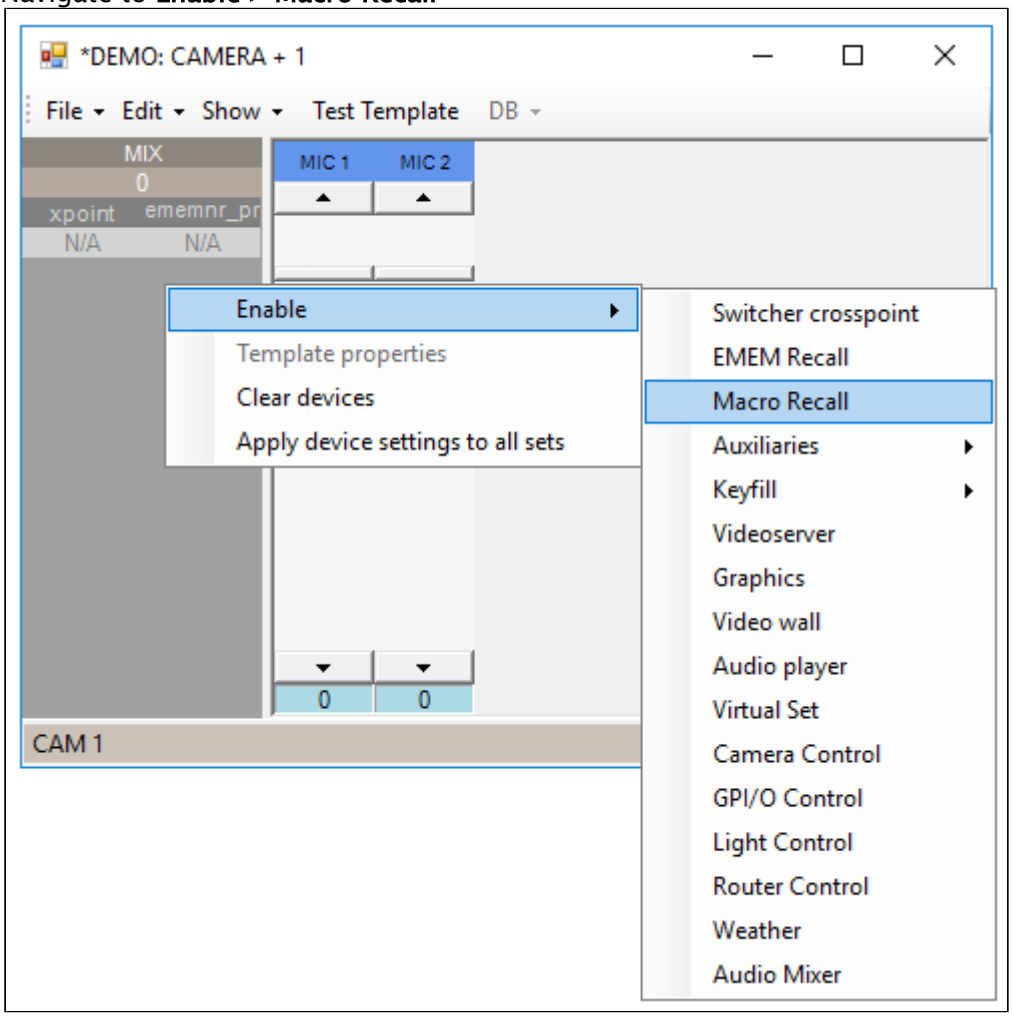
Macro Recall is available when the following vision mixer protocols (as selected in the configuration menu Vision Mixer):

- Ross Acuity
- Ross Carbonite
- GV CPL
- GV KAHUNA (Kahuna/Kula)
- SONY SERIAL TALLY
- VIZRT-IPS (NewTek/Vizrt Group IP series mixers (for example, Viz Vectar Plus, TriCaster, VMC1).

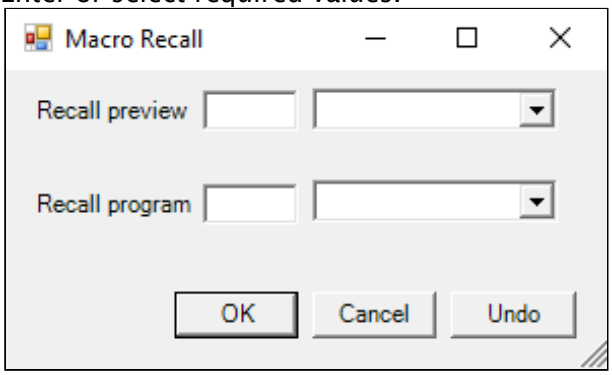
Copyright © 2024 Vizrt

To Open a Macro Recall

1. Open **Template Editor**.
2. Navigate to **Enable > Macro Recall**



3. Enter or select required values.



Copyright © 2024 Vizrt

Guidelines for Macro Recall Configuration

Item	General description	Drop-down	Text field
Recall preview	Action to be taken when entering preview mode.	See below	See below
Recall program	Action to be taken when entering program mode.	See below	See below

Drop-down Menu Alternatives

Drop-down option	Description
None	No action.
Recall	Macro is prepared (or cued).
Take	Macro is taken.
Recall and Take	Macro is prepared and taken.
Load project	Project is loaded (Kahuna/Kula only). **Warning: Load Project** could affect the performance of the switcher. Use with caution during a running show.

The format of the text fields depends on the mixer chosen:

Mixer	Text field	Remark
Acuity	One bank character and two macro characters. Example: 102 (recall custom control 02 on bank 1)	Recall (and Recall and take) Take does nothing. Note that Device is named **C** instead of **Macro Recall**.
Carbonite	One bank character and two macro characters. Example: 215 (recall custom control 15 on bank 2)	Recall (and Recall and take) Take does nothing. Note that Device is named **C** instead of **Macro Recall**.
GV CPL	Macro number (integer) in the range `1-999`.	Recall (and Recall and take) Take just takes macro alread

Copyright © 2024 Vizrt

<table>
<tr><th>Mixer</th><th>Text field</th><th>Remark</th></tr>
<tr><td>Kahuna/Kula</td><td colspan="2">The meaning and format of the text fields depend on the selected drop-down option:
<table>
<tr><th>Drop-down option</th><th>Description</th></tr>
<tr><td>Recall</td><td>Two-digit project number and three-digit macr
the format pp.mmm.</td></tr>
<tr><td>Take</td><td>As for Recall above, or three-digit macro numb</td></tr>
<tr><td>Recall and Take</td><td>As for Take above (if mmm only, the Recall is</td></tr>
<tr><td>Load project</td><td>Project number in the range <code>0-100</code> (<code>100</code> me
project).</td></tr>
</table></td></tr>
<tr><td>SONY SERIAL TALLY</td><td>Macro number (integer) in the range <code>1-250</code>.</td><td>Recall (and Recall and take)
Take just takes macro alread</td></tr>
<tr><td>VIZRT-IPS</td><td>Plays a macro having the name provided</td><td>Take only</td></tr>
</table>

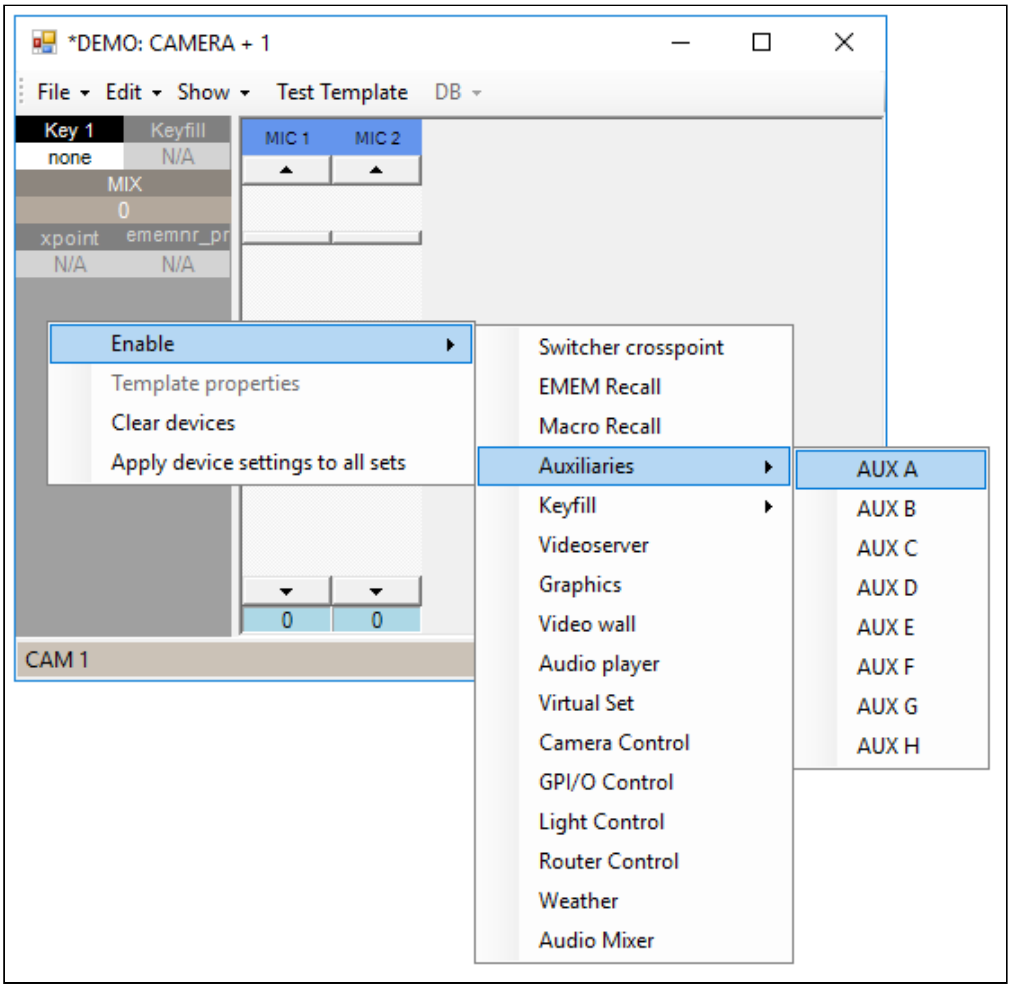
Note: Some Tricaster drivers use a non-standard macro in which the argument is the macro text itself. However, this does not apply the driver currently used by Viz Mosart.

8.7.7 Video Switcher Auxiliary Bus Delegation

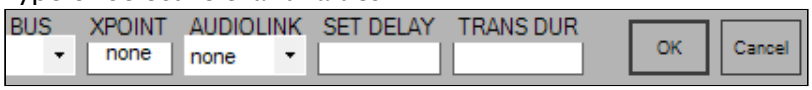
Auxiliary bus delegation enables routing of internal video switcher signals to the auxiliary outputs of your video switcher.

Note: VIZRT-IPS: Video switchers supporting VIZRT-IPS use *NDI outputs*, not auxiliaries. So on the template, the **Auxiliaries** device is instead named *NDI Router Output*.

Copyright © 2024 Vizrt

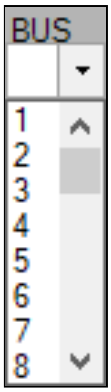
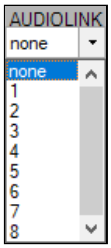


- Type or select relevant values.



Copyright © 2024 Vizrt

Guidelines for Auxiliary Bus Delegation Configuration

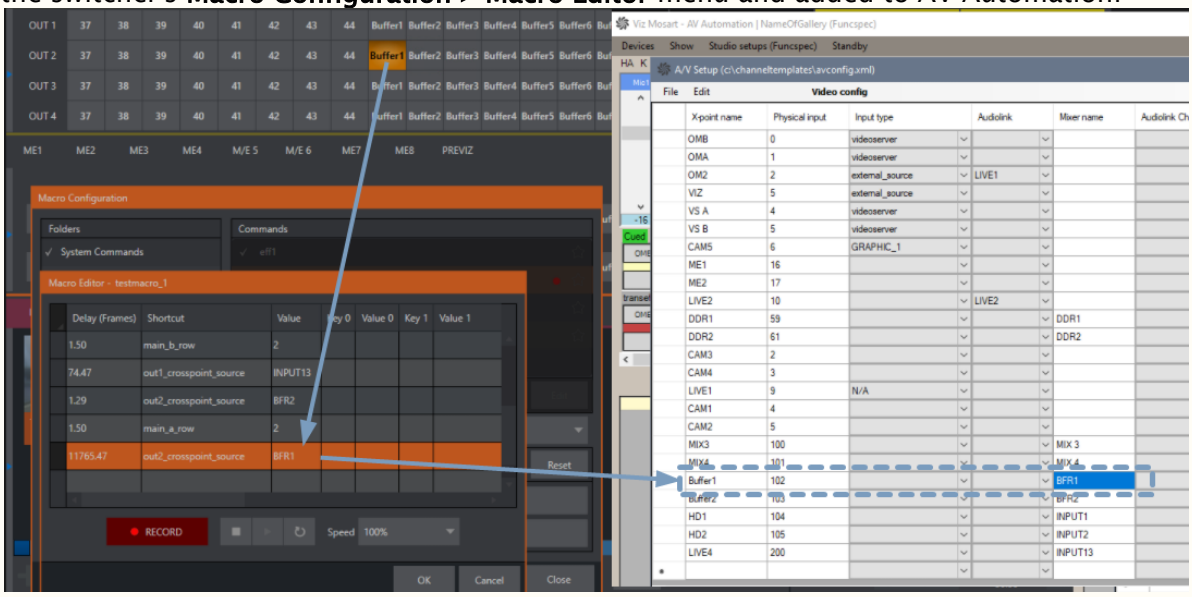
Item	Description
BUS	Selects the auxiliary to assign a new source.  ⚠ **Note:** Newsroom tag available.
XPOINT	Crosspoint on the video switcher. ⚠ **Note:** Newsroom tag available.
AUDIOLINK	Assigns a link group to this delegation. Please refer to Audio and Video Setup for the use of this feature. 
SET DELAY	Set the delay in frames of the key delegation when taking the template to Program. If the field is left blank, the key is set in Preview
TRANS DUR	If the vision mixer has the ability to mix on the AUX channels, this is where the mix rate is set, `0` is a cut

Copyright © 2024 Vizrt

Configuring an NDI Router (Viz Vectar Plus or Tricaster VMC)

⚠ **Note**: For NDI output crosspoints, you **must refer to actual crosspoint names recognized by VMC1** (and *not* any user-friendly names that would be mapped *to* the physical inputs).
These default naming on the NDI router side is typically:
- *INPUTn*
- *DDRn*
- *BFRn*
- *MIX n*.

The same naming must be used on the Viz Mosart side.
In **AV Automation > Devices > A/V Setup > Video Config > Mixer name column**, you must provide the hardcoded name associated with the user-friendly name of the crosspoint. This name is found on the switcher side.
In the example below, the value of the literal crosspoint name is the router output found in the switcher's **Macro Configuration > Macro Editor** menu and added to AV Automation:



There is a new setting **UseOutputCrosspointNames** in the IP-protocol config VizrtIpSystemsVideoSwitcherConfig.xml set by default to expect the *crosspoint name*, instead of the Viz Mosart standard of using physical input naming.

8.7.8 Keyfill: Video Switcher Key Bus Delegation

Video switcher key bus delegation enables routing of internal video switcher signals to the keyers on the different mixer effects of the switcher.

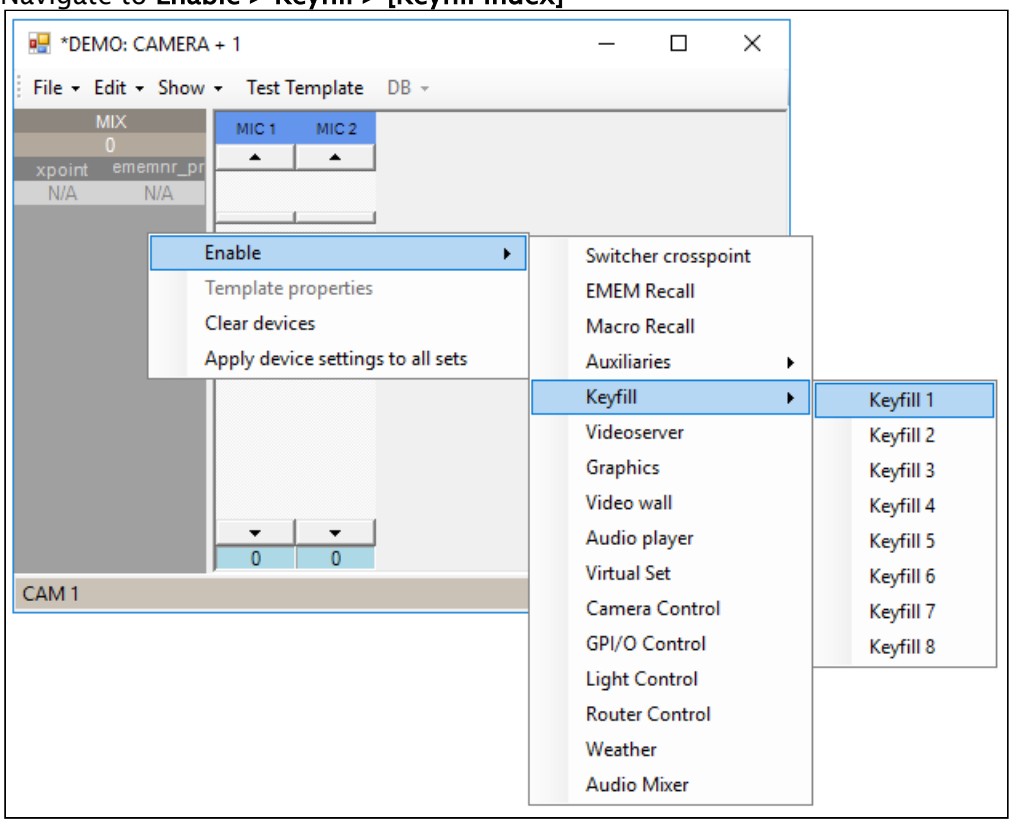
⚠ **Note**: VIZRT-IPS: For video switchers supporting M/E effects with 4 layers, the device is named *Key/Layer*.

To Enable Keyfill by Bus Delegation

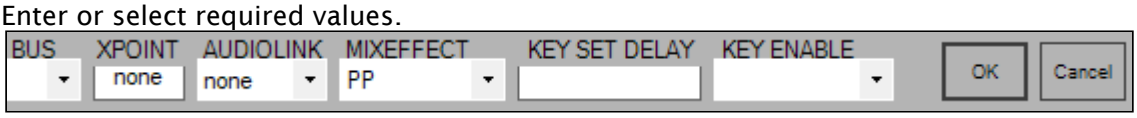
1. Open **Template Editor**.

Copyright © 2024 Vizrt

2. Navigate to **Enable > Keyfill > [Keyfill index]**



3. Enter or select required values.



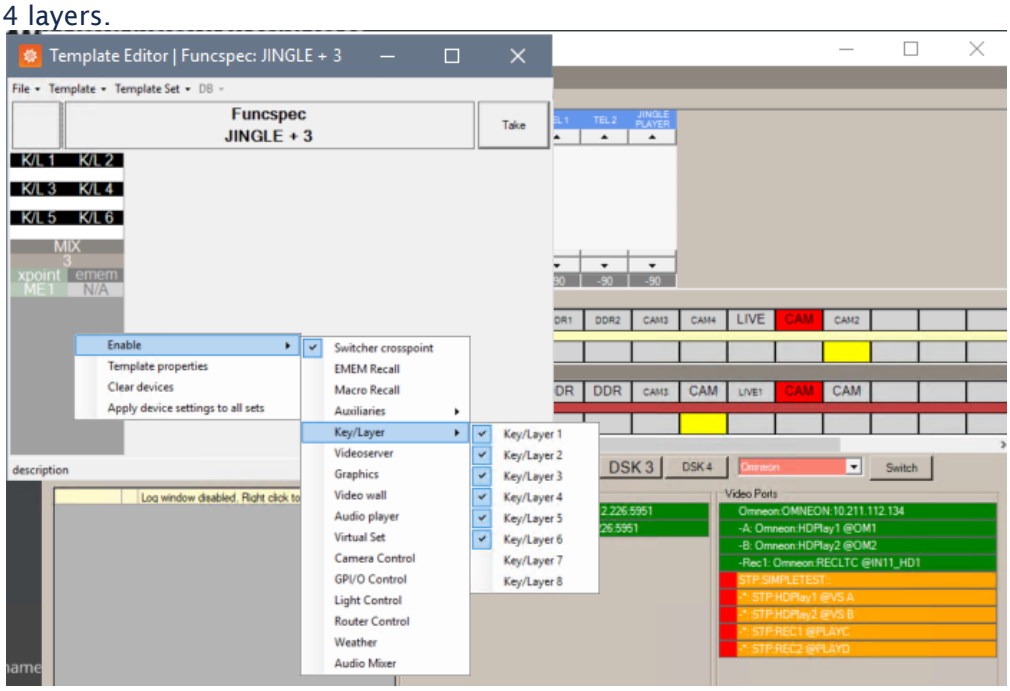
To Enable Keyfill by Bus Delegation for M/E effects with 4 layers

Devices like Viz Vectar Plus or VMC1 let you configure M/Es with effects supporting up to 4 primary sources/input layers (along with Key layers), called *A*, *B*, *C* and *D* buses.
Viz Mosart can control these buses from a template or keyboard shortcut when VIZRT-IPS is used for a vision mixer.

1. Open **Template Editor**.
2. Navigate to **Enable > KeyLayer > [KeyLayer index]**
   The Key/Layer device on a template can be used for VMC1 vision mixer to either set the crosspoint for a Key bus or to set the crosspoint for *A*, *B*, *C* and *D* buses of a M/E effect with

Copyright © 2024 Vizrt

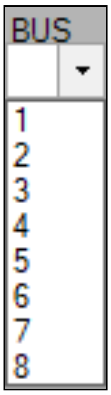
4 layers.



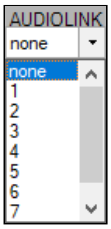
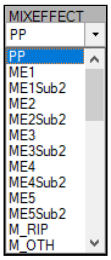
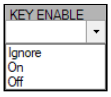
3. **Right-click > Device properties**.
   Select from the **BUS** drop-down menu:
   a. If the device is used for setting the crosspoint for a *key bus*, set the **BUS** drop-down menu to a number between *1* and *8* on the Key/Layer device properties.
   b. If the device is used for setting the crosspoint for a layer of a *M/E effect* with 4 layers, then choose one of the letters *A*, *B*, *C* and *D* from the **BUS** drop-down menu.

Copyright © 2024 Vizrt

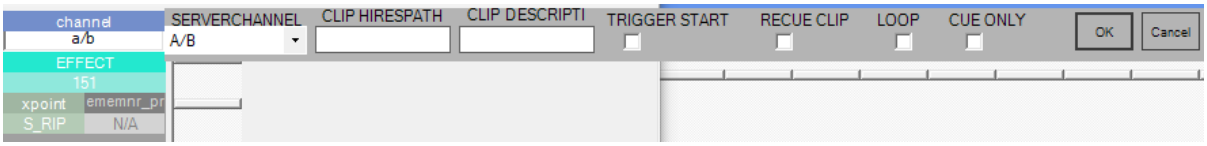
Guidelines for Keyfill Configuration

Item	Description
BUS	Selects the key bus on the selected ME.  If the video switcher supports M/Es with 4 layers (e.g. VIZRT-IPS), 4 more buses are available corresponding to each layer, A, B, C and D: **Note:** Newsroom tag available.
XPOINT	Crosspoint on the video switcher. Default value: *none* **Note:** Newsroom tag available.

Copyright © 2024 Vizrt

Item	Description
AUDIOLINK	Assigns a link group to this delegation. Please refer to Audio and Video Setup (previously *Linking video and audio sources)* for the use of this feature. 
MIXEFFECT	Select the ME where the key crosspoint should be set.  **Note:** Newsroom tag available.
KEY SET DELAY	Set the delay in frames of the key delegation when taking the template to a program. If the field is left blank the key is set in preview.
KEY ENABLE	Key enable drop-down menu.  **Note:** Newsroom tag available.

8.7.9 Video Server



Guidelines for Video Server Configuration

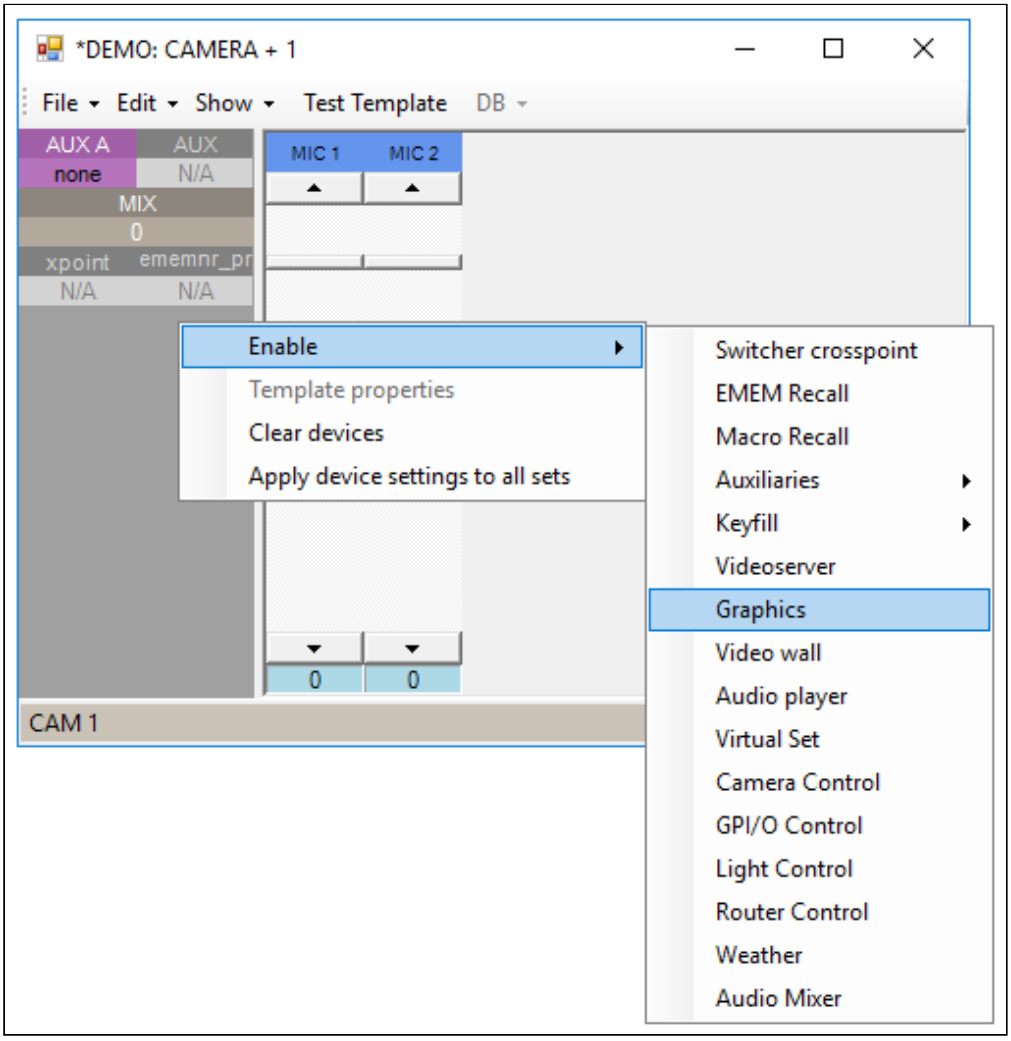
Setting	Description
SERVERCHANNEL	The Mosart video port / port group where the clip is to be played out /recorded. For recording the corresponding video port has to be part of the “Rec” port group. Port groups are created as part of the video server configuration in menu **AvAutomation > Devices > Preferences**. Configurations are described in Video Servers, section *Working with Mosart Port Configurations*. Notably, the defined virtual ports are among the available options. In addition, the non-empty groups (groups with at least one port) are available. If a group has the same name as one of its ports (as may be the case for the *Rec* and *P* groups, if defined), the *group* as such is not available, but the *port* is.
CLIP HIRESPATH	The (default) clip ID or name. ⚠ **Note:** Newsroom tag available, meaning that a value from the NRCS will override the value specified here.
CLIP DESCRIPT	You can enter a Clip description, however a clip description from the NRCS or video server overrides this value. ⚠ **Note:** Newsroom tag available.
TRIGGER START	Enables the Trigger start function from control commands, a shortcut or continue point could be set to PLAY the Clip.
RECUE CLIP	In an A/B roll situation, when several Adlibs are played back-to-back, this setting determines how a clip will be restarted after it has been paused. For example, if Clip A is paused when Clip B is taken On Air, then next time Clip A is taken on the same channel, it can either be played from where it was paused, or be re-cued (played from the beginning). This is *only* used for AdlibPix templates.
LOOP	Sets the server port to loop the clip. Beware! Not all video servers can loop.

Copyright © 2024 Vizrt

Setting	Description
CUE ONLY	Cues only the clip when the template is in Preview. The template will *not* start playing the clip when the template is taken.

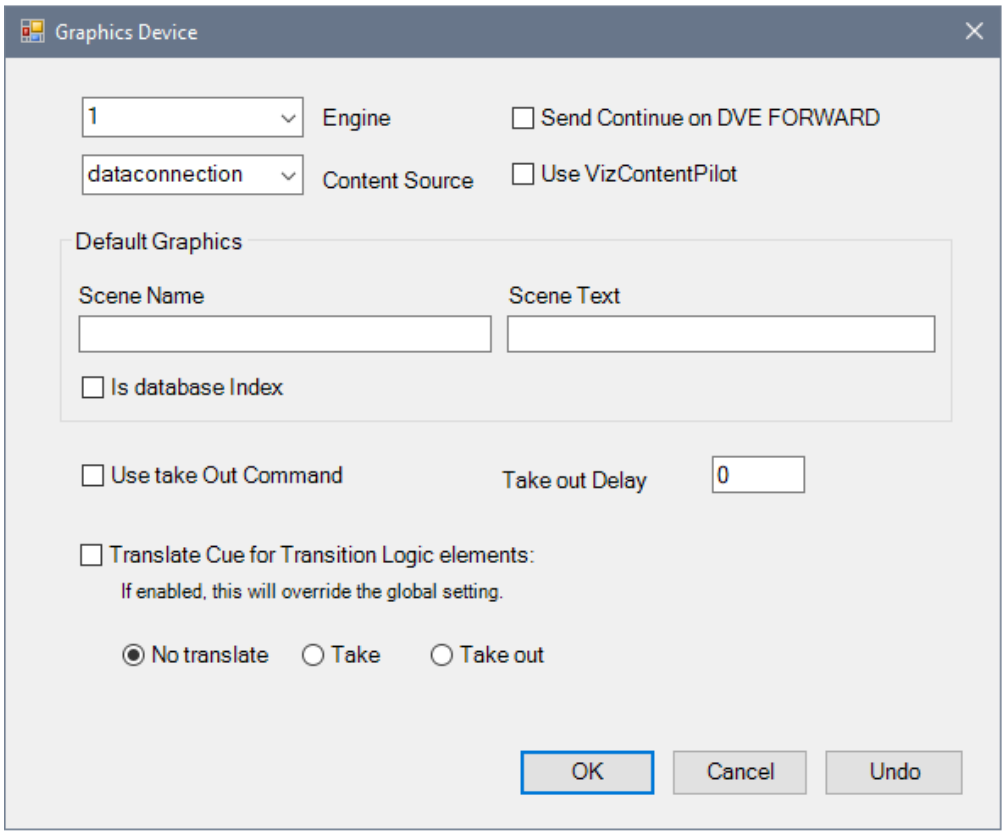
8.7.10 Graphics

The Graphics device enables recall of graphic elements.

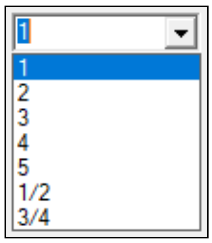
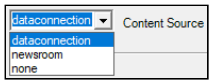


Copyright © 2024 Vizrt

- Graphic elements are cued (loaded) in Preview and played (started) when taken to Program.



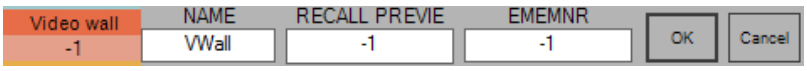
Guidelines for Graphics Configuration

Engine 	Selects the engine to operate on when using this template, based on your configuration. See section Graphics.
Content Source 	• **dataconnection:** Use content from the attached graphics database (only available when *Vizrt graphics* are used) • **newsroom:** Use content from the attached NRCS. • **none:** Use content written into the template.
Send Continue on DVE FORWARD	If selected, a continue command is sent together with a DVE Forward command.

Copyright © 2024 Vizrt

Use VizContentPilot	Enable to run this element through a Viz Pilot client. This allows for OnLiveUpdate events in Viz Pilot data elements.
Scene Name	The default scene name is used, if not overridden by the NRCS. ⚠ **Note:** Newsroom tag available.
Scene Text	The default scene text will be used, if not overridden by the NRCS. ⚠ **Note:** Newsroom tag available.
Is database Index	
Use take Out Command	If selected, a take-out command is sent when the template status changes from On Air to Off Air.
Take Out Delay	Delay of sending takeout command after template status changes from On Air to Off Air.
Translate Cue for Transition Logic elements	If active, this setting overrides the global setting with same name set in AV Automation > Devices > Properties > Graphics > General > [*VIZRT*] > Miscellaneous. This setting only applies to transition logic for Vizrt full screen graphics. The Media Sequencer ignores standard cue command for these items. · **No translate**: Send cue command to the Media Sequencer (on Media Sequencer versions below 1.20 no cue will be performed). · **Take**: Use a Take command when cuing. · **Take out**: Use a Take out command when cuing.

8.7.11 Video Wall Register Recall



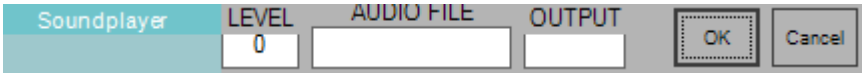
Setting	Description
NAME	Internal name of the shot.

Setting	Description
RECALL PREVIE	For WATCHOUT, see notes below. For the other brands, this is the number of the recall to be taken when the template is cued in Preview.
EMEMNR	For WATCHOUT, see notes below. For the other brands, this is the number of the preset to take when the template is taken in Program.

Notes for WATCHOUT

- The fields RECALL PREVIE and EMEMNR should contain the command or sequence of commands to be sent when the template comes in Preview or Program, respectively.
- The general format is: `[load "show"] [resetrun "timeline"]`
- Both `load` and `run` are optional, however at least one of them must be present. The show and timeline parameters may not contain double quotes.
- The `reset` value is optional. For more information see the connection string property *ResetAnyway*, described in the section Video Wall Connection Strings.
- In most cases, RECALL PREVIE contains `load "show"`, and EMEMNR contains `run "timeline"`.

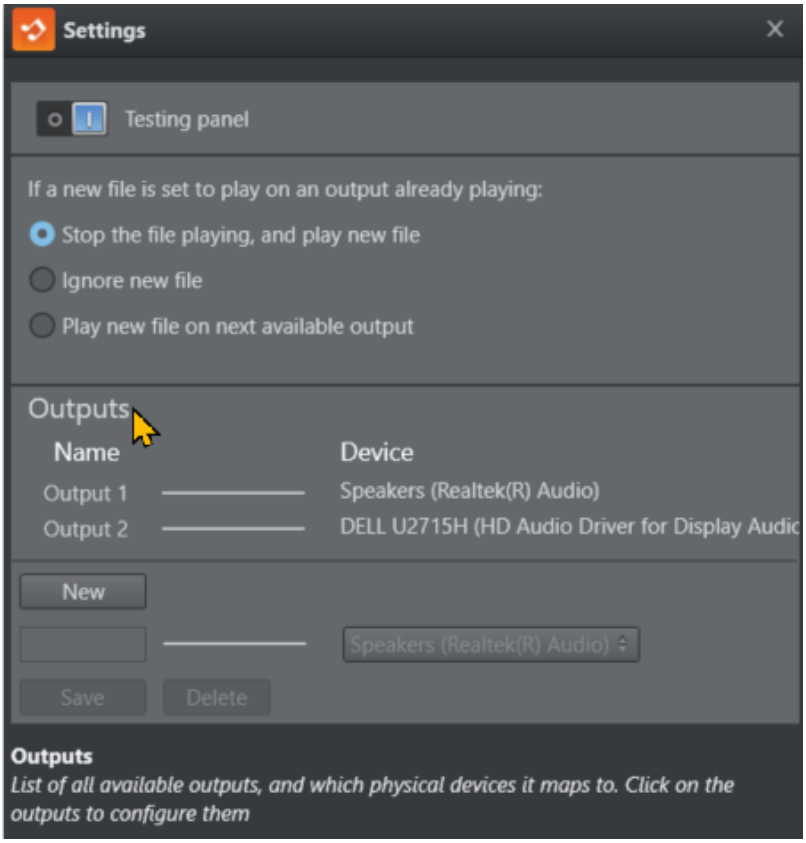
8.7.12 Audio Player



- **LEVEL:** This value *does not* affect the level. Instead, a value > -90 (the value must be an integer) *starts* playout, whereas a value <= -90 *stops* the playout.
- **AUDIO FILE:** Optionally, enter an audio file name here. If a newsroom tag is added, and a value is provided by the NRCS, that value is used.
- **OUTPUT:** (Viz Mosart Audio Player only) Choose the output on the Audio Player to be used, if nothing is inserted, the first port is used. This value is overwritten by values coming from the NRCS.
  Outputs are displayed on the Audio Player UI (see the *Audio Player* section at AV Automation

Copyright © 2024 Vizrt

Devices > Audio).

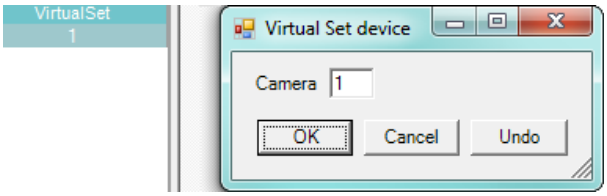


> **Note:** Newsroom tags available for both AUDIO FILE and OUTPUT.

For further details, including third party audio players, see section Audio Players.

8.7.13 Virtual Set

You can specify the camera to take in a virtual set.



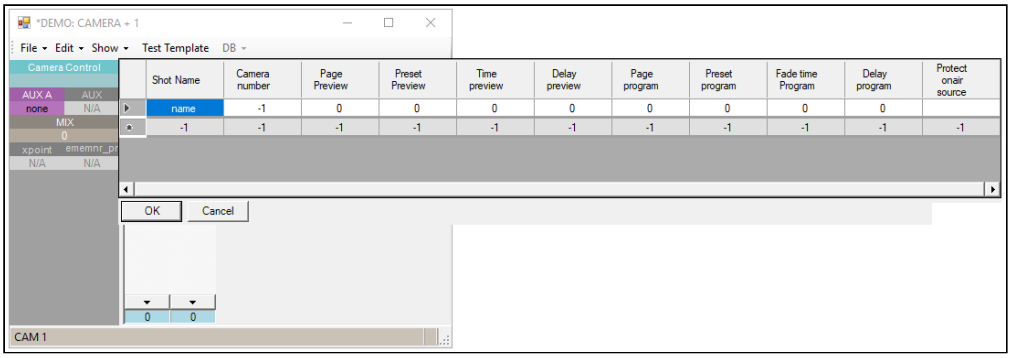
- Insert the camera number in the virtual set that is to be taken when the template is taken to Program.

> **Note**: For Vizrt's virtual set manager, Viz Arc, please refer to section Viz Arc AR Support in a Mosart Show.

Copyright © 2024 Vizrt

8.7.14 Robotic Camera Control

The camera control device allows for recalling camera shots (and pre-programmed moves) when the template is cued in preview or aired in program. The device editor supports salvo recalls of shots by adding multiple lines for each shot recall.



The Template Device Functions are described below, as well as exceptions for specific camera types:

- Guidelines for Standard Robotic Camera Control Configuration
- Cambotics Properties
- Camerobot Properties
- Cinneo System Properties
- Fx-Motion Properties
- Panasonic Properties
- Technodolly Properties
- Shotoku TR_T Properties
- Telemetrics Properties
- Vinten 200 Properties
- Electric Friends Robotic Camera

Guidelines for Standard Robotic Camera Control Configuration

Shot Name	Not used.
Camera number	• -1 : Shot is sent to all cameras that are not in standby. • >=0 : Shot is sent to the camera with the same number. The number refers to those given in the Cameras fields in Camera Robotics.

Preview	*The exact meaning and format of the* ***Preview*** *fields depend on the type of camera robotics. (See the sections below.)* These following three fields govern what happens when the template is taken in preview.
Page Preview	This field (together with Template Device Functions) determines the shot/move.
Preset Preview	This field (together with Template Device Functions) determines the shot/move.
Time preview	This field (together with Template Device Functions and Template Device Functions) determines the duration of a move.
Delay preview	Not used.
Program	*The exact meaning and format of the* ***Program*** *fields depend on the type of camera robotics. (See the sections below.)* These following three fields govern what happens when the template is taken to air.
Page program	This field (together with Template Device Functions) together determine the shot/move.
Preset program	This field (together with Template Device Functions) together determine the shot/move.
Fade time Program	This field determines the duration of a move.
Delay program	Delay of move in Program (frames). ⚠ **Note:** Only available for Vinten robotics systems
Protect onair source	Protected video switcher cross-point. If the video switcher cross-point is On Air, the shot/move recall will be ignored. For *Technodolly*, this protection can be confined to Preview and/or Program, and to Cuts and/or Moves.

Copyright © 2024 Vizrt

Cambotics Properties

- **Camera Number:** Set camera number to `1`.
- **Page Preview/ Page program:** Robotic camera number.
- **Preset Preview/ Preset program:** Shot number.
- **Time preview/ Fade time Program:** [Optional] Duration for move in deciseconds (10 deciseconds = 1 second).

> ⚠ **Note:** All parameters not defined specifically, *must* be set to `-1`.

Camerobot Properties

- **Page Preview/ Page program:** Name of shot's matrix in preview/program.
- **Preset Preview/ Preset program:** Shot's cell number in the selected matrix - the shot is adjustable from NRCS and GUI.
- **Time preview/ Fade time Program:** 0 ~ cut (fastest move) to position, ?0 ~ programmed move speed (adjustable from NRCS and GUI).

Cinneo System Properties

- **Page Preview/ Page program:** Name of the camera to recall.
- **Preset Preview/ Preset program:** Name of shot to recall - adjustable from NRCS/GUI.
- **Time preview/ Fade time Program:** Desired moving time (seconds).

Fx-Motion Properties

- **Page Preview/ Page program:** Identifies the camera to recall (identical to *Camera Number* > 0), optionally followed by :category for temporary overriding any NRCS or GUI settings.
- **Preset Preview/ Preset program:** Name of shot or move to recall.
- **Time preview/ Fade time Program:** =0 cut, <0 ~ default time to shot, >0 ~ wanted time (frames).

Panasonic Properties

- **Camera Number:** Identifies the Controller that the command is linked to.
- **Page Preview/ Page program:** Identifies the Camera Number to be controlled in Preview/ Program.
- **Preset Preview/ Preset program:** Identifies the Preset Stored Move in Preview/Program.
- **Time preview/ Fade time Program:** Sets the Timed Delay in Preview/Program. Has no effect for Panasonic TCP.

Copyright © 2024 Vizrt

Technodolly Properties

- **Page Preview/ Page program:** Identifies the camera, identical to *Camera Number* when >0.
- **Preset Preview/ Preset program:** Name of the move to recall.
- **Time preview/ Fade time Program:** =0 ~ go to start of move, ?0 ~ move.

Shotoku TR_T Properties

- **Page Preview/ Page program:** Page of shot to recall.

Telemetrics Properties

- **Page Preview/ Page program:** Preset page number to recall.
- **Preset Preview/ Preset program:** Preset number to recall.
- **Time preview/ Fade time Program:** =0 cut, <0 ~ default time to shot, >0 ~ wanted time (seconds).

Vinten 200 Properties

- **Page Preview/ Page program:** Name of the show to recall.
- **Preset Preview/ Preset program:** Name of shot or move to recall.
- **Time preview:** <0 ~ go to end position of a move, = 0 ~ cut to shot, >0 ~ go to start position of a move, or move to a shot (frames).
- **Fade time Program:** For moves, move forward or backward according to *Time preview*. For shots, = 0 ~ cut to shot, ?0 ~ move to shot (frames).

Electric Friends Robotic Camera

For Preview use:

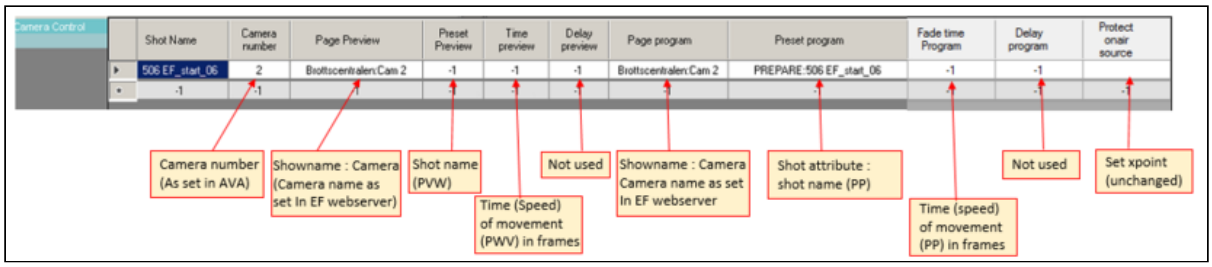
- **Page Preview:** Show and Robotic camera name separated with a colon (e.g.: *ShowName:Cam 1*).
- **Preset Preview** [Optional]: Shot name. Shot attributes: prepare, prepareplay and loop. Separated with a colon (for example, *prepare:ShotName* and prepareplay:ShotName).
- **Time Preview** [Optional]: Duration for move in frames. The time is rounded up to the nearest second.

For Program use:

- **Page Program:** Show and Robotic camera name separated with a colon (e.g.: *ShowName:Cam 1*).
- **Preset Program** [Optional]: Shot name. Shot attributes: prepare, prepareplay and loop. Separated with colon (for example, *prepare:ShotName* and prepareplay:ShotName).

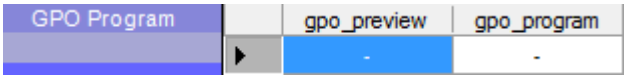
Copyright © 2024 Vizrt

- **Fade time Program** [Optional]: Duration for move in frames. The time is rounded up to the nearest second.



8.7.15 GPO

The GPO editor is used for sending GPOs to external equipment.

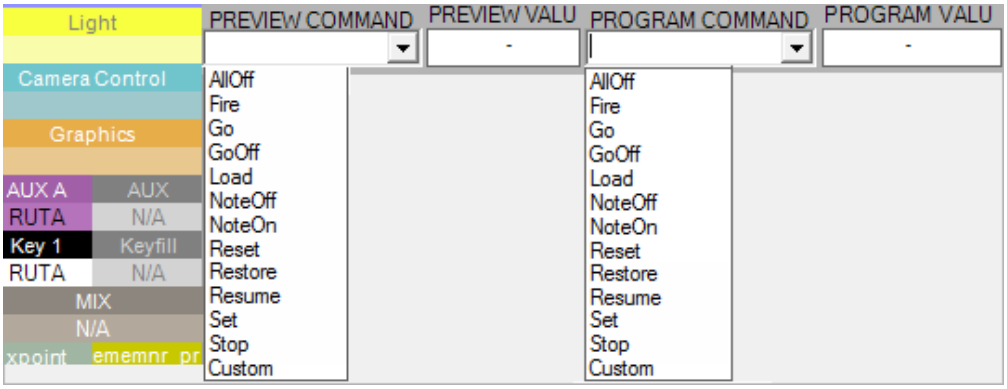


- The editor supports multiple GPO sends.
  Add a new line in the editor for multiple GPO sends.

> **Tip**: For GPO values, refer to section GPIO.

  - **gpo_preview:** GPO to send when a template is cued in Preview.
  - **gpo_program:** GPO to send when a template is taken in Program.

8.7.16 Light Control



> **Note:** The commands to use are driver-dependent and may vary.

Guidelines for Light Control Configuration

Option	Description
PREVIEW COMMAND	Command to use for the light mixer cue to be recalled when template is cued in Preview. ⚠ **Note:** Newsroom tag available.
PREVIEW VALUE	Light mixer cue to use when template is cued in Preview. ⚠ **Note:** Newsroom tag available.
PROGRAM COMMAND	Command to use for the light mixer cue to be recalled when a template is taken in Program. ⚠ **Note:** Newsroom tag available.
PROGRAM VALUE	Light mixer cue to use when a template is taken in Program. ⚠ **Note:** Newsroom tag available.

Preview Command and **Program Command** options are available from the drop-down list:

Command	Parameter count
AllOff	<none>. For details, see Template Device Functions below.
Fire	1.For details, see Template Device Functions below.
Go	cue number. For details, see Template Device Functions below.
GoOff	cue number. For details, see Template Device Functions below.
Load	cue number. For details, see Template Device Functions below.
NoteOff	numeric note # (Std. MIDI command). For details, see Template Device Functions below.
NoteOn	numeric note # (Std. MIDI command). For details, see Template Device Functions below.

Copyright © 2024 Vizrt

Command	Parameter count
Reset	<none>. For details, see Template Device Functions below.
Restore	<none>. For details, see Template Device Functions below.
Resume	variable. For details, see Template Device Functions below.

Copyright © 2024 Vizrt

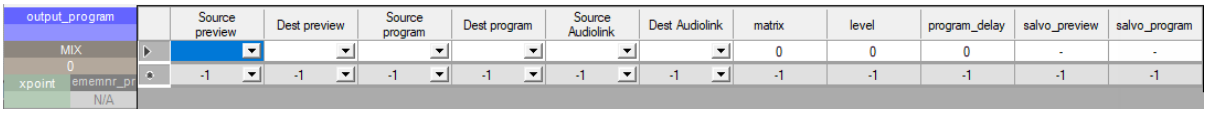
<table>
<tr><th>Command</th><th>Parameter count</th></tr>
<tr><td>Set</td><td>The value of SET is presented below in the context of Midi Show Control Specifications (RP-002-014_v1-1-1_MIDI_Show_Control_Specification_96-1-4.pdf available at https://midi.org/specifications/midi1-specifications/midi-show-control-2)
<table>
<tr><th>Hex value</th><th>Command</th><th>Number of data bytes</th><th>Recommended minimum sets</th></tr>
<tr><td>00</td><td>reserved for extensions</td><td></td><td></td></tr>
<tr><td>01</td><td>GO</td><td>variable</td><td>1 2 3</td></tr>
<tr><td>02</td><td>STOP</td><td>variable</td><td>1 2 3</td></tr>
<tr><td>03</td><td>RESUME</td><td>variable</td><td>1 2 3</td></tr>
<tr><td>04</td><td>TIMED_GO</td><td>variable</td><td>2 3</td></tr>
<tr><td>05</td><td>LOAD</td><td>variable</td><td>2 3</td></tr>
<tr><td>06</td><td>SET</td><td>4 or 9</td><td>2 3</td></tr>
<tr><td>07</td><td>FIRE</td><td>1</td><td>2 3</td></tr>
<tr><td>08</td><td>ALL_OFF</td><td>0</td><td>2 3</td></tr>
<tr><td>09</td><td>RESTORE</td><td>0</td><td>2 3</td></tr>
<tr><td>0A</td><td>RESET</td><td>0</td><td>2 3</td></tr>
<tr><td>0B</td><td>GO_OFF</td><td>variable</td><td>2 3</td></tr>
</table>
Note: You can include an optional time value (key/value[/time]), in the form: key=value@hh:mm:ss:ff
If unexpected program conditions occur, use a (dummy) time parameter.
For example, 511=22@00:00:00:00, meaning set key 511 to the value 22 at time (hh:mm:ss) 23:59:59, frame 24.</td></tr>
</table>

Copyright © 2024 Vizrt

Command	Parameter count
Stop	cue number. For details, see Template Device Functions above.
Custom	SysEx bytesSend multiple custom SysEx bytes in the format `\xnn` where `nn` is a numeric value (`0-255`). The global prefix and postfix set in the device config is used in the message (for example, `abc\x20def` will send seven bytes as a SysEx message). For details, see Template Device Functions above.
TimedGo	- (unsupported). For details, see Template Device Functions above.

8.7.17 Router Control

The router control device allows setting crosspoints for source/destination pairs when the template is both cued in Preview and aired in Program.



- Add new lines in the router device editor to support setting multiple crosspoints.

Guidelines for Router Control Configuration

Item	Description
Source preview	Source to route when a template is cued in Preview. ⚠ **Note:** Newsroom tag available.
Dest preview	Destination of routing when a template is cued in Preview.
Source program	Source to route when a template is aired in Program.
Dest program	Destination to route when a template is aired in Program.
matrix	If the router system supports multiple matrices, select the matrix to use here.

Copyright © 2024 Vizrt

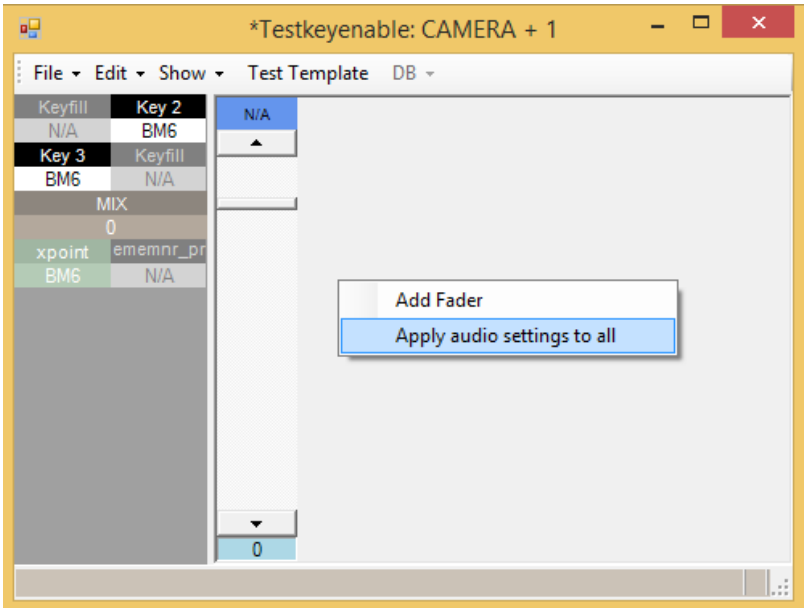
Item	Description
level	· **0:** route on all levels. · **1:** video routing only. · **2:** audio routing only. · **3:** GPI/O routing only
program_delay	Delay, in frames after the template has been taken to the air, for 'program' routing to happen.
salvo_preview	Salvo name to recall when a template is cued in Preview.
salvo_program	Salvo name to recall when a template is aired in Program.

8.7.18 Audio Faders

Adding, removing and configuring audio faders

To Add Audio Faders

1. Add audio faders to the template by right-clicking in an empty part (large grey area to the left) of the audio fader area and selecting **Add fader**.
2. Set the audio mixer crosspoint from the drop-down menu by clicking the label at the top of the fader.



Note: Clicking **Apply audio settings to all** applies all the changes to all templates of the same type and variant throughout all template sets.

To Set Audio Fader Level

- The **main level** for the fader is set by dragging the notch or clicking the arrows.
  The default fader level when adding a new fader is `0` dB.
- To set **level 2** of the fader press **SHIFT** while dragging.
  A tool-tip opens to signal the editing of the level 2 fader level.
- To set **level 3** of the fader press **SHIFT+CTRL** while dragging.
  A tool-tip opens to signal the editing of level 3 fader level.
  Level 3 is the *mute level* of the audio fader.

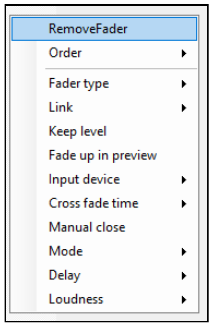
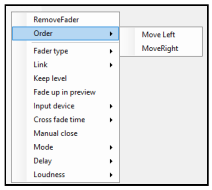
To Remove a Fader

- Right-click over the fader and select **Remove fader**.

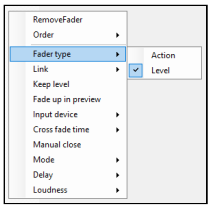
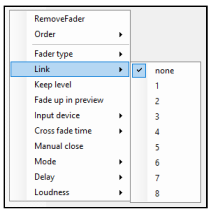
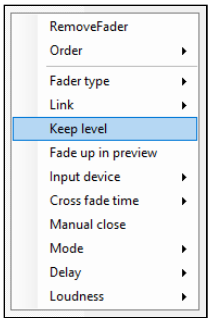
To Configure a Fader

- Right-click over the fader to open the context menu and follow the *Guidelines* below.

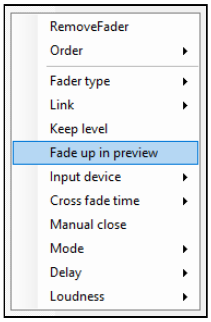
Guidelines for Audio Fader Configuration

Remove Fader	Removes the fader from the template. 
Order	Re-arranging the order: • *Move Left* • *Move Right* 

Copyright © 2024 Vizrt

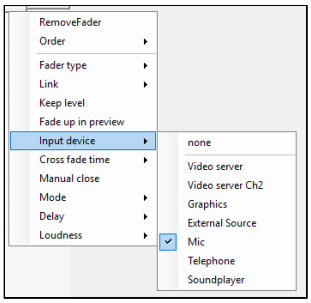
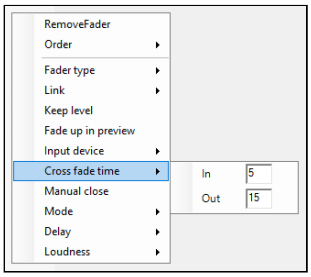
Fader type	**Action:** When a fader control is set as an action fader, the controls define whether the fader is enabled or disabled. **Level:** For normal operation. · *Action* · *Level* 
Link	Assigns a video/audio link group to this fader. · *none* · *1* · *2* · *3* · *4* · *5* · *6* · *7* · *8* 
Keep level	Keeps the fader level until manually changed. Enabled to let the automation open the audio fader, leaving it open until is is closed from another template, or the Viz Mosart client operator closes it with the **FADE_OUT_KEEPS** control command (recalled from an **Audio** keyboard shortcut). 

Copyright © 2024 Vizrt

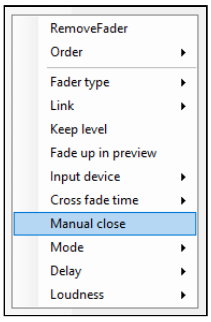
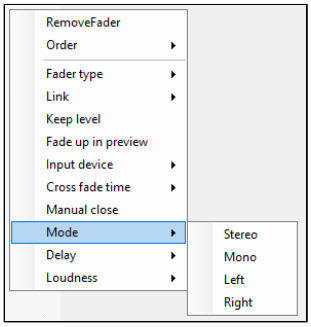
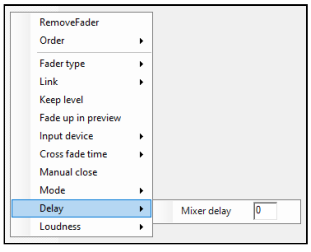
Fade up in preview	Fades up the audio in preview, when camera is in program. To let the audio fader open when the template is cued in program, set to *enabled*.  Enabling this functionality only works with specific template combinations, as shown below:

Program	Preview (with Fade up in preview enabled)	Result
CAMERA	LIVE	✓
	DVE	✓
	TELEPHONEINTERVIEW	✓
	PACKAGE	✓
	Other Templates	✗
Other Templates	PACKAGE	✓
	Other Templates	✗

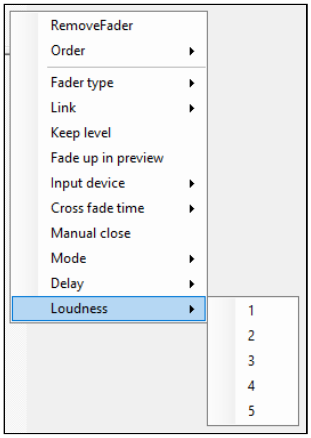
Copyright © 2024 Vizrt

Input device	Sets the type of the audio fader. The setting is used in combination with primary template type and On Air status to identify which fader should be treated as secondary audio. This special handling is applied to the types below: · **none:** No special treatment. --- · **Video server:** The fader controls a video server channel audio output. · **Video server Ch2:** The fader controls a second video server channel audio output. · **Graphics:** The fader controls audio output from the Graphics Engine. · **External source:** The fader controls an external audio source. · **Mic:** The fader controls a studio microphone. · **Telephone:** The fader controls a telephone hybrid. · **Soundplayer:** The fader controls the Mosart Audio Player output. 
Crossfade time	Sets the in and out crossfade time for the fader. · *In* · *Out* 

Copyright © 2024 Vizrt

Manual close	Select to ignore sending Close-level commands to audio mixer. When selected, sending the Take out for the fader at the end of the element is ignored; The fader has to be manually taken down. 
Mode	Only for Studer Vista and Lawo! Sends the chosen value to the mixer board for the audio fader. · *Stereo (not for LAWO EMBER+)* · *Mono (not for LAWO EMBER+)* · *Left (LAWO EMBER+: Stereo faders only)* · *Right (LAWO EMBER+: Stereo faders only)* 
Delay	Only for Studer Vista and non-Ember+ Lawo! Gives a delay value for that audio fader that will be sent to the audio board. · *Mixer delay*. Default: 0 

Copyright © 2024 Vizrt

Loudness	Sends chosen recall value to send to the Junger audio processor. · *1* · *2* · *3* · *4* · *5* 

8.7.19 Generic REST commands

See Controlling a REST-based Driver.

Copyright © 2024 Vizrt

9 Operational Examples

Viz Mosart can be configured to support multiple specialist scenarios. In this section, we list individual workflows that showcase the flexibility of Viz Mosart.

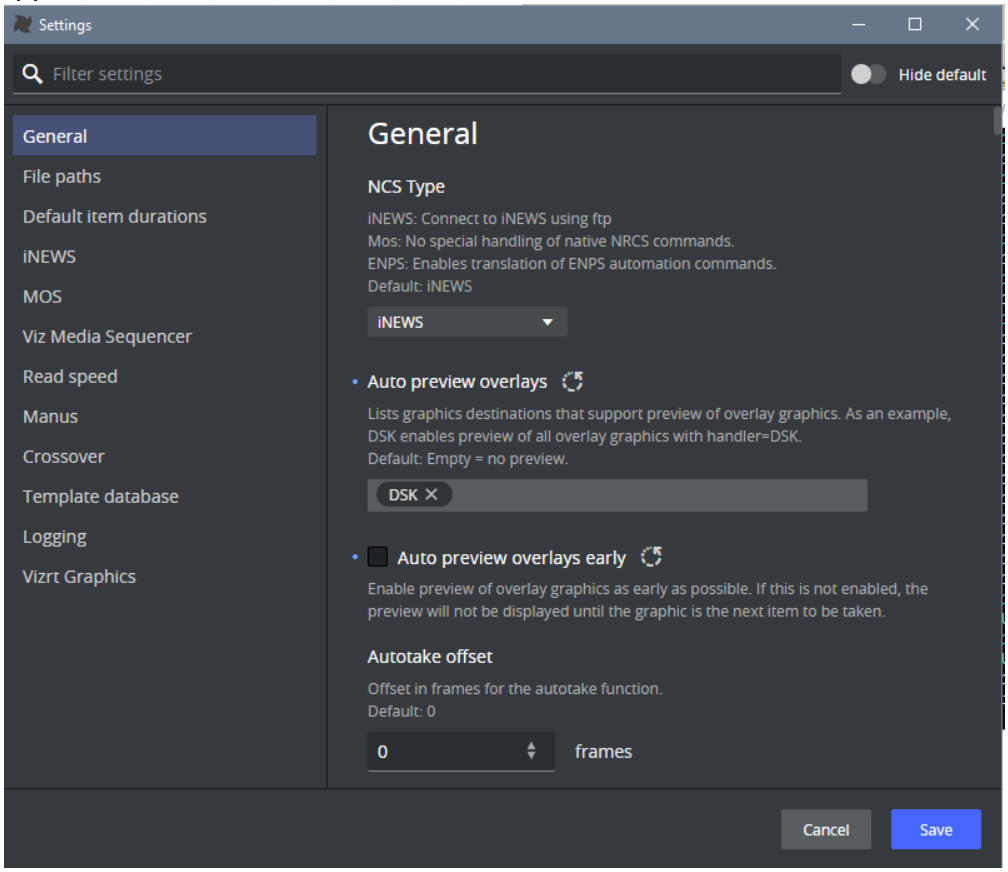
- Viz Mosart Settings and Configuration
- Story Recorder Mode
- Viz Engine as a Switcher
- Media Sequencer to Power Vizrt Graphics
- Audio Players
- Viz Arc AR Support in a Mosart Show
- Frame Accurate System Operations

Copyright © 2024 Vizrt

9.1 Viz Mosart Settings And Configuration

The Viz Mosart **Settings** menu provides a single point to configure all server-side parameters and manage entire operations from a single menu.

- In upcoming versions of Viz Mosart, settings for other Mosart Server applications and operations will be transferred to this centralized **Settings** menu, to ease the inventory of the Mosart Server.
  This will be implemented gradually, spanning several Viz Mosart versions.
- For AV Automation, all Device Properties (**Devices > Properties >[Device type** tab**]** will gradually be migrated to this centralized **Settings** menu.
- For examples of simplified operational setups, see Setup Examples below.
- The new **Settings** menu enables hot-reloading, eliminating the need for restarting the application.



9.1.1 Working with the Settings Menu

The web-based **Settings** menu is accessible from the the combined Manus Administrator that was introduced with Viz Mosart 5.3.

Settings from the legacy Manus applications (MMConsoleAdmin_2007.exe and MMConsoleAdmin_MOS.exe) are consolidated and presented in this improved menu.

Copyright © 2024 Vizrt

- Opening the Settings Menu
- Architecture
- Setup Examples

Opening the Settings Menu

To open the Settings menu

1. Ensure that your installation is using the *combined Manus*, as introduced in Viz Mosart 5.3.
2. Either:
    a. In AV Automation, navigate to **Devices > Settings**, or press **Ctrl**+**Shift**+**P**
    or
    b. In the Manus Administrator console window, type typing *settings.*

Architecture

- Settings are stored in the Mosart Server's local file system, specifically under C:\ProgramData\Mosart Medialab\ConfigurationFiles\manus_settings.json.
- The new settings editor uses Microsoft Edge WebView2 as the browser.
  Please refer to the Prerequisites page during installation for more information.

Setup Examples

The following operations/configurations, managed from the **Settings** menu are presented in detail:

- Settings Editor iNews
- Settings Editor MOS
- Generic REST Drivers
- Vizrt Graphics

Copyright © 2024 Vizrt

9.1.2 Generic REST Drivers

Working with Generic REST Drivers

Viz Mosart offers an easy to create, generic driver which is able to send REST commands to an external device. Once created, the driver's features can be controlled from a standard Mosart template, as described in section Controlling a REST-based Driver.

Key features of Viz Mosart Generic REST drivers:

- **Device configuration**: Ability to configure multiple devices that are providing REST API services.
- **Authentication support**: Options for basic, digest, and API key authentication methods.
- **Connection status monitoring**: Ability to monitor the connection status of the device.
- **Heartbeat**: Integrated heartbeat mechanism for consistent device status checks.
- **Standby mode**: Facility to set devices in standby mode.
- **Template-based command execution**: Options to configure and execute commands on cue, take or take-out.
- **HTTP command requests**: Support for GET, POST, and PUT HTTP request methods.
- **Integration with Newsroom Systems**: Enables the use of newsroom tags in request commands. This allows journalists to overwrite with content from Newsroom System.
- **Directtakes functionality**: Execute directtakes with REST commands for a configured device from templates or keyboard shortcuts in the Viz Mosart UI.

Out-of-the-box features available with a Generic REST driver:

- **Redundancy** using a reverse proxy such as Caddy or HAProxy.
- **Control a Mosart Server from a template** using the Mosart Remote Control REST API instead to use GPI control commands.
- **Control other Mosart servers** from a Mosart server using the Mosart Remote Control REST API.

This section presents the following topics:

- Setting up and Connecting to a REST-based Device
- Device Management
- Controlling a REST-based Driver

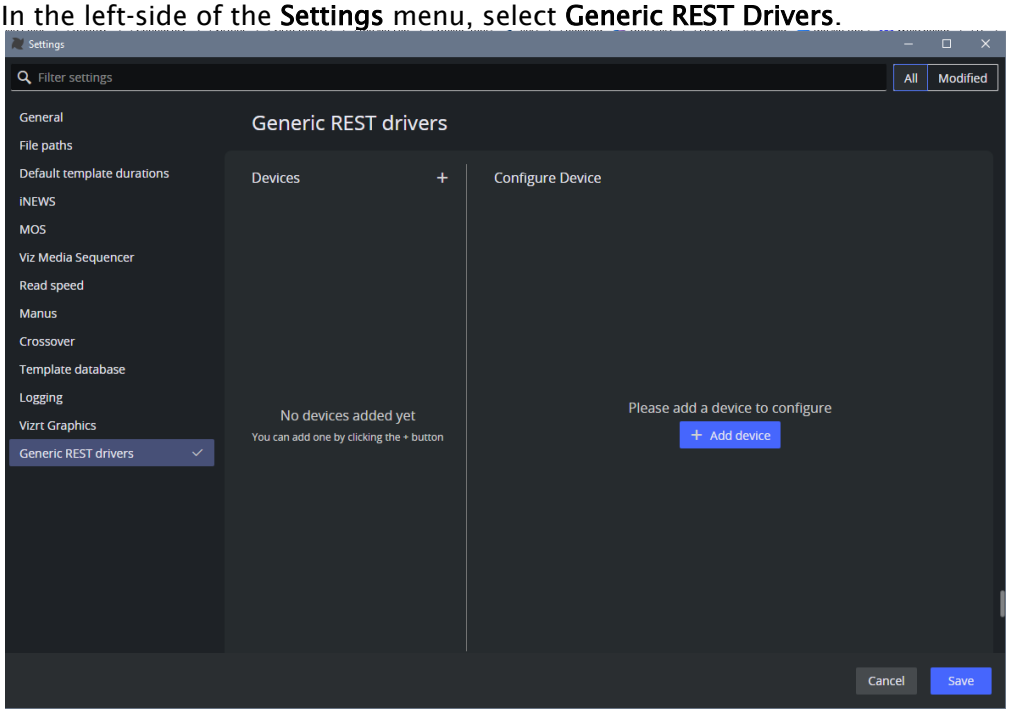
Setting up and Connecting to a REST-based Device

You construct your REST-based driver from the **Settings** menu in the combined Mosart Administrator utility.

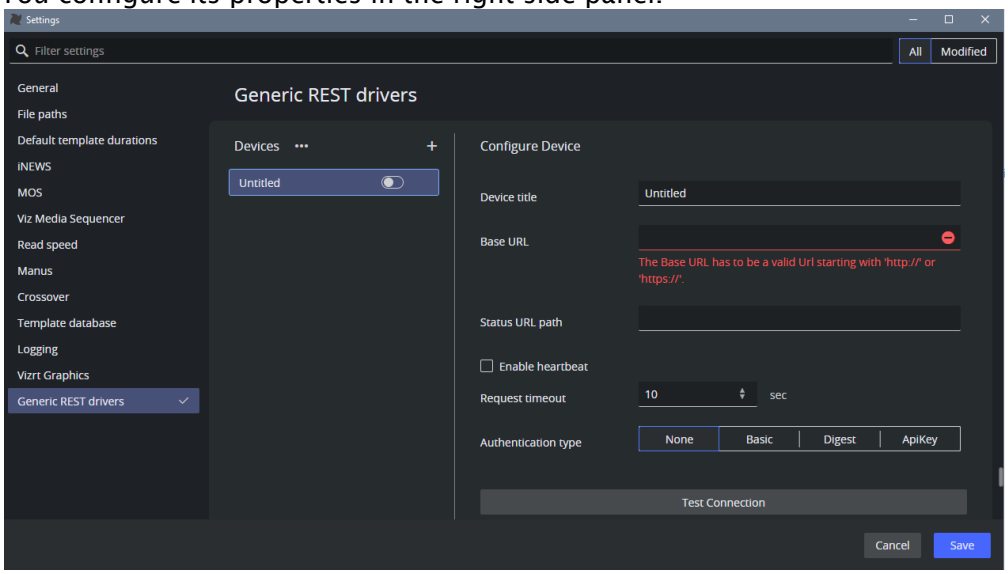
1. Open the **Settings** window. There are two methods:
   a. From the Manus Administrator console, type *settings*.
   b. From AV Automation menu bar, navigate to **Devices > Settings** (**Ctrl +Shift +P**).

Copyright © 2024 Vizrt

2. In the left-side of the **Settings** menu, select **Generic REST Drivers**.



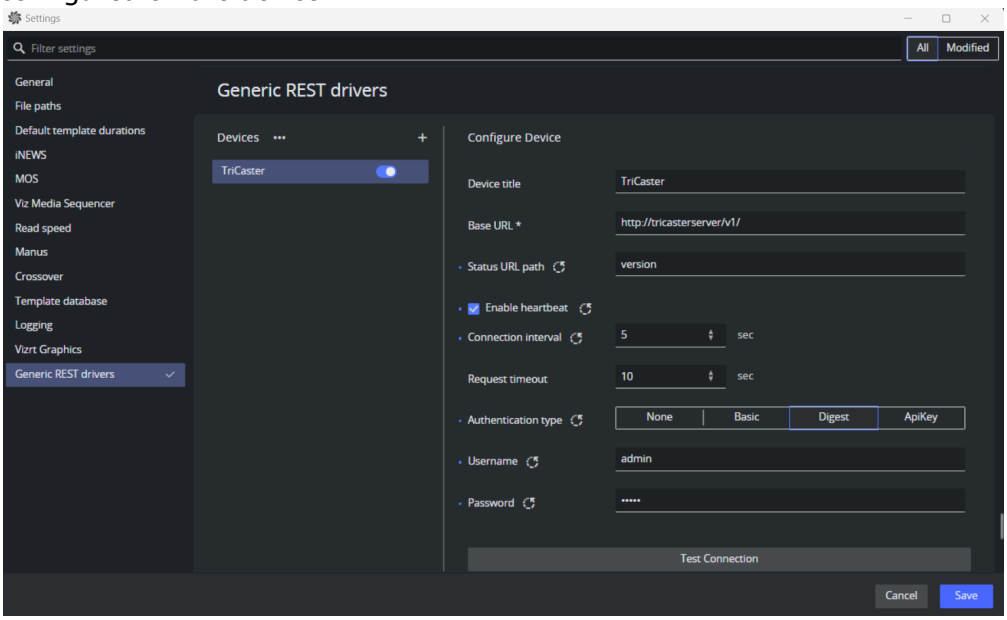
3. Add a new device by either
   a. Clicking **Add device** or
   b. Clicking the plus (+) sign next to **Devices**.
   You can add up to 50 devices.
4. A new device appears in the device list (below heading *Generic REST drivers*). You configure its properties in the right side panel.



   a. **Device title**: A name that uniquely identifies the generic REST device. Default: *Untitled*.
   b. **Base URL**: This mandatory field must include the minimum necessary components of the REST API's URL: the scheme (http:// or https://) and the hostname or IP address of the device. Additionally, it can include a common resource path that forms the base for all requests to the device. Essentially, the *Base URL* should represent the shared part of the URL used in all requests to the device.

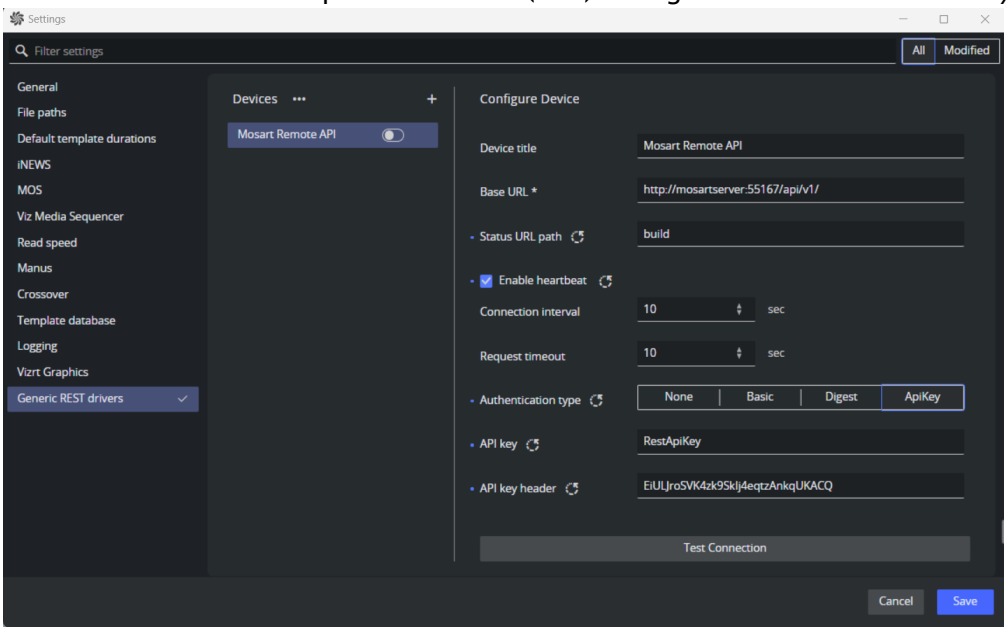
Copyright © 2024 Vizrt

c. **Status URL path**: The resource path to be attached to the base URL for testing the connection to the device's REST API or checking the device's heartbeat.
Provide this if the base URL alone is insufficient for testing the connection. Default: *empty*.
d. **Enable heartbeat:** When enabled, Viz Mosart will periodically send a request to the combined URL (*Base URL* concatenated with the *Status URL Path*) to verify the device's endpoint.
This check occurs at the interval specified in the setting **Connection interval**. Default: *disabled*.
e. **Connection interval**: The interval, in seconds, for sending the heartbeat request. Default: *10* seconds. Minimum: *1* second.
f. **Request timeout**: The maximum time Viz Mosart will wait for a response from the server when an HTTP request is sent to the device, before timing out. Default: *10* seconds.
g. **Authentication type**: Authentication type required by the device. Options include:
    - *None*: No authentication required by the device. This is also the default.
    - *Basic*: A simple authentication scheme using a *Username* and *Password* as configured on the device.
    - *Digest*: An authentication scheme using a *Username* and *Password* as configured on the device.



    - *ApiKey*: An authentication scheme requiring the API Key name and value as configured on the device.
    For example, when configuring Mosart Remote Control REST API as a Generic Rest driver, add X-API-Key as the *key header* and the value of `RestApi` setting

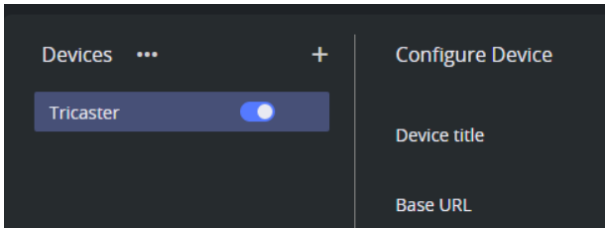
from Remote Control Dispatcher Service (RCS) configuration file as the *API key*:



5. Test the connection by clicking **Test Connection**.
   If the connection fails, an error message is displayed. See the Troubleshooting section for more details.

   > ⚠ **Note**: A REST-based device might not require authentication for certain HTTP request commands, such as those used to check the API version. These commands are commonly utilized for testing the connection to the device's REST API.
   > However, a successful response from this type of test does *not* necessarily confirm that authentication has been verified. In scenarios where authentication is crucial, it is recommended to configure the Status URL path to a resource that demands authentication.
   > Successful testing with this path ensures proper authentication setup. Afterwards, you can change the path to the one intended for regular heartbeat checks.
   > For example, the Mosart REST API does not require authentication for the endpoint `http://lbu-pc:55167/api/v1/build`. Consequently, a connection test using this URL will not validate the validity of the APIKey settings.

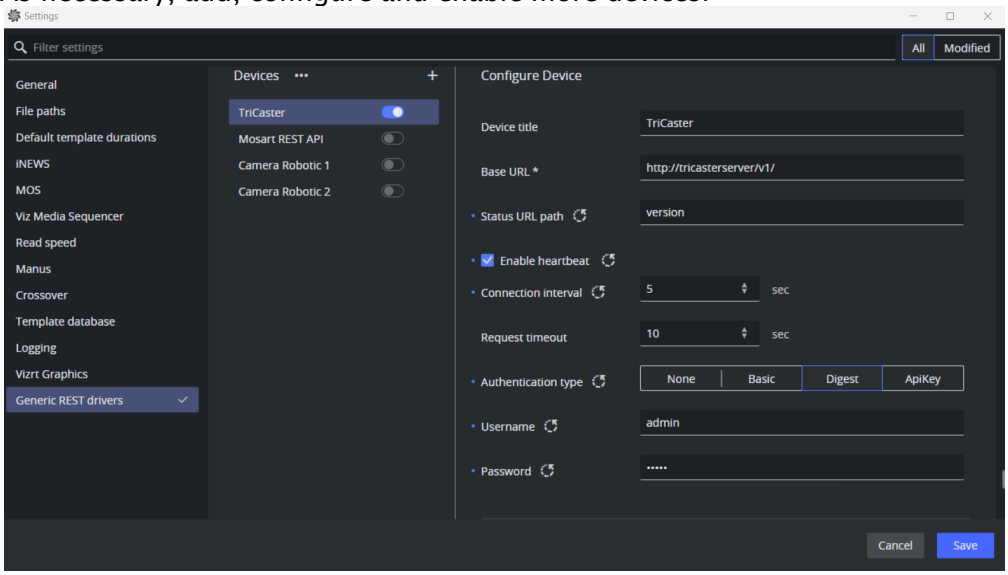
6. Once connection is successfully established, *enable* or *activate* the device by toggling the **Switch button** next to the device's name.



   This action initiates an HTTP connection to the device, preparing it to receive commands from Viz Mosart.

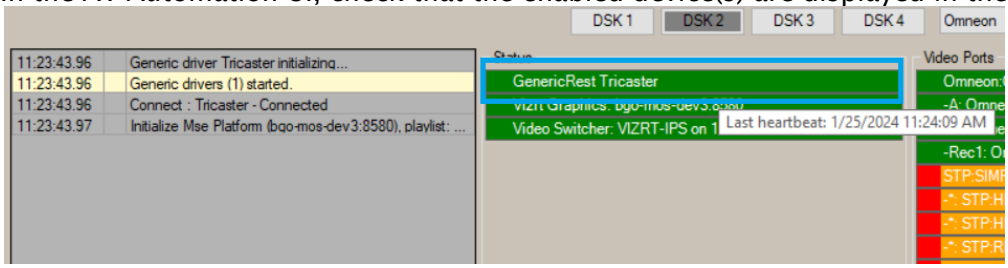
   > ⚠ **Note**: If the device is not enabled, Viz Mosart will *not* transmit any commands to it!

7. As necessary, add, configure and enable more devices:



> **Maximum number of enabled devices**
>
> **Note**: In the initial release (Viz Mosart version *5.5*), only **one Generic REST device can be enabled** at a time. To enable an alternative device, first disable the currently active one.
> For this reason, the *Enable all devices* action on the *Devices* context menu is disabled in v5.5.

8. After configuring the device(s), save all settings by clicking **Save**.
9. In the AV Automation UI, check that the enabled device(s) are displayed in the Status panel.



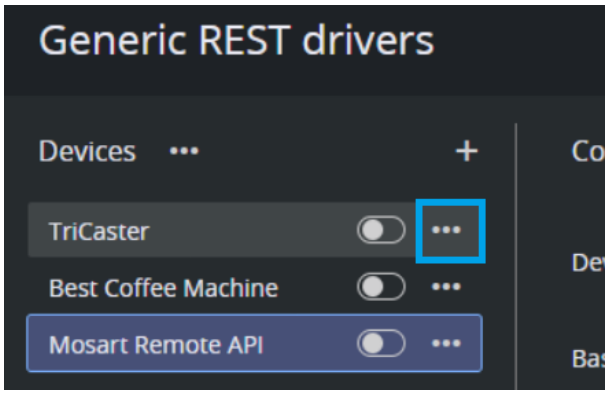
   a. If no heartbeat is configured, Mosart only checks the connection once and shows the correct status (green for OK, red for not OK).
   b. If a heartbeat is configured, hovering over the device in the status area displays last heartbeat check details.
   c. You can re-connect to the device by pressing **Ctrl+Shift+Z** in AV Automation.

Copyright © 2024 Vizrt

Device Management

To Remove A Configured Device

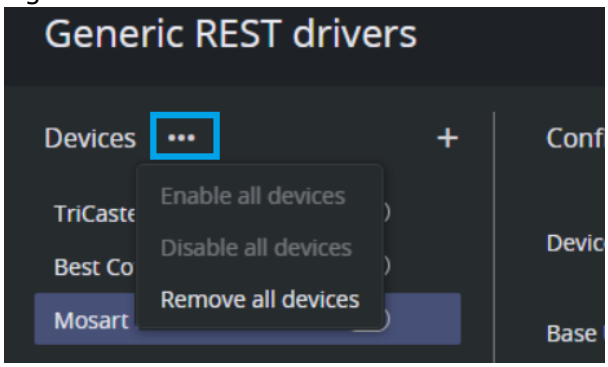
- Click on the three dots to the right of an item in the list of devices and select **Remove device**.



To Enable, Disable Or Remove All Configured Devices

(See the Note above related to which features are available).

- Right-click the three dots next to **Devices** and select from the context menu.



Controlling a REST-based Driver

Standard Viz Mosart templates can be used to control a REST-based driver.

These show-specific operations are described in section Controlling a REST-based Driver.

Troubleshooting

When testing, if the connection to the device fails:

- Ensure the *Base URL* is correctly configured.
- Consider adding a *Status URL Path* in addition to the *Base URL*, if needed, and ensure it is correctly formatted.
- Check that the correct authentication is set up if the device requires it.
- Confirm that AV Automation is operational.

Copyright © 2024 Vizrt

- Review the Mosart logs for further details.

Copyright © 2024 Vizrt

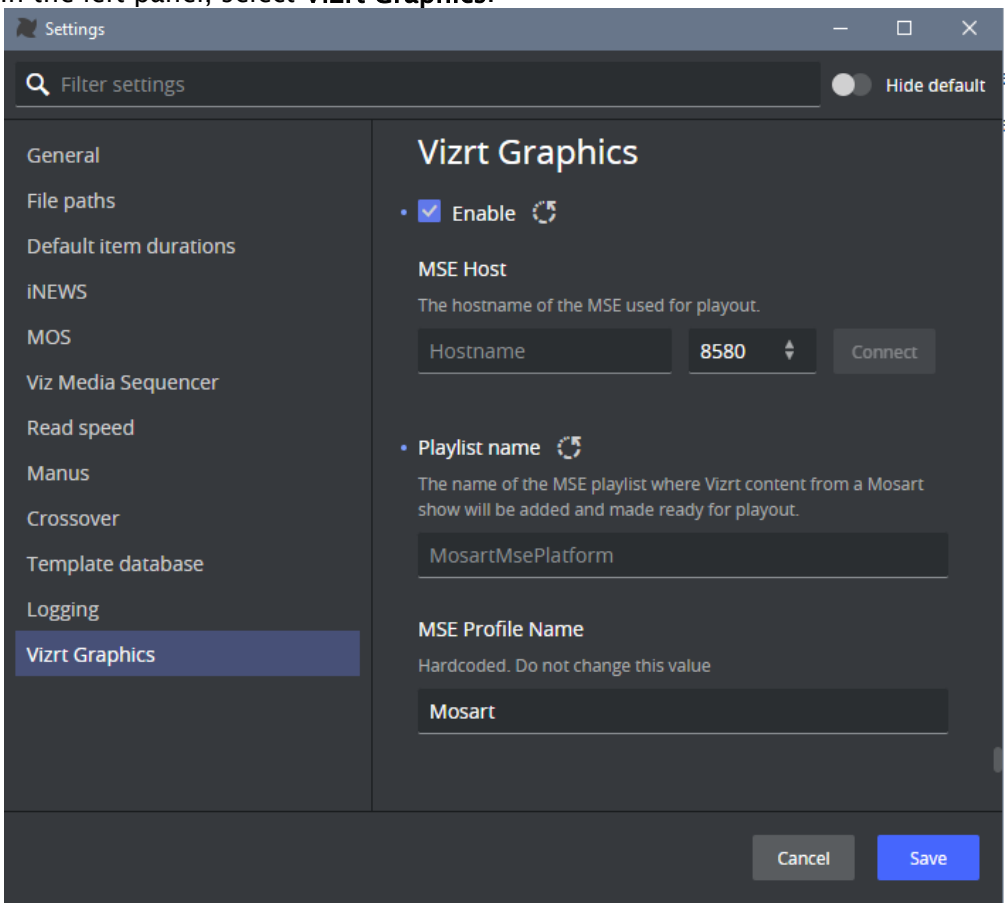
9.1.3 Vizrt Graphics

For Vizrt graphics (Viz Arc AR elements and limited support for overlays, full screens and clips) setup and workflow is simplified with a single driver that communicates with Vizrt's central media broker, the Media Sequencer (MSE). MSE communicates with AV Automation.

Connection to the Vizrt Graphics driver is configured through the new web-based **Settings** menu.

To configure Vizrt graphics

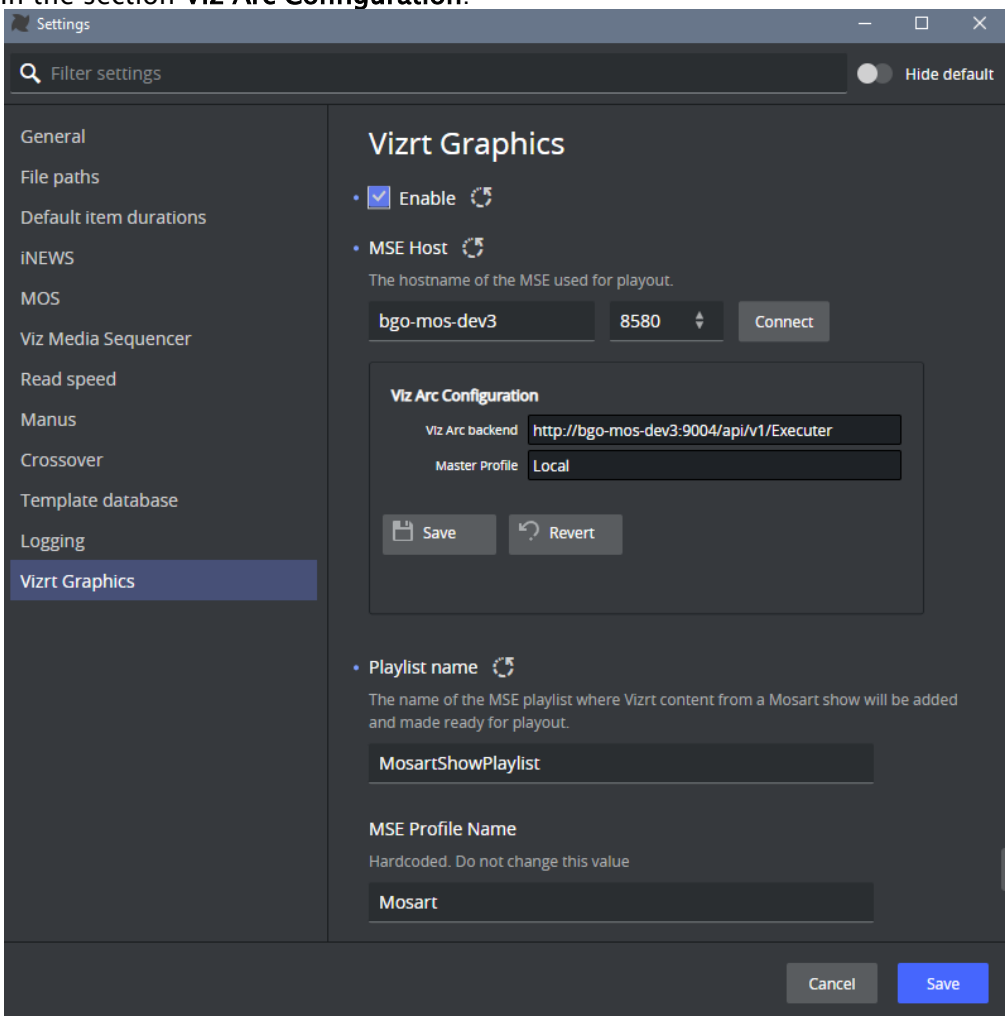
1. Open the **Settings** menu, as described in Viz Mosart Settings and Configuration.
2. In the left panel, select **Vizrt Graphics**.



3. In the panel to the right, select **Enable**.
4. In the MSE Host section provide:
   a. Host name
   b. Port number (default 8580).
5. Click **Connect**.
   On successful connection to MSE, a page served by Media Sequencer is displayed, just below the host and port fields.

Copyright © 2024 Vizrt

6. In the section **Viz Arc Configuration**:



   a. Enter a value for **Viz Arc backend**, as a URL, in the form: *http://<Viz Arc REST service hostname>:<Viz Arc REST service port>/api/v1/Executer*. (default Viz Arc REST service port is *9004*. See the Viz Arc User Guide).
   b. The **Master profile** is defined in Viz Arc. Here is an example for Viz Arc 1.8.
   A *profile* is used to configure the different groups (channels) of either the *Viz* or *Unreal Engines* available for user control.
7. Click **Save** to save to the MSE (or **Revert** to undo your changes).

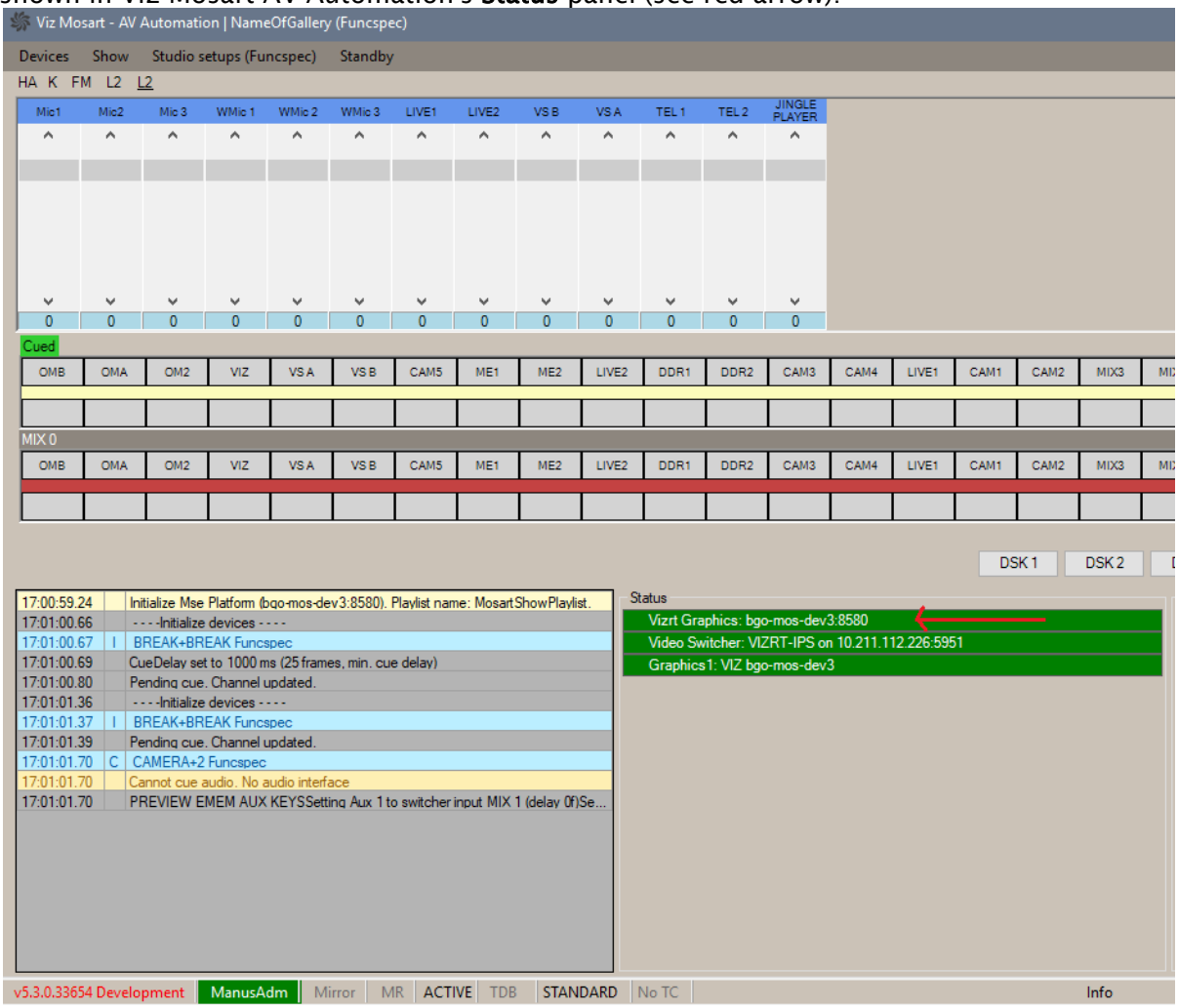
> ⚠ **Note**: These settings are saved on the Media Sequencer side (*not* Viz Mosart).

8. Enter a value for the MSE **Playlist name**.
   This is where the Viz content playout elements added in a Viz Mosart show, planned in an NRCS, will be added to be handled through the Vizrt Graphics driver.
   This value is *case sensitive*. Default: *MosartMsePlaylist*
9. Enter the **MSE Profile name**.
   This is used for playing out elements. The selected profile must first be *created* and *activated*, using a profile editor available in either Viz Trio, Viz Pilot (Director) or Viz Multiplay.
   This value is *case sensitive*. Default: *MOSART.*

Copyright © 2024 Vizrt

10. Click **Save** to apply the setting.
    There is no need to restart the Manus Administrator.

With the settings applied, the MSE connection status that employs the *Vizrt Graphics* driver is shown in Viz Mosart AV Automation's **Status** panel (see red arrow):



Copyright © 2024 Vizrt

9.2 Story Recorder Mode

- Enabling Story Recorder mode
- Prerequisites
- Overview
- Activating Story Recorder Mode
- Verification
- Troubleshooting
- Housekeeping

9.2.1 Enabling Story Recorder mode

To enable Story Recorder (SR) mode in your setup, follow the configuration procedures below.

> ⚠ **Note**: Once SR mode is activated, you will need to establish system-wide, frame accurate clock synchronization. This is described in section Frame Accurate System Operations.

End user (Viz Mosart operator) documentation for Story Recorder is provided in the section **Story Recorder** of the Viz Mosart User Guide.

9.2.2 Prerequisites

- The Viz Mosart system is already fully operative in live (non-SR) mode.
- The system, including studio equipment, can be calibrated for frame-accurate operations, as described in section Frame Accurate System Operations.
- A dedicated set of frame-accurate templates has been created for Story Recorder.
  Frame-accurate operations are automatically latency-compensated for primary events, introducing timing differences in the execution of the templates, compared to Standard mode.
  Additionally, not every feature in Standard mode is available or possible in Frame Accurate mode.
- An operating license is obtained from Vizrt Support.

  > ⚠ **Note**: Please refer to the Viz Mosart Release Notes for details of which devices are supported for Story Recorder mode.

9.2.3 Overview

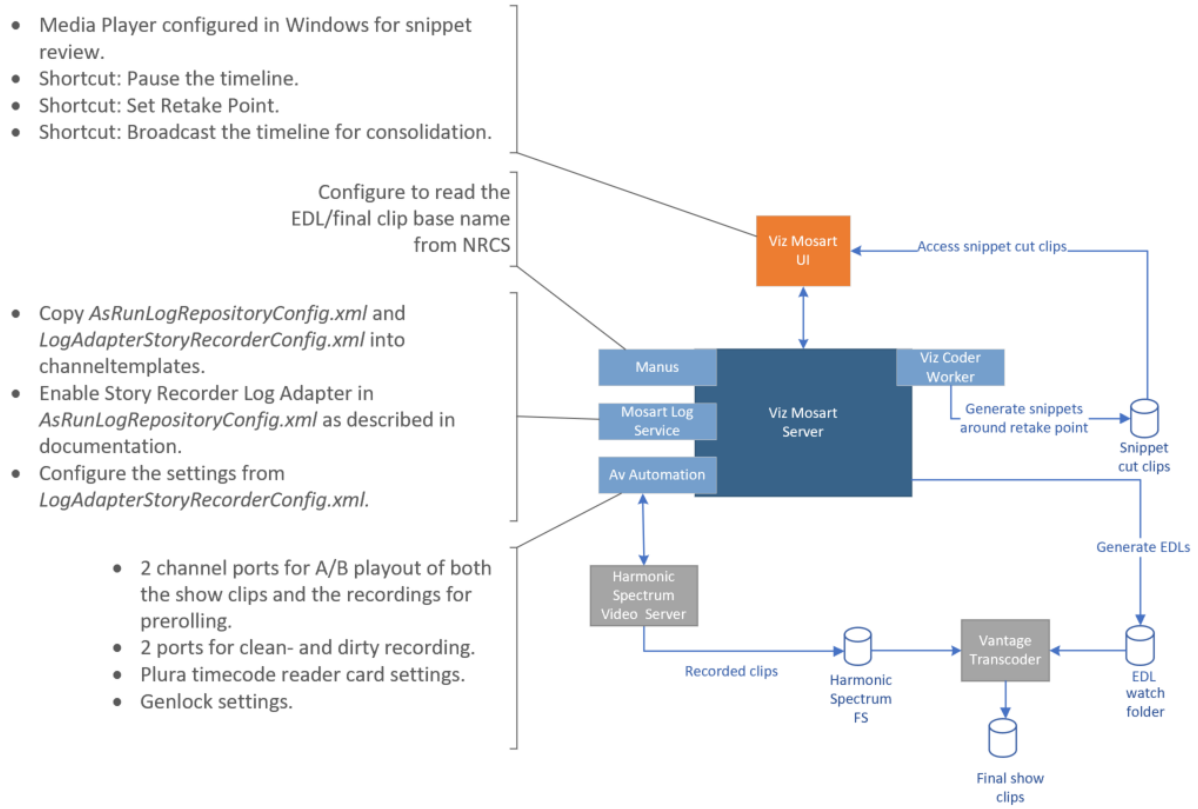
Story Recorder (SR) both introduces new system components and repurposes existing functionality. Important to note are:

- The *Mosart Logging Service* plays a significant role together with a new Story Recorder Log Adapter, for capturing events for recording.
  This new log adapter must be configured *after* installing the Mosart server.

Copyright © 2024 Vizrt

- A snippet transcoder, Vizrt's *Coder*, is added to the installation.

The relationships between the system elements introduced with SR are presented below. This example has a Harmonic Spectrum video server and Vantage for EDL transcoding:



Note: The folder where the final show clips reside is not controlled by Viz Mosart. This is set on the EDL transcoder side. In the above setup, the EDL transcoder is *Vantage*.

9.2.4 Activating Story Recorder Mode

Execute each of these procedures:

- Upgrading to a Version of Viz Mosart that includes Story Recorder
- Enabling the Story Recorder Log Adapter
- Configuring Video Server Settings for Playout and Recording
- Enabling Snippet Generation at the Cut Point
- Managing Naming of the Final Clip Show
- Pre-Roll Duration
- Automatic Pause at Show End
- Failover

Copyright © 2024 Vizrt

Installation order

For setups where several Mosart applications are installed on the same machine and are using their own logging service (for example, Viz Mosart UI, Viz Mosart Audio Player and Viz Mosart Server), please ensure that the Viz Mosart Server was the *last* component to be installed and that the Mosart Logging Service points to the Viz Mosart Server installation folder. This is relevant for Story Recorder because any supported log adapters (in this procedure, the newly introduced Story Recorder Log Adapter) reside on the Mosart Server side. For additional information, refer to the topic System Logging.

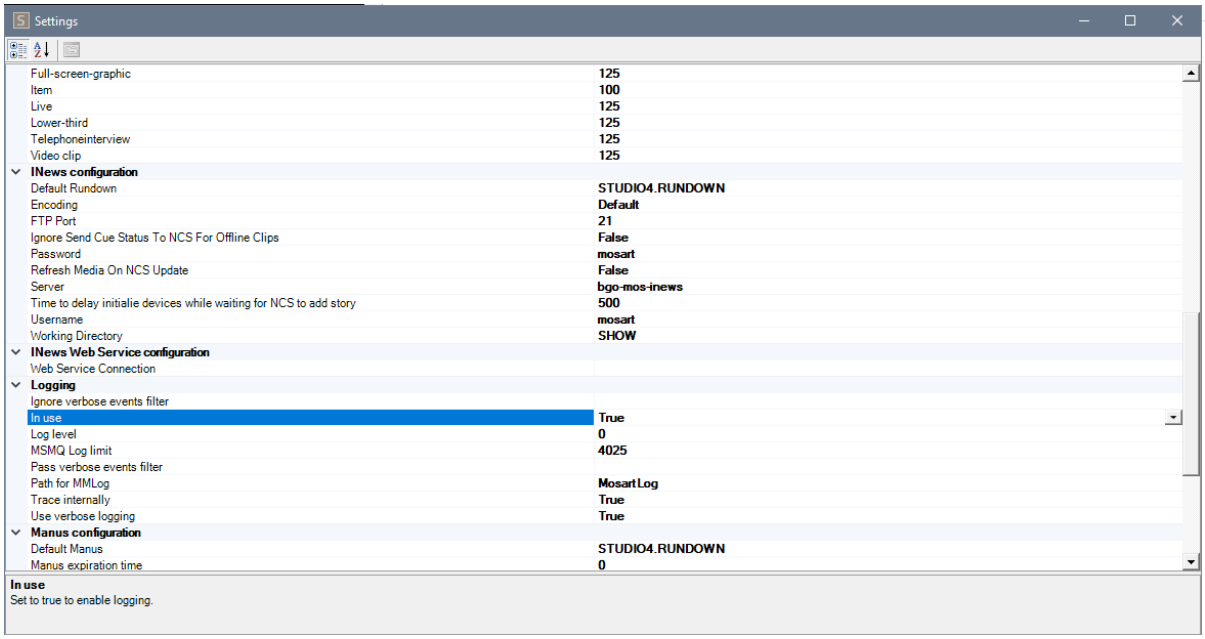
Upgrading to a Version of Viz Mosart that includes Story Recorder

Story Recorder was introduced with Viz Mosart 5.0, so the first step will be to install the new software bundle.

1. Stop the service *Mosart Logging Service*.
2. Install the new software bundle taking into account the previous note regarding the installation order.

Enabling the Story Recorder Log Adapter

1. In the Manus Administrator console, type **settings** and verify that **Logging** > **In use** is set to *True*:



2. Check if you have the following two files in your working folder, Channeltemplates or ProgramData\Mosart Medialab\ConfigurationFiles\.
   If yes, go to the next step.
   If not, copy

1. *AsRunLogRepositoryConfig.xml*

2. *LogAdapterStoryRecorderConfig.xml*
   from `C:\Program Files (x86)\Mosart Medialab\Mosart Server\ConfigurationFiles` into your working folder.

- Open **AsRunLogRepositoryConfig.xml** in an editor and add or uncomment the following line:

```
<Adapter type="LogAdapterStoryRecorder" name="LogAdapterStoryRecorder" configuration="LogAdapterStoryRecorderConfig.xml" assembly="LogAdapterStoryRecorder.dll" />
```

- Open **LogAdapterStoryRecorderConfig.xml** in an editor and configure the following settings:

Setting	Description
RemoteDispatcherHost	The host for service Mosart Remote Dispatcher (MosartRemoteControlService) is always local, running on the Mosart server machine. Default: *localhost* . No other value should be given.
RemoteDispatcherPort	The port where the Mosart Remote Dispatcher service (MosartRemoteControlService) resides. This should be the same as configured in *RemoteDispatcherServiceConfig.xml.* Default: *8090* (locally on the Mosart server machine).
BackupHost	The hostname of the Mosart backup server.
BackupPort	The port for connecting to the Backup Story Recorder Adapter. This should be the same on both the main and backup Mosart machine. Default: *5001*.
EdlWatchFolder	Location of a shared folder to which Viz Mosart has access and monitored (watched) by your EDL transcoder. In the figure above, this folder is named *EDL watch folder* . This setting can not be empty and can be any shared network path where both Viz Mosart and the EDL transcoder have access (for example, *somehostname**EDL*). The folder must have read and write permissions for the *Mosart Server Logging Service.* Viz Mosart does *not* clean up any files created in this folder. See more details in section Housekeeping.

Copyright © 2024 Vizrt

Setting	Description
EdlLocalFolder	Location of a local folder where Viz Mosart drops the Story recorder EDL files. This folder is a secure 'backup' for the *EdlWatchFolder*. The EDL transcoding node does not have access to this folder, but if ever the EDL files fail to be generated in the *EdlWatchFolder* because of say, connection issues, the user can later manually copy the files to the EDL transcode node. The setting can not be empty and it must be a different location than the *EdlWatchFolder* (for example, *C:\MMLogs\EDL or* \ *\mosart_server_hostname\MMLogs\EDL*). This folder must have read and write permissions for the *Mosart Server Logging Service.* Viz Mosart does *not* clean up any files created in this folder. See more details in section Housekeeping.
StoryRecorderSnippetClipsFolder	When a retake of a story item is performed, a rendered snippet around the cut point will be placed in this folder shortly after the item was retaken. If this property is not configured or left empty, *no* snippet of the cut can be created. The location should be a network shared folder where the user running *Mosart Server Logging Service* has read and write access and the user running *Mosart GUI* has at least read access. (for example, *\\somehostname\StoryRecorderSnippetClips*) In the figure above, this is *Snippet cut clips*. To set up a snippet recorder see Enabling Snippet Generation at the Cut Point. Default: *empty*. This implies that a snippet recorder is optional.
SnippetGenerationRequestDelay	How long in seconds after the retake process started, that generation of the snippet clip (from around the cut) shall begin. (The retake process is the last part from the following chain of actions : Show is paused → Pre-roll is taken on-air → Pre-roll auto-takes the item to be retaken). See section **Story Recorder** of the Viz Mosart User Guide*)*. Default and minimum recommended: *15 seconds*.
SnippetLengthForEachRecording	The amount of time (in timecode format) from each recording to be included in the snippet clip. Default: *00:00:05:00* (5 seconds from each recorded clip).

Copyright © 2024 Vizrt

Setting	Description
RecordingClipNamePrefix	Prefix added to the recording clips generated during a Story Recorder session using the configured video recorder. This is used by Viz Mosart to easily identify the recorded clips for housekeeping. (Only clip file names prefixed with this value will be subject to Viz Mosart housekeeping.) Default: *SRVizrt*.
RecordingFolder	Location of the video server's file system folder where it stores recordings created by Story Recorder. In the figure above, this folder is named *Harmonic Spectrum FS*. *(for example, \\somehost\fs0\clip.dir*). This location is defined on the video server side. This folder must have read and write permissions for the user running *Mosart Server Logging Service* which performs housekeeping tasks.
RecordingRetentionDays	Number of days to keep the recording clips (minimum 1 day) on the video server. An empty, zero or a negative value means the housekeeping of the intermediate recording files is disabled, there is *no* deletion of the intermediate files. If enabled, the housekeeping task is run every time *Mosart Server Log Service* is (re)started and keeps running at regular intervals defined by *RecordingRetentionDays* setting. Note that the recorded clips can be *very large* files easily filling the disk space. See also section Housekeeping. Default: *0* (disabled).
PrerollTemplate	The name of the variant of the pre-roll template. This will be used to automatically create the pre-roll template in the current studio setup. Default: *PREROLL*. **Note**: If another package template with this same name exists on your system which was manually created, *delete* or *rename* it. The pre-roll template for Story Recorder must retain its default minimal characteristics and can not contain additional control parameters. This unique minimal template can be reused after an upgrade.

Copyright © 2024 Vizrt

Setting	Description
MinPrerollDuration	The minimum and default preroll duration in seconds that the Preroll template will insert. ⚠ This time needs to be sufficient for the video server to reset and be ready to record, after being paused. Typically 10-15 seconds. The preroll duration can be increased by the user in the Story Recorder UI, see Minimum Pre-Roll Duration. Default: *10*.
MaxPrerollDuration	The maximum preroll duration in seconds that the Preroll template will insert. (See MinPrerollDuration above.) Default: *30*.

- Any changes in the configurations above require a **restart** of the service *Mosart Server Logging Service*.

⚠ For further details about logging in Viz Mosart, refer to section System Logging.

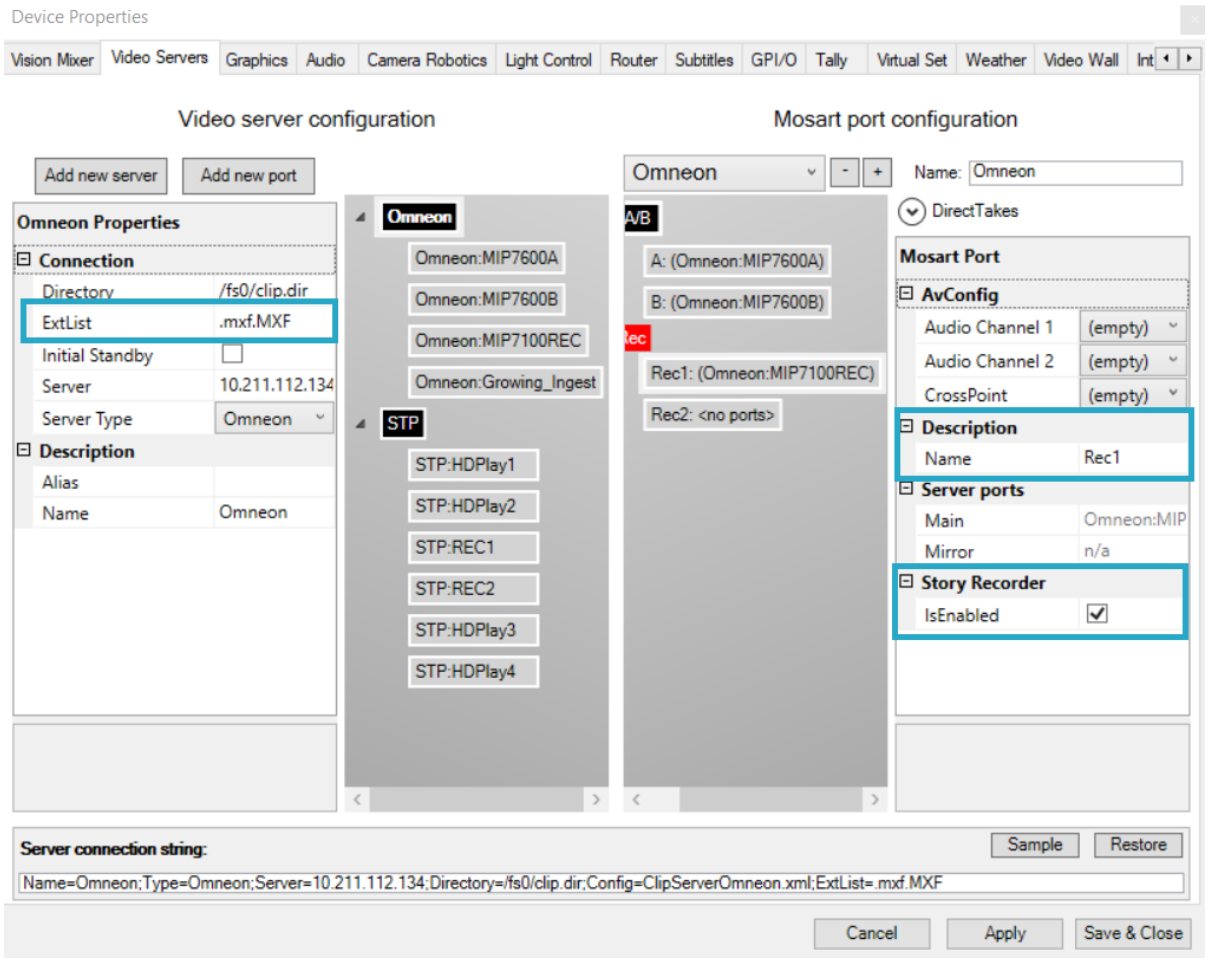
Configuring Video Server Settings for Playout and Recording

The video server is used for both playout of clips during the show and the recording of the story. Ports are configured for both dirty and clean recording.
These port settings on the video server depend on defined signal inputs and are not defined from Viz Mosart. To enable Story Recorder, some small changes must be made on the Viz Mosart side.

To configure video server recorder ports for Story Recorder

1. In AV Automation, navigate to **Devices > Properties > Video Servers**.
2. The settings for the **Video server configuration** and **Mosart port configuration** are already OK for a working system.
   Otherwise, refer to section Video Servers.
3. Configure the recording port(s) dedicated for Story Recorder.
4. For the recording port(s) mentioned in the step above, select **IsEnabled** in the **Story Recorder** section.

Copyright © 2024 Vizrt



This setting reserves the selected port for Story Recorder.

5. Click **Apply** then **Save & Close**.
6. Force a reconnect to the video server with **Ctrl+Shift+V**.
   (or restart of Av Automation).

Clip naming pattern of temporary recordings

The temporary recording files generated by Viz Mosart are automatically named.

- When only one recorder is configured, the clip name pattern is*<RecordingClipNamePrefix>_<counter_<timestamp>.<extension>*, where
  - The *RecordingClipNamePrefix* is the setting configured in step (4), when enabling the log adapter earlier.
  - The *counter* is reset to 1 for each Story Recorder session.
  - The *extension* is configured in **AV Automation > Devices > Properties > Video Servers > ExtList** (the first in the list will be chosen). See the previous screenshot.

  For example:
  SRVizrt_1_211001153956.mxf
  SRVizrt_2_211001153956 .mxf
  SRVizrt_3_211001153956.mxf

Copyright © 2024 Vizrt

- When more than one recorder are configured, the clip name pattern is *<RecordingClipNamePrefix>_<recorder name>_<counter_<timestamp>.<extension>*, where
  - The *<recorder name>* is the *Name* setting configured in *Devices/Properties/ Video Servers* on the virtual recorder port properties (see setting *Name* set to Rec1 in the previous screenshot).
  - The *counter* is reset to 1 for each Story Recorder session.
  - The *extension* is configured in **AV Automation > Devices > Properties > Video Servers > ExtList** (the first in the list will be chosen). See the previous screenshot.

  For example:
  SRVizrt_Rec1_1_211014172330.mxf
  SRVizrt_Rec1_2_211014172330.mxf
  SRVizrt_Rec1_3_211014172330.mxf
  SRVizrt_Rec2_1_211014172521.mxf
  SRVizrt_Rec2_2_211014172521.mxf
  SRVizrt_Rec2_3_211014172521.mxf

Enabling Snippet Generation at the Cut Point

This setup is optional and generates a few seconds (configurable) of verification video, intended to reveal the integrity of a fresh cut, after a story item has been paused and then retaken. This snippet creation employs Vizrt's versatile transcoder, Coder.

1. Download the most recent .exe version of Coder from the customer FTP, by navigating to /products/VizOne/Latest Versions/Individual Installers and Docs/Coder/Latest Version.
2. You need only install the Coder *Worker* .
   Install Coder on the same machine as the Viz Mosart Server. Coder's default installation location is in C:\Program Files\Vizrt\Coder.

   Note: If you install Coder anywhere but the default location, you must then update the following configuration, to specify the new path to the Coder executable, **amlbatch.exe**:

   ```
   <?xml version="1.0" encoding="utf-8"?>
   <configuration>
       <configSections>
           <sectionGroup name="userSettings" type="System.Configuration.UserSettingsGroup, System, Version=4.0.0.0, Culture=
               <section name="CoderTools.Properties.Settings" type="System.Configuration.ClientSettingsSection, System, Vers
           </sectionGroup>
       </configSections>
       <userSettings>
           <CoderTools.Properties.Settings>
               <setting name="ExecutableFileLocation" serializeAs="String">
                   <value>C:\Program Files\vizrt\Coder\bin\amlbatch.exe</value>
               </setting>
           </CoderTools.Properties.Settings>
       </userSettings>
   <startup><supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8"/></startup></configuration>
   ```

   This config file is C:\Program Files (x86)\Mosart Medialab\Mosart Server**CoderTools.dll.Config**.

3. On the machine that runs the Viz Mosart GUI, define a default media player for *.mxf files (for example, VLC).

Copyright © 2024 Vizrt

4. On the machine that hosts the Mosart server, in your working folder (ChannelTemplates or ProgramData), locate the file **LogAdapterStoryRecorderConfig.xml.**
   If not already set in step (4) earlier, provide values for:
   a. **StoryRecorderSnippetClipsFolder**
   b. **SnippetGenerationRequestDelay**
   c. **SnippetLengthForEachRecording.**
5. Restart both the
   a. Mosart Log Service (Windows Services menu)
   and
   b. Viz Mosart UI.

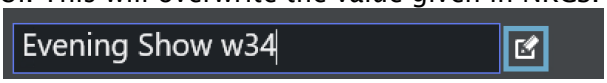
Managing Naming of the Final Clip Show

This step is optional. By default, naming of the show (and EDL files) is derived from the rundown name.

As a convenient option, you can instead provide a flexible and predictable naming scheme through a mapping field defined in the NRCS story.
Viz Mosart will then pick up this value and use it as the base for file naming, including the final show clip name. This is explained in Configure the NRCS and Viz Mosart for automatic naming of the Final Clip Show below.

> To prevent configuring the EDL/final clip base name every time the first story is changed, the rundown can be designed to, for example, start with a *Break* which will not be part of the show.

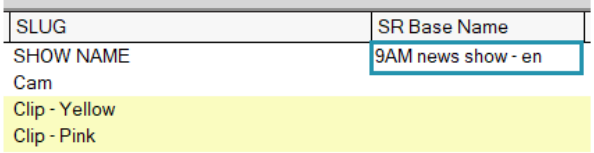
In addition, the user can also *manually* enter a name for the show's final clip, in the Story Recorder UI. This will overwrite the value given in NRCS.



Configure the NRCS and Viz Mosart for automatic naming of the Final Clip Show

From the NRCS, you can add a name that becomes the base of the final clip name in Story Recorder.

1. In the NRCS, open the corresponding rundown for the show.
2. Add an NRCS column (here for example, *SR Base Name*).
   The procedure is NRCS-specific and may require administrator privileges.
   The following example is from *iNews*.



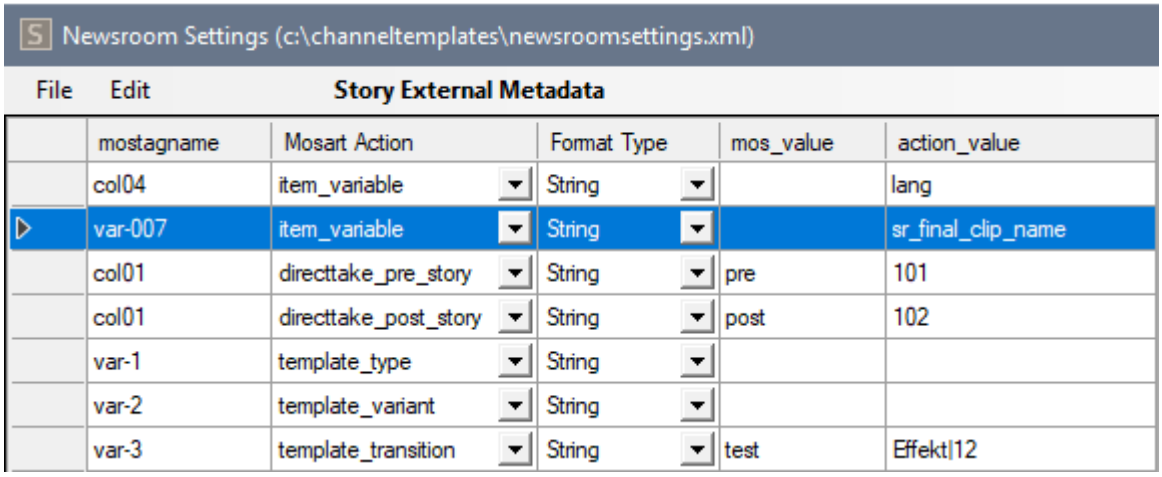
3. Under this new column, provide a *name* for the show, in **the first story in the rundown** (here, *9AM news show - en*).

> **Note**: Viz Mosart will only read the naming at *story* level, from the first story in the NRCS rundown. If the name is placed in any other story, it will be ignored.
> **Tip**: If your NRCS operates on rundown level, not story level (for example

OpenMedia), you can create a script that will propagate your desired final story name (as a text field on rundown level) to *all* stories in your rundown. As this also populates the *first* story, Viz Mosart will then successfully read the name.

This name will be mapped in Mosart to appear in the Story Recorder UI, when SR is enabled, as illustrated at the end of this procedure.
The NRCS-side configuration is complete, next is to arrange value-mapping on the Viz Mosart side.

4. In the Viz Mosart console application *Manus Administrator*, type *ns*. This opens the Newsroom Settings menu.
5. From the menu bar select **Edit > Story External Metadata** and add the following row if not already present:



	mostagname	Mosart Action	Format Type	mos_value	action_value
	col04	item_variable	String		lang
▶	var-007	item_variable	String		sr_final_clip_name
	col01	directtake_pre_story	String	pre	101
	col01	directtake_post_story	String	post	102
	var-1	template_type	String		
	var-2	template_variant	String		
	var-3	template_transition	String	test	Effekt|12

where
mostagname: the column name as specified in the NRCS (in this iNews example, *var-007*).

> ⚠ **Note**: For details on how to derive this naming (here var-007) from the NRCS, refer to section Story External Metadata in Newsroom Settings Editor .

Mosart Action : must be set to *item_variable* .
Format Type : must be set to *String* .
mos_value : empty.
action_value : must be *sr_final_clip_name* .

6. Restart the Viz Mosart Server.
7. Verify the setup:
   a. Provide the Final Clip Show name as described.
   b. Load the rundown in Mosart UI.
   c. Enable Story Recorder mode (see *Viz Mosart User Guide*, in the **Story Recorder** section).
   d. Check that the name entered in Managing Naming of the Final Clip Show, appears in the Story Recorder UI when the associated rundown is loaded and Story Recorder mode activated.

Copyright © 2024 Vizrt

⚠ Any changes made in NRCS for the Final Clip Show name are not taken into consideration during a Story Recorder session, but only when Story Recorder mode is again activated (so in the next Story Recorder session).

Clip naming pattern for EDL files and the final show clips

The EDL and final clip recording files are automatically named by Viz Mosart.

The name pattern used to generate the EDL files at show level and the final show clips generated by the EDL transcoder is:

- *<Final Clip Show name>_<timestamp>.<extension>*, when the show is broadcasted first time.
- *<Final Clip Show name>_v<counter>_<timestamp>.<extension>*, when the show is broadcasted more than once.
  where
  - *Final Clip Show name>* is the rundown name or as given from NRCS or as given in Story Recorder panel (see Managing Naming of the Final Clip Show).
  - *extension* for the EDL files is the extension for the supported EDL transcoder (for example, *tsedl* for a Vantage transcoder).
  For the final clip shows, this is the extension configured on the EDL transcoder itself.

  For the example used in this procedure. the *Show Final Clip* name is *9AM news show - en*:
  - 9AM news show - en_211011175050.tsedl
  - 9AM news show - en_211011175050.mxf
  - 9AM news show - en_v2_211011175050.tsedl
  - 9AM news show - en_v2_211011175050.mxf

Pre-Roll Duration

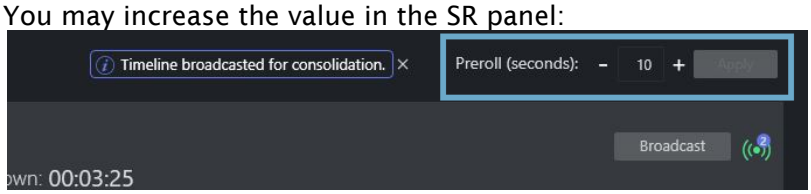
The pre-roll period must be long enough for the video server hardware to reset itself and be ready to continue recording, after a pause.

To set a pre-roll duration

1. Determine the time required by your video server hardware to commence recording. This is typically 10-15 seconds.
2. Configure this value as described in MinPrerollDuration.

Copyright © 2024 Vizrt

3. You may increase the value in the SR panel:



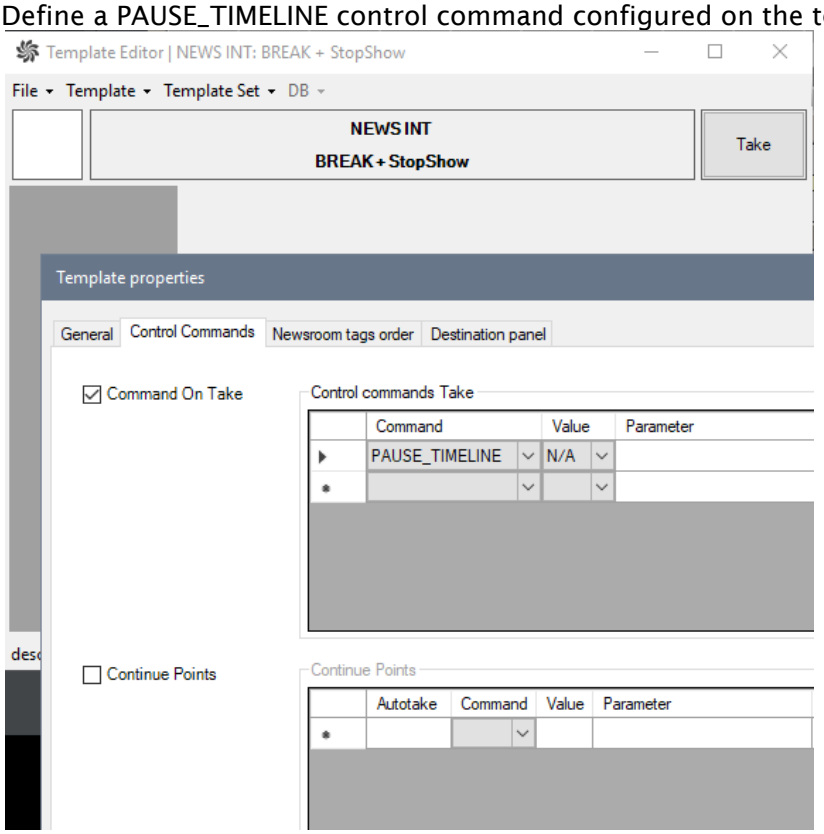
This setting is retained between SR sessions.

Automatic Pause at Show End

As an option, you can automatically pause the show and recording.

To set automatic pause

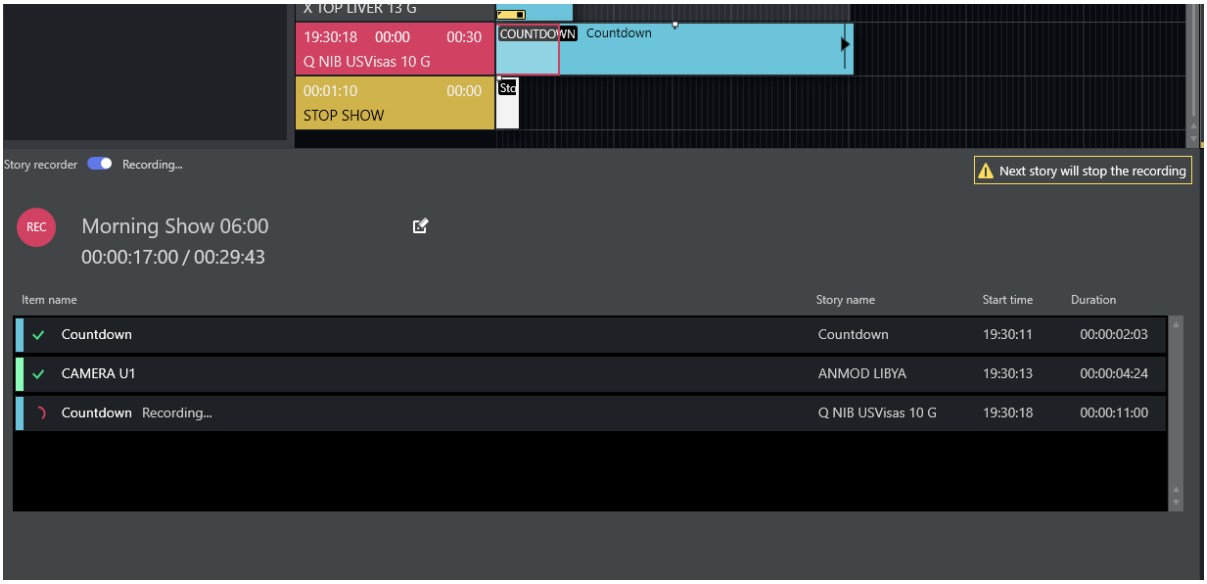
1. Create a dedicated template, typically a Break.
2. Define a PAUSE_TIMELINE control command configured on the template.



3. Insert this template at the end of the rundown.

In the Story Recorder session, when the story before this template is taken on air, a message warns the operator that the next story is going to pause the show and thus the recording too:

Copyright © 2024 Vizrt

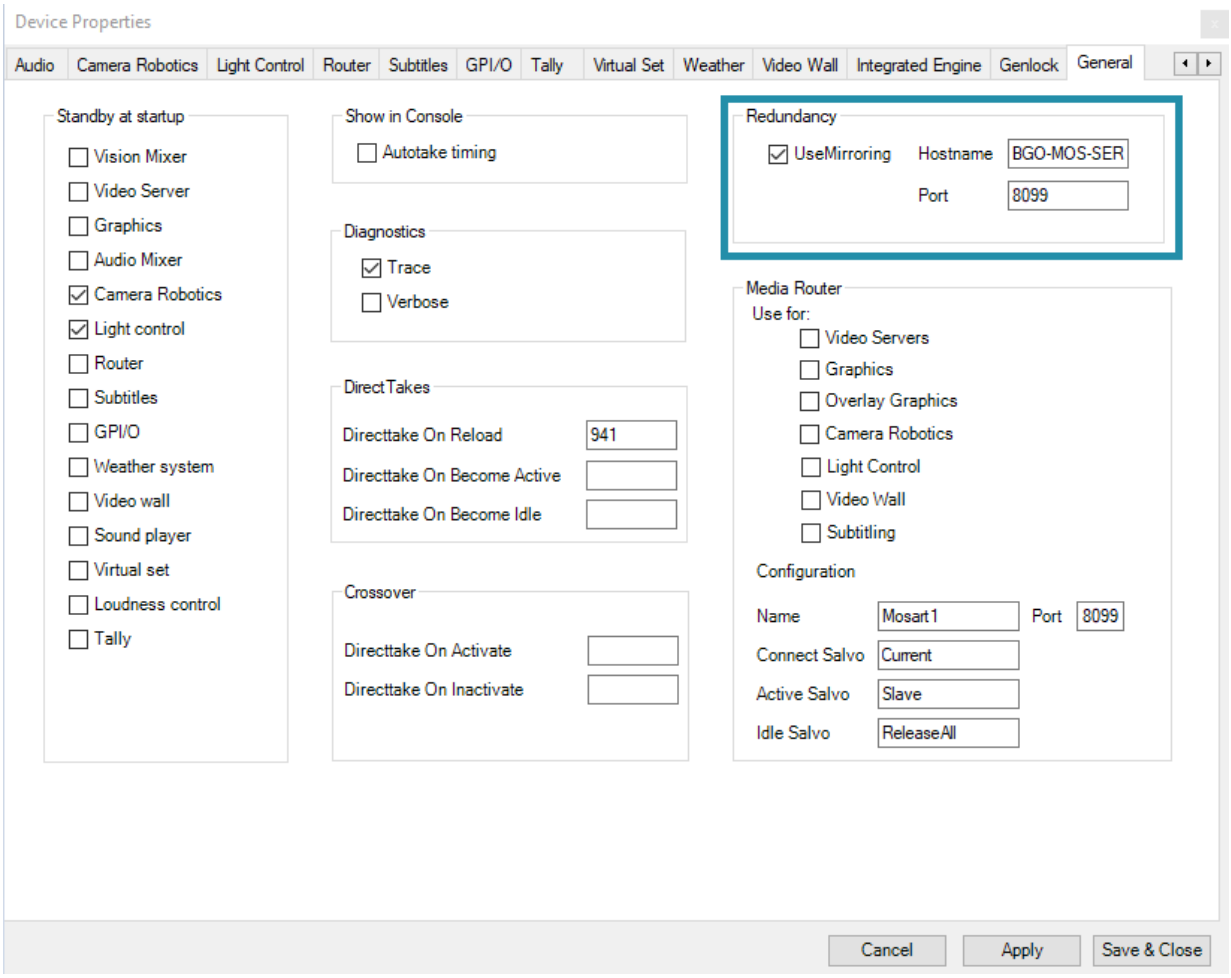


Failover

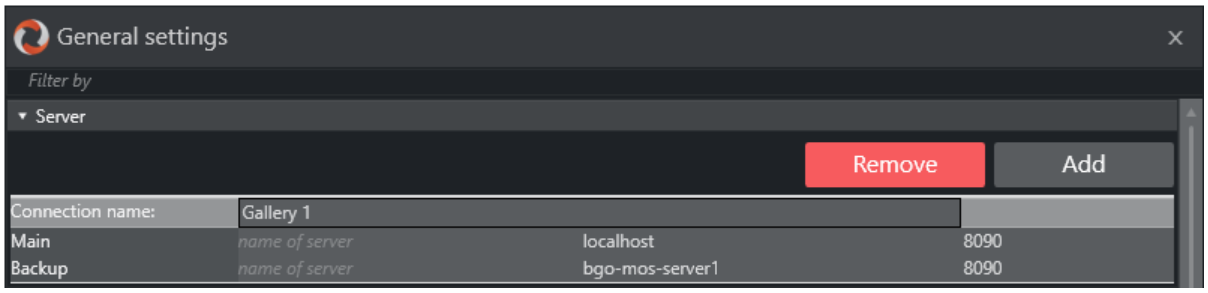
This step is optional and it assumes that Viz Mosart is already set up with redundancy as described in section Redundancy.
This means that:

- Both the Viz Mosart main and backup server have same configurations for at least Mosart Administrator, AV Automation, Media Administrator and Overlay Graphics Interface.
- AV Automation is configured for mirroring on both main and backup servers (with main pointing to backup and backup pointing to main):

Copyright © 2024 Vizrt



- The Viz Mosart UI is configured to connect to the backup server:



For Story Recorder, there are additional steps:

- Story Recorder is configured on both main and back in the same way. For simplicity, once main has been configured for Story Recorder mode as describe in section Enabling the Story Recorder Log Adapter, copy *AsRunLogRepositoryConfig.xml* and *LogAdapterStoryRecorderConfig.xml* in your working folder from the backup server (*Channeltemplates* or *ProgramData*).
- Open *LogAdapterStoryRecorderConfig.xml* from Mosart main server in an editor and provide the hostname and port to the backup server (**BackupHost** and **BackupPort**) as described in Enabling the Story Recorder Log Adapter, step 4.

Copyright © 2024 Vizrt

- Open *LogAdapterStoryRecorderConfig.xml* from Mosart backup server in an editor and provide the hostname and port to the main server (same **BackupHost** and **BackupPort** settings) as described in Enabling the Story Recorder Log Adapter, step 4.
- Both main and backup machines have Plura card installed and Viz Mosart configured to connect to Plura as described in Frame Accurate System Operations.
- (Optional) If Snippet generation around the cut point has been set up for main, this should be set up in the same way for backup. Install Viz Coder on *both* main and backup server. The snippet folder configured in *LogAdapterStoryRecorderConfig.xml* where the Viz Mosart UI has access, needs to be configured to be the *same* on main and backup.
- Same calibration settings from AV Automation: **Devices > Properties > Genlock** tab should be configured for main and backup.

9.2.5 Verification

- The video server is prepared for recording.
- Viz Mosart is prepared to run in Story Recorder mode.
- Snippet generation around the cut point is configured (optional).
- The integration is now established between Viz Mosart and the NRCS (optional).
- Failover is setup as described (optional).
- Ensure you have calibrated the system to frame accuracy, as described in section Frame Accurate System Operations.

The Viz Mosart operator instructions, for switching on and running Story Recorder are described in the *Viz Mosart User Guide*, in the **Story Recorder** section.

> ⚠ *We strongly suggest to build a new template set for Story Recorder, mainly because of the timing differences in the execution of the templates in Frame Accurate mode (now automatically latency-compensated for primary events), and partly because not every feature in Standard mode is available or possible in Frame Accurate mode.*

9.2.6 Troubleshooting

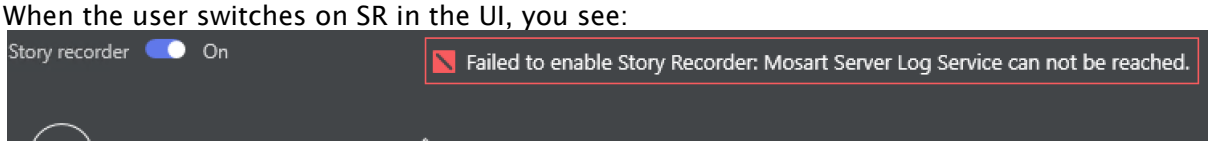
Story Recorder must always first be switched *On* in Viz Mosart UI.

- Failure to Enable Story Recorder Session
- Recording Ports not Found
- Failure to Start Recording or Story Items not Recorded in Story Recorder Panel
- Naming of Final Clip Show from NRCS not appearing in Story Recorder UI
- Unreliable genlock. See more details in AV Automation status bar
- Genlock is not enabled
- Retake position no longer valid. Please set retake point
- Timeline can not be paused while pre-rolling

Copyright © 2024 Vizrt

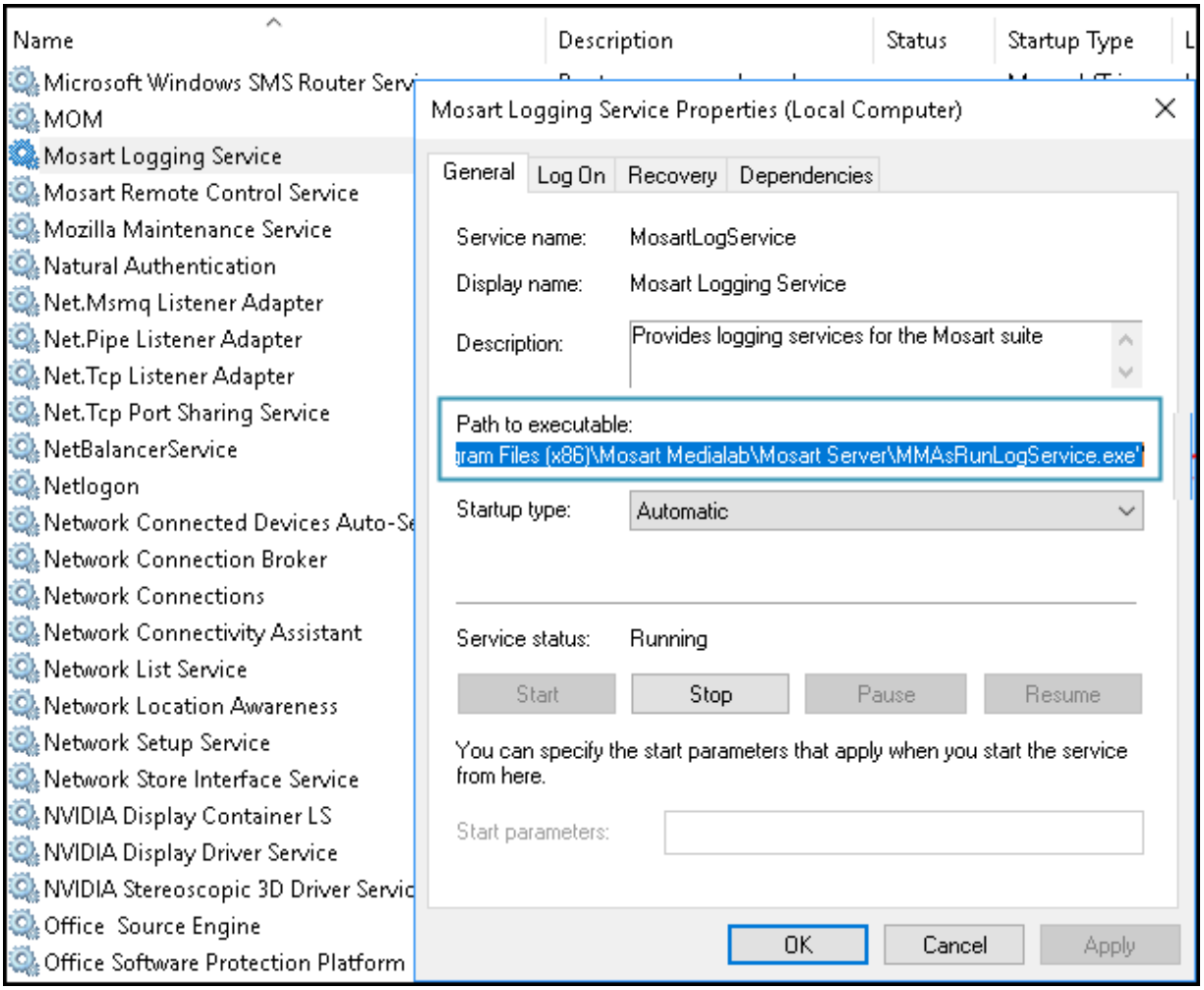
Failure to Enable Story Recorder Session

When the user switches on SR in the UI, you see:



Check these scenarios, in this order:

- Viz Mosart Server and Mosart UI or Mosart Audio Player are on the same machine and the server *was not installed last*.
  If you installed the server *before* the UI or Mosart Audio Player, Mosart Log Service will incorrectly point to a copy of `MMAsRunLogService.exe` that is located in the Mosart UI or Mosart Audio Player folder installation. Instead, this should point to the `MMAsRunLogService.exe` located in Mosart Server installation folder:



To fix this, you should **either** stop the Mosart Log Service from Services and start it as console from *Mosart Medialab\Mosart Server\ MMAsRunLogService.exe* (although you will still have

same problem after a machine restart) **or** re-install the applications in this order: first the UI/ Mosart Audio Player and then Server.

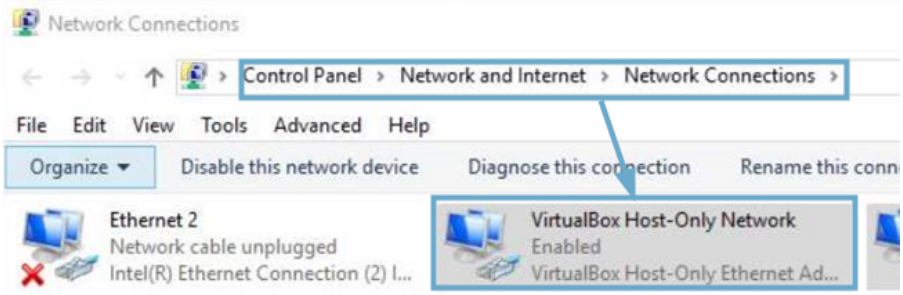
After Mosart Log Service is started, enable SR again.

- Verify that the (new) Story Recorder Log Adapter, and the config *LogAdapterStoryRecorderConfig.xml* are setup according to section above, Enabling the Story Recorder Log Adapter, around step (**3**).

A common oversight is that someone forgot to un-comment the lines for *Enabling the Story Recorder Log Adapter for Story Replay functionality* in `AsRunLogRepository.xml`.

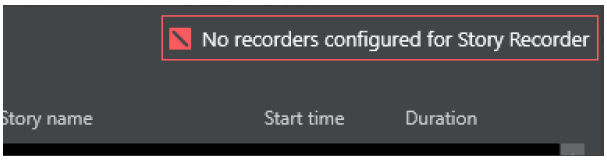
For additional help, refer to the general topic Troubleshooting Log Adapters.

- If the Viz Mosart UI is connected to a Mosart server on another machine, make sure there is *no VirtualBox* or similar configured on the Mosart server machine.



Recording Ports not Found

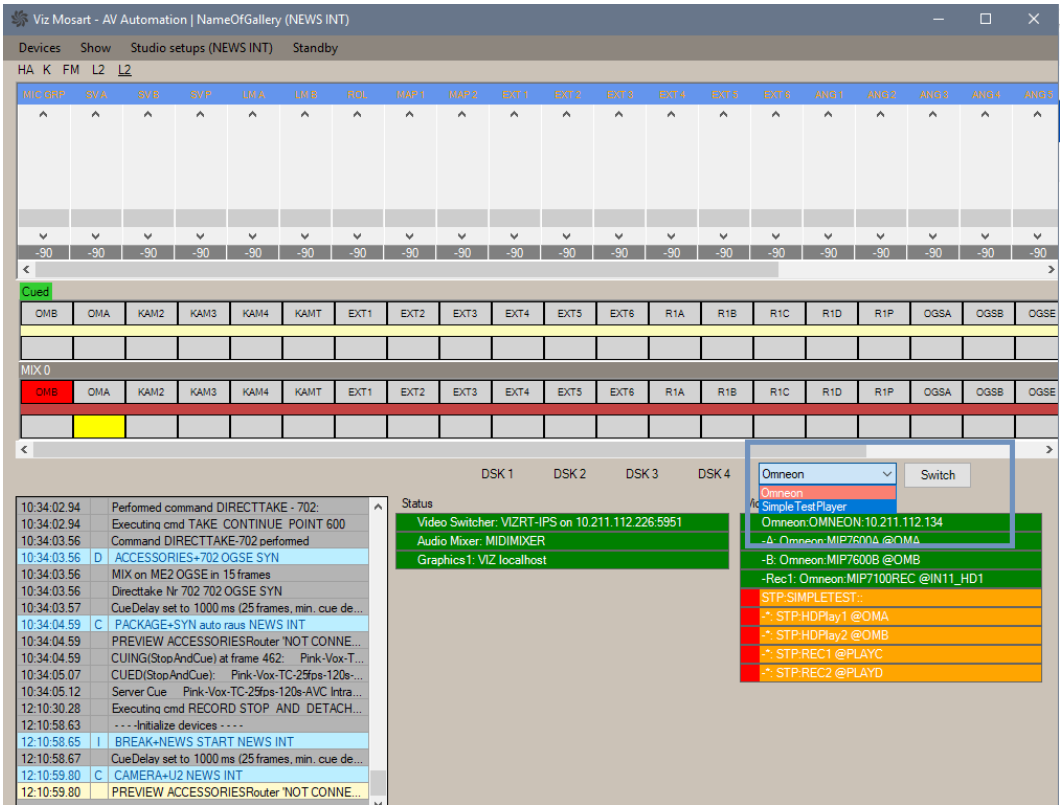
If when you switch on SR, you get errors about *No recorders configured for Story Recorder*, this indicates that the connection to the video server that performs the recorder is faulty.



- Ensure you set up Viz Mosart, with correct configuration settings selected, as explained above in To configure video server recorder ports for Story Recorder.

Even if the configured recording ports *are* enabled for Story Recorder as described above, you may also get this error message *after switching to a different video server salvo*.

Copyright © 2024 Vizrt



After switching to a different salvo, perform the following steps:

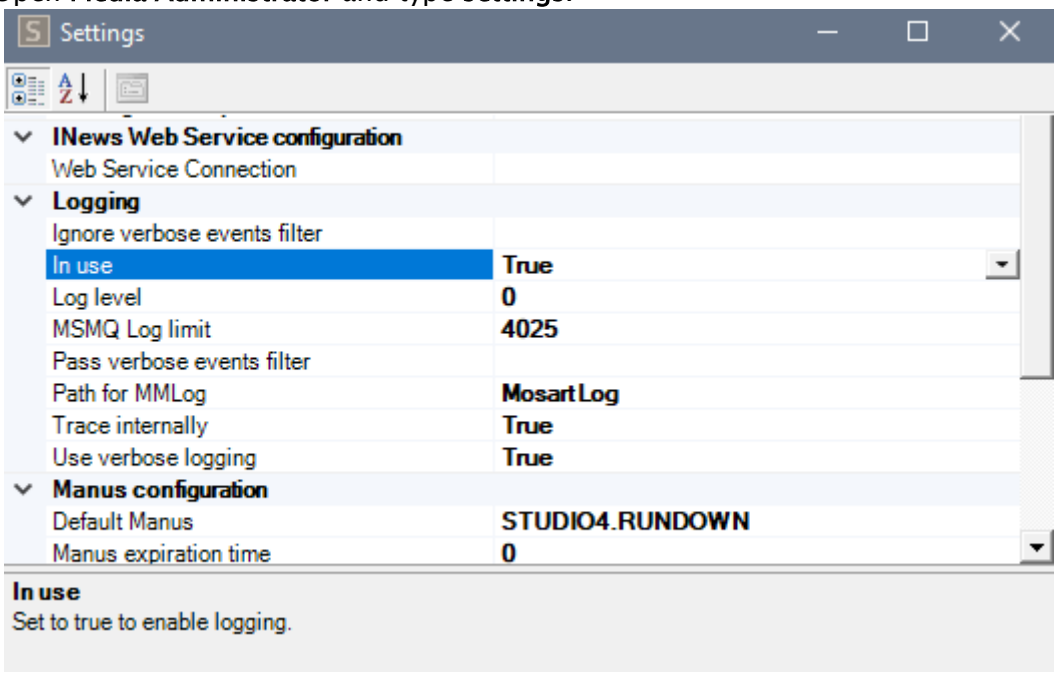
1. Check that there are recorder port(s) configured and enabled for Story Recorder as explained above in To configure video server recorder ports for Story Recorder.
2. Restart AV Automation or force a reconnect to the new video server (use **Ctrl+ Shift + V**). This step should be done regardless whether the previous step was necessary or not.

Failure to Start Recording or Story Items not Recorded in Story Recorder Panel

Recording does not start and no story items appear in the Story Recorder panel.

Copyright © 2024 Vizrt

- Verify in Viz Mosart **Manus Administrator** that logging is enabled:
  Open **Media Administrator** and type **settings**.



- On the machine hosting the Viz Mosart server, check that service *Viz Mosart Logging Service* is up and running (as service or console).
  If running as a service, verify that it points to *MMAsRunLogService.exe* from the Viz Mosart *server installation* folder.
- If the *Viz Mosart Logging Service* is restarted while other applications are running OK, then an **action in the Viz Mosart UI** (for example, reloading the rundown) must be performed to reestablish the connection to the Viz Mosart Logging Service.
  Until a Viz Mosart UI action is made, no events are logged in the AsRunLog for the first story item taken on-air.
  This leads to Story Recorder not functioning correctly. For example, recording may appear to start OK, but no story items show up in Story Recorder panel.
- If the *Viz Mosart Logging Service* is restarted when the Overlay Graphic Interface is being used for controlling the graphics systems, it must be restarted in order to re-establish the connection.
  Until the Overlay Graphic Interface is restarted, no graphic-related log events are logged in AsRunLog for the first graphic encountered in the rundown, meaning Story Recorder will be unable to backtrack the graphic.

> ⚠ **Note**: After any *Mosart Logging Service* (re)start, as a rule of thumb, always restart *AV Automation* and the *Overlay Graphics* interface.

- Make sure *Mosart Logging Service* does *not* run both as service *and* console at the same time.
  To fix, stop both and then start *either* as a console or as a service.

Copyright © 2024 Vizrt

Naming of Final Clip Show from NRCS not appearing in Story Recorder UI

You have set up your NRCS to provide Viz Mosart with a base naming for your final clip produced with Story Recorder, but the naming is not displayed in the Story Recorder panel.

Viz Mosart parses the MOS object corresponding to a story, and on the *first* primary story item found in the rundown it will add the show final clip show given in NRCS.

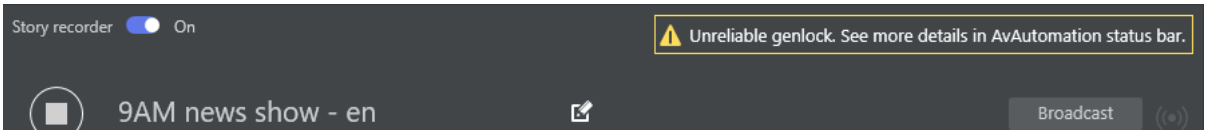
- Examine the underlying XML:
  Do **Ctrl + Shift + Alt + double click** in the Viz Mosart UI on the first story item in the rundown to open this window and verify if the value given from NRCS is present:

If this is not present, here are possible causes:

- The name is read from the *first* story in the NRCS rundown. Check the status of the first story.
  If for example it has been set to *float*, and is not displayed in Viz Mosart, you will need to add the name to whatever story in the NRCS is now the *first* story.
- If another story is *Set As Next* from Mosart UI, then you will need to add the name to this story in the NRCS which is now the *first* story.
- If the Show Final Clip name was changed in NRCS during a Story Recorder session, this will be taken into consideration only at the next Story Recorder session (disable Story Recorder mode and enable again for a new show).

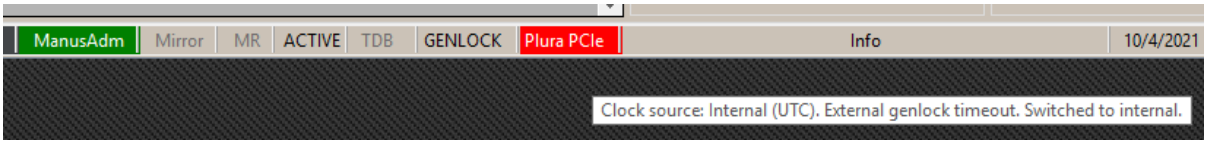
Unreliable genlock. See more details in AV Automation status bar

If this warning is shown in the Story Recorder panel while running a show or in pause mode, this means that Viz Mosart detected problems in the configured clock source (refer to Frame Accurate System Operations) and automatically fell-back to the internal clock source.



The AV Automation status bar will also indicate that there is an issue with the selected clock source:

Copyright © 2024 Vizrt



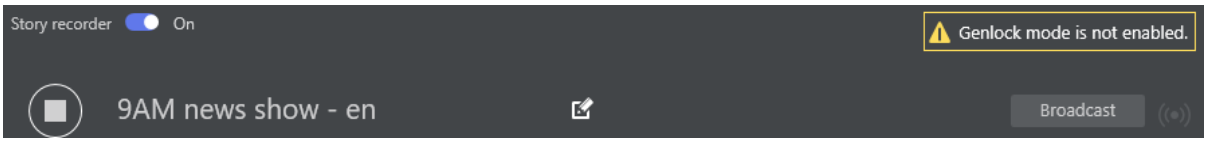
If the warning is shown infrequently, then the message can be safely ignored and the show can continue as planned.

Viz Mosart has a built-in safety mechanism which takes care of intermittent or lost external genlock and/or timecode signal (refer to section **Genlock and Timecode Reliability** in the topic Frame Accurate System Operations).

If the warning appears frequently or it is persistent, the genlock signal and/or timecode signal have become unreliable and must be fixed.

Genlock is not enabled

In the Story Recorder panel, this warning means the genlock has not be activated for Viz Mosart. Refer to topic Frame Accurate System Operations for details on how to activate Genlock mode.



Story Recorder can be used without having proper genlock setup and a timecode reader card installed on the Mosart Server machine, *but only for testing*. In this case, the time of the day will be used.

However, in order for Story Recorder to generate high quality final clip shows, frame accuracy is a requirement.

Retake position no longer valid. Please set retake point

- This is a user error, described in the Viz Mosart User Guide , under topic **Story Recorder Mode**.

Timeline can not be paused while pre-rolling

- This is a user error, described in the Viz Mosart User Guide , under topic **Story Recorder Mode**.

9.2.7 Housekeeping

Viz Mosart automatically performs regular maintenance tasks to archive or delete working folders. For Story Recorder operations, the following are important:

- No management of *local EDL* folder (**EdlLocalFolder,** typically set to *C:\MMLogs\EDL*) and *EDL transcoder* folder (EdlWatchFolder).
  See Enabling the Story Recorder Log Adapter.
  Whilst these files are not large, they will require an occasional purging.

Copyright © 2024 Vizrt

- Viz Mosart can perform some management of the clips generated during a Story Recorder session.
  - Intermediate recording files, prefixed with the string specified in the config RecordingClipNamePrefix (default *SRVizrt)* can be routinely purged from the video server, after a time lapse defined in **RecordingRetentionDays** .
  - When enabled, this task is run every time the service *Mosart Server Log Service* is (re)started .
  - Default value for the number of days to keep files on the video server, RecordingRetentionDays is 'keep forever'.

    ⚠ **Note**: Recorded clips can be *very large* files, easily filling the disk space.

9.3 Viz Engine As A Switcher

9.3.1 Engine Switcher Mode

With Viz Mosart in Engine Switcher mode, you can create an IP-based workflow which eliminates hardware studio switcher/mixers. In this mode:

- Viz Engine performs the function of a switcher/mixer, switching IP inputs and all rendering of DVE effects (box effects, butterflies, picture in picture, etc.) and full screen/overlay graphics.
- Engine Switcher works under exclusive control of Viz Mosart, but is *not* an integrated single unit. In Engine Switcher mode, Viz Engine and Viz Mosart run in their own environments. Viz Mosart templates can then control a broad scope of traditional and unique switcher features.
- On the Viz Engine side, the solution uses Viz Engine DVE based-*superchannels* for live, graphics and clip playback, instead of texture based sources.
  Superchannels can automatically swap sources in an A/B mode with animated transitions, providing further flexibility and gains in overall performance.

  > **Superchannels**
  >
  > **Note**: A *superchannel* is an abstraction on top of a Viz Engine media channel. Superchannels support various asset types like images, graphics, clips, streams, live content. Each superchannel has two sub-channels, so content in each channel can be played out over an A/B workflow. One sub-channel (A) is active, while the second (B) is inactive. New content can be prepared on the inactive sub-channel, a transition effect can then run the transition from the active to inactive sub-channel, switching their roles.

- As standard, the Viz Engine processes the *dirty* feed.
  Optionally, a second Viz Engine can be set up, for a clean feed.

The following topics are presented:

- Engine Switcher Mode
- Features of the Engine Switcher
- Requirements and Limitations
- Installation
- Configuration
- Template Design
- Custom Content Transitions
- Labels
- NDI Clip Preview
- Summary of Configuration Points

Copyright © 2024 Vizrt

> **Note**: For most recent documentation, created after this software was distributed, please visit the most recent version of this page here at the Vizrt Documentation Center.

9.3.2 Features of the Engine Switcher

- Preview and Program: Crosspoint can be set in preview or program (For example: cameras, external sources, clips, full screen graphics).
- Auto-trans: Instructs the switcher to perform the transition from preview to program. Viz Mosart switches to preview and sends the switcher a command to initiate the transition at a specified rate.
  This is typically achieved through a mix with rate > 0, a wipe transition, or an effect transition (using effect scenes) specified on the Viz Mosart template, in the story item provided by the NRCS or from the Viz Mosart UI.

  > **Note**: An *effect* scene is a scene loaded into the Viz Engine's front layer on top of the current switcher scene. These scenes are typically used to animate on top of the actual switcher during a switch.

- Key fill crosspoint set
  The number of key buses correspond to the number of superchannels (DVE box placeholders) supported in Viz Engine.
  The switcher scene available for download contains *8* superchannels by default. You can add more superchannels, to a maximum of 16.

  > **Note**:
  > - While the theoretical amount of sources/superchannels can be up to 16, a high amount of sources loaded on air at the same time might lead to performance issues.
  > - A key bus is represented with a superchannel. The number of superchannels is configured in VizEngineSwitcherConfig.xml.

- *Top Scene* support
  - A *top scene* is a scene loaded on to the Viz Engine front layer, on top of the current switcher scene. This type of scene is typically used for *opener* or *closer* animations for a show.
    Top scenes are called by adding a *top scene* attribute to the Viz Mosart template
  - Load (cue) and play a top scene on the Viz Engine front layer.
  - Cue a scene on-air (using a Viz Mosart *break* template).
  - Take a top scene on air.
- *Preset* support
  - The layouts for BOX effects (split screens, picture-in-picture, etc.) are stored within the scene in a *Presets* plugin. A preset contains the transformation, alpha and cropping attributes for each Viz Engine superchannel and its sub-channels.
  - Set a preset in preview by specifying the preset name and optionally providing content for the preset labels from a Viz Mosart template or from the NRCS.

Copyright © 2024 Vizrt

    - Cue a preset on air.
    - Take a preset on air, either by hard cut or transition. This transition can be specified in either a Viz Mosart template or from the NRCS.
    - Trigger the preset style using a keyboard shortcut.
    - Go to the previous preset style using a keyboard shortcut.
- Clip support
    - Cue a clip in preview.
    - Cue a clip on air (using a Viz Mosart *break* template).
    - Play a clip.
    - Scrubbing.
    - Pause (where camera and adlibs are back-to-back).
    - Up to 4 clips can be cued. Using a 2x ripple, this enables an A to B transition for the single box.

9.3.3 Requirements and Limitations

Requirements

- Viz Engine 5.0+
- Mosart version 5.1+

Limitations

Viz Mosart 5.1 introduced Viz Engine as a switcher. The feature set is being continually expanded in successive versions. Please note the current state of the product:

- No M/Es (Mix/Effects) support other than PP (Program/Preset mixer section).
- No support for macro recall.
- No support for auxiliaries.
- No enable/disable of downstream keyers.
- No cueing of full screen graphics on air.
- No transition logic scenes.
- Overlay graphics are not a standard part of the Engine Switcher workflow. These elements must be loaded on a separate Viz Engine instance/hardware.
- Whilst both the clean and dirty feed can be enabled, the clean feed requires an optional, additional dual-channel setup.
- No main and backup support for Viz Engine from within Viz Mosart.
- Only one clip can be visible (playing in PGM). This can be achieved by either using a key fill crosspoint to play the clip within a DVE box, or via a crosspoint that presents a full screen clip. In addition,
    - Box effects with more than one clip playing *in* the box are not possible.
    - These clips and GFX scenes are always bound to the main *superchannel 1 (SC1)*.
- A preview setup has to run on separate hardware.
  This requires separate PGM and PRV Viz Engines. However, a simple clip preview is configurable.

Copyright © 2024 Vizrt

- Viz Engine supports up to 16 superchannels. This is a theoretical maximum, operations may be limited by potential performance issues.

9.3.4 Installation

Sample Scenes

In addition to a standard Viz Mosart setup, Engine Switcher builds on Viz Engine *scenes* for template-driven operations in Viz Mosart.
To simplify the show design process, sample scenes are provided by Vizrt.

To obtain sample Engine Switcher scenes

1. Authenticate to Vizrt's FTP at https://download.vizrt.com/
2. Navigate to products/VizMosart/Engine Switcher Scenes
3. Download locally, then import the switcher scene to Viz Graphic Hub
4. Set the path to the Switcher Scene in VizEngineSwitcherConfig.xml

> **Note**: For latest installer files and template samples, always refer to the latest version of this documentation at the Vizrt Documentation Center.

9.3.5 Configuration

For first-time configuration, all the setup steps below must be followed:

1. AV Setup with Cross Point Naming and Mapping.
2. Video Switcher Setup
3. Graphics
4. Video Server and Port Configurations
5. Manual Configuration of the Vizrt Media Sequencer Engine (MSE)
   and then to perform
6. Template Design.

AV Setup with Cross Point Naming and Mapping

When Viz Mosart instructs a standard video switcher to put something in preview or on air, it is sufficient to configure AV Config file with a list of crosspoints that correspond to the video signal input configurations defined on the switcher.

In Engine Switcher mode, employing a Viz Engine as a video switcher, these standard rules no longer apply. There are no video signal inputs for the video server ports in a Viz Engine. Instead, the crosspoints on the video server ports in AV Automation must be configured to point directly to the port names configured on the Viz Engine.

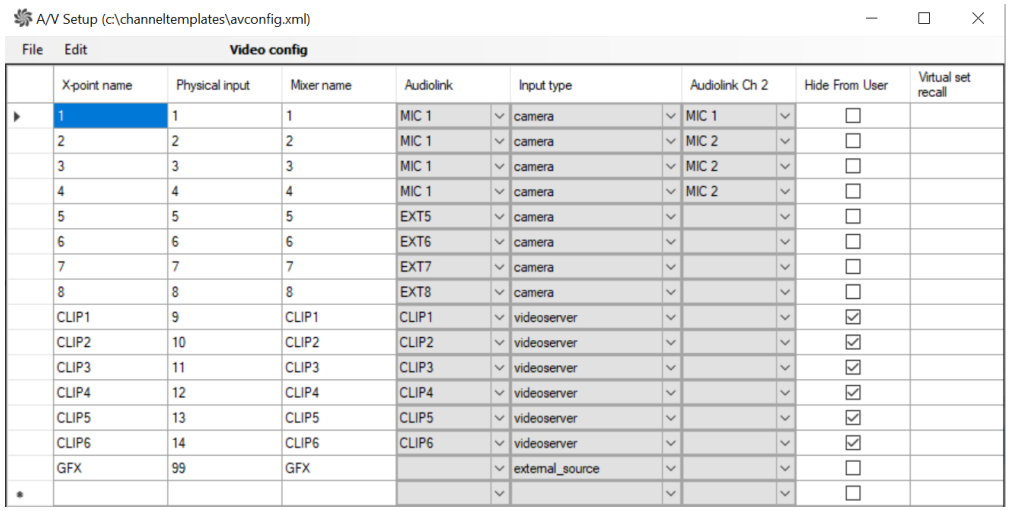
Defining Sources

- Source ID from 1 - N.

Copyright © 2024 Vizrt

- CLIP X-points from 1-6 are mandatory and used for internal switching.
- GFX X-points are mandatory for graphics.

Tip: The X-point name can either be
- the source ID as defined on the Switcher side
  or
- a *Viz Mosart user-friendly name*.
  In this case, then *source ID* (as defined on the Switcher side) should be entered in the **Mixer name** column.

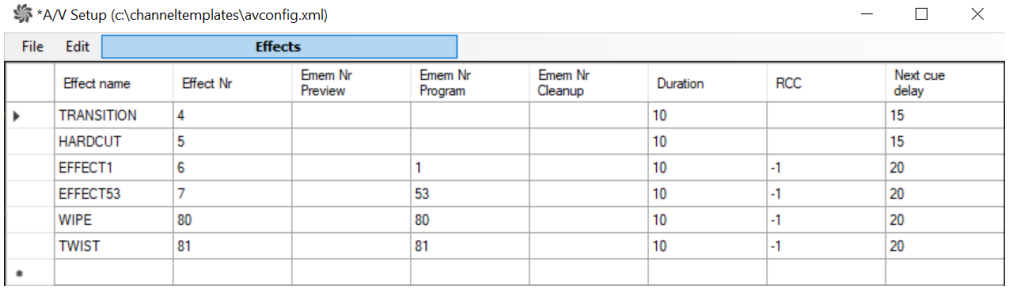


X-point name	Physical input	Mixer name	Audiolink	Input type	Audiolink Ch 2	Hide From User	Virtual set recall
1	1	1	MIC 1	camera	MIC 1	☐	
2	2	2	MIC 1	camera	MIC 2	☐	
3	3	3	MIC 1	camera	MIC 2	☐	
4	4	4	MIC 1	camera	MIC 2	☐	
5	5	5	EXT5	camera		☐	
6	6	6	EXT6	camera		☐	
7	7	7	EXT7	camera		☐	
8	8	8	EXT8	camera		☐	
CLIP1	9	CLIP1	CLIP1	videoserver		☑	
CLIP2	10	CLIP2	CLIP2	videoserver		☑	
CLIP3	11	CLIP3	CLIP3	videoserver		☑	
CLIP4	12	CLIP4	CLIP4	videoserver		☑	
CLIP5	13	CLIP5	CLIP5	videoserver		☑	
CLIP6	14	CLIP6	CLIP6	videoserver		☑	
GFX	99	GFX		external_source		☐	
						☐	

By configuring these crosspoints on the video server ports in AV Automation or referring to them in Viz Mosart templates, these definitions enable standard Viz Mosart clip execution.

Effects

- Effects need a unique *Emem Nr* for calling a specific effect scene.
- Scenes are always prefixed with the string *effect_x* in the designated Graphic Hub effect Scene folder.
  x represents the Emen Nr defined in the configuration.
- Effect IDs must not be higher than *80*, since numbers 80+ are reserved for transition effects within superchannels.



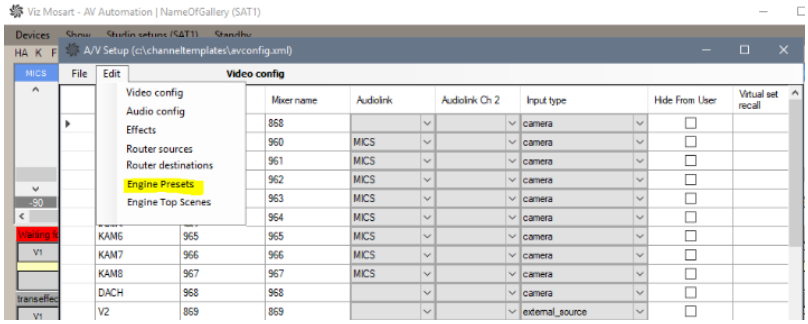
Effect name	Effect Nr	Emem Nr Preview	Emem Nr Program	Emem Nr Cleanup	Duration	RCC	Next cue delay
TRANSITION	4				10		15
HARDCUT	5				10		15
EFFECT1	6		1		10	-1	20
EFFECT53	7		53		10	-1	20
WIPE	80		80		10	-1	20
TWIST	81		81		10	-1	20

Copyright © 2024 Vizrt

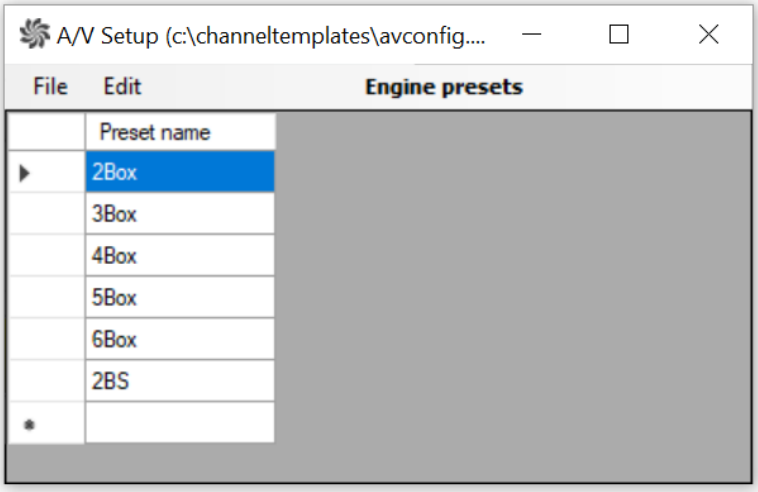
Presets

Tip: The concept *Presets*, is introduced below.

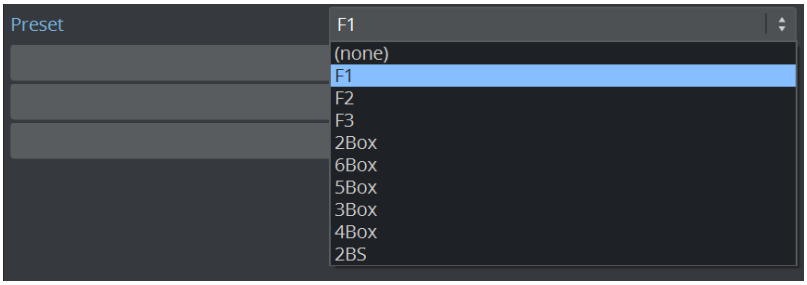
- In Viz Mosart, you see the preset names in the A/V Setup (`AvConfig.xml`) file.



- Preset names correspond directly to the presets defined in the switcher scene:



- A Preset plugin, as it appears in a switcher scene:



Note: F1, F2 etc. are the default full screen options for each superchannel. F1 is *mandatory*.

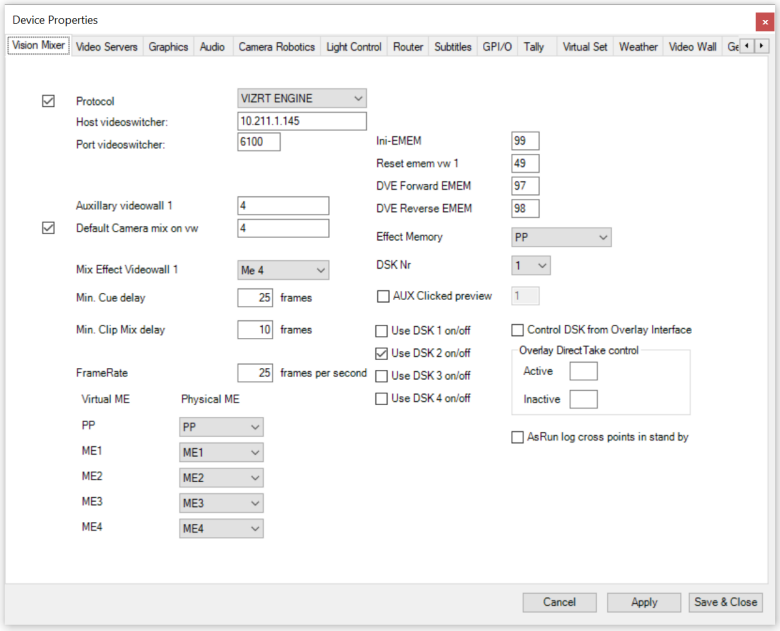
Copyright © 2024 Vizrt

Video Switcher Setup

Video switcher setup is performed in Viz Mosart.

In a Viz Engine Switcher setup, there are no MEs, just PP.

1. Navigate to **AV Automation > Devices > Properties > Vision Mixer** tab.

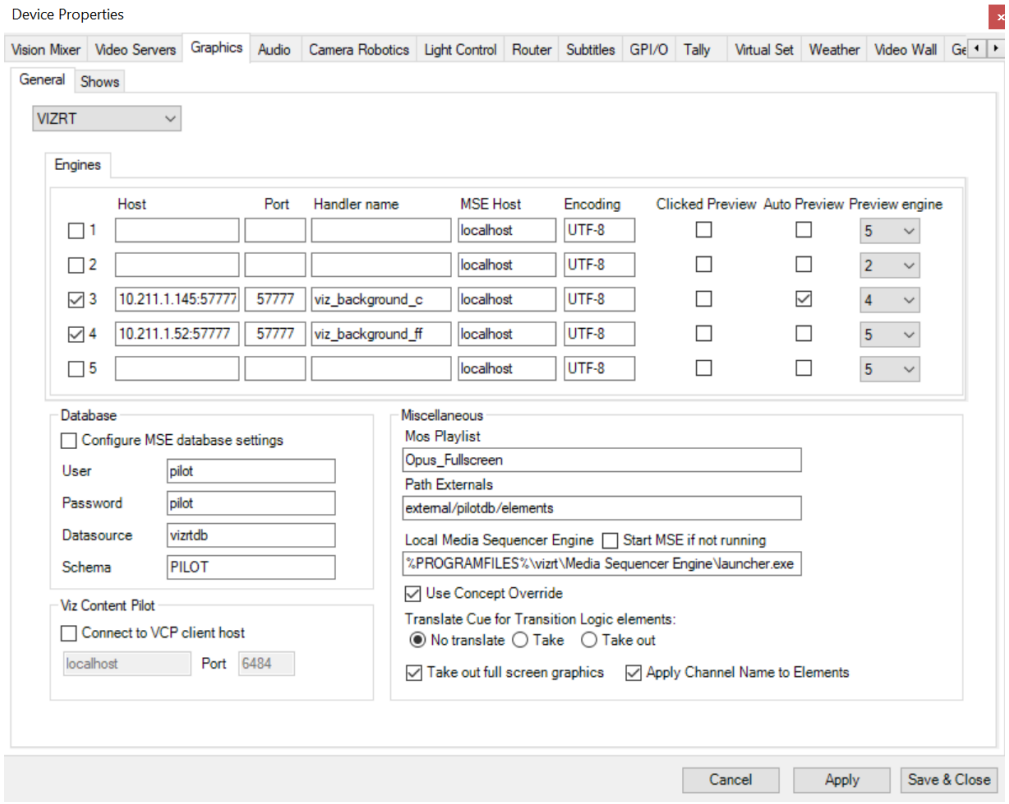


2. From the **Protocol** drop-down menu, select *VIZRT ENGINE*.
3. In addition to the default values, enter the following:
    a. **Host videoswitcher**: Location of the Viz Engine.
    b. **Port videoswitcher**: default `6100`.
    c. Tick the box to the left of **Protocol**. This indicates that you want to control the vision switcher (mixer).
4. Click **Save & Close**.
5. From the *Mosart Installation folder > Mosart Server > Configuration Files* copy VizEngineSwitcherConfig.xml file into your Mosart working folder (*C:\channeltemplates* or *C:\ProgramData\Mosart Medialab\ConfigurationFiles*) and configure:
    a. **ControlScenePat**: The path to the Switcher scene located in the Viz Engine.
    b. **NumberOfKeyBuses**: The number of superchannels defined in the Viz Engine.

Copyright © 2024 Vizrt

Graphics

1. Select the **Graphics** tab.

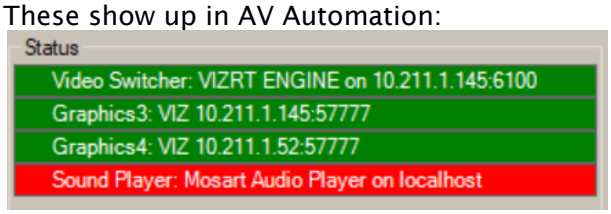


2. In the field **Host**, add the Viz Engine entry for PGM.

> ⚠ **Note**: A second Viz Engine, running on separate hardware, can be added for optional PRV.
>
> - Enable the checkbox **Auto Preview** option and link it to the PGM engine. See Clean and Dirty-Feed Setup.

3. Add the designated *superchannel* port for each Viz Engine (default is *57777*).
4. Tick the **Auto Preview** box for the PGM Viz Engine.
5. From the dropdown box under the **Preview engine** column, assign the identity of the PRV Viz Engine to it.
6. Follow the other guiding values as illustrated above, then click **Save & Close**.

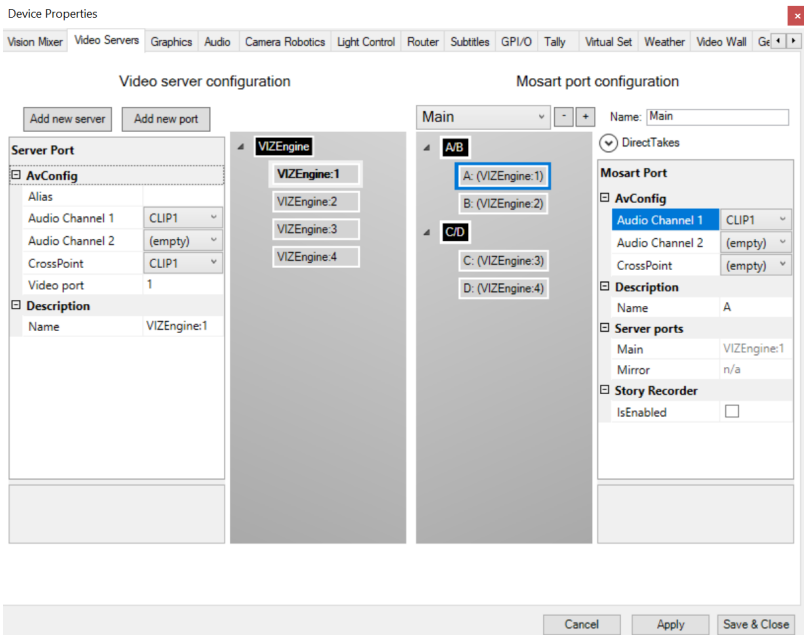
These show up in AV Automation:



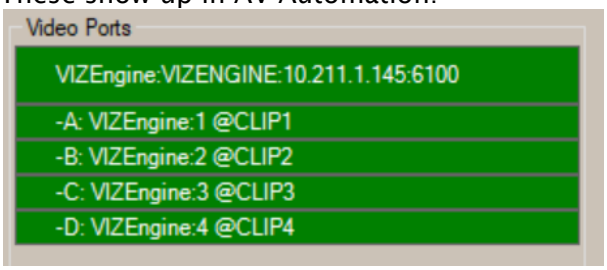
Copyright © 2024 Vizrt

Video Server and Port Configurations

- Select the **Video Servers** tab and add four server ports to create two clip server pairs:
  - A/B
  - C/D



These show up in AV Automation:



Viz Engine Configuration

The following setup is performed on the Viz Engine.

Standard Viz Engine configurations

You will need to provide basic Viz Engine configurations that include definitions of:

- Graphic Hub connection
- Output format
- Licenses
- Amount of inputs.

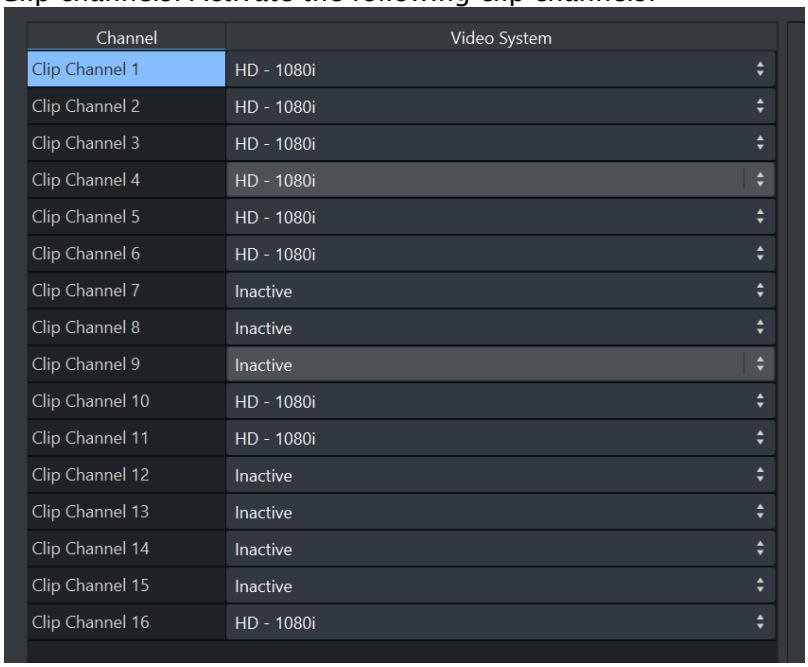
If these are not already in place, refer to the *Viz Engine Administrator Guide*.

Copyright © 2024 Vizrt

Additional Viz Engine configurations for an Engine Switcher setup

1. Navigate to the configuration section **Communication**.
2. Select **Super Channel Port** and set it to an available Port (for example, *57777*)

3. Clip channels: Activate the following clip channels:



- Clip channels *1-4* and *10-11* are for *full screen* and *box* clip playback.
- Clip channels *5-6* are optional for *videowall* integration playback (not included in default setup).
- Clip channel *16* is for clips loaded in *top* scenes.

Switcher Scene Configuration

The Switcher scene comes with a separate configuration file.

- Copy this file to C:\ProgramData\vizrt\VizEngine.

This file enables modification of switching behavior, delays and the adding of a separate Preview engine.

Manual Configuration of the Vizrt Media Sequencer Engine (MSE)

The MSE handlers that you use must be set to *superchannel* mode.

This is performed manually in the VDOM configuration file.

1. Navigate to http://localhost:8580/app/vdomconfig/vdomconfig.html, node */scheduler/<handler name>*. For example, in the AV Automation screenshot above of the **Graphics** tab, the handler name is *viz_background_c.*
2. Add the following entries to both handlers:

Copyright © 2024 Vizrt

```
<entry name="superchannel"/>
<ignore_cue_operation>no</ignore_cue_operation>
<ignore_out_operation>yes</ignore_out_operation>
```

Example of the MSE VDOM File

```
<handler connected="yes" mode="enabled" status="good: Commands processed" type="viz"
 name="viz_background_c">
    <entry name="encoding">UTF-8</entry>
    <entry name="state">
        <entry name="last">
            <entry name="element">/external/pilotdb/elements/406</entry>
        </entry>
    </entry>
    <entry name="renderer">
        <entry name="10.211.1.145:57777">
            <entry name="active"/>
        </entry>
    </entry>
    <superchannel/>
    <entry name="publishing_point_uri"/>
    <entry name="publishing_point_atom_id"/>
    <entry name="info"/>
    <entry name="superchannel"/>
    <ignore_cue_operation>no</ignore_cue_operation>
    <ignore_out_operation>yes</ignore_out_operation>
    <command_timeout>30.</command_timeout>
</handler>
```

9.3.6 Template Design

A show with video switching performed by Engine Switcher, is controlled exactly like any standard video switcher-driven show, from a rundown with associated Viz Mosart templates.

Tip:

- For an overview of template design with examples, please refer to the section Template Examples, section Switcher Plugin.
- To download sample scenes, see To obtain sample Engine Switcher scene.

Introduction to Presets, Top scenes and Effects

Viz Engine Switcher operations depend on distinct criteria, controlled from a Viz Mosart template. All settings are made during template design, as described in the section Template Editor. Three concepts for Engine Switcher operations are:

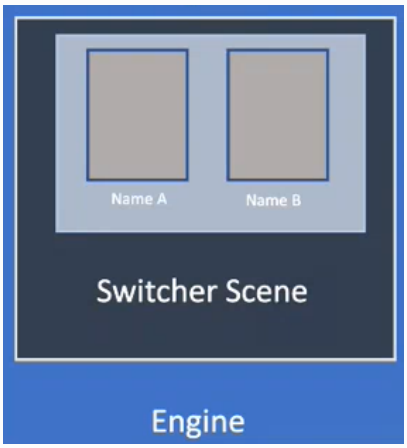
Copyright © 2024 Vizrt

- Presets
- Top Scenes
- Effect Scenes.

Presets

Info: The term *preset* originates from scene design in the Viz Engine. The composition of the scene is defined by x,y positions of a collection of 'windows' (two in the illustration below) in a Viz Engine *superchannel*. These definitions describe:

- The *number* of windows
- The p*osition* of the windows
- The size of each window.

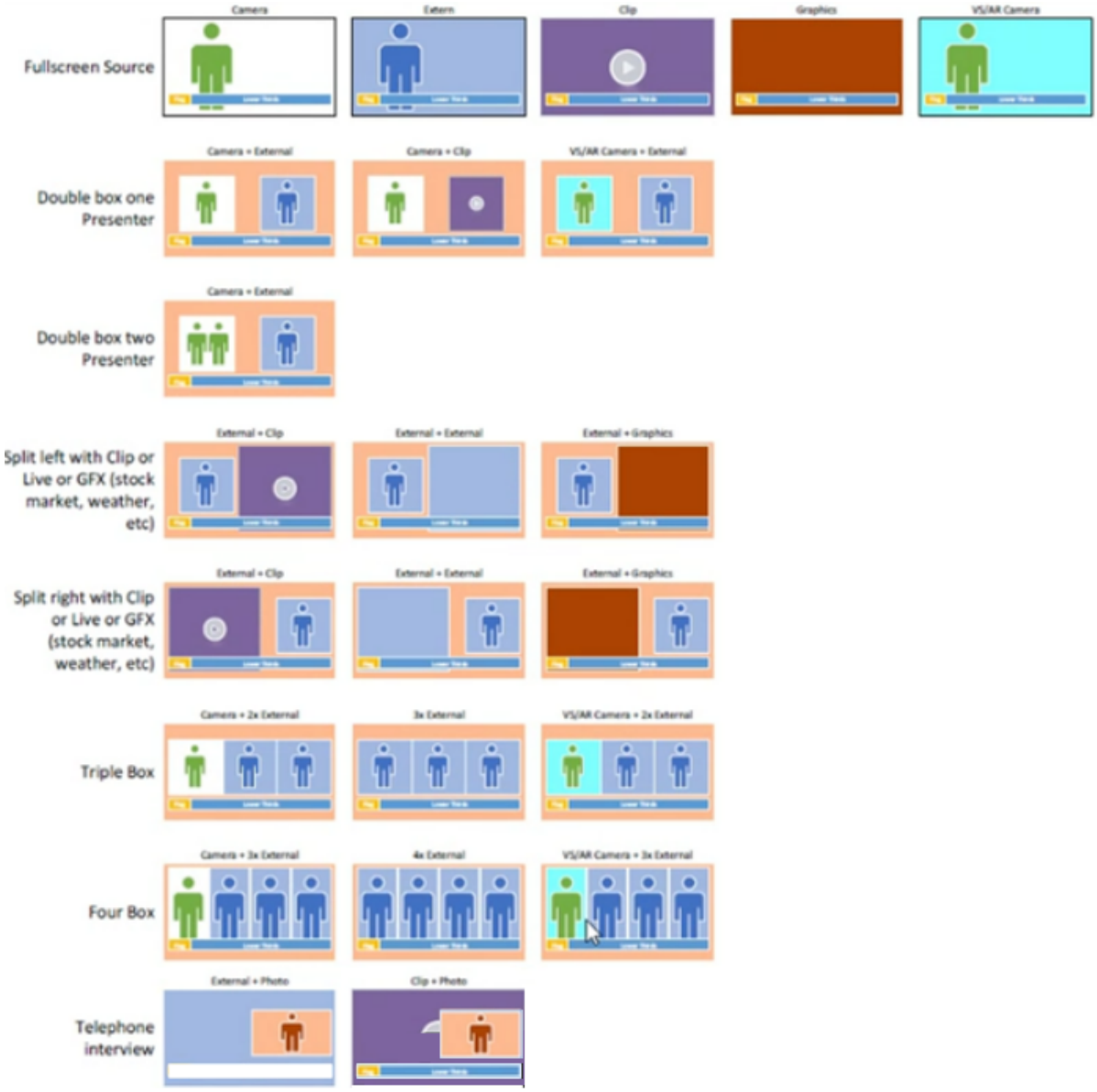


Each composition, saved on the Viz Engine, is called a *preset*.

Note: Superchannel 1 is considered the main superchannel for full screen program switches. The preset *F1* is mandatory for this workflow and must *not* be altered.

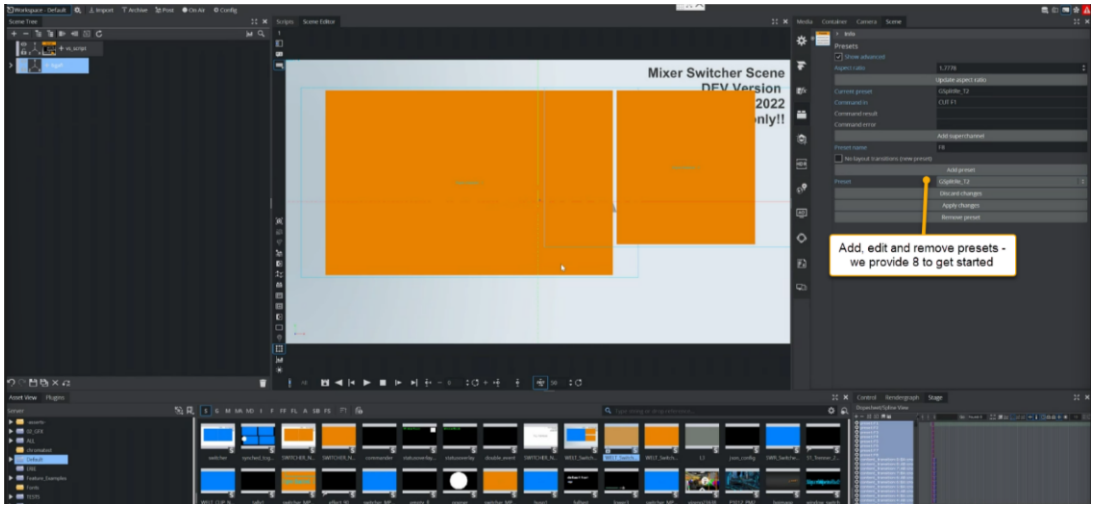
Viz Mosart includes a sample Switcher scene with eight presets.

Copyright © 2024 Vizrt



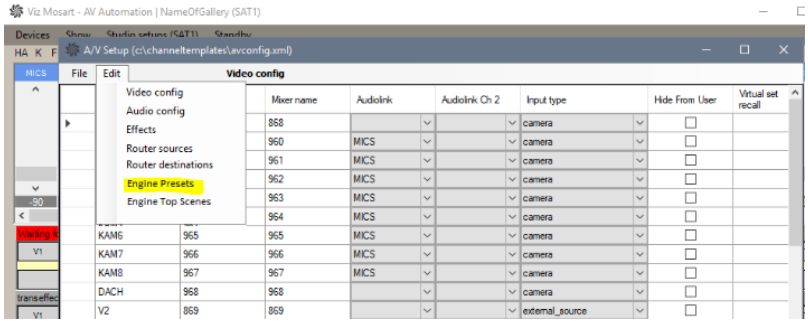
Copyright © 2024 Vizrt

- Use Viz Artist to edit any preset.



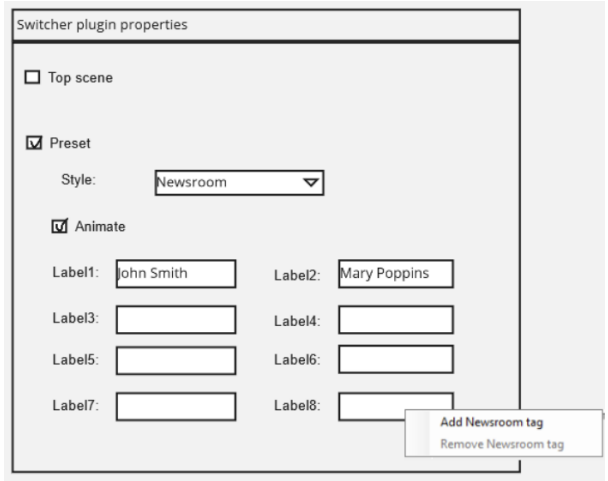
For more details, refer to the *Viz Artist User Guide*.

- In Viz Mosart, you select the Preset name from the list of preset names that have been configured in A/V Setup (AvConfig.xml) file.



- Transition style

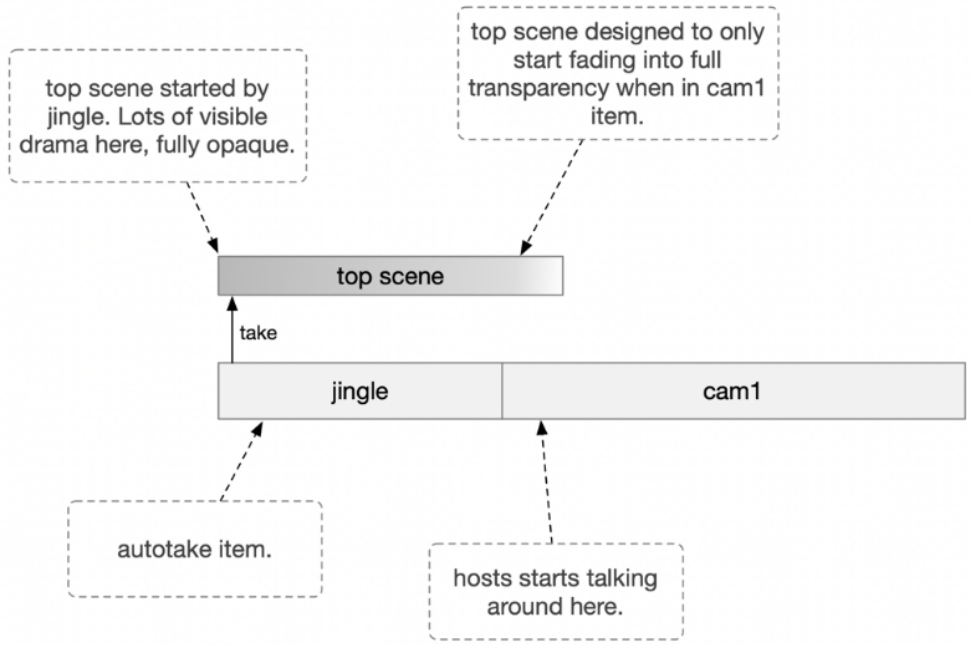
  The transition *style* (hard-cut or animation), and several *label* values can be specified. For example, in a DVE template (where **Switcher plugin** for the device is available), you can select the layout and look of the DVE effect by specifying a preset style (**Style**), the preset transition and the DVE window labels (**Label1** - **Label8**), as required:



You can also provide static values directly on the template. Additionally, newsroom tags can be configured to enable control from the NRCS.

Top Scenes

A typical use of top scenes is as an *opener*. For example, the top scene is played to introduce the Eight O'clock News show with many visual effects that fade to the following camera scene, when the news host takes over.

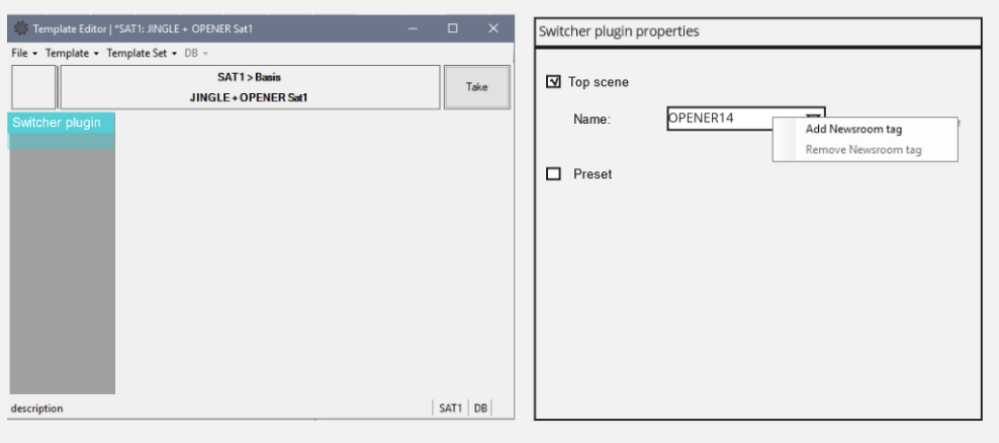


To Create a template to play a top scene

1. Select an appropriate template, for example *Graphic* or *Jingle*, corresponding to your Engine Switcher-controlled operation.
2. Enable **Switcher plugin** for the device.
3. From the **Switcher plugin properties** menu, specify a top scene name:
    a. None (**Top scene** unchecked)

Copyright © 2024 Vizrt

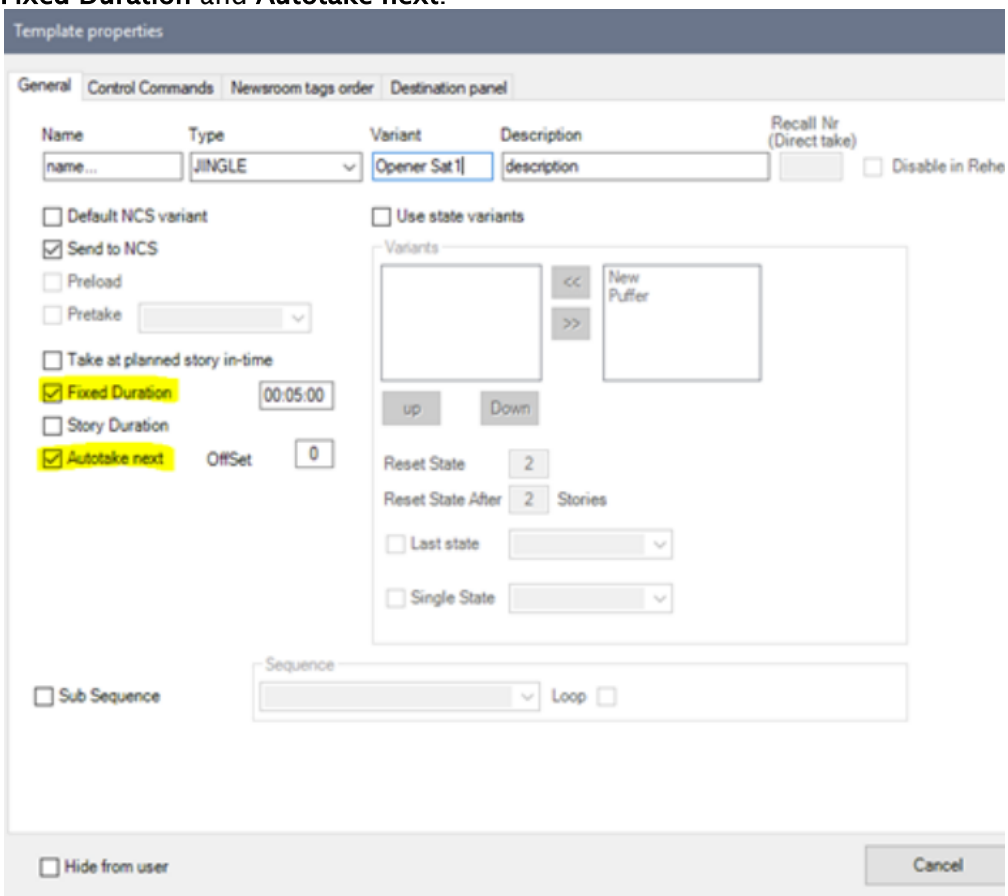
b. Check **Top scene** and select from the drop-down list of top scenes (configured in the A/V Setup).



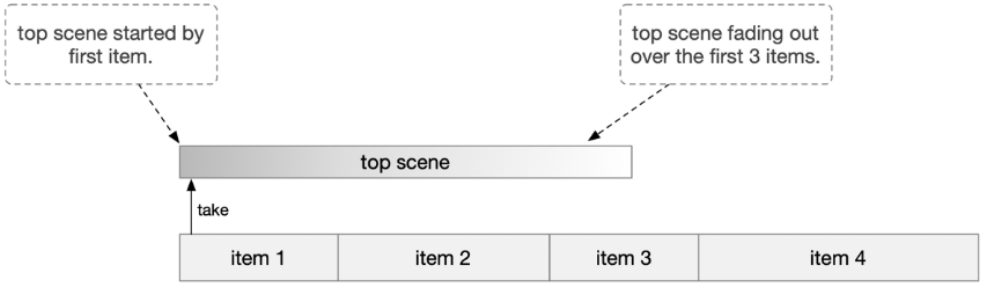
Time-related configurations

There are two time-related aspects for top scenes (similar to handling clips):

1. The time it takes for the scene to *load*. This is controlled automatically by Viz Mosart when cueing the story item.
2. The time it takes for the scene to *run*. This is controlled with the template configurations **Fixed Duration** and **Autotake next**.



- Top scenes are started from within a story item and if long enough, continue to play even when the item ends.
- Top scenes ignore story items boundaries. The take out of a story item must match the animation length of the top scene.
- Top scenes can be played on top of anything else.



To create a template that plays a **Viz Engine Top scene**

The sample script below (code block) performs this operation:

1. Initially the scene is fully opaque, covering the switcher completely.
2. On *Take,* an animation starts.
3. After a pre-defined time, the openers content disappears (animates or fades out).

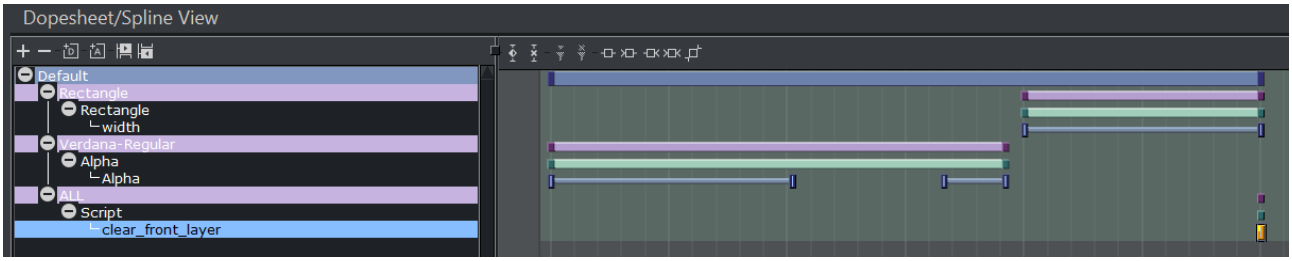
The Engine Switcher workflow does not need to contain a script; it is purposefully designed for simplicity. A script *can* be added, to call the "clear_front_layer" routine of the switcher script which will force-remove the top scene from the front layer after the animation is finished.

Example Script

```
'-------------------------------------------------------------------------------------------
-----------------------------------
sub OnInitParameters()
    RegisterPushButton("clear_front_layer", "Opener Close", 1)
end sub
'-------------------------------------------------------------------------------------------
-----------------------------------
sub OnExecAction(buttonId As Integer)
    if buttonid = 1 then clear_front_layer()
end Sub
'-------------------------------------------------------------------------------------------
-----------------------------------
sub clear_front_layer()
    dim commandStr as string = "RENDERER*MAIN_LAYER*SCRIPT INVOKE clear_front_layer"
    println 5, "--------------- TOP SCENE OUT ----------------"
    System.SendRemoteCommand("localhost", 6100, commandStr, 0)
end sub
'-------------------------------------------------------------------------------------------
-----------------------------------
```

In this case a **clear_front_layer** button is added to the animation in the Stage

Copyright © 2024 Vizrt



Effect Scenes

Effect scenes, similar to Top scenes, are loaded to the front layer of the Engine Switcher for an upcoming switch between sources. These scenes are loaded automatically once a transition effect is triggered in Viz Mosart. Scene names must be added to AV configuration and are typically named "effect_<id>".

Example Effect

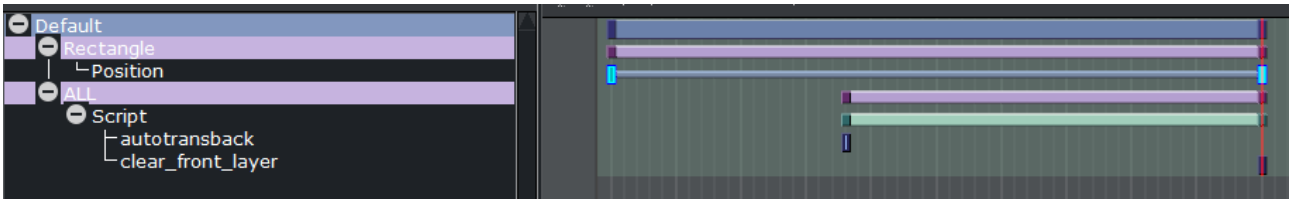
1. The scene is fully transparent at the beginning of its animation.
2. At *Take,* an animation starts to show content, animating to fully cover the scene underneath it (switcher).
3. After a predefined time, the animation reaches the *autotransback* script routine, which triggers the switch in the Switcher Scene.
4. The animation fades out, being fully transparent again.

Script

Two routines, one for triggering the take (auto_trans) and one for its removal when the animation ends:

```
sub OnInitParameters()
    RegisterPushButton("autotransback", "Auto trans back", 1)
    RegisterPushButton("clear_front_layer", "Effect Close", 2)
end sub

sub OnExecAction(buttonId As Integer)
    if buttonid = 1 then autotransback()
    if buttonid = 2 then clear_front_layer()
end sub


sub autotransback()
    dim commandStr as string = "RENDERER*MAIN_LAYER" & "*SCRIPT INVOKE auto_trans"
    println 5, "---------------- TOP SCENE TRANS FROM FL SCENE ----------------"
    System.SendRemoteCommand("localhost", 6100, commandStr, 0)
end sub

sub clear_front_layer()
    dim commandStr as string = "RENDERER*MAIN_LAYER*SCRIPT INVOKE clear_front_layer"
    println 5, "---------------- TOP SCENE OUT ----------------"
    System.SendRemoteCommand("localhost", 6100, commandStr, 0)
end sub
```

Copyright © 2024 Vizrt



Note: Using effects scenes on a *separate/external* Viz Engine.

- Edit the script to send the RemoteCommand for "autotransback" to the Engine Switcher. In this case the IP needs to be manually changed to the correct external one in the effect scene.
- Keep in mind that using an external engine for triggering effect transition switches might lead to timing inaccuracies due to possible network delays.

9.3.7 Custom Content Transitions

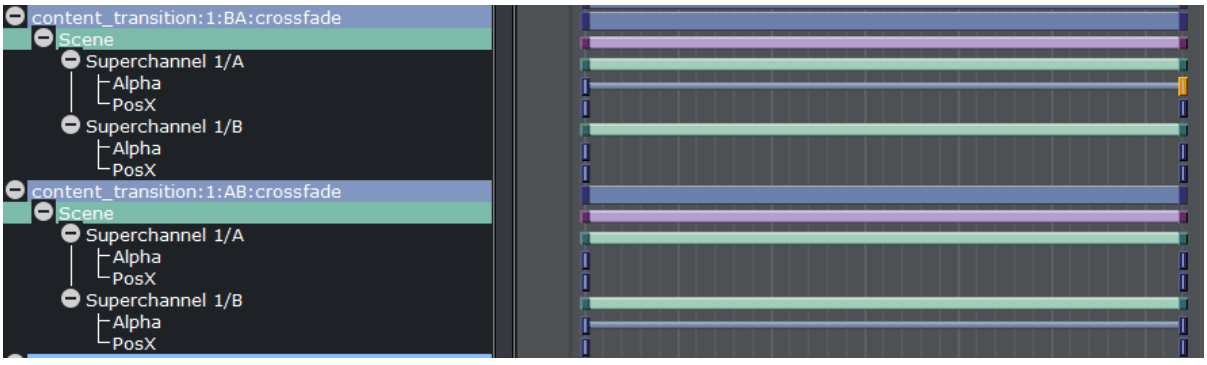
This section describes the actions required in Viz Artist, when working with Engine Switcher-driven Viz Mosart templates.

Note: Content transitions are only used for *full screen* content transitions. Transitions in between Boxes / Splits are always hard cut.

Preset Plugin

The Preset plugin creates a default content transition for a crossfade. Additional transitions for changing content within superchannels are created manually using Viz Artist.

Default Preset plugin crossfades



- A custom content transition is saved to a special director name, called by the Switcher from Viz Mosart.
  This director can contain the required transition animation.
- Typically, a content transition director contains the keyframes related to the superchannels sub-channels *A* and *B*.
- Content transitions are created in both directions. A separate director is used for a transition between A to B, and B to A.

Copyright © 2024 Vizrt

To define a director

- A director has the following naming format:
  `content_transition:<ID>:<direction>:<transition_name>` where
  - ID: Superchannel ID
  - direction: *AB* or *BA*
  - transition_name: Name the transition that will be called in the UI (for example *wipe*).

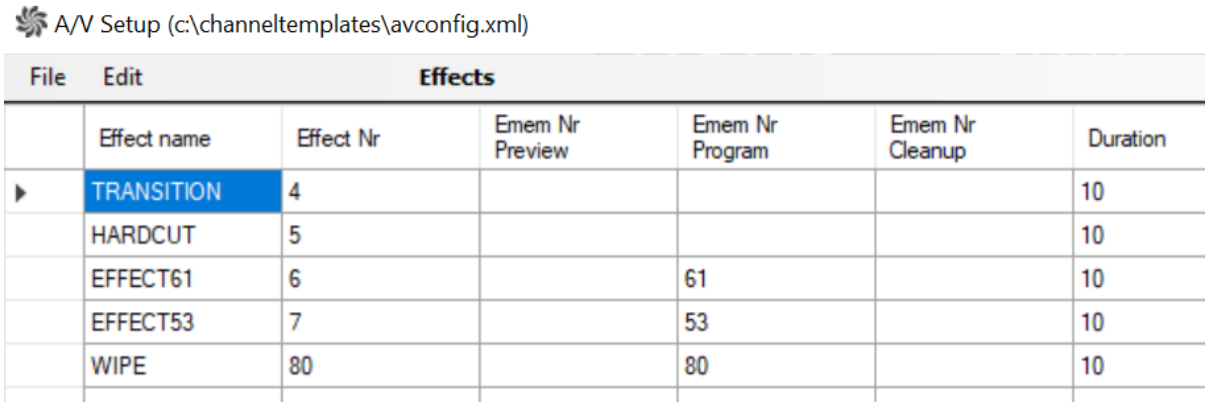
Custom Effects

- Custom effects scenes are manually added to the Engine Switcher's configuration file.

> **Note**: Custom effects IDs start at *80*.

To create an Effect

1. In Viz Mosart Mosart AV Automation:



	Effect name	Effect Nr	Emem Nr Preview	Emem Nr Program	Emem Nr Cleanup	Duration
▶	TRANSITION	4				10
	HARDCUT	5				10
	EFFECT61	6		61		10
	EFFECT53	7		53		10
	WIPE	80		80		10

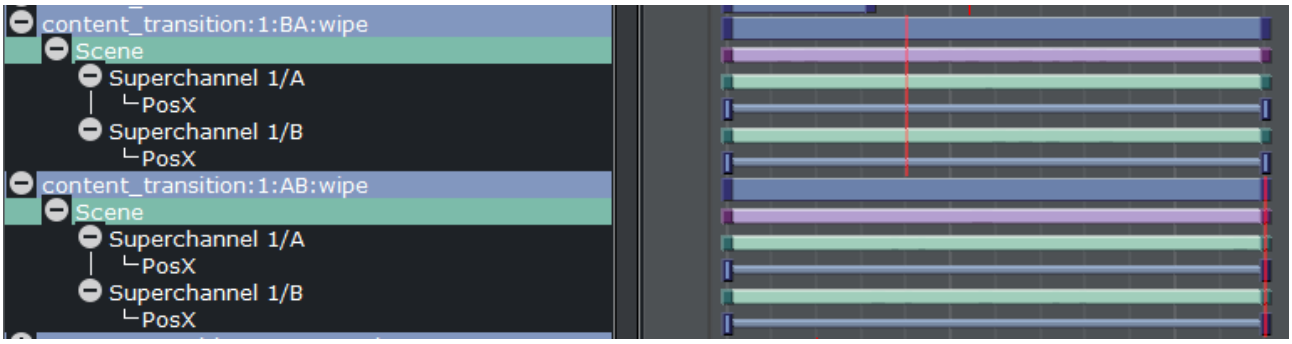
2. In the Viz Engine Engine Switcher Configuration file, C:\ProgramData\vizrt\VizEngine\switcher_config.cfg

```
#custom content transition effects
<content_transitions>80:wipe,81:rotate,82:twist</content_transitions>
```

3. The effect is now available for selection in Viz Mosart:

Example - Effect 80 - Wipe:

A typical wipe is set up by animating the subchannel's X position from left to right, then having the other subchannel follow.



Values:

(1) Direction AB (content_transition:1:AB:wipe):

- Assuming sub-channel 1/A is visible and should fade out to the right
- Keyframes for Pos X are set to
  - Superchannel 1/A:
  - Start : 0.0
  - End : 100.0
- Sub-channel 1/B starts on left to 1/A and animates onto the screen
  - Superchannel 1/B:
  - Start : -100.0
  - End : 0.0

(2) Direction BA (content_transition:1:BA:wipe):

The roles are now switched, this means 1/B starts on the screen, animates to the right and 1/A now appears from the left.

Copyright © 2024 Vizrt

- Keyframes for Pos X are set to
    - Superchannel 1/A:
    - Start : -100.0
    - End : 0.0
- Sub-channel 1/B starts on left to 1/A and animates onto the screen
    - Superchannel 1/B:
    - Start : 0.0
    - End : 100.0

9.3.8 Labels

Label names (newsroom tags) and their content, defined in a Viz Mosart template and filled out in the NCRS, are stored in *shared memory* when the story item containing the box template is cued.

When a new DVE box is cued, the designer dynamically reacts to any label change. These changes can either come from a script included in the Switcher scene background or from another scene located on another networked Viz Engine.

Label Format

The format of the received string comprises label names and label values. Each label pair is separated with a "|" character, giving:

```
<label_name1>:<label_content1>|<label_name2>:<label_content2>|
<label_name2>:<label_content3>|<label_name4>:<label_content4>...
```

Example: Labels populated from a DataPool plugin

The default map name is *MIXER_LABELS*. This can be changed within the Engine Switcher Configuration.

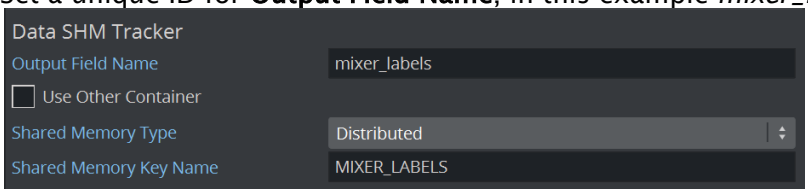
In this example you use a Vizrt **Datapool** plugin to access the content of the shared memory and deliver its content to a specified text container.

- An *SHMTracker* plugin reads the memory content.
- The content is forwarded to any other **Datapool** plugin within the scene.
- Here, *ataText* receives the memory content from *SHMTracker* and parses the values into a readable format
- The content is pasted into the geometries **Text** plugin.

Copyright © 2024 Vizrt

1. Create a container, name it *labels*.

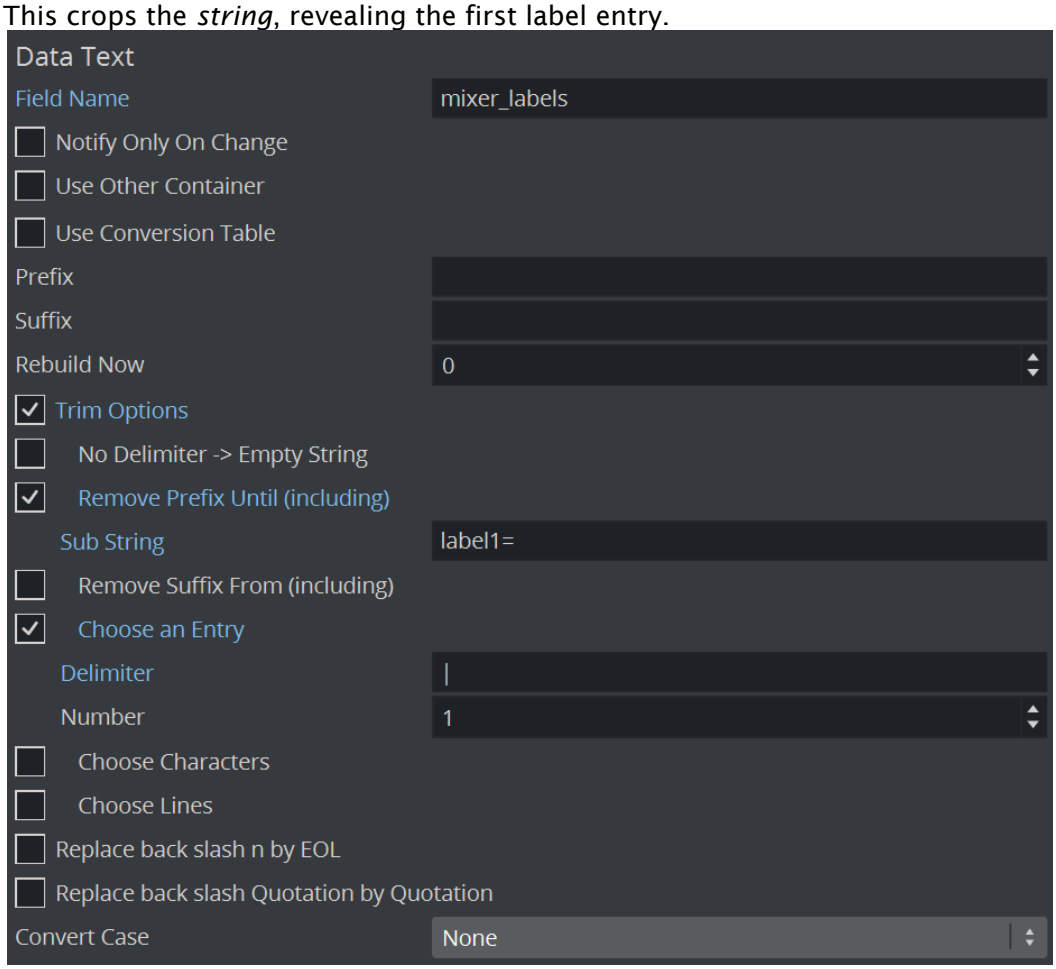
2. Add the *SHMTracker* plugin to it.
3. Set a unique ID for **Output Field Name**, in this example *mixer_labels*.



4. To access the Switcher scenes used SHM, set **Shared Memory Type** to *Distributed*.
5. Assign **Share Memory Key Name**. Default is set to *MIXER_LABELS*.
6. Create a child container and name it *label1*.
7. Add the *Data Text* plugin.
8. Set **Field Name** to the same *ID as* used in the parent plugin. In this example, *mixer_labels*.
9. In **Rebuild Now**
    a. Check **Trim Options**.
    b. Check **Remove Prefix Until** and enter **Sub String** *label1=*
    This crops the *label* from the incoming string.
    c. Check **Choose an Entry**.

Copyright © 2024 Vizrt

i. Set **Delimiter** to |
This crops the *string*, revealing the first label entry.



10. Repeat these steps for all required labels, modifying the sub-strings accordingly to *label2=* etc.

Example of a container script

The default map name is *MIXER_LABELS*. This can be changed within the Engine Switcher Configuration.

This simple script example shows how to read in the content of a shared memory map, parsing the content for *label name* and *content* and pasting it onto an existing text container.

The given example reacts on a mapping change that typically is triggered from cueing the next *Box* item.

```
'-------------------------------------------------------------------------------------
-----------------------------------
sub OnInit()
    VizCommunication.map.Registerchangedcallback("MIXER_LABELS")
end sub

sub OnSharedMemoryVariableChanged(map As SharedMemory, mapKey As String)
```

```
    dim labels as String = VizCommunication.map["MIXER_LABELS"]

'parsing and reacting routines
    Parse_Labels(labels)
    Set_Labels()
end sub

sub Parse_Labels(input as String)
    Dim params as Array[String]
    Dim content as Array[String]

    input.Split("|",params)

    For each param in params
        param.Split(":",content)
        labels[content[0]] = content[1]
    Next
end sub

'setting labels (container name equals label name)
sub Set_Labels()
    dim label_keys as Array[String]
    labels.GetKeys(label_keys)

    for each key in label_keys
        This.FindSubContainer(key).Geometry.Text = labels[key]
    next
end sub

'-------------------------------------------------------------------------------------
-----------------------------------
```

Note: For more advanced use cases, please contact Vizrt Support.

Note: The actual toggle event of the next story item can be accessed by registering on the the *<toggle_status>* map, as defined in the Engine Switcher Configuration.
This allows the designer to trigger animations related to an upcoming change of the layout.

9.3.9 NDI Clip Preview

If a simple preview of the upcoming clip layout/switch is needed and a separate Viz Engine cannot be used, you can instead add a small nested scene to the Switcher scene. This can stream the next cued clip via NDI (Network Device Interface, a protocol).

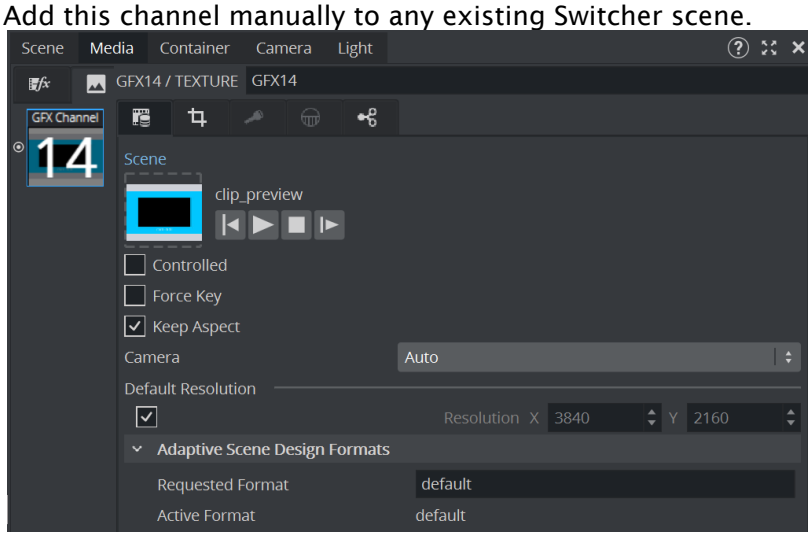
Note: The *Media Preview* feature used in Viz Opus is no longer supported.

Copyright © 2024 Vizrt

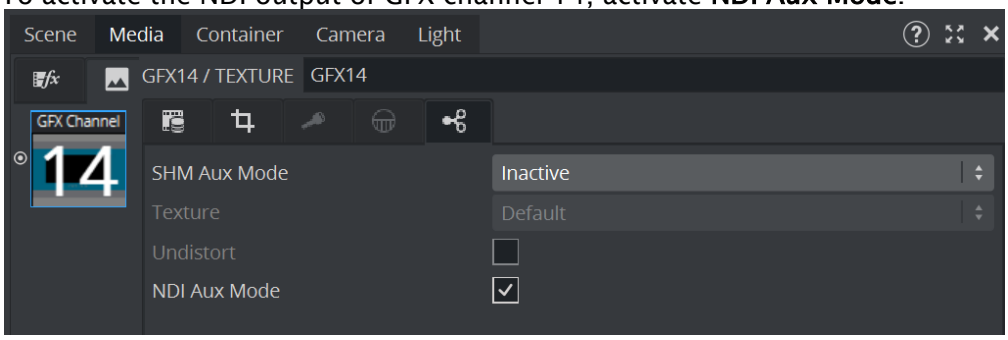
To stream a preview with NDI

Engine Switcher is configurable to send clip cueing commands to a separate scene if needed.

1. Add the scene to a GFX channel within the Engine Switcher scene.
   By default, GFX channel 14.
2. Add this channel manually to any existing Switcher scene.



3. To activate the NDI output of GFX channel 14, activate **NDI Aux Mode**.



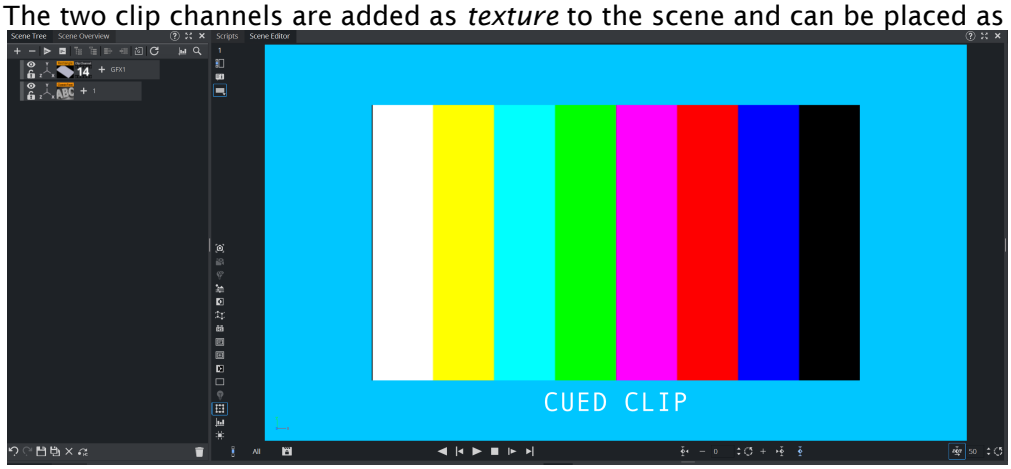
GFX channel 14 will now be received as a separate NDI stream on the network.

Previewing a scene
The preview scene setup is simple.

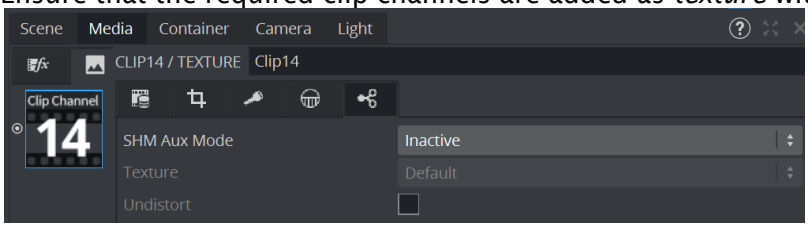
4. Clip channel 14 and 15 are used to load the next cued *full screen* or *box* clip.
   The layout of the scene itself does not matter and requires no further logic.

Copyright © 2024 Vizrt

5. The two clip channels are added as *texture* to the scene and can be placed as required.

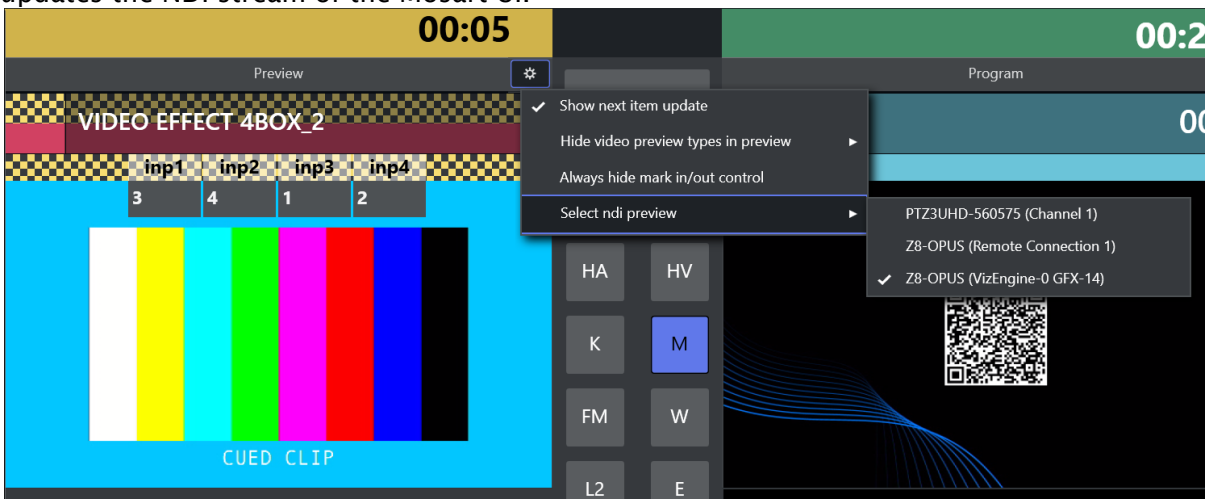


6. Ensure that the required clip channels are added as *texture* within the scene.



Setting up Viz Mosart

7. In Viz Mosart, select the source in the Preview window.
8. As soon as a clip is cued as next item, the GFX channel scene loads the paused clip. This updates the NDI stream of the Mosart UI.



9.3.10 Summary of Configuration Points

Viz Engine Configuration

- C:\ProgramData\vizrt\VizEngine\VizEngine-0.cfg
  (**Note**: Here -0 refers to which GPU will be used).

Copyright © 2024 Vizrt

> **Tip**: A copy of this config is included in this Administration Guide under Device Configuration Files > Engine Switcher Configuration File.

Switcher scene configuration

> The separate config file "switcher_config_<versionNumber>.cfg" **must** be renamed to drop the versionNumber, so that it just reads "switcher_config.cfg".
> For example, if the default name is *switcher_config_0.cfg*, this file must be renamed to *switcher_config.cfg*.

- C:\ProgramData\vizrt\VizEngine\switcher_config.cfg

Viz Media Sequence Configuration

- localhost:8580/app/vdomconfig/vdomconfig.html

Viz Mosart

- C:/channeltemplates/VizEngineSwitcherConfig.xml
- C:/channeltemplates/ClipServerVizEngine.xml

> **Note**: VizEngineSwitcherConfig.xml and ClipServerVizEngine.xml are originally located in the Viz Mosart installation folder C:\Program Files (x86)\Mosart Medialab\Mosart Server\ConfigurationFiles.

See Also

- Vision Mixer

Copyright © 2024 Vizrt

9.3.11 Clean and Dirty-Feed Setup

By default, Engine Switcher is set up to show only a combined dirty feed output. A clean feed configuration must be set up manually.

Clean and dirty feed

- A video signal that does not have any graphics or effects embedded, is called a *clean feed*.
- A video signal that has graphics or effects enabled is called a *dirty feed*.

- Working with Clean and Dirty Feed
  - Engine Switcher Terminology
  - Hardware Configuration
  - Software Configuration
  - Configuration Example
  - Viz Mosart Overlay Graphics Configuration
  - Additional Software Configuration

- Running **two** Viz Engine instances on the Engine Switcher server enables production of both dirty and clean feed.
- This setup will use one program output on the clean feed and one video input on the dirty feed.
- The clean feed signal is routed back into Engine Switcher through the Matrox video card, using the zero-frame delay at one of the video inputs.
  Zero-frame delay at the input prevents any additional delay to the final program signal.
- All overlay graphics and transition effects are then added on the clean feed signal, producing the dirty feed.
- The dirty feed is then sent out on two program-out channels.

Note: Clean and Dirty feed are rendered by 2 instances running on the *same* GPU. As resources are shared, creating a heavy load on one instance might affect the performance of the second one.

Working with Clean and Dirty Feed

Engine Switcher Terminology

Clean Feed

The clean feed provides:

- Full screen video live sources
- Full screen clip sources

Copyright © 2024 Vizrt

- Full screen GFX elements
- Butterfly effects (PiP effects)

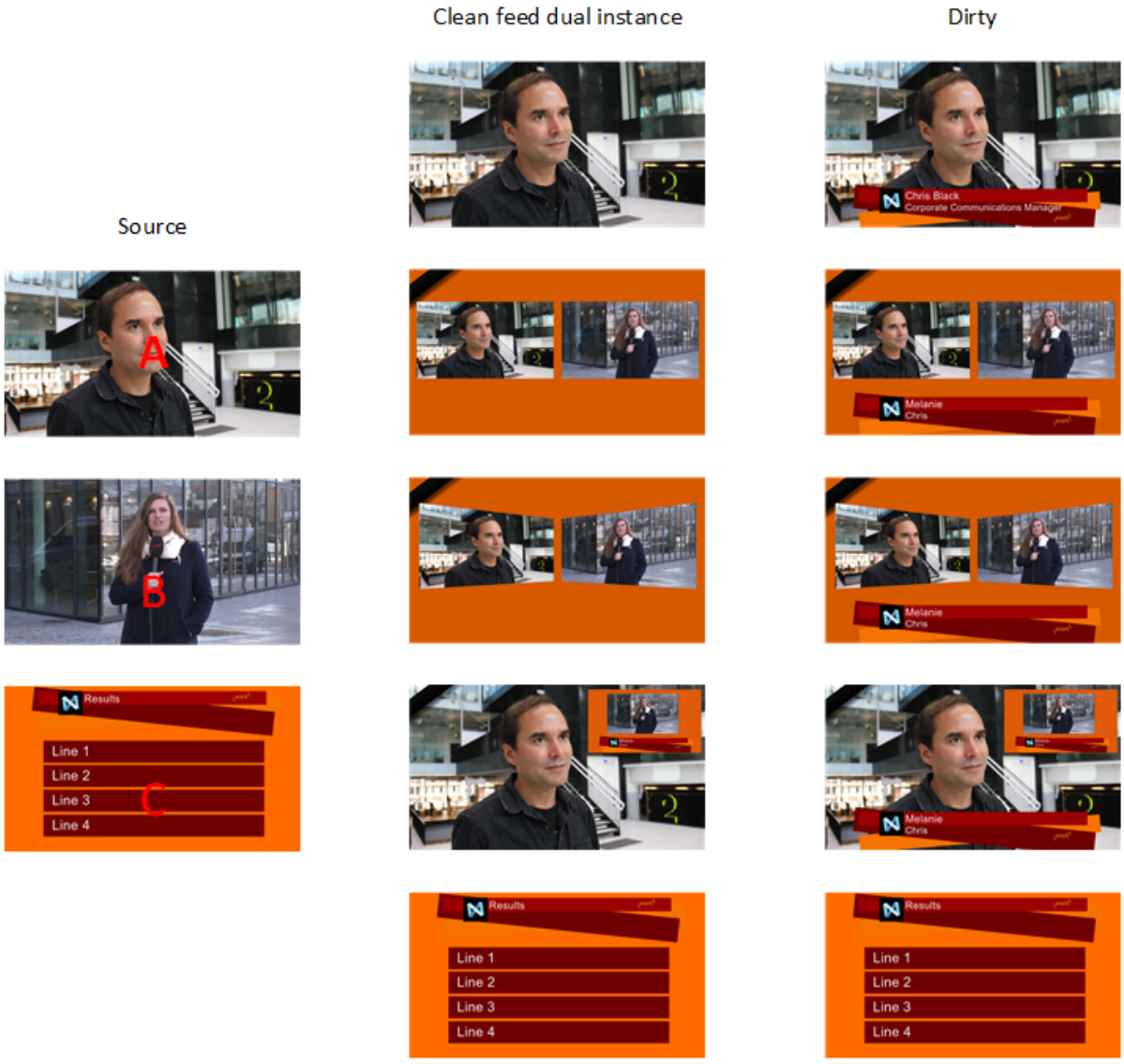
Dirty Feed

The dirty feed adds overlay graphics on top of the clean feed output.

	Clean Feed Engine	Dirty Feed Engine
Front Layer	Transition Effects	Overlay scenes
Main Layer	Full screen GFX	Overlay scenes
Back Layer	Not used	

Copyright © 2024 Vizrt

Example: Output Of Clean And Dirty Feed.



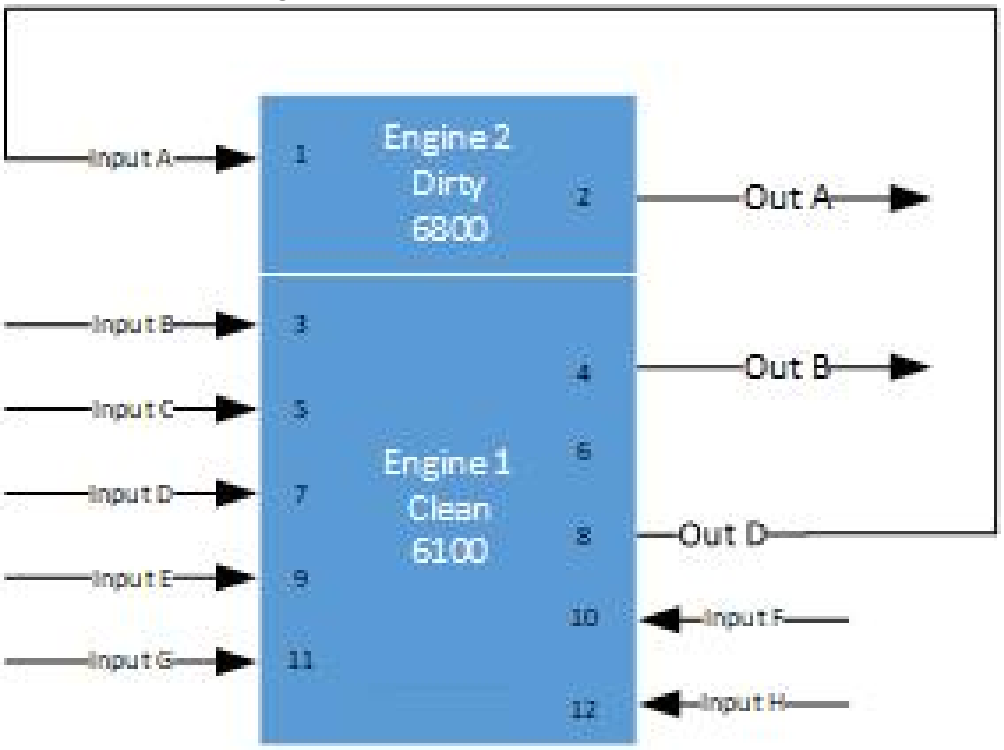
Hardware Configuration

The current clean feed solution needs a hardware wiring using a short SDI cable from the SDI OUT D (pin 8 on the X.mio3) to SDI IN A (Pin 1). Engine Switcher is using the "Zero Frame Delay" option within Viz Engine 5.1. This provides a direct signal copy from Input A to Output A without introducing an additional delay. Therefore there is no need to load a keyer scene to the dirty feed engine.

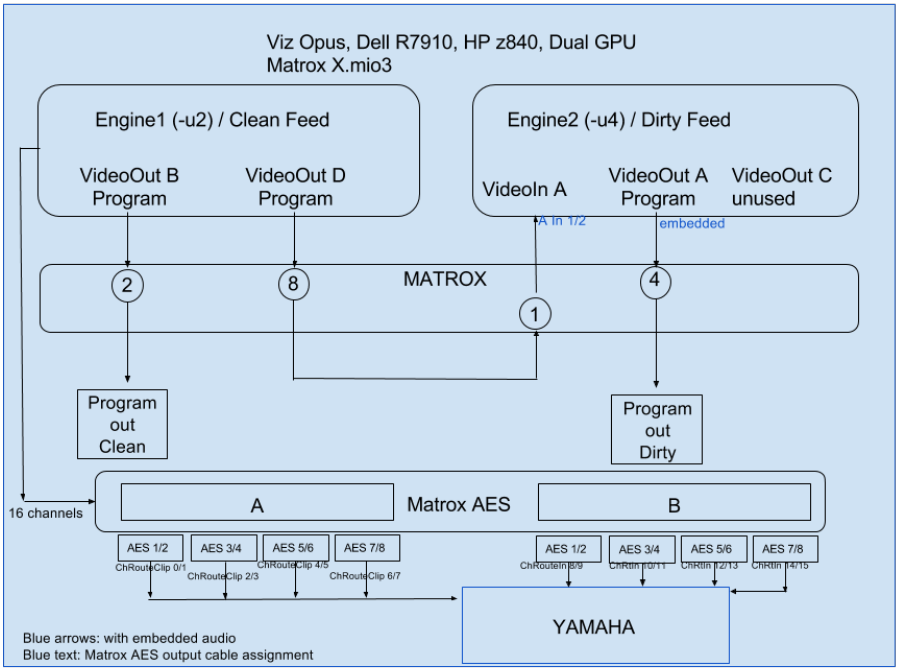
Copyright © 2024 Vizrt

To Configure The Matrox Board

- Matrox Pin Out for Engine Switcher clean and dirty feed setup (Zero Frame delay):



- Routing of video signal on the Matrox board.



Copyright © 2024 Vizrt

Software Configuration

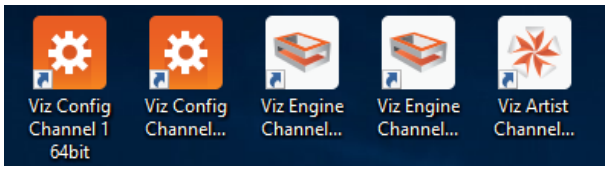
This sets up Viz Engine in dual channel mode.

To Configure The Viz Engine In Dual Channel Mode

1. Install Viz Artist and select **Dual Channel** mode.

   > **Tip**: To create the necessary shortcuts, refer to the *Dual Channel Mode* section in the latest version of the Viz Engine Administrator Guide (here as an example, version 5.1).

   Dual Channel installs and adds these icons to the desktop:

2. Replace the default configuration files with those provided on the Vizrt FTP.
   Files are in the folder /Product/VizMosart/Engine Switcher Scenes/
3. Rename them according to the computers name:
   a. Engine 1 - clean feed -> <machinename>-0.cfg
   b. Engine 2 - dirty feed -> <machinename>-2.cfg
4. Modify the dual channel shortcuts (different scenarios related to GPU count mentioned above).
   For Media Preview to work in Engine Switcher the clean feed instance needs to run in *console mode* (no Viz Artist).
   a. Configure the desktop shortcuts properly by adding parameters:
      i. -x (no UI, *mandatory*)
      ii. -u1 or -u3 depending on the graphics board used.
         1. -u1 points to the first engine on the first GPU (used for clean feed)
            a. this uses -> <machinename>-0.cfg
         2. -u3 points to the second engine on the first GPU (used for dirty feed)
            a. this uses -> <machinename>-2.cfg
      iii. -n for engine mode
      iv. -c config mode
      - u1 or -u3 for config shortcut of engine 1 or 2.
5. Port Configuration.
   Make sure that the ports are set correctly. This should be set when using the
   a. Clean feed runs on port 6100
   b. Dirty feed runs on port 6800.

Configuration Example

To Configure SIngle GPU Dual Engine Setup

Since all engines do use the same GPU, the "-u" parameter in the shortcut is only relevant to point the engine to the right config file. A clean- and dirty channel setup typically will use:

1. **Engine 1**: Clean feed
   a. <machinename>-0.cfg
   b. Desktop shortcut uses "-u1"

Copyright © 2024 Vizrt

2. **Engine 2** : Dirty feed
    a. <machinename>-2.cfg
    b. Desktop shortcut uses "-u3"

To Configure Dual GPU Dual Engine Setup

Typically the Engine Switcher's dual engine setup comes with two GPUs. The first card is used for connecting the monitor, the second for actual rendering purposes.

Config files and desktop shortcuts will have to be named as follows:

1. **Engine 1**: Clean feed
    a. <machinename>-1.cfg
    b. Desktop shortcut uses "-u2"
2. **Engine 2**: Dirty feed
    a. - <machinename>-3.cfg
    b. - Desktop shortcut uses "-u4"

> **Note**:
> - The digit behind the "u" depends on how the graphics board is installed in the machine.
>   If there are 2 GFX boards, the weaker one is used for the monitor, not for rendering. This weaker board is typically not addressed.
>   So the second video boards gets -u2 (clean) and -u4 (dirty).
> - Since Viz Artist always runs on the *first* GPU, a third config file named <machinename>-0.cfg must be employed, in case changes in the scene have to be performed on this particular scene.
> - As Viz Artist will run on the weaker GPU, performance of the scene while being loading on this Engine will be limited.

Output Format

Please note that the video format must be manually modified for the affected config files.

- The default is set to **1080i50.**

To Set The Output Format

Changing the format is most easily done by loading the appropriate configuration profiles.

1. Start the Configuration Tool for the clean feed Viz Engine.
2. Click **Load** and choose **Installed Profile**.
3. Navigate to C:\Program Files\vizrt\VizEngine\Configuration Profiles and choose the correct option
    - video-resolution-1080i50-audio-xmio3.cfg
    - video-resolution-1080i60-audio-xmio3.cfg
    - video-resolution-720p50-audio-xmio3.cfg
4. Save the configuration and close the Tool.
5. Start the Configuration Tool for the dirty feed engine and repeat steps 2-4.

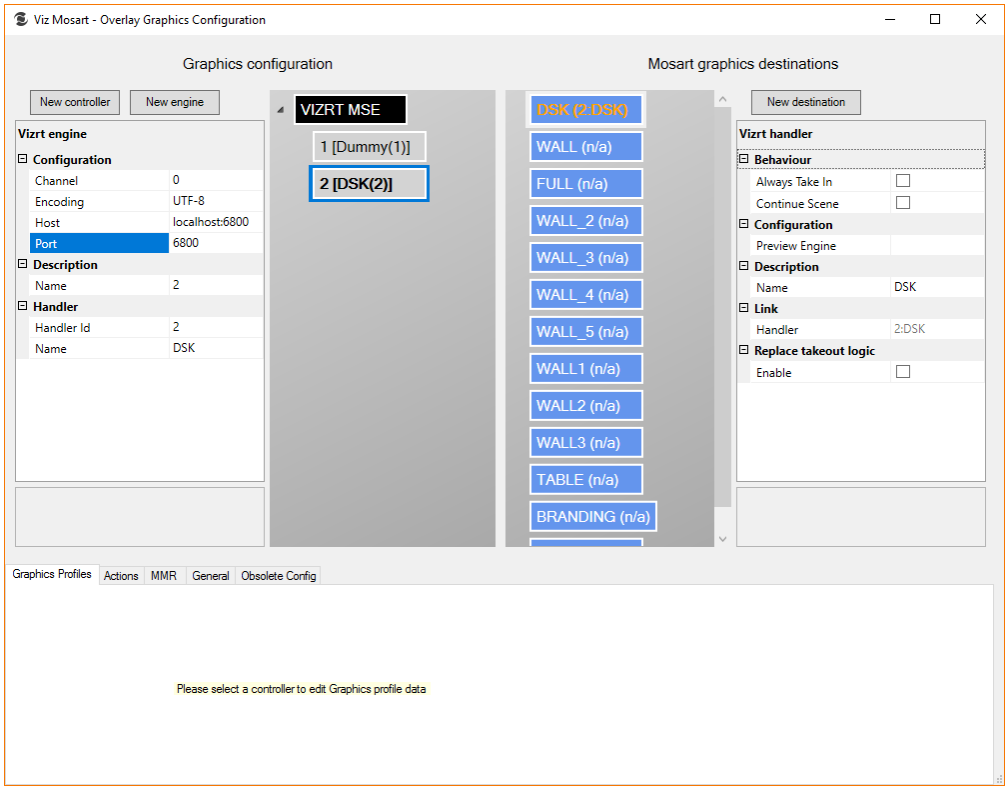
Copyright © 2024 Vizrt

Viz Mosart Overlay Graphics Configuration

To Set Up Overlay Graphics Configuration

The Overlay Graphics applications configuration needs modifying, to send Overlay scenes to the dirty feed on Port 6800 (instead of the default port 6100).

1. From the Viz Mosart server machine, open the Mosart Graphics Interface application (MMOverLayGraphicsInterface.exe).
2. From the menu bar, select **Settings > Properties**.
3. Fill-in as illustrated:



4. Restart the applications to save changes.

Additional Software Configuration

Configure the Viz Engine in dual channel mode, as described in the *Viz Engine Guide*, section *Software Configuration > Dual Channel Mode*.
Whilst this can initially be installed as a single engine, you will still need to configure the shortcuts and adjust the Viz Config.

Execute all three procedures:

1. Set dual channel mode (Viz Engine)
2. Configure clean feed
3. Configure dirty feed

Copyright © 2024 Vizrt

To Set Dual Channel Mode (Viz Engine)

1. For default configuration of a dual channel, load the dual channel profiles for Channel 1 (Clean) and 2 (dirty). This will set the correct Ports.
   C:\Program Files\vizrt\VizEngine\Configuration Profiles
   - dualchannel -0.cfg
   - dualchannel -1.cfg
2. Load the correct audio profile on both channels via the configuration tool
   C:\Program Files\vizrt\VizEngine\Configuration Profiles
   - video-resolution-1080i50-audio-xmio3.cfg

To Configure Clean Feed

The clean feed configuration should be based on the file provided in the bundle installer.

1. Configure **Program** and **Contains Alpha** to *Inactive*.
   See Video out documentation to configure the program output of the engine to output program (fill) on VideoOutB and VideoOutD assigned by setting the Map to Viz channel t for reference.
2. Configure the video inputs.
   a. Start by disabling video 1 by setting VideoInA to Unused.
   b. Set all other to the corresponding Video number (VideoInB to Video2, VideoInC to Video3 etc).
3. Changes done with the configuration tool are stored in `<machinename>-1-0.cfg`.
4. **Video** (Matrox board configuration)
   a. Mapping Live Outputs:
      i. Set Out B and Out D to *Program*
      ii. Set Out A and Out C to U *unused*
      iii. Set In A to *unused*
   b. Mapping Live Inputs:
      i. In A set to *unused*
      ii. In B set to *Live 1*
      iii. In C set to *Live 2*
      ...
      iv. In H set to *Live 7*
   c. Activate Live 1 to Live 7 in Input config.
5. **Audio**
   a. Audio Mode is set to AES
   b. AES audio routing for clips needs to be done directly in the config file.
   Default settings for 2 separate audio channels:

ChannelRouteClip0_0	0
ChannelRouteClip1_0	2
ChannelRouteClip0_1	1
ChannelRouteClip1_1	3

Copyright © 2024 Vizrt

To Configure Dirty Feed

See the *Viz Engine Guide*, section *Software Configuring Viz > Video Output* for reference.

1. Changes should be done uses the Configuration Tool.
2. Configure the program output of the engine to output program (fill) on VideoOutA assigned:
3. Set **Map to Viz channel** to *Program*
4. Set **Contains Alpha** to *active*.
5. Configure the video inputs by setting all of them to *Unused*.
6. If not already set by the preloaded dual channel profiles, set communication **General communication port** on this engine to *6800*.
7. Verify that the changes are stored in `<machinename>-3-0.cfg`.
8. **Video (Matrox board configuration)**
    a. Mapping Live Outputs
        i. Set Out A to Program
        ii. Set Out B, Out C and Out D to *Unused*
    b. Mapping Live Inputs
        i. Set all Inputs to *Unused*
9. **Audio**
   Audio on the dirty feed is *not* supported by default.
   Routing of the AES channels is performed manually in the Viz Engine's configuration file.
10. **Additional settings**
    - Turn on *Zero Frame Delay* option.

Copyright © 2024 Vizrt

9.4 Media Sequencer To Power Vizrt Graphics

This section introduces the most recent developments for graphics control managed by Vizrt's middleware component, Media Sequencer (MSE). It also introduces main/backup switching of the Media Sequencer.

- Configuring Vizrt Graphics
- Main/Backup Topology

9.4.1 Configuring Vizrt Graphics

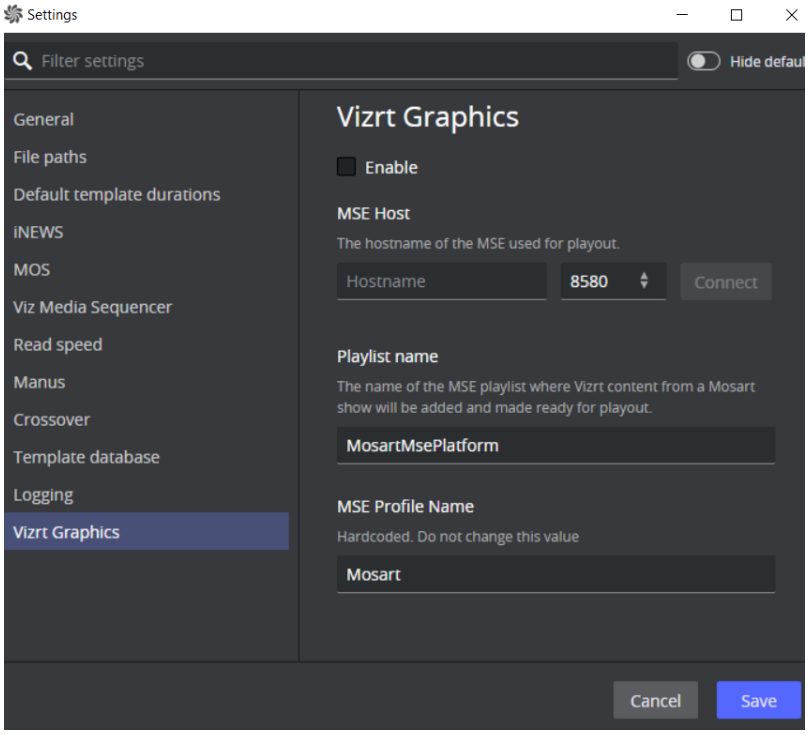
A Viz Mosart driver communicates with the MSE. MSE-driven functionality is being continually improved to control various graphic elements.

- From the AV Automation menu bar, navigate to **Devices > Properties > Vizrt Graphics tab** and follow the **Vizrt Graphics** link.

The current supported graphics are listed below:

Viz Arc Elements

- Use the **Settings** menu to set up Vizrt Graphics.



Copyright © 2024 Vizrt

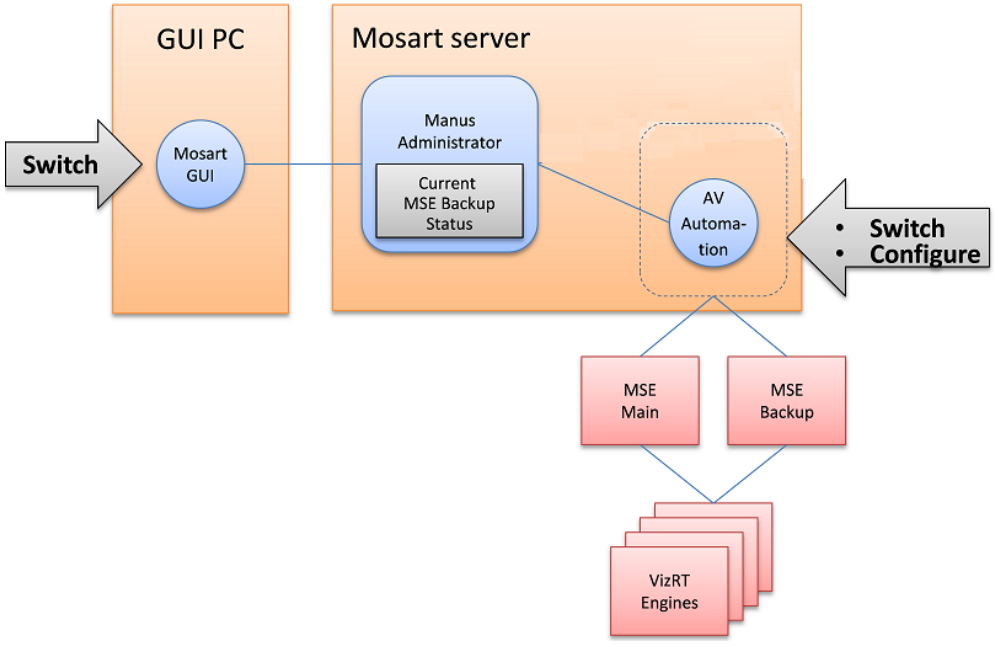
9.4.2 Main/Backup Topology

Selecting the Media Sequencer

- It is possible to select the *main* or *backup* Media Sequencer, from Viz Mosart. See Media Sequencer to Power Vizrt Graphics.
- There may be a delay of several seconds after switching from one to the other, before the Media Sequencer responds to Viz Mosart commands.

Configuration Alternatives

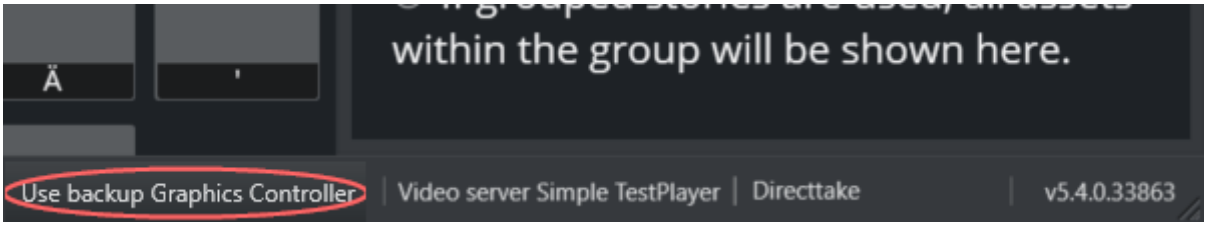
The diagrams below show the configuration and operations possibilities of the various components in the Media Sequencer backup concept.
Gray arrows show which operations are available in each Viz Mosart module.



To configure MSE backup

- See section Redundancy.

Switching Media Sequencer from the Viz Mosart GUI



Switching between main and backup Media Sequencer is normally done in the Viz Mosart GUI.

To Switch Media Sequencer

- At the bottom right corner of the Viz Mosart UI is the text: **Use backup Graphic Controller**.
  This text is clickable and switches Viz Mosart to the backup Media Sequencer.
  Once switched, a warning text on yellow background, **Use main Graphics Controller** visibly alerts to the exceptional state.

During the switching process, the alarm indicators in AV Automation momentarily go red, but return to green when the switching process is finished.

> **Note**: The Media Sequencer requires several seconds to initialize all parts, so it is recommended to wait some time before continuing graphics operations.
> The exact delay varies with graphics size and complexity.

Copyright © 2024 Vizrt

9.5 Audio Players

- Default Viz Mosart Audio Player
  - Supported File Formats
- Third Party Audio Player Devices
  - Setting up a SpotOn Audio Player
  - Setting up a TriCaster Audio Player
- Working with Audio Players
  - Working with the Viz Mosart Audio Player
  - Working with a Third Party Audio Player

A Viz Mosart user can play out audio files using an audio player. This is useful, for example for playinwindow for configuring the Audiog regularly used audio, like openers and audio-beds.

A basic audio player is always included as standard in a Viz Mosart. You can also install third party audio drivers (for example *SpotOn*) as standard player, as described below.

- User actions are presented in the Viz Mosart User Guide in section *Playing Audio*.

Requirements

- Refer to general section Prerequisites.

9.5.1 Default Viz Mosart Audio Player

Supported File Formats

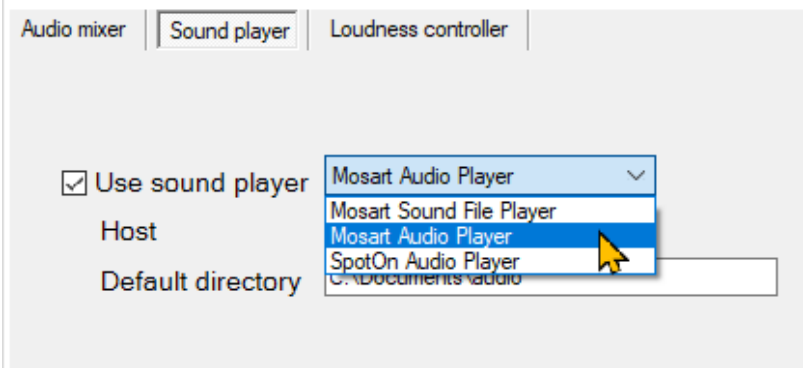
The standard Viz Mosart audio player supports formats: mp3, wav, wma, aac, m4a, mp4, aiff, avi.

To set up the default Viz Mosart audio player

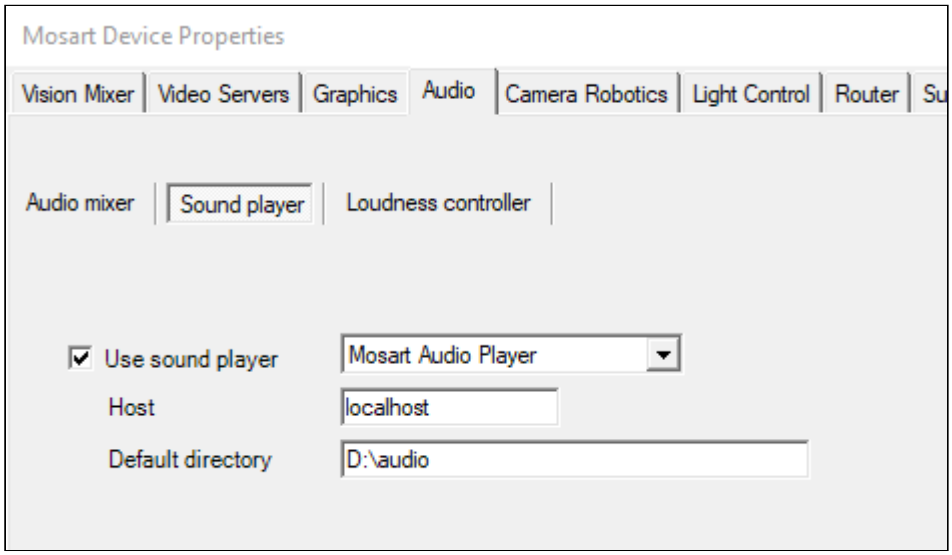
1. On the machine that will be used for the audio player, open the Viz Mosart installation folder and locate the Audio Player installer file, named something like VizMosartAudioPlayer-5.0.2.32554.msi
2. Install Audio Player, then start it.
3. On the Viz Mosart server, open AV Automation.
4. Go to **Devices** > **Preferences** > **Audio** > **Sound player.**

Copyright © 2024 Vizrt

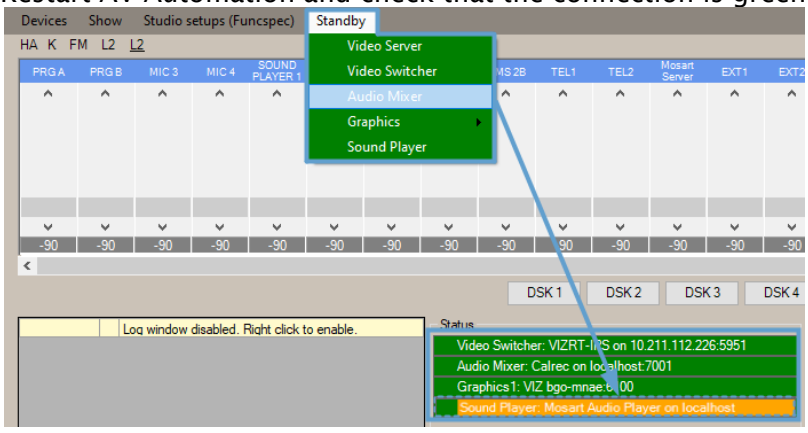
5. Check **Use sound player** and choose from the players in the drop-down list.



6. Under **Host** write the IP address to the computer where audio player is running.
7. Under **Default directory**, write the folder path where the audio files are stored.



8. Restart AV Automation and check that the connection is green.



9.5.2 Third Party Audio Player Devices

You can connect to and enable a compatible third party audio player.

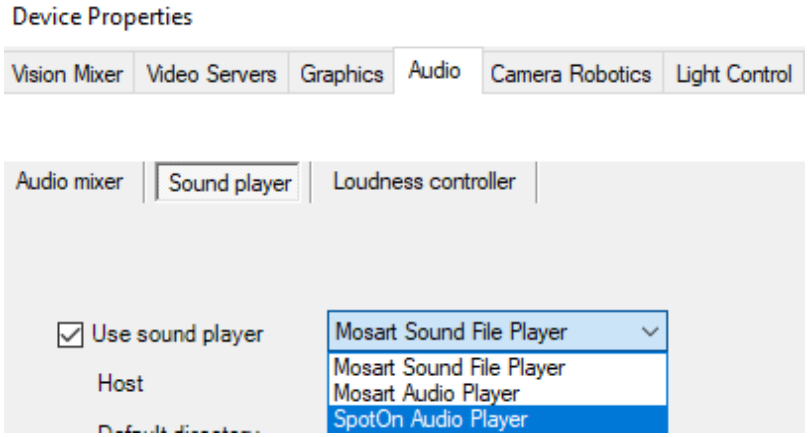
- Setting up a SpotOn audio player
- Setting up a TriCaster audio player

Setting up a SpotOn Audio Player

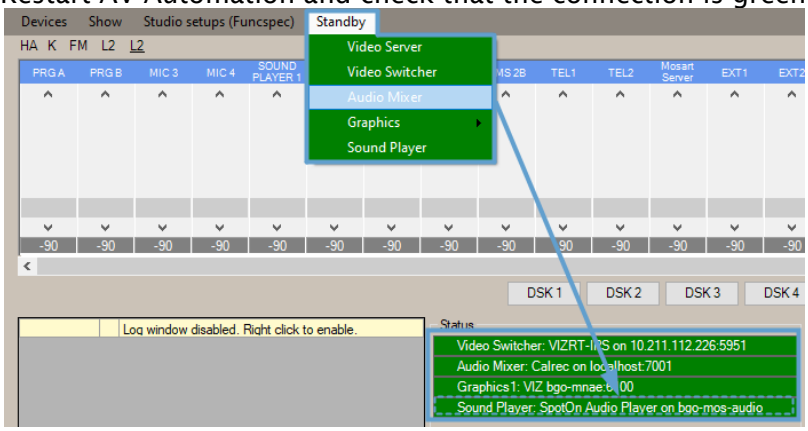
To set up a SpotOn audio player

This procedure prepares a *SpotOn* audio player.

1. Download, install and start the device software to a host machine available to the Viz Mosart server.
2. In AV Automation, go to **Devices** > **Preferences** > **Audio** > **Sound player.**
   a. Check **Use sound player**.
   b. Choose *SpotOn Audio Player* from the players in the drop-down list.



   c. Under **Host** write the IP address to the machine where the SpotOn player is running.
   d. Click **Save & Close**.
3. Restart AV Automation and check that the connection is green.



Setting up a TriCaster Audio Player

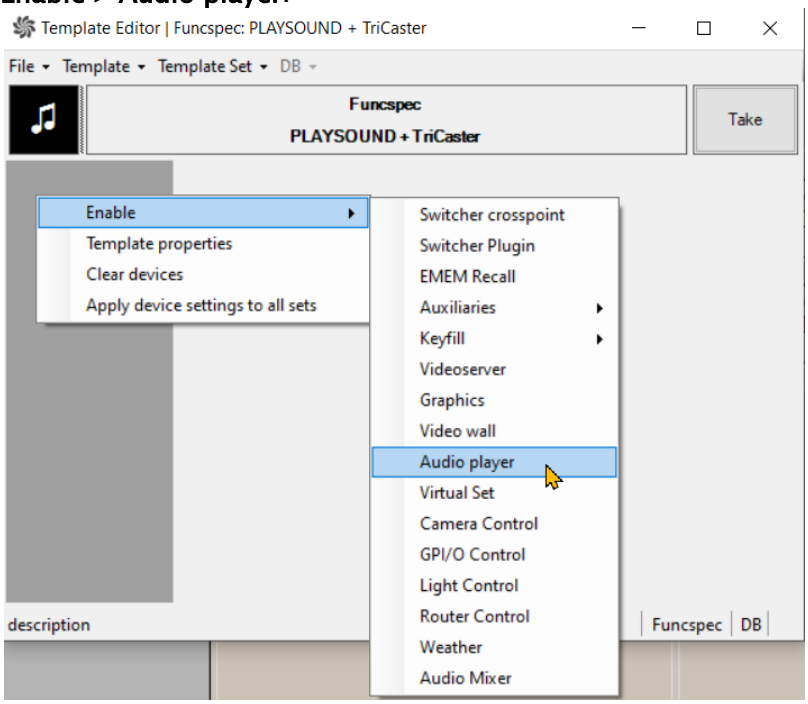
By adding a driver, you can control (access and play audio clips) the TriCaster onboard sound player from a Viz Mosart template.

To set up a TriCaster audio player

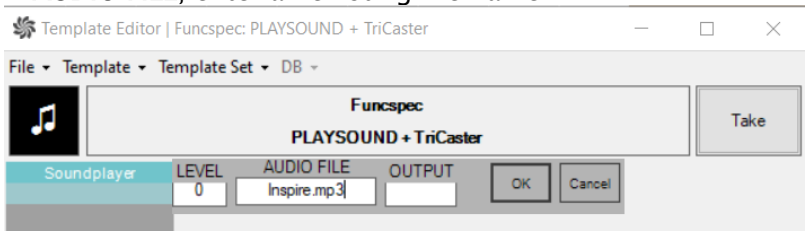
1. Start the TriCaster on a host machine available to the Viz Mosart server.
2. In AV Automation, navigate to **Devices > Device Properties > Audio** (tab) **> Sound player** (tab).
   a. From the drop-down menu to the right of **Use sound player**, select *vizrt-ips*.
   b. In the **Host** field, enter the hostname or IP address of the TriCaster host.
   c. In **Default directory**, enter the sound clips directory on the TriCaster machine (for example, C:\ProgramData\vizrt\Media\Sound and Music\Sample\www.SoundTraxxMusic.com).

      > **Note**: The current implementation enables playing of files from only a *single* folder.

   d. Click **Save & Close**.
3. Restart AV Automation and make sure you have a green connection (as in the SpotOn audio player example above).
4. In AV Automation, navigate to **Template Editor > Template > New**.
5. Choose an appropriate template, right-click in the dark grey area to the left and select **Enable > Audio player**.

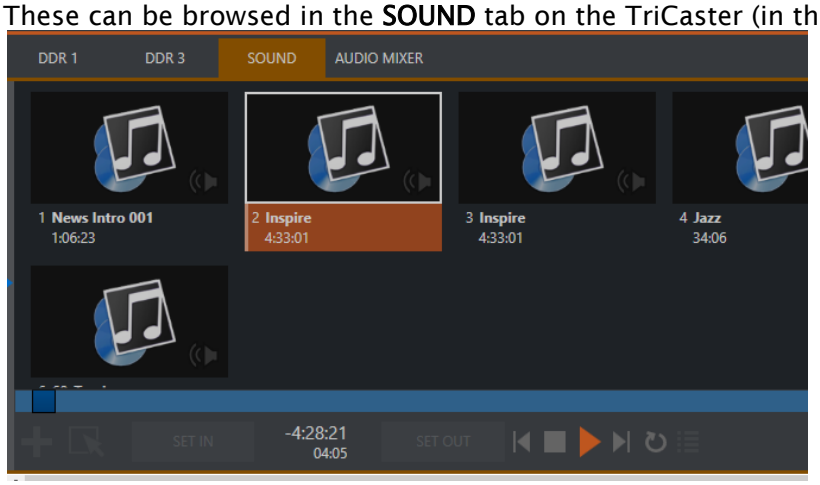


6. In **AUDIO FILE**, enter an existing file name.



Copyright © 2024 Vizrt

These can be browsed in the **SOUND** tab on the TriCaster (in this example, *Inspire.mp3*).



> **Note**: The **OUTPUT** field is ignored, as the TriCaster has a single output.

7. On the template, click **OK**.

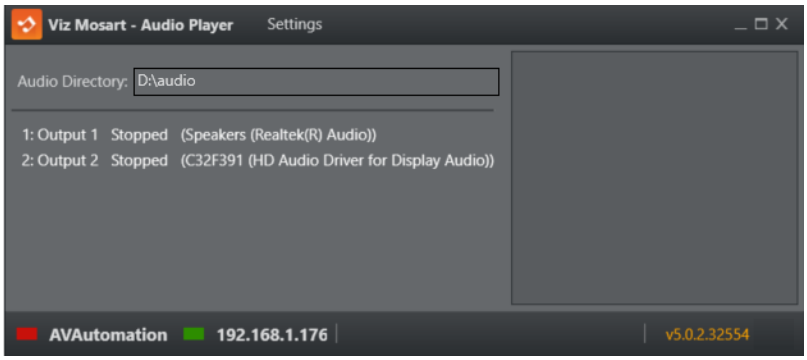
9.5.3 Working with Audio Players

Working with the Viz Mosart Audio Player

When the default Viz Mosart Audio Player is configured, the user opens the UI from a desktop shortcut
or file location C:\Program Files (x86)\Mosart Medialab\Mosart Audio Player\Mosart Audio Player.exe.

> **Note**: User actions are described in section *Playing Audio* of the *Viz Mosart User Guide*.

Viz Mosart Audio Player UI

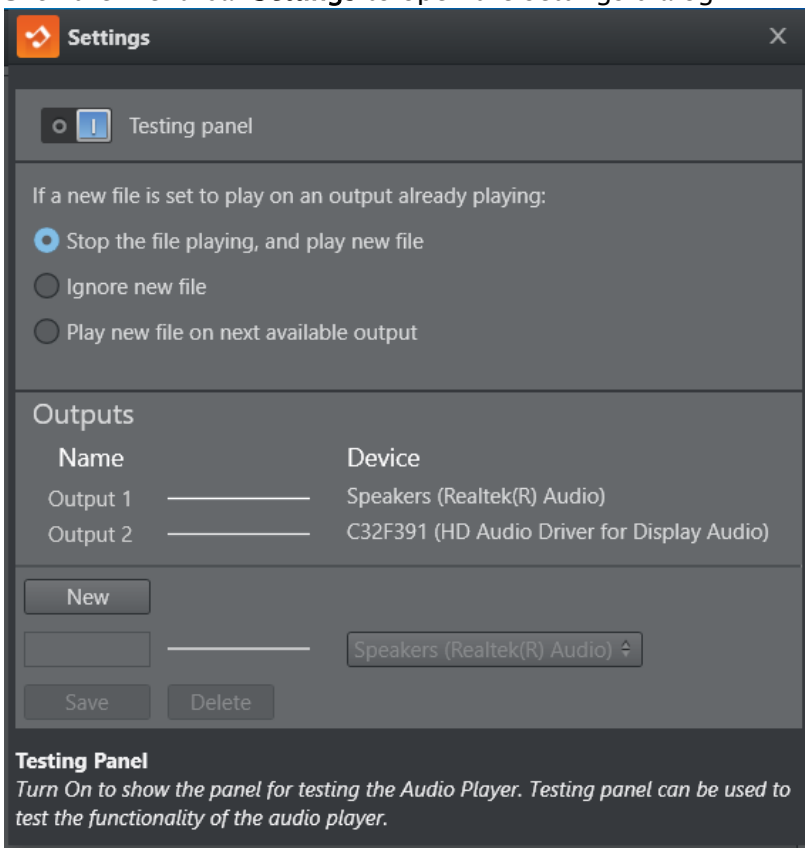


UI Element	Description
Settings button	Open the **Settings** window for configuring the Audio Player. See Audio Players below
Audio Directory	Folder path to where your audio files are. All files in this folder are ready to be played (This location is set in AV Automation **Devices > Properties > Audio > Sound Player**)
Numbered Output list (left hand panel)	A list of all available devices. The first number is the number you use in AV Automation or the NRCS to specify the audio file playout.
Log panel (right hand panel)	All log messages that the audio player creates.
AVAutomation status	Green: AV Automation is connected Red: No AV Automation connection.
IP Address	IP address of the computer Audio Player is running on.
Version	Version number of the Audio Player.

Copyright © 2024 Vizrt

Audio Player Settings

- Click the menu bar **Settings** to open the settings dialog.



Testing panel

Opens a panel for testing the functionality of the selected Audio Player.

File already playing behavior

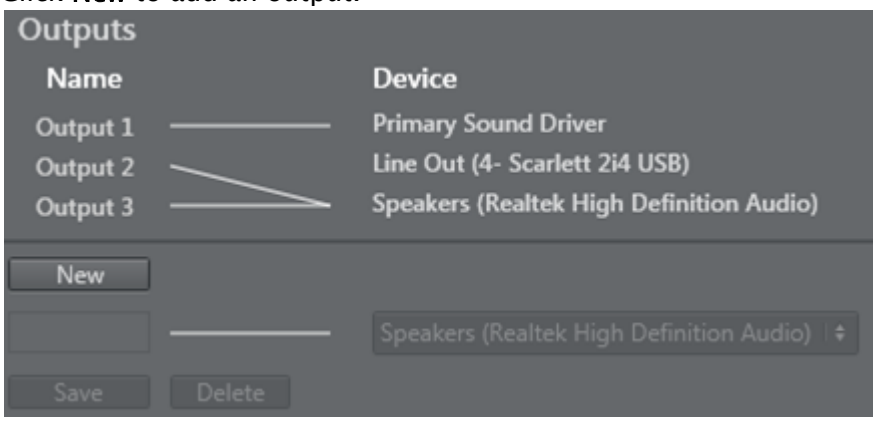
Controls what the Audio Player does if a new file is played when a file is already playing.

Outputs

The list on the left shows all the outputs, and the list on the right shows available playback devices on the computer. The line between them shows playback device mapping.

- Click an output to map to playback device.

- Click **New** to add an output.



- You can *rename* or *delete* an output.

Audio Player Configuration File

The Audio Player Configuration file `Mosart Audio Player.exe.config` is at `%ProgramFiles(x86)%\Mosart Medialab\Mosart Audio Player\`.

- Change the *baseAddress* attribute to use another port:

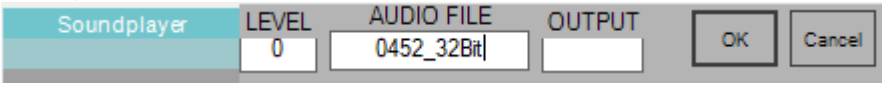
```
<add baseAddress="http://localhost:8084/LiveSoundService.svc/"/>
```

Working with a Third Party Audio Player

This example features the SpotOn audio player.

SpotOn Audio Player

Addressing an audio file that is stored in a SpotOn client is similar to the default player method above, but the file identifier is different.

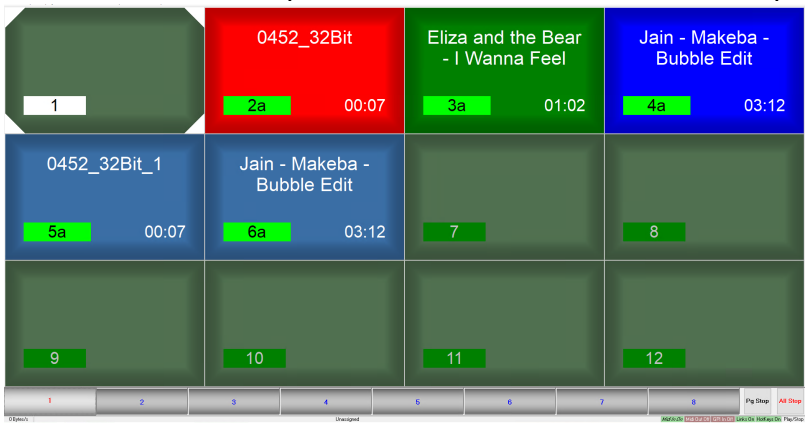


- **LEVEL:** This value *does not* affect the audio fader that is taken to Program. Instead a special command option is offered. Set LEVEL to *-90* to send a **STOP** (playing audio) command. All other values are ignored.
- **OUTPUT:** Enter the panel *button number* to identify the associated sound file (in this example "2a").

> **Tip**:

Use the *button number* (mapped to OUTPUT) as reference (instead of the audio file name). You can then replace the audio file assigned to a button (for example, when a show is rebranded and new music selected) and still keep the same template information.
This avoids having to change the file name.

- **AUDIO FILE:** Enter the panel button name here (in this example 0452_32Bit).



Note:

- Newsroom tag available. If a newsroom tag is added, and a value is provided by the NRCS, that value for the file name (with either AUDIO FIE or OUTPUT) is used.
- When both panel button *number* and panel button *name*. are used as file identifier, the value of panel button *number* takes precedence.

See also

- Audio in AV Automation Devices
- *Playing Audio* in the Viz Mosart User Guide (user actions).

Copyright © 2024 Vizrt

9.6 Viz Arc AR Support In A Mosart Show

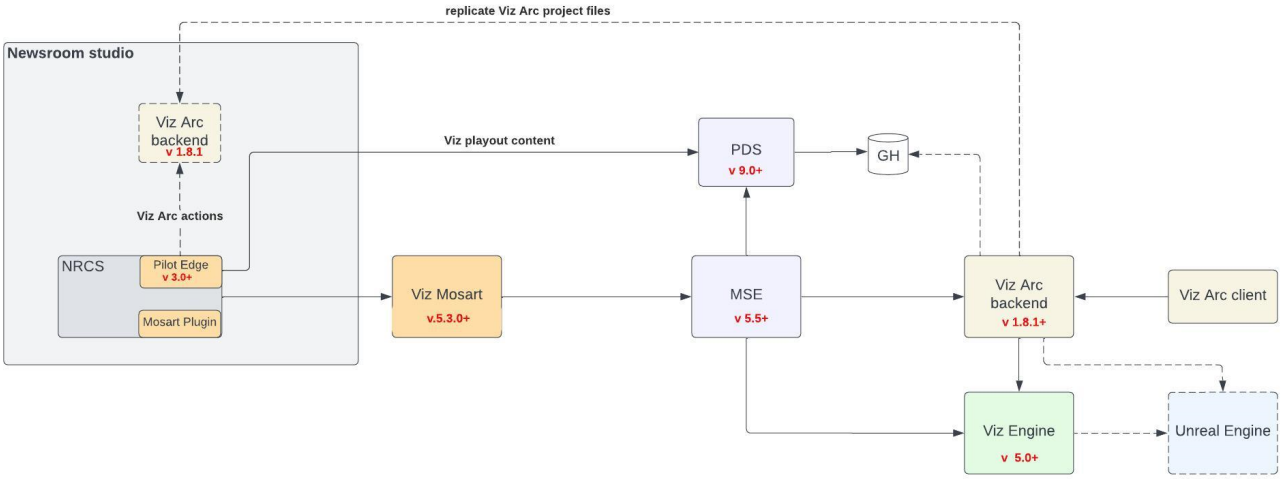
Viz Arc is an augmented reality control system, which enables users to drive virtual sets, AR graphics, Viz Engine and Unreal® Engine graphics from a single interface. These elements can also be driven from a Viz Mosart controlled show.

- Requirements
- MOS integration
- Driver Configuration
- Viz Mosart Workflow

9.6.1 Configuration and Setup

Requirements

The illustration below highlights the main actors and minimum component versions required for configuring Viz Mosart to manage Viz Arc actions within an NRCS-created rundown.



MOS integration

To setup Viz Pilot Edge with Viz Arc

- Refer to the Viz Pilot Edge documentation.

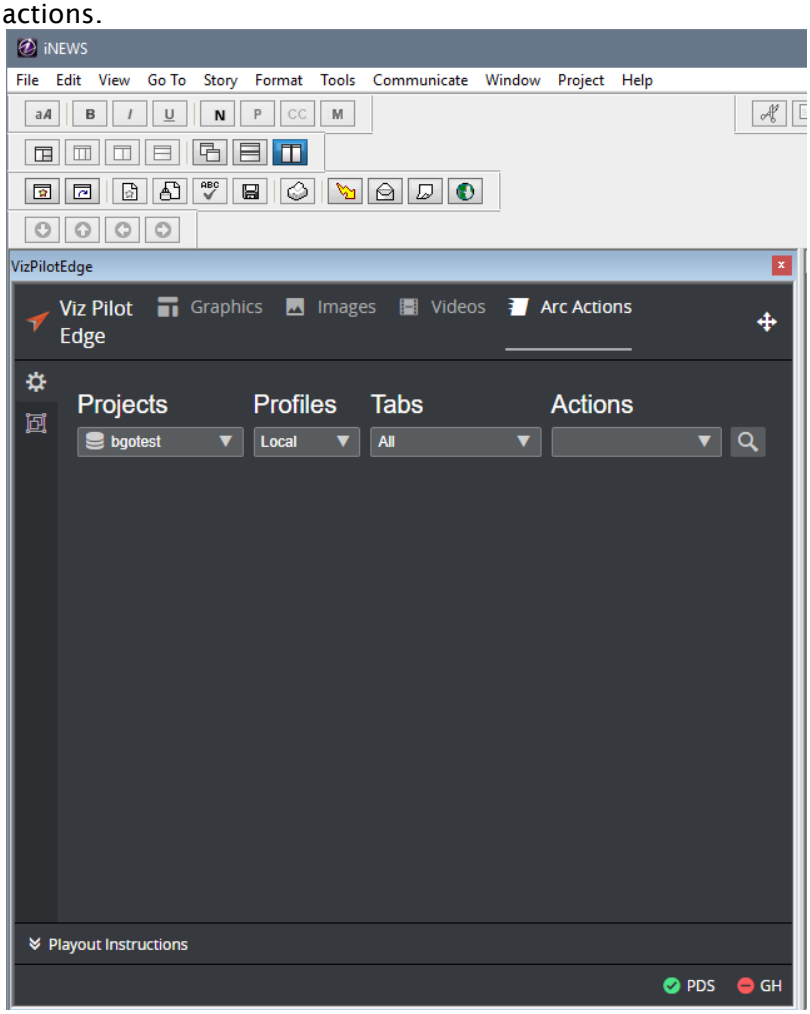
> ⚠ **Note**: In the diagram above
> - A *second* Viz Arc backend is set up on the Newsroom System machine for security reasons.

Copyright © 2024 Vizrt

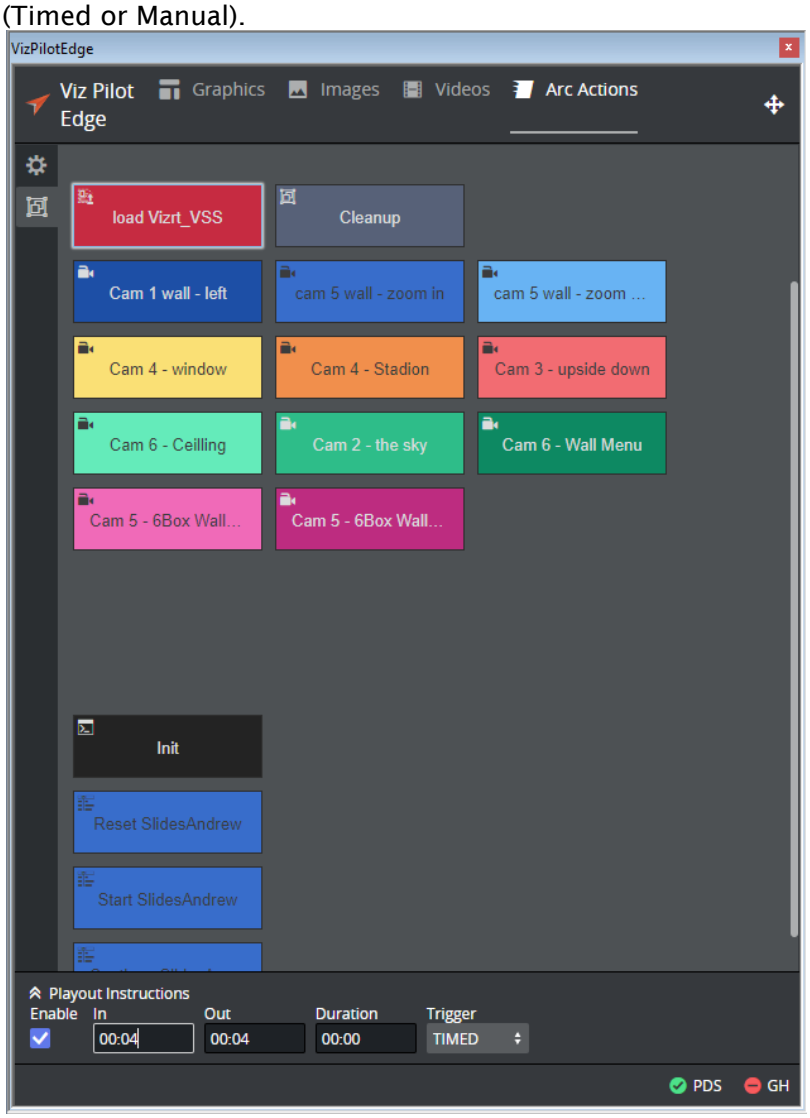
- Viz Arc does not currently support authentication. If this is not an issue, Viz Pilot Edge can be set up to directly fetch data from the Viz Arc playout backend.

To create a Mosart show containing pre-planned or manual Viz Arc actions

1. Open Viz Pilot Edge from the NRCS.
2. From the respective drop-down menus, select **Project**, **Profile** and **Tab** to browse the Viz Arc actions.



3. Select a Viz Arc action and enable the *Playout Instructions* to add timing and the trigger (Timed or Manual).

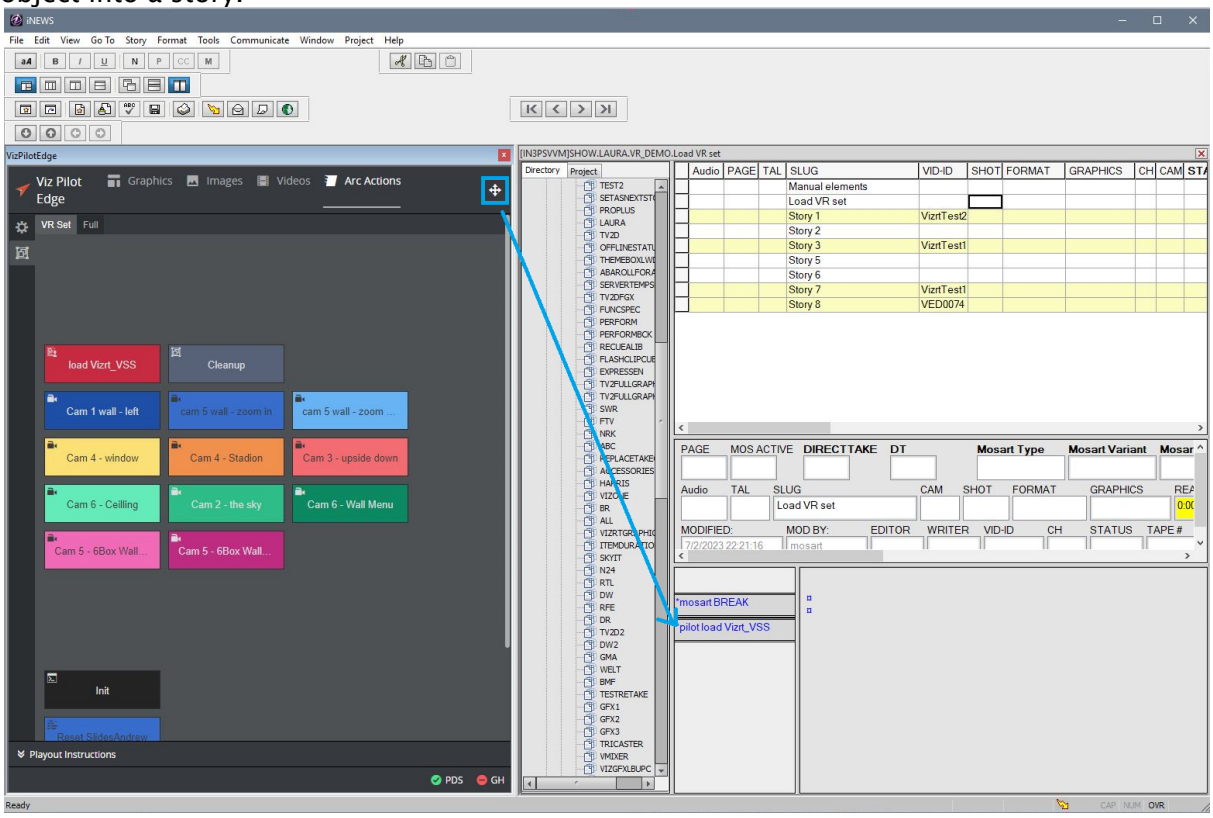


4. When the trigger is set to *TIMED*, the **In** field can be set to give the offset in minutes and seconds relative to the story item start.
    a. In this case, Viz Mosart will play the action when the story item is taken or with the given offset.
    b. If the Trigger is set to MANUAL, Viz Mosart will not automatically play this item, but mark it as a Manual item in the UI.

> ⚠ **Note: Out** and **Duration** are not yet supported for Viz Arc actions as these actions do not support such events. Only **In** time is relevant.

Copyright © 2024 Vizrt

5. Use the **plus** icon in the top-right corner in Pilot Edge to drag and drop the generated MOS object into a story.



In this example, a Viz Arc action was added on a *Break* story to load a virtual set scene.

Driver Configuration

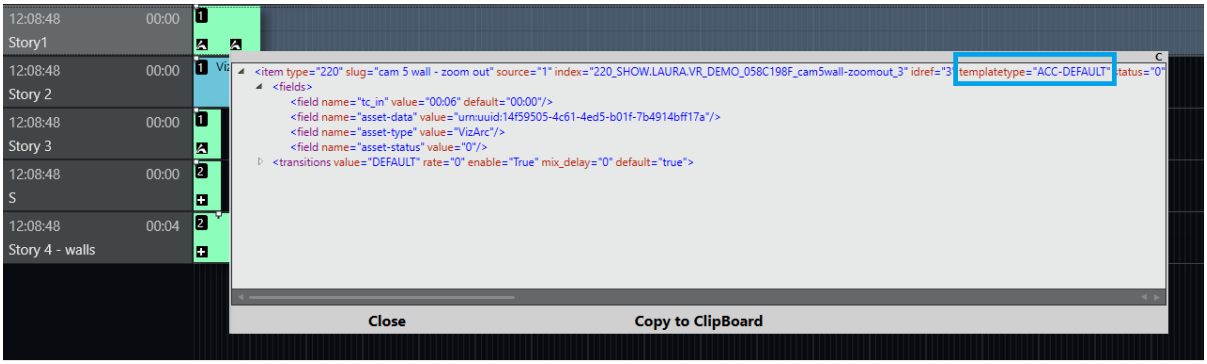
The *Vizrt Graphics* driver to communicate with Vizrt Media Sequencer through a REST API exposed by Media Sequencer.

- Refer to the Vizrt Graphics section of this guide.

Viz Mosart Workflow

When a rundown containing Viz Arc MOS objects is loaded in Viz Mosart, Mosart parses and attaches the asset data to a default Accessory template:

Copyright © 2024 Vizrt



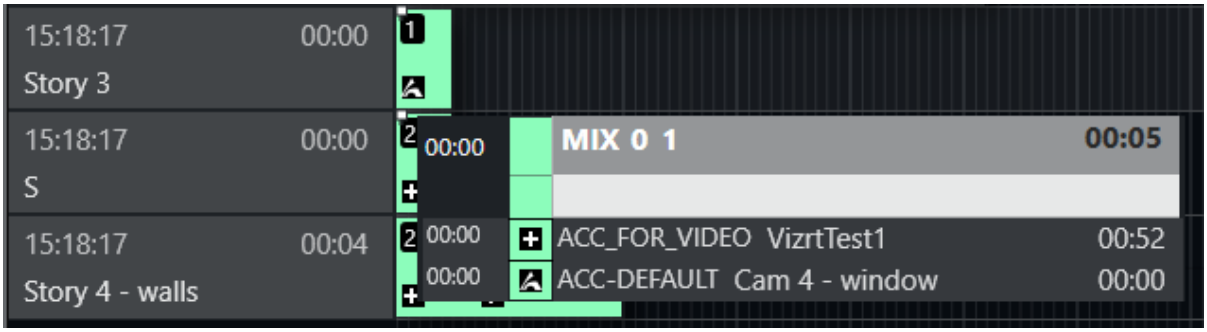
Because the Accessory item has a Viz Arc asset type attached, the visualization of the Accessory item is different to the standard *plus on a black background*.

Please refer to the *Viz Mosart User Guide* (section **Rundown Window > Story Elements > Secondary Story Elements**), for more on the visualization of Accessory items.

If the template does not exist in the current studio setup (template set), it is automatically generated by Mosart.

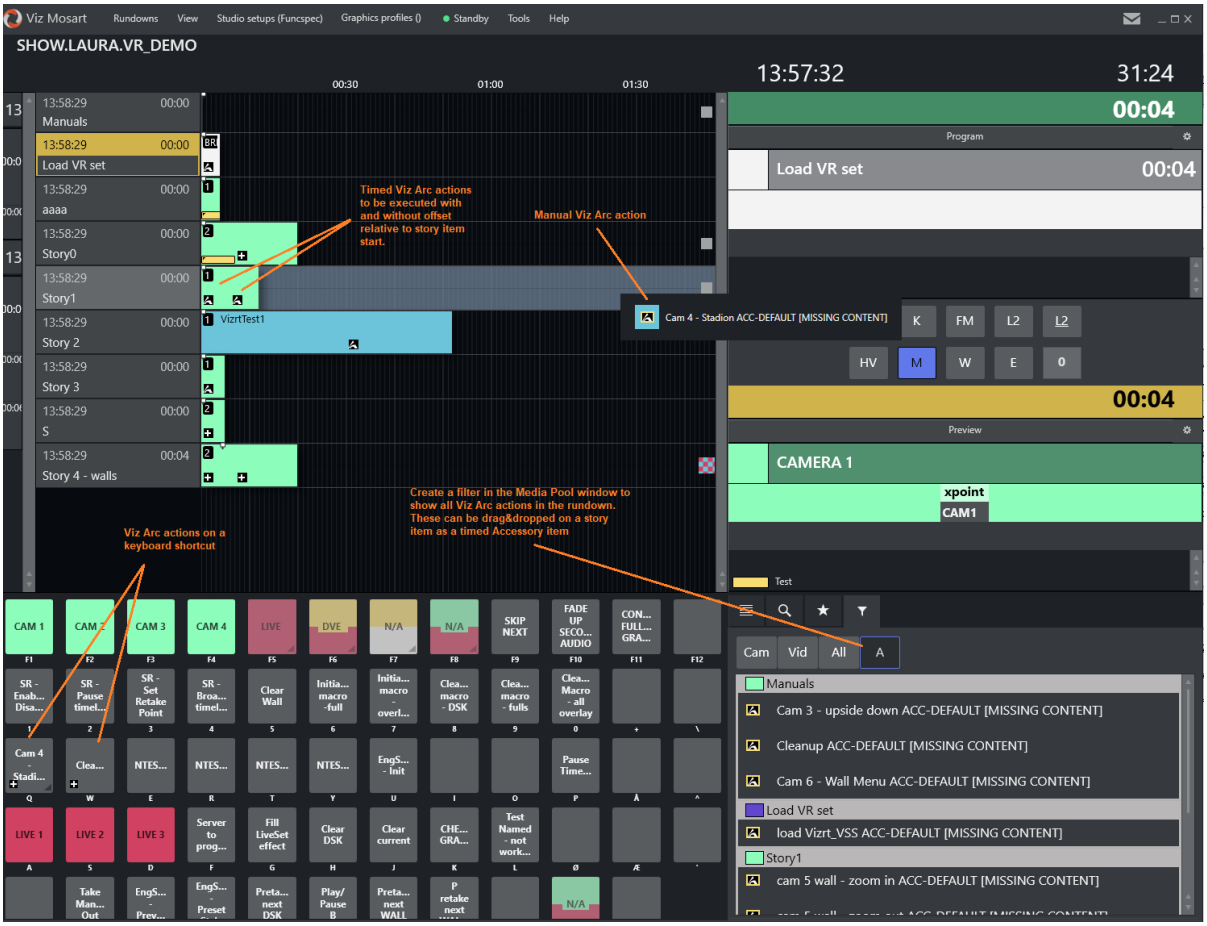
> **Note**: Attaching Viz Arc MOS object to a custom template is *not currently* supported. If modifications are made to the default Accessory template using the Template Editor and a function is enabled within the template, that logic will be applied to *all* items utilizing that template.

To execute other secondary actions in the same time with the Viz Arc action, a typical Accessory item can be used to execute in the same time with the Viz Arc action:

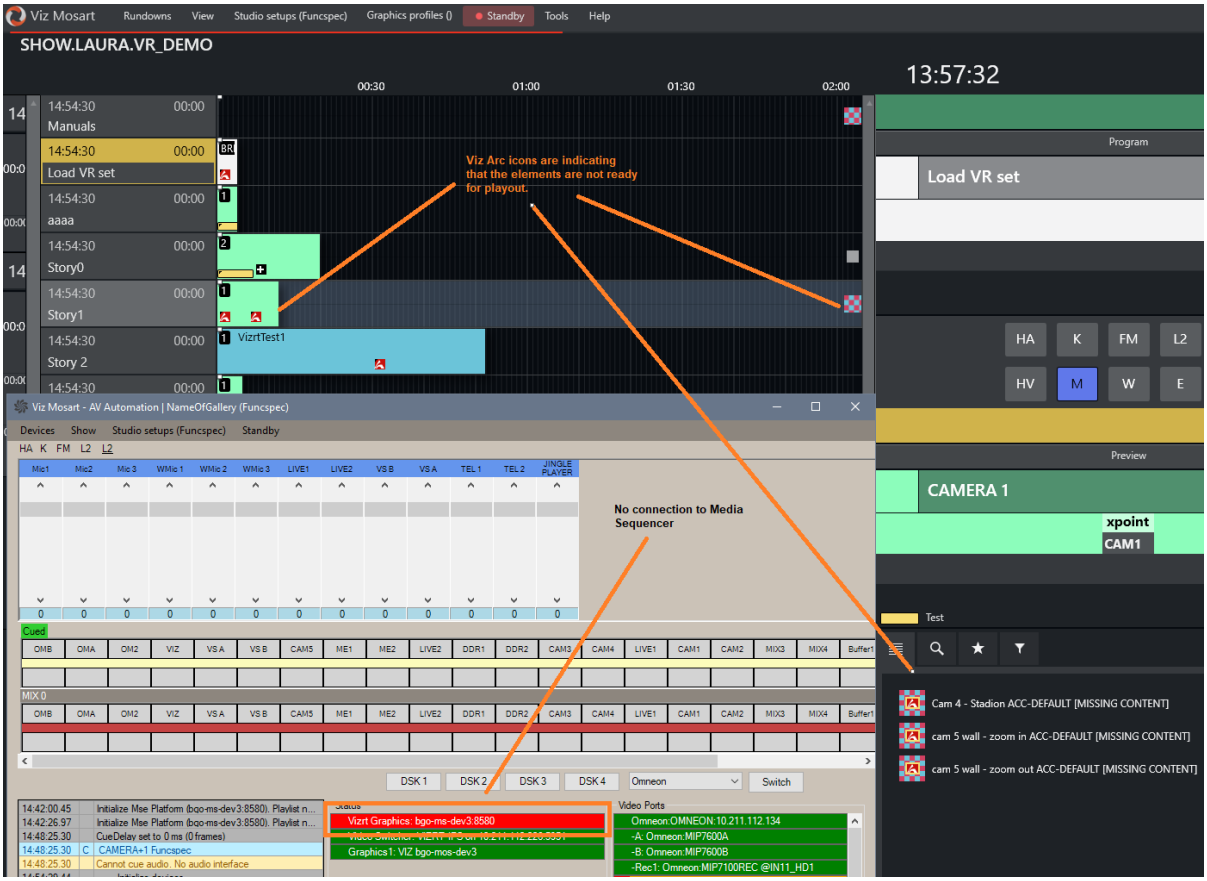


The timing and behavior (whether timed or manual) of the Accessory item are inherited from the MOS object received from the Newsroom System. Additionally, all known functionality applicable to Accessory items can also be applied to Viz Arc elements.

Copyright © 2024 Vizrt



If the Viz Arc elements are not ready for playout because, for example, there is no connection to the Media Sequencer, this is indicated in Mosart UI with a *red* background:



You can create Viz Arc actions to

- *Initialize* the Media Sequencer (MSE) playlist when a show is loaded in Viz Mosart
- Clean up the MSE memory after each show.

These are added as manual items in the Viz Mosart rundown and activated by the operator from keyboard shortcuts.

Current Limitations

- No loading status support: A virtual set loading on the MSE may take a considerable amount of time. Currently, Viz Mosart does not show the progress of the loading status or when it is ready for playout.
- No redundancy support for the new driver: A Media Sequencer backup server can not currently be configured in case of failover.
- No Viz Arc actions supported as *control commands* on a template.

Copyright © 2024 Vizrt

9.7 Frame Accurate System Operations

To ensure reliable operational synchronization between Viz Mosart and the studio equipment that it controls, the system can be calibrated to perform on a frame accurate level of timing precision. This section introduces:

- Why Frame Accuracy Matters
- Requirements
- Enabling Frame Accuracy Operation
- Calibrating the System
- Genlock Logging
- Frame Accuracy in Story Recorder
- Limitations

9.7.1 Why Frame Accuracy Matters

There are two areas where frame accuracy in a Viz Mosart controlled system are significant:

- Frame Accurate System Operations
- Frame Accurate System Operations.

Predictable Behavior of Equipment

Unless it is known where in the frame the Viz Mosart command is issued, the response from the equipment will always be arbitrary.

In the figure below, the red bars represent the start of a frame. The outgoing Viz Mosart commands (*cmd 1*, *cmd 2*) are now aligned with the start of the frame grid (frame accurate).

Tip: The illustrations in this section are *animated*. If you are viewing a PDF, we recommend the online version.

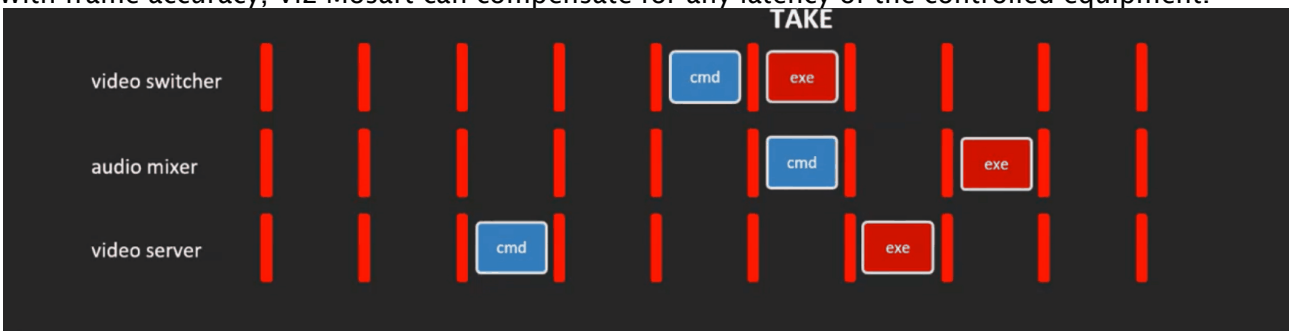
This provides deterministic behavior from the controlled equipment, as the equipment is always given exactly the same amount of time to respond (in the above figure, after *one* frame). Without frame accuracy, *cmd 1* might be fired already somewhere in the first or second frame above, and the gear will not operate at the anticipated frame, giving unpredictable results.

Copyright © 2024 Vizrt

Latency Compensation

In non-frame accurate systems, Viz Mosart only compensates for delays in the behavior of the playout *video server*.

With frame accuracy, Viz Mosart can compensate for any latency of the controlled equipment.



As a pay-off of Frame Accurate System Operations, a calibration process can reveal the latency of each piece of gallery equipment.

Viz Mosart can then compensate for these delays, by always executing commands earlier.

In the figure above, the three devices (*video switcher*, *audio mixer* and *video server*) have millisecond latencies of 1, 2 and 3 frames respectively.

After compensating by executing each command specifically earlier, Viz Mosart can now successfully perform the TAKE command, synchronized for all three devices, from the same frame.

9.7.2 Requirements

- On the Mosart server machines (main and backup), a Plura genlock/timecode card must be installed.
  This implies that the machine must have a free slot for this type of card.
  Vizrt has only verified Plura PCI 3G, with digital genlock (SDI) and supports, for time being, only the traditional digital VITC or ATC-VITC as timecode formats.
- Calibration has been performed.
- A dedicated template set.
  There will be timing differences for template execution under frame accurate operation. Latencies are automatically compensated for primary events.

> ⚠ **Note**: Whilst frame accurate operations are being continually developed and improved, not every feature from standard mode may be available or possible in frame accurate mode. This is also why a dedicated template set is essential.

9.7.3 Enabling Frame Accuracy Operation

For frame accurate behavior, both Viz Mosart and the equipment it controls, require setup of both a genlock and timecode feed.

Copyright © 2024 Vizrt

- **Genlock Settings**
  The switcher, video server (playout and recording) and Viz Mosart Server, all require genlock.
- **Timecode Feed**
  Both the recorder and Viz Mosart Server require a timecode feed.

> ⚠ **Note**: Story Recorder can be used without having proper genlock setup and a timecode reader card installed on the Mosart Server machine, *but only for testing*. In this case, the time of the day will be used.

Configuring Viz Mosart for Frame Accurate Operation

Once the system, including all equipment has been calibrated, frame accurate operation can be easily switched on or off.

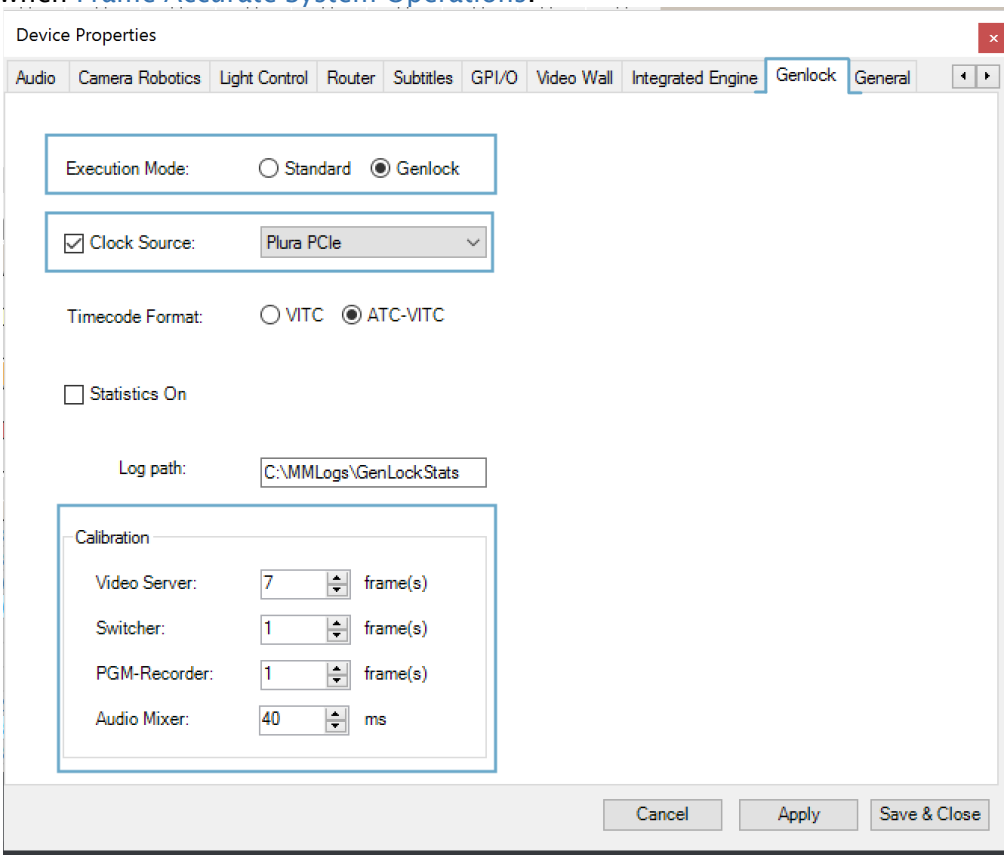
To activate frame accurate operations

Frame accuracy related settings and options are made in Av Automation.

1. Navigate to **Devices > Properties > Genlock** tab.
2. Select
   a. **Execution Mode**: Select *Genlock* mode.
   b. **Clock Source**: When checked, select which genlock / timecode card is installed, if any. By default, this is Plura timecode reader card (and for time being, the only option).

Copyright © 2024 Vizrt

c. **Calibration**: These are the individual latencies of the gear being controlled, as derived when Frame Accurate System Operations.



3. Click **Apply**, then **Save & Close**.

Combinations of the configuration above for **Execution Mode** and **Clock Source** settings:

Execution mode	Clock Source	Logic
Standard	Not selected	Classic mode. No timecode in the Viz Mosart logs. No statistics can be performed.
Standard	Selected	Classic mode. Timecode from the selected clock source (by default, this is Plura PCIe and the only supported option) is written in the Viz Mosart logs generated from AV Automation.
Genlock	Selected	Frame accurate behavior when executing Viz Mosart templates during a show. A clock source is required. Timecode from the selected clock source (Plura) is written in the Viz Mosart logs, generated from AV Automation. Viz Mosart falls back to the *internal* clock source if the selected clock source is detected as inaccurate.

Copyright © 2024 Vizrt

Genlock	Not selected	Not allowed. Viz Mosart will fall back to Standard execution mode, with no timecode.

To deactivate frame accurate operations

Open AV Automation.

1. Navigate to **Devices > Properties > Genlock tab**.
2. **Execution Mode**: Select *Standard* mode.
3. **Clock Source**: Deselect.
4. Click **Apply**, then **Save & Close**.

Genlock and Timecode Reliability

Viz Mosart contains a high resolution internal genlock clock as a built-in safety mechanism which takes care of intermittent or lost external genlock and/or timecode signal.

- Whenever the external *genlock signal* drops out or becomes unreliable, Mosart will fallback to the internal genlock clock. Mosart will return to external once the signal is reliable again.
- Mosart effectively runs on the *internal timecode clock*, constantly synchronized to the external clock when available, keeping Viz Mosart in phase.

9.7.4 Calibrating the System

- For Viz Mosart to function in frame accurate mode, a calibration process has been created to better understand the timings associated with the gear being operated.
  Contact Vizrt Support for further details.
- The guide defines *six tests that must be executed* to detect the *latencies* (delays) introduced when operating gallery gear, so that Viz Mosart can Frame Accurate System Operationsfor these delays and verify equipment Frame Accurate System Operations.

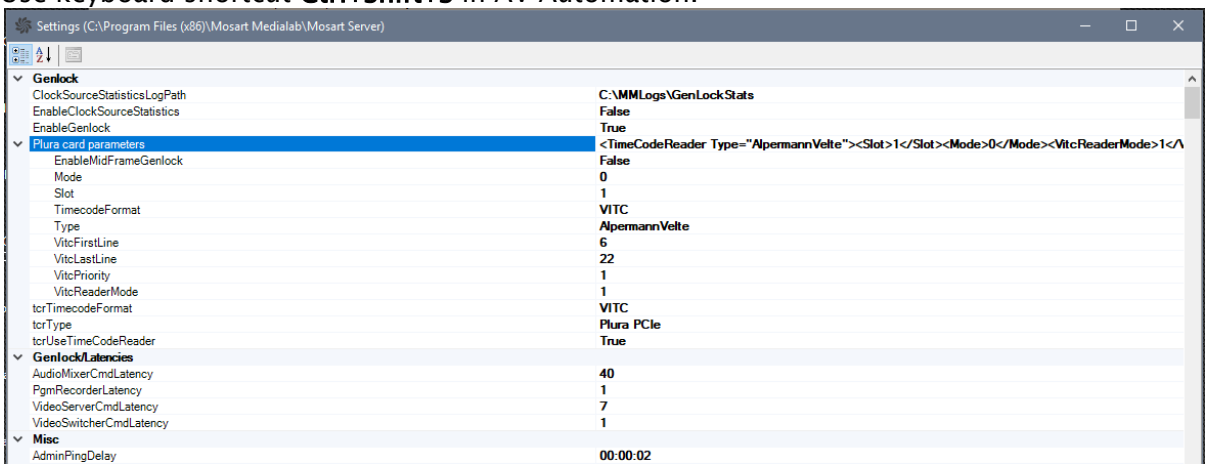
Timecode Readers

In Av Automation, you can change properties related to the Plura timecode reader card.

Copyright © 2024 Vizrt

To configure the Plura timecode reader card

1. Open the Av Automation Settings (*user.config* file).
   Use keyboard shortcut **Ctrl+Shift+S** in AV Automation.



2. If any changes have been done on the **Genlock** tab in AV Automation properties (see Frame Accurate System Operations above),
   click the **Apply** button or close the Properties window and enter **Ctrl+Shift+Q** for the changes to take effect.

Gathering Clock Source Statistics for Vizrt Support

Once Viz Mosart is set up for frame accuracy, it is recommended to perform clock source statistics that will verify the reliability of the genlock and timecode signals.

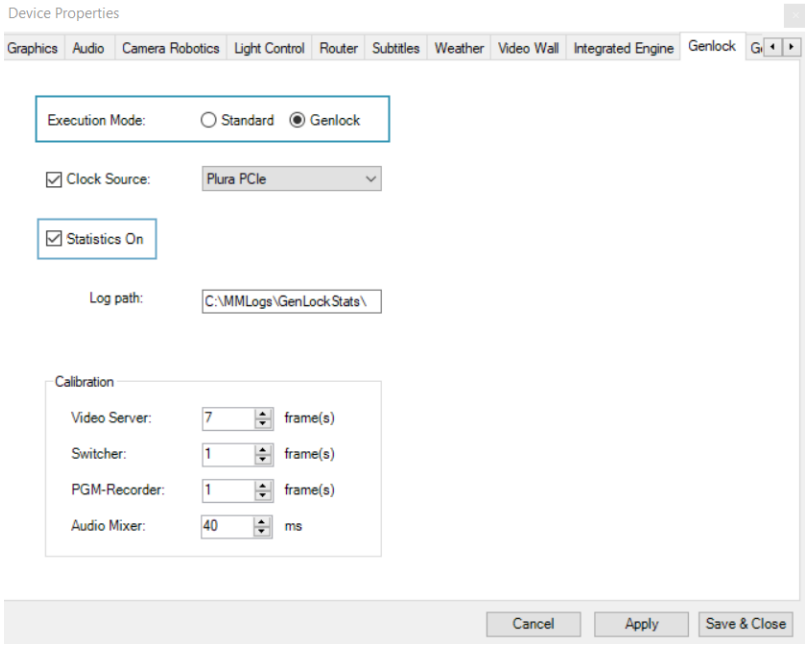
Follow the procedure below, and forward the results to Vizrt customer support, who will perform an analysis and report results.

To start clock source statistic gathering

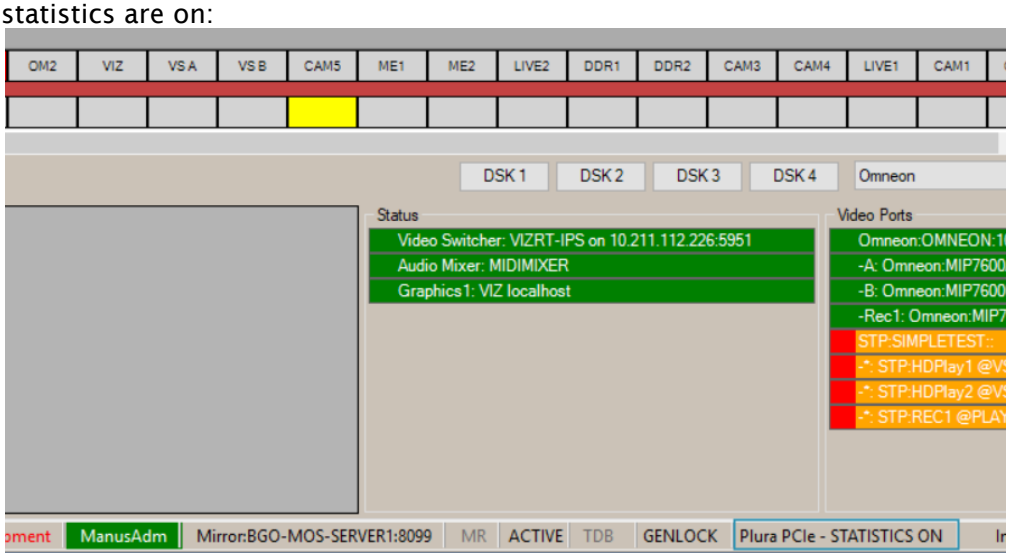
1. You start/stop statistics in AV Automation.
   a. Navigate to **Devices > Properties > Genlock tab**.
   b. **Execution Mode**: *Genlock*
   c. Clock source checked and a clock source selected.

Copyright © 2024 Vizrt

d. **Statistics On:** *Selected*



2. Specify the location where you wish to save the stats in field **Log path**.
3. Click **Apply**, then **Save & Close**.
4. the configured external clock source, if available. The file generated
5. Check the indicator in the AV Automation status bar at the bottom of the screen, that statistics are on:



6. Measurements will be performed for about 8-10 minutes
   You can stop the statistic measurements earlier by unselecting Statistics On, as described in step (1) above.
7. Collect the saved statistics from the location defined in **Log path** (above) for:
   a. The configured external clock source:
      ExternalClockSourceStats_<uid>.csv.
   b. The Mosart frame clock, driven by the selected clock source (internal or external).
      Logs are stored in *FrameClockStats_<uid>.csv*.

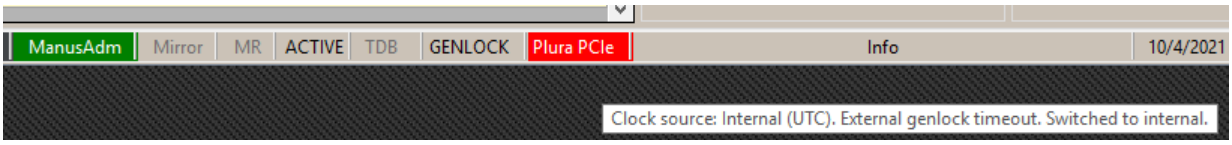
Copyright © 2024 Vizrt

Tip: You can start/stop the clock source statistics by toggling the keyboard shortcut **Ctrl+Shift+I** directly in AV Automation, without the need to open the **Genlock** settings tab in the **Properties** menu.

To analyze clock source statistics

This is a complex operation performed by Vizrt Support.

- The frame clock (FrameClock) is used internally by Viz Mosart to execute templates frame accurately.
- If the external clock source is not reliable (shown in red on the AV Automation status bar), the frame clock is driven by the internal clock source:



- If the external clock source is reliable, the internal FrameClock should behave more or less as the external clock source.
  This can be seen in MS Excel by comparing the chart generated (in Excel) from GenlockPeriodMs or JitterMs columns in FrameClockStats_<uid>.csv file,
  with the chart generated from the same columns in ExternalClockSourceStats_<uid>.csv.

9.7.5 Genlock Logging

A new type of logging has been introduced with frame accuracy in Viz Mosart to help identifying problems related to genlock, if this is enabled.

- To enable the genlock logging, copy the configuration file **GenLogRepositoryConfig.xml** from the *Viz Mosart installation folder/Mosart Server/ConfigurationFiles* to your working folder (*Channeltemplates* or *ProgramData*).
- Open **GenLogRepositoryConfig.xml** in an editor and configure it in a similar way as the other type of logging (refer to System Logging). These configurations also include settings about when the logging files shall be purged.
- The genlock logging is by default generated in *C:\MMLogs\Genlock* every time a new show is run in Mosart (whether Story Recorder mode is enabled or not).
- A new genlock logging file is generated when a rundown is (re)loaded in Viz Mosart or when the show is paused whilst Story Recorder is enabled.

Tip: Contact Vizrt Mosart Support for assistance with
- Statistics analysis
- Calibration

Copyright © 2024 Vizrt

9.7.6 Frame Accuracy in Story Recorder

For Story Recorder to provide high quality final shows, frame accuracy is a requirement. SR can also be run with Viz Mosart running in Standard mode, *but only for testing*.

See also the section **Story Recorder** of the Viz Mosart User Guide.

9.7.7 Limitations

Frame accuracy is only available on events driven from primary templates (clips, switcher, audio and full screen graphics), not secondary events (like overlay graphics or accessories).

Copyright © 2024 Vizrt

10 Viz Mosart Template Database

With the Viz Mosart Template Database (TDB), template information can be shared among the following clients:

- Between galleries automated with Viz Mosart (where templates are shared across galleries and Mosart servers).
- Between a Mosart main and backup server, in a redundancy setup.
- To NRCS clients using the Mosart Web Apps NRCS Plugin.
- To NRCS clients using Mosart ActiveX or a custom ActiveX.

The following Template Database topics are covered:

- Architecture
- Template Properties
- Information Stored in the Template Database
- Prerequisites for Connecting to TDB

10.1 Architecture

- The template database is stored in a MySQL relational database (or a valid MySQL equivalent such as MariaDb).
- The database may be installed separately for the Mosart servers or as a part (schema) of an in-house or cloud database, or even be virtualized.
- The only main requirement that the database must be MySQL compliant.

10.2 Template Properties

A Mosart template as stored in the template database has the following properties:

Property	Description
Gallery	The Mosart gallery naming where this template is implemented, and 'owner' of the templates. **Note:** A special *global* gallery is used for TDB templates that are shared, replicated and available between *all* galleries. This dedicated, virtual gallery for a global template set, is by default named *SHARED*.

Copyright © 2024 Vizrt

Property	Description
Template set	The Viz Mosart template set to which a template belongs. Can be *shared* or *local*. ⚠ **Note:** A local template is **gallery-specific** meaning it is 'owned' by the corresponding gallery and is independent and not visible to any all other galleries. ⚠ **Note:** The special Mosart template set **Directtakes** must be defined as a *shared* template set.
Template type	The Mosart template type. For example, CAMERA, CLIP, DVE.
Template variant	The Viz Mosart template variant. For example a specific camera is identified with a template named *CAMERA+3*.
Template description	Common attributes for the template which is shared among all galleries. The template description also contains a list of input fields normally known as *newsroomtags*.
Template implementation	Gallery-specific information used to control systems using Viz Mosart for automation. This is the main part of the template and has to be implemented in every gallery.

10.3 Information Stored In The Template Database

Details from the template database are stored in two Mosart server applications:

- **AV Automation**
  Stores templates and the file AvConfig.xml.
- **Manus Administrator**
  Stores NRCS aliases as configured in the **Newsroom settings** dialog.

  ⚠ No other information in `newsroomsettings.xml` is stored in the database.

Copyright © 2024 Vizrt

10.4 Prerequisites For Connecting To TDB

Before connecting to the template database the first time, make a backup of `channeltemplates.xml`, `AvConfig.xml` and `newsroomsettings.xml`

- Both AV Automation and Manus Administrator extract their information from the database and store the information locally in `channeltemplates.xml`, `AvConfig.xml` and `newsroomsettings.xml` respectively.
  Any existing files are overwritten.

This section continues with:

- Installing the Viz Mosart Template Database
- Creating and Upgrading a Viz Mosart Template Database
- Configuring Viz Mosart Server
- Backup and Recovery
- Database Maintenance
- Viz Mosart Template Database Specification

Copyright © 2024 Vizrt

10.5 Installing The Viz Mosart Template Database

This section presents both installing and creating a database for Viz Mosart templates.

- MySQL Database Servers and Clients
- MySQL Users
- Initial Mosart Template Database
  - File Location
  - Creating an Initial Template Database

Prerequisites

- A MySQL or equivalent, database server (MariaDB, AWS Aurora MySQL) installed and accessible from the Mosart server.
  This database server can be either dedicated to the Mosart automation servers or be an existing in-house or cloud-based server.
- A database user with credentials sufficient for creating and altering the database. This is also mandatory for Av Automation, which automatically updates the template database.
- A local MySQL client or tool that can connect and run sql scripts towards the database.

> **Note**: The Viz Mosart server can operate with *both* MySQL **5.x** and **8**.x.

10.5.1 MySQL Database Servers and Clients

- Wampserver: A complete Windows web development environment using Apache2 and PHP. It contains both MySQL and MariaDb database servers and supports PhpMyAdmin or Adminer for database management and configuration.

  > **Note**: Wampserver contains MySQL 5.x

  This method is recommended when installing a dedicated template database for the Mosart automation servers. It provides remote access to the database, via http.
- MySQL server: Use the MySQL Installer to install the last native MySQL database server. Recommended to also install the MySQL Workbench as a client for managing and configuring the database server.
- MySQL Workbench: Provides an integrated tools environment for database management and administration.
- MariaDB: Open source equivalent of MySQL server.
- HeidiSQL: Simple free MySql/ProgreSQL/Microsoft SQL client.
- dbVisualizer: Client for multiple database servers.

Refer to section Prerequisites for details of supported versions.

> **Verify the connection to the template database**

Copyright © 2024 Vizrt

Before configuring the Mosart Server, verify the *connection* between the template database and the Mosart Server, using a MySql client.
Typical issues at this stage relate to firewall and credentials/permissions.

10.5.2 MySQL Users

There are four types of users of a Mosart template database:

Role	Description	Application
Server management	Full rights, at least the possibility to create and alter databases	
Database management	Full rights within a single database / schema	Minimum rights for AV Automation
Database use (read/write)	Needs CRUD operations on a single database	Minimum rights for Manus Admin
Database use (read only)	Only need read operations on a single database	Minimum rights for ActiveX

It is recommend is to create *two* users for Mosart automation.
- One for the Mosart Server (AvAutomation and ManusAdmin) will full admin rights on the Mosart database.
- One for the Mosart ActiveX, which only requires rights to perform read operations.

10.5.3 Initial Mosart Template Database

File Location

The initial database is located in the *Sql* directory as part of the Mosart Server installation.

- Pick the *.sql* file with the highest version number.
  In this example, 1.0.5:

```
InitialMosartTemplateDatabase_1.0.5.sql
```

This sql file (containing the initial Mosart Template Database) is used to create the initial database on the database server.

Copyright © 2024 Vizrt

Creating an Initial Template Database

To Create a Template Database

You can create the initial template database in one of two ways:

1. By importing the initial database sql file using a database client, All MySql database clients given above supports import of sql scripts. Normally requires root access.
2. Using the MySQL command line tool *mysql.exe*. The script *RunDbScripts.cmd* contains command lines to import the initial database using mysql. This script is located in the *BatchFiles* directory as part of the Mosart Server installation.

   ⚠ Option 2 (below) requires that you have first created an empty database `mosarttemplatedb`.

   RunDbScripts.cmd

   ```
   ::: RunDbScripts.cmd:
   ::: Command script to upload an initial and empty database as the Mosart
   template database:

   ::: Option 1: Invoke mysql with user credentials on the command line.
   ::: Note localhost is assumed if host is not given.
   mysql [--host=hostname] --user=username --password=password mosarttemplatedb
   < ../Sql/InitialMosartTemplateDatabase_1.0.5.sql

   ::: Option 2: Invoke mysql with user credentials from the corresponding
   mysql.ini file.
   mysql --defaults-extra-file=mysql.ini < ../Sql/
   InitialMosartTemplateDatabase_1.0.5.sql
   ```

3. After import, a database *mosarttemplatedb* is created

mysql.exe

mysql.exe is part of MySQL Server, MySQL Workbench or Wampserver installations

Copyright © 2024 Vizrt

10.6 Creating And Upgrading A Viz Mosart Template Database

The Viz Mosart Template Database uses a schema called *mosarttemplatedb*.

Prerequisties

- Before continuing, ensure Viz Mosart has been installed. Refer to section Installation, and the latest Release Notes.
  A Viz Mosart installation includes an essential database creation batch file *RunDbScripts.cmd* written to: *%ProgramFiles%\Mosart Medialab\Mosart Server\BatchFiles.*
- An installed WampServer (refer to section Installing WampServer).

This section contains the following topics:

- Upgrading MySQL 5.x to MySQL 8.x
- Verifying the WampServer
- Running Database Scripts
- Running Scripts Automatically
- Running Scripts Manually

10.6.1 Upgrading MySQL 5.x to MySQL 8.x

Migrating the Mosart templatedb

In connection with the migration of the template database to the new version of the database server, it will be necessary to transfer all data from the old, following the recommendations in Template Database section Backup and Recovery.

To Migrate the Template Data

1. Transfer tables and data to the new server.
2. Run the command below.

   ```
   ALTER USER 'username'@'*' IDENTIFIED WITH 'mysql_native_password' BY
   'password';
   ```

   replacing `'username'` and `'password'` placeholders to their actual values.

10.6.2 Verifying the WampServer

To Verify WampServer is Running

1. Check the WampServer icon is present in the system tray.
   a. If not, start WampServer from the **Start** menu.

Copyright © 2024 Vizrt

   b. It should then appear *green* in the System tray.
2. If the WampServer icon is not green, click the icon and select **Start All Services**.

10.6.3 Running Database Scripts

The Viz Mosart Database can be created automatically using the *Db00Create.sql* script.

Location of Script

Navigate to the following locations, for x86 and x64 machines respectively:

- `%ProgramFiles%\Mosart Medialab\Mosart Server\Sql`
- `%ProgramFiles(x86)%\Mosart Medialab\Mosart Server\Sql`

There are also several *upgrade* scripts:

- Db01Change….sql
- Db02Change….sql
- and so on

Database changes are implemented as upgrade scripts rather than as new versions of the create script. This is to avoid destroying existing data.

- Scripts may also be run manually. It is recommended to run scripts automatically.
- If problems arise when running scripts automatically, more information on manually running scripts can be found later in this section.

10.6.4 Running Scripts Automatically

Scripts run from a batch file, *RunDbScripts.cmd*.

Location of Scripts

Navigate to the following locations, for x86 and x64 machines respectively:

- *%ProgramFiles%\Mosart Medialab\Mosart Server\BatchFiles*
- *%ProgramFiles(x86)%\Mosart Medialab\Mosart Server\BatchFiles*

The file refers to the MySQL command line tool mysql and an .ini file used by that tool.

- The .ini file is installed into the same folder as the batch file, but contents should be verified and amended as needed.
- The reference to the tool itself should also be verified.

To Run the Database Scripts

1. Edit, do not open, *RunDbScripts.cmd*. File contents should be similar to the following:

```
REM This file MUST be saved with Encoding = ANSI
for %%f in (<file list>) do C:\wamp\bin\mysql\mysql5.5.8\bin\mysql --defaults-
extra-file=mysql.ini < %%f
pause
```

Copyright © 2024 Vizrt

2. Check if the stated folder (in this example, *C:\wamp\bin\mysql\mysql5.5.8\bin*) exists and contains *mysql.exe*.
   If so, continue with step 5. If not, continue to step 3.
3. Locate *mysql.exe* and change the contents of *RunDbScripts.cmd* accordingly.
4. Save *RunDbScripts.cmd* with ANSI encoding.
   For example, in Notepad, choose **File > Save As**, then **Encoding: ANSI**.
5. Edit, do not open, *mysql.ini*. File contents should be similar to the following:

   ```
   [mysql] host=localhost database=mosarttemplatedb user=root password=
   ```

6. Verify entries in the file and change if necessary:
   - `host=localhost` : If MySql is running on another PC, replace localhost by the name or IP address of this other PC.
   - `database= mosarttemplatedb` : Should not be changed.
   - `user=root` : This is the default admin user. If you want to run the scripts with another user, replace with the name of a user with sufficient privileges.
   - `password=` : The default root password is empty. Replace if you have changed the root password or have replaced root.
7. Save the file if you have changed it.

10.6.5 Running Scripts Manually

To Run the Database Scripts Manually

1. Start phpMyAdmin.
2. Create the database, *mosarttemplatedb*, if not already done.
3. Upgrade the database using the scripts from the appropriate location:
   - *%ProgramFiles%\Mosart Medialab\Mosart Server\Sql*
   - *%ProgramFiles(x86)%\Mosart Medialab\Mosart Server\Sql*

Copyright © 2024 Vizrt

10.7 Configuring Viz Mosart Server

Both AV Automation and Manus Administrator can be configured to use the Viz Mosart Template Database.

- AV Automation: For storing templates and the xml configuration file, AvConfig.
- Manus Administrator: For storing NRCS *template type aliases* as configured in the newsroomsettings dialog.

> ⚠ All other information in *newsroomsettings.xml* is **not** stored in the database.

Both applications will extract information from the database and store the information locally on file:

- AV Automation: In file *channeltemplates.xml* and *AvConfig.xml*.
- Manus Administrator: In file *newsroomsettings.xml*.

> ⚠ Any existing files will be overwritten. Before connecting to the database for the first time, always back-up these files.

The following topics are presented:

- Connecting AV Automation to the Mosart Template Database
- Connecting Manus Administrator to the Mosart Template Database
- Viz Mosart Template Database and Mosart Web Apps NRCS Plugin
- Viz Mosart Template Database and Viz Mosart ActiveX

10.7.1 Connecting AV Automation to the Mosart Template Database

Connection String Settings

Connecting AV Automation to the Viz Mosart Template Database is done using the following settings in the general settings dialog (**CTRL + SHIFT + S**):

- **UseTemplateDb**: Set to *true* to enable connection to the Viz Mosart Template Database (Default: *false*).
- **ThisGallery**: The name of the DB gallery representing the gallery of this Viz Mosart installation. This DB gallery may not exist, but will be inserted when saving in A/V Setup or saving to DB in the Template Editor. Must be identical for both the Mosart main and backup servers, but a unique name for each gallery (Default: *NONSHARED*).
- **SharedGallery**: This setting SHOULD NOT be changed from its default value, and not be changed using the Settings dialogue. However, it *can* be changed by changing the MMAVAutomation.exe.config file, but this is *not* recommended (Default: *SHARED*). *SharedGallery* is the DB name of the (fictitious) gallery containing the shared template sets (containing the shared templates). If Viz Mosart has been installed to use the Template DB

Copyright © 2024 Vizrt

functionality, such a gallery has been inserted into the DB, along with a shared template set 'GLOBAL'. (The name of this template set MAY be changed in the Template editor.)

- **TemplateDbConnectionString**: Contains the connection string to the database.

  Example

  ```
  server=hostname;User Id=username;Password=password;database=mosarttemplatedb
  ```

- **TemplateDbDefaultInserter**: The name to be used for the _...*insertedby* and _... *updatedby* columns, the entities responsible for a database update. Only visible in the database itself where the *_insertedby* and *_updatedby* columns are populated with this property.
  Recommended set equal to *ThisGallery* + main/backup, to identify main and backup servers respectively.
- **TemplateDbEnableAutoSynchronize**: When *true,* Mosart Servers targeting the same gallery (*ThisGallery*) are synchronized automatically by AV Automation. Changes made to a template on one Mosart Server are automatically be obtained by the other Mosart Servers.
  This is typically used by Mosart main/backup servers. Recommended value: *true* (unless intensive show design with frequent template changes occurs).

  > ⚠ Only *idle* servers will update automatically. *Live* servers requires a manual operation, for example by opening the Template Editor, clicking on the Status bar or by restarting AV Automation.

- **TemplateDbPollIntervalSeconds**: Number of seconds between the Mosart server's polling of the database for updates. Used by the synchronization mechanism to check whether a synchronization operation is necessary. If set to *0*, 5 seconds is used. Recommended value: 5 seconds (Default: *0*, giving 5 seconds).
  Set to a higher value than the default (5 seconds) if multiple frequent template changes occurs.
- **TemplateDbEnableLocking** (deprecated): Leave default value (*true*). Only used on legacy systems, to ensure that only one Mosart server gets write access to a template in the template database at a time, preventing two users from simultaneously editing the same template.
- **TemplateDbProviderName**: The provider name for the Template DB. (Default: *MySql.Data.MySqlClient*).

For initial connection open the settings dialog and set the properties *UseTemplateDb*, *ThisGallery*, *TemplateDbConnectionString* and *TemplateDbDefaultInserter* then restart AV Automation. If everything goes fine an entry for the gallery should be added to the database and initial templates (*channeltemplates.xml*) and AvConfig (*avconfig.xml*) uploaded.

AV Automation Actions

When connected to the Template Database, different actions occur at the following times in AV Automation.

On launch of AV Automation

- avconfig.xml is replaced by the value tied to ThisGallery
  - If ThisGallery does not yet exist, any existing avconfig.xml is used as is, facilitating bringing an existing setup into Template Sharing.
- channeltemplates.xml is replaced
  - Any shared template sets are written as channels elements to the new channeltemplates.xml with a gallery attribute of SharedGallery.
  - Any template sets tied to ThisGallery are written as channels elements to the new channeltemplates.xml with a gallery attribute of ThisGallery.
  - If ThisGallery does not yet exist, any channels elements in the old channeltemplates.xml, except those with a gallery attribute of SharedGallery, are transferred as is to the new one. This avoids naming conflicts and facilitates bringing an existing set-up into Template Sharing.

When saving in A/V Setup

- If ThisGallery does not exist in the database, it is inserted
- The new contents of avconfig.xml is stored in the database tied to ThisGallery
- The contents of selected enumerations are stored separately in the database.
  - These are used when storing channeltemplates.xml in the database.
  - New or changed A/V Setup should always be saved before saving to database in the Template editor.

When saving to Template Database in the Template Editor

- If ThisGallery does not exist in the database, it is inserted.
- The new contents of channeltemplates.xml is stored in database
  Channel elements, Template Sets, with a gallery attribute of SharedGallery are stored as shared Template Sets - for example, tied to SharedGallery.
  Each template in these sets will have
  - A template description tied to SharedGallery
  - A template description tied to ThisGallery, the two descriptions will be equal
  - A template implementation tied to ThisGallery.
    Any existing descriptions and implementations tied to other galleries are retained.
- Other channels elements, including any new template sets are tied to ThisGallery. Each template in these sets will have:
  - A template description tied to ThisGallery
  - A template implementation tied to ThisGallery.

Manual Verification of a AV Automation Database Connection

1. Do the initial connection steps as outlined in the former section.
2. Using any MySql database client, verify the content of the *ga_gallery* table. This table shall now contain a single row where *ga_name*=ThisGallery.

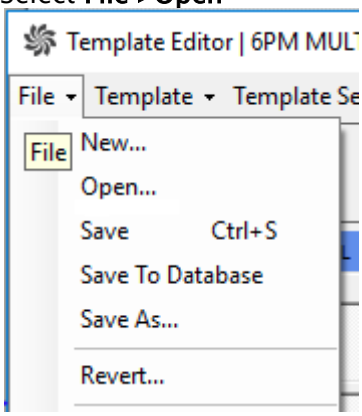
Copyright © 2024 Vizrt

Manual Import of channeltemplates.xml to the Template Database

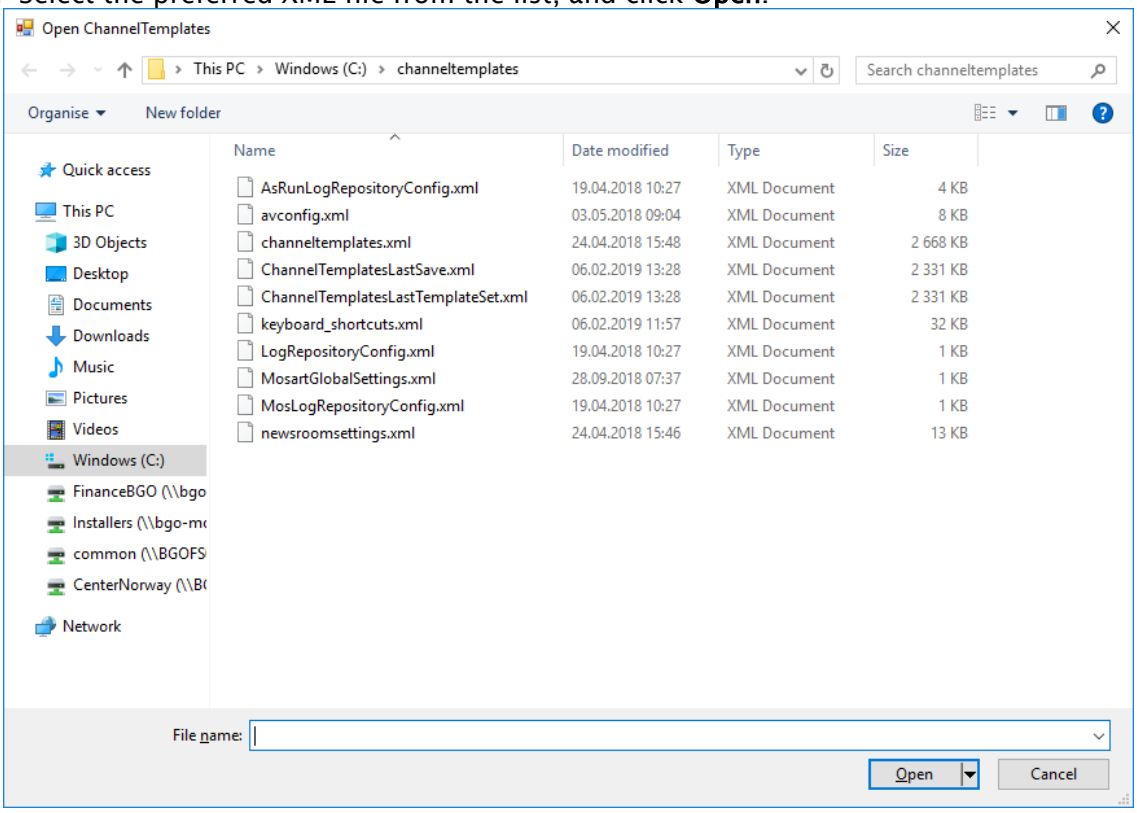
You can use a ready-made template set, stored as an XML-file. and manually import to the database. This is useful to populate the template database with the content from a *channeltemplates.xml* file. Either initially or as backup / transfer operation.

Import *channeltemplates.xml* file via the Template Editor in AV Automation. From inside the Template Editor:

1. Open any *channeltemplates.xml* from file.
   a. Select **File >Open**



   b. Select the preferred XML-file from the list, and click **Open**.



Copyright © 2024 Vizrt

> ⚠ After opening a template file in this way, **File > Save** will *not save to the database*, but to the **opened file**.
> Only after completing step [2] below, **File > Save To Database** does this operation save to the template database.

2. Select the **File > Save To Database**. The content of the selected *channeltemplates.xml* file is now uploaded to the Template Database.

> **Note:** When importing a *channeltemplates.xml* to the database any existing templates will be overwritten. No templates are deleted from the database.

Manual Import of AvConfig.xml to the Template Database

It is possible to manually import *AvConfig.xml* to the database. *AvConfig.xml* is stored entirely related to the gallery. Additionally, the lists of vision mixer crosspoints and effects are extracted and stored as global lists within the database. This information is used by the Viz Mosart ActiveX

Import *AvConfig.xml* via the A/V Setup dialog in AV Automation. From the A/V Setup dialog:

1. Open any *AvConfig.xml* file using the **File > Open** menu option.
2. Select the **File > Save** menu option. The content of the selected *AvConfig.xml* file is now stored in the template database.

> **Note:** The content of the *AvConfig.xml* file is stored as it is, as a string, in the *ga_gallery* table.

10.7.2 Connecting Manus Administrator to the Mosart Template Database

The only information stored in the database from Manus Administrator is the newsroom aliases for the Mosart template types. This is done to make the newsroom aliases the same for all galleries connected to the database. This information is global and shared among all galleries. Therefore there is no gallery-specific setting necessary within Manus Administrator (unlike *ThisGallery* for AV Automation)

Connecting Manus Administrator to the Mosart Template Database is done using the following settings in the general settings dialog (**CTRL + SHIFT + S**):

- *Use* **template** *database*: Set to true to enable connection to the Mosart Template Database.
- **ConnectionString**: Contains the database connection string to the database:

```
server=hostname;User Id=username;Password=password;database=mosarttemplatedb
```

- **Default inserter**: Identifies which entity is responsible for a database update.
- **Provider name**: Leave the default value for MySql (MySql.Data.MySqlClient).

Template Database Settings in Manus Administrator

In addition, the following settings are required for Template Database:

- TemplateDbConnectionString
- TemplateDbDefaultInserter
- TemplateDbProviderName
- TemplateToMosObject
- UseTemplateDb

For details see Settings Editor - iNews and Settings Editor - MOS.

Manual Import of newsroomsettings.xml Content to the Template Database

It is possible to manually import the NRCS template type aliases located in *newsroomsettings.xml* to the database.

1. In Manus Administrator, type newsroomsettings.
   This opens a dialog menu.
2. Open any newsroomsettings.xml using with **File > Open**.
3. Select the **File > Save**. The newsroom aliases for template types are now uploaded to the Template Database.

Note: It is only the *template type aliases* that are stored in the template database. All other information remains locally as is the *newsroomsettings.xml*.

10.7.3 Viz Mosart Template Database and Mosart Web Apps NRCS Plugin

Journalists use the Mosart Web Apps NRCS Plugin to insert and customize Viz Mosart template information into their NRCS script.

- From within their NRCS, the journalist search for a Viz Mosart template, customize the content, then drag and drop it into their script.
- When the script is saved, information is pushed to Manus Administrator creating a corresponding entry in the active rundown in the Viz Mosart GUI.

The Mosart Web Apps NRCS Plugin must be connected to your Viz Mosart Server.

Connection is handled in a simple configuration wizard at Web App installation, offering one of two connection choices:

1. The Template Database.
2. Directly to the Viz Mosart Server.

For more information, see the latest Mosart Web Applications Guide.

10.7.4 Viz Mosart Template Database and Viz Mosart ActiveX

The Viz Mosart ActiveX is a legacy method for inserting Viz Mosart template information into the NRCS script. This product is superseded by the Mosart Web Apps NRCS Plugin.

- The user selects a Viz Mosart type and variant to create an entry in the Viz Mosart rundown.
- When the script is saved, information is pushed to Manus Administrator and onto the active rundown in the Viz Mosart GUI.

Copyright © 2024 Vizrt

For the ActiveX to function correctly, some parameters are required to be set that link the ActiveX content to your Viz Mosart Server installation.

There are two ways to link the ActiveX to your Viz Mosart Server installation:,

1. The Template Database
2. Directly to the Viz Mosart Server.

Copyright © 2024 Vizrt

10.8 Backup And Recovery

The Viz Mosart Template Database is stored in a MySQL database and any backup and recovering strategies offered by the database vendor is valid.
In addition, most MySQL clients offers the possibility to backup and restore the entire database from a single .sql file.

In this section, the MySQL command line tools *mysqldump* and *mysql*are used for backup and restoring respectively.

Tip: *mysql.exe* and *mysqldump.exe* are part of MySQL Server, MySQL Workbench or Wampserver installations.

- Working with the Server Database
  - Database Backup
  - Database Restore
- Working with Database Clients
  - Backup and Restore from a MySQL Client

10.8.1 Working with the Server Database

Database Backup

To backup using mysqldump

- The command above creates a single .sql file containing the entire Viz Mosart Template Database.

```
mysqldump [--host=hostname] --user=username --password=password --add-drop-database
mosarttemplatedb > backupfile.sql
```

Database Restore

To Restore using mysql

- The command below restores the Viz Mosart Template Database from a single .sql file as created using the *mysqldump* command shown in the former section.

```
mysql [--host=hostname] --user=username --password=password mosarttemplatedb <
backupfile.sql
```

Copyright © 2024 Vizrt

10.8.2 Working with Database Clients

Backup and Restore from a MySQL Client

Backup and restore operations using the *mysqldump* and *mysql* tools are also available in all recommended MySQL client applications.

When using a client application ensure the following:

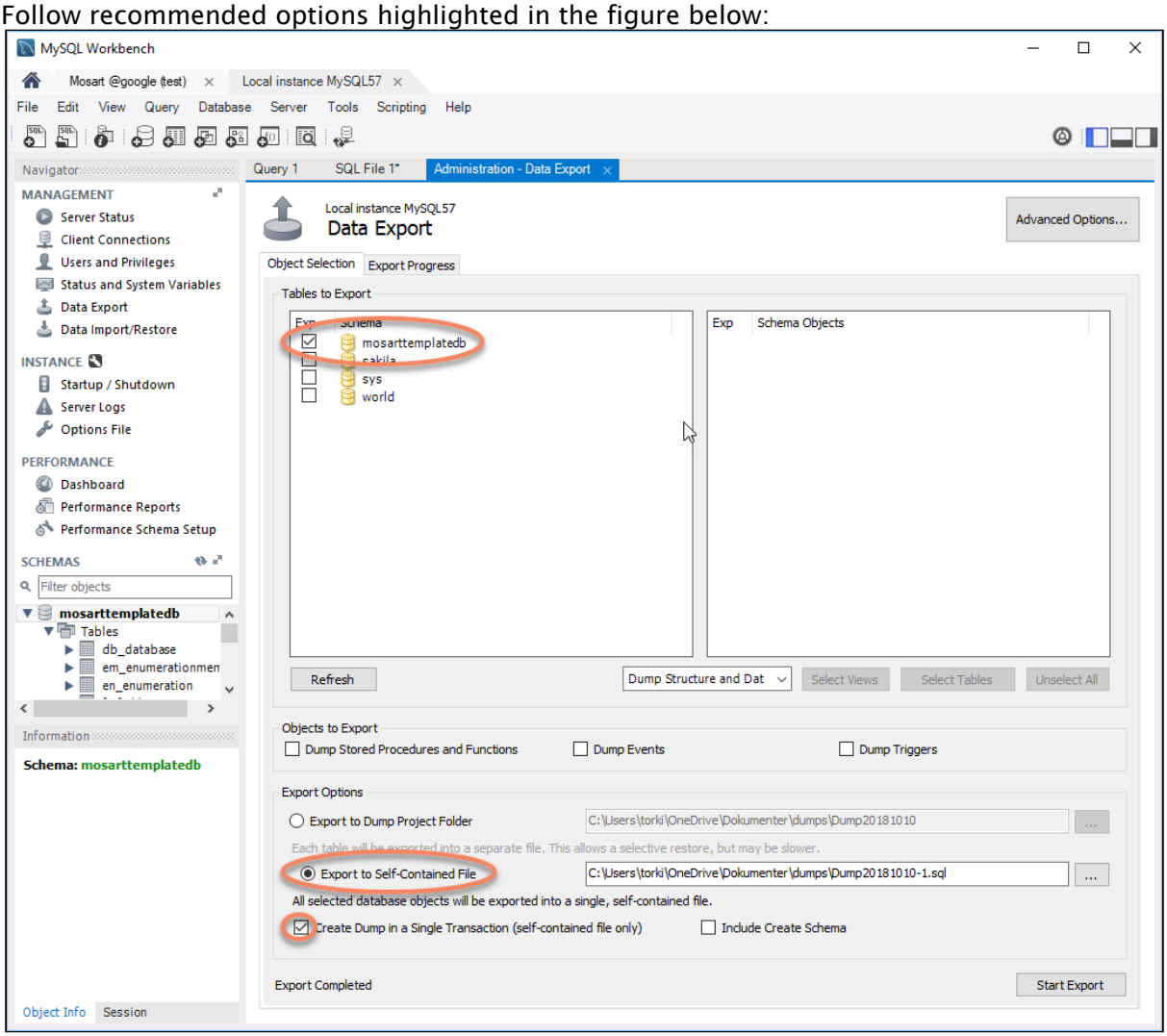
- Database schema = mosarttemplatedb.
- Export to a *single* file. The entire database must be saved to a single file.
- Include *drop* and *create* functions. This ensures that the database is restored as stored on the file.

Example: Backup from MySQL Workbench

1. In MySQL Workbench navigate to **Server > Data Export > Export**
   This opens the Data Export dialog menu.

Copyright © 2024 Vizrt

2. Follow recommended options highlighted in the figure below:

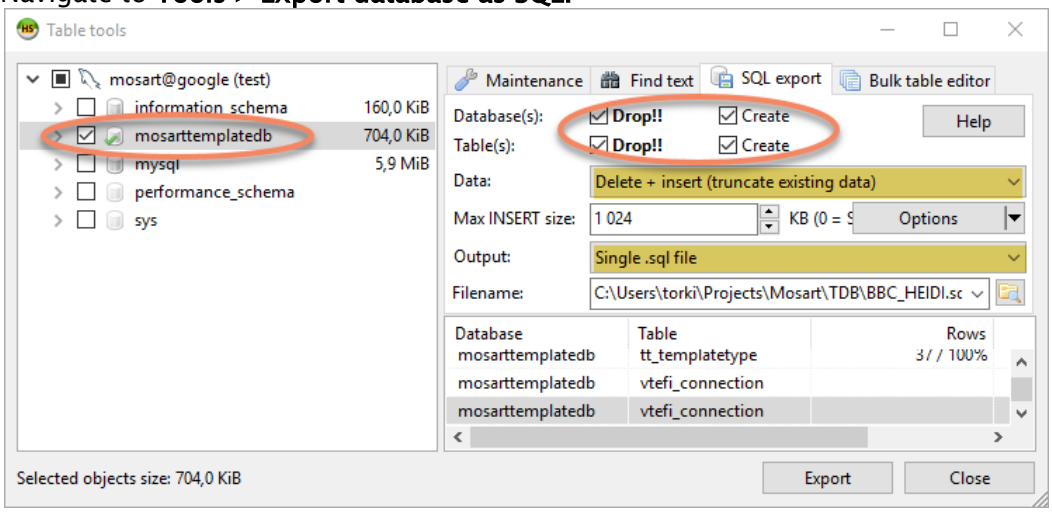


3. Restore from a single .sql file in MySQL Workbench.
   Navigate to **Server > Data Import**.
4. Select the .sql file as created during backup.

Copyright © 2024 Vizrt

Example: Backup from HeidiSQL

1. Navigate to **Tools > Export database as SQL.**



2. Ensure the following:
   a. For both **Database(s)** and **Table(s)**, *Drop* and *Create* are selected
   b. **Data**: is set to *Delete + insert (truncate existing data)*
   c. **Output**: is set to *Single .sql file.*
3. Navigate to **File > Run SQL file**
   and restore from a single .sql file in HeidiSQL.

Copyright © 2024 Vizrt

10.9 Database Maintenance

After initialization, most changes to the Viz Mosart Template Database are performed in AvAutomation and the Template Editor.

These changes may include upgrading the database, adding and removing galleries, adding and removing template sets and adding, removing and editing templates.

This section presents the following topics:

- Upgrading the Database
  - Upgrade History
- Viz Mosart Template Editor
- TemplateSetEditor
  - Configuration of TemplateSetEditor
  - Configuring the Required Template Database
  - Configuring an External Tool for Displaying Differences
  - Configuring an External Editor for XML Content in a Single Template
  - TemplateSetEditor Commands
  - Compare [set1] [set2] [type] [template]

10.9.1 Upgrading the Database

AvAutomation will automatically upgrade the database. All upgrades and current database version are stored in the db_database table.

> **Note:** Automatic upgrades shall never delete or modify any content in the database. The database should be kept backward compatible at all times.
> Upgrades normally add new functionality to the database

Upgrade History

The following table summarizes the upgrade history of the Viz Mosart Template Database:

Version	Date	Version	Description
0.1.0	01.06.2012	3.2	Initial version.
1.0.0	01.12.2012	3.2	First version of automatic upgrade. New table: db_database.
1.0.1	01.10.2013	3.2	Added support for sequences. New table: sq_sequence.

Copyright © 2024 Vizrt

Version	Date	Version	Description
1.0.2	07.05.2013	3.3	Added templateset properties. New table: tsp_templatesetproperties.
1.0.3	27.06.2013	3.3	Verified default transitions: Added initial content to the tr_transition table.
1.0.4	05.09.2013	3.5	Added order field in tdfi_connection table. New column: tdfi_order.
1.0.5	26.08.2014	3.5	Added scope to ta_templatetypealias table. New column: ta_scope.
1.1.0	17.01.2016	4.0	Added support for hierarchical template sets. New columns in ts_templateset table: ts_basedon, ts_hidefromuser.

10.9.2 Viz Mosart Template Editor

The Viz Mosart Template Editor is normally used to add, delete and modify Viz Mosart templates. When attached to a Mosart Template database every save operation pushes the changes incrementally to the database.

Use the Template Editor for the following changes:

- Adding and removing template sets. Note when adding a new template set it is possible to choose whether the template set shall be shared or gallery specific (local). It is not possible to change this afterwards.
- Adding, removing and editing templates. This is the normal use case of the template editor.

10.9.3 TemplateSetEditor

TemplateSetEditor is a console application that is part of the Mosart Test Suite. This application may be useful for simple maintenance and is currently the only way to remove galleries from the database.
The TemplateSetEditor was made as part of adding hierarchical templates to the database.

Note: As of Viz Mosart 4.0, the TemplateSetEditor is a standard component.

Configuration of TemplateSetEditor

Tip: Configuration of TemplateSetEditor is done in the application configuration file, TemplateSetEditor.exe.config

Copyright © 2024 Vizrt

userSettings

Within the application configuration file locate the userSettings sections where you can:

- Select an external tool for showing differences. Used by the **compare** function
- Select an external tool for viewing single templates or template sets in XML. Used by the **edit** function.
- Select the template database to use with corresponding credentials (connectionstring).
- Select template respository (future). Only *C:\channeltemplates.xml* is currently supported.

Configuring the Required Template Database

You can register three different databases to connect to.

1. Enter the respective connectionstrings using the properties *Database1*, *Database2* and *Database3*.
2. Select which database to use by assigning the desired server (hostname) to the *Database* property.

Example

- *Database1*: gallery=StudioE;server=localhost;User Id=user;password=password;database=mosarttemplatedb
- *Database2*: gallery=StudioE;server=146.148.9.188;User Id=user;password=password;database=mosarttemplatedb
- *Database3*: <empty> (not used)

Setting the property Database to localhost makes use of the connectionstring given in the *Database1* property, setting the *Database* property to 146.148.9.188 makes use of the connectionstring given in *Database2.*

Configuring an External Tool for Displaying Differences

You can reveal differences to template sets or templates by comparing their corresponding XML representations. This is used by the **compare** function. Register up to three different tools for comparison using the properties *DiffTool1*, *DiffTool2* and *DiffTool3*.
Use {0} and {1} as placeholders for the two different contents to compare. Then select which tool to use by setting the property *DiffTool* to the name of the desired tool.

Example

Tool	Code Example
DiffTool1: Visual Studio Code	Name=VSCode;Program=C:\Program Files\Microsoft VS Code\Code.exe;Args=--diff {0} {1}
DiffTool2: Beyond Compare	Name=BComp;Program=C:\Program Files\Beyond Compare 4\BCompare.exe;Args={0} {1}

Copyright © 2024 Vizrt

Tool	Code Example
DiffTool3: ExamXml	Name=ExamXml;Program=C:\Program Files\ExamXMLPro\examxmlpro.exe;Args=--diff {0} {1}

Setting the property *DiffTool* to BComp makes use of Beyond Compare as the external tool to use for the comparison.

Configuring an External Editor for XML Content in a Single Template

You can edit or view the XML content of a single template.

Register up to three different tools for editing xml using the properties *EditTool1*, *EditTool2* and *EditTool3*.
Use {0} as placeholder for the content to edit. Then select which tool to use by setting the property *EditTool* to the name of the desired tool.

Example

Tool	Code Example
EditTool1: Visual Studio Code	Name=VSCode;Program=C:\Program Files\Microsoft VS Code\Code.exe;Args={0}
EditTool2: XmlMarker	Name=XmlMarker;Program=C:\Program Files (x86)\XML Marker\xmlmarker.exe;Args={0}
EditTool3: Notepad++	Name=Notepad++;Program=C:\Program Files\Notepad++\notepad++.exe;Args={0}

Setting the property *EditTool* to VSCode makes use of Visual Studio Code as the external tool to use for editing.

TemplateSetEditor Commands

Below is a list of commands available in the TemplateSetEditor:

TemplateSetEditor help message

```
TemplateSetEditor
/?              Outputs this help message
Clear           Clears the console window
CleanUp         Clean up templates, removes depricated stuff etc.
Compare         [set] [baseset] [type] [template] Compares template sets or a single
template
Copy            [set] [newset] Makes a copy of set with name equals newset
```

Copyright © 2024 Vizrt

```
Edit             [set] [type] [template] Opens a single template in an external xml
editor
End              Terminates Mosart Template Set Editor
Equals           [set] [baseset] [type] [regex] Performs equality test on templateset
or set of templates
Exit             Terminates Mosart Template Set Editor
Flatten          [set] [type] [template] Flattens a templateset or a single template
Gallery          [gallery|add|select|delete|rename|list] [gallery] [newname] Gallery
operation when using template database, default select
Help             Outputs this help message
Info             [set] [type] [template] Outputs information of a templateset or a
single template
List             [set] [type] [regex] List templates / template sets
Open             {filename} Opens the given channeltemplates file
Quit             Terminates Mosart Template Set Editor
Rebase           [set] [baseset] [-c] [-f] Bases a templateset upon another set
Remove           [set] [type] [template]  Removes templateset or a single template
Rename           [set] [newname] Renames templateset
Save             Saves channeltemplates
SaveAs           [filename] [ChangesOnly] Saves channeltemplates to the given
filename. If ChangesOnly is given only changed templates are saved.
Verbose          Turns verbose logging to console on/off
```

Compare [set1] [set2] [type] [template]

Compares two template sets or a single template between two template sets. If any differences are encountered an external diff tool is launched to investigate the differences. Configuration of diff tool is done via the application configuration file, *TemplateSetEditor.exe.config*.
Syntax:

- **set1:** Name of template set one.
- **set2:** Name of template set two.
- **type:** Template type of single template (optional).
- **template:** Name or variant of single template (optional).

Example Compare

```
# Compares template sets LDN and LDN2
compare LDN LDN2
# Compares template Camera+1 from template sets LDN and LDN2
compare LDN LDN2 0 1
compare LDN LDN2 Camera 1
```

Copy [set] [newset]

Makes a copy of a template set with name equal to newset.

Copyright © 2024 Vizrt

Equals [set1] [set2] [type] [template]

Performs the equality test on two template sets or for a single template.
Syntax:

- **set1:** Name of template set one.
- **set2:** Name of template set two.
- **type:** Template type of single template (optional).
- **template:** Name or variant of single template (optional).

Example Equals

```
# Performs the equality test on the template sets LDN and LDN2
equals LDN LDN2
# Performs the equality test on template Camera+1 from template sets LDN and LDN2
equals LDN LDN2 0 1
equals LDN LDN2 Camera 1
```

Edit [set] [type] [template]

Opens a single template in an external xml editor. The tool to use is configured in the application configuration file, TemplateSetEditor.exe.config
Syntax:

- **set:** Name of template set.
- **type:** Template type of single template.
- **template:** Name or variant of single template.

Example: Edit

```
# Opens template Camera+1 in template sets LDN in an external xml editor
edit LDN 0 1
edit LDN Camera 1
```

Flatten [set] [type] [template]

Flattens an entire templateset to its base set or a single template.
Syntax:

- **set:** Name of template set to flatten.
- **type:** Optional: Template type of single template (optional).
- **template:** Name or variant of single template (optional).

Copyright © 2024 Vizrt

Example: Flatten

```
# Flattens template set LDN
flatten LDN
# Flattens template Camera+1 in template sets LDN
flatten LDN 0 1
flatten LDN Camera 1
```

Gallery [gallery|add|select|delete|rename|list] [gallery] [newname]

Gallery operations. Requires template database.

Tip: The recommended way to remove a gallery from the template database. Currently not possible from the template editor.

Syntax:

- **operation:** Gallery operation. Either add,select,delete,rename or list. Default = select.
- **gallery:** Name of gallery in the Mosart template database.
- **newname:** New gallery name when operation=rename.

Example: Gallery

```
# List all galleries in the template database
gallery list
# Selects StudioC as the current gallery, both examples are valid
gallery StudioC
gallery select StudioC
# Add StudioR as new gallery to the template database
gallery add StudioR
# Deletes StudioC from the template database. This operation removes all content
related to the gallery.
gallery delete StudioC
# Renames StudioB to StudioB2
gallery rename StudioB StudioB2
```

Info [set] [type] [template]

Outputs the XML content of a templateset or a single template to the console and the Windows clipboard.

Tip: Copies the XML content of a single template to the Windows clipboard.

Syntax:

- **set:** Name of template set.
- **type:** Template type of single template (optional).

Copyright © 2024 Vizrt

- **template:** Name or variant of single template (optional).

Example: Info

```
# Outputs information of template set LDN
info LDN
# Outputs information of template Camera+1 in template sets LDN
info LDN 0 1
info LDN Camera 1
```

List [set] [type] [regex] [file=filename]

List all templates sets or templates within a template sets.
Syntax:

- **set:** Template set to list templates from (optional).
- **type:** Restrict listing to templates of given types (optional).
  - '*' = All types
  - Comma separated list of valid types.
- **regex:** Restrict listing to templates matching the given regular expression (optional).
- **filename:** filename to store listed templates (optional).

Example: List

```
# List all template sets within the current channeltemplates.xml file
list
# List all templates within template set LDN
list LDN
# List all camera templates within template set LDN
list LDN 0
list LDN Camera
# List all camera templates within template set LDN where name contains "RIGHT"
list LDN 0 RIGHT
list LDN Camera RIGHT
# List all camera and live templates within template set LDN where name contains
"RIGHT"
list LDN 0,3 RIGHT
list LDN Camera,Live RIGHT
# List all templates within template set LDN where name contains "RIGHT"
list LDN * RIGHT
```

Open [filename]

Opens tempateset / *channeltemplates.xml* file. Default: *C:\channeltemplates\ChannelTemplates.xml.*

Copyright © 2024 Vizrt

Rebase [set] [baseset] [-c] [-f]

Bases a template set upon another template set. If a baseset is not given, the given template set is made 'stand alone' (i.e. not based upon any other template set).
Syntax:

- **set:** The template set to rebase.
- **baseset:** The template set to used as the baseset.
- **-c:** Makes a copy of the baseset if the given templateset does not exist.
- **-f:** Flattens template set after the rebase operation.

Example: Rebase

```
# Bases template set LDN2 to template set LDN
rebase LDN2 LDN
# Bases template set LDN2 to template set LDN.
# If LDN2 does not exist, a copy of LDN is made before rebasing
rebase LDN2 LDN -c
# Bases template set LDN2 to template set LDN.
# After rebase, a flattening templateset operation is performed
rebase LDN2 LDN -f
# Makes template set LDN2 'stand alone' (i.e. not based upon any other template set).
rebase LDN2
```

Remove [set] [type] [template]

Removes the given template set or a single template.

Syntax:

- **set:** Name of template set
- **type:** Template type of single template (optional).
- **template:** Name or variant of single template (optional).

Example: Remove

```
# Removes template set LDN
remove LDN
# Removes template Camera+1 from template sets LDN
remove LDN 0 1
remove LDN Camera 1
```

Rename [set] [newname]

Renames the given template set to the given new name.

Copyright © 2024 Vizrt

Save

Saves any changes to the template database and the current *channeltemplates.xml* file.

SaveAs [filename]

Saves the current templatesets to the given filename.

Copyright © 2024 Vizrt

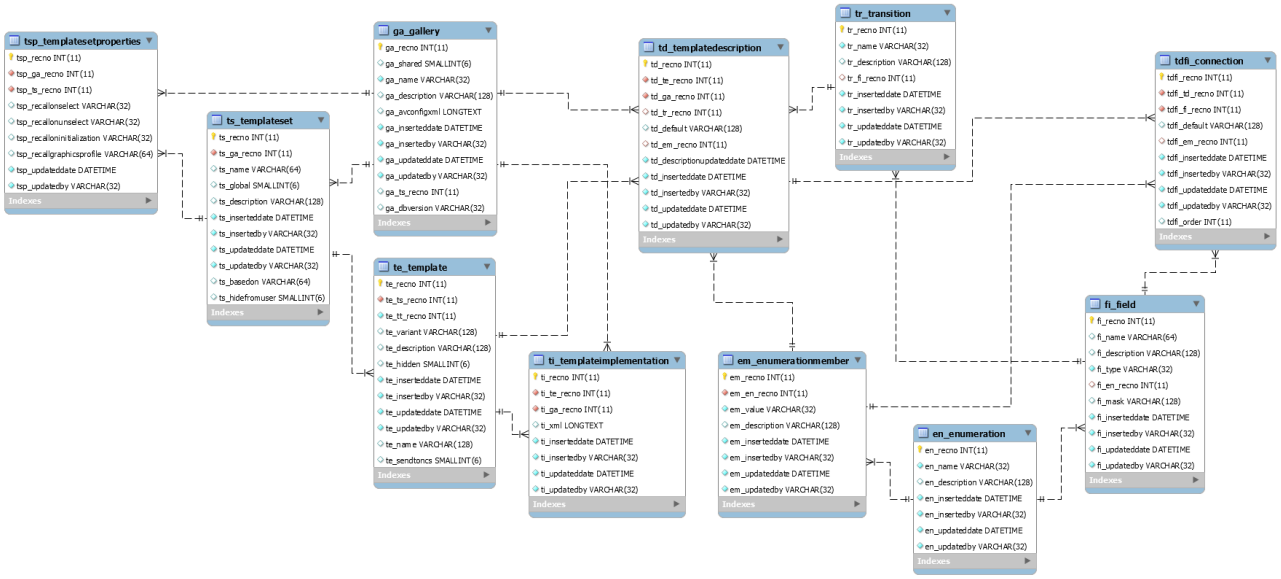
10.10 Viz Mosart Template Database Specification

This section provides database specifications for the Viz Mosart Template Database:

- Introductions and Notations
- Viz Mosart Database Tables

10.10.1 Entity Relationships

The following figure shows the relationships between the most important tables within the database.



Copyright © 2024 Vizrt

10.10.2 Introductions and Notations

By default Mosart organizes all templates in a single file, *C:\ChannelTemplates\channeltemplates.xml*. The template database makes it possible to store the templates in a MySQL database. The main benefits for this is follows:

1. To share templates among Mosart installations and galleries.
2. To provide synchronized template sets between Mosart main and backup servers.

Viz Mosart Template database tables

Table name	Prefix	Short description
db_database	db	Mosart Template Database update table. Contains an overview of database update.
em_enumerationmember	em	The enumeration members that may be used as values for template fields of enumeration type.
en_enumeration	en	The enumerations that may be used as types (i.e. sets of admissible values) for template fields.
fi_field	fi	The fields used as template inputs.
ga_gallery	ga	The galleries.
lo_lock	lo	Contains locks for editing database content.
sq_sequence	sq	Contains sequences stored as part of the Mosart template sets.
ta_templatetypealias	ta	The template type aliases (a template type may be referred to by several aliases).
td_templatedescription	td	The template descriptions. Input fields and transition input fields. A template may have different descriptions for different galleries.
tdfi_connection	tdfi	Relation, relating template description and fields.
te_template	te	The templates.

Copyright © 2024 Vizrt

Table name	Prefix	Short description
ti_templateimplementation	ti	The template implementations. A template may have different implementations for different galleries.
tr_transition	tr	Available transitions to be applied between templates.
ts_templateset	ts	The template sets (the templates are organized in sets).
tsp_templatesetproperties	tsp	Template set properties
tt_templatetype	tt	The Mosart template types.

Conventions and notations

Table prefix

Each table has a unique prefix consisting abbreviating the rest of the table name followed by an underscore (_). This prefix is used both in the table name and for all columns in the table. Using the prefix helps identifying relationships between column names and tables.

Syntax :

Table	<prefix>_<tableName>	ts_templateset
Column	<prefix>_<columnName>	ts_recno, ts_name
Foreign key	<prefix>_<foreignColumnName>	ts_ga_recno

Primary keys

Each table has an integer column …_recno (with … representing the prefix, except the _) representing an Primary key, and thereby as value of foreign keys, where applicable.

Foreign keys

A foreign key (referencing a …_recno column of some table) usually has the name …_…_recno (with the two …s representing the prefixes, still except the _, of the containing and referenced tables, respectively). (In the case that one table has more than one foreign key to the same table, other names have to be invented.)

Data types

The following data types are used in this specification:

Copyright © 2024 Vizrt

Type	Comment
datetime	Data type capable of storing a point in time (date and time of day).
varchar(n)	Variable length string.
integer	General integer.
smallint	Small integer, normally used to store boolean values (`0` =false, `1` =true).
longtext	Data type capable of storing a Unicode string of virtually any length.

Fixed columns

Every table contains the following four fixed columns to store update and insert information for the individual rows. Note that these columns are not stated in the forthcoming table descriptions.

Name	Type	Allow Null	Description
<prefix>_inserteddate	datetime	No	The point in time at which the row was inserted.
<prefix>_insertedby	varchar(32)	No	Mosart server responsible for the insert. Given by AvAutomation TemplateDbDefaultInserter property.
<prefix>_updateddate	datetime	No	The point in time at which the row was last updated (or inserted).
<prefix>_updatedby	varchar(32)	No	Mosart server responsible for the insert. Given by AvAutomation TemplateDbDefaultInserter property.

Copyright © 2024 Vizrt

10.10.3 Viz Mosart Database Tables

This section contains a brief overview of all tables contained in the Viz Mosart Database schema.

- db_database
- em_enumerationmember
- en_enumeration
- fi_field
- ga_gallery
- lo_lock
- sq_sequence
- ta_templatetypealias
- td_templatedescription
- te_template
- ti_templateimplementation
- ts_templateset
- tt_templatetype
- tdfi_connection
- tr_transition
- tsp_templatesetproperties

Copyright © 2024 Vizrt

db_database

The *db_database* table is a maintenance table that contains the current version of the Viz Mosart Template Database along with all its upgrades.

The table is used by Mosart AvAutomation to verify whether an automatic upgrade is required.

Name	Type	Allow Null	Description
db_recno	integer	No	Primary key.
db_version	integer	No	The database version number.
db_versiondate	varchar(32)	No	The date when the upgrade was introduced to the Mosart Template Database.
db_mosartversion	varchar(32)	Yes	The Viz Mosart version number responsible for doing the upgrade.

Copyright © 2024 Vizrt

em_enumerationmember

The em_enumerationmember contains the enumeration values for the corresponding enumeration types.

Name	Type	Allow Null	Description
em_recno	integer	No	Primary key.
em_en_recno	integer	No	Foreign key referencing the corresponding enumeration in the *en_enumeration* table.
em_value	varchar(32)	No	The value of the enumeration member. Normally presented to the users.
em_description	varchar(128)	Yes	Optional enumeration member description.

Copyright © 2024 Vizrt

en_enumeration

The table *en_enumeration* contains all enumerations used to set values for template fields.

Name	Type	Allow Null	Description
en_recno	integer	No	Primary key.
en_name	varchar(32)	No	The enumeration name. Normally used in drop-down lists presented to users.
en_description	varchar(128)	Yes	Optional enumeration description.

Copyright © 2024 Vizrt

fi_field

The table *fi_field* contains valid fields or newsroomtags used by Mosart templates for receiving user input. Normally these fields are exposed to the uses in the NRCS via dedicated ActiveX components. Note that a single field can be shared by many templates.

Name	Type	Allow Null	Description
fi_recno	integer	No	Primary key.
fi_name	varchar(32)	No	The field name. Normally presented to users.
fi_description	varchar(128)	Yes	Optional field description.
fi_type	varchar(32)	No	The field type. Supported values are 'ENUMERATION', 'NUMBER', 'STRING'.
fi_en_recno	integer	Yes	Optional foreign key for ENUMERATION fields. If set, referencing the corresponding enumeration in the en_enumeration table.
fi_mask	varchar(128)	Yes	Optional input mask. Used for STRING fields only.

Copyright © 2024 Vizrt

ga_gallery

The table *ga_gallery* contains all Mosart galleries using the database.

> **Note:** In Mosart AvAutomation the property *ThisGallery* shall reflect the gallery name as stored in the *ga_name* column.

Name	Type	AllowNull	Description
ga_recno	integer	No	Primary key.
ga_shared	smallint	Yes	Set to true (1) for the *shared* gallery. The *shared* gallery may be used by clients to obtain all shared template sets within the database. Only one gallery shall be used as the *shared* gallery. By default, a virtual gallery named SHARED is used for this purpose.
ga_name	varchar(32)	No	The gallery name.
ga_description	varchar(128)	Yes	Optional gallery description.
ga_avconfigxml	longtext	Yes	The XML formatted contents of the *avconfig.xml* file used in this gallery.
ga_ts_recno	integer	Yes	Optional foreign key referencing the 'default' template set of the gallery.

Copyright © 2024 Vizrt

Name	Type	AllowNull	Description
ga_dbversion	string	Yes	The database version used by the Mosart applications for the gallery. This database version reflects the current Mosart version.

Note: One row only should have *ga_shared* = 1. The name of this gallery is by default set to 'SHARED'. All shared templates belong to this gallery.

Copyright © 2024 Vizrt

lo_lock

The table *lo_lock* contains information used to restrict editing access to a template. The template database locks at template level (i.e. only one user may edit a single template at any time).

The lock table also contains rows used to store timestamps for last updates within a single gallery or the database overall. These values may be used to poll the database for changes. These timestamps are stored in the *lo_updatedate* field.

Note that the *lo_insertedby* field is used to identify the lock type:

- **Template:** This is a true lock, locking the corresponding template. The template is identified by the value in the *lo_recno* column which contains a foreign key to the *te_recno* value in the *te_template* table.
- **Gallery:** Contains a timestamp when the corresponding gallery was last updated. The gallery is identified by the *lo_recno* which for gallery types contains a foreign key to the corresponding gallery, *ga_recno* in the *ga_gallery* table. The foreign key is a function of the *lo_recno* value as follows: ga_recno = 10 - lo_recno
- **Database:** Contains a timestamp when the database was last updated.
- Other values are deprecated.

Name	Type	Allow Null	Description
lo_recno	integer	No	Primary key. Also used to identify either a gallery or a template depending upon the lock type. See description above.
lo_ga_recno	integer	Yes	Optional foreign key referencing the gallery owning the lock.
lo_ga_name	varchar (32)	Yes	The name of the gallery owning the lock.
lo_insertedby	varchar (32)	No	Contains the lock type. Either Template, Gallery or Database. See description above.

Copyright © 2024 Vizrt

sq_sequence

The *sq_sequence* table contains recorded sequences from a Viz Mosart Automation. Sequences are gallery dependent and related to the template set in use when recorded. A sequence makes it possible to record any sequence from a rundown in Viz Mosart automation for later playback.

Name	Type	Allow Null	Description
sq_recno	integer	No	Primary key.
sq_ga_recno	integer	No	Foreign key referencing the corresponding gallery *ga_recno* in the *ga_gallery* table.
sq_ts_recno	integer	No	Foreign key referencing the corresponding template set *ts_recno* in the *ts_templateset* table.
sq_name	varchar(32)	No	Name of sequence.
sq_description	varchar(128)	Yes	Optional description of the sequence.
sq_xml	longtext	No	The gallery specific implementation of the sequence. Stored as received from Mosart Automation in XML.

Copyright © 2024 Vizrt

ta_templatetypealias

The *ta_templatetypealias* table contains custom user aliases for the Mosart template types. These aliases are used by Mosart ManusAdministrator to translate user aliases to the corresponding Mosart template types.

Name	Type	AllowNull	Description
ta_recno	integer	No	Primary key.
ta_name	varchar(32)	No	The template type alias name. Should be unique, as this is presented to the users.
ta_description	varchar(128)	Yes	Optional template type alias description.
ta_tt_recno	integer	No	Foreign key referencing the corresponding Mosart template type in the *tt_templatetype* table.
ta_default	smallint	Yes	A value of 1 is used for the 'default' alias among those belonging to the same template type. For each template type, exactly one alias belonging to that type should have ta_default = `1`.

Information: The ta_templatetypealias table is the only table that is maintained and used by Mosart ManusAdministrator containing the NRCS template type aliases.

Copyright © 2024 Vizrt

td_templatedescription

The table *td_templatedescription* contains template information to be presented to users that may differ among the galleries (i.e. this information is gallery dependent).

Name	Type	Allow Null	Description
td_recno	integer	No	Primary key.
td_te_recno	integer	No	Foreign key referencing the corresponding template *te_recno* in the *te_template* table.
td_ga_recno	integer	No	Foreign key referencing the corresponding gallery *ga_recno* in the *ga_gallery* table
td_tr_recno	integer	Yes	Optional foreign key referencing a transition *tr_recno* in the *tr_transition* table.
td_default	varchar(128)	Yes	Optional default value for transitions not of type ENUMERATION. Normally for MIX and WIPE transitions.
td_em_recno	integer	Yes	Optional foreign key for the default value for ENUMERATION transition types. Normally EFFECT transition. Reference to a enumeration value em_recno in the em_enumerationmember table.
td_descriptionupdateddate	Datetime	No	(deprecated)

Note that the overall template description is stored in three parts / tables:

1. **td_templatedescription:** The base template description.
2. **tr_transition:** The transition used by the template. In Mosart this equals the transition when taking the template. Normally either Mix, Wipe or Effect.
3. **tdfi_connection:** Relation that list all fields (or newsroomtags) used by the templates. A newsroomtag is a template property allowing users to input information to the template. Normally newsroomtags are exposed to the users in the NCS.

Copyright © 2024 Vizrt

te_template

The table *te_template* contains the individual templates with information that is shared among all galleries.

Note that information regarding a single template is divided into three parts / tables:

1. **te_template:** Contains basic template information. This information is shared among all galleries (i.e. gallery independent).
2. **td_templatedescription:** Contains typically template information to be presented to users that is gallery dependent.
3. **ti_templateimplementation:** Contains the template implementation as created by the Mosart template editor. This information is gallery dependent.

Name	Type	Allow Null	Description
te_recno	integer	No	Primary key.
te_ts_recno	integer	No	Foreign key referencing the corresponding template set in the *ts_templateset* table.
te_tt_recno	integer	No	Foreign key referencing the corresponding template type in the *tt_templatetype* table.
te_variant	varchar(32)	No	The Mosart template variant.
te_description	varchar(128)	Yes	Optional template description.
te_hidden	smallint	Yes	Set to true (1) if this template is to be hidden from any user presentations.
te_name	varchar(128)	Yes	Optional template name.
te_sendtoncs	smallint	Yes	Set to true (1) if the template implementation is to be sent to NCS. Used by some NCS only.

Copyright © 2024 Vizrt

ti_templateimplementation

The table *ti_templateimplementation* contains the gallery specific template implementation as created by the Mosart template editor (i.e. this information is gallery dependent).

Name	Type	Allow Null	Description
ti_recno	integer	No	Primary key.
ti_te_recno	integer	No	Foreign key referencing the corresponding template *te_recno* in the *te_template* table.
ti_ga_recno	integer	No	Foreign key referencing the corresponding gallery *ga_recno* in the *ga_gallery* table.
ti_xml	longtext	Yes	The actual xml-formatted template implementation as created by the Mosart template editor.

Information: The template implementation must be created per gallery and is stored using its native XML representation, as received from the Mosart template editor

Copyright © 2024 Vizrt

ts_templateset

The table *ts_templateset* contains the template sets.

Name	Type	Allow Null	Description
ts_recno	integer	No	Primary key.
ts_ga_recno	integer	No	Foreign key referencing the corresponding gallery in the *ga_gallery* table. Template sets belonging to the shared gallery (*ga_shared*=1) are by definition *shared* template sets. Shared template sets will be present in all galleries. Template sets belonging to any other gallery are by definition *private* template sets owned by the corresponding gallery.
ts_name	varchar(64)	No	The name of the template set.
ts_global	smallint	Yes	(deprecated)
ts_description	varchar(128)	Yes	Optional template set description.
ts_ga_updating	integer	Yes	(deprecated)
ts_basedon	varchar(64)	Yes	Hierarchical template sets. Specifies the base template set, if applicable.
ts_hidefromuser	smallint	Yes	Non-zero (true) if the template set shall be hidden from users (i.e. from Mosart GUI and NCS presentations).

Copyright © 2024 Vizrt

tt_templatetype

The table *tt_templatetype* contains the Mosart template types.

Name	Type	Allow Null	Description
tt_recno	integer	No	Primary key.
tt_name	varchar(32)	No	The template type name. Should be unique, as this is presented to the users.
tt_description	varchar(128)	Yes	Optional template type description.

The Mosart template types

Id: tt_recno	Name: tt_name	Description
-1	UNKNOWN	
0	CAMERA	
1	PACKAGE	
2	VOICEOVER	
3	LIVE	
4	FULLSCREENGRAPHICS	
5	DVE	
6	JINGLE	
7	TELEPHONEINTERVIEW	
8	ADLIBPIX	
9	BREAK	
10	MACRO	

Copyright © 2024 Vizrt

Id: tt_recno	Name: tt_name	Description
17	VIDEOWALL	
25	CLIP	
100	LOWERTHIRDS	
110	OVERSHOULDERGRAPHICS	
215	PLAYSOUND	
216	SETSOUNDLEVEL	
217	LIGHT	
220	ACCESSORIES	
250	SETCROSSPOINT	
275	COMMAND	
310	EXTERNALCOMMAND	
320	TEXTTELEPROMPTER	
330	CLIPDUR	
331	ENDFRASE	
332	STORYNR	
333	ATTACHA	
334	CLIPSTART	
335	CLIPHIRESPATH	
340	GRAPHICSID	
350	MARKER	
360	TIMINGINFO	

Copyright © 2024 Vizrt

Id: tt_recno	Name: tt_name	Description
999	IGNORE	

Copyright © 2024 Vizrt

tdfi_connection

The table *tdfi_connection* which contains a list of fields (or newsroomtags) that corresponds to a template description. See also information given in the td_templatedescription table.

This is a relation table defining a many-to-many relationship between the *td_templatedescription* and *fi_field* tables.

Name	Type	Allow Null	Description
tdfi_recno	integer	No	Primary key.
tdfi_td_recno	integer	No	Foreign key referencing the corresponding template description td_recno in the td_templatedescription table.
tdfi_fi_recno	integer	No	Foreign key referencing the corresponding field fi_recno in the fi_field table.
tdfi_default	varchar(128)	Yes	Optional default value for fields not of type ENUMERATION. I.e for STRING and NUMBER types.
tdfi_em_recno	integer	Yes	Optional foreign key for the default value for ENUMERATION field types. Reference to a enumeration value em_recno in the em_enumerationmember table.

Copyright © 2024 Vizrt

tr_transition

The table *tr_transition* contains the transition types available for Mosart templates. A transition is defined as the transition to take when taking a template (i.e. the transition from the former template).

Name	Type	Allow Null	Description
tr_recno	integer	No	Primary key.
tr_name	varchar(32)	No	The transition name.
tr_description	varchar(128)	Yes	Optional transition description.
tr_fi_recno	integer	Yes	Optional foreign key referencing an input field, *fi_recno* in the *fi_field* table.

Copyright © 2024 Vizrt

tsp_templatesetproperties

The *tsp_templatesetproperties* table contains some optional gallery specific template set properties.

Name	Type	Allow Null	Description
tsp_recno	integer	No	Primary key.
tsp_ga_recno	integer	No	Foreign key referencing the corresponding gallery ga_recno in the ga_gallery table.
tsp_ts_recno	integer	No	Foreign key referencing the corresponding template set ts_recno in the ts_templateset table.
tsp_recallonselect	varchar(32)	Yes	Optional direct take to be recalled when the template is selected in Viz Mosart automation.
tsp_recallonunselect	varchar(32)	Yes	Optional direct take to be recalled when the template is unselected in Viz Mosart automation.
tsp_recalloninitialization	varchar(32)	Yes	Optional direct take to be recalled when a rundown is reloaded.
tsp_recallgraphicsprofile	varchar(32)	Yes	Optional direct take to be recalled when a graphics profile (or graphics concept) is selected in Viz Mosart automation.

Copyright © 2024 Vizrt

11 Maintenance

This section contains an overview of log configuration and maintenance procedures that may help streamline aspects of your broadcast environment.

This section contains the following topics:

- System Logging
- Server Maintenance
- General Advice on System Operations
- Redundancy

Copyright © 2024 Vizrt

11.1 System Logging

All logging in Viz Mosart is done through the Log Service. This service typically runs on the server hosting the Viz Mosart Server and is accessed through Windows Services.

All applications transmit log events to the server, including Viz Mosart GUI. When the Viz Mosart GUI is running on a different machine than the Viz Mosart server (which is the normal case), a selected subset of local log messages are transferred, via the Remote Control Service, to the Viz Mosart server. This enables these messages to also be logged in the Viz Mosart server logs.

There are three distinct log streams available:

- The Technical Log, which contains the normal logging from all applications.
- The As-run Log, which contains information about on-air events.
- The MOS Log, which contains information about MOS communication between NCS (MOS based) and Manus Administrator.

> **Note**: For setups where several Mosart applications are installed on the same machine and these are using their own logging service (for example, Viz Mosart UI, Viz Mosart Audio Player and Viz Mosart Server), if logging services do not provide the expected results, please ensure that the Viz Mosart Server was the *last* component to be installed and that Mosart Logging Service points to the Mosart Server installation folder. This is mainly relevant if one of the log adapters supported by Mosart is used. See section Log Adapters.

This section contains the following topics:

- Log Viewer
- Log File Characteristics
- Log Content
- Log Adapters
- Configuring Logging
- Starting-up a Log Adapter
- Troubleshooting Log Adapters

11.1.1 Log Viewer

A Viz Mosart installation includes a dedicated log file viewing utility. **Log Viewer** is in the Viz Mosart Server installation directory and is also available from Vizrt Support as a separate installation package.

> **Tip:** To improve filtering and visualization, the XML log files are readable in Microsoft Excel.

Copyright © 2024 Vizrt

11.1.2 Log File Characteristics

Location

Viz Mosart logs' default directory is:

- *C:\MMLogs*
  The technical log stream is located at the root of this directory, while the other two log streams have their own corresponding subdirectories.
  The location is configurable.

Format

All Viz Mosart logs are stored as XML files.

File Naming

Log file names are configurable, but typically follow the pattern:

```
{0}.{1:yyyyMMdd-hhmmss}.xml
```

Where:

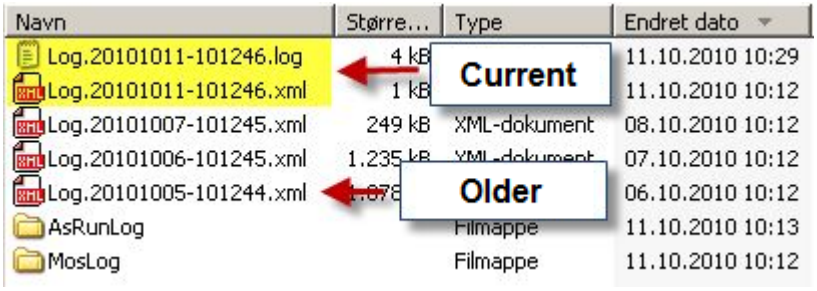
- {0} - User configurable prefix.
  Default: *Log*, *AsRunLog* and *MosLog* for the three logs respectively.
- {1} - File creation time. “year-month-day hour-minutes-seconds”

Current and Historical

There are two sets of log files:

- The current, active log file.
  - For efficiency purposes, this log is divided into two separate files; one XML file and one log file.
  - When reading the current file it is recommended to open the XML, as this contains a reference to the corresponding log file.
- Historical log files.
  - When a new log file starts, the current active log is merged into a single XML. The figure below illustrates a default setup, with a list of log files belonging to the technical log and two directories, each containing similar log files for as-run and MOS logs respectively.

Copyright © 2024 Vizrt



11.1.3 Log Content

Technical Log

This log contains standard log events, as produced by the running Viz Mosart Applications.

> ⚠ **Note:** This logging includes any Viz Mosart application normally run on other host machines, but belonging to the same control room.

Event Types

All events in the technical log stream are classified according to the event types as below:

- **AsRunLogger:** See As-run Log.
- **AudioMixer:** Events to/from the audio mixer.
- **AudioPlayer:** Events from the Audio Player and messages from AV Automation to Audio Player.
- **AvAutomation:** General events issued from AV Automation.
- **ConsoleController:** Used by default by GenericController. For generic console applications (services).
- **Crossover:** Events to/from the Crossover component.
- **Database:** Events associated with the SNMP Service.
- **GeneralInfo:** General events not assigned to any device or application.
- **GPIControl:** Events to/from the GPI control.
- **Graphics:** Events related to control of graphics.
- **GUI:** Events from the Viz Mosart GUI, such as mouse and button events.
- **GUILocal:** Used to log messages to the local log service from the Viz Mosart GUI. This is an extended set of local GUI events, logged to local GUI PC. Excludes pressed keys, these are sent to the Server log.
- **INewsWebServices:** Events associated with iNews web service.
- **IntegratedEngine:** Events to/from the integrated engine.
- **Instrumentation:** Events related to the Instrumentation component in AV Automation.
- **LightControl:** Events to/from the light control.
- **Loudness:** Events to/from the loudness control.
- **ManusAdministrator:** General events issued from Manus Administrator.
- **MediaAdministrator:** General events issued from Media Administrator.

Copyright © 2024 Vizrt

- **MediaAssetManagement:** Used principally for the Amadeus component.
- **MediaRouter:** Media router events.
- **Mimic:** Events associated with Mimic functionality.
- **MosartRemote:** Events related to remote control of Viz Mosart.
- **MosConnection:** MOS events between MOS based NCS and Viz Mosart.
- **RoboticCameraControl:** Events to/from the robotic camera control.
- **RouterControl:** Events to/from the router control.
- **SoundFilePlayer:** Events to/from the Sound File Player.
- **SpeakNoticer:** Events associated with the Asio Speak Notifier (for Mimic).
- **Tally:** Events to/from the tally control.
- **TemplateSharing:** Template sharing events.
- **Texting:** Events to/from the subtitles control.
- **TimeDisplay:** Events associated with Timing Info display.
- **Timing:** For Timing Display logging.
- **UserMessage:** Logs user messages from Control Commands.
- **VideoServer:** Events to/from the video server controllers.
- **VideoSwitcher:** Events to/from the video switcher control.
- **VideoWallController:** Events to/from the video wall control.
- **VirtualSet:** Principally for BrainstormVS component.
- **WeatherControl:** Events to/from the weather control.

Verbose Logging

Similar to other Viz Mosart applications, the Viz Mosart GUI can be configured to perform verbose logging of application events. Verbose logging is only intended for troubleshooting, so by default it is turned *off*.

> **Note**: When the Viz Mosart GUI is running on a different machine than the Viz Mosart server (which is the normal case), a selected subset of local log messages are transferred, via the Remote Control Service, to the Viz Mosart server, enabling these messages to be logged in the Viz Mosart Server logs.
> **It is important to note** that when the Viz Mosart GUI runs on a separate machine, filtering of verbose log messages ***will not, by default work as intended***.
> The unwanted effect is *all* verbose log messages are transferred to the Viz Mosart server, even if verbose logging is turned *off* in the Viz Mosart GUI.
>
> - This issue was addressed in Viz Mosart 5.5, but as the setting is local and preserved at upgrade, any earlier (not a clean installation of 5.5) version of Viz Mosart **requires a configuration change** in the Viz Mosart GUI configuration file *%localappdata%/Mosart_Medialab/MosartMultiGui.exe_.../user.config*, **changing "VerboseLevel" from the default "*0*" to "*-1*"**.

All Viz Mosart Applications have a set of configuration properties to control what events are passed to the Log Service:

- Verbose:
  - Turns on/off verbose logging.
- VerboseIgnoreEvents:

Copyright © 2024 Vizrt

    - A comma separated list of log event types to ignore during verbose logging.
    - Default: <empty> (no events are ignored).
- VerbosePassEvents:
    - A comma separated list of log event types that are passed during verbose logging.
    - Default: <empty> (all events are passed).

> **Example:** Setting `VerbosePassEvents = “VideoServer,AudioMixer”` ensures that only verbose events to/from video servers and audio mixer are passed to the Log Service.

As-run Log

This log stream contains events connected to content that has been on-air, typically all video and graphic content. The Log Service may be configured to extract information from the as-run log, for input into media asset management or presentation automation systems. All events in the as-run log stream are classified according to the event type, as shown below:

- **CROSSPOINT:** Issued as response to a crosspoint change from a tally feedback.
- **LOG:** Issued at start and stop of logging to the as-run log stream.
- **NEXT_STORY:** Issued when the next story element for on-air is changed.
- **RUNDOWN_RELOAD:** Issued when the rundown is reloaded.
- **STORY_ENDED:** Issued when the current story is ended.
- **STORY_STARTED:** Issued when the current story is started.
- **TAKE_EXTERNALS:** Issued when externals are taken on air.
- **TAKE_ITEM:** Issued when a story item is taken on air.

> **Note:** Customer specific as-run Log Adapters will use a unique syntax.

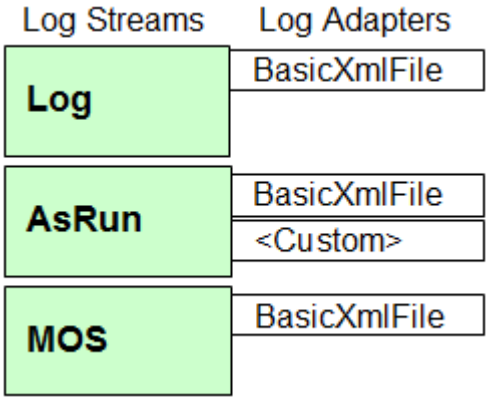
MOS Log

This log stream contains events communicated between a MOS-based Newsroom Computer System and Manus Administrator.

11.1.4 Log Adapters

Each log stream may be processed simultaneously by a set of *log adapters* as illustrated in the figure below:

Copyright © 2024 Vizrt



In this example, three log streams are configured as follows:

- The Technical Log Stream has one adapter:
  - **BasicXmlFile**: A default adapter that creates XML-based log files.
- The AsRun Log Stream has two adapters:
  - **BasicXmlFile**: As above, the default adapter for creating XML-based log files.
  - **Custom adapter**: Processes the as-run log stream according to site specific-requirements. These adapters are custom built, and not included in the Viz Mosart installer.
- The MOS Log Stream has one adapter:
  - **BasicXmlFile**: As above.

11.1.5 Configuring Logging

You can configure log content, granularity and naming.

- Configuration Files
- Configuration of Log Adapters
- Configuration of Log Filters
- Customizing an As-run Log Adapter

Configuration Files

Each log has its own configuration file, enabling adjustment of log properties like filename, or how often a new log file is generated.
The log configuration files are **located** in the following directory:

```
%ProgramFiles%\Mosart Medialab\Mosart Server\ConfigurationFiles
```

The log configuration files are **named** as follows:

- **Technical log**: *LogRepositoryConfig.xml*.
- **MOS log:** *MosLogRepositoryConfig.xml*.
  This log file comprises three segments:
  - **Adapters**: Configuration of one or more log adapters.
  - **FilterBank**: Configuration of optional log filters.

Copyright © 2024 Vizrt

    - **Properties**: Configuration of log properties.
- **As-run log**: *AsRunLogRepositoryConfig.xml*.

Example of an As-run Log

A sample log configuration file for the as-run log stream:

```
<?xml version=”1.0” encoding=“utf-8“ ?>
<LogRepositoryConfig>
<!- No specified adapters shall result in default adapter -->
<Adapters>
<!- Default adapter -->
<Adapter type=“BasicXmlFileAdapter“ name=“AsRunLog“ />
<!- Custom adapter -->
<Adapter type=“LogAdapterDRMorpheus“ name=“DRMorpheus“
configuration=“DRMorpheusAsRunLogConfig.xml“ assembly=“LogAdapterDRMorpheus.dll“ />
</Adapters>
<!-- Empty filter bank shall result in logging of all events -->
<FilterBank>
<LogFilter event=“TAKE_ITEM“ pattern=““ action=“Log“ />
<LogFilter event=“LOG“ pattern=““ action=“Ignore“ />
<LogFilter event=“LOG“ pattern=“.*break.*“ action=“Break“ />
</FilterBank>
<Properties>
<item name=“AdapterType“ value=“BasicXmlFileAdapter“ />
<item name=“RepositoryPath“ value=“C:\MMLogs\AsRunLog“ />
<item name=“FilePrefix“ value=“AsRunLog“ />
<item name=“MaxDaysInRepository“ value=“60“ />
<item name=“MaxFileNumberOfEvents“ value=“10000“ />
<item name=“MaxFileTimePeriodHours“ value=“24“ />
</Properties>
</LogRepositoryConfig>
```

Tip: For customer specific as-run log files, see section Custom-built AsRunLog Adapters.

Configuration File Properties

All configuration files have the same set of properties:

- **AdapterType:** The default log adapter. Default: *BasicXmlFileAdapter*.
- **RepositoryPath:** The directory where the log files are stored. Default: *C:\MMLogs*, *C:\MMLogs\AsRunLog*, *C:\MMLogs\MosLog*.
- **FilePrefix:** Custom file prefix for the log file names. Default: *Log*, *AsRunLog*, *MosLog*.
- **MaxDaysInRepository:** Maximum number of days before the logfile is automatically deleted. Default: *60*.
- **MaxFileNumberOfEvents:** Maximum number of log events before a new logfile is created. Default: *10000*.

Copyright © 2024 Vizrt

- **MaxFileTimePeriodHours:** Maximum number of hours before a new logfile is created. Default: *24*.
- **UseLocalTime:** If set to *true*, all timestamps are converted from UTC to local time before being passed to the log adapter. Default: *False*.

Configuration of Log Adapters

A log stream may have one, or several, log adapters. These are configured in the *Adapters* section as shown in Example of an As-run Log above.
If the *Adapters* section is omitted, or empty, one single log adapter of the default type is created.

> ⚠ **Important note**: When initially configuring a new log adapter, after an installation or upgrade, you must complete all configurations before installing and starting the Mosart Server.
> If this sequence is not observed, the log adapter will not perform as expected.
> See Troubleshooting Log Adapters.

The default log adapter type is configurable in the *Properties* section, but is normally set to `BasicXmlFileAdapter`.

A single adapter has the following attributes:

- **type:** The adapter type.
- **name:** Name used for identifying the log activities associated with the adapter.
- **configuration:** Optional configuration file for the log adapter. If no configuration file is given the log adapter inherits the values from the current configuration file.
- **assembly:** Optional .Net assembly (.dll) containing the log adapter. This enables dynamic log adapters, loaded at run-time.

Configuration of Log Filters

A log adapter may have a set of filters that remove log events before they are handled by the adapter.

If no filters are defined, or the filter bank is empty, all events pass to the event handling log adapter.

Each filter has three attributes:

- **event:** Which event the filter applies to. Events are listed in sections Technical Log and AsRun Log. Setting this attribute to “DEFAULT“ changes the settings for the default filter bank. The default filter is used for all events that do not have a designated filter bank.
- **pattern:** Optional regular expression pattern that is applied to the log event value. The filter only returns the corresponding action if the pattern returns a match. If no pattern is given, the corresponding action is returned for all events.
- **action:** The action to return. The following actions are available:
  - **Ignore:** The event is ignored.
  - **Log:** The event is logged (forwarded to the log adapter).
  - **Break:** The event is treated as a break event.

Copyright © 2024 Vizrt

Example of Filtering a Technical Log Stream

```
<!- Empty filter bank, passes all log events -->
<FilterBank />
<!- Only passes events of type “MosConnection“ -->
<FilterBank>
<LogFilter event=“DEFAULT“ action=“Ignore“ />
<LogFilter event=“MosConnection“ action=“Log“ />
</FilterBank>
<!- Passes all log events, but treats “InstantCollect“ events and all Events
containing the word “break“ as a break event -->
<FilterBank>
<LogFilter event=“InstantCollect“ pattern=“” action=“Break“ />
<LogFilter event=“DEFAULT“ pattern=“.*break.*“ action=“Break“ />
</FilterBank>
<!- Passes all log events, but ignores all GUI events -->
<FilterBank>
<LogFilter event=“Gui“ pattern=“” action=“Ignore“ />
</FilterBank>
```

Customizing an As-run Log Adapter

Customizing is performed on a site-specific basis, by Vizrt staff. Some samples are presented below for reference.
Customization is based on the standard *AsRunLogRepositoryConfig.xml* as presented in Example of an As-run Log, and then include two more configuration files that can contain elements that override this standard configuration file.

- *xxxAsRunLogConfig.xml*, (Overrides AsRunlogRepositoryConfig)
- *xxxSchedulerConfig.xml*, (Overrides xxxAsRunLogConfig)

> ⚠ **Note:** xxx is here substituted for the actual customer’s name.

Customization Example: *xxxAsRunLogConfig.xml*

```
<?xml version=”1.0“ encoding=“utf-8“ ?>
<LogRepositoryConfig>
<FilterBank>
<!-- Triggers all “NEXT_BREAK“ events as breaks... -->
<LogFilter event=“NEXT_BREAK“ pattern=“” action=“Break“ />
<LogFilter event=“STORY_STARTED“ pattern=“BREAK“ action=“Break“ />
<LogFilter event=“STORY_ENDED“ pattern=“BREAK“ action=“Break“ />
</FilterBank>
<Properties>
<!-- Special configuration for the xxx log adapter -->
<item name=“SchedulerConfig“ value=“xxxSchedulerConfig.xml“ />
```

Copyright © 2024 Vizrt

```
<!-- These values may be overridden in SchedulerConfig -->
<item name="RepositoryPath" value="C:\MMLOGS\AsRunLog\xxx" />
<item name="FilePrefix" value="UPD" />
<!-- Set to true to use local time codes instead of UTC -->
<item name="UseLocalTime" value="True" />
<!-- Clean up command for files. Set to 0 if no cleanup -->
<item name="MaxDaysInRepository" value="0" />
</Properties>
</LogRepositoryConfig>
```

Customization Example: *xxxSchedulerConfig.xml*

```
<?xml version="1.0" encoding="utf-8" ?>
<SchedulerConfig>
<ShowPrefix>xxx</ShowPrefix>
<FilePattern>{Name}_{Prefix}_{Date}_{DayOfWeek}_{Intime}_{Rundown}_{StoryTitle}
_{StoryId}_{StoryIndex}</FilePattern>
<Directory>C:\MMLOGS\AsRunLog\xxx</Directory>
<Extension>xml</Extension>
<DayOfWeek>Dim,Lun,Mar,Mer,Jeu,Ven,Sam</DayOfWeek>
<!--<DayOfWeek>Dimanche,Lundi,Mardi,Mercredi,Jeudi,Vendredi,Samedi</DayOfWeek> -->
<NoLoggingDuringRehearsal>true</NoLoggingDuringRehearsal>  <!-- If false, logging are
permitted during rehearsal -->
<UseAbsoluteTime>true</UseAbsoluteTime> <!-- If false, time codes are relative to
show start -->
<MinLogDelay>-1</MinLogDelay> <!-- Specifies a minimum logging delay after configured
show start (in seconds) -->
<OneFilePerStory>true</OneFilePerStory> <!-- True if a file is to be generated per
story -->
<FlushAtStoryEnd>false</FlushAtStoryEnd> <!-- True if file should be flushed when
story ends. Default when show ends. -->
<IgnoreEmptyShows>true</IgnoreEmptyShows> <!-- True if file should be flushed when
story ends. Default when show ends. -->
</SchedulerConfig>
```

11.1.6 Starting-up a Log Adapter

After modifying or creating your log adapter as described in the sections above, you must ensure that Mosart Logging Service is running on the machine where the Viz Mosart UI and Viz Mosart server are installed.

There are two alternatives:

1. Re-install (UI) Mosart Logging Service or
2. Stop the (UI) Mosart Logging Service and start the Mosart Logging Service as console, or alternatively, configure to start it as as service.

Copyright © 2024 Vizrt

11.1.7 Troubleshooting Log Adapters

Issues with log adapters usually appear in the Viz Mosart UI with errors/warnings mentioning the service *Mosart Server Log as unavailable*.

Log adapter configuration will usually be made after an initial installation or upgrade that is introducing a requirement for a new/changed log adapter.

- It is essential to wait with installation of the Mosart Server (from the installer bundle) till the log adapter configuration is complete, and changed files saved in the `C:\ChannelTemplates` folder.
  If you did not observe this installation sequence, simply uninstall and re-install Mosart Server.

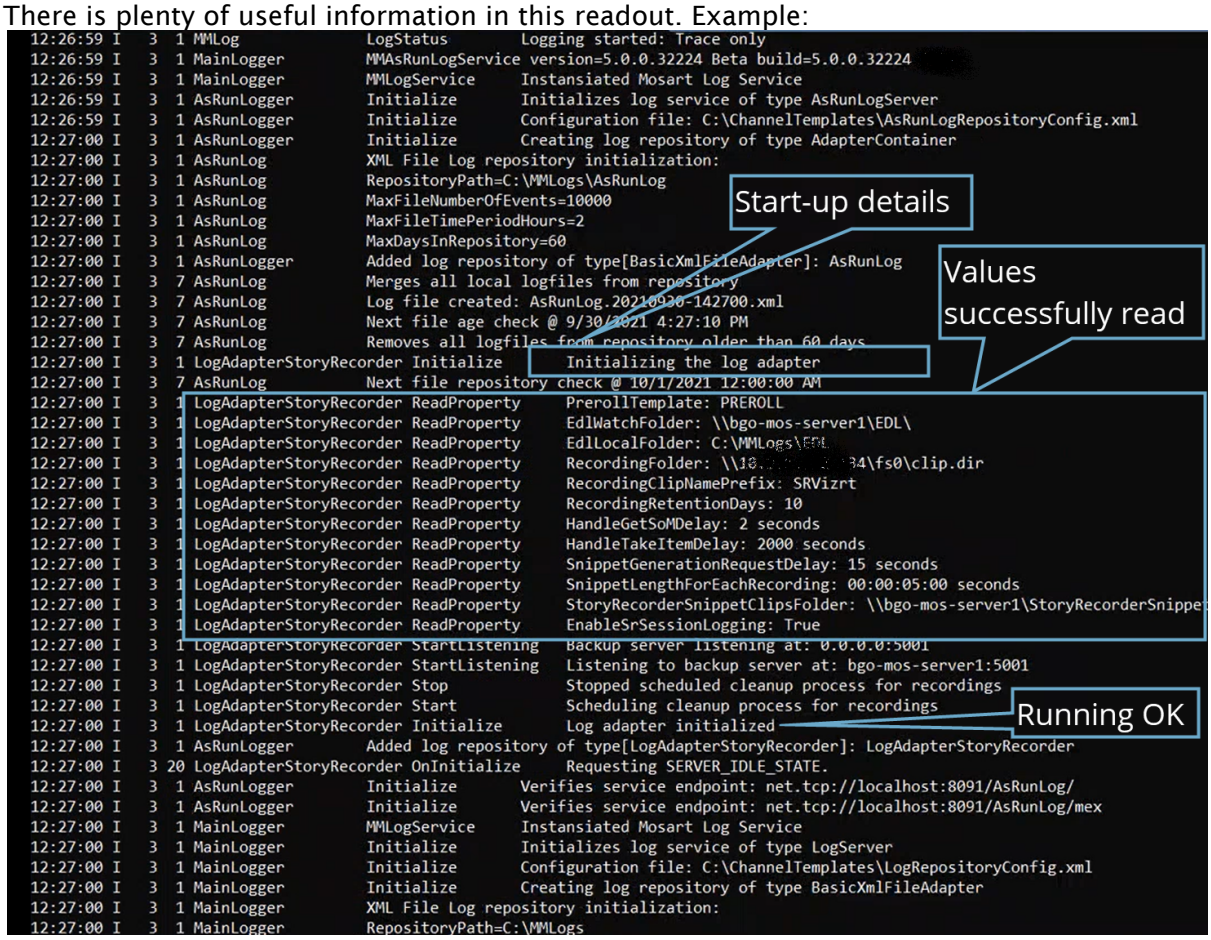
Verifying Log Adapter Performance

If you have unexpected logging service events, you can check the status of a log adapter.

1. Stop *Mosart Logging Service*.
2. Ensure that the affected logging configuration files, that initially reside in the `Program Files (x86)\Mosart Medialab\Mosart Server\` folder
   - *AsRunLogRepositoryConfig.xml*
   - *LogAdapterxxxConfig.xml* (here *xxx* is the naming of your new log adapter)
   already exist in *either*:
   - under `%ProgramData%\Mosart Medialab` folder
   or
   - `C:\ChannelTemplates` folder
   (but *never* in both places! Remove the oldest).
3. Ensure that in `A`*sRunLogRepositoryConfig.xml,* the section corresponding to your log adapter (for example `<!-- Adapter for Viz One -- ...`)
   is *uncommented* around the config parameters.
4. Save the file if you had to make changes.
5. Open your log adapter config file (*LogAdapterxxxConfig.xml*) and ensure the configurations match your system and save, if you made any changes.
6. In `Program Files (x86)\Mosart Medialab\Mosart Server\`, start the Mosart as-run service, `MMAsRunLogService.exe` as a *console*.
7. In the console, check that your log adapter initialized, and that it has successfully read the settings from the configuration file.

Copyright © 2024 Vizrt

There is plenty of useful information in this readout. Example:



8. After this verification step, *close* the console.
9. Start `MMAsRunLogService` as a service.

11.2 Server Maintenance

This section describes various maintenance aspects of Viz Mosart, and is divided into the following:

- Server File Structure
  - Static Configurations
  - User Configurations
  - Viz Mosart Applications and Services
  - Manus Administrator Repository
  - Log Directory
- Files for Backup
- File Purging

11.2.1 Server File Structure

The Viz Mosart Server stores files in different locations on the workstation as indicated by the list below:

Location	Definition
C:\Program Files (x86)\Mosart Medialab	Mosart applications. Various sub-directories for components such as Mosart GUI, Mosart Server and more. Either *\Program Files* or *\Program Files (x86)*.
C:\channelTemplates	XML-files defining the channel templates. Normally contains at a minimum *channeltemplates.xml*, *newsroomsettings.xml*, *AVConfig.xml*.
C:\manus	The Run-down (playlist) files fetched from the Newsroom systems.
C:\MMLogs	Default location for Log-files. Rotated.
C:\ProgramData\Mosart Medialab\ConfigurationFiles	System-wide configuration settings (XML).
C:\Users\<USERNAMEHERE>\AppData\Local\Mosart_Medialab	User-specific application configuration files.

Note: The list above is an example and may differ for various installations and Windows versions. See below for further details.

Copyright © 2024 Vizrt

Static Configurations

This is configuration that is static and common to all versions of Viz Mosart installed on the computer. Any upgrades do not overwrite any the content mentioned below but new configuration parameters are appended if present.

- **Location:** *C:\channeltemplates*.
- **Type:** Configuration.
- **Backup:** Yes.
- **For Support:** Yes.
- **Content:**
  - Video and audio setup.
  - Story templates.

User Configurations

This is configuration that is dependent upon the location and version of the application. Typically under the location there is a set of directories each referring to a particular installation of a Viz Mosart application. For example, for the Media Administrator:

- *%UserProfile%\Local Settings\Application Data\Mosart_Medialab\MMConsoleAdmin_2007.exe_Url_<*>\<version number>.* Any upgrades create a new subdirectory name with the corresponding version number and do a merge from the current version.
- **Location:** *%AppData%\Mosart_Medialab* or *%UserProfile%\Local Settings\Application Data\Mosart_Medialab* (i.e. for a particular non-roaming user: *C:\Documents and Settings\<User>\Local Settings\Application Data\Mosart_Medialab*)
- **Type:** Configuration.
- **Backup:** Yes.
- **For Support:** Yes.
- **Content:** Settings for all Viz Mosart Applications, for example the content of the various settings dialogs.

Viz Mosart Applications and Services

This is the location where the Viz Mosart installers place program files.

- **Location:** *%ProgramFiles(x86)%/Mosart Medialab* (i.e. for a default English language setup: *C:\Program Files(x86)\Mosart Medialab*).
- **Type:** Applications.
- **Backup:** Yes* - Most of the executables (.exe) files have corresponding configuration files. Normally the content in there is static and is not changed during an upgrade. Nevertheless, custom configuration is possible.
- **For Support:** No.
- **Content:** One or several subdirectories. One for each Viz Mosart installer.

Copyright © 2024 Vizrt

Manus Administrator Repository

The Manus Administrator repository contains all rundowns that have been loaded into Viz Mosart from the newsroom system. All rundowns are stored in an XML format, making it possible to extract rundown information to use for other purposes, for example video clip usage statistics.

- **Location:** *C:\manus*.
- **Type:** Rundowns.
- **Backup:** No* - Only for selected rundowns.
- **For Support:** Yes.
- **Content:** One or several subdirectories.

Log Directory

By default, log files are stored in *C:\MMLogs*.

- **Location:** *C:\MMLogs*
- **Type:** Log.
- **Backup:** No.
- **For Support:** Yes.
- **Content:** A set of log files from the various Viz Mosart Applications.

> ⚠ **Note:** For support purposes, collect all log files with timestamps that cover the period of time where an issue was discovered.

11.2.2 Files for Backup

All locations stated in Server File Structure should have a safety copy taken whenever changes are made to the active Viz Mosart Server. It is recommended to copy files and corresponding directories to a common share, outside Main and Backup Viz Mosart Servers, marked with Viz Mosart version and dated when the backup was performed.

11.2.3 File Purging

The Manus Administrator Repository and the Log Directory above, have built in mechanisms for purging files. The content of these directories grows unless a maintenance procedure is established.

For the Manus Administrator Repository, the retention period can be configured in the Manus Administrator by writing `settings` in the Manus Administrator console window, and then setting the value *ManusExpirationTime* to any number of days. The default value for *ManusExpirationTime* is 60 (days). If the manus files have an in-house maintenance procedure in place, use -1 and Manus Administrator will not delete any files. 0 is not a legal value. Only xml files auto-generated by Manus Admininstrator will be deleted.

The Log Directory system is preset to a 60 days retention period, but this can be changed, see System Logging.

Copyright © 2024 Vizrt

11.3 General Advice On System Operations

This section contains the following topics:

- Rebooting and Restarting - General Notes
- Rebooting Machines
- Application Restart
- Viz Mosart Application Updates
- Windows Updates
- Viz Mosart Main/Backup Server Testing

11.3.1 Rebooting and Restarting – General Notes

All software system must be restarted occasionally. This applies to both the operating and Viz Mosart system.

11.3.2 Rebooting Machines

The actual reboot frequency cannot be accurately determined up front, but good practice dictates rebooting all Windows computers and servers at least once a month. Should any ill effects occur periodically even with this monthly reboot interval, it is advised to reduce the interval to once a week. Likewise, if there still are issues, the next step should be daily reboots. In this case, steps should be taken to determine the problem cause.

11.3.3 Application Restart

For restart of the Viz Mosart Applications, we have not firmly stated any interval, as this depends on the actual practice in the stations, the connected equipment and the total workflow. Just like for system reboots, good practice dictates restarting the Viz Mosart Applications at least once a month.

11.3.4 Viz Mosart Application Updates

Vizrt regularly provides feature and maintenance updates for Viz Mosart system applications. Please review the associated Release Notes for each release to help with assessing whether you should upgrade.
After an upgrade, we always recommend performing some test runs in between other transmission schedules, focused on your mission-critical workflows.

However, if your operation is running according to your expectations, and is not dependent on the published fixes of a specific update, or as long as you don't require the new functionality that the update introduces, or when your servers are running in network isolation, then as long as your system has a valid maintenance contract with Vizrt Support, there may not be a need to perform an upgrade of your Viz Mosart system.

Copyright © 2024 Vizrt

11.3.5 Windows Updates

The interval for running Windows updates depends on the environment. If the computers are accessible from other zones, both Windows and antivirus updates must be done just as any other machine in the station's network. Based on the possibility of antivirus runaway, it is not recommended to run real-time virus scanning during transmissions. 24/7 stations must give this possible issue thorough considerations.

> **AV Automation Slow Start**
>
> There can be an approximately 30 second delay when starting AV Automation. This delay is either caused by
> 1. The Viz Mosart SNMP service being enabled but with incorrect SNMP connection settings.
> 2. A pending Windows update, queued to run.
>   In this case, follow the appropriate Microsoft procedure to update Windows, using downloaded files (when you are not permitted to connect to the Internet).

11.3.6 Viz Mosart Main/Backup Server Testing

|Our recommended usage pattern is to alternate between main and backup every day, to make sure that the vital backup server is ready to take over the day you really need it. This way, you are continuously testing the backup function.

> **Note:** There can be an approximately 30 second delay when starting AV Automation. This delay is either caused by
> 1. The Viz Mosart SNMP service being enabled but with incorrect SNMP connection settings.
> 2. A pending Windows update, queued to run.
>   In this case, follow the appropriate Microsoft procedure to update Windows, using downloaded files (when you are not permitted to connect to the Internet).

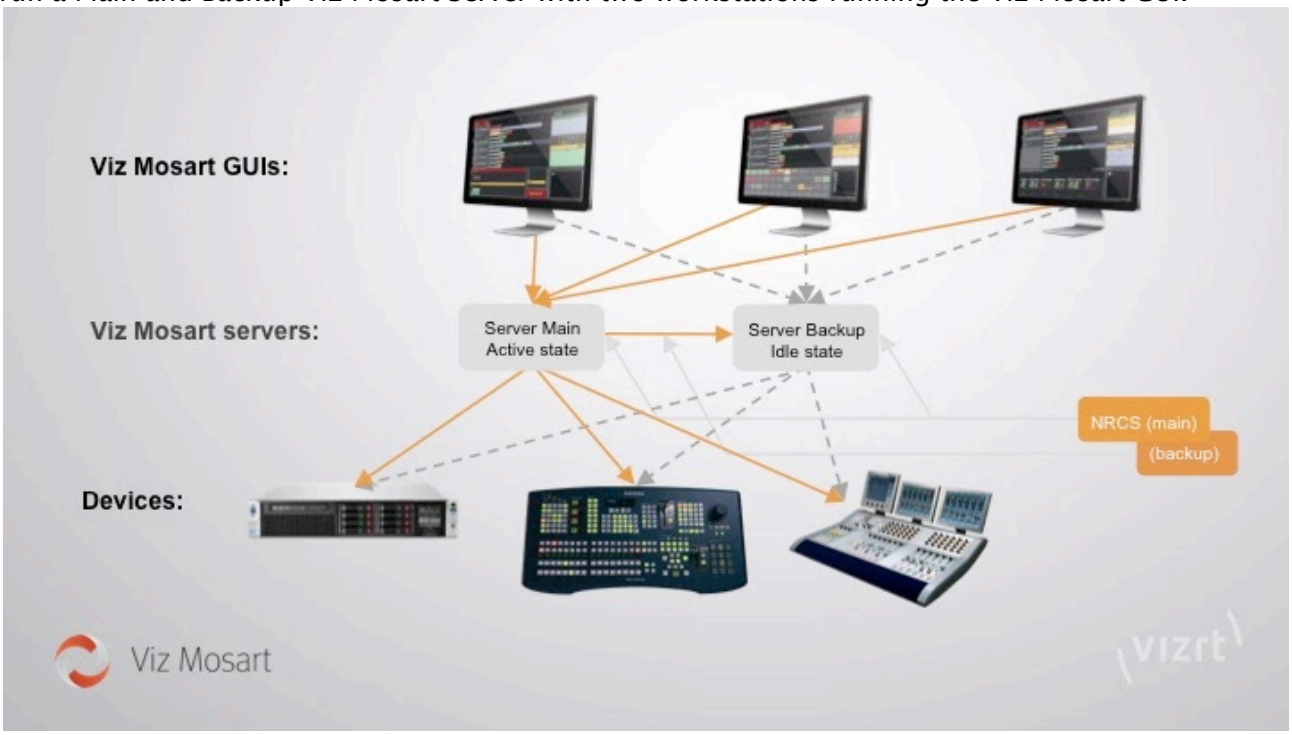
Copyright © 2024 Vizrt

11.4 Redundancy

As all Viz Mosart applications and services are interconnected using TCP/IP, it is possible to run all components on different computers, allowing multiple redundancy setup possibilities.

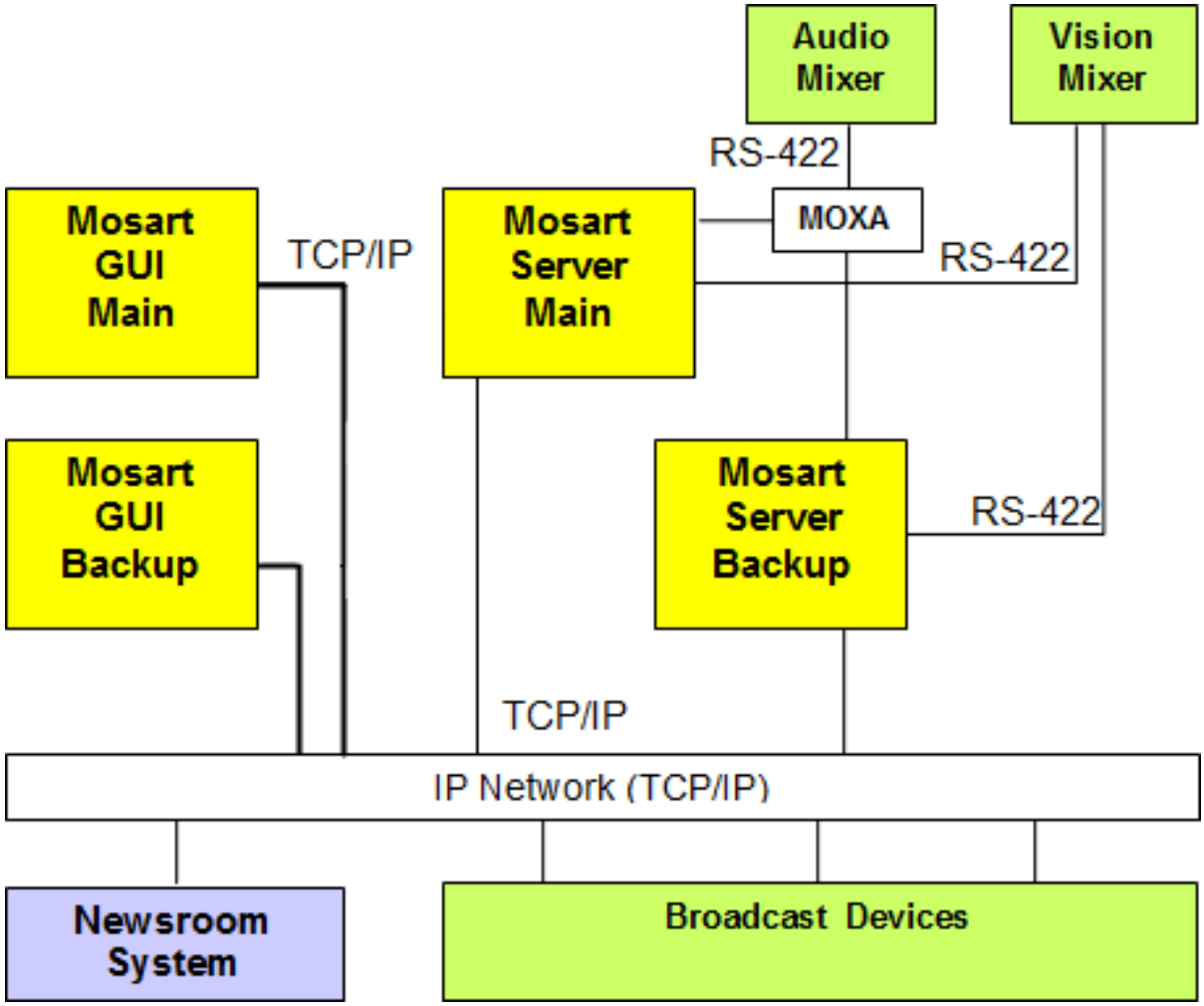
- Automation Server Mirroring
- Connected GUIs
- Graphical Interfaces
  - AV Automation - The Engineering Interface
  - Viz Mosart GUI (Operator's Interface)
- Device-specific Redundancy
  - Video server redundancy
  - Video Switcher Backup
  - Newsroom System Switch-over

Viz Mosart is based on a dual redundant server and a GUI PC configuration with dual redundant connections (Network Teaming) to broadcast devices. The recommended redundancy setup is to run a Main and Backup Viz Mosart Server with two workstations running the Viz Mosart GUI.



An example of a recommended redundancy setup is shown schematically below (the actual solution will depend on the characteristics of the connected devices):

Copyright © 2024 Vizrt



The system in the example above has the following characteristics:

- Dual redundant system.
- All Viz Mosart applications and services in the Viz Mosart Server suite are interconnected and running on the same server.
- Both main and backup Viz Mosart GUI applications are configured to both main and backup Viz Mosart servers. Initially they use the *main* Viz Mosart Server.
  Only one click is needed to instead connect them to the backup server.
- All external equipment and systems are shared between the main and backup servers.
- The vision mixer is connected using serial RS-422, with both main and standby cabling (as the mixer supports multiple serial connections).
- The audio mixer in this example has only one serial connection. A serial/ethernet converter (for example MOXA Terminal Server) has been used for enabling both servers to communicate with the mixer.
- All other equipment is connected via TCP/IP on a common network. IP redundancy is presumed on a network level through the use of Network Teaming.

Copyright © 2024 Vizrt

11.4.1 Automation Server Mirroring

The automation servers work in an active/idle setup, where the active server pushes the various states to the idle server. If the active server fails, the GUI operator can manually select the idle server, which then seamlessly takes over, becoming the active server.
In the case of a server failure, the servers operate independently of each other.

11.4.2 Connected GUIs

Viz Mosart supports an unlimited number of user interface clients. In the case of a server switch-over, all GUI clients follow this switch-over.

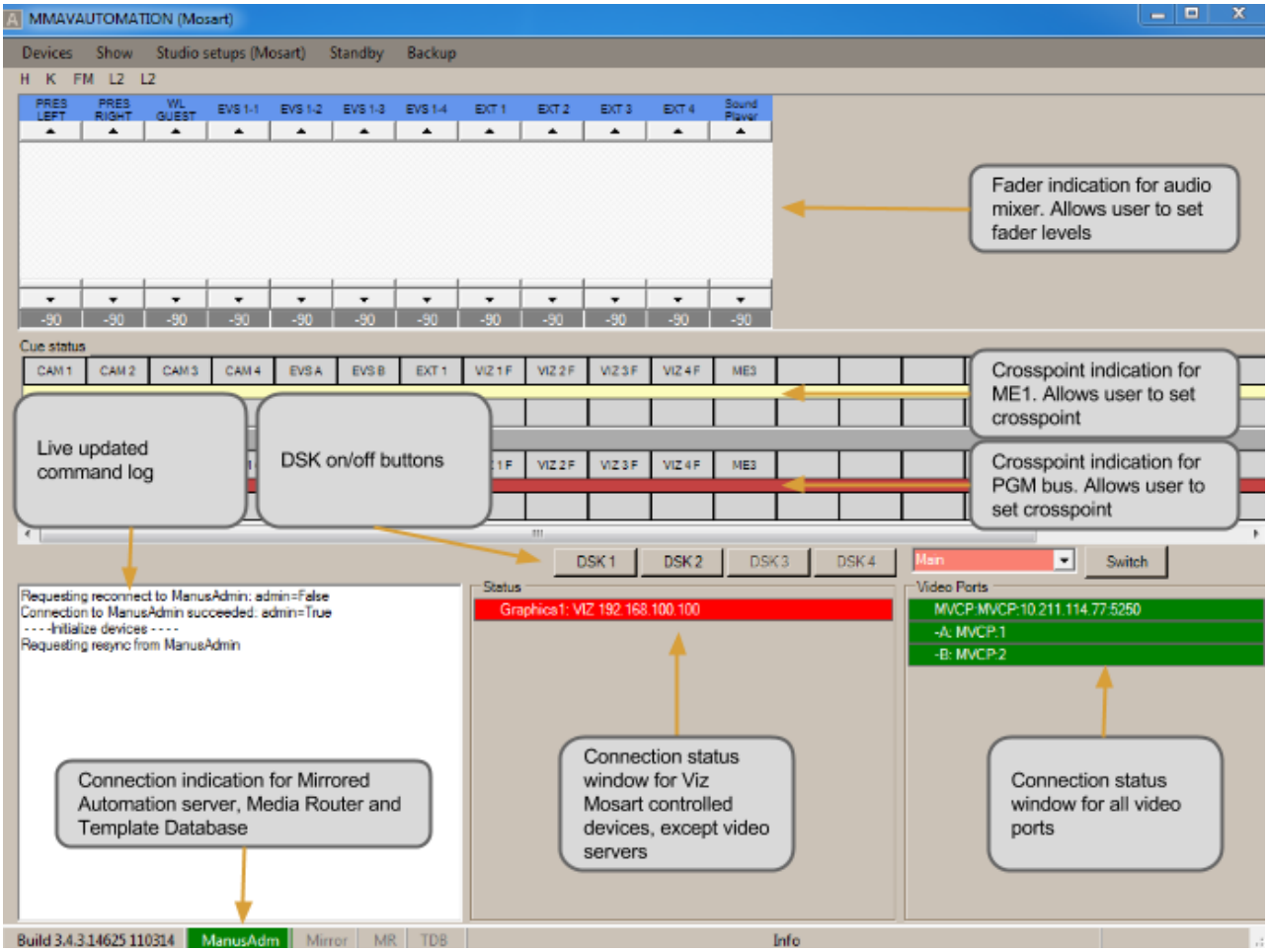
11.4.3 Graphical Interfaces

Viz Mosart has two main interfaces that expose redundancy features of the system to operators:

AV Automation - The Engineering Interface

AV Automation provides connection statuses of the connected devices and enables the user to access several functions connected to system redundancy. AV Automation runs on each of the two automation servers.

Copyright © 2024 Vizrt



Viz Mosart GUI (Operator's Interface)

While the GUI primarily is for on air operation, the *lower part* of the interface allows the operator to do the following:

- Switch between the automation servers (while the system is on air)
- Set video server salvos (for example main to backup salvo)
- Switch to the mirrored video servers seamlessly
- Switch to backup graphics controller.

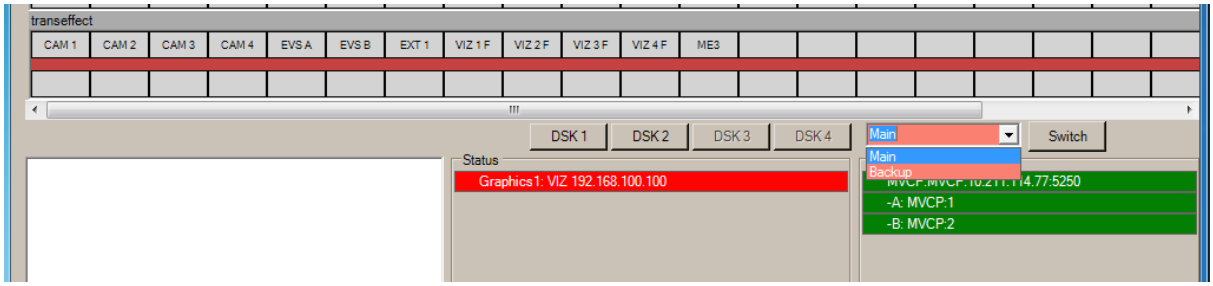
Copyright © 2024 Vizrt

11.4.4 Device-specific Redundancy

Video server redundancy

Viz Mosart supports connection to unlimited amounts of video servers, configurable with the following possibilities:

- Multiple A/B ripple groups
- Mirrored video playout in any group
- Salvo setups for quick recalls of various video server control setups.
  Video server salvo switch-over is illustrated below.



Video Switcher Backup

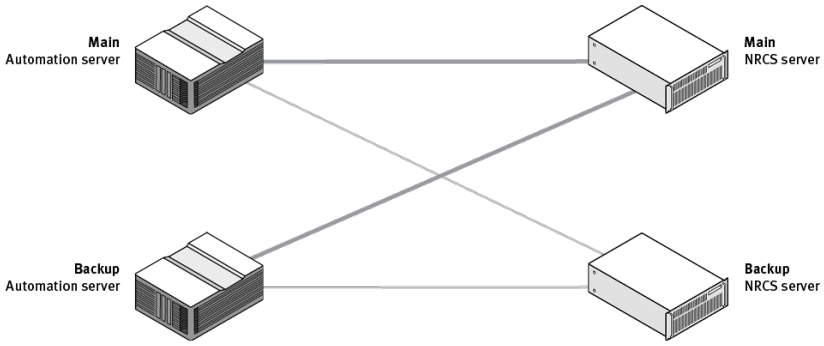
Viz Mosart supports router control which can be used as an emergency video switcher. The system allows the broadcaster to set crosspoints on the video router simultaneously with the video switcher. This can be used as an emergency output at video switcher breakdown.

Copyright © 2024 Vizrt

Newsroom System Switch-over

Viz Mosart supports dual NRCS servers and will follow the backup switch-over mechanism of the newsroom system.

Viz Mosart



Copyright © 2024 Vizrt

12 ActiveX

Note:

ActiveX is superseded by the Mosart Web Apps NRCS Plugin

Rather than using the legacy ActiveX feature, journalists can now use the Mosart Web Apps NRCS Plugin to insert and customize Viz Mosart template information into their NRCS script.

- From within their NRCS, the journalist searches for a Viz Mosart template, customize the content, then drag and drop it into their script.
- When the script is saved, information is pushed to Manus Administrator creating a corresponding entry in the active rundown in the Viz Mosart GUI.

The Mosart Web Apps NRCS Plugin must be connected to your Viz Mosart Server. Connection is handled in a simple configuration wizard at Web App installation, offering one of two connection choices:

1. The Template Database.
2. Directly to the Viz Mosart Server.

For more information, see the latest Mosart Web Applications Guide.

For details of the legacy ActiveX connection, please refer to pre-5.4 versions of this Viz Mosart Administrator Guide.

Copyright © 2024 Vizrt

13 Appendix

This section covers the following topics:

- Avid iNews Web Service for Status Feedback
- GPIO Sample XML Config
- Jupiter Systems PixelNet Video Wall Control
- Mosart Remote Control REST API
- Named Overlay Graphics
- Stagetec Driver Configuration
- Structure of MOS Objects Sent to NRCS

Copyright © 2024 Vizrt

13.1 Avid INews Web Service For Status Feedback

This section shows how to set up and use Viz Mosart with Avid iNews Web Services in order to get stories status updates in the NRCS (Newsroom Computer System). This section covers the following topics:

- Introduction
- Installation and Configuration
  - Supported Versions
  - Avid iNews Web Service installation
  - Checking the Current Version of iNews Web Service
  - Configuration
  - API Session Limits
  - Configurations in Viz Mosart Manus Administrator
  - Other Manus Administrator settings affecting NRCS status feedback
  - Configurations in AvidInewsWebServiceConfig.xml
- Operation
  - Status for Stories Containing More Than One Item
  - Status of an Off-air Story Containing Multiple Clips
  - Clip Duration Feedback to iNews Server
- Troubleshooting and Known Issues
- Useful Links

13.1.1 Introduction

Avid iNews Web Service can be configured with Viz Mosart connected to iNews through FTP. The main purpose is sending status for stories that contain clips (Online/Offline for clip availability, Cued, Play or Stop and duration, as provided by the video server), but the system can also be configured to send status for all types of events (Cameras, DVEs etc.).

13.1.2 Installation and Configuration

Supported Versions

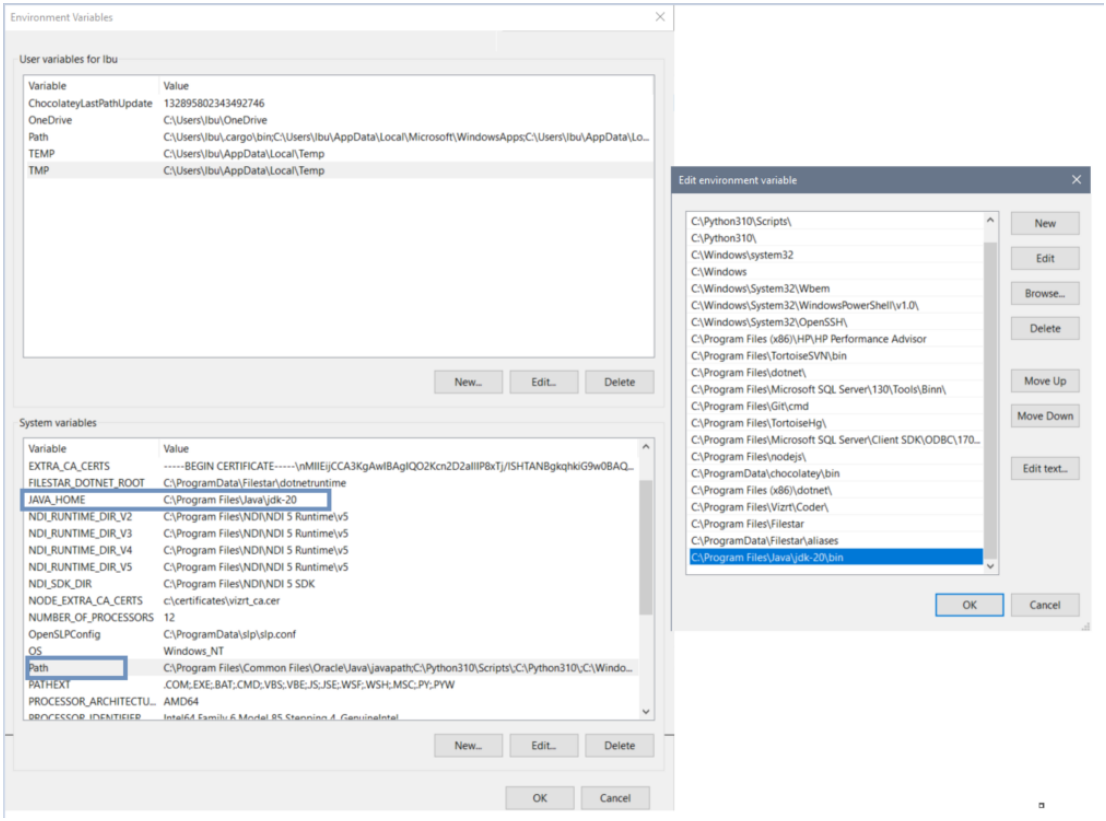
iNews WebService	Tomcat	JDK	iNews server
1.6.0.5	6.0.37	1.8	up to 2022.12
1.6.0.5	up to 9.0.79	20	up to 2022.12
1.6.3	up to 9.0.79	20	up to 2022.12

Copyright © 2024 Vizrt

Avid iNews Web Service installation

Refer also to the documentation that comes with Avid iNews Web Service installation package.

- Download and install the latest JRE or JDK, available from oracle.com (Windows offline (64-bit)).
- Run the executable JRE installation file (e.g. jdk-20_windows-x64_bin.exe).
- Set JAVA_HOME environment variable and the path to the bin folder in the PATH variable:



- Download and install the latest stable Tomcat version, available from tomcat.apache.org, 32bit/64-bit windows service installer (e.g. apache-tomcat-9.0.79.exe).
- Run the executable Tomcat installation file:
  a. Select “Normal” installation type.
  b. Create a Tomcat Administrator login user name and password (Do not call the user tomcat!).
  c. Select the JRE installation path if it is not already selected (e.g. C:\Program Files\Java\jdk-20).
- Allow C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin\Tomcat9.exe through the Windows Firewall.
- After the Tomcat installation is finished and has started the Tomcat management page should be available at http://<server>:8080/manager/html.
- To log in on the Tomcat Management page, user name and password were defined during Tomcat installation at step 5. b.

Copyright © 2024 Vizrt

- The default connector port can be modified in Tomcat's .\conf\server.xml file.
  If port 8080 is used by other applications, this can be changed in Tomcat Installation/config/server.xml file:

```
<Connector port="8080" protocol="HTTP/1.1"   connectionTimeout="20000"

redirectPort="8443" />
```

- Restart the Tomcat service after port changed from Services.msc or from Start/All Programs/Apache Tomcat/Monitor Tomcat.
- An icon is created in the Windows Taskbar.
  Right-click the icon for a list of actions to Start and Stop Tomcat service.
- Deploy iNews web service.war
- After deployment, the iNews Web Service should be available at http://<server>:<port>/inewswebservice/services.
- If you have problems accessing the Web Services page (http://<server>:<port>/inewswebservice/services), try deleting the Tomcat cache by following the four steps below:
  a. Stop the Tomcat service.
  b. Delete folder Tomcat installation\webapps\inewswebservice (only the folder, not the war file).
     You may replace the war file, for example, with a new one when upgrading. (This solution is same as deploying from Tomcat manager).
  c. Delete folder Tomcat installation\work\Catalina\localhost\inewswebservice
  d. Start the Tomcat service and try again to access the Web Services page.
- We recommend to reduce the session timeout in Apache Tomcat to *1 min*, freeing-up the API session 1 min after Manus Administrator was closed.
  If Manus Admin is closed and opened more than maximum API sessions allowed in 1 min, it will reconnect automatically to Web Service after 1 min when at least one session has been released. To change the session timeout interval do the following:
  a. Stop the Tomcat service.
  b. Change Tomcat session timeout interval in Tomcat installation/conf/web.xml to 1 min:
     <session-config>
     <session-timeout>1</session-timeout>
     </session-config>
  c. Start the Tomcat service.

⚠ **Note:** Config files for Tomcat are found here: C:\Program Files\Apache Software Foundation\Tomcat 8.5\conf.

Checking the Current Version of iNews Web Service

- Go to the Apache Tomcat installation folder /webapps/inewswebservice/META-INF and open Manifest.mf.
- Check the Implementation-version. This version number should be identical the version number specified in the **Note** above.

Copyright © 2024 Vizrt

C:\Program Files\Apache Software Foundation\Tomcat 8.5\webapps\inewswebservice\META-INF

- Recommended version is 1.6.0.5 or 1.6.3.
  It should look like this:

```
Manifest-Version: 1.0
Ant-Version: Apache Ant 1.8.4
Created-By: 1.7.0_25-b17 (Oracle Corporation)
Implementation-Build: MSN_INEWS_WEB_SERVICE_1.6.3.1
Implementation-Title: Avid iNEWS Web Service
Implementation-Version: 1.6.3.1
Implementation-Vendor: Avid Technology, Inc.
Copyright: Copyright (c) 2009-2022 Avid Technology, Inc.
```

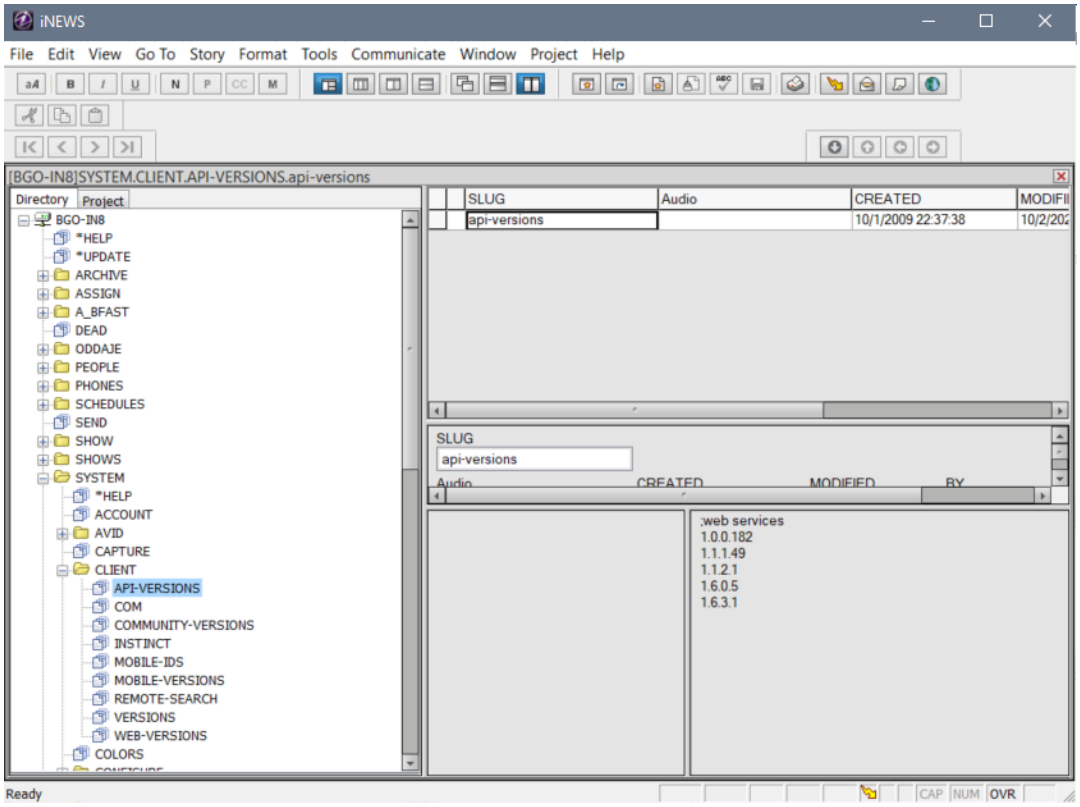
Configuration

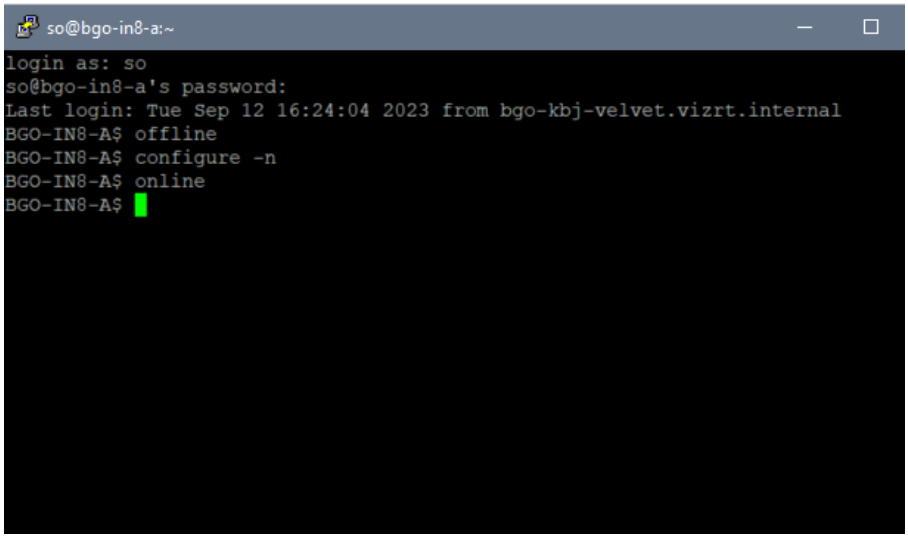
> **Note**: iNews can only display information at *story* level and therefore rules have been defined when a story contains more than one clip. However, for better usage of this functionality, it is recommended to have stories containing only one clip.

- Open iNews client and add the Avid iNews Web Service version in first story of rundown SYSTEM.CLIENT.API-VERSIONS.
  The version number should match the version of Avid iNews web service you are running (see section Checking the Current Version of iNews Web Service).
  For example:

  ```
  ;web services

  1.0.0.182

  1.1.1.49

  1.1.2.1

  1.6.0.5

  ; run offline; configure -n; online after doing changes in iNews server console
  ```

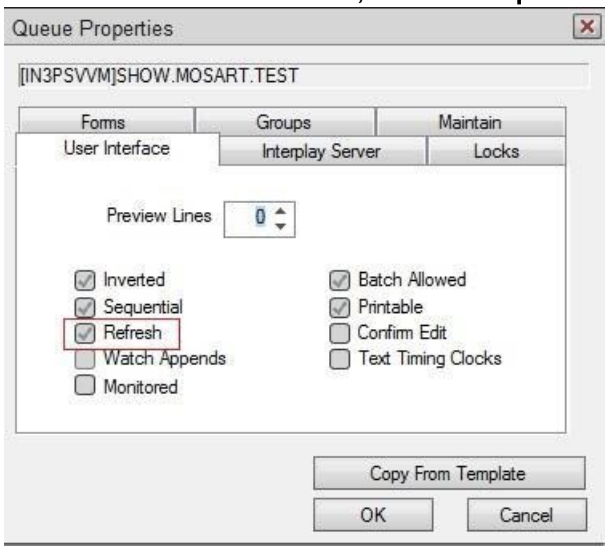
- On the iNews server console, execute the following commands one by one:
  offline, **configure -n** and **online**.





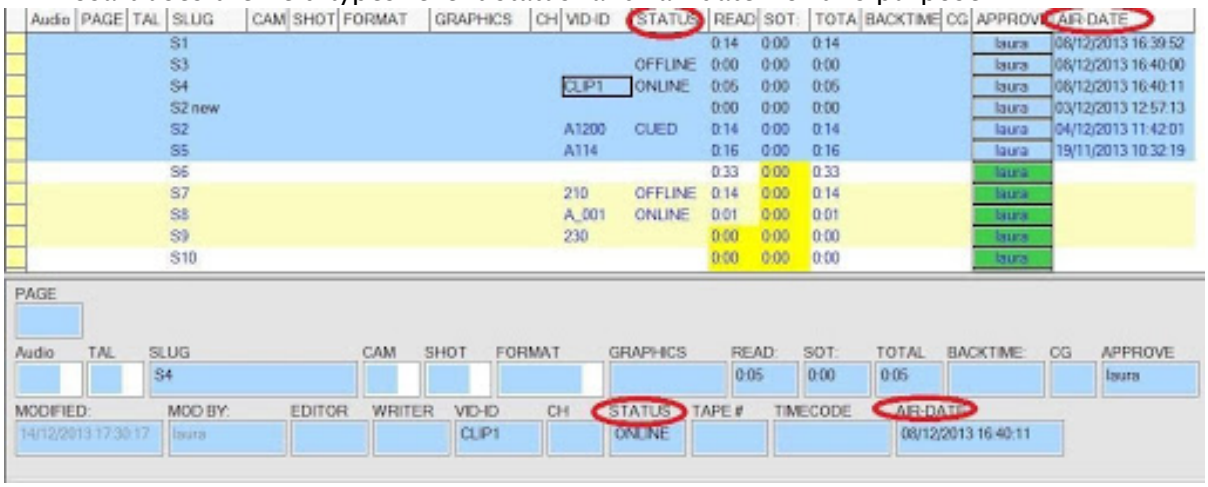
Copyright © 2024 Vizrt

- Enable **Refresh** for the rundown used for receiving status updates.
  Click on the rundown name, choose **Properties > User Interface**.



> **Note:** From iNews Setup and Configuration Guide: *Use the Refresh trait only on important queues, like rundown queues that are often modified by multiple users simultaneously. To automatically refresh a queue, your system must spend a lot of time monitoring workstations where users are viewing that queue.*
> *Assigning the refresh trait to too many queues that are often accessed at the same time greatly increases the amount of work your system has to do and may severely degrade its overall performance.*

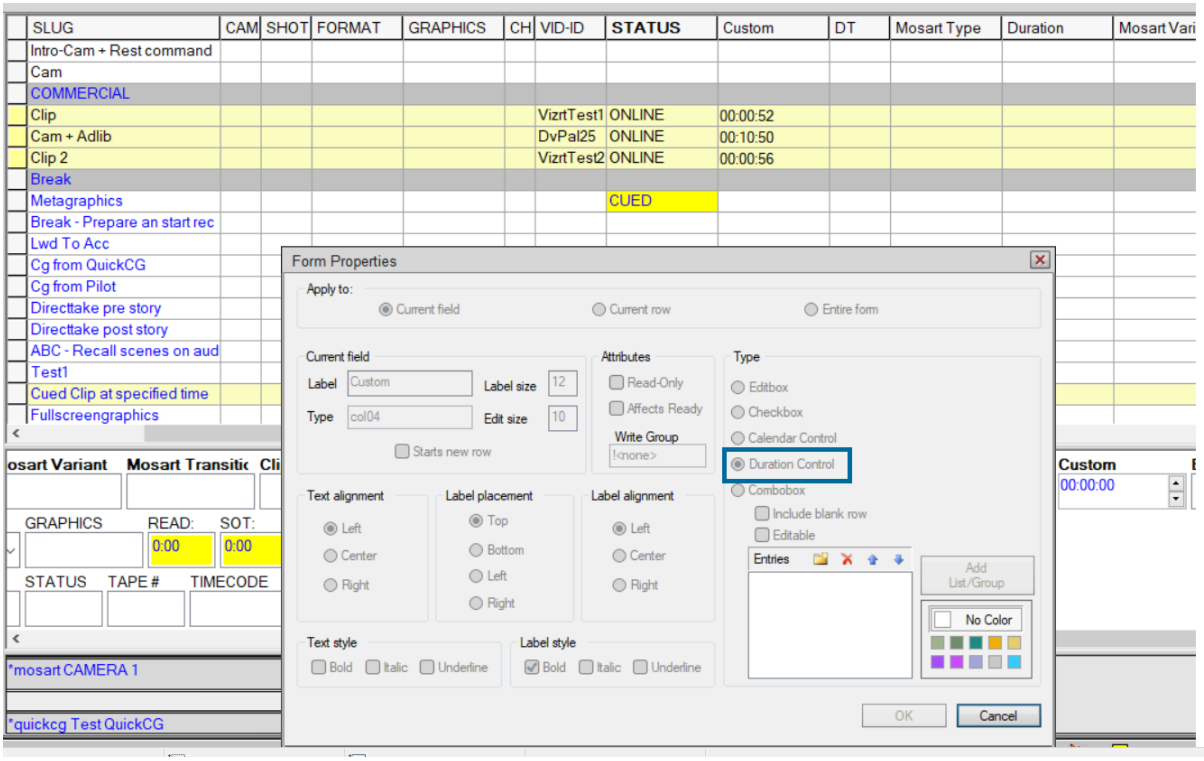
- Configure the columns to receive status and air-date updates from Viz Mosart. By default, Viz Mosart uses the field types “event-status” and “air-date“ for this purpose.



- To enable the feature that allows Viz Mosart to send clips duration, please configure a designated column in iNews to receive this information.
  If the column type is set to 'Duration,' then iNews will be responsible for controlling the timecode format displayed in that column.
  On the other hand, if the column type is 'Editbox,' Viz Mosart will send the clip duration in

Copyright © 2024 Vizrt

the timecode format specified within its own settings, calculated based on the frame rate configured in Viz Mosart.



API Session Limits

Maximum API-session connections to iNews Server

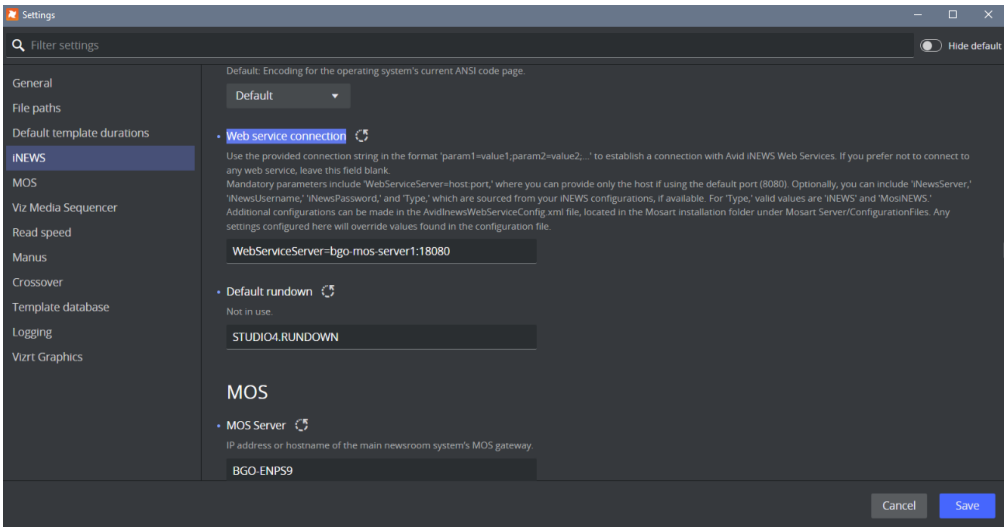
The default license for Avid iNews Web Service is for *five* concurrent API- sessions.
These are not the same as the iNews workstation licenses.

- Type "/exc/list type=api c" at the iNEWS server to see the number configured.
  Additional concurrent API-session licence is available from Avid.
- Viz Mosart will use a maximum of two API-sessions (in a redundancy setup).
- If you experience issues with no available API-session, please see section Troubleshooting and Known Issues.

Configurations in Viz Mosart Manus Administrator

From the Manus Admin Console, type *settings* and configure the **Web service connection** string:

Copyright © 2024 Vizrt



Leave the field blank if you do not wish to connect to any web service.
Use the provided connection string in the format param1=value1;param2=value2;...' to establish a connection with Avid iNEWS Web Services.

Required parameters are:

- **WebServiceServer =** host:port. Only the host can be provided if the default port (8080) is used.

Optional parameters are:

- **iNewsServer** Only one iNews server can be used by the WebService. Specify which iNews server the WebService should connect through. This field is optional if only one iNews server is set in the iNews Configuration.
- **iNewsUsername** Optional as it can be taken from iNews Configuration.
- **iNewsPassword** Optional as it can be taken from iNews Configuration.
- **Type** Valid values are 'iNEWS' and 'MosiNEWS

Additional configurations can be made in the AvidInewsWebServiceConfig.xml file, located in the Mosart installation folder under Mosart Server/ConfigurationFiles.

Other Manus Administrator settings affecting NRCS status feedback

- **Send Clip Status To NRCS** (default: false)

  Set to true to send clip status (Online/Offline) when Media Administrator is running.

- **Send only availability status for clips to NRCS** (default: false)

  When enabled, this setting will restrict the statuses sent to NRCS to only the availability statuses for clips, specifically 'Online' and 'Offline'. Playback statuses such as 'Cue', 'Play', and 'Stop' will not be sent.
  If the rundown is in *rehearsal mode*, Viz Mosart will only send availability statuses for clips, regardless of how this setting is configured.

- **Ignore Send Cue Status to NRCS for Offline Clips** (default: false)

Copyright © 2024 Vizrt

Select to prevent the NRCS receiving CUED or READY statuses of offline video clips for the first story set as next when the timeline is not running.

- **Ignore Updates If No Changes** (default: false)

  If enabled, stories with no relevant changes will not be processed by Viz Mosart. This is to avoid issues with story updates from GUI to be ignored. Later changes in Viz Mosart iNews Web Services API prevented such problems, but for future changes is good to keep in mind this setting together with **StoryCompareIgnoreAttributes** which is explained below.

- **Story Compare Ignore Attributes** (default: empty)
  List of xml attributes to ignore during comparison. Internally, the default list of attributes is set to [date_0, date_1, timestamp] and what is configured in this setting will be added to this list.

  When Viz Mosart receives a story in NSML format, it will parse the story and create its own XML format. When irrelevant updates come from NRCS, some attributes in the Viz Mosart XML story are updated. Currently these are date_0, date_1 and timestamp (see the format of a Viz Mosart XML story in any xml file in C:\manus folder). Attribute date_0 was recently added to the default attributes to be ignored due to the problems with updates caused by Viz Mosart iNews Web Services.
  In future if other attributes cause such issues, than they can be added to the list of attributes to be ignored.

- **Refresh Media On NRCS Update** (default: false)

  If enabled, it forces updating clips when story is changed in NRCS. This is an old setting introduced before Viz Mosart iNews Web Services.

- **Keep Story Status On NRCS Update** (default: false)
  If enabled, will keep the story status on an NRCS update, thus keeping the on-air status. The content of the story will be updated, but the story will be gray in the Mosart GUI. This is also an old setting introduced before Viz Mosart iNews Web Services.

Configurations in AvidInewsWebServiceConfig.xml

Copy the configuration file from in *Mosart Installation folder\Mosart Server\ConfigurationFiles* to the *ChannelTemplates* folder on C:\.

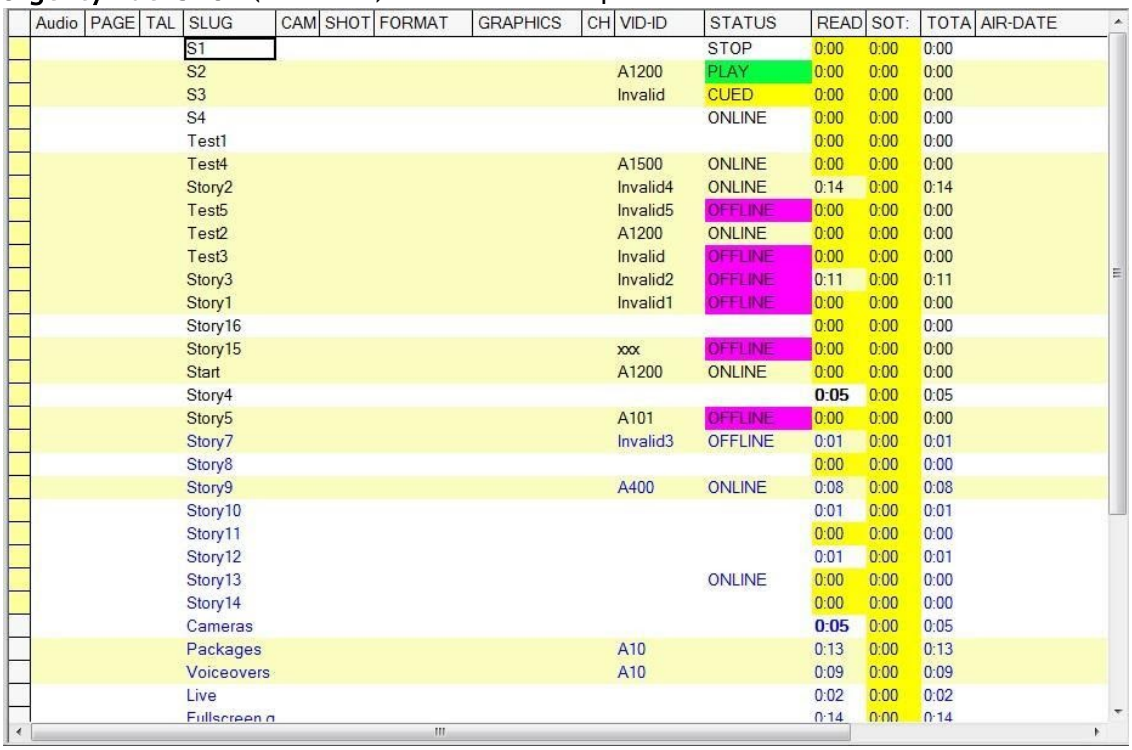
- **SendUpdateStatusForAllItems** (default: false)
  By default, only status of items containing clips will be sent (Online/Offline/Cued/Play/Stop). Not only clips given through Package or Voiceover are taken into consideration, but also Adlibpix or other ways to insert clip into a story. So for example if a story contains a Camera and an Adlibpix, than the story gets status.
- **StatusColId** (default: event-status) The id of the column to display the story item status sent by Viz Mosart.
- **AirDateColId** (default: air-date) The id of the column to display the air-date sent by Viz Mosart.
- **ClipDurationColId** (default: empty) The id of the column to display the clip duration sent by Viz Mosart.

Copyright © 2024 Vizrt

- **ClipDurTcFormat** This setting determines the timecode format used for displaying clip durations in the column specified by "ClipDurationColId". If the iNews column is not of type 'duration', the default format is 'mm:ss'. Other available formats include: 'ff', 'ss', 'hh:mm:ss:ff', 'mm:ss:ff', and 'ss:ff'. When the iNews column is of type 'duration', Mosart will transmit the duration in seconds. Conversion from frames to the selected timecode format is carried out based on the frame rate configured in the Manus Administrator settings.
- **ClearStatusWhenReload** (default: false)
  If enabled, only status column is cleared on reload of the rundown. This is to avoid inconsistent status if, for example, stories failed to be updated because of a crash of Viz Mosart or iNews or Viz Mosart is closed while rundown is playing etc.
- **ClearStatusWhenUnload** (default: false)
  If enabled, the status will be cleared when rundown is unloaded.
- **SetStoryAirDate** (default: true)
  Set to false to disable setting the AIR-DATE field and moving the "aired line" (the blue line). Note that the color of the line can be changed from Tools/Options/Colors in the iNews client and choose “Aired” from drop-down box. By default this is blue.
- **ClearAirDateWhenReload** (default: false)
  If enabled, the air-date field will be cleared during reload of the rundown (empty status). Note that this setting is taken into consideration only if **SetStoryAirDate** is true.
- **ClearAirDateWhenUnload** (default: false)
  If enabled, the air-date field will be cleared during unload of the rundown (empty status). Note that this setting is taken into consideration only if **SetStoryAirDate** is true.
- **Color index attributes**
  The following parameters are colour index referring to values defined in SYSTEM.COLORS on the iNews system.
  The AIR-DATE field needs to be empty in order to see the correct color for the STATUS field:
  - **UrgencyIndex** (default 1)
  - **UrgencyIndexONLINE** (default 1) Set color for Online status.
  - **UrgencyIndexOFFLINE** (default 4 - purple) Set color for Offline status.
  - **UrgencyIndexCUED** (default 2 - green) Set color for Cued status.
  - **UrgencyIndexPLAY** (default 3 - yellow) Set color for Play status.

Copyright © 2024 Vizrt

- **UrgencyIndexSTOP** (default 1) Set color for Stop status.



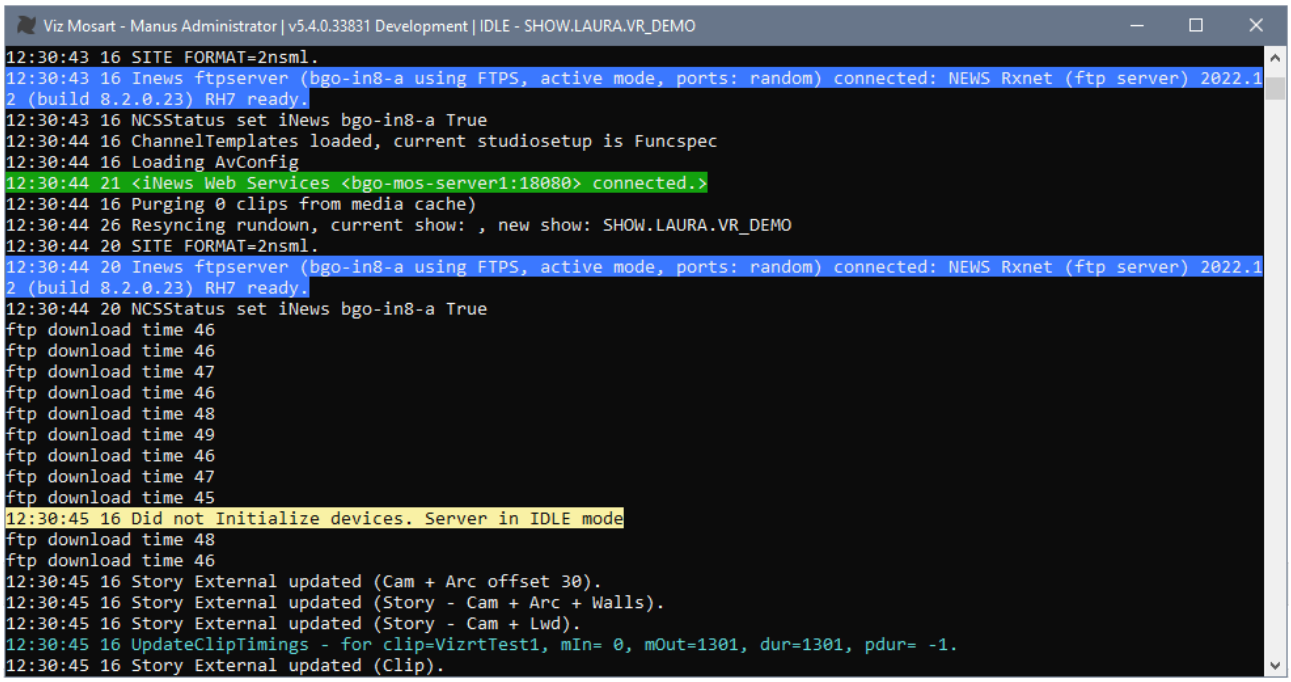
Audio	PAGE	TAL	SLUG	CAM	SHOT	FORMAT	GRAPHICS	CH	VID-ID	STATUS	READ	SOT:	TOTA	AIR-DATE
			S1							STOP	0:00	0:00	0:00	
			S2						A1200	PLAY	0:00	0:00	0:00	
			S3						Invalid	CUED	0:00	0:00	0:00	
			S4							ONLINE	0:00	0:00	0:00	
			Test1								0:00	0:00	0:00	
			Test4						A1500	ONLINE	0:00	0:00	0:00	
			Story2						Invalid4	ONLINE	0:14	0:00	0:14	
			Test5						Invalid5	OFFLINE	0:00	0:00	0:00	
			Test2						A1200	ONLINE	0:00	0:00	0:00	
			Test3						Invalid	OFFLINE	0:00	0:00	0:00	
			Story3						Invalid2	OFFLINE	0:11	0:00	0:11	
			Story1						Invalid1	OFFLINE	0:00	0:00	0:00	
			Story16								0:00	0:00	0:00	
			Story15						xxx	OFFLINE	0:00	0:00	0:00	
			Start						A1200	ONLINE	0:00	0:00	0:00	
			Story4								**0:05**	0:00	0:05	
			Story5						A101	OFFLINE	0:00	0:00	0:00	
			Story7						Invalid3	OFFLINE	0:01	0:00	0:01	
			Story8								0:00	0:00	0:00	
			Story9						A400	ONLINE	0:08	0:00	0:08	
			Story10								0:01	0:00	0:01	
			Story11								0:00	0:00	0:00	
			Story12								0:01	0:00	0:01	
			Story13							ONLINE	0:00	0:00	0:00	
			Story14								0:00	0:00	0:00	
			Cameras								**0:05**	0:00	0:05	
			Packages						A10		0:13	0:00	0:13	
			Voiceovers						A10		0:09	0:00	0:09	
			Live								0:02	0:00	0:02	
			Fullscreen								0:14	0:00	0:14	

- **LockFormSectionOnly** (default: true)
  If true will issue a lock on the FORM section of a story, if false will lock the complete story.
- **HeartbeatInterval** (default: 3 seconds)
  Interval to check Avid Web Service heartbeat.
- **UpdateStoriesCacheInterval** (default: 1 second)
- **RetryStoryUpdateStatusInterval** (default: 5 seconds)
- When a story failed to be updated (either because the story is locked for editing, the WebService is down or the iNews server is down), an error message will be displayed in console.
  The retrial to update the story status will continue until the problem is fixed or Manus Admin is closed.
- **NoOfStoriesToGet** (default: 50) – number of stories to get from iNews for caching

13.1.3 Operation

When the connection to the Avid Web Service is successful the following message will appear in Manus Admin console: <iNews Web Services <host:port> connected>.
The connection will be made after loading a rundown in the GUI.

Copyright © 2024 Vizrt



The following statuses will be shown in iNews:
Cued, Play, Stop, Online and Offline.

> ⚠ **Note:** Offline status also will be assigned for a clip that is to be cued but will fail cuing.

Users can move stories in iNews up or down. If rundown is on-air and user has been changed a position of On-Air or Next stories statuses of clips will be changed also according its statuses.

> ⚠ **Note:** If an iNews story with status OFFLINE due to "cue failed" is moved in the iNews playlist (even if it will be the Next story), it will be still OFFLINE and the status should stay as OFFLINE. Check iNews server settings (MCS dictionary located at /site/dict/mcs) to translate the statuses shown in iNews client to local statuses.

> ⚠ **Note:** No status is sent for overlays.

Operational Status

The table below shows examples of status behavior for setting **SendUpdateStatusForAllItems**.

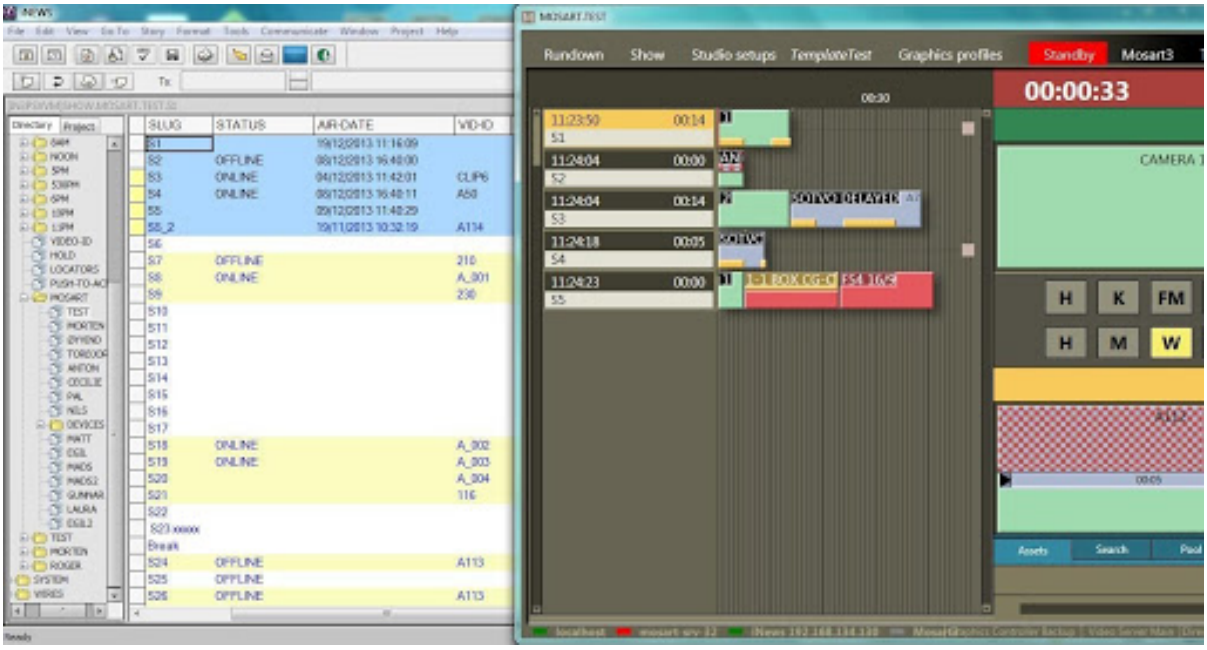
- If **SendUpdateStatusForAllItems** is *false*, the entire column "*Story without video clips*" will be cleared (no status sent for these stories).
- **ClearStatusWhenReload** should be set to *true* to clean up old history.
- The last row (*Story has been On air*) means that after the story has been run but the rundown is still on-air, the clip is no longer monitored.
  If the clip subsequently becomes available or unavailable, this is NOT be reflected in iNEWS.
  Only after the rundown is reloaded, is clip status be updated.

Copyright © 2024 Vizrt

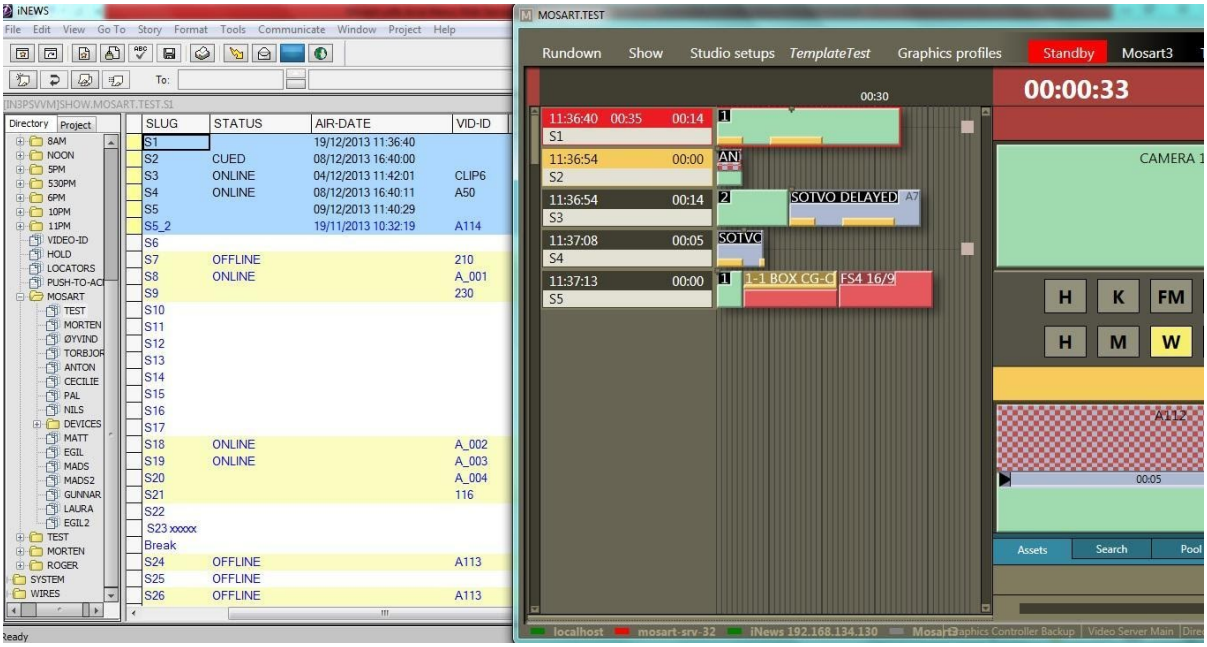
Time sequence	Story with video clips	Story without video clips (**SendUpdateStatusForAllItems** = *true*)	Story without video clips (**SendUpdateStatusForAllItems** = *false*)
Before On air	ONLINE/OFFLINE		
Next story	CUED	CUED	
Story On air	PLAY	PLAY	
Story just taken Off air	STOP	STOP	
Story has been On air	STOP	STOP	

Air-date is also updated to the NRCS if the parameter **SetStoryAirDate** is set to *true* in the AvidInewsWebServiceConfig.xml file.
When the parameter for **SendUpdateStatusForAllItems** is false, only **air-date** for stories that do not contain any clips will be updated.

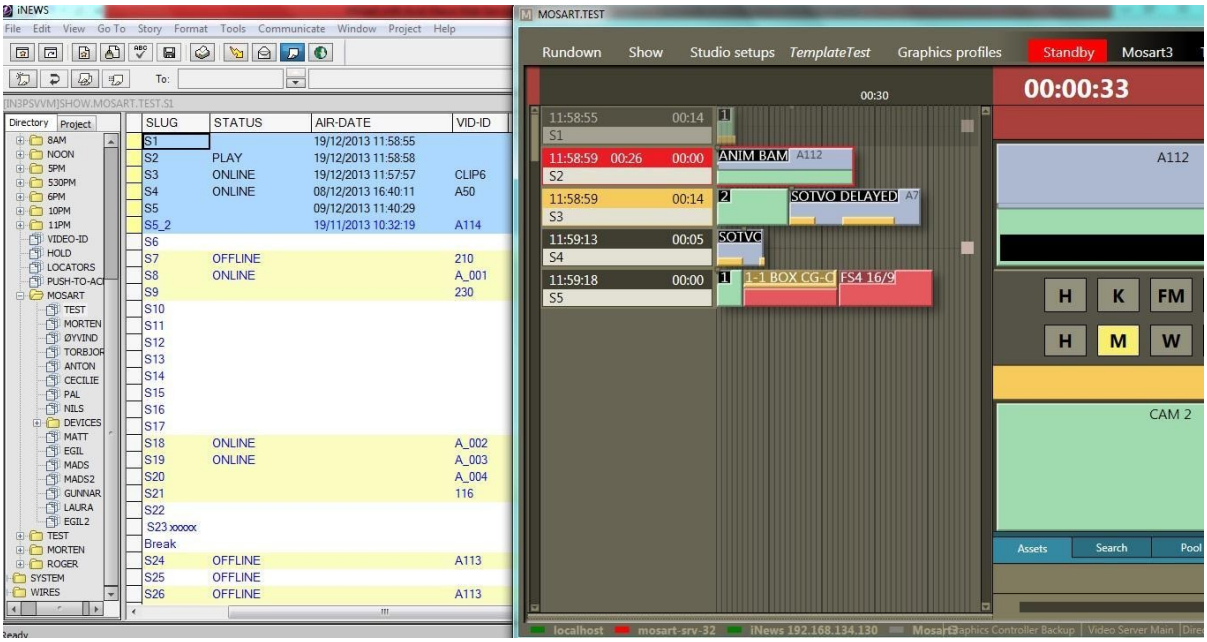
Below are three examples of status behavior:



Above, the rundown is off-air. Stories S2, S3 and S4 were updated with status of the clip it contains.

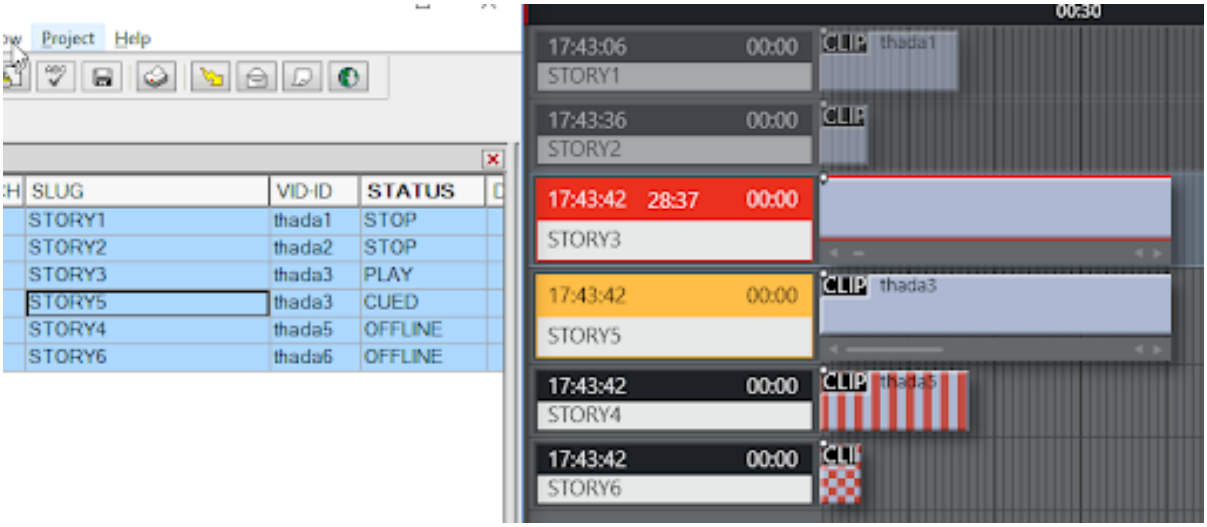


Above, the rundown is on-air. Story S1 is currently playing, but the status was not sent to the NRCS. Since S2 contains a clip, its status is set to “CUED”.

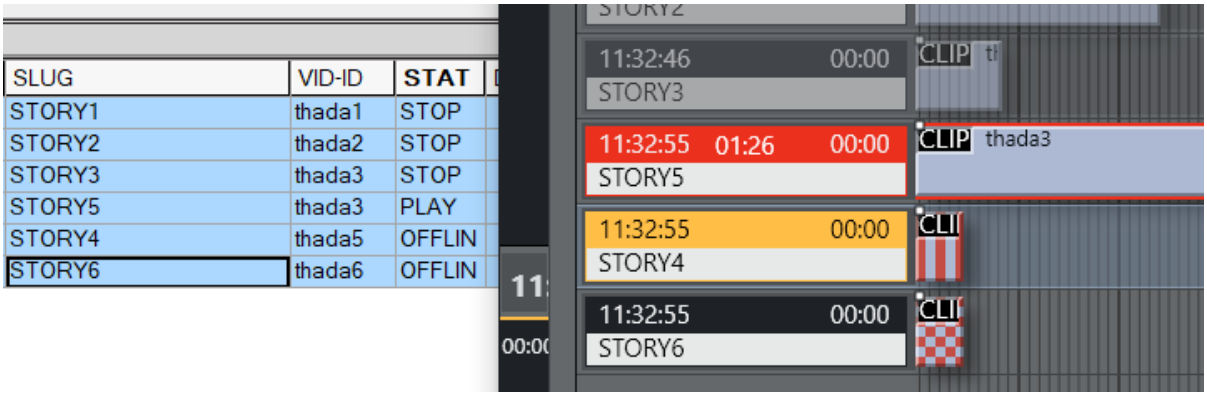


In the illustration above, Story S2 is playing and S3 is “CUED”. Since the first item in S3 is a camera the “CUED” status is not sent unless **SendUpdateStatusForAllItems** is set to *true*.

Copyright © 2024 Vizrt



Above, Story 3 is now playing and Story 5 is “CUED”. Clip thada5 for Story 4 is broken and will be failed during cuing



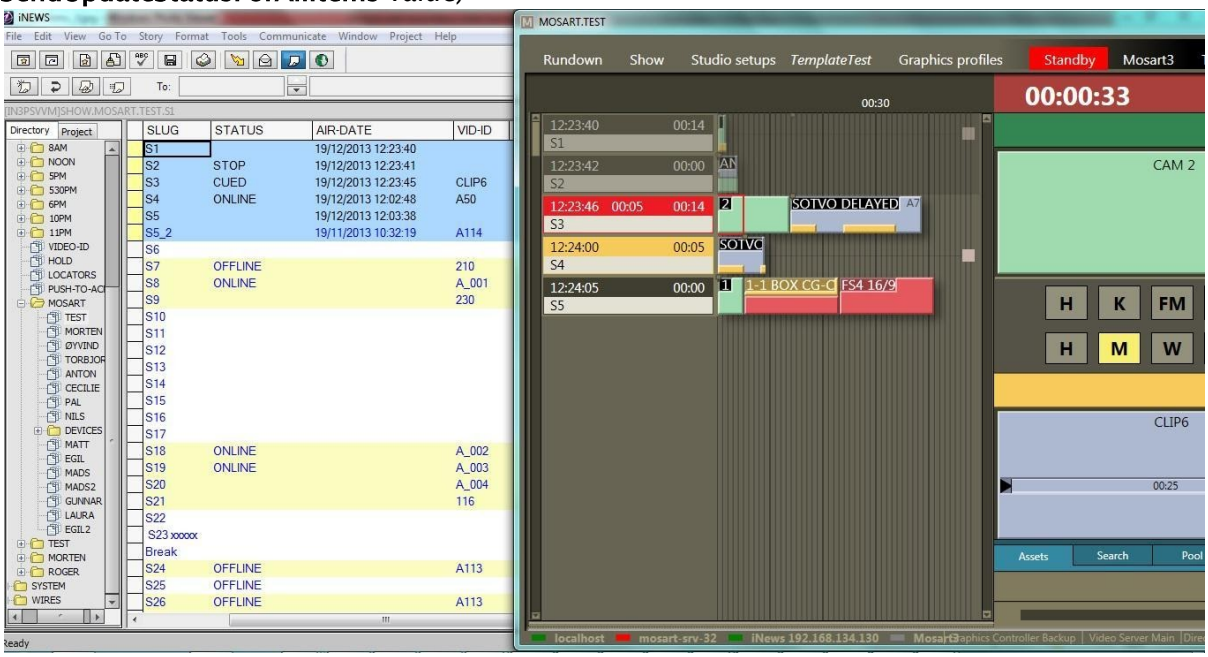
And finally, Story 5 is playing and clip from Story 4 should be cued. Zebra stripes in the Viz Mosart GUI indicate an issue with availability and Story 4 marked as OFFLINE in iNews

Status for Stories Containing More Than One Item

If a story has more than one item, the following statuses are sent when the story is on-air (keeping in mind that if **SendUpdateStatusForAllItems** is *false* the status is not sent if the story does not contain clips):

Copyright © 2024 Vizrt

- PLAY for first item, CUED for next item => thus the story status will be CUED (regardless of **SendUpdateStatusForAllItems** value)



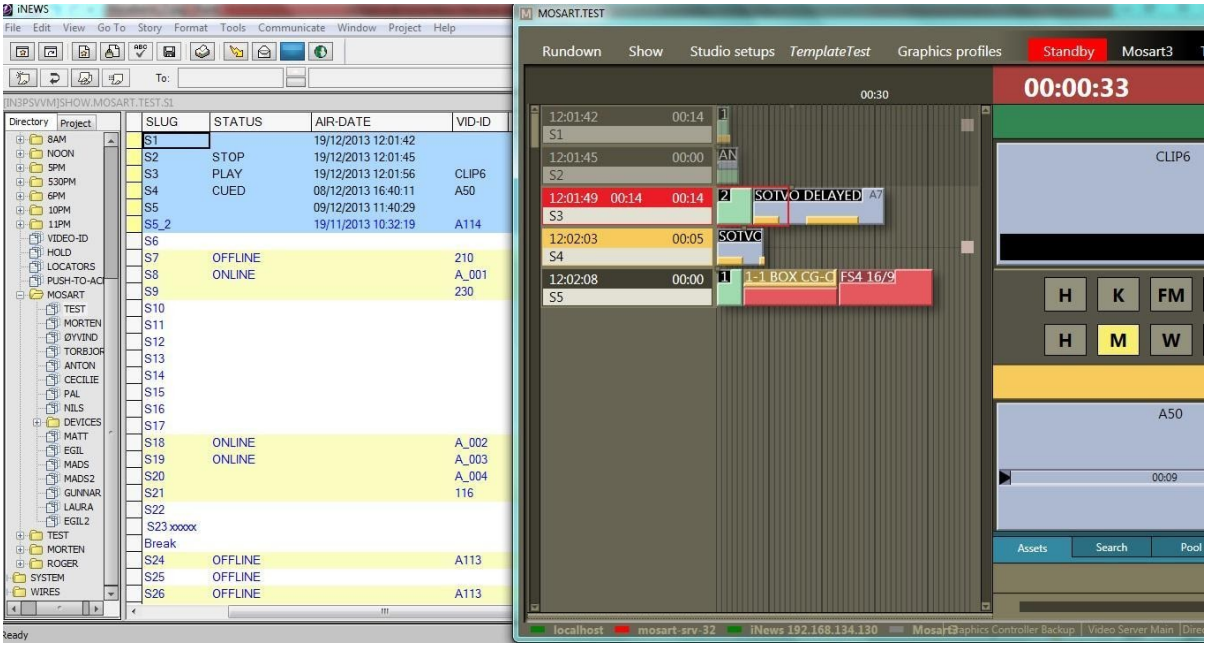
Above illustrates when the first item (camera) in S3 is playing, no “PLAY” status is sent. The Next item is a Package so “CUED” status is sent. In this case, even if **SendUpdateStatusForAllItems** is *true*, the “CUED” status will still be sent for the story.

- When F12 is pressed STOP is sent for the first item, PLAY for the second item and CUED for the third item => thus the story end-up with CUED status (regardless of **SendUpdateStatusForAllItems** value)
  And this will go on until we reach the last item in the story.
- It is only the last item that will send the status PLAY. At any time it is possible to check the AIR-DATE to identify the story currently on-air.

Continuing this example (2 screenshots below), the story S3 has only two items, a Camera and a Package. When we play the clip in the Package, the story gets PLAY status.

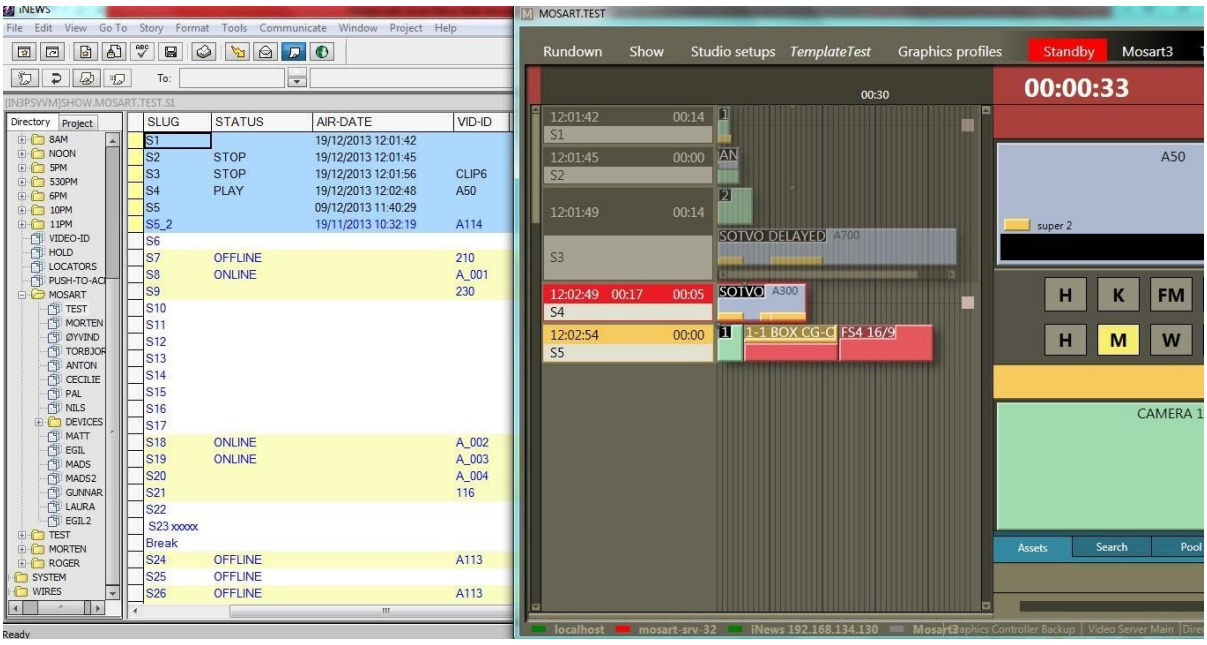
The Package from S3 is playing, this is the last item in the story so “PLAY” status is set. Story S4 will get “CUED” because it contains a clip.

Copyright © 2024 Vizrt



Example 1

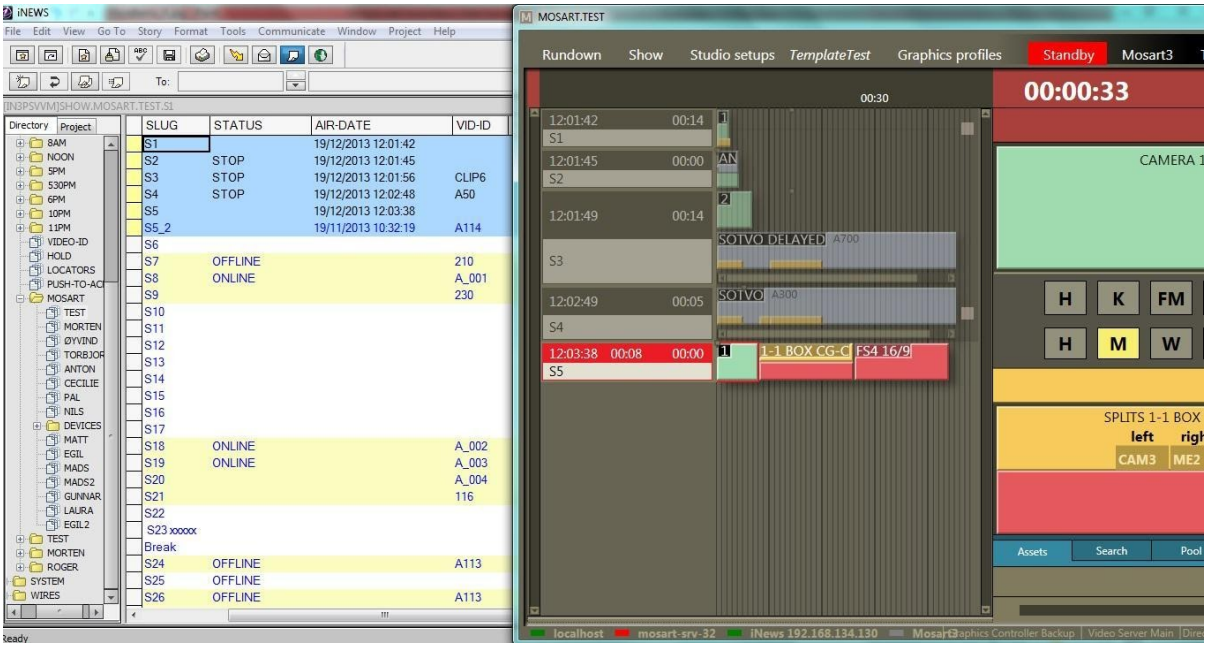
Story S4 containing only a clip is playing. No “CUED” status has been sent for next story since **SendUpdateStatusForAllItems** is set to *false*.



Example 2

Story S5 is playing and since the story does not contain any clips no status will be seen for this unless **SendUpdateStatusForAllItems** is set to *true*.
Only the “AIR-DATE” field shows that this story is on-air.

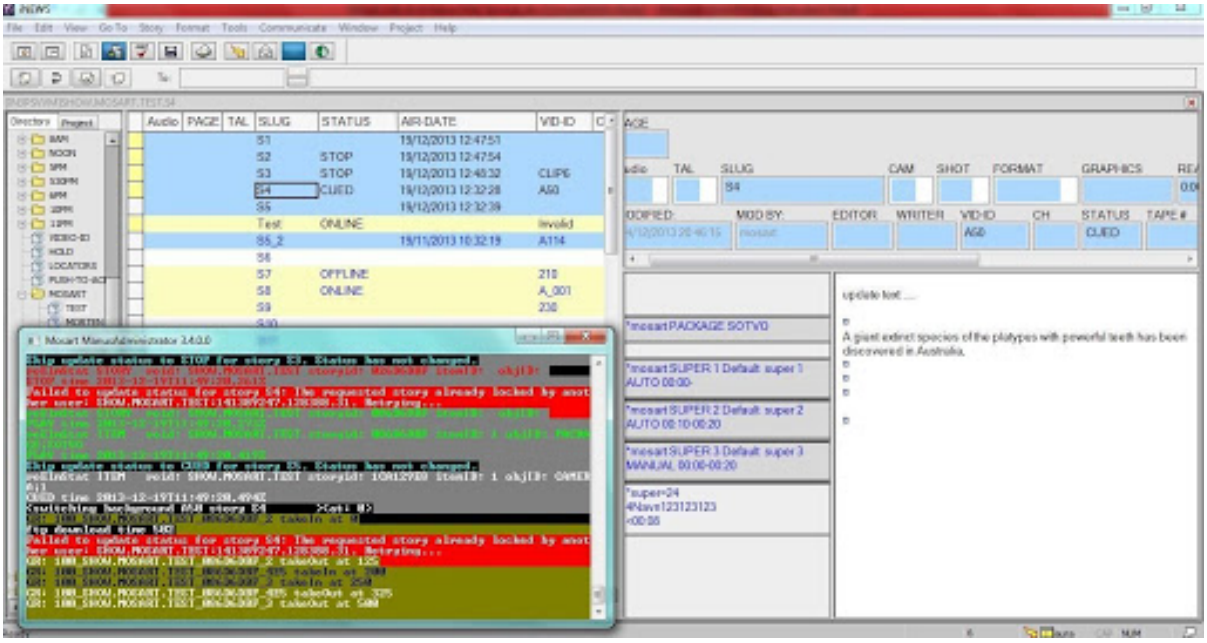
Copyright © 2024 Vizrt



Example 3

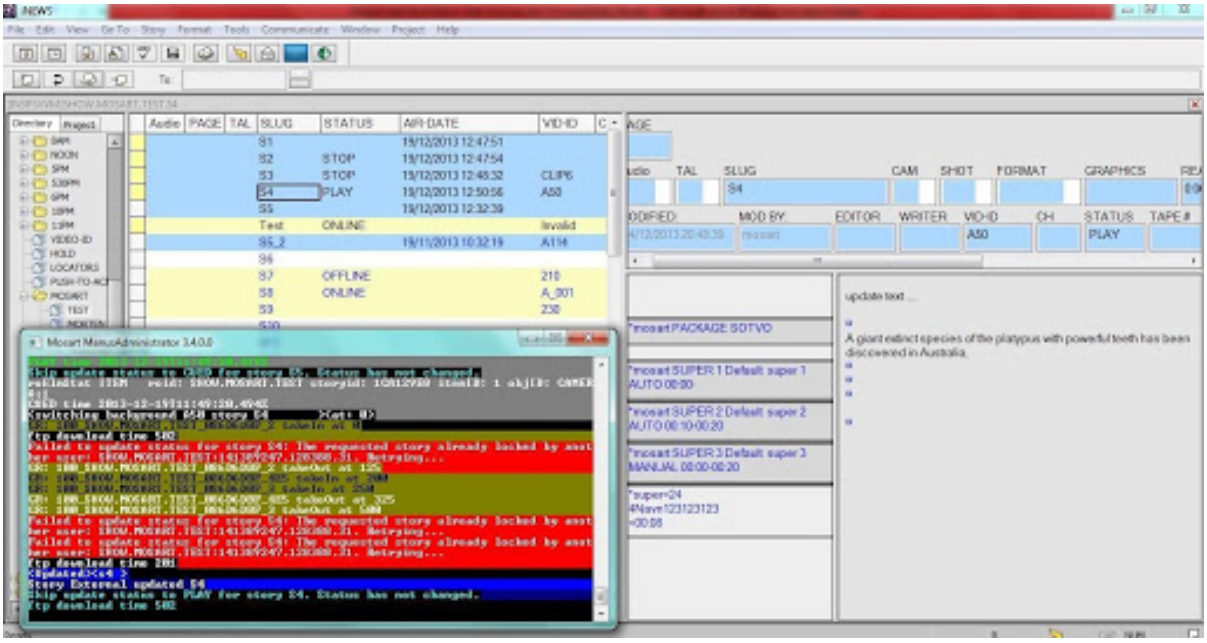
Play status could not be sent for story S4.

When Viz Mosart is doing the updates it locks the form of the story or the entire story depending on the value for **LockFormSectionOnly** parameter and unlocks the story after the update is performed.
If the story form is locked by an iNews user, a message in the Manus Admin console warns the user that the status could not be sent because the story form is locked. Once the story form is unlocked, the status is updated (see Example 4).



Example 4

Story S4 is now unlocked in iNews and “PLAY” status has been sent.



If the timeline is not running and the first story is floated, you may have to reload the rundown in GUI in order to set CUED status for the new first story.

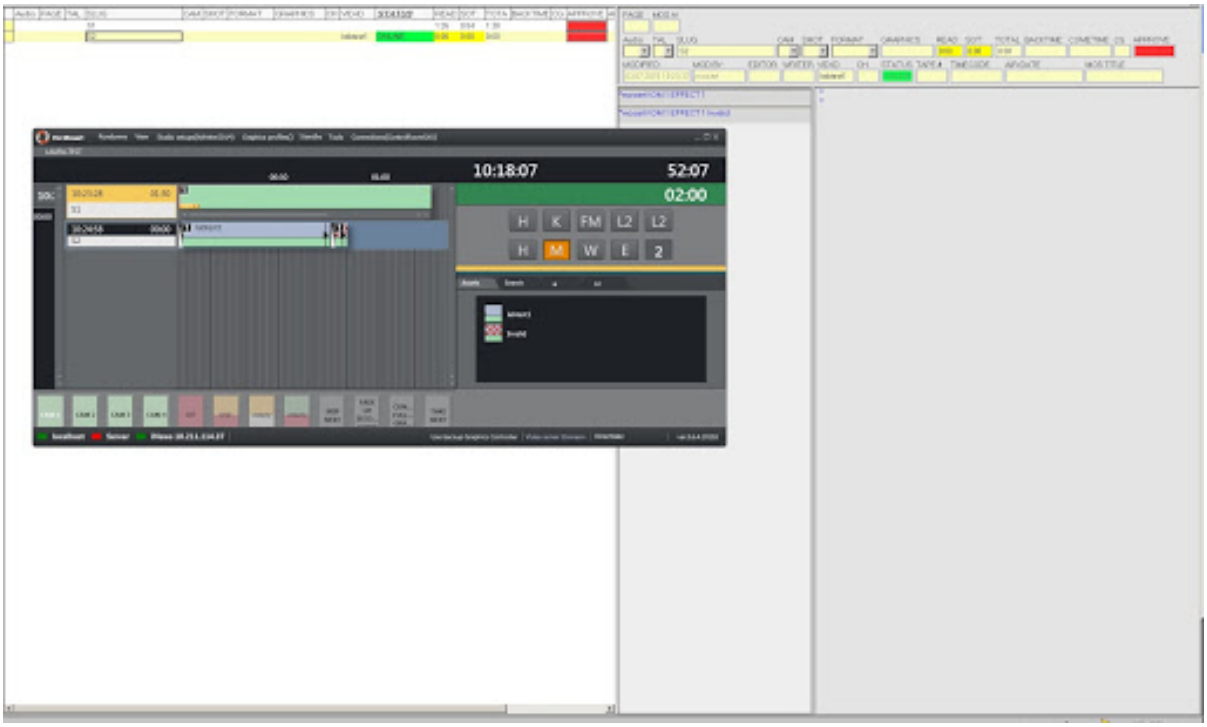
Status of an Off-air Story Containing Multiple Clips

The status of an off-air story that contains more than one clip is given by the first clip in the story.

Consider the following example:

1. A story with VID-ID set and containing two clip items.
2. First clip item has id given by the VID-ID field and it’s an Online clip.
3. Second clip item is given an id, for example, through clip_hirespath attribute (Mosart ActiveX) and it is an Offline clip.
4. The story will have the status depending on the availability of the first clip i.e. OFFLINE.
5. If the position of the clips is changed, then the first clip will get 2 IDs, the one from VID-ID and the one given through Mosart ActiveX.
6. The id given through Mosart ActiveX will give the status of the story, i.e. OFFLINE.

Copyright © 2024 Vizrt

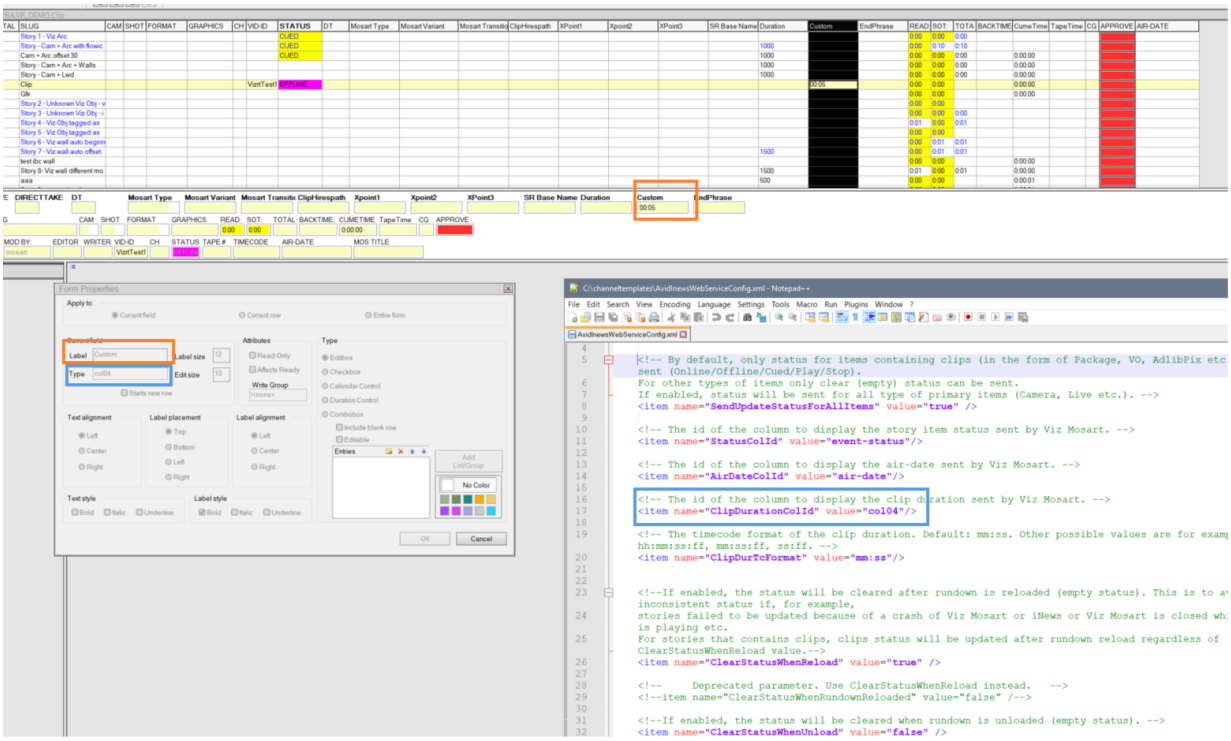


Story *S2* gets the status of the first clip.

Clip Duration Feedback to iNews Server

- The clip duration sent to iNews will be in mm:ss timecode format by default. Other formats are possible to configure. A system-wide frame rate for conversion is assumed and can be configured in Mosart Manus Admin settings.
- iNews can only display information at story level and therefore rules have been defined when a story contains more than one clip. However, for optimal usage of this functionality, it is *recommended to have stories containing only one clip*.
- A column must be configured in iNews to show the duration of a clip found in the story. A column id must also be configured in Mosart, in AvidInewsWebServiceConfig.xml.
  For status and air date, the default iNews columns have been used, but from Mosart 5.1.1, any column can be configured to show the status or the airdate.

Copyright © 2024 Vizrt



- When a rundown is loaded/reloaded in Mosart, Mosart will update all the stories in iNews with **the duration of the first clip** found in each story, in addition to the already existing clip availability status.
- When a clip is offline, the default duration configured in Mosart will be sent.
- When a clip update is received in Mosart, Mosart will send the updated clip duration to iNews.
- In general, when a given story item is cued/played/stopped, Mosart will send the clip duration if that item is a clip.
- If the timeline is not running and the item following the current item cued is a clip, the story will be updated with CUED status and the duration of the clip.
- If a clip that is currently being recorded or copied to the MAM system is included in the rundown, its duration is automatically transmitted to iNews, assuming that Mosart supports the 'growing clip' functionality for the configured video server.
- If a story has multiple items, the following rules apply (this is not new, but the current functionality has impact on how the clip duration is displayed):
    - When the **story is cued**, the status CUED and the duration of the first item in the story, if this a clip, will be sent to iNews.
    - When the **first item in the story is taken**, Mosart sends status STOP for the previous item and the duration if this was a clip.
    - Mosart also sends PLAY **for the first item in the story** and CUED **for the next item in the same story**. Thus, the story ends up with status CUED and the clip duration of the cued item if this is a clip.
    - The same logic is applied until the last item in the story is taken.

Copyright © 2024 Vizrt

- When the **last item in the story is taken** Mosart sends PLAY status and the clip duration if this is a clip.

Example:

In the following rundown:

Stroy1: Clip1
Story2: Cam1 Cam2 Clip2 Clip3 Cam3
Story3: Cam Clip4
Story4: Cam

The following settings are configured:

- **StatusColId**, **AirDateColId** and **ClipDurationColId***:* Set to the corresponding iNews column id.
- **SendUpdateStatusForAllItem:** *true*
- **SetStoryAirDate***: true*
- **Send Clip Status To NRCS**: *true*

When the rundown is loaded, the following information is sent to iNews:

Action	Description	Status (iNews)	Duration (iNews)	Air-Date (iNews)
S1 is cued		CUED	Clip1 duration	
S2		ONLINE/OFFLINE	Clip2 duration	
S3		ONLINE/OFFLINE	Clip4 duration	
S4				

When the rundown is played, the following information displayed in the iNews columns:

Action	Description	Status (iNews)	Duration (iNews)	Air-Date (iNews)
S1 played		S1 PLAY S2 CUED	S1 Clip1 duration	Updated
S2 Cam1 taken	Cam1 is played and Cam2 is cued on S2.	S1 STOP S2 CUED	S1 Clip1 duration	Updated
S2 CAM2 is taken	Cam1 is stopped, Cam2 is played and Clip2 is cued.	S2 CUED	S2 Clip2 duration	
S2 Clip2 is taken	Cam2 is stopped, Clip2 is played, Clip3 is cued.	S2 CUED	S2 Clip3 duration	

Copyright © 2024 Vizrt

S2 Clip3 is taken	Clip2 is stopped, Clip3 is played, Cam3 is cued.	S2 CUED		
S2 CAM3 is taken	Clip3 is stopped, Cam3 is played, S3 Cam is cued.	S2 PLAY S3 CUED		
S3 CAM is taken	S2 Cam3 is stopped, S3 Cam is played, S3 Clip4 is cued.	S2 STOP S3 CUED	S3 Clip4 duration	Updated
S3 Clip4 is taken	S3 Cam is stopped, Clip4 is played, and S4 Cam is cued.	S3 PLAY S4 CUED	S3 Clip4 duration	
S4 Cam is taken	S3 Clip4 is stopped, S4 Cam is played S4 Cam is cued	S3 STOP S4 CUED	S3 Clip4 duration	Updated

13.1.4 Troubleshooting and Known Issues

- Not able to access Web Service page after installation
- The connection to iNews server times out
- "No sessions available on X" – Maximum session
- "Cannot write to iNews Webservice, connect before selectQueue"
- Cannot get any status feedback in iNews
- The Web Service version is not allowed to connect to iNews server

Not able to access Web Service page after installation

If you have problems accessing the Web Services page after installation and deployment (http://<server>:8080/inewswebservice/services), try to delete the Tomcat cache as follow:

1. Stop the Tomcat service.
2. Delete folder Tomcat installation\webapps\inewswebservice (only the folder, not the *war* file).
   You may replace the war file, for example, with a new one when upgrading (this solution is same as deploying from Tomcat Manager).
3. Delete folder Tomcat installation\work\Catalina\localhost\inewswebservice
4. Start the Tomcat service and try again to access the Web Services page.

Copyright © 2024 Vizrt

The connection to iNews server times out

If the connection to the iNews server through the iNews Web Service times out, check first the iNews server logs for more details.

If the iNews server logs contain "SSL_access failed", check what version of Java you are using. SSLv3 is disabled in Java 8 by default.

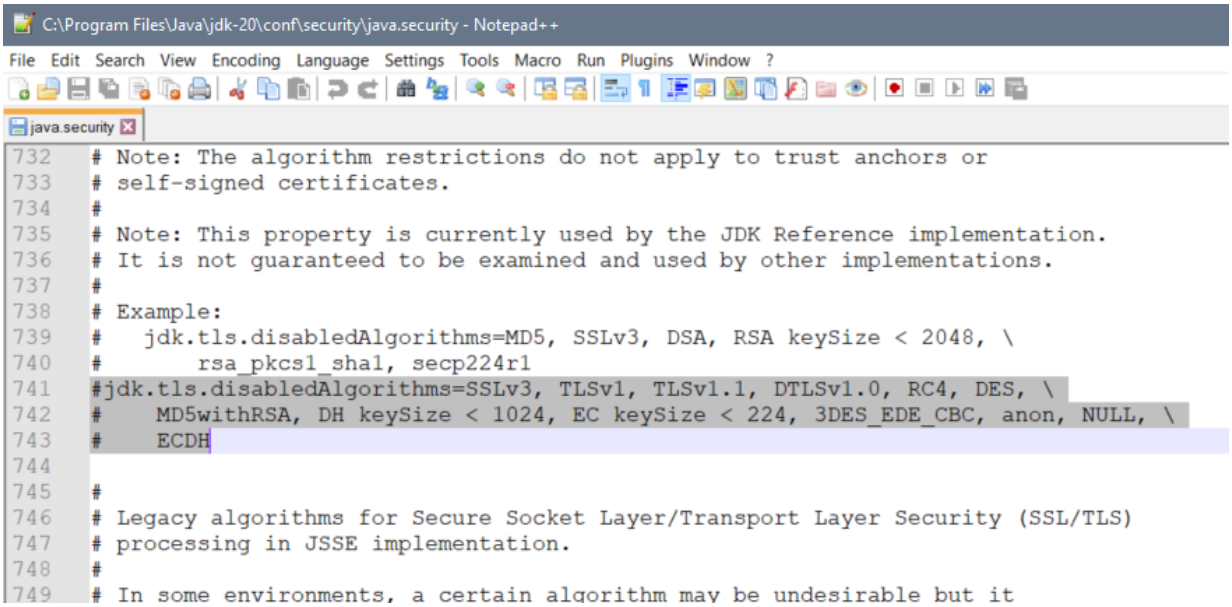
A default Java 8 installation disables the SSLv3 algorithm, preventing WSAPI from connecting to the Newsroom Management Server.

- To re-enable SSLv3 algorithm, open as administrator the file java.security located in the *lib/security* or *conf\security* in your JRE or JDK installation folder

  /user/lib/jvm/java-.1.8.0-openjdk-1.8.0.xxxxx/jre/lib/security

  C:\Program Files\Java\jdk-20\conf\security\java.security

  and comment out the lines that disables SSLv3. Restarting the Apache Tomcat service once the modifications are done.



"No sessions available on X" – Maximum session

If the above error is shown in Manus Admin's console or the log (where X is the iNews server IP address or hostname), refer to the section API Session Limits.
If this is not the case, then most probably the issue is that the Manus Admin console failed to release the occupied API session to connect to iNews server through Avid Web Services, when the API session was closed.

Workaround: There are two solutions for this problem.

1. Either restart Apache Tomcat service or
2. Reduce the session timeout in Tomcat to *1 min*, freeing-up the API session 1 min after Manus Administrator was closed.
   If Manus Admin is closed and opened more than maximum API sessions allowed in 1 min, it will reconnect automatically to Web Service after 1 min when at least one session has been

released.
To change the session timeout interval do the following:

1. Stop the Tomcat service.
2. Change Tomcat session timeout interval in Tomcat installation/conf/web.xml to 1 min:
   <session-config>
   <session-timeout>1</session-timeout>
   </session-config>
3. Start the Tomcat service.

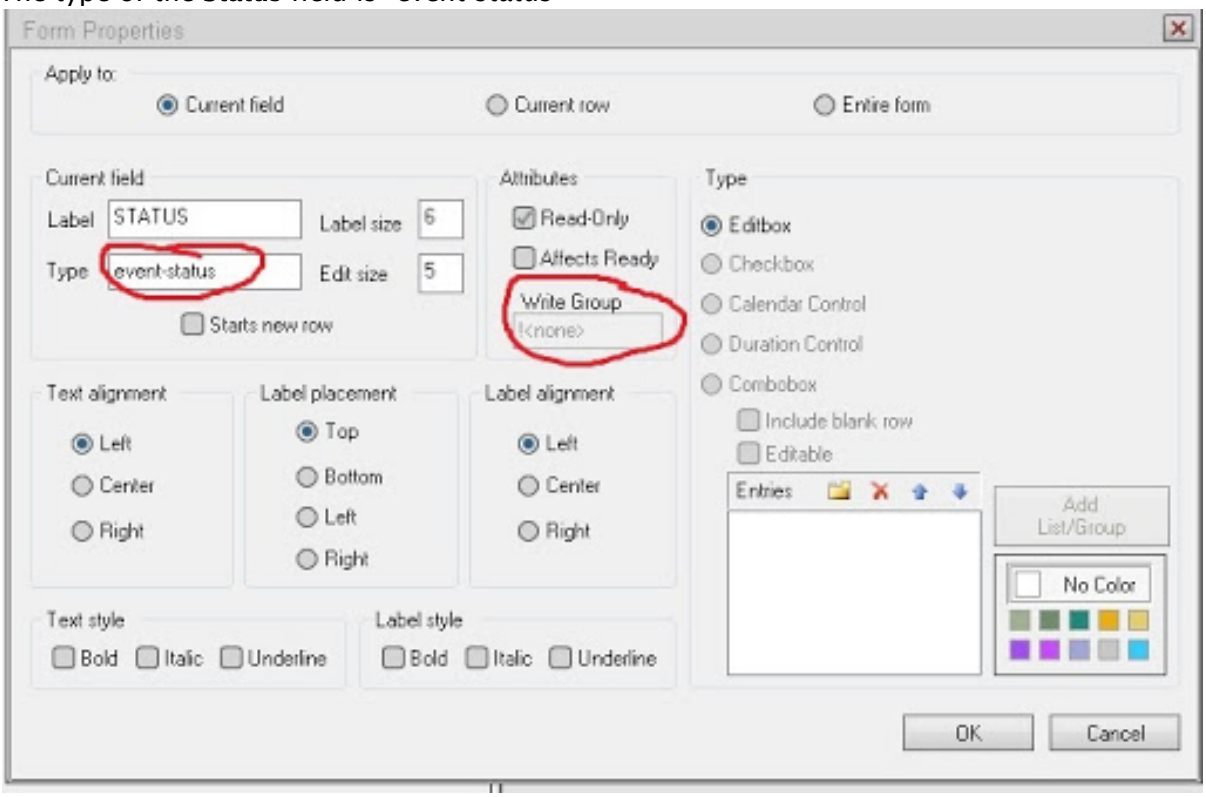
"Cannot write to iNews Webservice, connect before selectQueue"

If you get the error message "cannot write to iNews Webservice, connect before selectQueue" check that the version of Web Service running corresponds with the version set in iNews.
See section Checking the Current Version of Tomcat.

Cannot get any status feedback in iNews

If after setting up everything as described in the document, no status is written in the **Status** field, check the following:

- The **Status** field has write permissions for the user that connects Viz Mosart to the iNews system.
- The type of the **Status** field is "event-status"



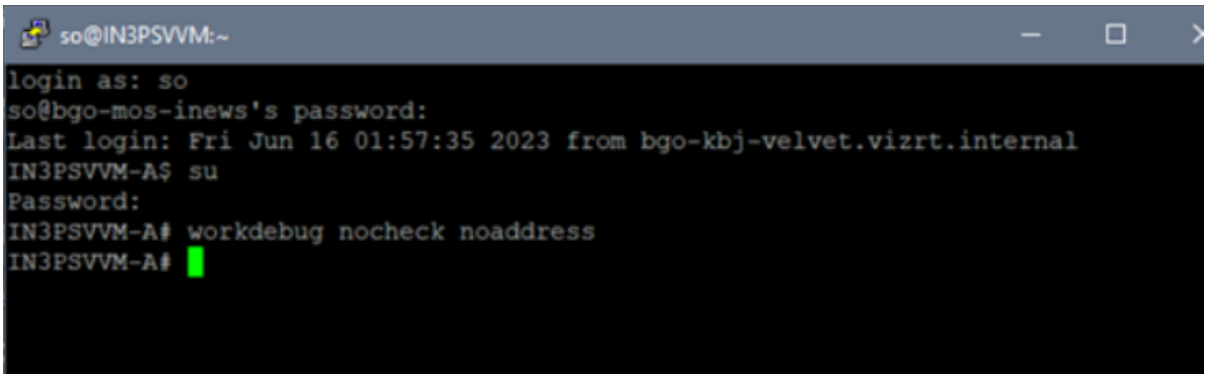
- The queue must *not* be monitored with the iNews Monitor tool.

Copyright © 2024 Vizrt

Verify also the Mosart logs or the iNews Web service logs located under C:\Program Files\Apache Software Foundation\Tomcat 9.0\logs\iNEWS_WS for any error related to status feedback to iNews.

The Web Service version is not allowed to connect to iNews server

If you also encounter the issue "*The Web Service version is not allowed to connect to...*", then disable the version check on the iNews server by executing *workdebug nocheck noaddress* on the iNews server:



13.1.5 Useful Links

- http://tomcat.apache.org/whichversion.html
- http://java.com/en/download/index.jsp
- iNews Mos Gateway Installation and Operations Manual v2.6.0
- iNews Mos Gateway v4.0 ReadMe
- iNews v4.5 Setup and Configuration Guide
- Introduction to iNews Workstation

Copyright © 2024 Vizrt

13.2 GPIO Sample XML Config

```
Working configuration from Viz Mosart Lab in Bergen, October 2014
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
V2.30, Aug  1 2013, 12:55:23
-->
<io-Digital12x12.3>
    <Config>
        <Device>
            <Text>
                <DeviceName>WEBIO-<wut1></DeviceName>
                <DeviceText> From the terminal block directly to the network </
DeviceText>
                <Location></Location>
                <Contact></Contact>
            </Text>
            <Time_Date>
                <TimeZone>
                    <UTCoffset>01:00</UTCoffset>
                    <Enable>
                        <Apply_Time_Zone>ON</Apply_Time_Zone>
                    </Enable>
                    <Summertime>
                        <UTCoffset>02:00</UTCoffset>
                        <Enable>
                            <Apply_Summertime>ON</Apply_Summertime>
                        </Enable>
                        <Start>
                            <Month>
                                <January>OFF</January>
                                <February>OFF</February>
                                <March>ON</March>
                                <April>OFF</April>
                                <May>OFF</May>
                                <June>OFF</June>
                                <July>OFF</July>
                                <August>OFF</August>
                                <September>OFF</September>
                                <October>OFF</October>
                                <November>OFF</November>
                                <December>OFF</December>
                            </Month>
                            <Mode>
                                <first>OFF</first>
                                <second>OFF</second>
                                <third>OFF</third>
                                <fourth>OFF</fourth>
```

Copyright © 2024 Vizrt

```
                    <last>ON</last>
                </Mode>
                <Weekday>
                    <Sunday>ON</Sunday>
                    <Monday>OFF</Monday>
                    <Tuesday>OFF</Tuesday>
                    <Wednesday>OFF</Wednesday>
                    <Thursday>OFF</Thursday>
                    <Friday>OFF</Friday>
                    <Saturday>OFF</Saturday>
                </Weekday>
                <Time>02:00</Time>
            </Start>
            <Stop>
                <Month>
                    <January>OFF</January>
                    <February>OFF</February>
                    <March>OFF</March>
                    <April>OFF</April>
                    <May>OFF</May>
                    <June>OFF</June>
                    <July>OFF</July>
                    <August>OFF</August>
                    <September>OFF</September>
                    <October>ON</October>
                    <November>OFF</November>
                    <December>OFF</December>
                </Month>
                <Mode>
                    <first>OFF</first>
                    <second>OFF</second>
                    <third>OFF</third>
                    <fourth>OFF</fourth>
                    <last>ON</last>
                </Mode>
                <Weekday>
                    <Sunday>ON</Sunday>
                    <Monday>OFF</Monday>
                    <Tuesday>OFF</Tuesday>
                    <Wednesday>OFF</Wednesday>
                    <Thursday>OFF</Thursday>
                    <Friday>OFF</Friday>
                    <Saturday>OFF</Saturday>
                </Weekday>
                <Time>03:00</Time>
            </Stop>
        </Summertime>
    </TimeZone>
    <TimeServer>
        <UTCServer1>de.pool.ntp.org</UTCServer1>
        <UTCServer2>europe.pool.ntp.org</UTCServer2>
        <SyncTime>0</SyncTime>
```

Copyright © 2024 Vizrt

```
            <Enable>
                <Apply_TimeServer>OFF</Apply_TimeServer>
            </Enable>
        </TimeServer>
        <DeviceClock>
            <Time>15:56</Time>
            <Day>01</Day>
            <Month>01</Month>
            <Year>2004</Year>
        </DeviceClock>
    </Time_Date>
    <BasicSettings>
        <Network>
            <IP_Addr>10.211.114.124</IP_Addr>
            <Subnet_Mask>255.255.255.00</Subnet_Mask>
            <Gateway>10.211.114.01</Gateway>
            <BOOTPClient>
                <STATIC>ON</STATIC>
                <BOOTP_enable>OFF</BOOTP_enable>
                <DHCP_enable>OFF</DHCP_enable>
            </BOOTPClient>
            <DnsServer1></DnsServer1>
            <DnsServer2></DnsServer2>
            <KeepAliveTime>00</KeepAliveTime>
        </Network>
        <HTTP_Set>
            <Startup>
                <index.htm>ON</index.htm>
                <home.htm>OFF</home.htm>
                <user.htm>OFF</user.htm>
            </Startup>
            <Enable>
                <GET_Header_enable>OFF</GET_Header_enable>
                <GET_HTTP_enable>OFF</GET_HTTP_enable>
            </Enable>
            <_HTTP_Port>80</_HTTP_Port>
            <InputTrigger>
                <Input_0>ON</Input_0>
                <Input_1>ON</Input_1>
                <Input_2>ON</Input_2>
                <Input_3>ON</Input_3>
                <Input_4>ON</Input_4>
                <Input_5>ON</Input_5>
                <Input_6>ON</Input_6>
                <Input_7>ON</Input_7>
                <Input_8>ON</Input_8>
                <Input_9>ON</Input_9>
                <Input_10>ON</Input_10>
                <Input_11>ON</Input_11>
            </InputTrigger>
        </HTTP_Set>
        <HTTP_Client>
```

Copyright © 2024 Vizrt

```
    <HTTP_Client>
        <HTTP_Client_enable>OFF</HTTP_Client_enable>
    </HTTP_Client>
    <Enable>
        <GET_Header_enable>OFF</GET_Header_enable>
        <GET_HTTP_enable>OFF</GET_HTTP_enable>
    </Enable>
    <LocalPort>AUTO</LocalPort>
    <ServerPort></ServerPort>
    <Server_IP_Addr></Server_IP_Addr>
    <ServerUrl></ServerUrl>
    <InactiveTimeout></InactiveTimeout>
    <InputTrigger>
        <Input_0>OFF</Input_0>
        <Input_1>OFF</Input_1>
        <Input_2>OFF</Input_2>
        <Input_3>OFF</Input_3>
        <Input_4>OFF</Input_4>
        <Input_5>OFF</Input_5>
        <Input_6>OFF</Input_6>
        <Input_7>OFF</Input_7>
        <Input_8>OFF</Input_8>
        <Input_9>OFF</Input_9>
        <Input_10>OFF</Input_10>
        <Input_11>OFF</Input_11>
    </InputTrigger>
    <Timer_Cron></Timer_Cron>
</HTTP_Client>
<Mail>
    <Name>Web-IO 12xDigital</Name>
    <ReplyAddr>Web-IO@no.reply</ReplyAddr>
    <MailServer></MailServer>
    <Authentication>
        <_SMTP_authentication_off>ON</_SMTP_authentication_off>
        <_ESMTP>OFF</_ESMTP>
        <_SMTP_after_POP3>OFF</_SMTP_after_POP3>
    </Authentication>
    <User></User>
    <Password></Password>
    <POP3_Server></POP3_Server>
    <Enable>
        <Mail_enable>OFF</Mail_enable>
    </Enable>
</Mail>
<SNMP>
    <Community_String_Read></Community_String_Read>
    <Community_String_Read_Write></Community_String_Read_Write>
    <Manager_IP></Manager_IP>
    <SystemTraps>
        <Cold_Start>OFF</Cold_Start>
        <Warm_Start>OFF</Warm_Start>
        <Diag_Messages>OFF</Diag_Messages>
```

Copyright © 2024 Vizrt

```
    </SystemTraps>
    <Enable>
        <SNMP_enable>ON</SNMP_enable>
    </Enable>
</SNMP>
<UDP>
    <Port>42279</Port>
    <RemotePort></RemotePort>
    <Enable>
        <UDP_enable>OFF</UDP_enable>
    </Enable>
</UDP>
<Syslog>
    <SyslogServerIP></SyslogServerIP>
    <SyslogServerPort>514</SyslogServerPort>
    <SystemMessages>
        <Cold_Start>OFF</Cold_Start>
        <Warm_Start>OFF</Warm_Start>
        <Diag_Messages>OFF</Diag_Messages>
    </SystemMessages>
    <Enable>
        <System_Messages_enable>OFF</System_Messages_enable>
    </Enable>
</Syslog>
<FTP>
    <FtpServerIP></FtpServerIP>
    <FtpControlPort>21</FtpControlPort>
    <User></User>
    <Password></Password>
    <FtpAccount></FtpAccount>
    <Options>
        <PASV>OFF</PASV>
    </Options>
    <Enable>
        <FTP_enable>OFF</FTP_enable>
    </Enable>
</FTP>
<Binary1>
    <OperationMode>
        <TCP_Server>ON</TCP_Server>
        <TCP_Client>OFF</TCP_Client>
        <UDP_Peer>OFF</UDP_Peer>
    </OperationMode>
    <Enable>
        <Enable_BINARY_1>ON</Enable_BINARY_1>
    </Enable>
    <TcpServer1>
        <LocalPort>49153</LocalPort>
        <Client_HTTP_Port>80</Client_HTTP_Port>
        <InputTrigger>
            <Input_0>ON</Input_0>
            <Input_1>ON</Input_1>
```

```
            <Input_2>ON</Input_2>
            <Input_3>ON</Input_3>
            <Input_4>ON</Input_4>
            <Input_5>ON</Input_5>
            <Input_6>ON</Input_6>
            <Input_7>ON</Input_7>
            <Input_8>ON</Input_8>
            <Input_9>ON</Input_9>
            <Input_10>ON</Input_10>
            <Input_11>ON</Input_11>
        </InputTrigger>
        <ApplicationMode1>
            <Socket_device>ON</Socket_device>
            <Compatible_50xxx>OFF</Compatible_50xxx>
            <Box2Box_Slave>OFF</Box2Box_Slave>
            <OPC_Device>OFF</OPC_Device>
        </ApplicationMode1>
    </TcpServer1>
    <TcpClient1>
        <LocalPort>AUTO</LocalPort>
        <ServerPort>49153</ServerPort>
        <Server_HTTP_Port>80</Server_HTTP_Port>
        <Server_IP_Addr></Server_IP_Addr>
        <ServerPassword></ServerPassword>
        <InactiveTimeout>30</InactiveTimeout>
        <InputTrigger>
            <Input_0>ON</Input_0>
            <Input_1>ON</Input_1>
            <Input_2>ON</Input_2>
            <Input_3>ON</Input_3>
            <Input_4>ON</Input_4>
            <Input_5>ON</Input_5>
            <Input_6>ON</Input_6>
            <Input_7>ON</Input_7>
            <Input_8>ON</Input_8>
            <Input_9>ON</Input_9>
            <Input_10>ON</Input_10>
            <Input_11>ON</Input_11>
        </InputTrigger>
        <Interval></Interval>
        <ApplicationMode2>
            <Socket_device>ON</Socket_device>
            <Compatible_50xxx>OFF</Compatible_50xxx>
            <Box2Box_Master>OFF</Box2Box_Master>
            <ILOface>OFF</ILOface>
        </ApplicationMode2>
    </TcpClient1>
    <UdpPeer1>
        <LocalPort>45889</LocalPort>
        <RemotePort>45889</RemotePort>
        <Remote_IP_Addr></Remote_IP_Addr>
        <InputTrigger>
```

Copyright © 2024 Vizrt

```
                <Input_0>ON</Input_0>
                <Input_1>ON</Input_1>
                <Input_2>ON</Input_2>
                <Input_3>ON</Input_3>
                <Input_4>ON</Input_4>
                <Input_5>ON</Input_5>
                <Input_6>ON</Input_6>
                <Input_7>ON</Input_7>
                <Input_8>ON</Input_8>
                <Input_9>ON</Input_9>
                <Input_10>ON</Input_10>
                <Input_11>ON</Input_11>
            </InputTrigger>
            <Interval></Interval>
            <ApplicationMode3>
                <Socket_device>ON</Socket_device>
                <Compatible_50xxx>OFF</Compatible_50xxx>
            </ApplicationMode3>
        </UdpPeer1>
    </Binary1>
    <Binary2>
        <OperationMode>
            <TCP_Server>ON</TCP_Server>
            <TCP_Client>OFF</TCP_Client>
            <UDP_Peer>OFF</UDP_Peer>
        </OperationMode>
        <Enable>
            <Enable_BINARY_2>ON</Enable_BINARY_2>
        </Enable>
        <TcpServer2>
            <LocalPort>49154</LocalPort>
            <Client_HTTP_Port>80</Client_HTTP_Port>
            <InputTrigger>
                <Input_0>ON</Input_0>
                <Input_1>ON</Input_1>
                <Input_2>ON</Input_2>
                <Input_3>ON</Input_3>
                <Input_4>ON</Input_4>
                <Input_5>ON</Input_5>
                <Input_6>ON</Input_6>
                <Input_7>ON</Input_7>
                <Input_8>ON</Input_8>
                <Input_9>ON</Input_9>
                <Input_10>ON</Input_10>
                <Input_11>ON</Input_11>
            </InputTrigger>
            <ApplicationMode1>
                <Socket_device>ON</Socket_device>
                <Compatible_50xxx>OFF</Compatible_50xxx>
                <Box2Box_Slave>OFF</Box2Box_Slave>
                <OPC_Device>OFF</OPC_Device>
            </ApplicationMode1>
```

Copyright © 2024 Vizrt

```
</TcpServer2>
<TcpClient2>
    <LocalPort>AUTO</LocalPort>
    <ServerPort>49154</ServerPort>
    <Server_HTTP_Port>80</Server_HTTP_Port>
    <Server_IP_Addr></Server_IP_Addr>
    <ServerPassword></ServerPassword>
    <InactiveTimeout>30</InactiveTimeout>
    <InputTrigger>
        <Input_0>ON</Input_0>
        <Input_1>ON</Input_1>
        <Input_2>ON</Input_2>
        <Input_3>ON</Input_3>
        <Input_4>ON</Input_4>
        <Input_5>ON</Input_5>
        <Input_6>ON</Input_6>
        <Input_7>ON</Input_7>
        <Input_8>ON</Input_8>
        <Input_9>ON</Input_9>
        <Input_10>ON</Input_10>
        <Input_11>ON</Input_11>
    </InputTrigger>
    <Interval></Interval>
    <ApplicationMode2>
        <Socket_device>ON</Socket_device>
        <Compatible_50xxx>OFF</Compatible_50xxx>
        <Box2Box_Master>OFF</Box2Box_Master>
        <ILOface>OFF</ILOface>
    </ApplicationMode2>
</TcpClient2>
<UdpPeer2>
    <LocalPort>45890</LocalPort>
    <RemotePort>45890</RemotePort>
    <Remote_IP_Addr></Remote_IP_Addr>
    <InputTrigger>
        <Input_0>ON</Input_0>
        <Input_1>ON</Input_1>
        <Input_2>ON</Input_2>
        <Input_3>ON</Input_3>
        <Input_4>ON</Input_4>
        <Input_5>ON</Input_5>
        <Input_6>ON</Input_6>
        <Input_7>ON</Input_7>
        <Input_8>ON</Input_8>
        <Input_9>ON</Input_9>
        <Input_10>ON</Input_10>
        <Input_11>ON</Input_11>
    </InputTrigger>
    <Interval></Interval>
    <ApplicationMode3>
        <Socket_device>ON</Socket_device>
        <Compatible_50xxx>OFF</Compatible_50xxx>
```

Copyright © 2024 Vizrt

```
            </ApplicationMode3>
        </UdpPeer2>
    </Binary2>
</BasicSettings>
<OutputMode>
    <Output0>
        <_HTTP>ON</_HTTP>
        <_UDP_ASCII>OFF</_UDP_ASCII>
        <_SNMP>OFF</_SNMP>
        <Binary_1>OFF</Binary_1>
        <Binary_2>OFF</Binary_2>
        <_Logic>OFF</_Logic>
    </Output0>
    <Output1>
        <_HTTP>ON</_HTTP>
        <_UDP_ASCII>OFF</_UDP_ASCII>
        <_SNMP>OFF</_SNMP>
        <Binary_1>OFF</Binary_1>
        <Binary_2>OFF</Binary_2>
        <_Logic>OFF</_Logic>
    </Output1>
    <Output2>
        <_HTTP>ON</_HTTP>
        <_UDP_ASCII>OFF</_UDP_ASCII>
        <_SNMP>OFF</_SNMP>
        <Binary_1>OFF</Binary_1>
        <Binary_2>OFF</Binary_2>
        <_Logic>OFF</_Logic>
    </Output2>
    <Output3>
        <_HTTP>ON</_HTTP>
        <_UDP_ASCII>OFF</_UDP_ASCII>
        <_SNMP>OFF</_SNMP>
        <Binary_1>OFF</Binary_1>
        <Binary_2>OFF</Binary_2>
        <_Logic>OFF</_Logic>
    </Output3>
    <Output4>
        <_HTTP>ON</_HTTP>
        <_UDP_ASCII>OFF</_UDP_ASCII>
        <_SNMP>OFF</_SNMP>
        <Binary_1>OFF</Binary_1>
        <Binary_2>OFF</Binary_2>
        <_Logic>OFF</_Logic>
    </Output4>
    <Output5>
        <_HTTP>ON</_HTTP>
        <_UDP_ASCII>OFF</_UDP_ASCII>
        <_SNMP>OFF</_SNMP>
        <Binary_1>OFF</Binary_1>
        <Binary_2>OFF</Binary_2>
        <_Logic>OFF</_Logic>
```

Copyright © 2024 Vizrt

```
</Output5>
<Output6>
    <_HTTP>OFF</_HTTP>
    <_UDP_ASCII>OFF</_UDP_ASCII>
    <_SNMP>OFF</_SNMP>
    <Binary_1>ON</Binary_1>
    <Binary_2>OFF</Binary_2>
    <_Logic>OFF</_Logic>
</Output6>
<Output7>
    <_HTTP>OFF</_HTTP>
    <_UDP_ASCII>OFF</_UDP_ASCII>
    <_SNMP>OFF</_SNMP>
    <Binary_1>ON</Binary_1>
    <Binary_2>OFF</Binary_2>
    <_Logic>OFF</_Logic>
</Output7>
<Output8>
    <_HTTP>OFF</_HTTP>
    <_UDP_ASCII>OFF</_UDP_ASCII>
    <_SNMP>OFF</_SNMP>
    <Binary_1>ON</Binary_1>
    <Binary_2>OFF</Binary_2>
    <_Logic>OFF</_Logic>
</Output8>
<Output9>
    <_HTTP>OFF</_HTTP>
    <_UDP_ASCII>OFF</_UDP_ASCII>
    <_SNMP>OFF</_SNMP>
    <Binary_1>ON</Binary_1>
    <Binary_2>OFF</Binary_2>
    <_Logic>OFF</_Logic>
</Output9>
<Output10>
    <_HTTP>OFF</_HTTP>
    <_UDP_ASCII>OFF</_UDP_ASCII>
    <_SNMP>OFF</_SNMP>
    <Binary_1>ON</Binary_1>
    <Binary_2>OFF</Binary_2>
    <_Logic>OFF</_Logic>
</Output10>
<Output11>
    <_HTTP>OFF</_HTTP>
    <_UDP_ASCII>OFF</_UDP_ASCII>
    <_SNMP>OFF</_SNMP>
    <Binary_1>ON</Binary_1>
    <Binary_2>OFF</Binary_2>
    <_Logic>OFF</_Logic>
</Output11>
<Safety_Timeout></Safety_Timeout>
<Enable>
    <Load_Control_enable>OFF</Load_Control_enable>
```

Copyright © 2024 Vizrt

```
                </Enable>
            </OutputMode>
            <Alarm>
                <Alarm1>
                    <InputTrigger>
                        <Input_0><enable>OFF</enable><state>OFF</state></Input_0>
                        <Input_1><enable>OFF</enable><state>OFF</state></Input_1>
                        <Input_2><enable>OFF</enable><state>OFF</state></Input_2>
                        <Input_3><enable>OFF</enable><state>OFF</state></Input_3>
                        <Input_4><enable>OFF</enable><state>OFF</state></Input_4>
                        <Input_5><enable>OFF</enable><state>OFF</state></Input_5>
                        <Input_6><enable>OFF</enable><state>OFF</state></Input_6>
                        <Input_7><enable>OFF</enable><state>OFF</state></Input_7>
                        <Input_8><enable>OFF</enable><state>OFF</state></Input_8>
                        <Input_9><enable>OFF</enable><state>OFF</state></Input_9>
                        <Input_10><enable>OFF</enable><state>OFF</state></Input_10>
                        <Input_11><enable>OFF</enable><state>OFF</state></Input_11>
                    </InputTrigger>
                    <OutputTrigger>
                        <Output_0><enable>OFF</enable><state>OFF</state></Output_0>
                        <Output_1><enable>OFF</enable><state>OFF</state></Output_1>
                        <Output_2><enable>OFF</enable><state>OFF</state></Output_2>
                        <Output_3><enable>OFF</enable><state>OFF</state></Output_3>
                        <Output_4><enable>OFF</enable><state>OFF</state></Output_4>
                        <Output_5><enable>OFF</enable><state>OFF</state></Output_5>
                        <Output_6><enable>OFF</enable><state>OFF</state></Output_6>
                        <Output_7><enable>OFF</enable><state>OFF</state></Output_7>
                        <Output_8><enable>OFF</enable><state>OFF</state></Output_8>
                        <Output_9><enable>OFF</enable><state>OFF</state></Output_9>
                        <Output_10><enable>OFF</enable><state>OFF</state></Output_10>
                        <Output_11><enable>OFF</enable><state>OFF</state></Output_11>
                    </OutputTrigger>
                    <SystemTrigger>
                        <Load_Control>OFF</Load_Control>
                        <Interval_Timer>OFF</Interval_Timer>
                        <Cold_Start>OFF</Cold_Start>
                        <Warm_Start>OFF</Warm_Start>
                    </SystemTrigger>
                    <MaxCounterValue></MaxCounterValue>
                    <Interval></Interval>
                    <Timer_Cron></Timer_Cron>
                    <AlarmEnable>
                        <Mail_enable>OFF</Mail_enable>
                        <SNMP_Trap_enable>OFF</SNMP_Trap_enable>
                        <UDP_Client_enable>OFF</UDP_Client_enable>
                        <Send_special_alarm_to_pending_TCP_connections>OFF</
Send_special_alarm_to_pending_TCP_connections>
                        <TCP_Client_enable>OFF</TCP_Client_enable>
                        <Syslog_Messages_enable>OFF</Syslog_Messages_enable>
                        <FTP_Client_enable>OFF</FTP_Client_enable>
                    </AlarmEnable>
                    <Mail>
```

```
            <E-Mail-Addr></E-Mail-Addr>
            <Subject></Subject>
            <Mailtext></Mailtext>
            <Alarm_clear_subject></Alarm_clear_subject>
            <Alarm_clear_text></Alarm_clear_text>
        </Mail>
        <SNMP>
            <Manager_IP></Manager_IP>
            <TrapText></TrapText>
            <Alarm_clear_text></Alarm_clear_text>
        </SNMP>
        <UDP>
            <IP_Addr></IP_Addr>
            <Port></Port>
            <UdpText></UdpText>
            <Alarm_clear_text></Alarm_clear_text>
        </UDP>
        <TCP>
            <IP_Addr></IP_Addr>
            <Port></Port>
            <TcpText></TcpText>
            <Alarm_clear_text></Alarm_clear_text>
        </TCP>
        <Syslog>
            <IP_Addr></IP_Addr>
            <Port>514</Port>
            <SyslogText></SyslogText>
            <Alarm_clear_text></Alarm_clear_text>
        </Syslog>
        <FTP>
            <FtpLocalDataPort>AUTO</FtpLocalDataPort>
            <FileName></FileName>
            <FtpAlarmText></FtpAlarmText>
            <Alarm_clear_text></Alarm_clear_text>
            <Options>
                <STORE>OFF</STORE>
                <APPEND>ON</APPEND>
            </Options>
        </FTP>
    </Alarm1>
    <Alarm2>
        <InputTrigger>
            <Input_0><enable>OFF</enable><state>OFF</state></Input_0>
            <Input_1><enable>OFF</enable><state>OFF</state></Input_1>
            <Input_2><enable>OFF</enable><state>OFF</state></Input_2>
            <Input_3><enable>OFF</enable><state>OFF</state></Input_3>
            <Input_4><enable>OFF</enable><state>OFF</state></Input_4>
            <Input_5><enable>OFF</enable><state>OFF</state></Input_5>
            <Input_6><enable>OFF</enable><state>OFF</state></Input_6>
            <Input_7><enable>OFF</enable><state>OFF</state></Input_7>
            <Input_8><enable>OFF</enable><state>OFF</state></Input_8>
            <Input_9><enable>OFF</enable><state>OFF</state></Input_9>
```

Copyright © 2024 Vizrt

```
                        <Input_10><enable>OFF</enable><state>OFF</state></Input_10>
                        <Input_11><enable>OFF</enable><state>OFF</state></Input_11>
                    </InputTrigger>
                    <OutputTrigger>
                        <Output_0><enable>OFF</enable><state>OFF</state></Output_0>
                        <Output_1><enable>OFF</enable><state>OFF</state></Output_1>
                        <Output_2><enable>OFF</enable><state>OFF</state></Output_2>
                        <Output_3><enable>OFF</enable><state>OFF</state></Output_3>
                        <Output_4><enable>OFF</enable><state>OFF</state></Output_4>
                        <Output_5><enable>OFF</enable><state>OFF</state></Output_5>
                        <Output_6><enable>OFF</enable><state>OFF</state></Output_6>
                        <Output_7><enable>OFF</enable><state>OFF</state></Output_7>
                        <Output_8><enable>OFF</enable><state>OFF</state></Output_8>
                        <Output_9><enable>OFF</enable><state>OFF</state></Output_9>
                        <Output_10><enable>OFF</enable><state>OFF</state></Output_10>
                        <Output_11><enable>OFF</enable><state>OFF</state></Output_11>
                    </OutputTrigger>
                    <SystemTrigger>
                        <Load_Control>OFF</Load_Control>
                        <Interval_Timer>OFF</Interval_Timer>
                        <Cold_Start>OFF</Cold_Start>
                        <Warm_Start>OFF</Warm_Start>
                    </SystemTrigger>
                    <MaxCounterValue></MaxCounterValue>
                    <Interval></Interval>
                    <Timer_Cron></Timer_Cron>
                    <AlarmEnable>
                        <Mail_enable>OFF</Mail_enable>
                        <SNMP_Trap_enable>OFF</SNMP_Trap_enable>
                        <UDP_Client_enable>OFF</UDP_Client_enable>
                        <Send_special_alarm_to_pending_TCP_connections>OFF</
Send_special_alarm_to_pending_TCP_connections>
                        <TCP_Client_enable>OFF</TCP_Client_enable>
                        <Syslog_Messages_enable>OFF</Syslog_Messages_enable>
                        <FTP_Client_enable>OFF</FTP_Client_enable>
                    </AlarmEnable>
                    <Mail>
                        <E-Mail-Addr></E-Mail-Addr>
                        <Subject></Subject>
                        <Mailtext></Mailtext>
                        <Alarm_clear_subject></Alarm_clear_subject>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Mail>
                    <SNMP>
                        <Manager_IP></Manager_IP>
                        <TrapText></TrapText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </SNMP>
                    <UDP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <UdpText></UdpText>
```

```
            <Alarm_clear_text></Alarm_clear_text>
        </UDP>
        <TCP>
            <IP_Addr></IP_Addr>
            <Port></Port>
            <TcpText></TcpText>
            <Alarm_clear_text></Alarm_clear_text>
        </TCP>
        <Syslog>
            <IP_Addr></IP_Addr>
            <Port>514</Port>
            <SyslogText></SyslogText>
            <Alarm_clear_text></Alarm_clear_text>
        </Syslog>
        <FTP>
            <FtpLocalDataPort>AUTO</FtpLocalDataPort>
            <FileName></FileName>
            <FtpAlarmText></FtpAlarmText>
            <Alarm_clear_text></Alarm_clear_text>
            <Options>
                <STORE>OFF</STORE>
                <APPEND>ON</APPEND>
            </Options>
        </FTP>
    </Alarm2>
    <Alarm3>
        <InputTrigger>
            <Input_0><enable>OFF</enable><state>OFF</state></Input_0>
            <Input_1><enable>OFF</enable><state>OFF</state></Input_1>
            <Input_2><enable>OFF</enable><state>OFF</state></Input_2>
            <Input_3><enable>OFF</enable><state>OFF</state></Input_3>
            <Input_4><enable>OFF</enable><state>OFF</state></Input_4>
            <Input_5><enable>OFF</enable><state>OFF</state></Input_5>
            <Input_6><enable>OFF</enable><state>OFF</state></Input_6>
            <Input_7><enable>OFF</enable><state>OFF</state></Input_7>
            <Input_8><enable>OFF</enable><state>OFF</state></Input_8>
            <Input_9><enable>OFF</enable><state>OFF</state></Input_9>
            <Input_10><enable>OFF</enable><state>OFF</state></Input_10>
            <Input_11><enable>OFF</enable><state>OFF</state></Input_11>
        </InputTrigger>
        <OutputTrigger>
            <Output_0><enable>OFF</enable><state>OFF</state></Output_0>
            <Output_1><enable>OFF</enable><state>OFF</state></Output_1>
            <Output_2><enable>OFF</enable><state>OFF</state></Output_2>
            <Output_3><enable>OFF</enable><state>OFF</state></Output_3>
            <Output_4><enable>OFF</enable><state>OFF</state></Output_4>
            <Output_5><enable>OFF</enable><state>OFF</state></Output_5>
            <Output_6><enable>OFF</enable><state>OFF</state></Output_6>
            <Output_7><enable>OFF</enable><state>OFF</state></Output_7>
            <Output_8><enable>OFF</enable><state>OFF</state></Output_8>
            <Output_9><enable>OFF</enable><state>OFF</state></Output_9>
            <Output_10><enable>OFF</enable><state>OFF</state></Output_10>
```

Copyright © 2024 Vizrt

```
                        <Output_11><enable>OFF</enable><state>OFF</state></Output_11>
                    </OutputTrigger>
                    <SystemTrigger>
                        <Load_Control>OFF</Load_Control>
                        <Interval_Timer>OFF</Interval_Timer>
                        <Cold_Start>OFF</Cold_Start>
                        <Warm_Start>OFF</Warm_Start>
                    </SystemTrigger>
                    <MaxCounterValue></MaxCounterValue>
                    <Interval></Interval>
                    <Timer_Cron></Timer_Cron>
                    <AlarmEnable>
                        <Mail_enable>OFF</Mail_enable>
                        <SNMP_Trap_enable>OFF</SNMP_Trap_enable>
                        <UDP_Client_enable>OFF</UDP_Client_enable>
                        <Send_special_alarm_to_pending_TCP_connections>OFF</
Send_special_alarm_to_pending_TCP_connections>
                        <TCP_Client_enable>OFF</TCP_Client_enable>
                        <Syslog_Messages_enable>OFF</Syslog_Messages_enable>
                        <FTP_Client_enable>OFF</FTP_Client_enable>
                    </AlarmEnable>
                    <Mail>
                        <E-Mail-Addr></E-Mail-Addr>
                        <Subject></Subject>
                        <Mailtext></Mailtext>
                        <Alarm_clear_subject></Alarm_clear_subject>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Mail>
                    <SNMP>
                        <Manager_IP></Manager_IP>
                        <TrapText></TrapText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </SNMP>
                    <UDP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <UdpText></UdpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </UDP>
                    <TCP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <TcpText></TcpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </TCP>
                    <Syslog>
                        <IP_Addr></IP_Addr>
                        <Port>514</Port>
                        <SyslogText></SyslogText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Syslog>
                    <FTP>
```

Copyright © 2024 Vizrt

```
            <FtpLocalDataPort>AUTO</FtpLocalDataPort>
            <FileName></FileName>
            <FtpAlarmText></FtpAlarmText>
            <Alarm_clear_text></Alarm_clear_text>
            <Options>
                <STORE>OFF</STORE>
                <APPEND>ON</APPEND>
            </Options>
        </FTP>
    </Alarm3>
    <Alarm4>
        <InputTrigger>
            <Input_0><enable>OFF</enable><state>OFF</state></Input_0>
            <Input_1><enable>OFF</enable><state>OFF</state></Input_1>
            <Input_2><enable>OFF</enable><state>OFF</state></Input_2>
            <Input_3><enable>OFF</enable><state>OFF</state></Input_3>
            <Input_4><enable>OFF</enable><state>OFF</state></Input_4>
            <Input_5><enable>OFF</enable><state>OFF</state></Input_5>
            <Input_6><enable>OFF</enable><state>OFF</state></Input_6>
            <Input_7><enable>OFF</enable><state>OFF</state></Input_7>
            <Input_8><enable>OFF</enable><state>OFF</state></Input_8>
            <Input_9><enable>OFF</enable><state>OFF</state></Input_9>
            <Input_10><enable>OFF</enable><state>OFF</state></Input_10>
            <Input_11><enable>OFF</enable><state>OFF</state></Input_11>
        </InputTrigger>
        <OutputTrigger>
            <Output_0><enable>OFF</enable><state>OFF</state></Output_0>
            <Output_1><enable>OFF</enable><state>OFF</state></Output_1>
            <Output_2><enable>OFF</enable><state>OFF</state></Output_2>
            <Output_3><enable>OFF</enable><state>OFF</state></Output_3>
            <Output_4><enable>OFF</enable><state>OFF</state></Output_4>
            <Output_5><enable>OFF</enable><state>OFF</state></Output_5>
            <Output_6><enable>OFF</enable><state>OFF</state></Output_6>
            <Output_7><enable>OFF</enable><state>OFF</state></Output_7>
            <Output_8><enable>OFF</enable><state>OFF</state></Output_8>
            <Output_9><enable>OFF</enable><state>OFF</state></Output_9>
            <Output_10><enable>OFF</enable><state>OFF</state></Output_10>
            <Output_11><enable>OFF</enable><state>OFF</state></Output_11>
        </OutputTrigger>
        <SystemTrigger>
            <Load_Control>OFF</Load_Control>
            <Interval_Timer>OFF</Interval_Timer>
            <Cold_Start>OFF</Cold_Start>
            <Warm_Start>OFF</Warm_Start>
        </SystemTrigger>
        <MaxCounterValue></MaxCounterValue>
        <Interval></Interval>
        <Timer_Cron></Timer_Cron>
        <AlarmEnable>
            <Mail_enable>OFF</Mail_enable>
            <SNMP_Trap_enable>OFF</SNMP_Trap_enable>
            <UDP_Client_enable>OFF</UDP_Client_enable>
```

Copyright © 2024 Vizrt

```
                        <Send_special_alarm_to_pending_TCP_connections>OFF</
Send_special_alarm_to_pending_TCP_connections>
                        <TCP_Client_enable>OFF</TCP_Client_enable>
                        <Syslog_Messages_enable>OFF</Syslog_Messages_enable>
                        <FTP_Client_enable>OFF</FTP_Client_enable>
                    </AlarmEnable>
                    <Mail>
                        <E-Mail-Addr></E-Mail-Addr>
                        <Subject></Subject>
                        <Mailtext></Mailtext>
                        <Alarm_clear_subject></Alarm_clear_subject>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Mail>
                    <SNMP>
                        <Manager_IP></Manager_IP>
                        <TrapText></TrapText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </SNMP>
                    <UDP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <UdpText></UdpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </UDP>
                    <TCP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <TcpText></TcpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </TCP>
                    <Syslog>
                        <IP_Addr></IP_Addr>
                        <Port>514</Port>
                        <SyslogText></SyslogText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Syslog>
                    <FTP>
                        <FtpLocalDataPort>AUTO</FtpLocalDataPort>
                        <FileName></FileName>
                        <FtpAlarmText></FtpAlarmText>
                        <Alarm_clear_text></Alarm_clear_text>
                        <Options>
                            <STORE>OFF</STORE>
                            <APPEND>ON</APPEND>
                        </Options>
                    </FTP>
                </Alarm4>
                <Alarm5>
                    <InputTrigger>
                        <Input_0><enable>OFF</enable><state>OFF</state></Input_0>
                        <Input_1><enable>OFF</enable><state>OFF</state></Input_1>
                        <Input_2><enable>OFF</enable><state>OFF</state></Input_2>
```

Copyright © 2024 Vizrt

```
                        <Input_3><enable>OFF</enable><state>OFF</state></Input_3>
                        <Input_4><enable>OFF</enable><state>OFF</state></Input_4>
                        <Input_5><enable>OFF</enable><state>OFF</state></Input_5>
                        <Input_6><enable>OFF</enable><state>OFF</state></Input_6>
                        <Input_7><enable>OFF</enable><state>OFF</state></Input_7>
                        <Input_8><enable>OFF</enable><state>OFF</state></Input_8>
                        <Input_9><enable>OFF</enable><state>OFF</state></Input_9>
                        <Input_10><enable>OFF</enable><state>OFF</state></Input_10>
                        <Input_11><enable>OFF</enable><state>OFF</state></Input_11>
                    </InputTrigger>
                    <OutputTrigger>
                        <Output_0><enable>OFF</enable><state>OFF</state></Output_0>
                        <Output_1><enable>OFF</enable><state>OFF</state></Output_1>
                        <Output_2><enable>OFF</enable><state>OFF</state></Output_2>
                        <Output_3><enable>OFF</enable><state>OFF</state></Output_3>
                        <Output_4><enable>OFF</enable><state>OFF</state></Output_4>
                        <Output_5><enable>OFF</enable><state>OFF</state></Output_5>
                        <Output_6><enable>OFF</enable><state>OFF</state></Output_6>
                        <Output_7><enable>OFF</enable><state>OFF</state></Output_7>
                        <Output_8><enable>OFF</enable><state>OFF</state></Output_8>
                        <Output_9><enable>OFF</enable><state>OFF</state></Output_9>
                        <Output_10><enable>OFF</enable><state>OFF</state></Output_10>
                        <Output_11><enable>OFF</enable><state>OFF</state></Output_11>
                    </OutputTrigger>
                    <SystemTrigger>
                        <Load_Control>OFF</Load_Control>
                        <Interval_Timer>OFF</Interval_Timer>
                        <Cold_Start>OFF</Cold_Start>
                        <Warm_Start>OFF</Warm_Start>
                    </SystemTrigger>
                    <MaxCounterValue></MaxCounterValue>
                    <Interval></Interval>
                    <Timer_Cron></Timer_Cron>
                    <AlarmEnable>
                        <Mail_enable>OFF</Mail_enable>
                        <SNMP_Trap_enable>OFF</SNMP_Trap_enable>
                        <UDP_Client_enable>OFF</UDP_Client_enable>
                        <Send_special_alarm_to_pending_TCP_connections>OFF</
Send_special_alarm_to_pending_TCP_connections>
                        <TCP_Client_enable>OFF</TCP_Client_enable>
                        <Syslog_Messages_enable>OFF</Syslog_Messages_enable>
                        <FTP_Client_enable>OFF</FTP_Client_enable>
                    </AlarmEnable>
                    <Mail>
                        <E-Mail-Addr></E-Mail-Addr>
                        <Subject></Subject>
                        <Mailtext></Mailtext>
                        <Alarm_clear_subject></Alarm_clear_subject>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Mail>
                    <SNMP>
                        <Manager_IP></Manager_IP>
```

Copyright © 2024 Vizrt

```
            <TrapText></TrapText>
            <Alarm_clear_text></Alarm_clear_text>
        </SNMP>
        <UDP>
            <IP_Addr></IP_Addr>
            <Port></Port>
            <UdpText></UdpText>
            <Alarm_clear_text></Alarm_clear_text>
        </UDP>
        <TCP>
            <IP_Addr></IP_Addr>
            <Port></Port>
            <TcpText></TcpText>
            <Alarm_clear_text></Alarm_clear_text>
        </TCP>
        <Syslog>
            <IP_Addr></IP_Addr>
            <Port>514</Port>
            <SyslogText></SyslogText>
            <Alarm_clear_text></Alarm_clear_text>
        </Syslog>
        <FTP>
            <FtpLocalDataPort>AUTO</FtpLocalDataPort>
            <FileName></FileName>
            <FtpAlarmText></FtpAlarmText>
            <Alarm_clear_text></Alarm_clear_text>
            <Options>
                <STORE>OFF</STORE>
                <APPEND>ON</APPEND>
            </Options>
        </FTP>
    </Alarm5>
    <Alarm6>
        <InputTrigger>
            <Input_0><enable>OFF</enable><state>OFF</state></Input_0>
            <Input_1><enable>OFF</enable><state>OFF</state></Input_1>
            <Input_2><enable>OFF</enable><state>OFF</state></Input_2>
            <Input_3><enable>OFF</enable><state>OFF</state></Input_3>
            <Input_4><enable>OFF</enable><state>OFF</state></Input_4>
            <Input_5><enable>OFF</enable><state>OFF</state></Input_5>
            <Input_6><enable>OFF</enable><state>OFF</state></Input_6>
            <Input_7><enable>OFF</enable><state>OFF</state></Input_7>
            <Input_8><enable>OFF</enable><state>OFF</state></Input_8>
            <Input_9><enable>OFF</enable><state>OFF</state></Input_9>
            <Input_10><enable>OFF</enable><state>OFF</state></Input_10>
            <Input_11><enable>OFF</enable><state>OFF</state></Input_11>
        </InputTrigger>
        <OutputTrigger>
            <Output_0><enable>OFF</enable><state>OFF</state></Output_0>
            <Output_1><enable>OFF</enable><state>OFF</state></Output_1>
            <Output_2><enable>OFF</enable><state>OFF</state></Output_2>
            <Output_3><enable>OFF</enable><state>OFF</state></Output_3>
```

```
                        <Output_4><enable>OFF</enable><state>OFF</state></Output_4>
                        <Output_5><enable>OFF</enable><state>OFF</state></Output_5>
                        <Output_6><enable>OFF</enable><state>OFF</state></Output_6>
                        <Output_7><enable>OFF</enable><state>OFF</state></Output_7>
                        <Output_8><enable>OFF</enable><state>OFF</state></Output_8>
                        <Output_9><enable>OFF</enable><state>OFF</state></Output_9>
                        <Output_10><enable>OFF</enable><state>OFF</state></Output_10>
                        <Output_11><enable>OFF</enable><state>OFF</state></Output_11>
                    </OutputTrigger>
                    <SystemTrigger>
                        <Load_Control>OFF</Load_Control>
                        <Interval_Timer>OFF</Interval_Timer>
                        <Cold_Start>OFF</Cold_Start>
                        <Warm_Start>OFF</Warm_Start>
                    </SystemTrigger>
                    <MaxCounterValue></MaxCounterValue>
                    <Interval></Interval>
                    <Timer_Cron></Timer_Cron>
                    <AlarmEnable>
                        <Mail_enable>OFF</Mail_enable>
                        <SNMP_Trap_enable>OFF</SNMP_Trap_enable>
                        <UDP_Client_enable>OFF</UDP_Client_enable>
                        <Send_special_alarm_to_pending_TCP_connections>OFF</
Send_special_alarm_to_pending_TCP_connections>
                        <TCP_Client_enable>OFF</TCP_Client_enable>
                        <Syslog_Messages_enable>OFF</Syslog_Messages_enable>
                        <FTP_Client_enable>OFF</FTP_Client_enable>
                    </AlarmEnable>
                    <Mail>
                        <E-Mail-Addr></E-Mail-Addr>
                        <Subject></Subject>
                        <Mailtext></Mailtext>
                        <Alarm_clear_subject></Alarm_clear_subject>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Mail>
                    <SNMP>
                        <Manager_IP></Manager_IP>
                        <TrapText></TrapText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </SNMP>
                    <UDP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <UdpText></UdpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </UDP>
                    <TCP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <TcpText></TcpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </TCP>
```

```
        <Syslog>
            <IP_Addr></IP_Addr>
            <Port>514</Port>
            <SyslogText></SyslogText>
            <Alarm_clear_text></Alarm_clear_text>
        </Syslog>
        <FTP>
            <FtpLocalDataPort>AUTO</FtpLocalDataPort>
            <FileName></FileName>
            <FtpAlarmText></FtpAlarmText>
            <Alarm_clear_text></Alarm_clear_text>
            <Options>
                <STORE>OFF</STORE>
                <APPEND>ON</APPEND>
            </Options>
        </FTP>
    </Alarm6>
    <Alarm7>
        <InputTrigger>
            <Input_0><enable>OFF</enable><state>OFF</state></Input_0>
            <Input_1><enable>OFF</enable><state>OFF</state></Input_1>
            <Input_2><enable>OFF</enable><state>OFF</state></Input_2>
            <Input_3><enable>OFF</enable><state>OFF</state></Input_3>
            <Input_4><enable>OFF</enable><state>OFF</state></Input_4>
            <Input_5><enable>OFF</enable><state>OFF</state></Input_5>
            <Input_6><enable>OFF</enable><state>OFF</state></Input_6>
            <Input_7><enable>OFF</enable><state>OFF</state></Input_7>
            <Input_8><enable>OFF</enable><state>OFF</state></Input_8>
            <Input_9><enable>OFF</enable><state>OFF</state></Input_9>
            <Input_10><enable>OFF</enable><state>OFF</state></Input_10>
            <Input_11><enable>OFF</enable><state>OFF</state></Input_11>
        </InputTrigger>
        <OutputTrigger>
            <Output_0><enable>OFF</enable><state>OFF</state></Output_0>
            <Output_1><enable>OFF</enable><state>OFF</state></Output_1>
            <Output_2><enable>OFF</enable><state>OFF</state></Output_2>
            <Output_3><enable>OFF</enable><state>OFF</state></Output_3>
            <Output_4><enable>OFF</enable><state>OFF</state></Output_4>
            <Output_5><enable>OFF</enable><state>OFF</state></Output_5>
            <Output_6><enable>OFF</enable><state>OFF</state></Output_6>
            <Output_7><enable>OFF</enable><state>OFF</state></Output_7>
            <Output_8><enable>OFF</enable><state>OFF</state></Output_8>
            <Output_9><enable>OFF</enable><state>OFF</state></Output_9>
            <Output_10><enable>OFF</enable><state>OFF</state></Output_10>
            <Output_11><enable>OFF</enable><state>OFF</state></Output_11>
        </OutputTrigger>
        <SystemTrigger>
            <Load_Control>OFF</Load_Control>
            <Interval_Timer>OFF</Interval_Timer>
            <Cold_Start>OFF</Cold_Start>
            <Warm_Start>OFF</Warm_Start>
        </SystemTrigger>
```

Copyright © 2024 Vizrt

```
                    <MaxCounterValue></MaxCounterValue>
                    <Interval></Interval>
                    <Timer_Cron></Timer_Cron>
                    <AlarmEnable>
                        <Mail_enable>OFF</Mail_enable>
                        <SNMP_Trap_enable>OFF</SNMP_Trap_enable>
                        <UDP_Client_enable>OFF</UDP_Client_enable>
                        <Send_special_alarm_to_pending_TCP_connections>OFF</
Send_special_alarm_to_pending_TCP_connections>
                        <TCP_Client_enable>OFF</TCP_Client_enable>
                        <Syslog_Messages_enable>OFF</Syslog_Messages_enable>
                        <FTP_Client_enable>OFF</FTP_Client_enable>
                    </AlarmEnable>
                    <Mail>
                        <E-Mail-Addr></E-Mail-Addr>
                        <Subject></Subject>
                        <Mailtext></Mailtext>
                        <Alarm_clear_subject></Alarm_clear_subject>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Mail>
                    <SNMP>
                        <Manager_IP></Manager_IP>
                        <TrapText></TrapText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </SNMP>
                    <UDP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <UdpText></UdpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </UDP>
                    <TCP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <TcpText></TcpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </TCP>
                    <Syslog>
                        <IP_Addr></IP_Addr>
                        <Port>514</Port>
                        <SyslogText></SyslogText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Syslog>
                    <FTP>
                        <FtpLocalDataPort>AUTO</FtpLocalDataPort>
                        <FileName></FileName>
                        <FtpAlarmText></FtpAlarmText>
                        <Alarm_clear_text></Alarm_clear_text>
                        <Options>
                            <STORE>OFF</STORE>
                            <APPEND>ON</APPEND>
                        </Options>
```

Copyright © 2024 Vizrt

```
                    </FTP>
                </Alarm7>
                <Alarm8>
                    <InputTrigger>
                        <Input_0><enable>OFF</enable><state>OFF</state></Input_0>
                        <Input_1><enable>OFF</enable><state>OFF</state></Input_1>
                        <Input_2><enable>OFF</enable><state>OFF</state></Input_2>
                        <Input_3><enable>OFF</enable><state>OFF</state></Input_3>
                        <Input_4><enable>OFF</enable><state>OFF</state></Input_4>
                        <Input_5><enable>OFF</enable><state>OFF</state></Input_5>
                        <Input_6><enable>OFF</enable><state>OFF</state></Input_6>
                        <Input_7><enable>OFF</enable><state>OFF</state></Input_7>
                        <Input_8><enable>OFF</enable><state>OFF</state></Input_8>
                        <Input_9><enable>OFF</enable><state>OFF</state></Input_9>
                        <Input_10><enable>OFF</enable><state>OFF</state></Input_10>
                        <Input_11><enable>OFF</enable><state>OFF</state></Input_11>
                    </InputTrigger>
                    <OutputTrigger>
                        <Output_0><enable>OFF</enable><state>OFF</state></Output_0>
                        <Output_1><enable>OFF</enable><state>OFF</state></Output_1>
                        <Output_2><enable>OFF</enable><state>OFF</state></Output_2>
                        <Output_3><enable>OFF</enable><state>OFF</state></Output_3>
                        <Output_4><enable>OFF</enable><state>OFF</state></Output_4>
                        <Output_5><enable>OFF</enable><state>OFF</state></Output_5>
                        <Output_6><enable>OFF</enable><state>OFF</state></Output_6>
                        <Output_7><enable>OFF</enable><state>OFF</state></Output_7>
                        <Output_8><enable>OFF</enable><state>OFF</state></Output_8>
                        <Output_9><enable>OFF</enable><state>OFF</state></Output_9>
                        <Output_10><enable>OFF</enable><state>OFF</state></Output_10>
                        <Output_11><enable>OFF</enable><state>OFF</state></Output_11>
                    </OutputTrigger>
                    <SystemTrigger>
                        <Load_Control>OFF</Load_Control>
                        <Interval_Timer>OFF</Interval_Timer>
                        <Cold_Start>OFF</Cold_Start>
                        <Warm_Start>OFF</Warm_Start>
                    </SystemTrigger>
                    <MaxCounterValue></MaxCounterValue>
                    <Interval></Interval>
                    <Timer_Cron></Timer_Cron>
                    <AlarmEnable>
                        <Mail_enable>OFF</Mail_enable>
                        <SNMP_Trap_enable>OFF</SNMP_Trap_enable>
                        <UDP_Client_enable>OFF</UDP_Client_enable>
                        <Send_special_alarm_to_pending_TCP_connections>OFF</
Send_special_alarm_to_pending_TCP_connections>
                        <TCP_Client_enable>OFF</TCP_Client_enable>
                        <Syslog_Messages_enable>OFF</Syslog_Messages_enable>
                        <FTP_Client_enable>OFF</FTP_Client_enable>
                    </AlarmEnable>
                    <Mail>
                        <E-Mail-Addr></E-Mail-Addr>
```

Copyright © 2024 Vizrt

```
            <Subject></Subject>
            <Mailtext></Mailtext>
            <Alarm_clear_subject></Alarm_clear_subject>
            <Alarm_clear_text></Alarm_clear_text>
        </Mail>
        <SNMP>
            <Manager_IP></Manager_IP>
            <TrapText></TrapText>
            <Alarm_clear_text></Alarm_clear_text>
        </SNMP>
        <UDP>
            <IP_Addr></IP_Addr>
            <Port></Port>
            <UdpText></UdpText>
            <Alarm_clear_text></Alarm_clear_text>
        </UDP>
        <TCP>
            <IP_Addr></IP_Addr>
            <Port></Port>
            <TcpText></TcpText>
            <Alarm_clear_text></Alarm_clear_text>
        </TCP>
        <Syslog>
            <IP_Addr></IP_Addr>
            <Port>514</Port>
            <SyslogText></SyslogText>
            <Alarm_clear_text></Alarm_clear_text>
        </Syslog>
        <FTP>
            <FtpLocalDataPort>AUTO</FtpLocalDataPort>
            <FileName></FileName>
            <FtpAlarmText></FtpAlarmText>
            <Alarm_clear_text></Alarm_clear_text>
            <Options>
                <STORE>OFF</STORE>
                <APPEND>ON</APPEND>
            </Options>
        </FTP>
    </Alarm8>
    <Alarm9>
        <InputTrigger>
            <Input_0><enable>OFF</enable><state>OFF</state></Input_0>
            <Input_1><enable>OFF</enable><state>OFF</state></Input_1>
            <Input_2><enable>OFF</enable><state>OFF</state></Input_2>
            <Input_3><enable>OFF</enable><state>OFF</state></Input_3>
            <Input_4><enable>OFF</enable><state>OFF</state></Input_4>
            <Input_5><enable>OFF</enable><state>OFF</state></Input_5>
            <Input_6><enable>OFF</enable><state>OFF</state></Input_6>
            <Input_7><enable>OFF</enable><state>OFF</state></Input_7>
            <Input_8><enable>OFF</enable><state>OFF</state></Input_8>
            <Input_9><enable>OFF</enable><state>OFF</state></Input_9>
            <Input_10><enable>OFF</enable><state>OFF</state></Input_10>
```

Copyright © 2024 Vizrt

```
                        <Input_11><enable>OFF</enable><state>OFF</state></Input_11>
                    </InputTrigger>
                    <OutputTrigger>
                        <Output_0><enable>OFF</enable><state>OFF</state></Output_0>
                        <Output_1><enable>OFF</enable><state>OFF</state></Output_1>
                        <Output_2><enable>OFF</enable><state>OFF</state></Output_2>
                        <Output_3><enable>OFF</enable><state>OFF</state></Output_3>
                        <Output_4><enable>OFF</enable><state>OFF</state></Output_4>
                        <Output_5><enable>OFF</enable><state>OFF</state></Output_5>
                        <Output_6><enable>OFF</enable><state>OFF</state></Output_6>
                        <Output_7><enable>OFF</enable><state>OFF</state></Output_7>
                        <Output_8><enable>OFF</enable><state>OFF</state></Output_8>
                        <Output_9><enable>OFF</enable><state>OFF</state></Output_9>
                        <Output_10><enable>OFF</enable><state>OFF</state></Output_10>
                        <Output_11><enable>OFF</enable><state>OFF</state></Output_11>
                    </OutputTrigger>
                    <SystemTrigger>
                        <Load_Control>OFF</Load_Control>
                        <Interval_Timer>OFF</Interval_Timer>
                        <Cold_Start>OFF</Cold_Start>
                        <Warm_Start>OFF</Warm_Start>
                    </SystemTrigger>
                    <MaxCounterValue></MaxCounterValue>
                    <Interval></Interval>
                    <Timer_Cron></Timer_Cron>
                    <AlarmEnable>
                        <Mail_enable>OFF</Mail_enable>
                        <SNMP_Trap_enable>OFF</SNMP_Trap_enable>
                        <UDP_Client_enable>OFF</UDP_Client_enable>
                        <Send_special_alarm_to_pending_TCP_connections>OFF</
Send_special_alarm_to_pending_TCP_connections>
                        <TCP_Client_enable>OFF</TCP_Client_enable>
                        <Syslog_Messages_enable>OFF</Syslog_Messages_enable>
                        <FTP_Client_enable>OFF</FTP_Client_enable>
                    </AlarmEnable>
                    <Mail>
                        <E-Mail-Addr></E-Mail-Addr>
                        <Subject></Subject>
                        <Mailtext></Mailtext>
                        <Alarm_clear_subject></Alarm_clear_subject>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Mail>
                    <SNMP>
                        <Manager_IP></Manager_IP>
                        <TrapText></TrapText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </SNMP>
                    <UDP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <UdpText></UdpText>
                        <Alarm_clear_text></Alarm_clear_text>
```

Copyright © 2024 Vizrt

```
        </UDP>
        <TCP>
            <IP_Addr></IP_Addr>
            <Port></Port>
            <TcpText></TcpText>
            <Alarm_clear_text></Alarm_clear_text>
        </TCP>
        <Syslog>
            <IP_Addr></IP_Addr>
            <Port>514</Port>
            <SyslogText></SyslogText>
            <Alarm_clear_text></Alarm_clear_text>
        </Syslog>
        <FTP>
            <FtpLocalDataPort>AUTO</FtpLocalDataPort>
            <FileName></FileName>
            <FtpAlarmText></FtpAlarmText>
            <Alarm_clear_text></Alarm_clear_text>
            <Options>
                <STORE>OFF</STORE>
                <APPEND>ON</APPEND>
            </Options>
        </FTP>
    </Alarm9>
    <Alarm10>
        <InputTrigger>
            <Input_0><enable>OFF</enable><state>OFF</state></Input_0>
            <Input_1><enable>OFF</enable><state>OFF</state></Input_1>
            <Input_2><enable>OFF</enable><state>OFF</state></Input_2>
            <Input_3><enable>OFF</enable><state>OFF</state></Input_3>
            <Input_4><enable>OFF</enable><state>OFF</state></Input_4>
            <Input_5><enable>OFF</enable><state>OFF</state></Input_5>
            <Input_6><enable>OFF</enable><state>OFF</state></Input_6>
            <Input_7><enable>OFF</enable><state>OFF</state></Input_7>
            <Input_8><enable>OFF</enable><state>OFF</state></Input_8>
            <Input_9><enable>OFF</enable><state>OFF</state></Input_9>
            <Input_10><enable>OFF</enable><state>OFF</state></Input_10>
            <Input_11><enable>OFF</enable><state>OFF</state></Input_11>
        </InputTrigger>
        <OutputTrigger>
            <Output_0><enable>OFF</enable><state>OFF</state></Output_0>
            <Output_1><enable>OFF</enable><state>OFF</state></Output_1>
            <Output_2><enable>OFF</enable><state>OFF</state></Output_2>
            <Output_3><enable>OFF</enable><state>OFF</state></Output_3>
            <Output_4><enable>OFF</enable><state>OFF</state></Output_4>
            <Output_5><enable>OFF</enable><state>OFF</state></Output_5>
            <Output_6><enable>OFF</enable><state>OFF</state></Output_6>
            <Output_7><enable>OFF</enable><state>OFF</state></Output_7>
            <Output_8><enable>OFF</enable><state>OFF</state></Output_8>
            <Output_9><enable>OFF</enable><state>OFF</state></Output_9>
            <Output_10><enable>OFF</enable><state>OFF</state></Output_10>
            <Output_11><enable>OFF</enable><state>OFF</state></Output_11>
```

```
                    </OutputTrigger>
                    <SystemTrigger>
                        <Load_Control>OFF</Load_Control>
                        <Interval_Timer>OFF</Interval_Timer>
                        <Cold_Start>OFF</Cold_Start>
                        <Warm_Start>OFF</Warm_Start>
                    </SystemTrigger>
                    <MaxCounterValue></MaxCounterValue>
                    <Interval></Interval>
                    <Timer_Cron></Timer_Cron>
                    <AlarmEnable>
                        <Mail_enable>OFF</Mail_enable>
                        <SNMP_Trap_enable>OFF</SNMP_Trap_enable>
                        <UDP_Client_enable>OFF</UDP_Client_enable>
                        <Send_special_alarm_to_pending_TCP_connections>OFF</
Send_special_alarm_to_pending_TCP_connections>
                        <TCP_Client_enable>OFF</TCP_Client_enable>
                        <Syslog_Messages_enable>OFF</Syslog_Messages_enable>
                        <FTP_Client_enable>OFF</FTP_Client_enable>
                    </AlarmEnable>
                    <Mail>
                        <E-Mail-Addr></E-Mail-Addr>
                        <Subject></Subject>
                        <Mailtext></Mailtext>
                        <Alarm_clear_subject></Alarm_clear_subject>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Mail>
                    <SNMP>
                        <Manager_IP></Manager_IP>
                        <TrapText></TrapText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </SNMP>
                    <UDP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <UdpText></UdpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </UDP>
                    <TCP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <TcpText></TcpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </TCP>
                    <Syslog>
                        <IP_Addr></IP_Addr>
                        <Port>514</Port>
                        <SyslogText></SyslogText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Syslog>
                    <FTP>
                        <FtpLocalDataPort>AUTO</FtpLocalDataPort>
```

Copyright © 2024 Vizrt

```
            <FileName></FileName>
            <FtpAlarmText></FtpAlarmText>
            <Alarm_clear_text></Alarm_clear_text>
            <Options>
                <STORE>OFF</STORE>
                <APPEND>ON</APPEND>
            </Options>
        </FTP>
    </Alarm10>
    <Alarm11>
        <InputTrigger>
            <Input_0><enable>OFF</enable><state>OFF</state></Input_0>
            <Input_1><enable>OFF</enable><state>OFF</state></Input_1>
            <Input_2><enable>OFF</enable><state>OFF</state></Input_2>
            <Input_3><enable>OFF</enable><state>OFF</state></Input_3>
            <Input_4><enable>OFF</enable><state>OFF</state></Input_4>
            <Input_5><enable>OFF</enable><state>OFF</state></Input_5>
            <Input_6><enable>OFF</enable><state>OFF</state></Input_6>
            <Input_7><enable>OFF</enable><state>OFF</state></Input_7>
            <Input_8><enable>OFF</enable><state>OFF</state></Input_8>
            <Input_9><enable>OFF</enable><state>OFF</state></Input_9>
            <Input_10><enable>OFF</enable><state>OFF</state></Input_10>
            <Input_11><enable>OFF</enable><state>OFF</state></Input_11>
        </InputTrigger>
        <OutputTrigger>
            <Output_0><enable>OFF</enable><state>OFF</state></Output_0>
            <Output_1><enable>OFF</enable><state>OFF</state></Output_1>
            <Output_2><enable>OFF</enable><state>OFF</state></Output_2>
            <Output_3><enable>OFF</enable><state>OFF</state></Output_3>
            <Output_4><enable>OFF</enable><state>OFF</state></Output_4>
            <Output_5><enable>OFF</enable><state>OFF</state></Output_5>
            <Output_6><enable>OFF</enable><state>OFF</state></Output_6>
            <Output_7><enable>OFF</enable><state>OFF</state></Output_7>
            <Output_8><enable>OFF</enable><state>OFF</state></Output_8>
            <Output_9><enable>OFF</enable><state>OFF</state></Output_9>
            <Output_10><enable>OFF</enable><state>OFF</state></Output_10>
            <Output_11><enable>OFF</enable><state>OFF</state></Output_11>
        </OutputTrigger>
        <SystemTrigger>
            <Load_Control>OFF</Load_Control>
            <Interval_Timer>OFF</Interval_Timer>
            <Cold_Start>OFF</Cold_Start>
            <Warm_Start>OFF</Warm_Start>
        </SystemTrigger>
        <MaxCounterValue></MaxCounterValue>
        <Interval></Interval>
        <Timer_Cron></Timer_Cron>
        <AlarmEnable>
            <Mail_enable>OFF</Mail_enable>
            <SNMP_Trap_enable>OFF</SNMP_Trap_enable>
            <UDP_Client_enable>OFF</UDP_Client_enable>
```

Copyright © 2024 Vizrt

```
                        <Send_special_alarm_to_pending_TCP_connections>OFF</
Send_special_alarm_to_pending_TCP_connections>
                        <TCP_Client_enable>OFF</TCP_Client_enable>
                        <Syslog_Messages_enable>OFF</Syslog_Messages_enable>
                        <FTP_Client_enable>OFF</FTP_Client_enable>
                    </AlarmEnable>
                    <Mail>
                        <E-Mail-Addr></E-Mail-Addr>
                        <Subject></Subject>
                        <Mailtext></Mailtext>
                        <Alarm_clear_subject></Alarm_clear_subject>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Mail>
                    <SNMP>
                        <Manager_IP></Manager_IP>
                        <TrapText></TrapText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </SNMP>
                    <UDP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <UdpText></UdpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </UDP>
                    <TCP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <TcpText></TcpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </TCP>
                    <Syslog>
                        <IP_Addr></IP_Addr>
                        <Port>514</Port>
                        <SyslogText></SyslogText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Syslog>
                    <FTP>
                        <FtpLocalDataPort>AUTO</FtpLocalDataPort>
                        <FileName></FileName>
                        <FtpAlarmText></FtpAlarmText>
                        <Alarm_clear_text></Alarm_clear_text>
                        <Options>
                            <STORE>OFF</STORE>
                            <APPEND>ON</APPEND>
                        </Options>
                    </FTP>
                </Alarm11>
                <Alarm12>
                    <InputTrigger>
                        <Input_0><enable>OFF</enable><state>OFF</state></Input_0>
                        <Input_1><enable>OFF</enable><state>OFF</state></Input_1>
                        <Input_2><enable>OFF</enable><state>OFF</state></Input_2>
```

```
                        <Input_3><enable>OFF</enable><state>OFF</state></Input_3>
                        <Input_4><enable>OFF</enable><state>OFF</state></Input_4>
                        <Input_5><enable>OFF</enable><state>OFF</state></Input_5>
                        <Input_6><enable>OFF</enable><state>OFF</state></Input_6>
                        <Input_7><enable>OFF</enable><state>OFF</state></Input_7>
                        <Input_8><enable>OFF</enable><state>OFF</state></Input_8>
                        <Input_9><enable>OFF</enable><state>OFF</state></Input_9>
                        <Input_10><enable>OFF</enable><state>OFF</state></Input_10>
                        <Input_11><enable>OFF</enable><state>OFF</state></Input_11>
                    </InputTrigger>
                    <OutputTrigger>
                        <Output_0><enable>OFF</enable><state>OFF</state></Output_0>
                        <Output_1><enable>OFF</enable><state>OFF</state></Output_1>
                        <Output_2><enable>OFF</enable><state>OFF</state></Output_2>
                        <Output_3><enable>OFF</enable><state>OFF</state></Output_3>
                        <Output_4><enable>OFF</enable><state>OFF</state></Output_4>
                        <Output_5><enable>OFF</enable><state>OFF</state></Output_5>
                        <Output_6><enable>OFF</enable><state>OFF</state></Output_6>
                        <Output_7><enable>OFF</enable><state>OFF</state></Output_7>
                        <Output_8><enable>OFF</enable><state>OFF</state></Output_8>
                        <Output_9><enable>OFF</enable><state>OFF</state></Output_9>
                        <Output_10><enable>OFF</enable><state>OFF</state></Output_10>
                        <Output_11><enable>OFF</enable><state>OFF</state></Output_11>
                    </OutputTrigger>
                    <SystemTrigger>
                        <Load_Control>OFF</Load_Control>
                        <Interval_Timer>OFF</Interval_Timer>
                        <Cold_Start>OFF</Cold_Start>
                        <Warm_Start>OFF</Warm_Start>
                    </SystemTrigger>
                    <MaxCounterValue></MaxCounterValue>
                    <Interval></Interval>
                    <Timer_Cron></Timer_Cron>
                    <AlarmEnable>
                        <Mail_enable>OFF</Mail_enable>
                        <SNMP_Trap_enable>OFF</SNMP_Trap_enable>
                        <UDP_Client_enable>OFF</UDP_Client_enable>
                        <Send_special_alarm_to_pending_TCP_connections>OFF</
Send_special_alarm_to_pending_TCP_connections>
                        <TCP_Client_enable>OFF</TCP_Client_enable>
                        <Syslog_Messages_enable>OFF</Syslog_Messages_enable>
                        <FTP_Client_enable>OFF</FTP_Client_enable>
                    </AlarmEnable>
                    <Mail>
                        <E-Mail-Addr></E-Mail-Addr>
                        <Subject></Subject>
                        <Mailtext></Mailtext>
                        <Alarm_clear_subject></Alarm_clear_subject>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Mail>
                    <SNMP>
                        <Manager_IP></Manager_IP>
```

Copyright © 2024 Vizrt

```
                        <TrapText></TrapText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </SNMP>
                    <UDP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <UdpText></UdpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </UDP>
                    <TCP>
                        <IP_Addr></IP_Addr>
                        <Port></Port>
                        <TcpText></TcpText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </TCP>
                    <Syslog>
                        <IP_Addr></IP_Addr>
                        <Port>514</Port>
                        <SyslogText></SyslogText>
                        <Alarm_clear_text></Alarm_clear_text>
                    </Syslog>
                    <FTP>
                        <FtpLocalDataPort>AUTO</FtpLocalDataPort>
                        <FileName></FileName>
                        <FtpAlarmText></FtpAlarmText>
                        <Alarm_clear_text></Alarm_clear_text>
                        <Options>
                            <STORE>OFF</STORE>
                            <APPEND>ON</APPEND>
                        </Options>
                    </FTP>
                </Alarm12>
            </Alarm>
        </Device>
        <Ports>
            <Inputs>
                <Input0>
                    <Name>Input 0</Name>
                    <Text>Description Input 0</Text>
                    <Filter></Filter>
                    <InputMode>
                        <Short_Pulse_detection_enable>OFF</
Short_Pulse_detection_enable>
                        <Change_detection_enable>OFF</Change_detection_enable>
                        <Clear_counter_on_read>OFF</Clear_counter_on_read>
                        <Bidirectional_counter_enable>OFF</
Bidirectional_counter_enable>
                    </InputMode>
                    <Counter_Mode>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>ON</positive>
```

```
                        </PulsPolarity>
                        <InactiveTimeout>1</InactiveTimeout>
                    </Counter_Mode>
                </Input0>
                <Input1>
                    <Name>Input 1</Name>
                    <Text>Description Input 1</Text>
                    <Filter></Filter>
                    <InputMode>
                        <Short_Pulse_detection_enable>OFF</
Short_Pulse_detection_enable>
                        <Change_detection_enable>OFF</Change_detection_enable>
                        <Clear_counter_on_read>OFF</Clear_counter_on_read>
                        <Bidirectional_counter_enable>OFF</
Bidirectional_counter_enable>
                    </InputMode>
                    <Counter_Mode>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>ON</positive>
                        </PulsPolarity>
                        <InactiveTimeout>1</InactiveTimeout>
                    </Counter_Mode>
                </Input1>
                <Input2>
                    <Name>Input 2</Name>
                    <Text>Description Input 2</Text>
                    <Filter></Filter>
                    <InputMode>
                        <Short_Pulse_detection_enable>OFF</
Short_Pulse_detection_enable>
                        <Change_detection_enable>OFF</Change_detection_enable>
                        <Clear_counter_on_read>OFF</Clear_counter_on_read>
                        <Bidirectional_counter_enable>OFF</
Bidirectional_counter_enable>
                    </InputMode>
                    <Counter_Mode>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>OFF</positive>
                        </PulsPolarity>
                        <InactiveTimeout></InactiveTimeout>
                    </Counter_Mode>
                </Input2>
                <Input3>
                    <Name>Input 3</Name>
                    <Text>Description Input 3</Text>
                    <Filter></Filter>
                    <InputMode>
                        <Short_Pulse_detection_enable>OFF</
Short_Pulse_detection_enable>
                        <Change_detection_enable>OFF</Change_detection_enable>
```

```
                        <Clear_counter_on_read>OFF</Clear_counter_on_read>
                        <Bidirectional_counter_enable>OFF</
Bidirectional_counter_enable>
                    </InputMode>
                    <Counter_Mode>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>OFF</positive>
                        </PulsPolarity>
                        <InactiveTimeout></InactiveTimeout>
                    </Counter_Mode>
                </Input3>
                <Input4>
                    <Name>Input 4</Name>
                    <Text>Description Input 4</Text>
                    <Filter></Filter>
                    <InputMode>
                        <Change_detection_enable>OFF</Change_detection_enable>
                        <Clear_counter_on_read>OFF</Clear_counter_on_read>
                        <Bidirectional_counter_enable>OFF</
Bidirectional_counter_enable>
                    </InputMode>
                    <Counter_Mode>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>OFF</positive>
                        </PulsPolarity>
                        <InactiveTimeout></InactiveTimeout>
                    </Counter_Mode>
                </Input4>
                <Input5>
                    <Name>Input 5</Name>
                    <Text>Description Input 5</Text>
                    <Filter></Filter>
                    <InputMode>
                        <Change_detection_enable>OFF</Change_detection_enable>
                        <Clear_counter_on_read>OFF</Clear_counter_on_read>
                        <Bidirectional_counter_enable>OFF</
Bidirectional_counter_enable>
                    </InputMode>
                    <Counter_Mode>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>OFF</positive>
                        </PulsPolarity>
                        <InactiveTimeout></InactiveTimeout>
                    </Counter_Mode>
                </Input5>
                <Input6>
                    <Name>Input 6</Name>
                    <Text>Description Input 6</Text>
                    <Filter></Filter>
```

Copyright © 2024 Vizrt

```
                    <InputMode>
                        <Change_detection_enable>OFF</Change_detection_enable>
                        <Clear_counter_on_read>OFF</Clear_counter_on_read>
                        <Bidirectional_counter_enable>OFF</
Bidirectional_counter_enable>
                    </InputMode>
                    <Counter_Mode>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>OFF</positive>
                        </PulsPolarity>
                        <InactiveTimeout></InactiveTimeout>
                    </Counter_Mode>
                </Input6>
                <Input7>
                    <Name>Input 7</Name>
                    <Text>Description Input 7</Text>
                    <Filter></Filter>
                    <InputMode>
                        <Change_detection_enable>OFF</Change_detection_enable>
                        <Clear_counter_on_read>OFF</Clear_counter_on_read>
                        <Bidirectional_counter_enable>OFF</
Bidirectional_counter_enable>
                    </InputMode>
                    <Counter_Mode>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>OFF</positive>
                        </PulsPolarity>
                        <InactiveTimeout></InactiveTimeout>
                    </Counter_Mode>
                </Input7>
                <Input8>
                    <Name>Input 8</Name>
                    <Text>Description Input 8</Text>
                    <Filter></Filter>
                    <InputMode>
                        <Change_detection_enable>OFF</Change_detection_enable>
                        <Clear_counter_on_read>OFF</Clear_counter_on_read>
                        <Bidirectional_counter_enable>OFF</
Bidirectional_counter_enable>
                    </InputMode>
                    <Counter_Mode>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>OFF</positive>
                        </PulsPolarity>
                        <InactiveTimeout></InactiveTimeout>
                    </Counter_Mode>
                </Input8>
                <Input9>
                    <Name>Input 9</Name>
```

```
                    <Text>Description Input 9</Text>
                    <Filter></Filter>
                    <InputMode>
                        <Change_detection_enable>OFF</Change_detection_enable>
                        <Clear_counter_on_read>OFF</Clear_counter_on_read>
                        <Bidirectional_counter_enable>OFF</
Bidirectional_counter_enable>
                    </InputMode>
                    <Counter_Mode>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>OFF</positive>
                        </PulsPolarity>
                        <InactiveTimeout></InactiveTimeout>
                    </Counter_Mode>
                </Input9>
                <Input10>
                    <Name>Input 10</Name>
                    <Text>Description Input 10</Text>
                    <Filter></Filter>
                    <InputMode>
                        <Change_detection_enable>OFF</Change_detection_enable>
                        <Clear_counter_on_read>OFF</Clear_counter_on_read>
                        <Bidirectional_counter_enable>OFF</
Bidirectional_counter_enable>
                    </InputMode>
                    <Counter_Mode>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>OFF</positive>
                        </PulsPolarity>
                        <InactiveTimeout></InactiveTimeout>
                    </Counter_Mode>
                </Input10>
                <Input11>
                    <Name>Input 11</Name>
                    <Text>Description Input 11</Text>
                    <Filter></Filter>
                    <InputMode>
                        <Change_detection_enable>OFF</Change_detection_enable>
                        <Clear_counter_on_read>OFF</Clear_counter_on_read>
                        <Bidirectional_counter_enable>OFF</
Bidirectional_counter_enable>
                    </InputMode>
                    <Counter_Mode>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>OFF</positive>
                        </PulsPolarity>
                        <InactiveTimeout></InactiveTimeout>
                    </Counter_Mode>
                </Input11>
```

Copyright © 2024 Vizrt

```
            </Inputs>
            <Outputs>
                <Output0>
                    <Name>Output 0</Name>
                    <Text>Description Output 0</Text>
                    <GroupMode>
                        <_0_1_2_3_separated>ON</_0_1_2_3_separated>
                        <_0--1_2--3_parallel>OFF</_0--1_2--3_parallel>
                        <_0--1--2--3_parallel>OFF</_0--1--2--3_parallel>
                    </GroupMode>
                    <SafetyState>
                        <Safety_enable><enable>OFF</enable><state>OFF</state></
Safety_enable>
                    </SafetyState>
                    <Logic>
                        <InputMask>
                            <Input_0>OFF</Input_0>
                            <Input_1>OFF</Input_1>
                            <Input_2>OFF</Input_2>
                            <Input_3>OFF</Input_3>
                            <Input_4>OFF</Input_4>
                            <Input_5>OFF</Input_5>
                            <Input_6>OFF</Input_6>
                            <Input_7>OFF</Input_7>
                            <Input_8>OFF</Input_8>
                            <Input_9>OFF</Input_9>
                            <Input_10>OFF</Input_10>
                            <Input_11>OFF</Input_11>
                        </InputMask>
                        <InputInverter>
                            <Input_0>OFF</Input_0>
                            <Input_1>OFF</Input_1>
                            <Input_2>OFF</Input_2>
                            <Input_3>OFF</Input_3>
                            <Input_4>OFF</Input_4>
                            <Input_5>OFF</Input_5>
                            <Input_6>OFF</Input_6>
                            <Input_7>OFF</Input_7>
                            <Input_8>OFF</Input_8>
                            <Input_9>OFF</Input_9>
                            <Input_10>OFF</Input_10>
                            <Input_11>OFF</Input_11>
                        </InputInverter>
                        <LogicFunc>
                            <AND>ON</AND>
                            <OR>OFF</OR>
                        </LogicFunc>
                        <OutputInverter>
                            <Output_Inverter>OFF</Output_Inverter>
                        </OutputInverter>
                    </Logic>
                    <Puls>
```

```
                        <Duration>5000</Duration>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>ON</positive>
                        </PulsPolarity>
                    </Puls>
                </Output0>
                <Output1>
                    <Name>Output 1</Name>
                    <Text>Description Output 1</Text>
                    <GroupMode>
                        <_0_1_2_3_separated>ON</_0_1_2_3_separated>
                        <_0--1_2--3_parallel>OFF</_0--1_2--3_parallel>
                        <_0--1--2--3_parallel>OFF</_0--1--2--3_parallel>
                    </GroupMode>
                    <SafetyState>
                        <Safety_enable><enable>OFF</enable><state>OFF</state></
Safety_enable>
                    </SafetyState>
                    <Logic>
                        <InputMask>
                            <Input_0>OFF</Input_0>
                            <Input_1>OFF</Input_1>
                            <Input_2>OFF</Input_2>
                            <Input_3>OFF</Input_3>
                            <Input_4>OFF</Input_4>
                            <Input_5>OFF</Input_5>
                            <Input_6>OFF</Input_6>
                            <Input_7>OFF</Input_7>
                            <Input_8>OFF</Input_8>
                            <Input_9>OFF</Input_9>
                            <Input_10>OFF</Input_10>
                            <Input_11>OFF</Input_11>
                        </InputMask>
                        <InputInverter>
                            <Input_0>OFF</Input_0>
                            <Input_1>OFF</Input_1>
                            <Input_2>OFF</Input_2>
                            <Input_3>OFF</Input_3>
                            <Input_4>OFF</Input_4>
                            <Input_5>OFF</Input_5>
                            <Input_6>OFF</Input_6>
                            <Input_7>OFF</Input_7>
                            <Input_8>OFF</Input_8>
                            <Input_9>OFF</Input_9>
                            <Input_10>OFF</Input_10>
                            <Input_11>OFF</Input_11>
                        </InputInverter>
                        <LogicFunc>
                            <AND>ON</AND>
                            <OR>OFF</OR>
                        </LogicFunc>
```

Copyright © 2024 Vizrt

```
                        <OutputInverter>
                            <Output_Inverter>OFF</Output_Inverter>
                        </OutputInverter>
                    </Logic>
                    <Puls>
                        <Duration>2000</Duration>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>ON</positive>
                        </PulsPolarity>
                    </Puls>
                </Output1>
                <Output2>
                    <Name>Output 2</Name>
                    <Text>Description Output 2</Text>
                    <GroupMode>
                        <_0_1_2_3_separated>ON</_0_1_2_3_separated>
                        <_0--1_2--3_parallel>OFF</_0--1_2--3_parallel>
                        <_0--1--2--3_parallel>OFF</_0--1--2--3_parallel>
                    </GroupMode>
                    <SafetyState>
                        <Safety_enable><enable>OFF</enable><state>OFF</state></
Safety_enable>
                    </SafetyState>
                    <Logic>
                        <InputMask>
                            <Input_0>OFF</Input_0>
                            <Input_1>OFF</Input_1>
                            <Input_2>OFF</Input_2>
                            <Input_3>OFF</Input_3>
                            <Input_4>OFF</Input_4>
                            <Input_5>OFF</Input_5>
                            <Input_6>OFF</Input_6>
                            <Input_7>OFF</Input_7>
                            <Input_8>OFF</Input_8>
                            <Input_9>OFF</Input_9>
                            <Input_10>OFF</Input_10>
                            <Input_11>OFF</Input_11>
                        </InputMask>
                        <InputInverter>
                            <Input_0>OFF</Input_0>
                            <Input_1>OFF</Input_1>
                            <Input_2>OFF</Input_2>
                            <Input_3>OFF</Input_3>
                            <Input_4>OFF</Input_4>
                            <Input_5>OFF</Input_5>
                            <Input_6>OFF</Input_6>
                            <Input_7>OFF</Input_7>
                            <Input_8>OFF</Input_8>
                            <Input_9>OFF</Input_9>
                            <Input_10>OFF</Input_10>
                            <Input_11>OFF</Input_11>
```

Copyright © 2024 Vizrt

```
                        </InputInverter>
                        <LogicFunc>
                            <AND>ON</AND>
                            <OR>OFF</OR>
                        </LogicFunc>
                        <OutputInverter>
                            <Output_Inverter>OFF</Output_Inverter>
                        </OutputInverter>
                    </Logic>
                    <Puls>
                        <Duration>1000</Duration>
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>ON</positive>
                        </PulsPolarity>
                    </Puls>
                </Output2>
                <Output3>
                    <Name>Output 3</Name>
                    <Text>Description Output 3</Text>
                    <GroupMode>
                        <_0_1_2_3_separated>ON</_0_1_2_3_separated>
                        <_0--1_2--3_parallel>OFF</_0--1_2--3_parallel>
                        <_0--1--2--3_parallel>OFF</_0--1--2--3_parallel>
                    </GroupMode>
                    <SafetyState>
                        <Safety_enable><enable>OFF</enable><state>OFF</state></
Safety_enable>
                    </SafetyState>
                    <Logic>
                        <InputMask>
                            <Input_0>OFF</Input_0>
                            <Input_1>OFF</Input_1>
                            <Input_2>OFF</Input_2>
                            <Input_3>OFF</Input_3>
                            <Input_4>OFF</Input_4>
                            <Input_5>OFF</Input_5>
                            <Input_6>OFF</Input_6>
                            <Input_7>OFF</Input_7>
                            <Input_8>OFF</Input_8>
                            <Input_9>OFF</Input_9>
                            <Input_10>OFF</Input_10>
                            <Input_11>OFF</Input_11>
                        </InputMask>
                        <InputInverter>
                            <Input_0>OFF</Input_0>
                            <Input_1>OFF</Input_1>
                            <Input_2>OFF</Input_2>
                            <Input_3>OFF</Input_3>
                            <Input_4>OFF</Input_4>
                            <Input_5>OFF</Input_5>
                            <Input_6>OFF</Input_6>
```

Copyright © 2024 Vizrt

```
                        <Input_7>OFF</Input_7>
                        <Input_8>OFF</Input_8>
                        <Input_9>OFF</Input_9>
                        <Input_10>OFF</Input_10>
                        <Input_11>OFF</Input_11>
                    </InputInverter>
                    <LogicFunc>
                        <AND>ON</AND>
                        <OR>OFF</OR>
                    </LogicFunc>
                    <OutputInverter>
                        <Output_Inverter>OFF</Output_Inverter>
                    </OutputInverter>
                </Logic>
                <Puls>
                    <Duration>1000</Duration>
                    <PulsPolarity>
                        <negative>OFF</negative>
                        <positive>ON</positive>
                    </PulsPolarity>
                </Puls>
            </Output3>
            <Output4>
                <Name>Output 4</Name>
                <Text>Description Output 4</Text>
                <GroupMode>
                    <_4_5_6_7_separated>ON</_4_5_6_7_separated>
                    <_4--5_6--7_parallel>OFF</_4--5_6--7_parallel>
                    <_4--5--6--7_parallel>OFF</_4--5--6--7_parallel>
                </GroupMode>
                <SafetyState>
                    <Safety_enable><enable>OFF</enable><state>OFF</state></
Safety_enable>
                </SafetyState>
                <Logic>
                    <InputMask>
                        <Input_0>OFF</Input_0>
                        <Input_1>OFF</Input_1>
                        <Input_2>OFF</Input_2>
                        <Input_3>OFF</Input_3>
                        <Input_4>OFF</Input_4>
                        <Input_5>OFF</Input_5>
                        <Input_6>OFF</Input_6>
                        <Input_7>OFF</Input_7>
                        <Input_8>OFF</Input_8>
                        <Input_9>OFF</Input_9>
                        <Input_10>OFF</Input_10>
                        <Input_11>OFF</Input_11>
                    </InputMask>
                    <InputInverter>
                        <Input_0>OFF</Input_0>
                        <Input_1>OFF</Input_1>
```

```
                        <Input_2>OFF</Input_2>
                        <Input_3>OFF</Input_3>
                        <Input_4>OFF</Input_4>
                        <Input_5>OFF</Input_5>
                        <Input_6>OFF</Input_6>
                        <Input_7>OFF</Input_7>
                        <Input_8>OFF</Input_8>
                        <Input_9>OFF</Input_9>
                        <Input_10>OFF</Input_10>
                        <Input_11>OFF</Input_11>
                    </InputInverter>
                    <LogicFunc>
                        <AND>ON</AND>
                        <OR>OFF</OR>
                    </LogicFunc>
                    <OutputInverter>
                        <Output_Inverter>OFF</Output_Inverter>
                    </OutputInverter>
                </Logic>
                <Puls>
                    <Duration></Duration>
                    <PulsPolarity>
                        <negative>OFF</negative>
                        <positive>OFF</positive>
                    </PulsPolarity>
                </Puls>
            </Output4>
            <Output5>
                <Name>Output 5</Name>
                <Text>Description Output 5</Text>
                <GroupMode>
                    <_4_5_6_7_separated>ON</_4_5_6_7_separated>
                    <_4--5_6--7_parallel>OFF</_4--5_6--7_parallel>
                    <_4--5--6--7_parallel>OFF</_4--5--6--7_parallel>
                </GroupMode>
                <SafetyState>
                    <Safety_enable><enable>OFF</enable><state>OFF</state></
Safety_enable>
                </SafetyState>
                <Logic>
                    <InputMask>
                        <Input_0>OFF</Input_0>
                        <Input_1>OFF</Input_1>
                        <Input_2>OFF</Input_2>
                        <Input_3>OFF</Input_3>
                        <Input_4>OFF</Input_4>
                        <Input_5>OFF</Input_5>
                        <Input_6>OFF</Input_6>
                        <Input_7>OFF</Input_7>
                        <Input_8>OFF</Input_8>
                        <Input_9>OFF</Input_9>
                        <Input_10>OFF</Input_10>
```

```
                        <Input_11>OFF</Input_11>
                    </InputMask>
                    <InputInverter>
                        <Input_0>OFF</Input_0>
                        <Input_1>OFF</Input_1>
                        <Input_2>OFF</Input_2>
                        <Input_3>OFF</Input_3>
                        <Input_4>OFF</Input_4>
                        <Input_5>OFF</Input_5>
                        <Input_6>OFF</Input_6>
                        <Input_7>OFF</Input_7>
                        <Input_8>OFF</Input_8>
                        <Input_9>OFF</Input_9>
                        <Input_10>OFF</Input_10>
                        <Input_11>OFF</Input_11>
                    </InputInverter>
                    <LogicFunc>
                        <AND>ON</AND>
                        <OR>OFF</OR>
                    </LogicFunc>
                    <OutputInverter>
                        <Output_Inverter>OFF</Output_Inverter>
                    </OutputInverter>
                </Logic>
                <Puls>
                    <Duration></Duration>
                    <PulsPolarity>
                        <negative>OFF</negative>
                        <positive>OFF</positive>
                    </PulsPolarity>
                </Puls>
            </Output5>
            <Output6>
                <Name>Output 6</Name>
                <Text>Description Output 6</Text>
                <GroupMode>
                    <_4_5_6_7_separated>ON</_4_5_6_7_separated>
                    <_4--5_6--7_parallel>OFF</_4--5_6--7_parallel>
                    <_4--5--6--7_parallel>OFF</_4--5--6--7_parallel>
                </GroupMode>
                <SafetyState>
                    <Safety_enable><enable>OFF</enable><state>OFF</state></
Safety_enable>
                </SafetyState>
                <Logic>
                    <InputMask>
                        <Input_0>OFF</Input_0>
                        <Input_1>OFF</Input_1>
                        <Input_2>OFF</Input_2>
                        <Input_3>OFF</Input_3>
                        <Input_4>OFF</Input_4>
                        <Input_5>OFF</Input_5>
```

```
                        <Input_6>OFF</Input_6>
                        <Input_7>OFF</Input_7>
                        <Input_8>OFF</Input_8>
                        <Input_9>OFF</Input_9>
                        <Input_10>OFF</Input_10>
                        <Input_11>OFF</Input_11>
                    </InputMask>
                    <InputInverter>
                        <Input_0>OFF</Input_0>
                        <Input_1>OFF</Input_1>
                        <Input_2>OFF</Input_2>
                        <Input_3>OFF</Input_3>
                        <Input_4>OFF</Input_4>
                        <Input_5>OFF</Input_5>
                        <Input_6>OFF</Input_6>
                        <Input_7>OFF</Input_7>
                        <Input_8>OFF</Input_8>
                        <Input_9>OFF</Input_9>
                        <Input_10>OFF</Input_10>
                        <Input_11>OFF</Input_11>
                    </InputInverter>
                    <LogicFunc>
                        <AND>ON</AND>
                        <OR>OFF</OR>
                    </LogicFunc>
                    <OutputInverter>
                        <Output_Inverter>OFF</Output_Inverter>
                    </OutputInverter>
                </Logic>
                <Puls>
                    <Duration></Duration>
                    <PulsPolarity>
                        <negative>OFF</negative>
                        <positive>OFF</positive>
                    </PulsPolarity>
                </Puls>
            </Output6>
            <Output7>
                <Name>Output 7</Name>
                <Text>Description Output 7</Text>
                <GroupMode>
                    <_4_5_6_7_separated>ON</_4_5_6_7_separated>
                    <_4--5_6--7_parallel>OFF</_4--5_6--7_parallel>
                    <_4--5--6--7_parallel>OFF</_4--5--6--7_parallel>
                </GroupMode>
                <SafetyState>
                    <Safety_enable><enable>OFF</enable><state>OFF</state></
Safety_enable>
                </SafetyState>
                <Logic>
                    <InputMask>
                        <Input_0>OFF</Input_0>
```

```
                <Input_1>OFF</Input_1>
                <Input_2>OFF</Input_2>
                <Input_3>OFF</Input_3>
                <Input_4>OFF</Input_4>
                <Input_5>OFF</Input_5>
                <Input_6>OFF</Input_6>
                <Input_7>OFF</Input_7>
                <Input_8>OFF</Input_8>
                <Input_9>OFF</Input_9>
                <Input_10>OFF</Input_10>
                <Input_11>OFF</Input_11>
            </InputMask>
            <InputInverter>
                <Input_0>OFF</Input_0>
                <Input_1>OFF</Input_1>
                <Input_2>OFF</Input_2>
                <Input_3>OFF</Input_3>
                <Input_4>OFF</Input_4>
                <Input_5>OFF</Input_5>
                <Input_6>OFF</Input_6>
                <Input_7>OFF</Input_7>
                <Input_8>OFF</Input_8>
                <Input_9>OFF</Input_9>
                <Input_10>OFF</Input_10>
                <Input_11>OFF</Input_11>
            </InputInverter>
            <LogicFunc>
                <AND>ON</AND>
                <OR>OFF</OR>
            </LogicFunc>
            <OutputInverter>
                <Output_Inverter>OFF</Output_Inverter>
            </OutputInverter>
        </Logic>
        <Puls>
            <Duration></Duration>
            <PulsPolarity>
                <negative>OFF</negative>
                <positive>OFF</positive>
            </PulsPolarity>
        </Puls>
    </Output7>
    <Output8>
        <Name>Output 8</Name>
        <Text>Description Output 8</Text>
        <GroupMode>
            <_8_9_10_11_separated>ON</_8_9_10_11_separated>
            <_8--9_10--11_parallel>OFF</_8--9_10--11_parallel>
            <_8--9--10--11_parallel>OFF</_8--9--10--11_parallel>
        </GroupMode>
        <SafetyState>
```

```
                    <Safety_enable><enable>OFF</enable><state>OFF</state></
Safety_enable>
                </SafetyState>
                <Logic>
                    <InputMask>
                        <Input_0>OFF</Input_0>
                        <Input_1>OFF</Input_1>
                        <Input_2>OFF</Input_2>
                        <Input_3>OFF</Input_3>
                        <Input_4>OFF</Input_4>
                        <Input_5>OFF</Input_5>
                        <Input_6>OFF</Input_6>
                        <Input_7>OFF</Input_7>
                        <Input_8>OFF</Input_8>
                        <Input_9>OFF</Input_9>
                        <Input_10>OFF</Input_10>
                        <Input_11>OFF</Input_11>
                    </InputMask>
                    <InputInverter>
                        <Input_0>OFF</Input_0>
                        <Input_1>OFF</Input_1>
                        <Input_2>OFF</Input_2>
                        <Input_3>OFF</Input_3>
                        <Input_4>OFF</Input_4>
                        <Input_5>OFF</Input_5>
                        <Input_6>OFF</Input_6>
                        <Input_7>OFF</Input_7>
                        <Input_8>OFF</Input_8>
                        <Input_9>OFF</Input_9>
                        <Input_10>OFF</Input_10>
                        <Input_11>OFF</Input_11>
                    </InputInverter>
                    <LogicFunc>
                        <AND>ON</AND>
                        <OR>OFF</OR>
                    </LogicFunc>
                    <OutputInverter>
                        <Output_Inverter>OFF</Output_Inverter>
                    </OutputInverter>
                </Logic>
                <Puls>
                    <Duration></Duration>
                    <PulsPolarity>
                        <negative>OFF</negative>
                        <positive>OFF</positive>
                    </PulsPolarity>
                </Puls>
            </Output8>
            <Output9>
                <Name>Output 9</Name>
                <Text>Description Output 9</Text>
                <GroupMode>
```

```
                    <_8_9_10_11_separated>ON</_8_9_10_11_separated>
                    <_8--9_10--11_parallel>OFF</_8--9_10--11_parallel>
                    <_8--9--10--11_parallel>OFF</_8--9--10--11_parallel>
                </GroupMode>
                <SafetyState>
                    <Safety_enable><enable>OFF</enable><state>OFF</state></
Safety_enable>
                </SafetyState>
                <Logic>
                    <InputMask>
                        <Input_0>OFF</Input_0>
                        <Input_1>OFF</Input_1>
                        <Input_2>OFF</Input_2>
                        <Input_3>OFF</Input_3>
                        <Input_4>OFF</Input_4>
                        <Input_5>OFF</Input_5>
                        <Input_6>OFF</Input_6>
                        <Input_7>OFF</Input_7>
                        <Input_8>OFF</Input_8>
                        <Input_9>OFF</Input_9>
                        <Input_10>OFF</Input_10>
                        <Input_11>OFF</Input_11>
                    </InputMask>
                    <InputInverter>
                        <Input_0>OFF</Input_0>
                        <Input_1>OFF</Input_1>
                        <Input_2>OFF</Input_2>
                        <Input_3>OFF</Input_3>
                        <Input_4>OFF</Input_4>
                        <Input_5>OFF</Input_5>
                        <Input_6>OFF</Input_6>
                        <Input_7>OFF</Input_7>
                        <Input_8>OFF</Input_8>
                        <Input_9>OFF</Input_9>
                        <Input_10>OFF</Input_10>
                        <Input_11>OFF</Input_11>
                    </InputInverter>
                    <LogicFunc>
                        <AND>ON</AND>
                        <OR>OFF</OR>
                    </LogicFunc>
                    <OutputInverter>
                        <Output_Inverter>OFF</Output_Inverter>
                    </OutputInverter>
                </Logic>
                <Puls>
                    <Duration></Duration>
                    <PulsPolarity>
                        <negative>OFF</negative>
                        <positive>OFF</positive>
                    </PulsPolarity>
                </Puls>
```

```
                </Output9>
                <Output10>
                    <Name>Output 10</Name>
                    <Text>Description Output 10</Text>
                    <GroupMode>
                        <_8_9_10_11_separated>ON</_8_9_10_11_separated>
                        <_8--9_10--11_parallel>OFF</_8--9_10--11_parallel>
                        <_8--9--10--11_parallel>OFF</_8--9--10--11_parallel>
                    </GroupMode>
                    <SafetyState>
                        <Safety_enable><enable>OFF</enable><state>OFF</state></
Safety_enable>
                    </SafetyState>
                    <Logic>
                        <InputMask>
                            <Input_0>OFF</Input_0>
                            <Input_1>OFF</Input_1>
                            <Input_2>OFF</Input_2>
                            <Input_3>OFF</Input_3>
                            <Input_4>OFF</Input_4>
                            <Input_5>OFF</Input_5>
                            <Input_6>OFF</Input_6>
                            <Input_7>OFF</Input_7>
                            <Input_8>OFF</Input_8>
                            <Input_9>OFF</Input_9>
                            <Input_10>OFF</Input_10>
                            <Input_11>OFF</Input_11>
                        </InputMask>
                        <InputInverter>
                            <Input_0>OFF</Input_0>
                            <Input_1>OFF</Input_1>
                            <Input_2>OFF</Input_2>
                            <Input_3>OFF</Input_3>
                            <Input_4>OFF</Input_4>
                            <Input_5>OFF</Input_5>
                            <Input_6>OFF</Input_6>
                            <Input_7>OFF</Input_7>
                            <Input_8>OFF</Input_8>
                            <Input_9>OFF</Input_9>
                            <Input_10>OFF</Input_10>
                            <Input_11>OFF</Input_11>
                        </InputInverter>
                        <LogicFunc>
                            <AND>ON</AND>
                            <OR>OFF</OR>
                        </LogicFunc>
                        <OutputInverter>
                            <Output_Inverter>OFF</Output_Inverter>
                        </OutputInverter>
                    </Logic>
                    <Puls>
                        <Duration></Duration>
```

Copyright © 2024 Vizrt

```
                        <PulsPolarity>
                            <negative>OFF</negative>
                            <positive>OFF</positive>
                        </PulsPolarity>
                    </Puls>
                </Output10>
                <Output11>
                    <Name>Output 11</Name>
                    <Text>Description Output 11</Text>
                    <GroupMode>
                        <_8_9_10_11_separated>ON</_8_9_10_11_separated>
                        <_8--9_10--11_parallel>OFF</_8--9_10--11_parallel>
                        <_8--9--10--11_parallel>OFF</_8--9--10--11_parallel>
                    </GroupMode>
                    <SafetyState>
                        <Safety_enable><enable>OFF</enable><state>OFF</state></
Safety_enable>
                    </SafetyState>
                    <Logic>
                        <InputMask>
                            <Input_0>OFF</Input_0>
                            <Input_1>OFF</Input_1>
                            <Input_2>OFF</Input_2>
                            <Input_3>OFF</Input_3>
                            <Input_4>OFF</Input_4>
                            <Input_5>OFF</Input_5>
                            <Input_6>OFF</Input_6>
                            <Input_7>OFF</Input_7>
                            <Input_8>OFF</Input_8>
                            <Input_9>OFF</Input_9>
                            <Input_10>OFF</Input_10>
                            <Input_11>OFF</Input_11>
                        </InputMask>
                        <InputInverter>
                            <Input_0>OFF</Input_0>
                            <Input_1>OFF</Input_1>
                            <Input_2>OFF</Input_2>
                            <Input_3>OFF</Input_3>
                            <Input_4>OFF</Input_4>
                            <Input_5>OFF</Input_5>
                            <Input_6>OFF</Input_6>
                            <Input_7>OFF</Input_7>
                            <Input_8>OFF</Input_8>
                            <Input_9>OFF</Input_9>
                            <Input_10>OFF</Input_10>
                            <Input_11>OFF</Input_11>
                        </InputInverter>
                        <LogicFunc>
                            <AND>ON</AND>
                            <OR>OFF</OR>
                        </LogicFunc>
                        <OutputInverter>
```

Copyright © 2024 Vizrt

```
                        <Output_Inverter>OFF</Output_Inverter>
                    </OutputInverter>
                </Logic>
                <Puls>
                    <Duration></Duration>
                    <PulsPolarity>
                        <negative>OFF</negative>
                        <positive>OFF</positive>
                    </PulsPolarity>
                </Puls>
            </Output11>
        </Outputs>
    </Ports>
    <Manufacturer>
        <Name>Wiesemann & Theis GmbH</Name>
        <Address>Porschstr. 12<br>
42279 Wuppertal<br>
Germany</Address>
        <Support_Hotline>+49-(0)202-2680-0</Support_Hotline>
        <Internet>www.wut.de</Internet>
        <DeviceTyp>Web-IO Digital 12xIn, 12xOut</DeviceTyp>
        <DeviceNo>#57630 TB 900571</DeviceNo>
        <Logo>logo.gif</Logo>
    </Manufacturer>
</Config>
</io-Digital12x12.3>
```

13.3 Jupiter Systems PixelNet Video Wall Control

This section covers the following topics:

- Introduction
- Configuration
  - AVAutomation
  - Template Editor
  - Configuration File
- Implementation
  - Connection
- Requests to Implement
  - To Use LoadLayoutEx
  - To Use ActivateWall and LoadLayout
  - To Use LoadLayoutOnWall
  - To Use ActivateWall with LoadLayoutOnWall

13.3.1 Introduction

The protocol is text based commands and responses sent on TCP/IP. Text encoding is UTF-8. The protocol is synchronous - i.e. a command (request) shall always result in a response, and a new command shall not be sent until the response to the previous command has been received.

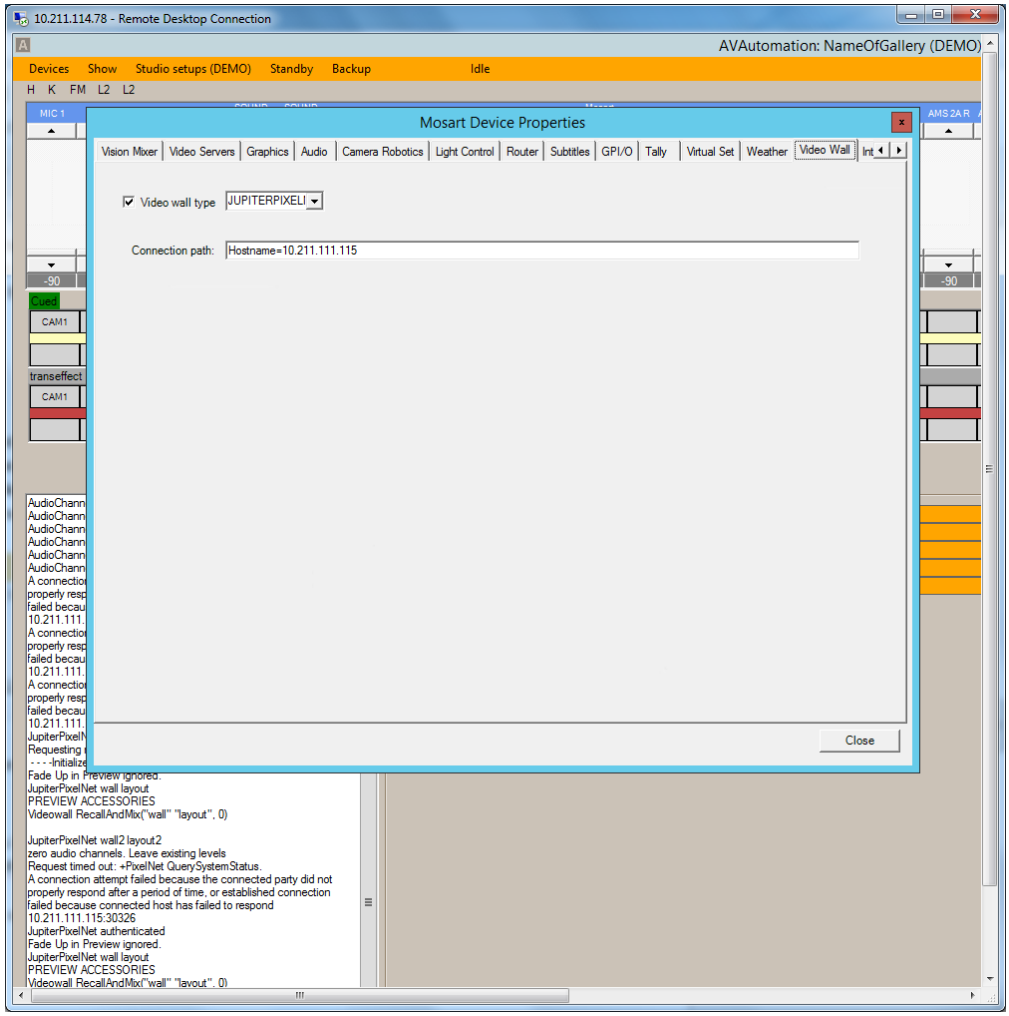
13.3.2 Configuration

AVAutomation

The router protocol is named JUPITERPIXELNET. Configuring the PixelNet video wall interface requires the following parameters:

- IP address of the PixelNet Domain Controller
- Port number (default is set to 30326)

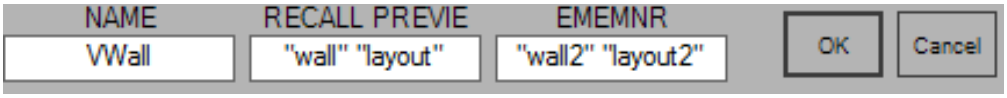
Copyright © 2024 Vizrt



Template Editor

In the template editor, enable Video Wall. In Device Properties, enter wall and layout (in double quotes) in that order, space separated, in both 'RECALL PREVIE' and EMEMNR:

Implementation may be based on the WATCHOUT protocol implementation.



Configuration File

The PixelNet configuration file is located in the program-folder under *\Mosart Medialab\Mosart Server\ConfigurationFiles\VideoWallJupiterPixelNet.xml*. To use it, copy it to c:\channeltemplates, and append

;Config=VideoWallJupiterPixelNet.xml

to the connection string.

The following configurations are available:

Setting	Description	Default
PingInterval	The amount of time (in ms) to wait before ping after connect, ping or RecallAndMix	10000 (10s)
Port	The port number of the PDC computer	30326
ReconnectInterval	The amount of time (in ms) to wait before reconnect after unsuccessful connect or lost connection	10000 (10s)
Server	The DNS name of the PDC computer	None
TimeoutActivateWall	The timeout (in ms) for the ActivateWall request	2000 (2s)
TimeoutLoadLayout	The timeout (in ms) for the LoadLayout request	2000 (2s)
TimeoutLoadLayoutEx	The timeout (in ms) for the LoadLayoutEx request	2000 (2s)
TimeoutLoadLayoutOnWall	The timeout (in ms) for the LoadLayoutOnWall request	2000 (2s)
TimeoutQuerySystemStatus	The timeout (in ms) for the QuerySystemStatus request	2000 (2s)
UseActivateWall	Whether (True/False) an ActivateWall request is sent before the LoadLayoutOnWall request. Has no effect unless UseLoadLayout is False.	False
UseDispatcherForRecallAndMixes	Whether (True/False) the RecallAndMix requests are executed in the dispatcher, i.e. in the same Thread as (re-)connects and pings.	False
UseLoadLayout	Whether (True/False) LoadLayout requests are sent instead of LoadLayoutEx requests	False
UseLoadLayoutOnWall	Whether (True/False) LoadLayoutOnWall requests are sent instead of LoadLayoutEx requests	False

Copyright © 2024 Vizrt

WallIsFirstArgument	Whether (True/False) the Wall argument of the LoadLayoutOnWall request is the first Argument	False
WallIsInt	Whether (True/False) the Wall argument of the LoadLayoutOnWall request is an integer	False

13.3.3 Implementation

Connection

- PixelNet uses the PixelNet Domain Control (PDC) protocol. PDC protocol is developed on top of the Remote Method Call (RMC) inter-process communication protocol, implemented over a TCP/IP connection.
- The Watchout video wall driver may serve as a model for this driver.
- The protocol can be tested over a Telnet session.
- Port number to connect to is 30326.

13.3.4 Requests to Implement

Five requests are implemented:

- ActivateWall (used only if UseLoadLayout is True, or if both UseActivateWall and UseLoadLayoutOnWall are True)
- LoadLayout (used only if UseLoadLayout is True)
- LoadLayoutEx (used if both UseLoadLayout and UseLoadLayoutOnWall are False)
- LoadLayoutOnWall (used only if UseLoadLayout is False, and UseLoadLayoutOnWall is True)
- QuerySystemStatus (used for (internal) Ping)

⚠ **Note:** Although LoadLayoutOnWall will likely not work, it was included since version 1.5 indicated that this was the request that should be utilized.

To Use LoadLayoutEx

Leave all Use... settings at their default False.

To Use ActivateWall and LoadLayout

Set UseLoadLayout = True.

To Use LoadLayoutOnWall

Set

Copyright © 2024 Vizrt

- UseLoadLayout = False
- UseLoadLayoutOnWall = True
- WallIsFirstArgument = True
- WallIsInt = True.

Note: LoadLayoutOnWall may not work.

To Use ActivateWall with LoadLayoutOnWall

- UseActivateWall = True.

Note: ActivateWall with LoadLayoutOnWall may not work.

AVAutomation Implementation

The following methods declared in MMVideoWall must be implemented:

Public MustOverride Function Connect(ByVal strConnection As String) As String Implements IConnect.Connect

Public MustOverride Function Disconnect() As String Implements IConnect.Disconnect

Public MustOverride Function Ping() As String Implements IConnect.Ping

Public MustOverride Sub RecallAndMix(ByVal presetName As String, ByVal transrate As Integer)

Note: The final method in the list above, RecallAndMix, calls LoadLayoutOnWall.

Copyright © 2024 Vizrt

13.4 Mosart Remote Control REST API

- Introduction
- Working with the REST API
  - Prerequisites/Conditions
  - Swagger Documentation
  - Command Descriptions
  - Authentication

13.4.1 Introduction

The Mosart Remote Control Service (RCS) provides a REST API for controlling Viz Mosart rundowns using REST. This interface in effect duplicates some GPI/O calls (0-4) for rundown operations.

13.4.2 Working with the REST API

Prerequisites/Conditions

- The descriptions below of the command assume that RCS and the user work on the same machine and its name is *localhost*.
- The API hosted by RCS and configured to run on port *55167*. This port can not be configured.
- Employs API key-based authentication where the key needs to be passed in a HTTP header called "X-API-key". See Mosart Remote Control REST API below.
- All routes under "/api/v1/command" are protected.
- All commands of this API are accessible only when RCS is running.

Swagger Documentation

- The Swagger documentation that describes every command can be found at http://localhost:55167/swagger/.

Command Descriptions

- Reload rundown
- Start/Continue rundown
- Start rundown from top
- Enable/disable rehearsal mode
- Take a template
- Trigger a DirectTake template

Copyright © 2024 Vizrt

Reload rundown

The reload endpoint allows you to reload the current rundown. It will trigger a rundown reload event. I.e. the timeline will be stopped and the current rundown will be reloaded from the NRCS cache. No additional parameters.

GET api/v1/command/reload

Start/Continue Rundown

Start or continue the rundown. You can use some optional parameters - see the swagger documentation.

GET api/v1/command/start-continue

Start Rundown from Top

Restart the rundown from the first story. No additional parameters.

GET api/v1/command/start-from-top

Enable/disable Rehearsal Mode

Enable/disable/toggle rehearsal mode:

GET api/v1/command/rehearsal-mode?state=true
GET api/v1/command/rehearsal-mode?state=false
GET api/v1/command/rehearsal-mode

Take a Template

Takes the template by its type and variant (name).

GET api/v1/command/template?type={template-type}&variant={variant}

For example,
GET api/v1/command/template?type=Camera&variant=CAM1

Here *Camera* and *CAM1* are the actual type and name of the desired template.

Trigger a DirectTake Template

Triger a given DirectTake template. You must provide the number of desired template.

GET api/v1/command/directtake?number={number}

an

Authentication

All commands require the special API key as a sign that the action has been authorized. This key should be passed in HTTP header called `X-API-key`.

Copyright © 2024 Vizrt

Remote Control Dispatcher Service (RCS) configuration file

The RCS configuration file contains a new setting `RestApi`. It's value can be randomly generated to be robust and sustainable against brute-force attack. Each REST call will compare the passed in header key `X-API-key` with the value from the `RestApi` setting first.

%ProgramData%\Mosart Medialab\ConfigurationFiles\RemoteDispatcherServiceConfig.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<RemoteServiceConfig>
  <Properties>
    ...
    <item name="RestApiKey" value="test-api-key" />
```

```
curl -X GET "http://localhost:55167/api/v1/command/reload" -H  "accept: */*" -H  "X-API-Key: test-api-key"
```

Important

Info: All the above commands will impact the live show on your Mosart Server.

Copyright © 2024 Vizrt

13.5 Named Overlay Graphics

Named CGs is a functionality allowing CGs (or overlay graphics) to be fired from within a template or via a control command (keyboard shortcut). All named CGs are placed within the file *NamedOverlayGraphics.xml*.

This section contains:

- Named CGs (Named Overlay Graphics)
- NCS Placeholders
- Named CG Actions
- Required Fields in XML
- Adding the Named CG to a Template
- Obtaining a valid overlay graphics as a Named Overlay Candidate

13.5.1 Named CGs (Named Overlay Graphics)

Use of Named CGs

Named CGs is a functionality allowing CGs (or overlay graphics) to be fired from within a template or via a control command (keyboard shortcut). These CGs are to be treated as constants but has the ability to extract information from the current rundown.

Prerequisites

All named CGs must be placed within a single XML file named *NamedOverlayGraphics.xml*. This file must be placed as part of the configuration files for the system, for example *C:\ChannelTemplates*.

NamedOverlayGraphics.xml contains all CGs to be accessible from templates.

The CGs within this file have the following properties:

- The “slug” attribute is used as the ‘name’ of the CG (i.e. used in templates to refer to the CG).
- The content of each CG shall be identical to the CG representation in Manus Administrator files. In most circumstances only a small set of the attributes and elements are necessary.
- The “templatetype” attribute is used to identify the behaviour of the CG and should be according to the lowerthird mapping found in *newsroomtags.xml*. Default mapping is AUTOOUT.
- Setting “templatetype” to “STORYSTART” takes the corresponding CG on story transitions (i.e. when the first item or background of the story is taken). Only one CG could be used for this purpose.

Copyright © 2024 Vizrt

NamedOverlayGraphics.xml Example

Below is an example showing two named CGs. One to be fired at every story transition (MosartStoryStart) and one that could be triggered from a template (StrapsOff).

This example is created for BIGVIZ which makes use only of the <objParams> field.

```
<?xml version=”1.0” encoding=”utf-8” ?>
<items>
<!--CG used to inform BigViz about story start -->
<item slug=“MosartStoryStart” templatetype=“STORYSTART” in=“0” dur=“25”
mosid=“BIGTED.W1.BBC.MOS” objid=“BIGTED”>
    <content>
     <storyItem>
       <objParams>MOSART STORY START</objParams>
     </storyItem>
    </content>
</item>
<!--CG used to inform BigViz to take all CGs off air -->
<item slug=“StrapsOff” templatetype=“AUTOOUT-DSK” in=“0” dur=“25”
     mosid=“BIGTED.W1.BBC.MOS” objid=“BIGTED”>
   <content>
     <storyItem>
       <objParams>STRAPS OFF</objParams>
     </storyItem>
   </content>
</item>
</items>
```

13.5.2 NCS Placeholders

Named CGs supports placeholders for NCS information. The following example shows a modified “MosartStoryStart” Named CG making use of placeholders:

```
<item slug=“MosartStoryStart” templatetype=“STORYSTART” in=“0” dur=“25”
mosid=“BIGTED.W1.BBC.MOS” objid=“BIGTED”>
   <content>
     <storyItem>
       <roID>{story:roid}</roID>
       <storyID>{story:id}</storyID>
       <storySlug>{story:slug}</storySlug>
       <objParams>MOSART STORY START</objParams>
     </storyItem>
   </content>
 </item>
```

Syntax: `{[story|item]:[story_value|item_value|xpath]}`

Where:

Copyright © 2024 Vizrt

- **story:** Obtains information from the current story:
  - **roid:** MOS rundown identity.
  - **id:** MOS story identity.
  - **slug:** MOS story slug.
  - **xpath:** Arbitrary xpath from Manus Administrator story element.
- **item:** Obtains information from the current story item:
  - **roid:** MOS rundown identity.
  - **id:** MOS story identity.
  - **slug:** MOS story slug.
  - **type:** Viz Mosart type.
  - **variant:** Viz Mosart variant.
  - **template:** Viz Mosart type + Viz Mosart variant.
  - **xpath:** Arbitrary xpath from Manus Administrator story item element.

Examples of valid placeholders

- **{story:roid}:** The rundown identity.
- **{story:id}:** MOS story identity.
- **{story:slug}:** MOS story slug.
- **{item:type}:** MOS story item, Viz Mosart template type.
- **{item:variant}:** MOS story item, Viz Mosart template variant.
- **{item:template}:** MOS story item, Viz Mosart template type+variant.
- **{item:@templatetype}:** xpath, Viz Mosart template variant.
- **{item:item[@type=100][1]/*/storyItem/objParams}:** xpath, pics legend from first lowerthird object within a story item.

13.5.3 Named CG Actions

It is possible to trigger defined actions within Overlay Graphics Interface when taking a Named CG. The actions are specified within an action list of the CG. A typical reason for using such actions is when the graphics systems require special graphics to be sent for taking out items.

A sample action list is defined in the XML as follows:

```
<item slug="LOGO OFF" templatetype="AUTOOUT-DSK" in="0" dur="12" >
   <actions>
     <action name="takeOut" value="[last¦lastLocator¦lastManual" />
     <action name="clear" />
   </actions>
   <content>
   </content>
</item>
```

The following actions are available:

- **takeout:** Takes out any stored On Air graphics matching the criteria specified in the value field:

- **last:** Takes the last taken graphics out.
- **lastLocator:** Takes the last locator graphics out.
- **lastManual:** Takes the last manual graphics out.

- **clear:** Takes out all On Air graphics.

> ⚠ **Note:** Named CG actions do not work with Trio Interface.

13.5.4 Required Fields in XML

Example of complete lower third graphics XML

Below is an example graphics XML extracted from a running order:

```
<item type="100" slug="In:00:00/LIVE// Mosart=L|00:00|B" source="1" index="100_NWATVN
T1;P_ATVNEWSWF_HOLD MOSART
TRAINING;AFD78EF5-3D19-48F1-860BB9B2B53A679F_NWATVNT1;P_ATVNEWSWF_HOLD MOSART
TRAININGR_AFD78EF5-3D19-48F1-860BB9B2B53A679F;21115F3E-B25A-46AD-89620E63F8077CCF_2"
idref="2" templatetype="BACKGROUNDEND-DSK" status="0" error="0" in="0" dur="500" pin="
0" pdur="500" rdur="0" externaleffect="" intimeline="true" date_0="" accessory="False
" static="false" endfrase="" rundown="NWATVNT1;P_ATVNEWS\W\F_HOLD MOSART
TRAINING;AFD78EF5-3D19-48F1-860BB9B2B53A679F" storyid="NWATVNT1;P_ATVNEWS\W\F_HOLD
MOSART TRAINING\R_AFD78EF5-3D19-48F1-860BB9B2B53A679F;21115F3E-
B25A-46AD-89620E63F8077CCF" typetext="" mosid="PILOT.ATV.TEN.MOS" objid="1863809"
ismoselement="true" use_graphics_id="true" graphics_id="1863809" handler_name="DSK"
graphics_out_on="BACKGROUNDEND" description="(DSK) - In:00:00/LIVE// Mosart=L|00:00|
B" owner="">
   <fields>
     <field name="graphics_description" fieldtype="TEXT" value="In:00:00/LIVE//
Mosart=L|00:00|B" />
     <field name="graphics_id" fieldtype="TEXT" value="1863809" />
     <field name="tc_dur" fieldtype="TIMECODE" inputmask="mm:ss" default="00:00"
value="00:20" />
     <field name="continuecount" value="-1" fieldtype="TEXT" />
     <field name="tc_in" fieldtype="TIMECODE" inputmask="mm:ss" default="00:00" value="
00:00" />
   </fields>
</item>
```

Required fields: `slug, templatetype, in, dur, use_graphics_id, graphics_id, handler_name, graphics_out_on, description`.

Extracted XML example

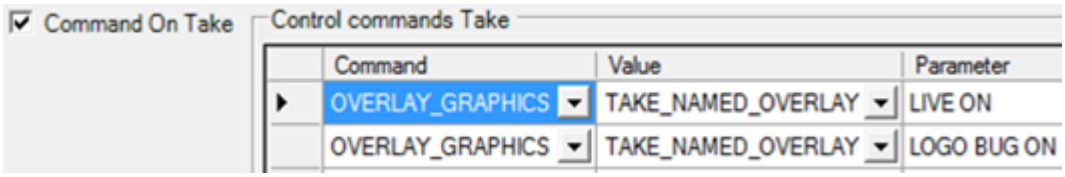
The required fields extracted look like this, please note that the slug and description have changed:

```
<items>
<!--LIVE ON -->
    <item slug="LIVE ON" in="0" dur="125" use_graphics_id="true" graphics_id="1863809
" handler_name="DSK" graphics_out_on="BACKGROUNDEND" description="(DSK) - LIVE ON"/>
</items>
```

For the above example the name used to recall the Named CG is “LIVE ON”.

13.5.5 Adding the Named CG to a Template

SLUG is where we name the overlay for our use – and we can now recall the super from a template as shown below:



13.5.6 Obtaining a valid overlay graphics as a Named Overlay Candidate

A tip to obtain candidates for named overlays is as follows:

1. Forward the graphics you want to be part of named overlay as ordinary overlay graphics (i.e. by assigning them in the NRCS). Mosart now reads these as part of overlays in a rundown.
2. Verify that the named overlay candidates appear in OverlayGraphics in the left column. All active overlay graphics in the rundown appears here.
3. Select a named overlay candidate in the OverlayGraphics left column and press the **Info** button below.
4. Now you see the XML representation of that particular overlay candidate. Copy this XML to the clipboard.
5. Paste the content of the clipboard into the named overlay XML file, *C:\ChannelTemplates\NamedOverlayGraphics.xml*.
6. Optionally, remove not required fields as described in the former section Required fields in XML.

Copyright © 2024 Vizrt

13.6 Stagetec Driver Configuration

This section describes how to configure the Stagetec driver. The configuration consists of three parts:

- If needed, change the AudioMixerStagetecDiamondRas.xml config file
- Configure one or more scenarios
- Configure Mosart to use these scenarios

13.6.1 AudioMixerStagetecDiamondRas.xml Config File

The default contents of this file are as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<DeviceConfig name="StagetecDiamondRasConfig">
  <Properties>
            <!--The strip number of the leftmost fader in the add-on panel-->
            <!--The default setting is for controlling strips 33-40-->
            <item name="FirstAddOnStrip" value="33"/>

            <!--This alternative setting is for controlling strips 1-8 of the Auratus
simulator-->
            <!--<item name="FirstAddOnStrip" value="1"/>-->

            <!--The initial scenario, if any-->
            <item name="InitialScenario" value=""/>
  </Properties>
</DeviceConfig>
```

- The default value (33) of the setting FirstAddOnStrip is for controlling the add-on panel described in the document linked to above. If that's what you want to do, you should not change it. The alternative setting is for controlling the Stagetec simulator.
- The InitialScenario setting is for setting a scenario to be invoked at start-up. The value should be a scenario name as described in the next chapter.
- If you want to change the file, please follow the normal procedure: Copy the file from ConfigurationFiles in the installation folder to Channeltemplates, then make changes to the COPY.

13.6.2 Scenario Configuration

One or more scenarios may be configured. Each scenario is represented by an xml file (i.e., with extension .xml and containing valid xml). The scenario config files must be put in one of the places normally used for config files, preferably Channeltemplates.

Copyright © 2024 Vizrt

The installation contains a sample scenario config file, Studio3Mics.xml. This file may be copied and used as a model for real scenarios. The sample scenario config file Studio3Mics.xml has the following contents:

```
<?xml version="1.0" encoding="utf-8" ?>
<StagetecDiamondRasStripsConfig>
  <Strips>
                <Strip Number="1" FaderType="DeviceType" Channel="MIC"
 PreviewOrProgram="Program" />
                <Strip Number="2" FaderType="Empty" PreviewOrProgram="Program" />
                <Strip Number="3" FaderType="Ignore" PreviewOrProgram="Program" />
                <Strip Number="4" FaderType="Static" Channel="Ch   3"
 PreviewOrProgram="Program" />
                <Strip Number="5" FaderType="DeviceType" Channel="MIC"
 PreviewOrProgram="Preview" />
                <Strip Number="6" FaderType="Empty" PreviewOrProgram="Preview" />
                <Strip Number="7" FaderType="Ignore" PreviewOrProgram="Preview" />
                <Strip Number="8" FaderType="Static" Channel="Ch   1"
 PreviewOrProgram="Preview" />
  </Strips>
</StagetecDiamondRasStripsConfig>
```

The outer parts are fixed:

```
<?xml version="1.0" encoding="utf-8" ?>
<StagetecDiamondRasStripsConfig>
  <Strips>
                ...
  </Strips>
</StagetecDiamondRasStripsConfig>
```

There are one or more Strip elements, each defining which channel to assign to the strip. The attributes are as follows:

Attribute	Valid values	Explanation
Number	1-8	This attribute determines which strip is defined. The strips should always be Number-ed 1-8; the actual strip number also depends on the value of the FirstAddOnStrip setting described above: Number - 1 + FirstAddOnStrip With the default config, Number="1" then corresponds to strip 33.

Copyright © 2024 Vizrt

FaderType	DeviceType OR Empty OR Ignore OR Static	If DeviceType: The actual channel will be determined as described below this table. If Empty: The fader/strip will be unassigned. If Ignore: The Strip element will be ignored altogether. If Static: The channel will simply be the one with name equal to the value of the Channel attribute.
Channel	Mixer channel name OR device type (none, server, server_ch2, graphics, external, mic, tlf, OR soundplayer)	If FaderType=”Static”: The mixer channel with the specified name is assigned to the stripe. If FaderType=”DeviceType”: The actual channel will be determined as described below this table.
PreviewOrProgram	Preview OR Program	Used only if FaderType=”DeviceType” (but must be present also in the other cases, but then without any effect).
Sticky	true OR false (default)	Only required if ● FaderType=”DeviceType” AND ● the fader allocation is reevaluated after the templates have changed (i.e., NOT when setting the initial scenario (see above) or changing the current scenario through a control command (see below)). The effect will be described below this table.

If FaderType=”DeviceType”, the actual channel will be determined as follows:

- A channel of the type specified by the Channel attribute (an ‘eligible’ channel) will be assigned.
- If PreviewOrProgram=”Preview”, the channel chosen will be among the eligible channels in the preview template. If the scenario has SEVERAL Strip elements with
  - FaderType=”DeviceType”,
  - the same Channel, and
  - PreviewOrProgram=”Preview”

each such Strip element will define a different channel (as long as there are enough eligible channels in the preview template).

Copyright © 2024 Vizrt

- Similarly, if PreviewOrProgram=”Program”, the channel chosen will be among the eligible channels in the program template. If the scenario has SEVERAL Strip elements with
  - FaderType=”DeviceType”,
  - the same Channel, and
  - PreviewOrProgram=”Program”

each such Strip element will define a different channel (as long as there are enough eligible channels in the program template).

- If there are not enough eligible channels (in the preview or program template, respectively) of the specified type, each surplus strip will be DEASSIGNED (i.e., not have a fader/channel assigned), UNLESS the strip is Sticky, AND the fader allocation is reevaluated after the templates have changed. (In this case, the strip is left as is; hence the designation ‘sticky’.)
- The Strip elements with FaderType=”Static” does not play any role whatsoever here; so the same channel may be assigned to several strips.

Assuming that the preview template contains the MIC channel Mic1 (but not Mic2 or any other), and that the program template contains the MIC channel Mic2 (but not Mic1 or any other), the sample scenario results in this assignment:

Strip	Channel
33	Mic2
34	(unassigned)
35	(unchanged)
36	Ch 3
37	Mic1
38	(unassigned)
39	(unchanged)
40	Ch 1

Note: the fader allocation described above is reevaluated whenever the templates changes.

13.6.3 Using the Scenarios

The scenarios will be invoked through *control commands*. Each such control command should be as follows:

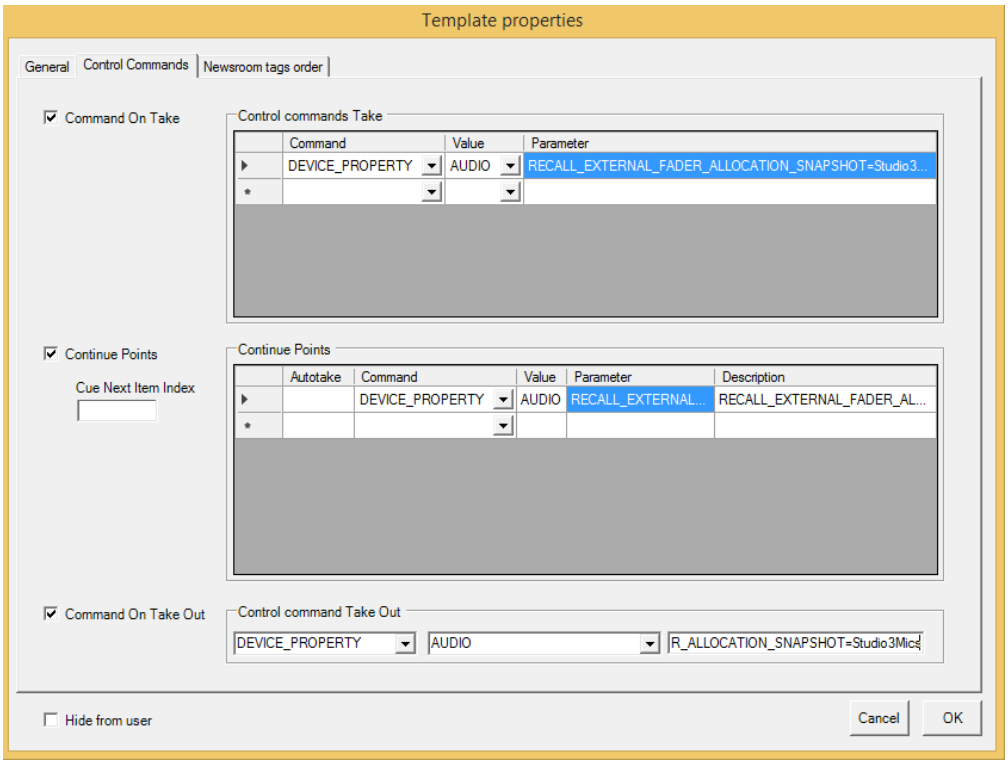
- Command: DEVICE_PROPERTY (usually selectable in a dropdown)
- Value: AUDIO (sometimes, but not always, selectable in a dropdown)

Copyright © 2024 Vizrt

- Parameter: RECALL_EXTERNAL_FADER_ALLOCATION_SNAPSHOT=Studio3Mics

> **Note:** *Studio3Mics* is an example only. In practice, the actual scenario name (= file name WITHOUT EXTENSION .xml) should be supplied.

In Template properties it looks like this:



On the UI:

Copyright © 2024 Vizrt

13.7 Structure Of MOS Objects Sent To NRCS

You can send Mosart templates as MOS objects to an NRCS and then use them to create items in a Viz Mosart rundown (as an alternative to the Mosart ActiveX method) as a brick system.
In addition to MOS objects that contain Viz Mosart templates, there is a special MOS object that can be sent containing lower thirds information.

This topic is covered in section Mosart Templates Feedback to the NRCS.

This section provides some sample MOS object files:

- MOS Objects with Mosart Templates Grouped by Type
- MOS Object with Mosart Templates not Grouped by Type
- MOS Object with Clip Templates Merged in a Mosart Template
- MOS Object with Lowerthird Information
- Octopus

Copyright © 2024 Vizrt

13.7.1 MOS Objects with Mosart Templates Grouped by Type

- Variants for Mosart Templates of Same Type
- Variants for Mosart Templates of Same Type and SendAllTemplateSets set to True

Variants for Mosart Templates of Same Type

A single MOS object will be created for Mosart templates of same type from the default template set. In this example

- <mosExternalMetadata> node contains the list of variants for the Mosart templates, of same type.
- <objID> of the MOS object is the template type, for example 0 for CAMERA or 1 for PACKAGE.
- <objGroup> is the default template set name.
- <type> node has the name set to the template type name (for example CAMERA, PACKAGE etc.), and *templateset* set to the name of the default template set.

```
<mosObj>
    <objID>0</objID>
    <objSlug>CAMERA</objSlug>
    <mosAbstract>CAMERA</mosAbstract>
    <objGroup>Talkshow</objGroup>
    <objType>STILL</objType>
    <objTB>50</objTB>
    <objRev>1</objRev>
    <objDur>0</objDur>
    <status>NEW</status>
    <objAir>NOT READY</objAir>
    <createdBy>
    </createdBy>
    <created>2019-03-04T13:23:38,000Z</created>
    <changedBy>
    </changedBy>
    <changed>2020-03-07T09:22:21,256Z</changed>
    <description>CAMERA</description>
    <mosExternalMetadata>
    <mosScope>PLAYLIST</mosScope>
    <mosSchema>http://www.mosartmedialab.no/schema/mositem.dtd</mosSchema>
    <mosPayload>
        <mosarttemplate>
        <type name="CAMERA" templateset=”Talkshow” category="">
            <variants fieldtype="LIST" value="1">
            <variant name="1">
                <transitions value="DEFAULT" enable="false">
                <transition name="DEFAULT">
                    <field name="" value="" fieldtype="TEXT" />
                </transition>
```

Copyright © 2024 Vizrt

```
                <transition name="MIX">
                    <field name="transitionrate" value="0" fieldtype="NUMBER" range="
0,999" />
                </transition>
                <transition name="WIPE">
                    <field name="transitionrate" value="" fieldtype="NUMBER" range="0
,999" />
                </transition>
                <transition name="EFFECT">
                    <field name="effectname" value="" fieldtype="LIST" keylist="effec
ts" />
                </transition>
                </transitions>
                <fields />
            </variant>
            <variant name="2">
                <transitions value="DEFAULT" enable="false">
                <transition name="DEFAULT">
                    <field name="" value="" fieldtype="TEXT" />
                </transition>
                <transition name="MIX">
                    <field name="transitionrate" value="0" fieldtype="NUMBER" range="
0,999" />
                </transition>
                <transition name="WIPE">
                    <field name="transitionrate" value="" fieldtype="NUMBER" range="0
,999" />
                </transition>
                <transition name="EFFECT">
                    <field name="effectname" value="" fieldtype="LIST" keylist="effec
ts" />
                </transition>
                </transitions>
                <fields />
            </variant>
            </variants>
            <keys name="mixerinputs">
            <k>
            </k>
            <k>CAM1</k>
            <k>CAM2</k>
            <k>GRAPHIC1</k>
            <k>GRAPHIC2</k>
            <k>LIVE1</k>
            <k>LIVE2</k>
            <k>SKYPE1</k>
            <k>SKYPE2</k>
            <k>PLAYA</k>
            <k>PLAYB</k>
            <k>PLAYC</k>
            <k>PLAYD</k>
            <k>ME1</k>
```

```
            <k>ME2</k>
            <k>ME3</k>
            </keys>
            <keys name="effects">
            <k>1</k>
            <k>2</k>
            <k>3</k>
            <k>4</k>
            <k>5</k>
            <k>6</k>
            <k>7</k>
            <k>8</k>
            <k>9</k>
            <k>10</k>
            <k>11</k>
            <k>12</k>
            <k>13</k>
            <k>14</k>
            <k>45</k>
            <k>47</k>
            </keys>
            <keys name="routersources">
            <k>SAT_10</k>
            <k>DEP_01_A</k>
            </keys>
            <keys name="routerdestinations">
            <k>M_VISU_2</k>
            <k>M_ROBO_2</k>
            <k>TESTLOW</k>
            <k>TESTHIGH</k>
            </keys>
        </type>
        </mosarttemplate>
    </mosPayload>
    </mosExternalMetadata>
</mosObj>
```

Variants for Mosart Templates of Same Type and SendAllTemplateSets set to True

A single MOS object will be created for Mosart templates of the same type for each template set. In this example:

- <mosExternalMetadata> node contains the list of variants for the Mosart templates of the same type.
  For example, if there are 3 template sets and 10 template types having 2 variants each, then there are *30* MOS objects created, each having 2 variants.
- <objID> is the “TemplateSet-TemplateType”.
  For example in the ’FuncSpec’ template set, FuncSpec-0 for CAMERA or FuncSpec-1 for PACKAGE.
- <objSlug> is the template type name.
  For example, CAMERA or PACKAGE.

Copyright © 2024 Vizrt

- <objGroup> is the template set name.
- <type> node has the name of the template type name (for example CAMERA, PACKAGE etc.), and *templateset* set to the name of the template set.
  In the example below there are there are 2 template sets:
  - *FuncSpec* having 3 Camera templates and
  - *Demo* having 2 Camera templates.

```
<mosObj>
    <objID>FUNCSPEC-0</objID>
    <objSlug>CAMERA</objSlug>
    <mosAbstract>CAMERA</mosAbstract>
    <objGroup>Funcspec</objGroup>
    <objType>STILL</objType>
    <objTB>50</objTB>
    <objRev>1</objRev>
    <objDur>0</objDur>
    <status>NEW</status>
    <objAir>NOT READY</objAir>
    <createdBy>
    </createdBy>
    <created>2016-04-01T11:42:09,000Z</created>
    <changedBy>
    </changedBy>
    <changed>2017-09-22T12:05:20,070Z</changed>
    <description>CAMERA</description>
    <mosExternalMetadata>
      <mosScope>PLAYLIST</mosScope>
      <mosSchema>http://www.mosartmedialab.no/schema/mositem.dtd</mosSchema>
      <mosPayload>
        <mosarttemplate>
          <type name="CAMERA" templateset="Funcspec" category="">
            <variants fieldtype="LIST" value="LIVE">
              <variant name="LIVE">
                <transitions value="DEFAULT" enable="false">
                  <transition name="DEFAULT">
                    <field name="" value="" fieldtype="TEXT" />
                  </transition>
                  <transition name="MIX">
                    <field name="transitionrate" value="0" fieldtype="NUMBER" range="
0,999" />
                  </transition>
                  <transition name="WIPE">
                    <field name="transitionrate" value="" fieldtype="NUMBER" range="0
,999" />
                  </transition>
                  <transition name="EFFECT">
                    <field name="effectname" value="" fieldtype="LIST" keylist="effec
ts" />
                  </transition>
                </transitions>
                <fields>
```

Copyright © 2024 Vizrt

```
                  <field name="xpoint1" value="CAM1" default="CAM1" fieldtype="LIST"
 keylist="mixerinputs" />
                </fields>
              </variant>
              <variant name="21">
                <transitions value="DEFAULT" enable="false">
                  <transition name="DEFAULT">
                    <field name="" value="" fieldtype="TEXT" />
                  </transition>
                  <transition name="MIX">
                    <field name="transitionrate" value="2" fieldtype="NUMBER" range="
0,999" />
                  </transition>
                  <transition name="WIPE">
                    <field name="transitionrate" value="" fieldtype="NUMBER" range="0
,999" />
                  </transition>
                  <transition name="EFFECT">
                    <field name="effectname" value="" fieldtype="LIST" keylist="effec
ts" />
                  </transition>
                </transitions>
                <fields />
              </variant>
              <variant name="20">
                <transitions value="DEFAULT" enable="false">
                  <transition name="DEFAULT">
                    <field name="" value="" fieldtype="TEXT" />
                  </transition>
                  <transition name="MIX">
                    <field name="transitionrate" value="2" fieldtype="NUMBER" range="
0,999" />
                  </transition>
                  <transition name="WIPE">
                    <field name="transitionrate" value="" fieldtype="NUMBER" range="0
,999" />
                  </transition>
                  <transition name="EFFECT">
                    <field name="effectname" value="" fieldtype="LIST" keylist="effec
ts" />
                  </transition>
                </transitions>
                <fields />
              </variant>
            </variants>
            <keys name="mixerinputs">
              <k>
              </k>
              <k>CAM1</k>
              <k>CAM2</k>
              <k>LIVE1</k>
              <k>VIZ</k>
              <k>IN11_HD1</k>
```

```
              <k>IN12_HD2</k>
              <k>VIZ2</k>
              <k>ME1</k>
              <k>ME2</k>
              <k>LIVE2</k>
              <k>DUMMY</k>
              <k>DDR1</k>
              <k>DDR2</k>
            </keys>
            <keys name="effects">
              <k>1</k>
              <k>2</k>
              <k>3</k>
              <k>4</k>
              <k>5</k>
              <k>6</k>
              <k>7</k>
              <k>8</k>
              <k>9</k>
            </keys>
            <keys name="routersources">
              <k>R CAM 1</k>
            </keys>
            <keys name="routerdestinations">
              <k>OPUS 1</k>
            </keys>
          </type>
        </mosarttemplate>
      </mosPayload>
    </mosExternalMetadata>
  </mosObj>
<mosObj>
    <objID>DEMO-0</objID>
    <objSlug>CAMERA</objSlug>
    <mosAbstract>CAMERA</mosAbstract>
    <objGroup>DEMO</objGroup>
    <objType>STILL</objType>
    <objTB>50</objTB>
    <objRev>1</objRev>
    <objDur>0</objDur>
    <status>NEW</status>
    <objAir>NOT READY</objAir>
    <createdBy>
    </createdBy>
    <created>2014-11-11T12:55:05,000Z</created>
    <changedBy>
    </changedBy>
    <changed>2016-05-04T11:01:27,903Z</changed>
    <description>CAMERA</description>
    <mosExternalMetadata>
      <mosScope>PLAYLIST</mosScope>
      <mosSchema>http://www.mosartmedialab.no/schema/mositem.dtd</mosSchema>
      <mosPayload>
```

```
        <mosarttemplate>
          <type name="CAMERA" templateset="DEMO" category="">
            <variants fieldtype="LIST" value="1">
              <variant name="1">
                <transitions value="DEFAULT" enable="false">
                  <transition name="DEFAULT">
                    <field name="" value="" fieldtype="TEXT" />
                  </transition>
                  <transition name="MIX">
                    <field name="transitionrate" value="0" fieldtype="NUMBER" range="
0,999" />
                  </transition>
                  <transition name="WIPE">
                    <field name="transitionrate" value="10" fieldtype="NUMBER" range="
0,999" />
                  </transition>
                  <transition name="EFFECT">
                    <field name="effectname" value="0" fieldtype="LIST" keylist="effe
cts" />
                  </transition>
                </transitions>
                <fields />
              </variant>
               <variant name="2">
                <transitions value="DEFAULT" enable="false">
                  <transition name="DEFAULT">
                    <field name="" value="" fieldtype="TEXT" />
                  </transition>
                  <transition name="MIX">
                    <field name="transitionrate" value="50" fieldtype="NUMBER" range="
0,999" />
                  </transition>
                  <transition name="WIPE">
                    <field name="transitionrate" value="50" fieldtype="NUMBER" range="
0,999" />
                  </transition>
                  <transition name="EFFECT">
                    <field name="effectname" value="0" fieldtype="LIST" keylist="effe
cts" />
                  </transition>
                </transitions>
                <fields />
              </variant>
            </variants>
            <keys name="mixerinputs">
              <k>
              </k>
              <k>CAM1</k>
              <k>CAM2</k>
              <k>LIVE1</k>
              <k>VIZ</k>
              <k>IN11_HD1</k>
              <k>IN12_HD2</k>
```

Copyright © 2024 Vizrt

```
            <k>VIZ2</k>
            <k>ME1</k>
            <k>ME2</k>
            <k>LIVE2</k>
            <k>DUMMY</k>
            <k>DDR1</k>
            <k>DDR2</k>
          </keys>
          <keys name="effects">
            <k>1</k>
            <k>2</k>
            <k>3</k>
            <k>4</k>
            <k>5</k>
            <k>6</k>
            <k>7</k>
            <k>8</k>
            <k>9</k>
          </keys>
          <keys name="routersources">
            <k>R CAM 1</k>
          </keys>
          <keys name="routerdestinations">
            <k>OPUS 1</k>
          </keys>
        </type>
      </mosarttemplate>
    </mosPayload>
  </mosExternalMetadata>
</mosObj>
```

Copyright © 2024 Vizrt

13.7.2 MOS Object with Mosart Templates not Grouped by Type

For each Mosart template, a MOS object will be created to be sent to NCS. In this example:

- <mosExternalMetadata>is the node containing the variant of the Mosart template it represents.
- <objID> is the format <TemplateSet>-<Type>-<Variant>.
- <objSlug> has <Type> <Variant>;<Name>.
- <objGroup> is set to the template set the template belongs to.
- <type> node has its name set to the type of the template.
  For example CAMERA, PACKAGE etc, and *templateset* has the name of the template set this template belongs to.
- Other information is:
  - The list of video crosspoints (“mixerinputs”)
  - Effect numbers (“effects”)
  - Router sources (“routersources”)
  - Router destinations (“routerdestinations”)
    as configured in AvConfig.xml file.

The name of these key lists must be exactly as given in the parentheses, because they can be referenced from a Mosart item when adding newsroomtags to a template.

```
<mosObj>
    <objID>Talkshow-1-FULL</objID>
    <objSlug>PACKAGE FULL;name...</objSlug>
    <mosAbstract>PACKAGE FULL;description</mosAbstract>
    <objGroup>Talkshow</objGroup>
    <objType>STILL</objType>
    <objTB>50</objTB>
    <objRev>1</objRev>
    <objDur>0</objDur>
    <status>UPDATED</status>
    <objAir>NOT READY</objAir>
    <createdBy/>
    <created/>
    <changedBy/>
    <changed/>
    <description>PACKAGE</description>
    <mosExternalMetadata>
    <mosScope>PLAYLIST</mosScope>
    <mosSchema>http://www.mosartmedialab.no/schema/mositem.dtd</mosSchema>
    <mosPayload>
        <mosarttemplate>
        <type name="PACKAGE" templateset=”Talkshow” category="CLIP">
            <variants fieldtype="LIST" value="FULL">
            <variant name="FULL">
                <transitions value="DEFAULT" enable="false">
                <transition name="DEFAULT">
                    <field name="" value="" fieldtype="TEXT" />
```

```
                </transition>
                <transition name="MIX">
                    <field name="transitionrate" value="0" fieldtype="NUMBER"
range="0,999" />
                </transition>
                <transition name="WIPE">
                    <field name="transitionrate" value="" fieldtype="NUMBER"
range="0,999" />
                </transition>
                <transition name="EFFECT">
                    <field name="effectname" value="" fieldtype="LIST"
keylist="effects" />
                </transition>
                </transitions>
                <fields>
                <field name="clip_hirespath" fieldtype="TEXT" />
                <field name="clip_description" fieldtype="TEXT" />
                <field name="clip_mark_in" default="00:00:00:00"
     fieldtype="TIMECODE" inputmask="hh:mm:ss:ff" />
                <field name="clip_dur" default="00:00:00"
          fieldtype="TIMECODE" inputmask="mm:ss:ff" />
                </fields>
             </variant>
             </variants>
             <keys name="mixerinputs">
             <k>
             </k>
             <k>CAM1</k>
             <k>CAM2</k>
             <k>GRAPHIC1</k>
             <k>GRAPHIC2</k>
             <k>LIVE1</k>
             <k>LIVE2</k>
             <k>SKYPE1</k>
             <k>SKYPE2</k>
             <k>PLAYA</k>
             <k>PLAYB</k>
             <k>PLAYC</k>
             <k>PLAYD</k>
             <k>ME1</k>
             <k>ME2</k>
             <k>ME3</k>
             </keys>
             <keys name="effects">
             <k>1</k>
             <k>2</k>
             <k>3</k>
             <k>4</k>
             <k>5</k>
             <k>6</k>
             <k>7</k>
             <k>8</k>
             <k>9</k>
```

Copyright © 2024 Vizrt

```
            <k>10</k>
            <k>11</k>
            <k>12</k>
            <k>13</k>
            <k>14</k>
            <k>45</k>
            <k>47</k>
            </keys>
            <keys name="routersources">
            <k>SAT_10</k>
            <k>DEP_01_A</k>
            </keys>
            <keys name="routerdestinations">
            <k>M_VISU_2</k>
            <k>M_ROBO_2</k>
            <k>TESTLOW</k>
            <k>TESTHIGH</k>
            </keys>
        </type>
        </mosarttemplate>
    </mosPayload>
    </mosExternalMetadata>
  </mosObj>
```

Newsroomtags for crosspoints and router control inputs

When enabling a Switcher Crosspoint or a Router Control device for the template given above and adding some newsroomtags for crosspoints and for router control inputs, the Mosart item created from the template above could have the following fields and transitions:

```
  <fields>
        <field name="input_preview" value="SAT_10" default="SAT_10" fieldtype="LIST"
keylist="routersources" />
            <field name="input_program" value="SAT_10" default="SAT_10" fieldtype="LI
ST"
keylist="routersources" />
            <field name="output_preview" value="M_VISU_2" default="M_VISU_2"
fieldtype="LIST"
keylist="routerdestinations" />
            <field name="output_program" value="M_VISU_2" default="M_VISU_2"
fieldtype="LIST"
keylist="routerdestinations" />
            <field name="xpoint" value="CAM2" default="CAM2" fieldtype="LIST" keylist="
mixerinputs" />
      </fields>
     <transitions value="EFFECT" enable="false" rate="9" next_cue_delay="10">
            <transition name="EFFECT">
            <field name="effectname" value="9" fieldtype="LIST" keylist="effects" />
            </transition>
      </transitions>
```

Copyright © 2024 Vizrt

Note: The *keylists* in the Mosart item refers to the keys in the MOS object for a Mosart template.

Copyright © 2024 Vizrt

13.7.3 MOS Object with Clip Templates Merged in a Mosart Template

A single MOS object is created for all Mosart templates of type PACKAGE or VOICEOVER from the default template set. In this example:

- <mosExternalMetadata> node contains the list of variants for the Mosart templates of type PACKAGE and VOICEOVER.
- <objID> of the MOS object is set to *25*, representing a Mosart item type CLIP.
- <objGroup> is the default template set name.
- <type> node has name and category set to “CLIP”, and *templateset* set to the name of the default template set.
- The *variant* for each template is built using some prefixes:
  1) for PACKAGE, the variant is set to MAZ-<template variant>
  2) for VOICEOVER, the variant is set to OFFMAZ-<template variant>

⚠ **Note**: When **GroupByType** is *false*, no MOS object will be sent for templates of type PACKAGE and VOICEOVER.

```
<mosObj>
<objID>LOWERTHIRDS</objID>
<objSlug>LOWERTHIRDS</objSlug>
<mosAbstract />
<objGroup>TEMPLATES</objGroup>
<objType>STILL</objType>
<objTB>50</objTB>
<objRev>1</objRev>
<objDur>0</objDur>
<!-- The status of the MOS object can be NEW, UPDATED or DELETED -->
<status>NEW</status>
<objAir>NOT READY</objAir>
<createdBy/>
<created/>
<changedBy/>
<changed/>
<description />
<mosExternalMetadata>
      <mosScope>PLAYLIST</mosScope>
      <mosSchema>http://www.mosartmedialab.no/schema/mositem.dtd</mosSchema>
      <mosPayload>
         <mosarttemplate>
        <type name="LOWERTHIRDS" category="GRAPHICS">
            <variants fieldtype="LIST" value="AUTOOUT">
            <variant name="AUTOOUT">
                <fields>
                  <field name="graphics_description" fieldtype="TEXT" />
                        <field name="graphics_id" fieldtype="TEXT" />
                       <field name="tc_in" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00"/>
```

```
                        <field name="tc_out" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00"/>
                        <field name="channel" value="DSK" fieldtype="LIST" keylist="ch
annels" />
                        <field name="take" value="AUTO" fieldtype="LIST" keylist="take
types" />
                </fields>
            </variant>
            <variant name="KEEPWHILEBACKGROUND">
                <fields>
                  <field name="graphics_description" fieldtype="TEXT" />
                        <field name="graphics_id" fieldtype="TEXT" />
                       <field name="tc_in" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00" />
                        <field name="tc_out" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00" />
                        <field name="channel" value="DSK" fieldtype="LIST" keylist="c
hannels" />
                         <field name="take" value="AUTO" fieldtype="LIST" keylist="ta
ketypes" />
                </fields>
            </variant>
            <variant name="KEEPWHILESTORY">
                <fields>
                    <field name="graphics_description" fieldtype="TEXT" />
                           <field name="graphics_id" fieldtype="TEXT" />
                        <field name="tc_in" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00" />
                        <field name="tc_out" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00" />
                         <field name="channel" value="DSK" fieldtype="LIST" keylist="
channels" />
                        <field name="take" value="AUTO" fieldtype="LIST" keylist="ta
ketypes" />
                </fields>
            </variant>
            <variant name="NONEAUTOOUT">
                <fields>
                    <field name="graphics_description" fieldtype="TEXT" />
                           <field name="graphics_id" fieldtype="TEXT" />
                        <field name="tc_in" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00" />
                        <field name="tc_out" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00" />
                         <field name="channel" value="DSK" fieldtype="LIST" keylist="
channels" />
                        <field name="take" value="AUTO" fieldtype="LIST" keylist="ta
ketypes" />
                </fields>
            </variant>
            </variants>
            <!-- List of active channels (graphics destinations) as configured in
OverlayGraphics Interface-->
```

```
            <keys name="channels">
            <k>WALL_2</k>
            <k>WALL</k>
            <k>TABLE</k>
            <k>DSK</k>
  </keys>
            <keys name="taketypes">
            <k>AUTO</k>
            <k>MANUAL</k>
            </keys>
         </type>
        </mosarttemplate>
    </mosPayload>
</mosExternalMetadata>
</mosObj>
```

The list of channels contains the active destinations (destinations linked to a graphic engine) as configured in the **Overlay Graphics** interface.

For details of configuration, refer to section xxx.

Copyright © 2024 Vizrt

13.7.4 MOS Object with Lowerthird Information

Along MOS objects corresponding to the Mosart channel templates, Mosart also sends to NCS a MOS object containing information about lower thirds.

```
<mosObj>
<objID>LOWERTHIRDS</objID>
<objSlug>LOWERTHIRDS</objSlug>
<mosAbstract />
<objGroup>TEMPLATES</objGroup>
<objType>STILL</objType>
<objTB>50</objTB>
<objRev>1</objRev>
<objDur>0</objDur>
<!-- The status of the MOS object can be NEW, UPDATED or DELETED -->
<status>NEW</status>
<objAir>NOT READY</objAir>
<createdBy/>
<created/>
<changedBy/>
<changed/>
<description />
<mosExternalMetadata>
      <mosScope>PLAYLIST</mosScope>
      <mosSchema>http://www.mosartmedialab.no/schema/mositem.dtd</mosSchema>
      <mosPayload>
         <mosarttemplate>
        <type name="LOWERTHIRDS" category="GRAPHICS">
            <variants fieldtype="LIST" value="AUTOOUT">
            <variant name="AUTOOUT">
                <fields>
                  <field name="graphics_description" fieldtype="TEXT" />
                         <field name="graphics_id" fieldtype="TEXT" />
                        <field name="tc_in" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00"/>
                       <field name="tc_out" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00"/>
                       <field name="channel" value="DSK" fieldtype="LIST" keylist="ch
annels" />
                       <field name="take" value="AUTO" fieldtype="LIST" keylist="take
types" />
                </fields>
            </variant>
            <variant name="KEEPWHILEBACKGROUND">
                <fields>
                  <field name="graphics_description" fieldtype="TEXT" />
                        <field name="graphics_id" fieldtype="TEXT" />
                       <field name="tc_in" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00" />
                        <field name="tc_out" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00" />
```

```
                        <field name="channel" value="DSK" fieldtype="LIST" keylist="c
hannels" />
                         <field name="take" value="AUTO" fieldtype="LIST" keylist="ta
ketypes" />
                </fields>
            </variant>
            <variant name="KEEPWHILESTORY">
                <fields>
                    <field name="graphics_description" fieldtype="TEXT" />
                           <field name="graphics_id" fieldtype="TEXT" />
                        <field name="tc_in" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00" />
                        <field name="tc_out" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00" />
                         <field name="channel" value="DSK" fieldtype="LIST" keylist="
channels" />
                        <field name="take" value="AUTO" fieldtype="LIST" keylist="ta
ketypes" />
                </fields>
            </variant>
            <variant name="NONEAUTOOUT">
                <fields>
                    <field name="graphics_description" fieldtype="TEXT" />
                           <field name="graphics_id" fieldtype="TEXT" />
                        <field name="tc_in" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00" />
                        <field name="tc_out" fieldtype="TIMECODE" inputmask="mm:ss"
default="00:00" />
                         <field name="channel" value="DSK" fieldtype="LIST" keylist="
channels" />
                        <field name="take" value="AUTO" fieldtype="LIST" keylist="ta
ketypes" />
                </fields>
            </variant>
            </variants>
            <!-- List of active channels (graphics destinations) as configured in
OverlayGraphics Interface-->
            <keys name="channels">
            <k>WALL_2</k>
            <k>WALL</k>
            <k>TABLE</k>
            <k>DSK</k>
  </keys>
            <keys name="taketypes">
            <k>AUTO</k>
            <k>MANUAL</k>
            </keys>
        </type>
       </mosarttemplate>
    </mosPayload>
</mosExternalMetadata>
</mosObj>
```

Note: The list of channels contains the active destinations (destinations linked to a graphic engine) as configured in **Overlay Graphics** interface.

Copyright © 2024 Vizrt

13.7.5 Octopus

When using template feedback to Octopus, templates are listed with the Mosart Type names only, not the translated names from the NRCS tags list in Newsroomsettings.

- For Viz Mosart to be able to create the correct timeline objects you must include the Mosart type names as NRCS tag names for all template types.
  As a partial example:

```
<tags>
<tag name="CAM" type="CAMERA" scope="All" />
<tag name="CAMERA" type="CAMERA" scope="All" />
<tag name="OOV" type="VOICEOVER" scope="All" />
<tag name="VOICEOVER" type="VOICEOVER" scope="All" />
```

Copyright © 2024 Vizrt