



Viz Mosart Administrator Guide

Version 4.0



Viz Mosart

vizrt



Copyright © 2020 Vizrt. All rights reserved.

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

Disclaimer

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

Technical Support

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

Created on

2020/09/22

Contents

1	Introduction.....	12
1.1	Related Documents.....	12
1.2	Feedback and Suggestions.....	12
1.3	Legal.....	12
2	About Viz Mosart.....	13
2.1	Naming Convention for Components.....	13
2.2	The Viz Mosart System.....	14
2.2.1	Viz Mosart Applications.....	15
2.3	Viz Mosart Server.....	16
2.3.1	Viz Mosart Server.....	16
2.3.2	Viz Mosart Server Components.....	17
2.4	Viz Mosart Client.....	25
2.4.1	Viz Mosart GUI.....	25
2.4.2	Audio Panel.....	27
2.4.3	Timing Display.....	28
2.4.4	Audio Player.....	28
2.4.5	ActiveX.....	29
2.4.6	iNews Timer.....	30
2.5	Other Viz Mosart Applications.....	30
2.5.1	Media Router and Mosart Template Database.....	30
2.5.2	Omnibus Controller.....	31
2.5.3	SNMP Service.....	31
2.5.4	Test Suite.....	31
3	Installation.....	32
3.1	Prerequisites.....	32
3.1.1	System Requirements.....	32
3.1.2	Hardware Requirements.....	35
3.2	Viz Mosart Installation Files.....	38
3.2.1	Viz Mosart Installer Files.....	38
3.3	Version Numbering.....	39
3.4	Manual Installation.....	40
3.4.1	To Perform a Manual Installation.....	40
3.4.2	Viz Mosart Backup.....	41
3.5	Viz Mosart Installation Administrator.....	41

3.5.1	Installing the Viz Mosart Installation Administrator (MIA)	42
3.5.2	Viz Mosart Installation Administrator Interface.....	43
3.5.3	Semi-Automated Installation	43
3.5.4	Settings	44
3.5.5	Take Snapshot	45
3.5.6	Start Mosart Application.....	45
3.5.7	Backup Files.....	46
3.5.8	Stop Services	46
3.5.9	Uninstall Mosart.....	47
3.5.10	Install Mosart.....	48
3.5.11	Start Services	50
4	Manus Administrator	51
4.1	NRCS Software Compatibility.....	51
4.2	Configuration Utilities	51
4.3	Settings Editor iNews	52
4.3.1	Configuration	53
4.3.2	Configuration File Paths.....	55
4.3.3	Crossover Configuration	55
4.3.4	Default Item Durations.....	55
4.3.5	iNews Configuration	56
4.3.6	iNews Web Service Configuration	56
4.3.7	Logging	56
4.3.8	Manus Configuration	57
4.3.9	Read Speed	57
4.3.10	Template Database Configuration	57
4.3.11	Viz Media Sequencer Configuration.....	58
4.4	Settings Editor MOS	58
4.4.1	Advanced.....	59
4.4.2	Configuration	59
4.4.3	Configuration File Paths.....	61
4.4.4	Crossover Configuration	62
4.4.5	Default Item Durations.....	62
4.4.6	iNews Web Service Configuration	63
4.4.7	Logging	63
4.4.8	Manus Configuration	64
4.4.9	NCS Configuration	64
4.4.10	Template Database Configuration	66

4.4.11	Notes.....	67
4.5	Configuring the MSE Playlist Panel	67
4.5.1	Setting User Access to MSE Playlist Panel	67
4.6	Newsroom Settings Editor.....	68
4.6.1	Saving Changes	69
4.6.2	Newsroom Settings Editor - Edit Menu.....	69
4.6.3	Story External Metadata	70
4.6.4	Graphic Destination Letters.....	73
4.6.5	MOS ID Mapping	86
4.6.6	Overriding Default Field Names when Parsing iNEWS Stories	86
4.7	Field Mapping.....	87
4.7.1	The Tool Set	88
4.7.2	What to Configure.....	88
4.7.3	Target Fields.....	88
4.7.4	Mapping Source Fields to Target Fields	89
4.7.5	Examples	89
4.8	Manus Administrator Commands	93
5	Media Administrator	95
5.1	Media Administrator Commands.....	95
5.2	Media Administrator Properties Editor.....	96
5.2.1	Configuration	97
5.2.2	Connection	98
5.2.3	Connection Media Search Servers.....	98
5.2.4	Logging	99
5.2.5	Manus Admin Connection	99
5.2.6	Media Router	99
6	Overlay Graphics Interface	100
6.1	[1] List of Overlay Graphics in the Rundown	102
6.2	[2] Server Status.....	104
6.3	[3] General Log	104
6.4	[4] Event Log.....	104
6.5	System Information.....	105
6.6	Menu Bar	105
6.7	Overlay Graphics Configuration	106
6.7.1	Overlay Graphics Configuration Window.....	106
6.7.2	Controllers, Engines and Destinations.....	108

6.7.3	Configuration Panels.....	109
6.7.4	Add Overlay Graphics Controllers and Engines.....	116
6.7.5	Common Graphic Controller Properties.....	118
6.7.6	Common Graphic Engine Properties.....	119
6.7.7	Overlay Graphics Types.....	120
6.7.8	Overlay Graphics Configuration Property Tabs.....	126
6.7.9	Adding a Mosart Graphics Destination.....	131
7	Audio Player.....	136
7.1	Setting Up the Audio Player.....	136
7.2	Audio Player Settings.....	137
7.3	Audio Player Configuration File.....	139
7.4	How to Set Up Audio Player.....	139
7.5	Audio Player Configuration File.....	140
8	Trio Interface.....	141
8.1	Trio Interface Configuration.....	142
8.1.1	Engines Setup Tab.....	142
8.1.2	Connections Setup Tab.....	144
8.2	Media Sequencer Redundancy.....	145
8.2.1	Switching Media Sequencer from the GUI.....	146
8.2.2	Configuration and Test.....	146
8.2.3	Test and Debug.....	147
9	ActiveX.....	149
9.1	Setting up the Registry for Viz Mosart ActiveX.....	149
9.1.1	Working with Viz Mosart Registry Keys.....	150
9.1.2	Viz Mosart ActiveX Registry Properties.....	152
9.1.3	Configuring the Newsroom System.....	158
9.2	Connecting the ActiveX to Viz Mosart Server.....	162
9.2.1	Connecting the ActiveX Directly to Viz Mosart Server via folder sharing.....	162
9.2.2	Connecting the ActiveX to Viz Mosart Server via Template Database.....	163
9.3	ActiveX Notes.....	165
10	AV Automation.....	166
10.1	Overview.....	166
10.2	Menu Description.....	167
10.3	Logging and Status.....	168
10.3.1	Logging Panel.....	168
10.4	Recording.....	168

10.4.1	To Start a Recording	169
10.4.2	To Stop a Recording.....	170
10.4.3	Supported Video Servers	170
10.5	Audio and Video Setup.....	171
10.5.1	Video Configurations	171
10.5.2	Audio Configurations.....	173
10.5.3	Vision Mixer Effects Configurations	173
10.5.4	Router Source Configurations	176
10.5.5	Router Destination Configurations	176
10.6	Template Editor	177
10.6.1	Working with Templates	178
10.6.2	Template Examples.....	196
10.6.3	Template Device Functions	205
10.6.4	Working with Template Sets	245
10.6.5	Other Template Functionality.....	252
10.7	Mosart Instrumentation Panel	256
10.8	Newsroom Tags	256
10.8.1	Adding a Newsroom tag to your Viz Mosart Template	257
10.8.2	Newsroom Tags Order	259
10.8.3	Advanced Features for Specific NRCs.....	260
11	AV Automation Device Properties.....	267
11.1	AV Automation Devices Vision Mixer.....	268
11.2	AV Automation Devices Video Servers	270
11.2.1	Working with Video Server Configurations	272
11.2.2	Working with Servers and Ports.....	279
11.2.3	Mosart Port Configurations	280
11.3	AV Automation Devices Graphics	283
11.3.1	Vizrt Settings	284
11.3.2	Deko Settings	287
11.3.3	XPression Settings	288
11.3.4	Orad Settings.....	288
11.3.5	Pixel Power Settings.....	289
11.3.6	Chyron Settings	290
11.4	AV Automation Devices Audio.....	290
11.4.1	Audio Mixer.....	291
11.4.2	Loudness Control.....	294
11.5	AV Automation Devices Camera Robotics.....	295

11.5.1	Configuring a Robotic Camera	295
11.6	AV Automation Devices Light Control	298
11.7	AV Automation Devices Router	298
11.8	AV Automation Devices Subtitles	299
11.9	AV Automation Devices GPIO	299
11.9.1	DirectTakes via GPI	301
11.9.2	Sample Configuration (W&T Web-IO)	303
11.10	AV Automation Devices Tally	305
11.11	AV Automation Devices Virtual Set	305
11.12	AV Automation Devices Weather	306
11.13	AV Automation Devices Video Wall.....	306
11.14	AV Automation Devices Integrated Engine.....	307
11.15	AV Automation Devices General	307
12	Viz Mosart Template Database.....	309
12.1	Mosart Template Definition.....	309
12.1.1	Shared Template Set	310
12.2	Installing the Viz Mosart Template Database.....	310
12.2.1	Prerequisites.....	310
12.2.2	MySQL Users	311
12.2.3	Initial Mosart Template Database	312
12.2.4	Creating the Initial Template Database.....	312
12.3	Creating and Upgrading Mosart templatedb.....	313
12.3.1	Verify WampServer is running	313
12.3.2	Running Database Scripts	313
12.3.3	Running Scripts Automatically.....	313
12.3.4	Running Scripts Manually.....	314
12.4	Configuring Viz Mosart Server.....	315
12.4.1	Connecting AvAutomation to the Mosart Template Database	315
12.4.2	Viz Mosart Template Database and Viz Mosart ActiveX	318
12.5	Backup and Recovery	319
12.5.1	Using mysqldump for backups.....	319
12.5.2	Using mysql for restore	319
12.5.3	Backup and restore from MySQL clients	319
12.6	Database Maintenance	321
12.6.1	Upgrading the Database	321
12.6.2	Viz Mosart Template Editor	322

12.6.3	TemplateSetEditor	322
12.7	Viz Mosart Template Database Specification	330
12.7.1	Entity Relationships	330
12.7.2	Introductions and Notations	331
12.7.3	Viz Mosart Database Tables	334
13	Maintenance	349
13.1	System Logging	349
13.1.1	Log File Structure.....	349
13.1.2	Log Viewer.....	350
13.1.3	Technical Log Stream.....	351
13.1.4	AsRun Log Stream.....	352
13.1.5	MOS Log Stream	353
13.1.6	Log Adapters	353
13.1.7	Log Configuration.....	353
13.1.8	Custom-built AsRunLog Adapters.....	356
13.1.9	Log Properties	357
13.2	Server Maintenance.....	358
13.2.1	Server File Structure.....	358
13.2.2	Files for Backup	360
13.2.3	File Purging	361
13.3	General Advice on System Operations	361
13.3.1	Rebooting and Restarting - General Notes	361
13.3.2	Rebooting Machines.....	361
13.3.3	Application Restart	361
13.3.4	Viz Mosart Application Updates	361
13.3.5	Windows Updates	362
13.3.6	Viz Mosart Main/Backup Server Testing	362
13.4	Redundancy.....	362
13.4.1	Automation Server Mirroring.....	364
13.4.2	Connected GUIs	365
13.4.3	Graphical Interfaces	365
13.4.4	AV Automation - The Engineering Interface	365
13.4.5	Device-specific Redundancy	366
14	General Configuration Files.....	368
14.1	Named Overlay Graphics.....	368
14.1.1	Named CGs (Named Overlay Graphics).....	368
14.1.2	NCS Placeholders	369

14.1.3	Named CG Actions	370
14.1.4	Required Fields in XML.....	371
14.1.5	Adding the Named CG to a Template	372
14.1.6	How to obtain valid overlay graphics as a named overlay candidate	373
15	Device Connection Strings	374
15.1	Video Server, MAM and Database Connection Strings	374
15.1.1	System-level Location of Device Configuration Files	374
15.1.2	Standard Connection String Properties	375
15.1.3	Device-specific Connection String Parameters	379
15.2	Video Wall Connection Strings	416
15.2.1	Watchout Connection String.....	416
16	Device Configuration Files	420
16.1	Robotic Camera Configuration Files	420
16.1.1	Cambotics Configuration File.....	420
16.1.2	Cinneo Configuration File	421
16.1.3	Technodolly Configuration File	421
16.1.4	Panasonic Configuration File.....	422
16.1.5	Electric Friends Configuration File.....	422
16.2	Graphics Configuration Files	423
16.2.1	Pixel Power Control Center (PPCC) Configuration File	423
16.2.2	Pixel Power (Clarity) Configuration File.....	424
16.2.3	Vizrt Graphics Configuration File	426
16.2.4	Vizrt Media Sequencer VDom Logic Macros	427
16.3	Subtitling Configuration Files.....	428
16.3.1	ScreenLL Configuration File.....	428
16.4	Video Router Configuration Files.....	428
16.4.1	Miranda NVision Configuration File.....	428
16.5	Audio Mixer Configuration Files	429
16.5.1	Calrec TCP/IP Configuration File	429
16.5.2	SSL Configuration File	430

This manual is organized as follows:

- [Introduction](#)
- [About Viz Mosart](#)
- [Installation](#)
- [Manus Administrator](#)
- [Media Administrator](#)
- [Overlay Graphics Interface](#)
- [Audio Player](#)
- [Trio Interface](#)
- [ActiveX](#)
- [AV Automation](#)
- [AV Automation Device Properties](#)
- [Viz Mosart Template Database](#)
- [Maintenance](#)
- [General Configuration Files](#)
- [Device Connection Strings](#)
- [Device Configuration Files](#)

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.

1.3 Legal

Copyright © Vizrt. All rights reserved.

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

Disclaimer

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

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](#)

2.1 Naming Convention For Components

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

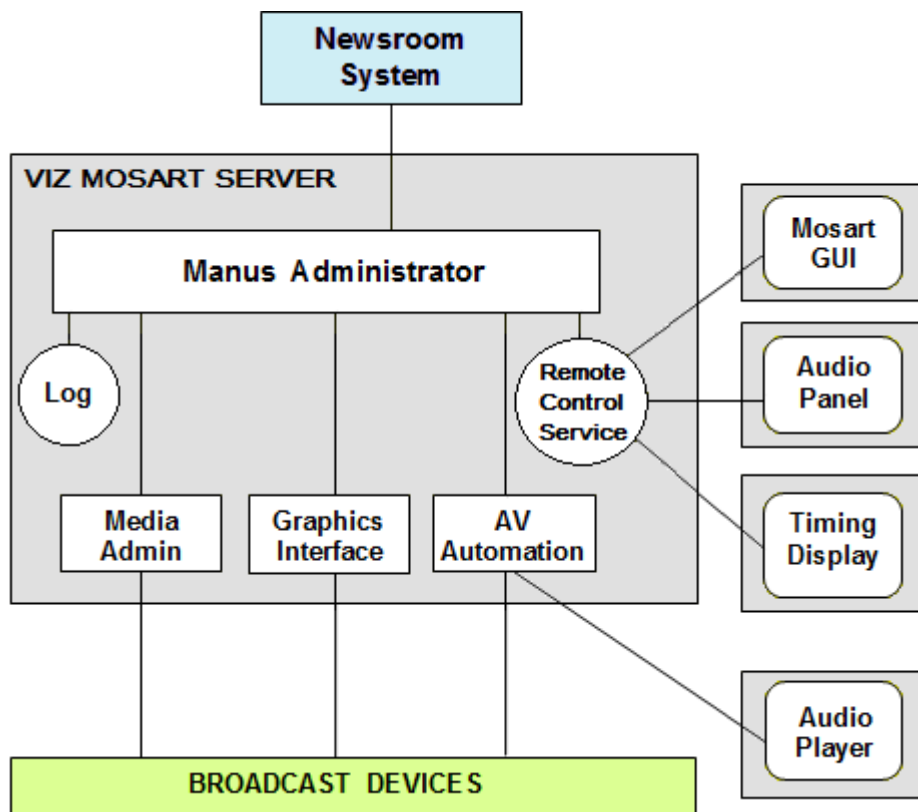
Short Name	Long Name	Description
AVA	AV Automation	Application controlling attached broadcast equipment.
GUI	Viz Mosart GUI	Main user control application.
KVM	Keyboard, Video, Mouse	Application allowing multiple computers to share the same monitor, keyboard and mouse.
Manus Admin	Manus Administrator	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 Interface instead).
URL	Uniform Resource Locator	In this context, address of web service.
Viz Mosart	Viz Mosart Newsroom Automation System	

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.

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 Interface
- AV Automation
- Log Service and Log Viewer
- Remote Control Service

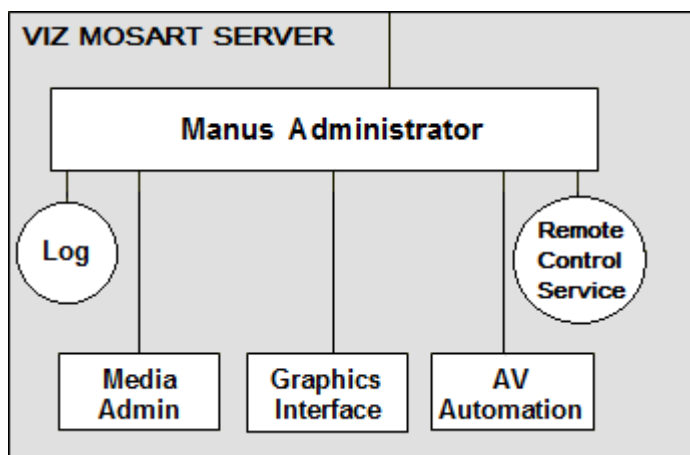
Viz Mosart Client Applications

- Viz Mosart Client
 - Viz Mosart GUI
 - Audio Panel
 - Timing Display
 - 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

2.3 Viz Mosart Server







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.

The **Viz Mosart Server** 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.

The **Viz Mosart Server** also 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.

2.3.2 Viz Mosart Server Components

Icon	Application
	Mosart Manus Administrator Executable(s): either <i>MMConsoleAdmin_2007.exe</i> (FTP, for iNEWS) or <i>MMConsoleAdmin_MOS.exe</i> (for MOS workflow Newsroom Control Systems)
	Mosart Media Administrator Executable: <i>MMMediaAdministrator.exe</i>
	Mosart AV Automation Executable: <i>MMAVAutomation.exe</i>
	Mosart Overlay Graphics Executable: <i>MMAOverLayGraphicsInterface.exe</i>

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 (NCS).
- **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:

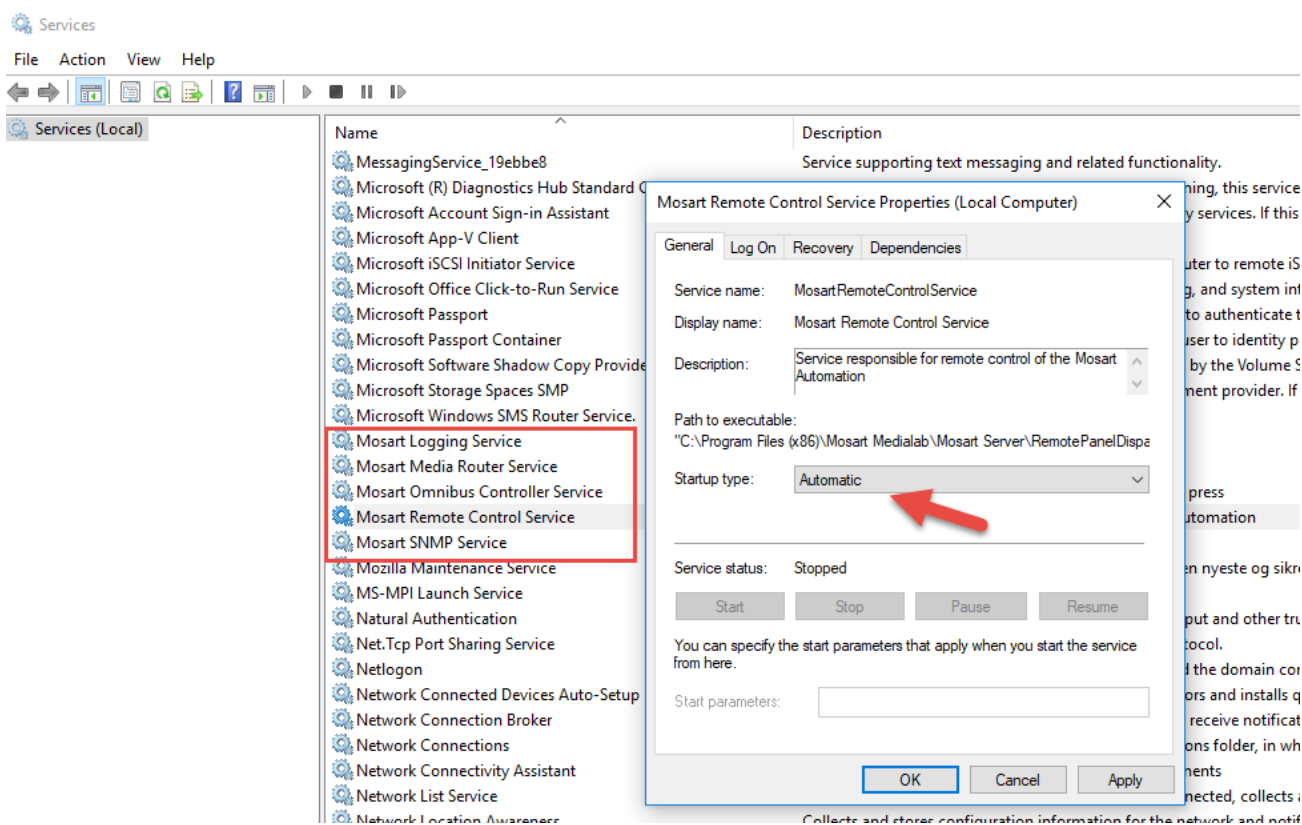
```

StartMosartServicesAndGUI.cmd
1  @echo off
2
3  @REM Assume we are installed at the default location, otherwise fix path
4  cd "C:\Program Files (x86)\Mosart Medialab\Mosart Server"
5
6  start "Manus Admin" /MIN MMConsoleAdmin_MOS.exe
7  timeout /T 1 > nul
8  start "Media Admin" /MIN MMMediaAdministrator.exe
9  timeout /T 1 > nul
10 start "AvAutomation" /MIN MMAVAutomation.exe
11 timeout /T 2 > nul
12 start "OverlayGraphics" /MIN MMOverLayGraphicsInterface.exe
13 timeout /T 1 > nul
14 cd "C:\Program Files (x86)\Mosart Medialab\Mosart GUI"
15 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:



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
```

Manus Administrator

```

Viz Mosart - ManusAdministrator 3.8.0.25778 IDLE - <no active rundown>
08:27:12 15 Loading AvConfig
08:27:12 I 3 15 ManusAdministrator_2007 ExSetAvConfig Loading AvConfig
08:27:12 I 3 15 MMAVSharedChannelTemplates getHSHvideoChannel Did not find Server A/B Roll defintion for SERVER08:27:1
2 15 Did not find Server A/B Roll defintion for SERVER
08:27:12 15 Did not find Server A/B Roll defintion for BACK UP SERVER
08:27:12 I 3 15 MMAVSharedChannelTemplates getHSHvideoChannel Did not find Server A/B Roll defintion for BACK UP SERVE
R
08:27:16 16 Status: stopped intime=0, storyTime=0, itemTime=0, tset=TVA 2017
settings
Checking for newsroomsettings version...
Newsroomsettings version OK.
08:27:54 I 3 18 NewsroomSettingsRepository Initialize Newsroomsettings configurations succesfully loaded into th
e repository.08:27:54 18 Newsroomsettings configurations succesfully loaded into the repository.
NEWSROOMSETTINGS: c:\channeltemplates\newsroomsettings.xml
08:27:54 18 Manus Directory is c:\manus
08:27:54 I 3 18 ManusAdministrator_2007 ManusDirectory Manus Directory is c:\manus
08:27:54 I 3 15 ManusAdministrator_2007 ExSetAvConfig Loading AvConfig08:27:54 15 Loading AvConfig
08:27:54 15 Did not find Server A/B Roll defintion for SERVER
08:27:54 I 3 15 MMAVSharedChannelTemplates getHSHvideoChannel Did not find Server A/B Roll defintion for SERVER
08:27:54 15 Did not find Server A/B Roll defintion for BACK UP SERVER
08:27:54 I 3 15 MMAVSharedChannelTemplates getHSHvideoChannel Did not find Server A/B Roll defintion for BACK UP SERVE
R
08:27:54 I 3 15 ManusAdministrator_2007 ExSetChannelTemplates ChannelTemplates loaded, current studiosetup is TVA 2017
08:27:54 15 ChannelTemplates loaded, current studiosetup is TVA 2017

```

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

The Manus Administrator handles the connection to the NCS, and executes commands at the appropriate time according to GUI input and template configuration.

The 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.

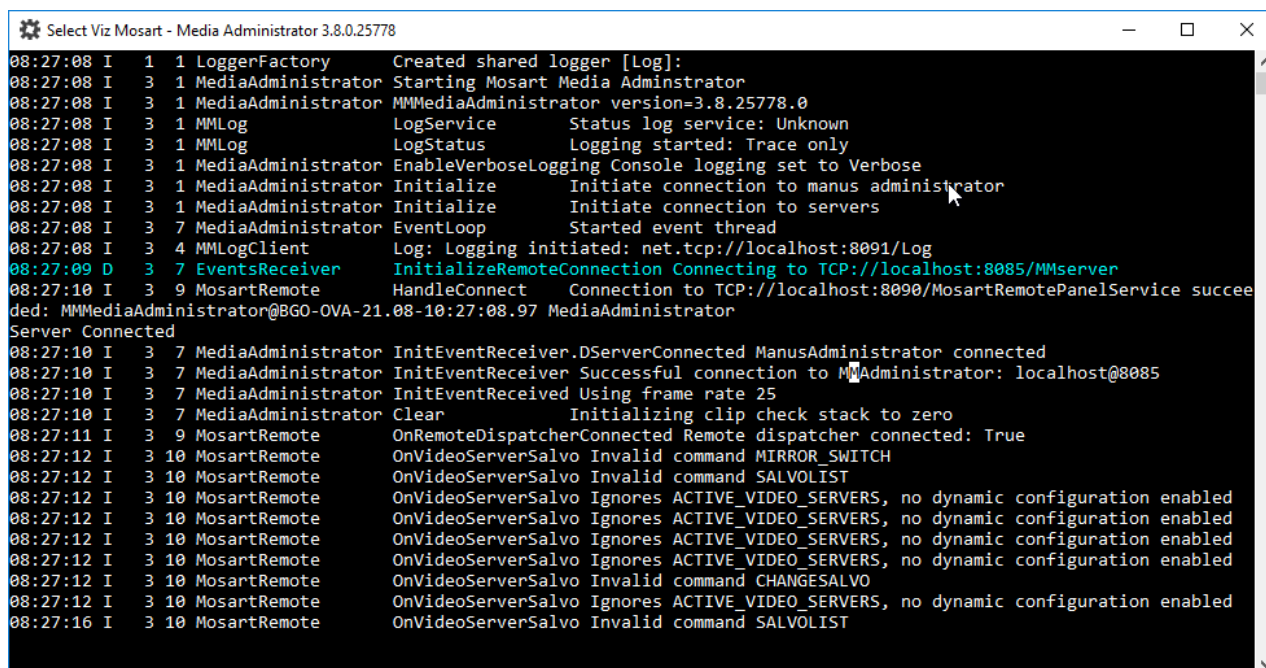
Variants

Manus Administrator has variants with slightly different settings that suit different broadcast environments. During installation, select the Manus Administrator appropriate to your environment:

- **MMConsoleAdmin_2007** for **iNews FTP Newsroom** connections
- **MMConsoleAdmin_MOS** for **MOS Newsroom** connections

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

Media Administrator



```

Select Viz Mosart - Media Administrator 3.8.0.25778
08:27:08 I 1 1 LoggerFactory Created shared logger [Log]:
08:27:08 I 3 1 MediaAdministrator Starting Mosart Media Administrator
08:27:08 I 3 1 MediaAdministrator MMediaAdministrator version=3.8.25778.0
08:27:08 I 3 1 MMLog LogService Status log service: Unknown
08:27:08 I 3 1 MMLog LogStatus Logging started: Trace only
08:27:08 I 3 1 MediaAdministrator EnableVerboseLogging Console logging set to Verbose
08:27:08 I 3 1 MediaAdministrator Initialize Initiate connection to manus administrator
08:27:08 I 3 1 MediaAdministrator Initialize Initiate connection to servers
08:27:08 I 3 7 MediaAdministrator EventLoop Started event thread
08:27:08 I 3 4 MMLogClient Log: Logging initiated: net.tcp://localhost:8091/Log
08:27:09 D 3 7 EventsReceiver InitializeRemoteConnection Connecting to TCP://localhost:8085/MMserver
08:27:10 I 3 9 MosartRemote HandleConnect Connection to TCP://localhost:8090/MosartRemotePanelService succeeded: MMediaAdministrator@BGO-OVA-21.08-10:27:08.97 MediaAdministrator
Server Connected
08:27:10 I 3 7 MediaAdministrator InitEventReceiver.DServerConnected ManusAdministrator connected
08:27:10 I 3 7 MediaAdministrator InitEventReceiver Successful connection to ManusAdministrator: localhost@8085
08:27:10 I 3 7 MediaAdministrator InitEventReceived Using frame rate 25
08:27:10 I 3 7 MediaAdministrator Clear Initializing clip check stack to zero
08:27:11 I 3 9 MosartRemote OnRemoteDispatcherConnected Remote dispatcher connected: True
08:27:12 I 3 10 MosartRemote OnVideoServerSalvo Invalid command MIRROR_SWITCH
08:27:12 I 3 10 MosartRemote OnVideoServerSalvo Invalid command SALVOLIST
08:27:12 I 3 10 MosartRemote OnVideoServerSalvo Ignores ACTIVE_VIDEO_SERVERS, no dynamic configuration enabled
08:27:12 I 3 10 MosartRemote OnVideoServerSalvo Ignores ACTIVE_VIDEO_SERVERS, no dynamic configuration enabled
08:27:12 I 3 10 MosartRemote OnVideoServerSalvo Ignores ACTIVE_VIDEO_SERVERS, no dynamic configuration enabled
08:27:12 I 3 10 MosartRemote OnVideoServerSalvo Ignores ACTIVE_VIDEO_SERVERS, no dynamic configuration enabled
08:27:12 I 3 10 MosartRemote OnVideoServerSalvo Invalid command CHANGESALVO
08:27:12 I 3 10 MosartRemote OnVideoServerSalvo Ignores ACTIVE_VIDEO_SERVERS, no dynamic configuration enabled
08:27:16 I 3 10 MosartRemote OnVideoServerSalvo Invalid command SALVOLIST

```

The Media Administrator (MMediaAdministrator) 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 NCS. 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.

Overlay Graphics Interface

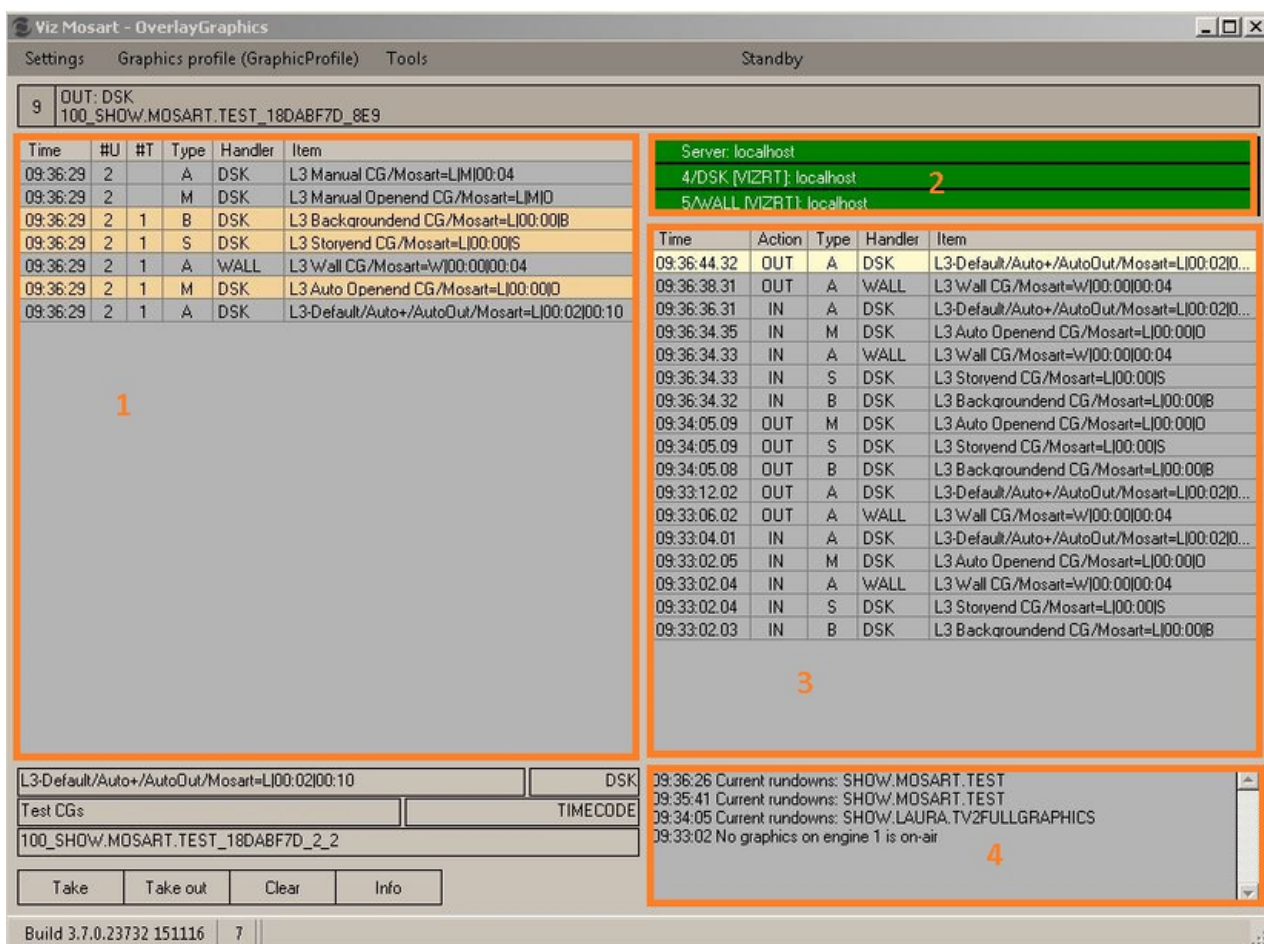
This is used for controlling and monitoring overlay graphics for graphics engines.

The two overlay graphics interfaces are:

- **OverlayGraphics** Interface (recommended).

- **Trio Interface** (to be 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](#).

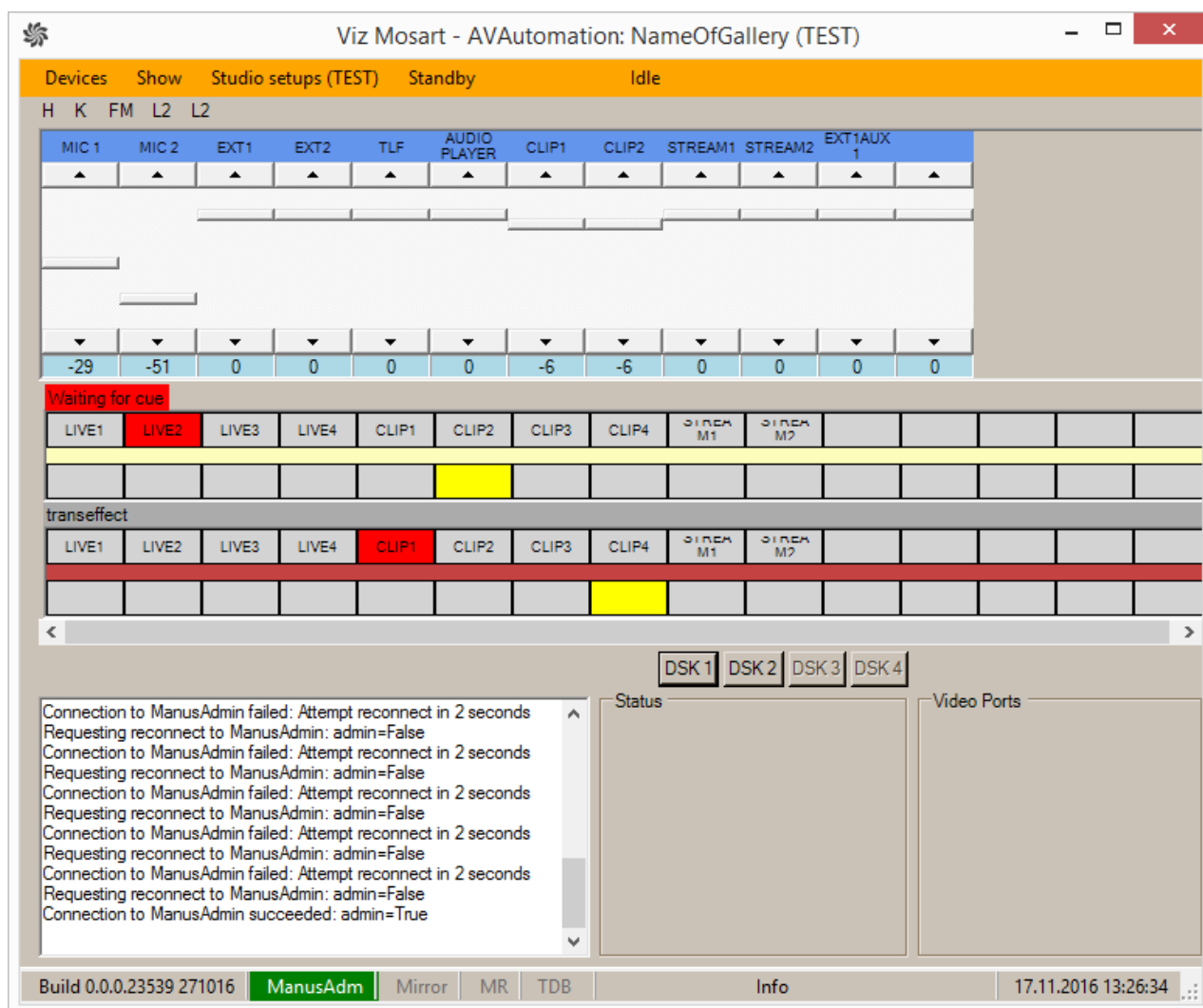


The [Overlay Graphics Interface](#) is used for controlling and monitoring the overlay graphics for all Viz Mosart approved graphics engines. Configure it using [Overlay Graphics Configuration](#). For a list of supported graphics devices, see [Overlay Graphics Types](#).

Trio Interface

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

AV Automation



AV Automation (MMAVAutomation) controls all Viz Mosart broadcast devices with the exception of Overlay Graphics engines. Full-screen graphics is 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 NCS.

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.

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.

Index	Log time	Hostname	Log module	Method	Log element	L	Log text
4560	2014-02-17T16:37:29...	SASTVQ01	MMAVAutomation	CueChannels	VideoSwitcher	3	CueDelay set to 1000 ms (25 frames, min. cue delay)
4561	2014-02-17T16:37:29...	SASTVQ01	MMConsoleAdmi..	SwitchBackGrou...	ManusAdministr	3	Slug=CAM BLACK, Story=1650 CLAPPER-VIZ, Templ
4562	2014-02-17T16:37:30...	SASTVQ01	MMAVAutomation	CueAccessoriesIt..	AvAutomation	2	PREVIEW ACCESSORIESRouter set matrix 0 level
4563	2014-02-17T16:37:30...	SASTVQ01	MMAVAutomation	PrepareGraphics	Graphics	3	Got Vizcommand from dataelement ' CLAPPER / BRIS
4564	2014-02-17T16:37:30...	SASTVQ01	MMAVAutomation	LoadGraphics	Graphics	3	Graphics loaded ' CLAPPER / BRISBANE (TVQ) UPDA1...
4565	2014-02-17T16:37:30...	SASTVQ01	MMAVAutomation	SetCrossPoint	VideoSwitcher	3	VIZ B FILL (20) ME2/BusB
4566	2014-02-17T16:37:30...	SASTVQ01	MMAVAutomation	ScheduleElement	Graphics	3	Scheduling; viz=viz_mosart_b, command=cue, show=/...
4567	2014-02-17T16:37:32...	SASTVQ01	MMAVAutomation	DoEarlyTakeInC...	GeneralInfo	3	EarlyCommand DIRECTTAKE - 800(STBY ALL MICs
4568	2014-02-17T16:37:32...	SASTVQ01	MMAVAutomation	TakeAccessories..	AvAutomation	2	PROGRAM ACCESSORIESRouter set matrix 0 level
4569	2014-02-17T16:37:32...	SASTVQ01	MMAVAutomation	TakeGraphics	Graphics	3	Graphics taken ' CLAPPER / BRISBANE (TVQ) UPDATE
4570	2014-02-17T16:37:32...	SASTVQ01	MMAVAutomation	RippleGraphicsGr	AvAutomation	3	Graphics ripple 2: current=2, next=2
4571	2014-02-17T16:37:32...	SASTVQ01	MMAVAutomation	MMAudio	AudioMixer	3	Post onair channels: SVR D ST, READER1 MAIN, SVR

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.

See Also

- [Manus Administrator](#)
- [AV Automation](#)
- [AV Automation Device Properties](#)
- [Media Administrator](#)

2.4 Viz Mosart Client

The **main components** that make up Viz Mosart Client are:

- **Viz Mosart GUI:** Fullscreen Windows application used to control and monitor the Viz Mosart rundown.
- **Audio Panel:** Fullscreen Windows application used to control and monitor software audio faders.
- **Timing Display:** Fullscreen Windows application used to display various timing information from the Viz Mosart rundown.
- **Audio Player:** Used to play out audio files located on the file system.

Additional Viz Mosart Client applications:

- **ActiveX**
- **iNews Timer**

 **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



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.

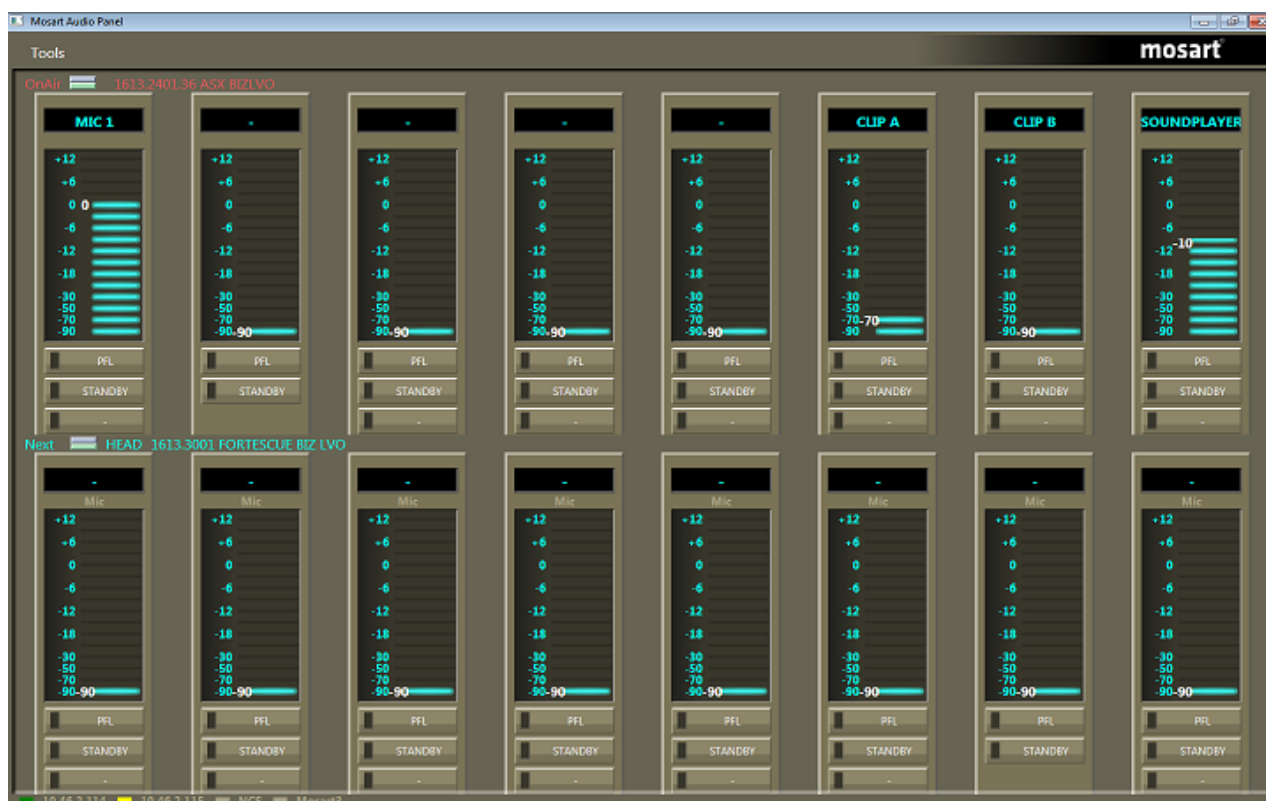
The Viz Mosart GUI is the main control interface for the Viz Mosart Server. The director can run the production from a single key press on the workstation keyboard.



Alternate configurations may also host a fader panel and a [Timing Display](#). 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 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)

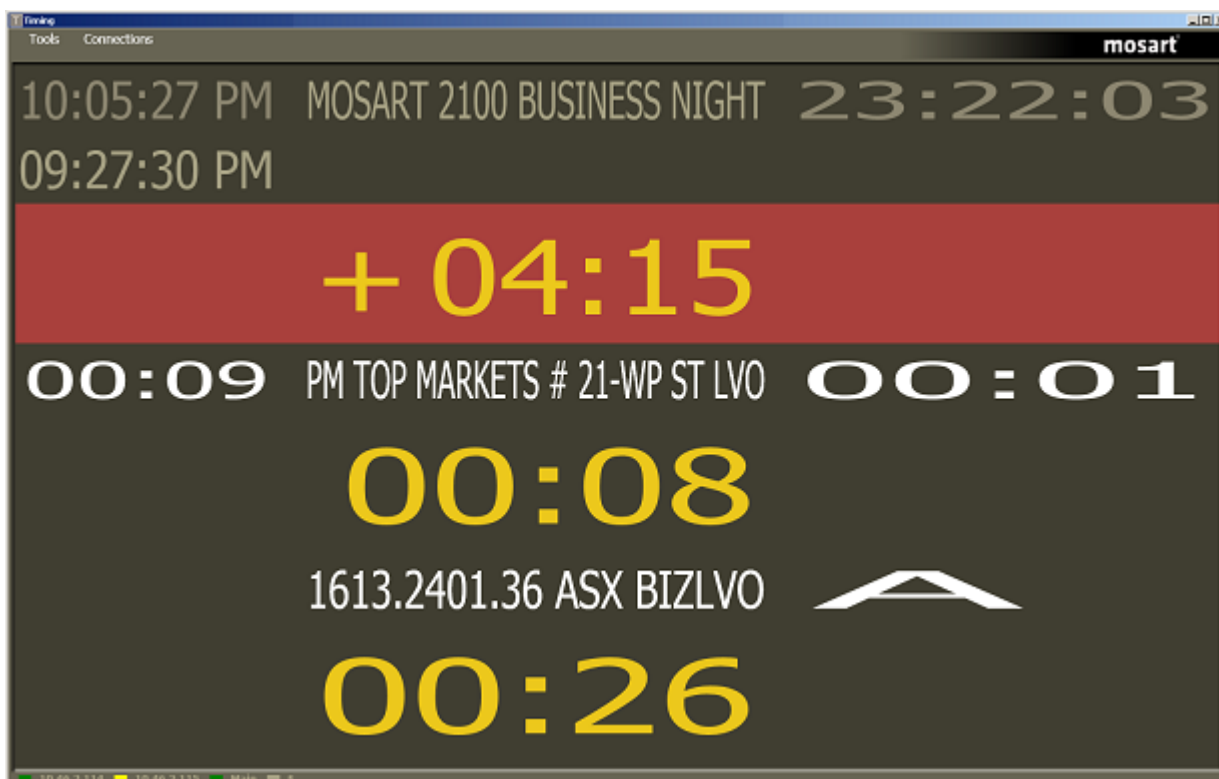


Mosart Audio Panel

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*.

2.4.3 Timing Display



The Timing Display (WPFTimingInfo) provides timing information to the studio control room and the studio floor.

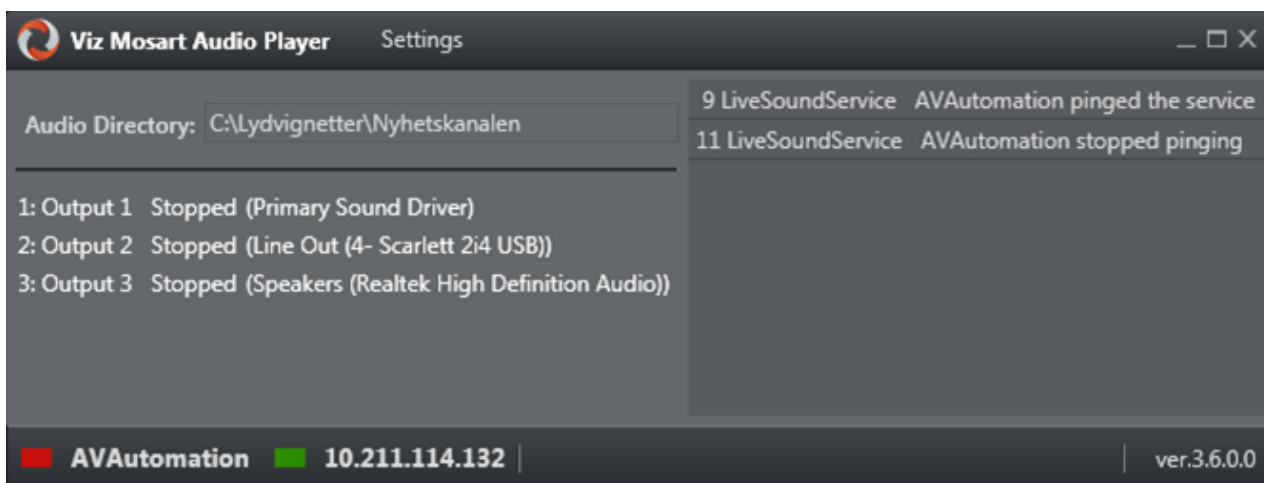
The Timing Display is synchronized to the current rundown in the Viz Mosart GUI. You can run as many customized Timing Display applications as you need for the production.

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

2.4.4 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.

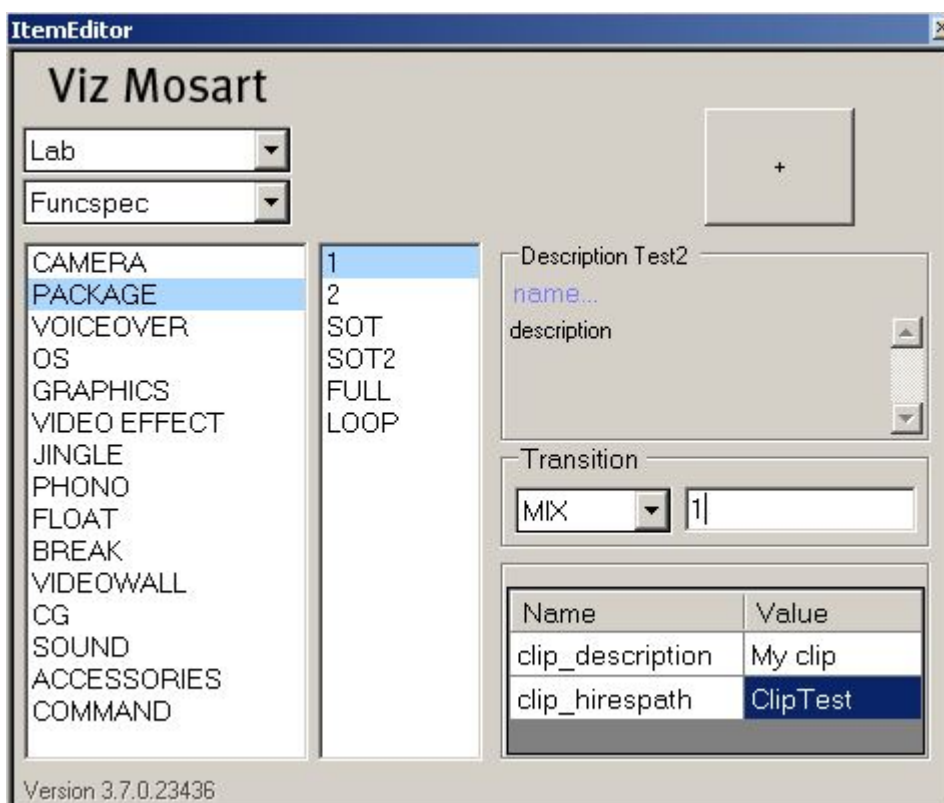
Note: A broadcast sound card is required for audio output. If you're running the Audio Player on a Windows Server, you must install Windows Audio features on the machine in order for the various audio formats to work.



Note: The Audio Player replaces the discontinued Soundfile Player.

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

2.4.5 ActiveX



The ActiveX is used to insert Viz Mosart template information into the Newsroom System script. It's compatible with any newsroom system that allows ActiveX plugins.

The ActiveX list displays the first template name result it finds under NCS tags in Newsroom Settings. The user selects a Viz Mosart template type and variant, which can then be dragged into an entry in the Viz Mosart rundown. When the script is saved, information is pushed to Manus Administrator and the active rundown in the Viz Mosart GUI.

The ActiveX is installed as a standalone component and is independent of the version of Viz Mosart Server used.

Viz Mosart has its own ActiveX NCS plugin. The plugin lets users insert Viz Mosart commands to the scripts from a list of available Viz Mosart templates.

The templates and transition effects that are shown in the ActiveX may also depend on the value of **Studio Setup** in [AV Automation > Audio and Video Setup](#).

When a particular **Studio Setup** has been selected in the ActiveX (in the second dropdown box), only effects with this studio setup (as defined in [AV Automation > Audio and Video Setup](#)) will be available in the Transition dropdown. Effects with no studio setup defined are global effects and always appear in the **Transition** dropdown.

2.4.6 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.

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
- **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

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](#)
 - [Version Numbering](#)
 - [Manual Installation](#)
 - [Viz Mosart Installation Administrator](#)
-

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](#)
 - [To determine which Microsoft .NET Frameworks are installed](#)
 - [Installing the Microsoft .NET Framework](#)
 - [Microsoft Visual C++ Redistributable Package \(x86\)](#)
- [Hardware Requirements](#)
 - [Viz Mosart Automation Servers](#)
 - [Minimum Recommended Hardware Configuration](#)
 - [Recommended Performance-related Components](#)
 - [Viz Mosart GUI PCs](#)
 - [Minimum Recommended Hardware Configuration](#)
 - [Recommended Performance-related Components](#)
 - [Viz Mosart Audio Player](#)
 - [Viz Mosart Timing Display and Audio Panel PCs](#)
 - [Viz Mosart Shared Template Database](#)
 - [Recommended MySQL Software](#)
 - [Minimum Recommended Hardware Configuration](#)
 - [Recommended Performance-related Components](#)
 - [Pre-Installation Tasks](#)
 - [Connections](#)

3.1.1 System Requirements

The following apply all machines running Viz Mosart software:

Prerequisite	Target	Description
Microsoft Windows Server 2008 R2 or later	Viz Mosart Server	Operating system
Microsoft Windows 10	Viz Mosart Client	Operating system. Microsoft Windows 7 Professional may also be used, but this version is no longer supported by Microsoft.
Microsoft .NET Framework	All	See the Viz Mosart Release Notes for the applicable Microsoft .Net version. Most of the Viz Mosart installers abort the installation process if .NET 4.5 is not detected.
Microsoft Visual C++ Redistributable (vcredist_x86.exe)	Viz Mosart Server	See the Viz Mosart Release Notes for the applicable Microsoft Visual C++ Redistributable Package.
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	<p>If MIDI devices are connected via direct conversion serial adapters, Windows Audio feature must be enabled on the server.</p> <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;"> <p> 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.</p> </div>
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.

Prerequisite	Target	Description
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.

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 (x86)

A *Microsoft Visual C++ Redistributable Package (x86)* 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.

Obtain the applicable Visual C++ Redistributable Package (x86) installer from the Microsoft Download Center.

Installing Microsoft Visual C++ Redistributable Package:

1. When downloading, please ensure that you select the **32-bit (x86)** version.
2. Run the installer 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
- 1280 x 1024 display adapter
- 8 GB Memory

Recommended Performance-related Components

- 2 Processors with performance as 2.0 GHz dual-core and with 4 MB L2 cache
- Minimum 50 GB free storage space
- Windows Server 2008, 2012 or 2016 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](http://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-MIDI
- 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 PC's: 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 64 bit operating system with applicable .NET Framework
- Gigabit Ethernet port
- Approved video adapter with resolution 1920 x 1080
- 8 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 2.0 GHz Dual-core
- 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 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.

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.

⚠ 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.

3.2 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.

3.2.1 Viz Mosart Installer Files

Installer	Machine Role	Description
VizMosartActiveX	NCS Client	Installer containing the <i>ActiveX</i> , used for displaying Viz Mosart Templates in the NCS client.
VizMosartAudioPanel	Audio Client (GUI or standalone PC)	Installer containing the <i>Audio Panel</i> , an application for controlling audio faders, both on air and in preview.
VizMosartAudioPlayer	Viz Mosart Server or Audio Client	Installer containing the <i>Audio Player</i> , 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 <i>Viz Mosart GUI</i> application, which is the Viz Mosart user interface. This is available both as .msi and .exe.
Viz MosartiNEWSTimer	iNews Client	Installer containing the <i>iNews Timer</i> , an application used to create accurate timing of the production in iNews.
Viz MosartInstallationAdministrator	All Viz Mosart PCs	Installer containing the Viz Mosart Installation Administrator , an (beta version, feedback is welcomed) installation assistant for Viz Mosart software.

Installer	Machine Role	Description
Viz MosartLogViewer	Viz Mosart Server	Installer containing the <i>Log Viewer</i> , an application used to decipher Viz Mosart logs.
Viz MosartMediaRouter	Viz Mosart Database Server	Installer containing the Media Router. For more details, see Media Router and Mosart Template Database Admin Guide.
Viz MosartOmnibusController	Viz Mosart Server	Installer containing the <i>Omnibus Controller</i> , an application used for Omnibus Server control.
Viz MosartServer	Viz Mosart Server	Installer containing the various Viz Mosart Server components; Manus Administrator, Media Administrator, AV Automation, and Overlay Graphics Interface.
Viz MosartSNMPService	Viz Mosart Server	Installer containing the <i>SNMP Service</i> , used for monitoring the Viz Mosart Server suite via SNMP.
Viz MosartTestSuite	Viz Mosart Server	Installer containing the <i>Test Suite</i> , test utilities used when installing Viz Mosart.
Viz MosartTimingDisplay	Timing Client	Installer containing the <i>Timing Display</i> , used to provide timing information to the studio control room and studio floor.

3.3 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 **3.4.5.12345**)

Where:

- Major.Minor.Revision denotes the Viz Mosart Version Number

- Build denotes a patch number

Example: `Installers\3.4.5.12345\VizMosartServer-3.4.5.12345.msi`

Here we can see that the computer has a folder for the Viz Mosart 3.4.5.12345 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.

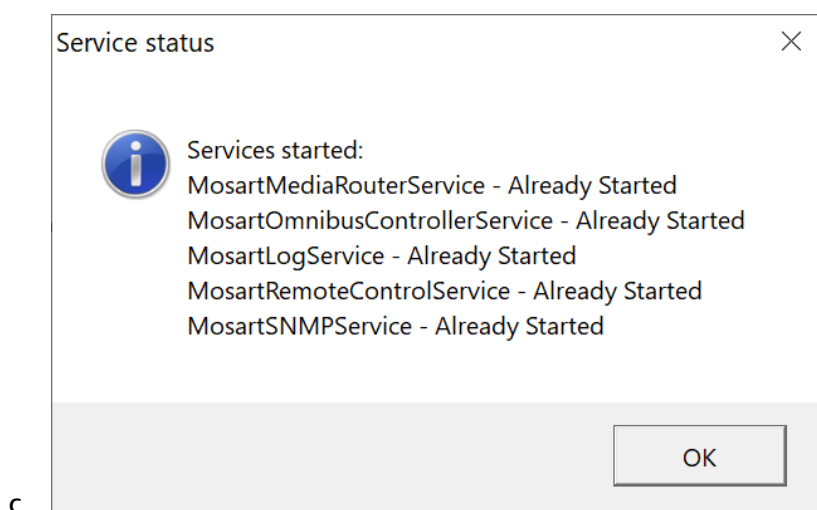
3.4 Manual Installation

When installing or upgrading Viz Mosart, run the [Viz Mosart Installation Files](#) manually from Windows Explorer.

✔ **Tip:** Vizrt provides a beta software utility [Viz Mosart Installation Administrator](#) that can speed up installation and upgrading of your Viz Mosart system. Refer to section [Viz Mosart Installation Administrator](#).

3.4.1 To Perform a Manual Installation

1. Download all relevant [Viz Mosart Installation Files](#) to the preferred location. The default location is `C:\Mosart\Installers`. You are advised to make a sub-directory for the installers for a particular version/build containing all the MSI installer-files and any other supplemental files. This directory should be named using the Release.Version.Patch.Build naming convention, for example: `C:\Mosart\Installers\3.4.5.12345` for Viz Mosart 3.4.5 build 12345.
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 [Viz Mosart Backup](#) below.
3. Double-click an **MSI installation file**, and follow the prompts to complete installation.
4. Repeat the above step for all relevant installation files.
5. You need to start a set of Windows services to make Viz Mosart run properly. These are automatically started when the computer is started.
 - a. The safest is to reboot the computer to verify that the automatic start of the services is working.
 - b. Service names will be displayed as in the screenshot below:



3.4.2 Viz Mosart Backup

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, e.g. Mosart Server, Mosart GUI, etc.

Windows registry settings for

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

3.5 Viz Mosart Installation Administrator

Viz Mosart has its own installation utility, Viz Mosart Installation Administrator (MIA). It can save a lot of time when installing and upgrading a Viz Mosart setup.

Note: The Viz Mosart Installation Administrator is **beta** software. Please provide any feedback to your Vizrt Support organization.

This section contains the following topics:

- [Installing the Viz Mosart Installation Administrator \(MIA\)](#)
- [Viz Mosart Installation Administrator Interface](#)
- [Semi-Automated Installation](#)

- Settings
- Take Snapshot
- Start Mosart Application
- Backup Files
- Stop Services
- Uninstall Mosart
- Install Mosart
- Start Services

3.5.1 Installing the Viz Mosart Installation Administrator (MIA)



Mosart
Installation
Administrator

You need to first download and install the Viz Mosart Installation Administrator (MIA) before working with Viz Mosart system installations.

To Install the Viz Mosart Installation Administrator

1. Download the `MosartInstallationAdministratorInstaller.<version>.msi` to the preferred location.
Default location is `C:\Mosart\Installers`.
2. Double-click the installation file, and follow the prompts to complete installation. Repeat this installation process on all PCs where you plan to install any Viz Mosart Applications.

⚠ Warning: The directory that holds the Viz Mosart installers must be named using the Release.Version.Patch.Build naming convention.

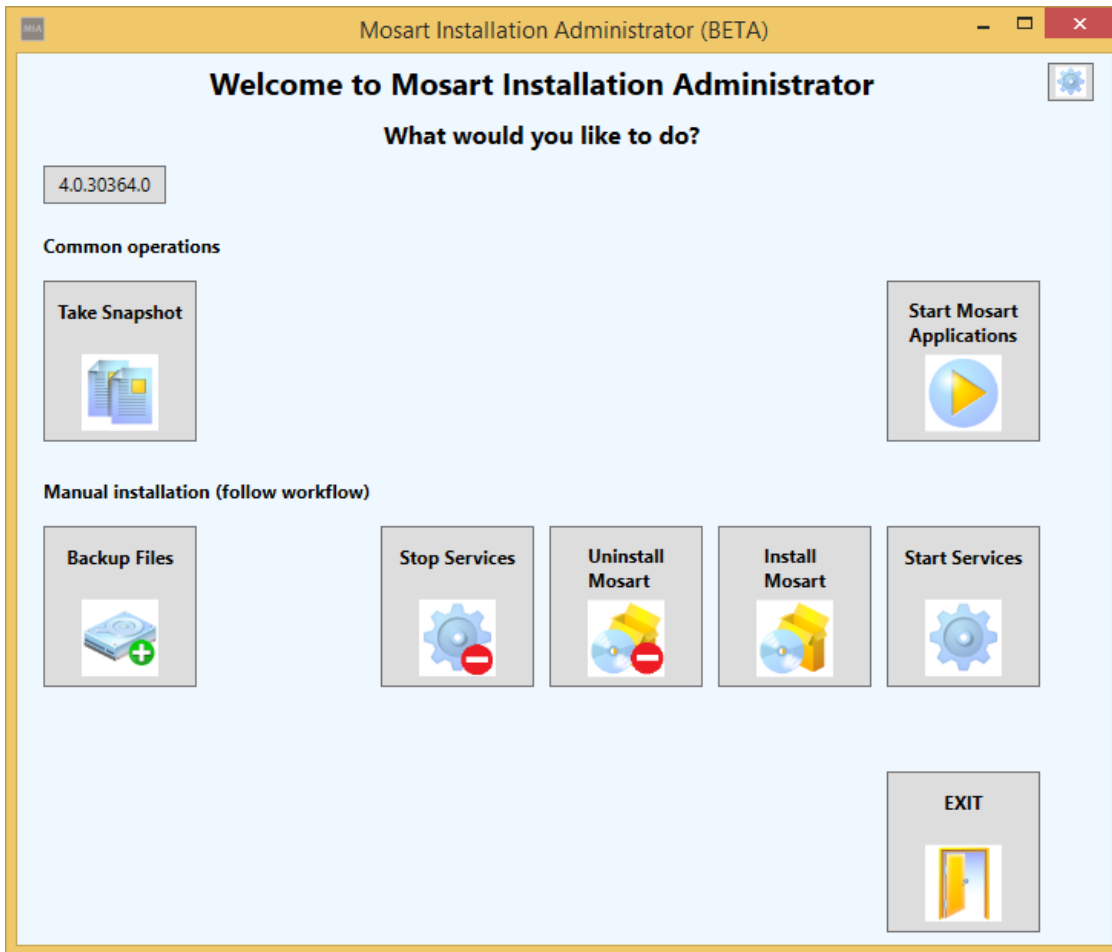
For example:

C:\Mosart\Installers\3.4.5.12345 for *Viz Mosart 3.4.5 build 12345*, or

C:\Mosart\Installers\4.0.0 for *VizMosart-4.0.0* etc.

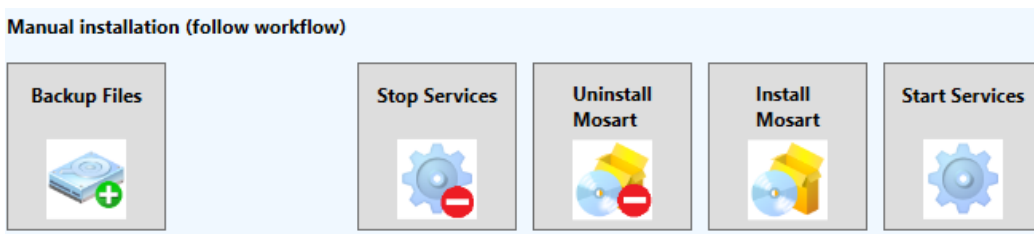
If this rule is not followed, MIA will not display the installation directory.

3.5.2 Viz Mosart Installation Administrator Interface



The MIA utility assists in installation and upgrade of Viz Mosart. The various tools are described below.

3.5.3 Semi-Automated Installation



The Viz Mosart Installation Administrator can run a semi-automated installation process. The buttons above are placed in the order you would normally progress through when upgrading an existing Viz Mosart installation.

New Installs

If Viz Mosart has not been installed before and installer files are in the correct installation location, begin from the **Install Mosart** button.

Upgrades

Follow a five-step process running from left to right when performing a semi-automated upgrade.

Note: If Viz Mosart has previously been manually installed on the system you are using *without* using the Viz Mosart Installation Administrator, the user *must* uninstall Viz Mosart manually through Windows' *Uninstall* or *change a program* before using the installation assistant. This is due to a filename mismatch in the Windows Registry and installation assistant that makes MIA unable to locate previous manually installed Viz Mosart Applications.

3.5.4 Settings



Settings are accessed by clicking on the **cog-wheel** icon, at the top right of MIA.

The screenshot below shows default settings.

Settings

Backup path

Installers path

Send mail to

Install timeout (sec)

Correct Mosart version numbers

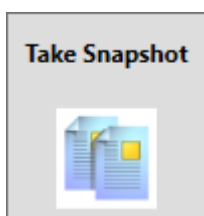
Start Mosart applications after upgrade

OK

Note: Any paths not found in MIA settings can be manually changed by editing the file:
 C:\Program Files (x86)\Mosart Medialab\Mosart Installation
 Administrator\MosartInstallationAdministrator.exe.config

- **Backup path:** Defines the path to where the installation assistant places backup files.
- **Installers path:** Defines the path to where the installation assistant should look for installation files.
- **Send mail to:** Defines the e-mail address where error reports should be sent when troubleshooting.
- **Install timeout (sec):** Viz Mosart Services attempt to start/stop for the defined number of seconds.

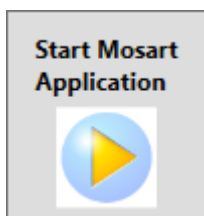
3.5.5 Take Snapshot



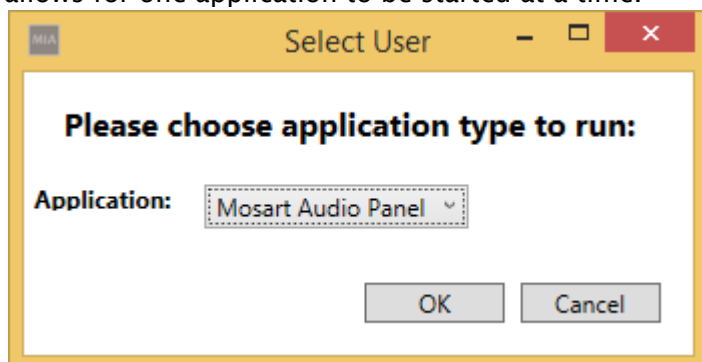
- A snapshot of current Viz Mosart logs and user configuration files. These are helpful for Viz Mosart Support to assist in diagnosing any issues you encounter.
- You must specify a date and time in the past for the snapshot.
- The following files are collected:
 - *C:\MMLogs*
 - *C:\MMLogs\MosLog*
 - *C:\MMLogs\AsRunLog*
 - *%appdata%\..\Local\Mosart_Medialab\.\user.config*

Once complete, the MIA copies the files to the configured Backup path (see [Settings](#)) and compresses the files in .ZIP format. An email interface appears with the .ZIP attached for the user to send the file to Viz Mosart Support.

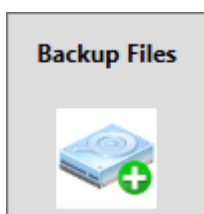
3.5.6 Start Mosart Application



- MIA provides an quick launch of any installed Viz Mosart application. This feature only allows for one application to be started at a time.

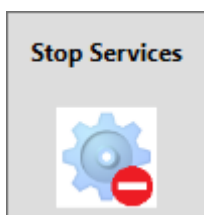


3.5.7 Backup Files



- This tool backs up all user.config files from folders with current version numbers and all files located in C:\channel\templates.
- The files are copied to the specified Backup path (see [Settings](#)) with the Viz Mosart version/username as folder name.

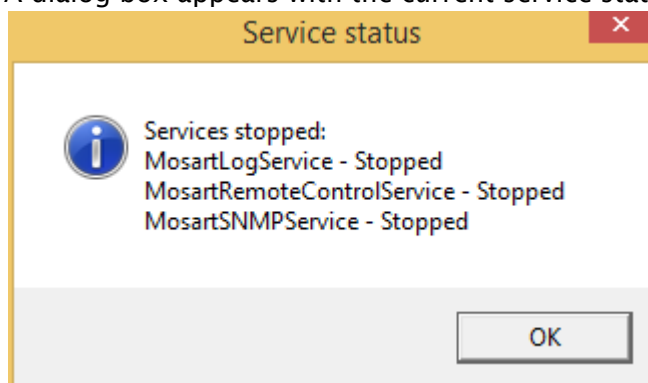
3.5.8 Stop Services



- Pressing this button stops all Viz Mosart services installed, with a timeout of six seconds.

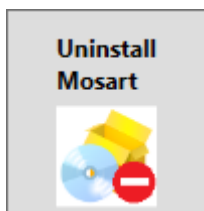
✓ **Tip:** The timeout length can be adjusted in the installation assistant [Settings](#).

- A dialog box appears with the current service status:

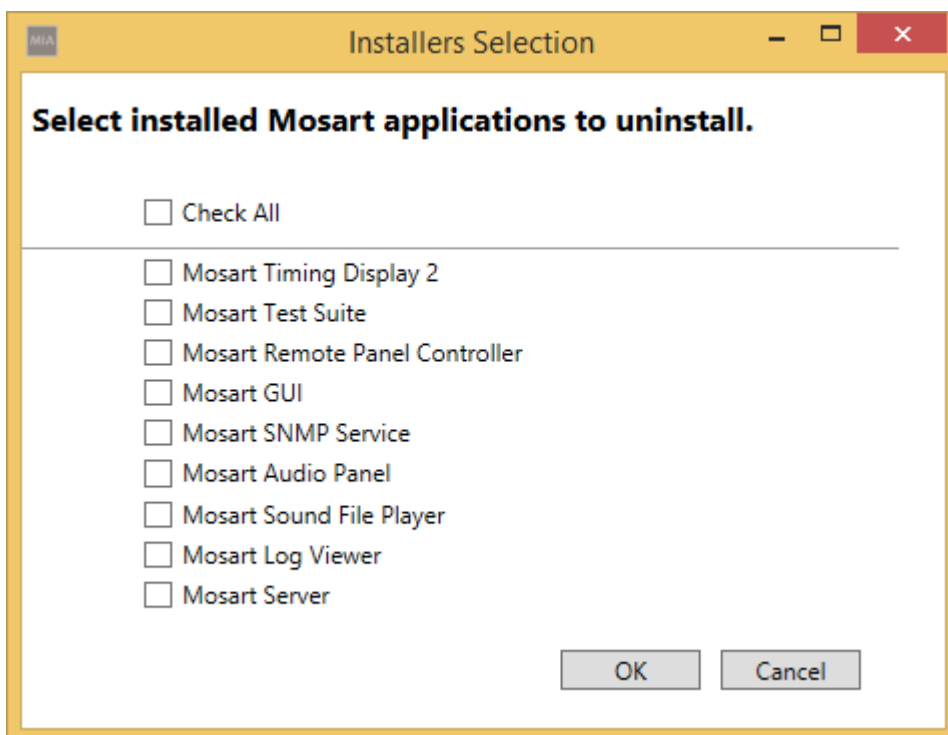


- Repeat as necessary.
Alternatively, perform a manual stop within Windows Services.

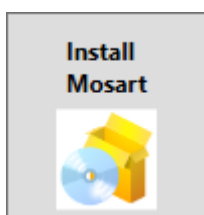
3.5.9 Uninstall Mosart



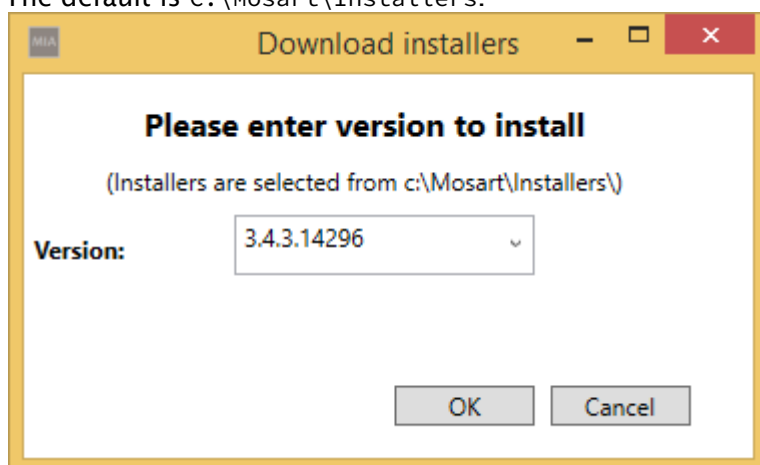
- Pressing this button displays all installed Viz Mosart Applications. You can then select one or all of the applications to uninstall.
- The MIA does not delete desktop shortcuts. Existing shortcuts can be reused on completion of an upgrade.



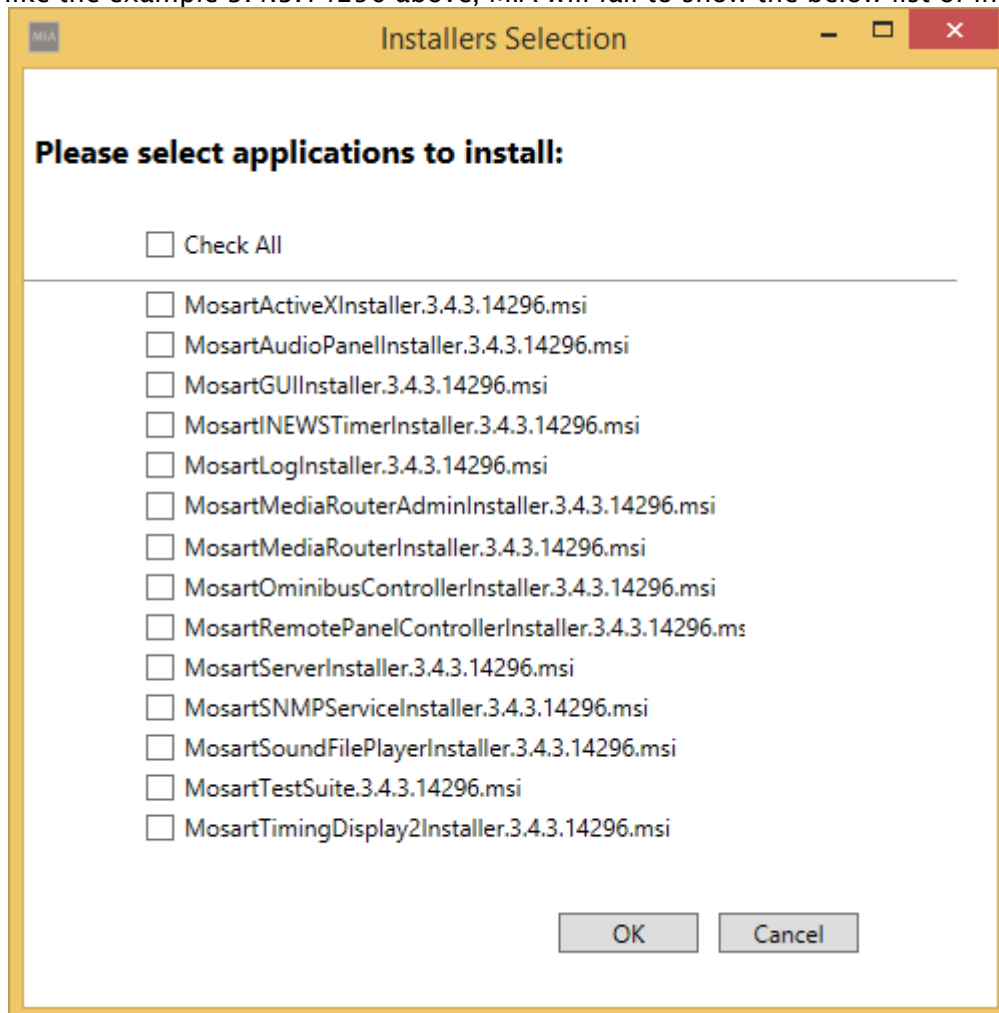
3.5.10 Install Mosart



1. Pressing this button displays all available Viz Mosart installers found in the configured Installers path (see [Settings](#)).
The default is C:\Mosart\Installers.




2. Select the Viz Mosart version you want to install, and click **OK**.
The next screen shows applications found in the installation folder. You can select one or all of the installers to run.
Note that if the installation folder is not named purely with the Viz Mosart version number, like the example 3.4.3.14296 above, MIA will fail to show the below list of installers.



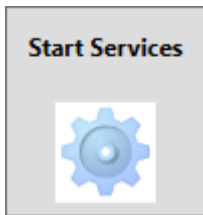
 **Tip:** For details on how to locate/change your installation folder path, see [Settings](#).

3. MIA then performs a silent installation of the selected applications, no user interaction is required.

The installation assistant does not create or delete desktop shortcuts. Any existing shortcuts from a previous version can be reused on completion of the upgrade.

 **Note:** Viz Mosart Server components must be selected on the first installation. This is done by creating shortcuts from C:\Program Files (x86)\Mosart Medialab\Mosart Server. Select the components relevant to your broadcast environment.

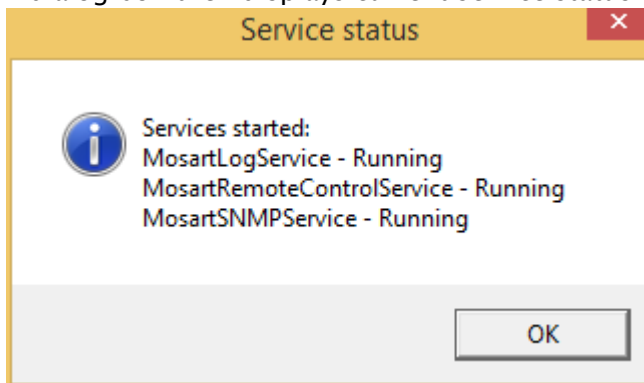
3.5.11 Start Services



- This button starts all installed Viz Mosart services, with a default timeout of six seconds.

✓ **Tip:** The timeout length can be adjusted in the installation assistant [Settings](#).

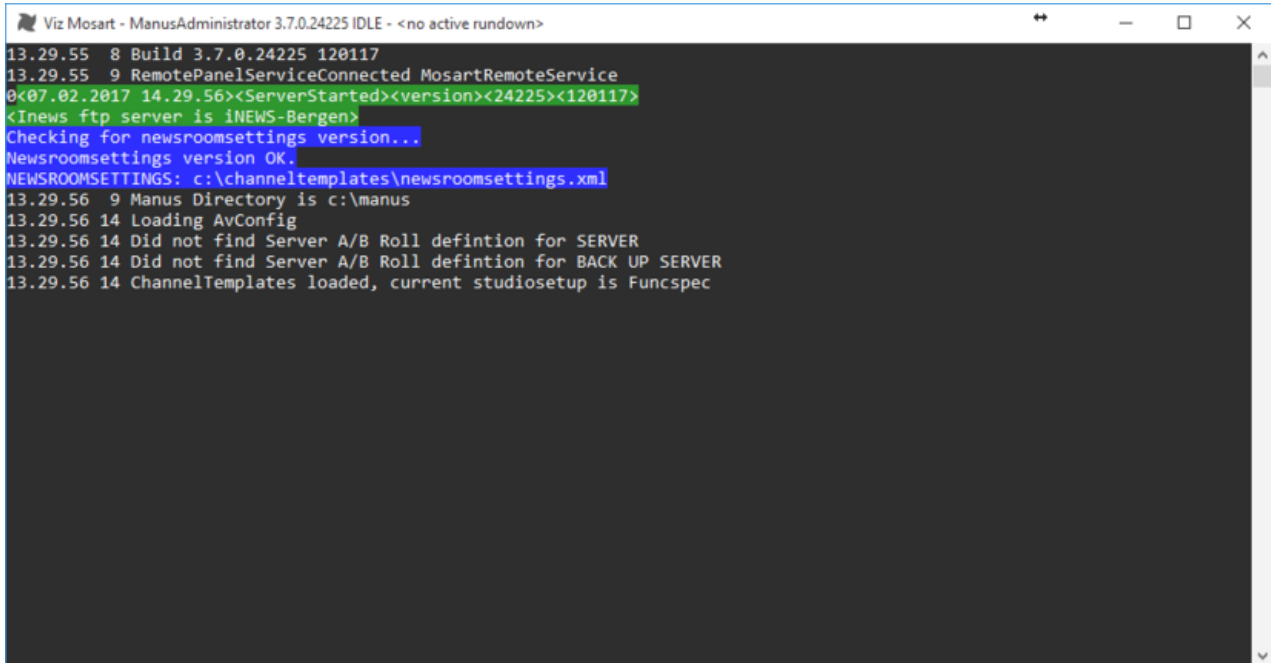
- A dialog box then displays current service status:



- Repeat as necessary, or perform a manual start within Windows Services.

4 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.



```
Viz Mosart - ManusAdministrator 3.7.0.24225 IDLE - <no active rundown>
13.29.55 8 Build 3.7.0.24225 120117
13.29.55 9 RemotePanelServiceConnected MosartRemoteService
0<07.02.2017 14.29.56><ServerStarted><version><24225><120117>
<Inews ftp server is iNEWS-Bergen>
Checking for newsroomsettings version...
Newsroomsettings version OK.
NEWSROOMSETTINGS: c:\channeltemplates\newsroomsettings.xml
13.29.56 9 Manus Directory is c:\manus
13.29.56 14 Loading AvConfig
13.29.56 14 Did not find Server A/B Roll definition for SERVER
13.29.56 14 Did not find Server A/B Roll definition for BACK UP SERVER
13.29.56 14 ChannelTemplates loaded, current studiosetup is Funcspec
```

Manus Administrator is a console application that must be kept running at all times on your Mosart server.

4.1 NRCS Software Compatibility

There are two versions of the Manus Administrator, corresponding to the newsroom system in use:

- **MMConsoleAdmin_2007.exe** For iNEWS (with ftp) workflow.
- **MMConsoleAdmin_MOS.exe** For newsroom systems with *MOS* workflow. Basically all systems except iNEWS with ftp workflow, including iNEWS with MOS workflow.

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](#).

4.2 Configuration Utilities

From the Manus Administrator console you can open the following configuration editors in a new window:

- To open the **frmSettings XML Editor**, type `settings` in the console followed by **ENTER**. For details see:
 - [Settings Editor iNews](#) or
 - [Settings Editor MOS](#) as appropriate.
- To open the **Newsroom Settings XML Editor**, type `ns` in the console window followed by **ENTER**. For details see:
 - [Newsroom Settings Editor](#)

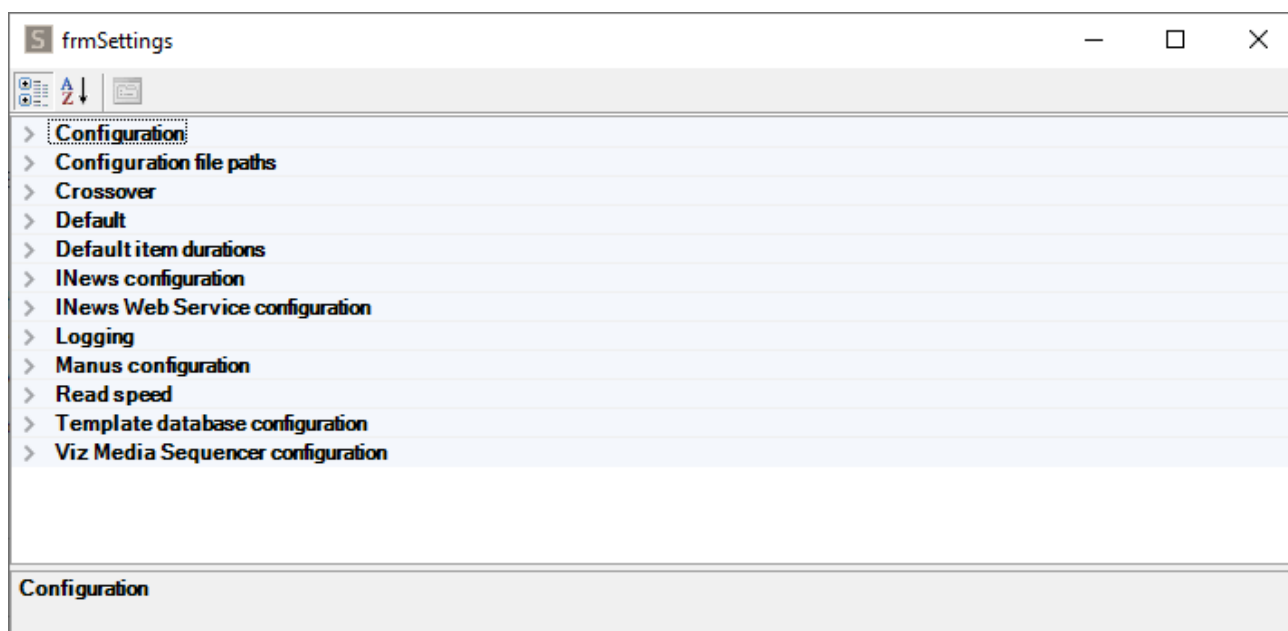
Note: The configuration editor opens in a new window. If you make changes to the configuration, restart Manus Administrator for the changes to be applied.

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](#)

4.3 Settings Editor INews

To open the **frmSettings XML Editor**, open the Manus Administrator console, then type `settings` followed by **ENTER**.



- [Configuration](#)
- [Configuration File Paths](#)
- [Crossover Configuration](#)


- [Default Item Durations](#)
- [iNews Configuration](#)
- [iNews Web Service Configuration](#)
- [Logging](#)
- [Manus Configuration](#)
- [Read Speed](#)
- [Template Database Configuration](#)
- [Viz Media Sequencer Configuration](#)

4.3.1 Configuration

- **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.
- **Auto Take Offset:** Sets offset for the autotake function. Default: 0.
- **Default Handler Name:** Defines a default handler name for all graphics that have no handler name. This can be configured using **DefaultHandlerName** setting in **Manus Administrator**. Default: *DSK*.
- **Default Lower Third Out Behaviour:** Default method for taking out a lower third.
- **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.
- **Ignore updates if nothing changes:** Ignores story update from NCS if the updated story is assumed to be equal to the current version of the story when set. Default: *False* (no NCS story updates are ignored).
- **Keep story status on updates from NCS:** Keeps played out stories gray if the story is updated in the NCS when enabled. Default: *False*
- **MetaData creator priority:** Overrides priority timing information from the video file's source. Select between Newsroom editorial time, MAM duration or full video clip length. Separate IDs with commas. Use the name of the Media Administrator entry. Special names are NCS (for information from the NCS) 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).
- **MyPort:** 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*
- **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 NCS 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 given in frames. Default = 6 frames, i.e. the accessory preload/pretake functions shall take place after the cue operation is done in AvAutomation with an additional and configurable delay of **PreloadAccessoryCueDelay** frames.
- **Reset AutoTake on 'Clear Loop':** Disables the autotake mode automatically when using the clear loop function from the GUI when enabled. Default: *True*
- **Server description:** User-defined descriptive name of the server that is displayed in the **Timing Display**.
- **Should upgrade:** For internal use only. Should not be manually changed. Default: *False*
- **Startup in idle mode:** 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 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 Take Out Logic:** Enables the CG takeout logic. 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. Default: *False*. When True, enables the takeout functionality for overlay graphics.
- **UseItemStatusToNCS:** Sends online offline CUE/PLAY/STOP status to the NCS when enabled. Default: *False*

4.3.2 Configuration File Paths

- **AvConfig:** 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`.
- **Newsroomsettings:** The path to the XML-file containing the mappings from newsroom system commands to Viz Mosart templates. Default: `c:\channeltemplates\newsroomsettings.xml`.

4.3.3 Crossover Configuration

- **ConnectionString:** The connection string of the crossover, for example: **controller=IP address, client=crossover**. Default: ""
- **Crossover Auto Take 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 Set Next On Switch Delay:** Delay of sending set next story from the server running the show when taking control. Default: 500 (milliseconds).

4.3.4 Default Item Durations

- **Break:** The minimum length given to the **BREAK** 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 NCS, 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 NCS, then this time is given to the element. Default: 125 (frames).
- **Full-screen-graphic:** The minimum length given to the **FULLSCREEN GRAPHIC** template type in the rundown. If no time is given in the NCS, then this time is given to the element. Default: 125 (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).
- **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 NCS, then this time is given to the element. Default: 100 (frames).
- **Telephoneinterview:** The minimum length given to the **PHONO** template type in the rundown. If no time is given in the NCS, 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 NCS, then this time is given to the element. Default: 125 (frames).

4.3.5 iNews Configuration

- **Default Rundown:** *Deprecated.*
- **Encoding:** Character encoding of the iNews story items. Should match the encoding used in iNews. Default: Encoding for the operating system's current ANSI code page. Other values: *UTF7, UTF8, UTF32, UNICODE, BIGENDIANUNICODE, ASCII*
- **FTP Port:** iNews FTP port.
- **Ignore Send Cue Status To NCS For Offline Clips:** Enable this flag to prevent the NCS from receiving CUED or READY statuses of offline video clips. Default: *False*
- **Password:** Password of a valid iNews FTP account. Normally password to a valid iNews user with access rights to the iNews FTP server.
- **Refresh Media On NCS Update:** Triggers clip refresh for all clips that belong to a story being updated, normally via story updates from NCS when set to *True*. This ensures that all clip information is synchronized. Default: *False*
- **Server:** Hostname or IP address where the iNews FTP server is running.
- **Time to delay initial devices while waiting for NCS to add story:** Delay of displaying the loading rundown when rundowns are published to Viz Mosart. Default: 500 (milliseconds)
- **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**.
- **Working Directory:** Initial directory in iNews. Viz Mosart gives access to all rundowns stored hierarchically within this directory. Default: *SHOW*

4.3.6 iNews Web Service Configuration

- **Web Service Connection:** Connection string.
Example: **WebServiceServer=localhost; iNewsServer=10.211.112.104; iNewsUsername=mosart; iNewsPassword=mosart; SendUpdatesStatusForAllItems=true; ClearStatusWhenRundownReloaded=true**

4.3.7 Logging

- **Ignore verbose events filter:** Semicolon based list of events to ignore in the log when using verbose logging. Default: *Empty* (log all events)
- **In use:** Enables or disables the logging of events from the Manus Administrator to the logfile. Default: *True*
- **Log level:** Sets the detail level of logging to the log file: 0=normal, 1=warnings, 2=errors, 3=info, 4=detailed
- **MSMQ Log limit:** Value to identify when the application should dump the log queue to file. Default: 4023
- **Pass verbose events filter:** Semicolon based list of events to log when using verbose logging. Default: *Empty* (log all events)


- **Path for MMLog:** The path where the **Viz Mosart Log** is stored. Default: *MosartLog*
- **Trace internally:** Enables or disables internal tracing to console, for debugging only. Default: *False*
- **Use verbose logging:** Enables or disables verbose logging. If **UseLogging** is set, verbose increases the details sent to the log. Default: *False*

4.3.8 Manus Configuration

- **Default Manus:** Selects the default rundown to use. If the setting **UseDefaultManus** is set to *True*, the system loads this rundown on startup.
- **Manus expiration time:** Time in days to keep Manus Administrator files, i.e. files older than the **ManusExpirationTime** are deleted when the Manus Administrator is started. Default: *empty* (no clean up)
- **Manus keep file pattern:** If **ManusExpirationTime** is given this property allows a list of file search patterns for files to keep in the Manus Administrator directory. 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.
- **TestManus:** Rundown to be used for maintenance purposes.
- **Use the default selected manus:** Initializes the rundown given in the **DefaultManus** setting automatically when enabled. Default: *True*

4.3.9 Read Speed

- **Read Speed:** Number of words read per minute by the anchor. (By default, words are taken to be the parts of text separated by space characters. See the setting **Use character for read speed word** below.) This is applied to the 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 which is only used when the NCS System fails to provide a **ReadRate** of its own. This function can be tested by changing any **readspeed="xx"** to **readspeed=""** on an item in a local Manus file.

- **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 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.

4.3.10 Template Database Configuration

- **ConnectionString:** The connection string for the **Template DB**. For example, for MySQL: `server=<hostname>; User Id=<user>; Password=<password>; database=mosarttemplatedb.`
- **Default inserter:** The name to be used for the **insertedby** and **updatedby** columns in the **Mosart Template Database**.

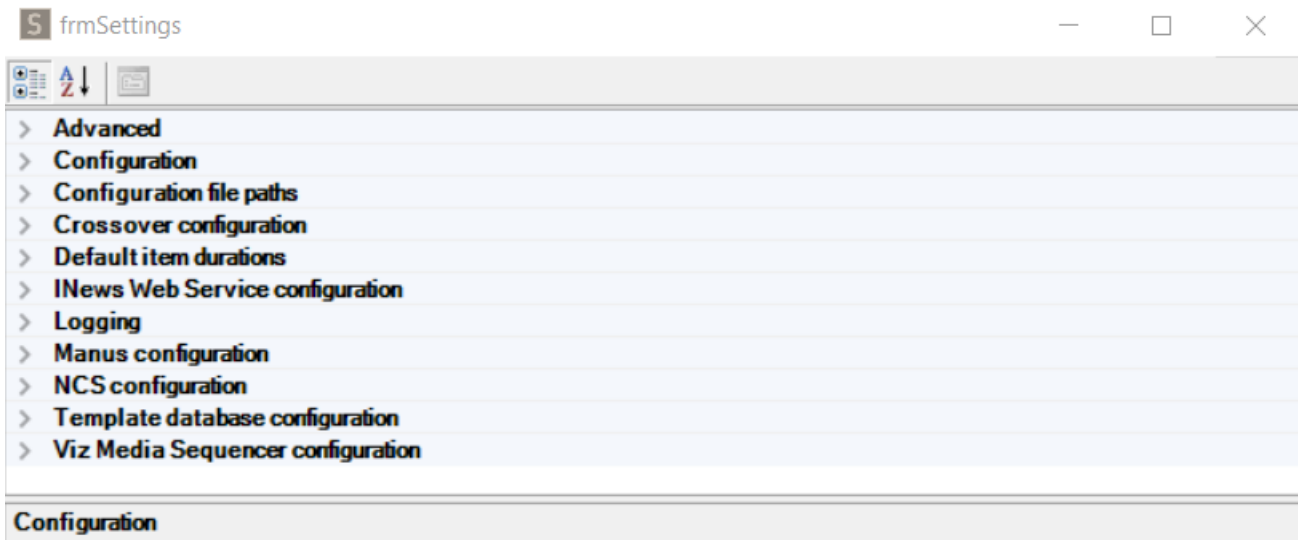
- **Name of provider:** The provider name for the **Template DB**. For example, *MySql.Data.MySqlClient* for MySQL.
- **Use template database:** When true, enables the Template DB functionality, i.e. importing template type aliases from DB to `newsroomsettings.xml` at startup, and exporting `newsroomsettings.xml` to DB after saving `newsroomsettings.xml`. Default: False.

4.3.11 Viz Media Sequencer Configuration

Mse Playlist Panel Host: The host address of the Media Sequencer.

4.4 Settings Editor MOS

- The **frmSettings XML Editor** lets you configure Manus Administrator in a structured way.
- To open the editor, start Manus Administrator and type **settings**, then press **ENTER**.



This section covers the following topics:

- [Advanced](#)
- [Configuration](#)
- [Configuration File Paths](#)
- [Crossover Configuration](#)
- [Default Item Durations](#)
- [iNews Web Service Configuration](#)
- [Logging](#)
- [Manus Configuration](#)
- [NCS Configuration](#)
- [Template Database Configuration](#)
- [Notes](#)

4.4.1 Advanced

- **ItemIdGenerator:** 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, 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 by setting the **ItemIdGenerator** property in Mosart Server settings.

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.

4.4.2 Configuration

- **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.
- **Auto take offset:** Sets offset for the autotake function.
Default: 0 (milliseconds).
- **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 initial synchronization:** Ignores all roReq on startup when synchronizing with NCS.
Default: *False*.
- **Ignore sending item status filter:** Lists regular expressions used to prevent sending MOS status back to NCS. 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).

- **Ignore updates if nothing changes:** If *True* , a story update from NCS is ignored if the updated story is assumed to be equal to the current version of that story.
Default: *False* (no NCS story updates are ignored).
- **Item status unavailable values:** Lists status values used to identify NCS content as invalid. Content as graphics or clip objects.
Default: *NOT_READY, ERROR*.
- **Keep story status on updates from NCS:** When enabled, keeps played out stories gray if the story is updated in the NCS.
Default: *False*.
- **MetaData creator priority:** Select between **Newsroom editorial time** , **MAM duration** or **full video clip length** . Separate IDs with commas. Overrides the timing information priority of the video file source. Use the name of the **Media Administrator** entry. Special names are **NCS** (for information from the NCS) and **GLOBAL** (global values from media object). If empty, Viz Mosart inserts **NCS** as the first entry and **GLOBAL** as the last entry.
Default: *Empty*
- **Min. time between story update:** If no update is received from the NCS within the minimum time stipulated, Viz Mosart forces a story update.
Default: *1000* (milliseconds).
- **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).
- **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 NCS 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 **AvAutomation** with an additional and configurable delay of **PreloadAccessoryCueDelay** frames.
- **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 which is only used when the NCS System fails to provide a ReadRate of its own. This function can be tested by changing any `readspeed="xx"` to `readspeed=""` on an item in a local Manus file.

- **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 the **Manus Administrator** is idle at start-up.
Default: *True*.
- **Story Compare Ignore Attributes:** Used for debugging purposes only.
Default: *Empty*.
- **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)*.
- **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 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**.
- **Use Take Out Logic:** Enables CG takeout logic. When two CG graphic objects with the same graphic ID are next to each other, the first one is not taken out - take in on the second object is executed only. **True** enables the takeout functionality for overlay graphics.
Default: **False**.

4.4.3 Configuration File Paths

- **AvConfig:** 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
- **Newsroomsettings:** The path to the XML file containing the mappings from newsroom system commands to **Viz Mosart** templates.
Default: c:\channeltemplates\newsroomsettings.xml.

4.4.4 Crossover Configuration

- **ConnectionString:** The connection string of the crossover, example: controller=IP address, client=crossover.
Default: **Empty**
- **Crossover Auto Take 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 Set Next On Switch Delay:** Delay of sending set next story from the server running the show when taking control.
Default: 500 (milliseconds).

4.4.5 Default Item Durations

- **Break:** The minimum length given to the BREAK template type in the rundown. If no time is given in the NCS, 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 NCS, 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 NCS, this time is given to the element.
Default: 125 (frames).
- **Full-screen-graphic:** The minimum length given to the FULLSCREEN GRAPHIC template type in the rundown. If no time is given in the NCS, this time is given to the element.
Default: 125 (frames).
- **Item:** The minimum length given to any template type in the rundown. If no time is given in the NCS, this time is given to the element.
Default: 100 (frames).

- **Live:** The minimum length given to the LIVE template type in the rundown. If no time is given in the NCS, this time is given to the element.
Default: 125 (frames).
- **Lower-third:** The minimum length given to the Lowerthird template type in the rundown. If no time is given in the NCS, this time is given to the element.
Default: 125 (frames).
- **Telephoneinterview:** The minimum length given to the PHONO template type in the rundown. If no time is given in the NCS, 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 NCS, this time is given to the element.
Default: 125 (frames).

4.4.6 iNews Web Service Configuration

- **Web Service Connection:** Example:
WebServiceServer=localhost;iNewsServer=10.211.112.104;iNewsUsername=mosart;iNewsPassword=mosart;SendUpdatesStatusForAllItems=true;ClearStatusWhenRundownReloaded=true.
Default: **Empty**.

4.4.7 Logging

- **Ignore verbose events filter:** Semicolon-based list of events to ignore in the log when using verbose logging.
Default: **Empty** (log all events).
- **In use:** Enables or disables the logging of events from the Manus Administrator to the log file.
Default: **True**.
- **Log level:** Sets the detail level of logging to the log file. **0** =normal, **1** =warnings, **2** =errors, **3** =info, **4** =detailed.
Default: **0** .
- **Mos communication is logged:** Enable logging of the communication to and from NCS.
Default: **False** .
- **MSMQ Log limit:** Value to identify when the application should dump the log queue to file.
Default: 4023.
- **Pass verbose events filter:** Semicolon based list of events to log when using verbose logging.
Default: **Empty** (log all events).
- **Path for MMLog:** The path where the Viz Mosart log is stored. Default: **MosartLog** .
- **Trace internally:** Enables or disables internal tracing to console for debugging only.
Default: **False** .
- **Use verbose logging:** Enables or disables verbose logging. If **UseLogging** is set, verbose increases the details sent to the log.
Default: **False**.

- **WTC Level:** Sets the detail level of logging to the console. **0** =normal, **1** =warnings, **2** =errors, **3** =info, **4** =detailed.
Default: **4** .

4.4.8 Manus Configuration

- **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: **0** (no files are deleted)
- **Manus keep file pattern:** A semicolon-separated list of file name patterns. Files with names satisfying any of these patterns will be kept even if they are 'too old' according to the **ManusExpirationTime** setting.
Default: **Empty** (no files to keep). Example: "TEST*;DEMO??.xml" keeps all files starting with "TEST" and all DEMO??.xml files where '?' denotes a wild character.
- **Use the default selected manus:** Initializes the rundown given in the **DefaultManus** setting automatically when enabled.
Default: **True**


4.4.9 NCS Configuration

These settings are for sending Mosart templates as MOS objects to NCS. MOS objects can then be used to create items in a rundown, for example, instead of using Mosart ActiveX. A MOS object containing lower-third information can also be sent.

- **Forced Mos protocol version:** Protocol version to use in NCS communication. Default (blank) setting detects the version from the NCS, otherwise, use "2.8.3", and so on.
Default: **Empty**.
- **Ignore Send Cue Status To NCS For Offline Clips:** Enable this flag to prevent the NCS from receiving **CUED** or **READY** statuses of offline video clips.
Default: **False**.
- **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. NCS to Mosart Lower port:** NCS heartbeat sent to Viz Mosart. If the time between received heartbeats exceeds the value, the connection is displayed as lost. Value= **0** means no timeout.
Default: **0** (seconds).
 - **2. NCS to Mosart Upper port:** NCS heartbeat sent to Viz Mosart. If the time between received heartbeats exceeds the value, the connection is displayed as lost. Value= **0** means no timeout.
Default: **0** (seconds).

- **3. NCS to Mosart High port:** NCS heartbeat sent to Viz Mosart. If the time between received heartbeats exceeds the value, the connection is displayed as lost. Value= 0 means no timeout.
Default: 0 (seconds).
 - **4. Mosart to NCS:** Viz Mosart to NCS timeout. If the time from heartbeat is sent to response is received exceeds the value, the connection is displayed as lost. Value= 0 means no timeout. Default: 0 (seconds).
- **MosId:** MOS identification of this instance of the Manus Administrator. Generic value is mosart.<galleryID>.<stationID>.mos.
Default value: **mosart.mos** .
- **NCS 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.
- **NCS Type:** Generic: No special handling of native NCS commands. **Generic** , **DaletPlus**, **ENPS**, **NcPower**, **Octopus**, **Open Media** and **MOSInews**.
Default: **Generic**.
- **NCSId:** Same form as mosID, but this value is the ID for the newsroom system. See [Notes](#) below.
Default: **NCSSERVER**.
- **NCSId Backup:** Same form as mosID, but this value is the ID for the backup newsroom system. See [Notes](#) below.
Default: **Empty**.
- **Newsroomtag keep keywords (NewsRoomTagKeepKeywords):** A comma (or semicolon) separated list of CAPITALIZED words. 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**. See **Keep Newsroom tag (with the same name) within a story** in Newsroom Tags . If ", " (only a comma) is used all newsroomtags will be kept.
- **ReadyToAir by default:** If the value is true, all MOS active rundowns are assumed to be ready to air.
Default: **False**.
- **Refresh Media On NCS Update:** Trigger clip refresh for all clips that belong to a story that is being updated, normally via story updates from NCS when set to **True** . This ensures that all clip information is synchronized.
Default: **False**
- **Reply with connection MosId:** If set, any MOS message sent to the NCS uses the connection MosId instead of the current mos object ID.
Default: **False**.
- **Schema:** The string is used to identify Viz Mosart items in the rundown. Default: <http://www.mosartmedialab.no/schema/mositem.dtd>.
- **Server:** IP address or hostname of the main newsroom system's MOS gateway. See [Notes](#) below.
Default: **NCSSERVER**.

- **Server BackUp:** IP address or hostname of the backup newsroom system's MOS gateway. See [Notes](#) below.
Default: **Empty**.
- **Template feedback to NCS:**
 - **Enabled:** Enables sending templates to NCS through MOS communication.

 **Note:** **Enabled** is the default value.
 - **GenerateUniqueObjId:** Create a unique identifying name when creating a new MOS Object. The NCS 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 NCS when enabled. Disable to send only the default template set.
 - **TemplateChangeWanted:** Set to **true**, the NCS will receive an update whenever a template is changed or deleted.
- **Use NCS backup server:** Enables the NCS backup configuration. See [Notes](#) below.
Default: **False**.
- **UseItemStatusToNCS:** Sends roltemStat/roElementStat messages to the NCS when clip updates are received from the Media Administrator. If the clip is available on the playout server, **READY** is sent. Otherwise **NOT READY** is sent.
Default: **False**.

4.4.10 Template Database Configuration

- **ConnectionString:** 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 inserter:** The name to be used for the ..._insertedby and ..._updatedby columns.
Default: **inserter**.
- **Provider name:** The provider name for the Template DB. For example, MySql.Data.MySqlClient for MySQL.
Default: **MySql.Data.MySqlClient**.
- **Use template database:** When True enables the Template DB functionality, i.e. exporting template type aliases from DB to newsroomsettings.xml at startup, and importing from newsroomsettings.xml to DB after saving newsroomsettings.xml.
Default: **False**.

 **Note:** Enter the **TemplateDbConnectionString** before enabling **UseTemplateDb**.

4.4.11 Notes

- If there is more than one main newsroom system, the setting parameters **MosServer**, **MosServerBackup**, **ncsID**, and **ncsIDBackup** 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 **MosServerBackup** and **ncsIDBackup**.
- 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 **mosID** should also be a space-separated list of values, in the same order as for **ncsID**.
- Use only a single space between list values. All values are case sensitive.

4.5 Configuring The MSE Playlist Panel

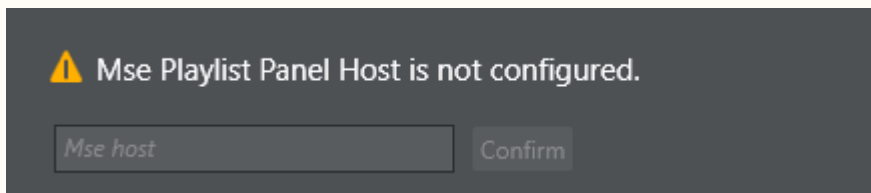
The MSE Playlist panel is presented on the Viz Mosart UI, offering users active (touch or click-sensitive) thumbnails of graphics held on an MSE.

4.5.1 Setting User Access to MSE Playlist Panel

The Viz Mosart UI needs a connection to MSE. You must configure or change an existing MSE host to activate the MSE Playlist panel.

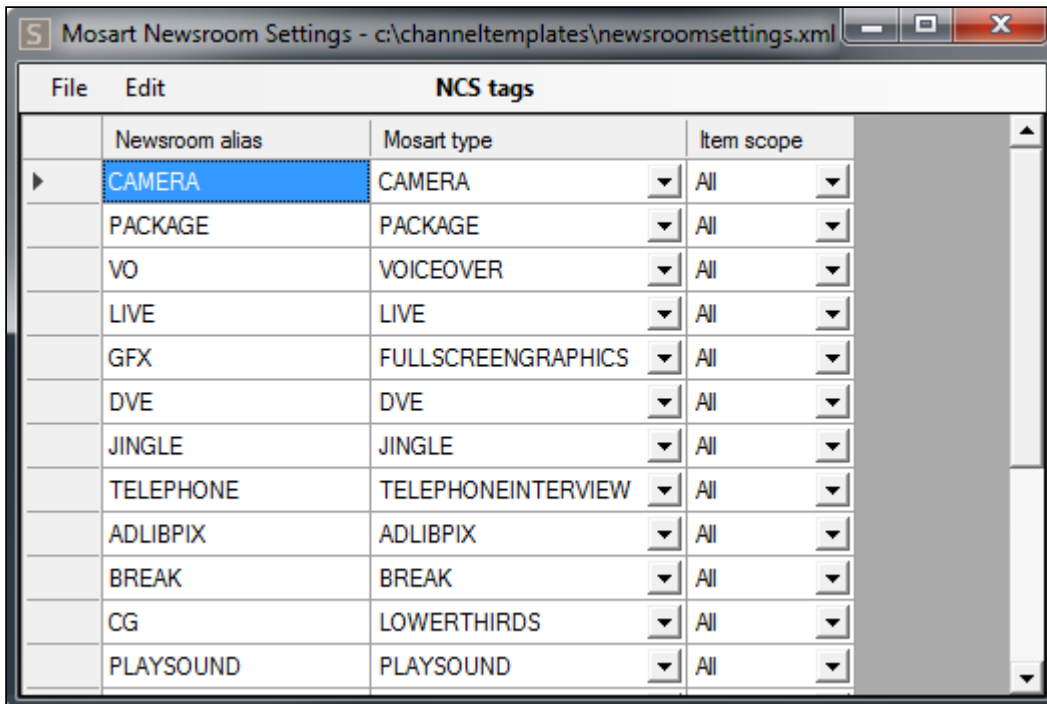
1. Open Viz Mosart ManusAdministrator console window
2. Type **settings**
3. Navigate to the **Viz Media Sequencer** configuration.
4. Modify the value of **Mse host**.
5. Restart the console.

Note: If the **MSE Playlist** panel is visible in the Viz Mosart UI, you can add the host address of the Media Sequencer directly. Both computer name and IP can be used.



- The **MSE Playlist Panel** can be hosted as a floating window or as a docked window in the Viz Mosart UI. This is controlled with **View > Floating windows > Playlist Panel**.

4.6 Newsroom Settings Editor



To open the **Newsroom Settings XML Editor**, open the **Manus Administrator** console, then type **ns** followed by **ENTER**.

This section contains the following topics:

- [Saving Changes](#)
- [Newsroom Settings Editor - Edit Menu](#)
- [Story External Metadata](#)
- [Graphic Destination Letters](#)
 - [Properties](#)
 - [Viz Mosart Object Structure](#)
 - [Viz Mos Item With Payload Data Nodes](#)
 - [Viz Mos Item Without Payload Data Nodes](#)
 - [Viz Mosart VCP MOS item integration](#)
 - [Basic requirements](#)
 - [VCP template requirements](#)
 - [Storing the Mosart field in the VCP data element description](#)
 - [Storing the Mosart field in the mosExternalMetadata section of the VCP data element](#)
 - [VB script Example](#)
 - [Viz Pilot plugin Prerequisites](#)
- [MOS ID Mapping](#)
- [Overriding Default Field Names when Parsing iNEWS Stories](#)

4.6.1 Saving 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.

4.6.2 Newsroom Settings Editor – Edit Menu


The **Edit** menu lists the configuration categories that are available.

- **NCS tags:** Menu for assigning NCS command aliases from the newsroom system to the corresponding Viz Mosart type. Each type can have multiple aliases, i.e. for the Camera type, the aliases can be **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:** Only applies to the iNews system. 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 story is tagged as break.
- **NCS accessory stories:** Viz Mosart perceives the story with the title as a story with accessories only.
- **NCS lowerthird mapping:** List for converting lower third template names from what is written in the NCS (origin) to the desired 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.
- **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:** Only applies to the iNews system. Add 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 Newsroom Control System can send external story values in a container called **mosExternalMetadata (MEM)**, which can be translated to Viz Mosart content. See [Story External Metadata](#) below.
- **Graphic Destination Letters:** **Graphic Destination Letters** is used to identify the output and behavior of graphic elements. See [Graphic Destination Letters](#) below.

 **Note:** Only applies Vizrt VCP MOS items and the use of the Viz Mosart extended data element description.

- **Lower thirds type translation:** Only applies for native ENPS 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:**
- **Omit text:** Omit text from iNEWS presenters area.
- **Properties:** iNEWS custom settings, see [Overriding Default Field Names When Parsing iNEWS Stories](#) below.
- **Ignore parenthesis:** Only applies to MOS. List of start and end parenthesis that encloses text ignored in Viz Mosart.
- **MOS Id mapping:** Enables Viz Mosart to recognize MOS elements from NCS and trigger specific Viz Mosart actions when encountered. Should not be changed by the user. See [MOS ID Mapping](#) 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.

4.6.3 Story External Metadata

The **Newsroom Control System** can send story values in a container called **MosExternalMetadata (MEM)**. These values can be translated to Viz Mosart content. Item-IDs are:

- **mostagname** : XML-tag name in the **MEM**
- **mos_value** : Value in the **mostagname** from the **NCS**
- **Mosart Action:**
 - **template_type:** Use a value from NCS to create a new Mosart item in the rundown story. The value from NCS can be a Newsroom tag of TEXT item scope. Newsroom tags are configured in newsroomsettings.xml, examples: **CAM**, **KAM**, **KAMERA**, **PKG**. Or, you can use the Viz Mosart type number, example: **0** for Camera, **1** for Package. If the value from NCS is empty, no item is created.
Example:

```
<mostag mostagname="var-1" mosart_action="template_type" format_type="String" />
```

A new Viz Mosart item of type given in the NCS column **var-1** is created.

- **item_variable:** Use the value from NCS 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 set to the value taken from columns **var-2** and **var-3** configured in NCS.

- **item_duration:** use the value from NCS 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 NCS tags from Newsroomsettings or as a number, i.e. **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, then the duration applies to those specific items in the story (e.g. **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 NCS to add or change the variant of the first primary item found in the story.
- **template_transition:** Use the value from NCS to add effect transitions to the first primary item found in the story.

Example:

```
<mostag mostagname="TranFromNcs" mos_value="SLIDE" format_type="String"
mosart_action="template_transition" action_value="MIX,24" />
```

Enables you to set the transition type (**MIX**, **WIPE**, **EFFECT**) of the first item in the story from the NRCS rundown. This also provides a good overview in the newsroom of which effects are used between the stories.

Use the value from NCS 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**.

- **story_variable**: Use the value from NCS 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 NCS 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 given in the **action_value** at the start of the story (in the switch from the current to the next story).
- **directtake_post_story**: Executes the direct take 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 NCS to override the **endphrase** attribute for the last primary item in a Mosart story.
- **segment_duration**: Use the value from NCS 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 NCS. 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 NCS in the column with id **ClipDescrCol**" and the field **clip_hirespath** are set to the value **MyClip** if the value from column **ClipIdCol** in NCS 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 info going to the **AsRunLog**.

4.6.4 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		%	
ogw	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 is used to identify output and behavior for graphics elements. Use this to parse a Viz 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:** fullscreen 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. (i.e., the value of the description field found at the specified index). The index is 0-based. Default is **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. Default is **empty** .
E.g. **descriptionfields="0|1"**
- **Use TC Out:** Specifies if the duration should be used as **tc_out**.
- **Inputtxtdelimiter:** Viz 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. The default is **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>

```

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 **destinationmapping** attribute can be set to any string of alpha-numeric 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 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.

The **Graphic Destination Letters** rules can not only be used to re-direct graphics to predefined destinations like **DSK**, **WALL**, **WALL_i**, **FULL** etc., but also to any desired destination. For example, having a rule like:

```
<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**.

Viz Mosart Object Structure

When building a graphic Viz Mosart object from a Viz 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" endphrase="" 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 the `MosAbstract` found in the Viz Mos object and the fields delimited by the `inputtxtdelimiter` are replaced with `outputtxtdelimiter`.

Viz Mosart can receive several types of Viz Mos graphic objects from NCS, including a [Viz Mos Item With Payload Data Nodes](#) and a [Viz Mos Item Without Payload Data Nodes](#).

Viz Mos Item With Payload Data Nodes

If a Viz 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 Mosart item (created from the Viz 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 Viz 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.

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 Viz 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>

```

Viz Mos Item Without Payload Data Nodes

If the Viz 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 Viz 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 Viz 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="Lowertthird" templatetypefield="" descriptionfields="1|3" inputtxtdelimiter="|" outputtxtdelimiter="%" /> </graphicdestinationletters>
```

The result is as follows:

```
<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="100" rdur="0" externaleffect="" intimeline="true" date_0="" accessory="False" static="false" endphrase="" mosid="VIZ.NPRO.MOS" objid="1529718" ismoselement="true" use_graphics_id="true" graphics_id="1529718" handler_name="DSK" graphics_out_on="TIMECODE" 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>
</item>
```


Viz Mosart VCP MOS item integration

Vizrt MOS items do not contain any information on play-out 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 contains Vizrt MOS item metadata that is required to allow Mosart to separate play-out channel and extract timing information.

Basic requirements

Mosart needs to know whether the graphics element is a lower third, a graphics played out on a video wall engine, or a full screen graphics. Any graphics needs 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

The optional Mosart item is 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 [Graphic Destination Letters](#).

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 fullscreen graphic)

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 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
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

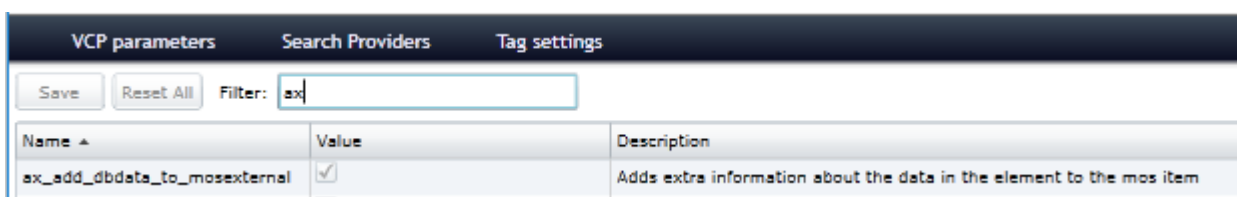
```

Viz Pilot plugin Prerequisites

To enable the **mosExternalMetadata** section in the Viz Pilot News (Active X) plugin, the system administrator needs to 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:

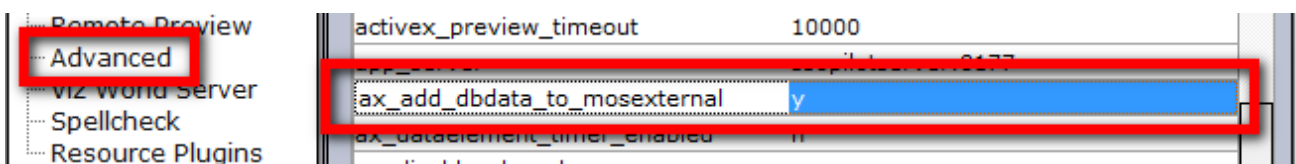
[http://\[hostname of PDS\]:8177/settings#Params](http://[hostname of PDS]:8177/settings#Params)



Name	Value	Description
ax_add_dbdata_to_mosexternal	<input checked="" type="checkbox"/>	Adds extra information about the data in the element to the mos item

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**.



activex_preview_timeout	10000
ax_add_dbdata_to_mosexternal	y
ax_dataelement_timer_enabled	n

4.6.5 MOS ID Mapping

A **MOS Id** mapping can be done in the `newsroomsettings.xml` file, as follows:

```
<mosids>
  <tag name="" value="" fieldmapping="" appendContent="" embeddedMosartItem=""
  keywords="" />
</mosids>
```

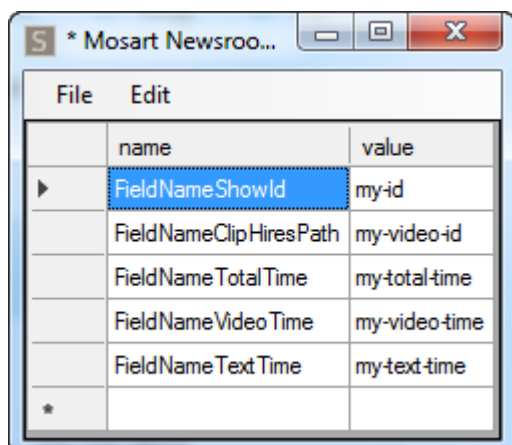
Where:

- **name:** The MOS ID that identifies the MOS elements coming from NCS.
- **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 NCS is ignored. For example:


```
<tag name="MOSART" value="" appendContent="false" fieldMapping="false"
          keywords="IgnoreTemplateSet=true"/>
```

4.6.6 Overriding Default Field Names when Parsing iNEWS 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**.



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.

4.7 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](#)

4.7.1 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.

4.7.2 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.

4.7.3 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

- **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).

4.7.4 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**.
- **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**.

4.7.5 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>

```

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"/>
  <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>

```

- 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** .

4.8 Manus Administrator Commands

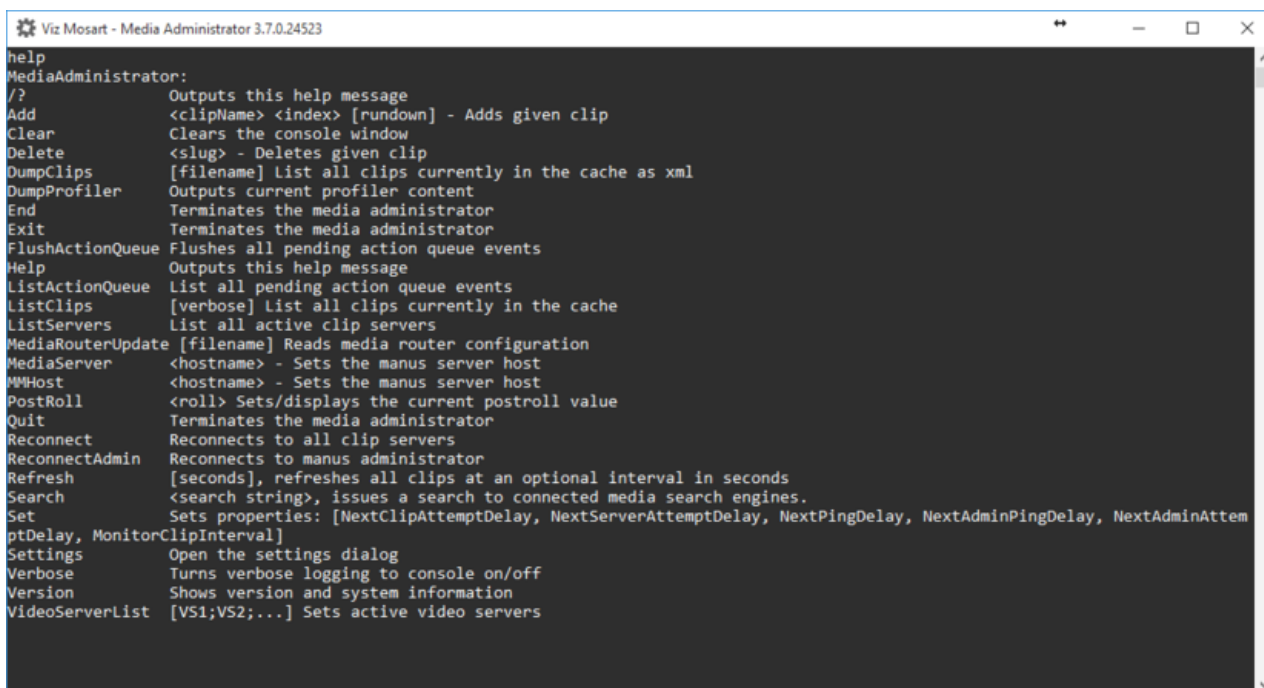
These console commands are only intended for advanced users, and principally for troubleshooting. They shall never be used on a live system.

Warning: Use of these commands may adversely affect system performance. Unauthorized use of these commands may invalidate a contracted Vizrt Support agreement.

- **DUMPNCSTRUNDOWN DNR /p[destination filepath]:** Dumps the current rundown in the native NSML format
- **DUMPNCSTORY, DNS [storyname] /p[destination filepath]:** Dumps a story in native NSML format
- **-FILEWATCHER -FW:** Starts the application in file watcher mode
- **Go to next story**
- **HIDELOG:** Hide logging elements
- **INWSDEFAULTRUNDOWN:** Sets the defaultShowpath
- **INWSFTPSERVER [servername] [username] [pathword]** Sets the ftp connection to inews FTP server. You must restart GTS or G
- **INWSWORKINGDIRECTORY:** Sets the working directory
- **INIFILEPATH** Sets the ini file
- **LISTCLIPS:** List all items in media cache
- **LM, LOADMANUS [filename]:** Loads manus from file
- **LOG:** Collect logs for all connections
- **LOG [filename]:** Write current log to file
- **NEWSROOMSETTINGS,NS:** Show settings dialog for newsroomsettings

- **NEWT**: Send Mosart templates to NCS as MOS objects with status NEW (Templates which are not marked "Send to NCS" in the Template properties, will not be sent to NCS)
- **NONEAUTOS**: List non-auto lowerthirds
- **PARSENATIVESTORY,PNS**: Parses a native NCS file to Mosart XMLPS [0] [0]
- **pcol**: Print col1
- **PRINTRUNDOWNS**: Print newsroom rundowns
- **PST**: Print states
- **PRINTSTORY**: Print story or story item
- **R**: Release Background
- **READSPEED [0]**: Set read speed in words per minute
- **RELOADMANUS, RL, RELOAD**: Reload current manus
- **RELOADTEMPLATES, RT**: Reload ChannelTemplates
- **RESYNC-resync**: Resync rundown with NCS
- **S**: Skip next story
- **SAVE [path]**: Save the rundown as a file.
- **SAVESEQUENCE [sequencename as string]**: Save the rundown as a sequence
- **setManusDirectory**: Set the directory in which manus are stored
- **setrundown, SR, CD [name]**: Set a new rundown
- **SETTINGS**: Show settings dialog
- **SHOWLOG**: Show logging elements
- **SS, STUDIOSETUP**: Set the current studiosetup
- **START**: Start or stop the timeline
- **STEST**: Start a series of automatic tests. **Y** to start, **N** to stop. (millisec).
- **TESTUPDATE**: Start test timer
- **tmpPrint**: Print the manus
- **TRACE**: Activate/deactivate the trace listener
- **UPDT**: Send Mosart templates to NCS as MOS objects with status UPDATED (Templates which are not marked "Send to NCS" in the Template properties, will not be sent to NCS)
- **VERSION**: Print the current version
- **WQ**: Write queue list

5 Media Administrator



```

Viz Mosart - Media Administrator 3.7.0.24523
help
MediaAdministrator:
/?          Outputs this help message
Add        <clipName> <index> [rundown] - Adds given clip
Clear      Clears the console window
Delete     <slug> - Deletes given clip
DumpClips  [filename] List all clips currently in the cache as xml
DumpProfiler Outputs current profiler content
End        Terminates the media administrator
Exit       Terminates the media administrator
FlushActionQueue Flushes all pending action queue events
Help       Outputs this help message
ListActionQueue List all pending action queue events
ListClips  [verbose] list all clips currently in the cache
ListServers List all active clip servers
MediaRouterUpdate [filename] Reads media router configuration
MediaServer <hostname> - Sets the manus server host
MWHost     <hostname> - Sets the manus server host
PostRoll   <roll> Sets/displays the current postroll value
Quit       Terminates the media administrator
Reconnect  Reconnects to all clip servers
ReconnectAdmin Reconnects to manus administrator
Refresh    [seconds], refreshes all clips at an optional interval in seconds
Search     <search string>, issues a search to connected media search engines.
Set        Sets properties: [NextClipAttemptDelay, NextServerAttemptDelay, NextPingDelay, NextAdminPingDelay, NextAdminAttemptDelay, MonitorClipInterval]
Settings   Open the settings dialog
Verbose    Turns verbose logging to console on/off
Version    Shows version and system information
VideoServerList [VS1;VS2;...] Sets active video servers

```

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 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 **MediaAdmin** is started.
- when the **ManusAdmin** is activated (i.e. from **idle** to **active** transition).
- when the **AvAutomation** changes the current setup.

To open the **Settings XML Editor**, type **settings** in **Manus 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](#)

5.1 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.
- **dumpprofiler**: 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 to 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.

5.2 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.

Mosart Media Administrator Properties	
Configuration	
Enable dynamic configuration	False
Post roll frames	75
ShouldUpgrade	False
Connection	
Existing clip check interval	00:00:30
Max clip duration for key frames	-1
Monitoring clip interval	00:00:05
Next admin attempt delay	00:00:05
Next admin ping delay	00:00:05
Ping delay	00:01:00
Server reconnect delay	00:00:30
Connection Media search Servers	
Video clip Server 1	Name=OM;Type=OmneonSearch;Server=10.211.114.103;LoadDatabase=true;Ext List=.mxf.MOV;Config=ClipServerOmneon.xml
Video clip Server 1 Backup	
Video clip Server 2	Name=K2;Type=GrassValleyK2;Server=10.208.13.21;Domain=MosartK2;User=Administrator;Pass=adminGV1;Suite=Mosartx;Volume=V.;Bin=Nyhe
Video clip Server 2 Backup	
Video clip Server 3	Name=ASMS;Type=AirSpeedMultiStreamSearch;Config=ClipServerMultiStream.xml;Server=localhost;LoadDatabase=true
Video clip Server 3 Backup	
Video clip Server 4	
Video clip Server 4 Backup	
Video clip Server 5	
Video clip Server 5 Backup	
Video clip Server 6	
Video clip Server 6 Backup	
Logging	
Ignore verbose events filter	
Log exist	True
Pass verbose events filter	
Use logging	True
Use verbose logging	True
Manus admin connection	
Host	localhost
Port	8085
Media router	
Configuration filename	MediaRouterConfigVideo.xml
Server index name	
Use	False
Manus admin connection	

5.2.1 Configuration

- **Enable dynamic configuration:** If *False* (default), *Media Administrator* connects to all servers configured. (See [Connection Media Search Servers](#), [Video clip Server <n>](#) below.) However, if this setting is *True*, it connects only to
 - those servers that are part of the salvo selected in *AV Automation* (See [AV Automation Devices Video Servers](#), Virtual Server Ports, Switch between salvos), and
 - those servers which have *Static=True* in the connection string. (See [Connection Media Search Servers](#), [Video clip Server <n>](#) below.)
- **Post roll frames:** Subtracts value from the actual clip length when sending clip info back to the **Manus Administrator**. Default: 75 (frames).
- **ShouldUpgrade:** For internal use only, should be set to **False**. Default: **False**.

5.2.2 Connection

- **Existing clip check interval:** 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:** Disables showing keyframe markers for video files greater than this value in frames. Default: **-1** .
- **Monitoring clip interval:** 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:** 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:** 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:** 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**.

5.2.3 Connection Media Search Servers

- **Video clip Server <n>:** Connection string to media server (video server or MAM system) <n>. A maximum of six media servers may be connected. In most cases only one server connection is necessary.

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 [AV Automation Devices 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

- those servers (identified by the *Name* property) that are part of the salvo selected in *AV Automation* (see [AV Automation Devices Video Servers](#), Virtual Server Ports, Switch between salvos), and
- those servers which have *Static=True* in the connection string.

⚠ Note: When MediaAdmin 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.

5.2.4 Logging

- **Ignore verbose events filter:** Semicolon based list of events to ignore when using verbose logging. Default: **<none>** .
- **Log exist:** Verifies the presence of the Viz Mosart Log Service at startup. Default: **True** .
- **Pass verbose events filter:** Semicolon based list of events to log when using verbose logging. Default: **<none>** .
- **Use logging:** Passes all logging to the Log Service if true. Default: **True**.
- **Use verbose logging:** Enables verbose logging if true. Default: **False**.

5.2.5 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**.

5.2.6 Media Router

- **Configuration filename:** 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 Overlay Graphics Interface

The Overlay Graphics Interface provides control and monitoring of overlay graphics for all Viz Mosart-approved graphic engines. 2

- For a list of supported graphics devices, see [Overlay Graphics Types](#).
- Configuration is done in [Overlay Graphics Configuration](#).

Start **Overlay Graphics** by double-clicking the application icon:

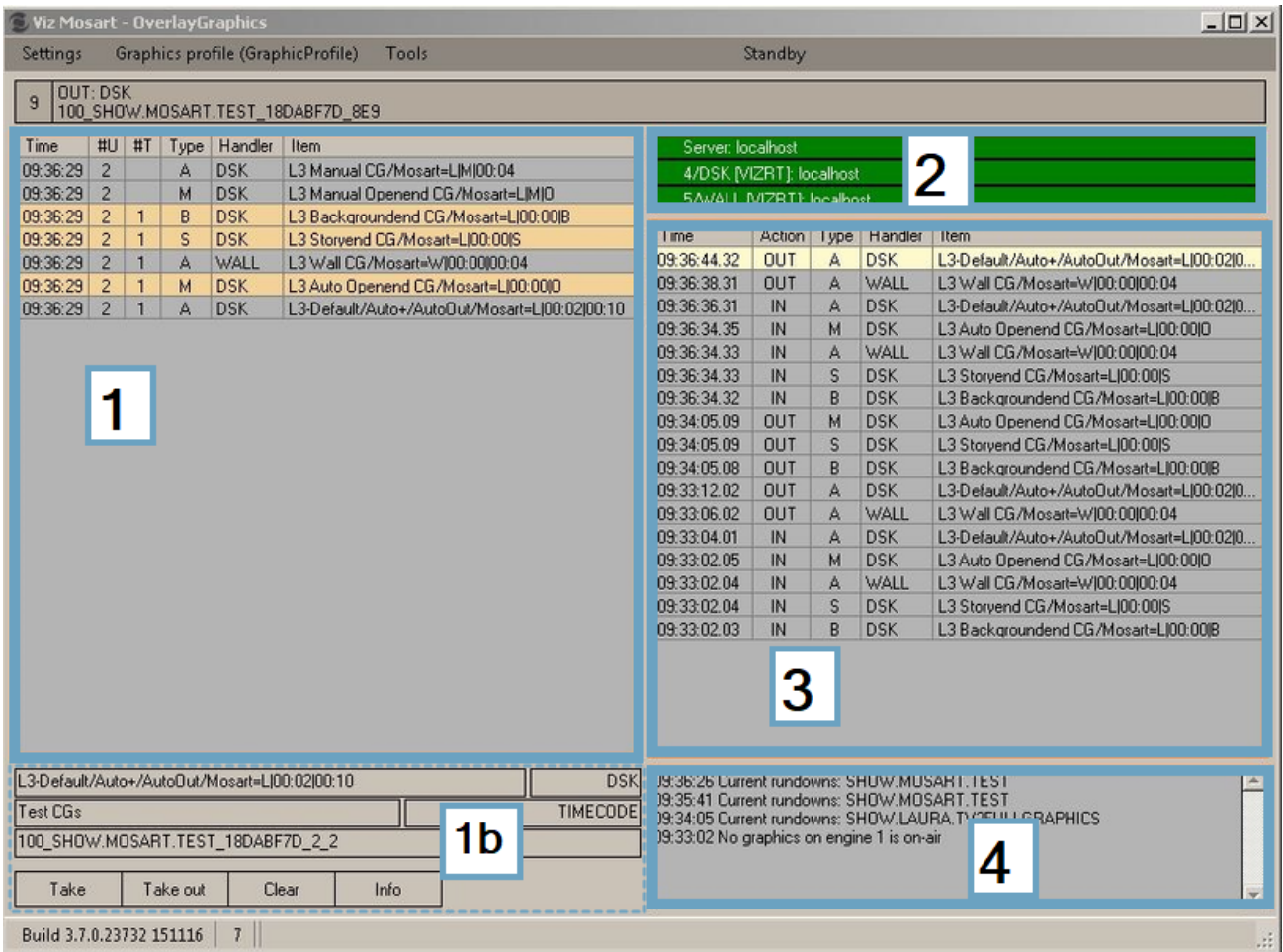


or by navigating to the installation directory (by default C:\Program Files (x86)\Mosart Medialab\Mosart Server) executable MMServer\MMOverlayGraphicsInterface.exe.

⚠ 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**, **Graphic Profile**, **Tools** and **Standby**.



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).

6.1 [1] List Of Overlay Graphics In The Rundown

The screenshot displays the 'Standby' window of the Viz Mosart Administrator. It features a main table of overlay graphics and a secondary log table. Annotations include:

- Named overlay:** Points to the row with Time 14:59:04, #U 9, #T 1, Type DSK, Handler Test Name, Item Test Name.
- Rundown/button overlay = same overlay is in rundown and on a keyboard shortcut:** Points to the row with Time 15:26:48, #U 1, #T 1, Type M, Handler WALL, Item Lowerthird 2/Gfx2.
- Rundown overlay:** Points to the row with Time 15:26:48, #U 1, #T 1, Type M, Handler WALL, Item Lowerthird 2/Gfx2.
- The row is selected:** Points to the row with Time 15:26:48, #U 1, #T 1, Type M, Handler WALL, Item Lowerthird 2/Gfx2.
- Button overlay which is not in the rundown (only on a keyboard shortcut):** Points to the row with Time 15:26:48, #U 1, #T 1, Type M, Handler WALL, Item Lowerthird 2/Gfx2.

The main column labels are: **Time**, **#U**, **#T**, **Type**, **Handler**, and **Item**.

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)

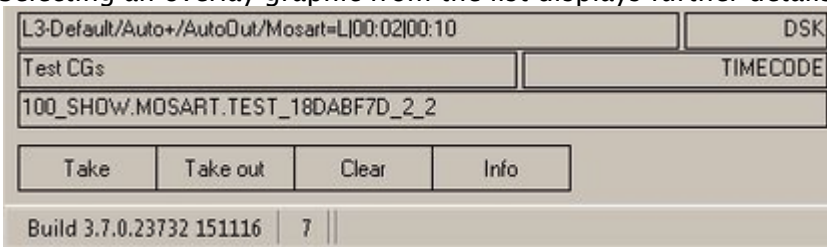
Type	Description
B	Background End
S	Story End
M	Manual (Open End)
L	Last Out
-	Stay Always
>	if template is STORYSTART
***	None of the above

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
	Graphic type (see [1] above)

Element	Description
<div style="border: 1px solid gray; padding: 2px; margin-bottom: 5px;">100_SHOW.MOSART.TEST_18DABF7D_2_2</div>	<p>Graphic ID</p> <div style="border: 1px solid yellow; padding: 5px; margin-top: 10px;"> <p>⚠ This is the <i>Viz Mosart</i> Graphic ID, not the graphic ID from graphic database (Graphic Hub/Oracle)</p> </div>
<div style="border: 1px solid gray; padding: 2px; margin-bottom: 5px;"> Take Take out Clear Info </div>	<p>Action Buttons</p> <p>Take: Takes graphic on air</p> <p>Take out: Takes graphic off air</p> <p>Clear: Takes out all taken graphics</p> <p>Info: Opens an editor window where the graphic MOS item can be changed and updated</p>

6.2 [2] Server Status

Viz Mosart server and configured graphic engines statuses are displayed (Green: OK, Red: Failure) in the panel at the top right:

Time	Action	Type	Handler	Item
13:27:57.56	IN	S	DSK	In:00:00/HEADLINE OPENER/Mail bombs ... / 19.08.2019 16:10:23
13:27:57.75	OUT	S	DSK	In:00:00/HEADLINE OPENER/Mail bombs ... / 19.08.2019 16:10:23

6.3 [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.

6.4 [4] Event Log

The bottom right contains an event log with messages, warnings and errors from the Overlay Graphics application.

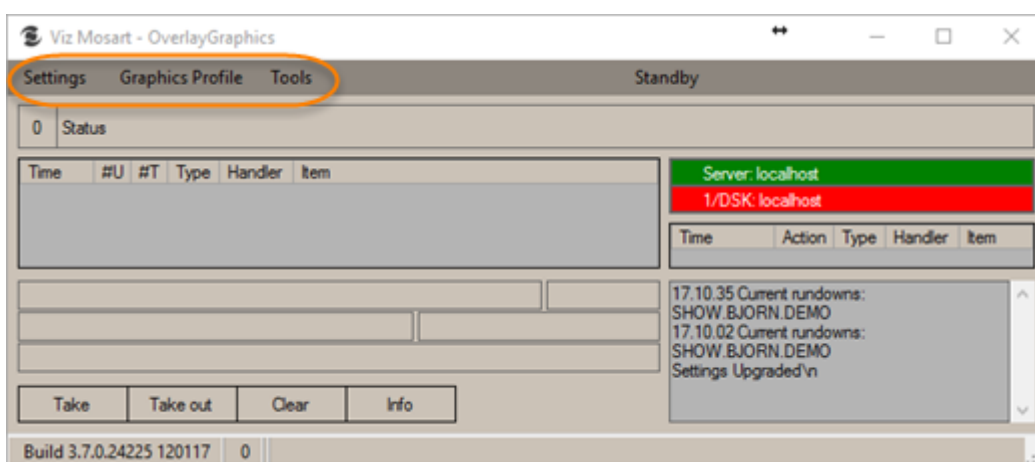
To clear this area, double-click inside it.

6.5 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:



6.6 Menu Bar



Settings

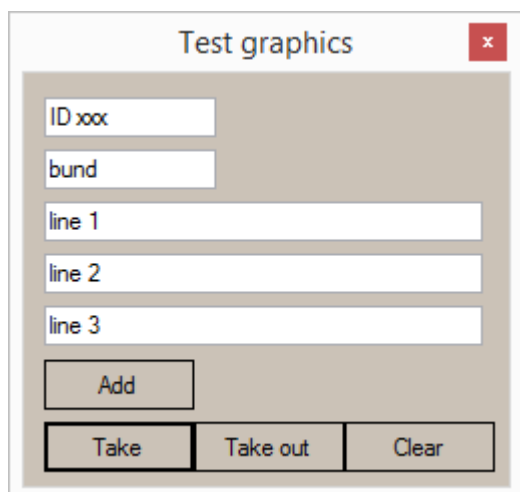
Properties: Opens [Overlay Graphics Configuration](#).

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.

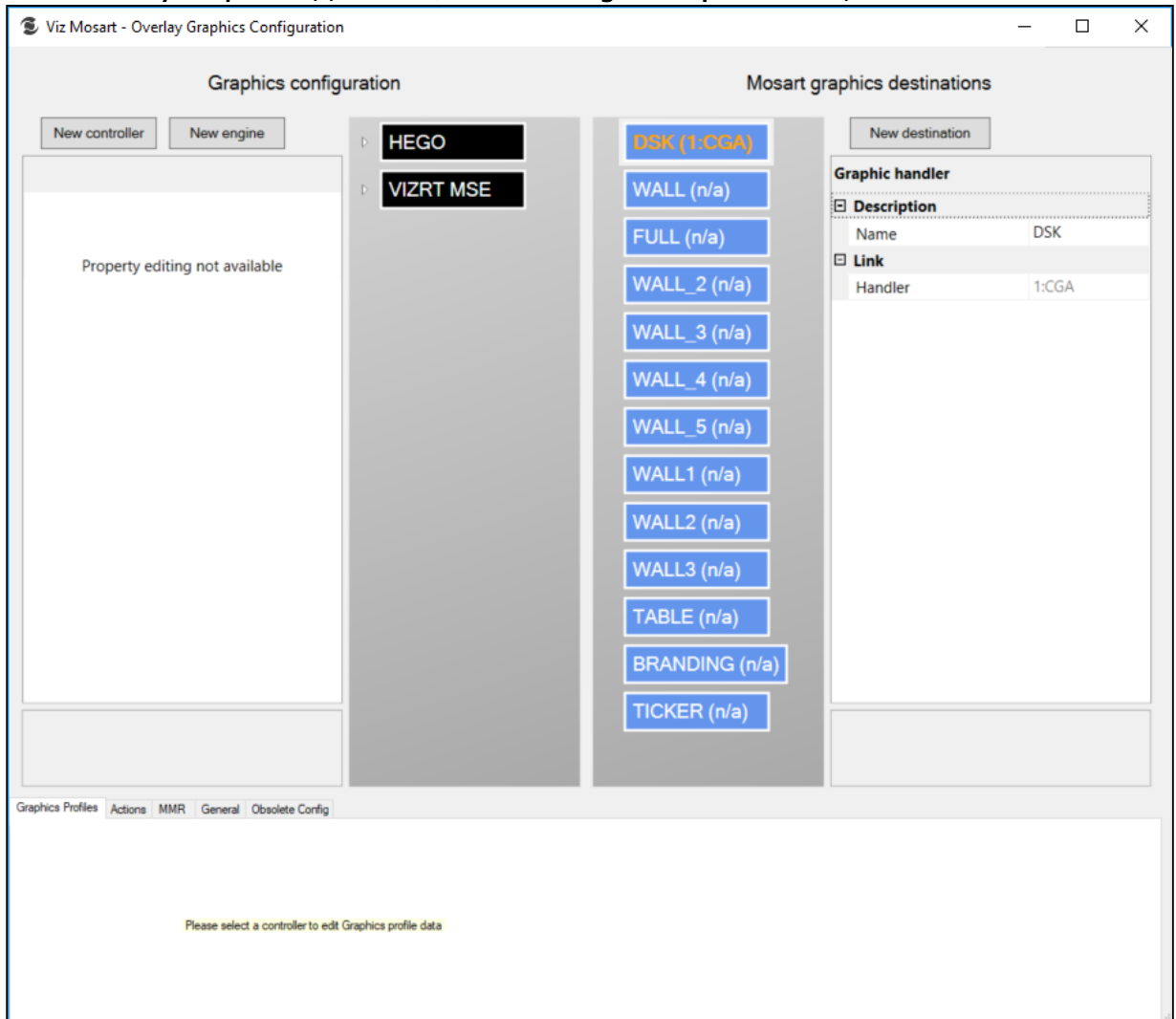
6.7 Overlay Graphics Configuration

6.7.1 Overlay Graphics Configuration Window

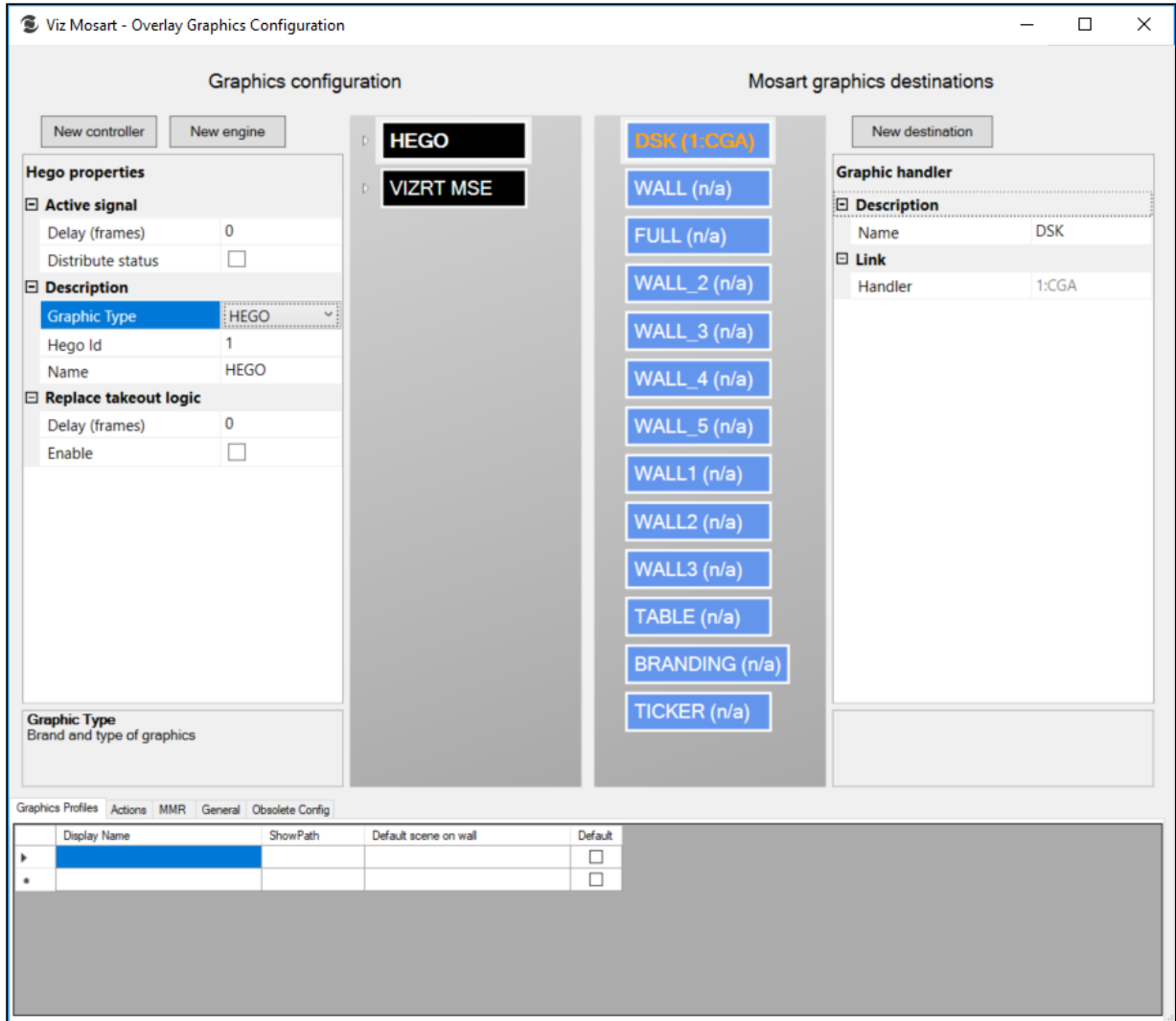
The **Overlay Graphics Configuration** (OGC) window provides options for the **Overlay Graphics** interface.

- [Overlay Graphics Configuration Window](#)
- [Controllers, Engines and Destinations](#)
 - [Controller](#)
 - [Engine](#)
 - [Destination](#)
- [Configuration Panels](#)
 - [Panel 1 - Graphics Configuration](#)
 - [Panel 2 - Mosart Graphics Destinations](#)
 - [Panel 3 - Property Tabs](#)

- In the **Overlay Graphics** application, select **Settings > Properties** to open:



- Select (double-click) an object (here: the controller named **HEGO**). The form displays more information.



6.7.2 Controllers, Engines and Destinations

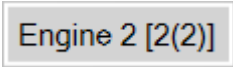
Controller



A controller defines the type of the graphics system that is to be used. For each controller, both common and specific properties can be configured as described below.

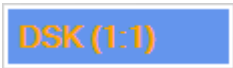
As a rule, connection settings for the graphics system are not defined at Controller level. However, for some graphic handlers, connection definitions to auxiliary components must be provided (for example, when connecting to the Media Sequencer for Vizrt graphics systems).

Engine



For each controller, one or more engines can be added. An engine defines the connection to the graphics system.

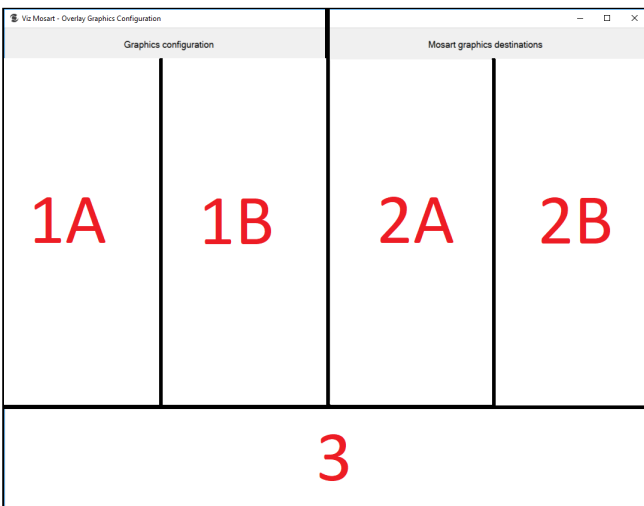
Destination



The destination defines the output channels where you play out the graphics.

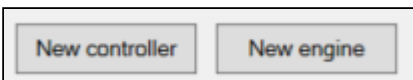
6.7.3 Configuration Panels

For explanation, each panel is presented in this style:



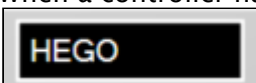
Panel 1 – Graphics Configuration

Panel 1A



- **New controller:** Select to add.
- **New engine:** Select to add.

When a controller has been clicked:



Hego properties	
Active signal	
Delay (frames)	0
Distribute status	<input type="checkbox"/>
Description	
Graphic Type	HEGO
Hego Id	1
Name	HEGO
Replace takeout logic	
Delay (frames)	0
Enable	<input type="checkbox"/>

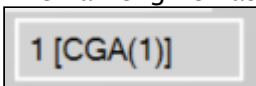
Hego example

- **Active signal**
 - **Delay (frames):** Displays an integer value indicating number of frames to be delayed. Default: 0 .
 - **Distribute status:** Displays a checkbox indicating whether the selected controller is set to **distribute status**. Default: **Unchecked**.
- **Description**
 - **Graphic Type:** Displays the selected controller type from a dropdown list of possible controllers.
 - **Hego Id:** Displays an integer value indicating the selected controller's ID number.
 - **Name:** Displays the written name of the selected controller.

Replace takeout logic

- **Delay (frames):** Displays an integer indicating a number of frames to be delayed. Default: 0 .
- **Enable:** Displays a checkbox indicating whether the **Replace takeout logic** has been enabled. Default: **Unchecked**.

When an engine has been clicked:



Hego engine	
Behaviour	
Always Take In	<input type="checkbox"/>
Continue Scene	<input type="checkbox"/>
Configuration	
Channel	0
Encoding	
Host	localhost
Port	10003
Description	
Name	1
Handler	
Handler Id	1
Name	CGA

Hego engine example

Behavior

- **Always Take In:** Whether the selected engine is set to always taking in .
Default: **Unchecked.**
- **Continue Scene:** Whether the selected engine is set to continue the scene .
Default: **Unchecked.**

Configuration

- **Channel:** Shows the integer number for the channel. Default: **0** .
- **Encoding:**
- **Host:** Displays the host identification name. Default: **localhost.**
- **Port:** Displays an integer value for the selected engine's port number. Default: **10003** .

Description

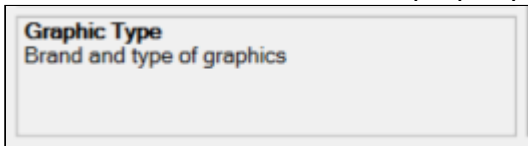
- **Name:** Displays an integer value for the selected engine's ID number. Default: **1** .

Handler

- **Handler Id:** Displays an integer value for the selected engine's ID number. Default: **1** .
- **Name:** Displays the name of the engine handler.

Bottom of Panel (information)

- Further details about the active property

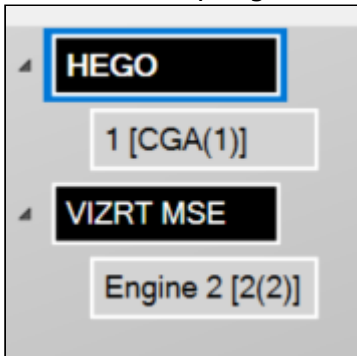


Panel 1B

- This section contains a list of controllers:



- A sub-list of any engines for each controller:



Panel 2 – Mosart Graphics Destinations

Panel 2A

- A list of destinations:



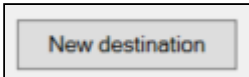
- The activated destination is **orange**.



- Multiple engines can be assigned to:
 - The same destination (**Mirrored Graphics Playout**), where you use identical graphics concepts.
 - Different destinations with different names, where you use separate concepts for each output.

- Different destinations with the same name, for example, playing out the same graphics on multiple parallel outputs, either in a virtual set configuration or in a multi-branding setup.

Panel 2B

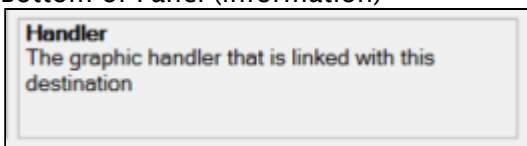


- **New destination:** Select to add.



Graphic handler

- **Description:** Displays the selected, named destination.
- **Link:** Displays a concatenation of the **Handler Id** and the **Handler Name** separated by a semi-colon. This information indicates the graphics handler linked to this specific destination.
- Bottom of Panel (information)



Further details about the active property

Panel 3 – Property Tabs

Graphics Profiles

Only available when the selected item is a controller.

Graphics Profiles					
	Display Name	ShowPath	VCP concept	Default scene on wall	Default
▶	Green1	Mosart/Overlays/Green1	Green		<input checked="" type="checkbox"/>
	Yellow1	Mosart/Overlays/Yellow1	Yellow		<input type="checkbox"/>
					<input type="checkbox"/>

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

- **Display Name**
- **ShowPath:**
- **Default scene on wall:**

- **Default:** Indicates whether the entry may be set as default.
Default: **Unchecked**

Actions

Only available when the selected item is an engine.

- **Event:**
- **Action:**
- **Value:**

MMR

The screenshot shows the 'MMR' configuration window with the 'General' tab selected. On the left, there is a checkbox for 'Use Media Router' which is unchecked. To the right is the 'Engine Destination Map' section, which contains eight rows for Engine 1 through Engine 8. Each row has a checkbox and a dropdown menu. Engine 1 has its checkbox checked and the dropdown set to 'DSK'. Engine 2 has its dropdown set to 'WALL'. Engines 3 through 8 have their checkboxes unchecked and their dropdown menus empty.

- **Use Media Router:** Default: **Unchecked**.
See the [Viz Mosart Media Router Guide](#).
- **Engine Destination Map**
 - **Engine 1 [to 8]:** Displays a drop-down list of optional destinations to be linked to this engine. Check the box to set the destination. Default: **Unchecked**.

General

The screenshot shows the 'General' configuration window. It is divided into several sections:

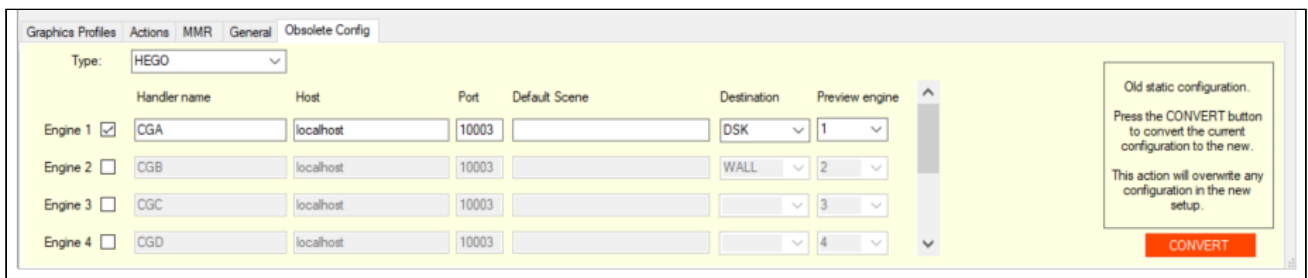
- Logging:** Contains two checkboxes: 'Verbose' (unchecked) and 'Trace' (checked).
- Mosart server:** Contains a text input field for 'Hostname' with the value 'localhost'.
- GUI Control:** Contains a checkbox for 'Show local id's' which is unchecked.
- Graphics Mirroring:** Contains a checkbox for 'Enable' which is unchecked.
- Connectivity:** Contains two checkboxes: 'Connected when in standby' (checked) and 'Connected when idle' (checked).

- **Logging**
 - **Verbose:** If checked, verbose actions are logged. Default: **Unchecked**.
 - **Trace:** If checked, trace actions are logged. Default: **Checked**.
- **GUI Control**
 - **Show local id's:** If checked, local IDs are displayed. Default: **Unchecked**.
- **Mosart server**

- **Hostname:** Shows the **Mosart Server** host name. Default: **localhost**.
- **Graphics Mirroring**
 - **Enable:** If checked, graphics mirroring is enabled. Default: **Unchecked**.
- **Connectivity**
 - **Connected when in standby:** If checked, Mosart Server remains connected in standby mode. Default: **Unchecked**.
 - **Connected when idle:** If checked, Graphics devices will remain connected in idle (when switching from main to backup, for example). Default: **Unchecked**.

Obsolete Config

⚠ Overwrite Alert: Clicking the red **CONVERT** button in the **Obsolete Config** tab overwrites your existing configuration. Make sure you understand the consequences before clicking **CONVERT**.



Engine	Handler name	Host	Port	Default Scene	Destination	Preview engine
Engine 1 <input checked="" type="checkbox"/>	CGA	localhost	10003		DSK	1
Engine 2 <input type="checkbox"/>	CGB	localhost	10003		WALL	2
Engine 3 <input type="checkbox"/>	CGC	localhost	10003			3
Engine 4 <input type="checkbox"/>	CGD	localhost	10003			4

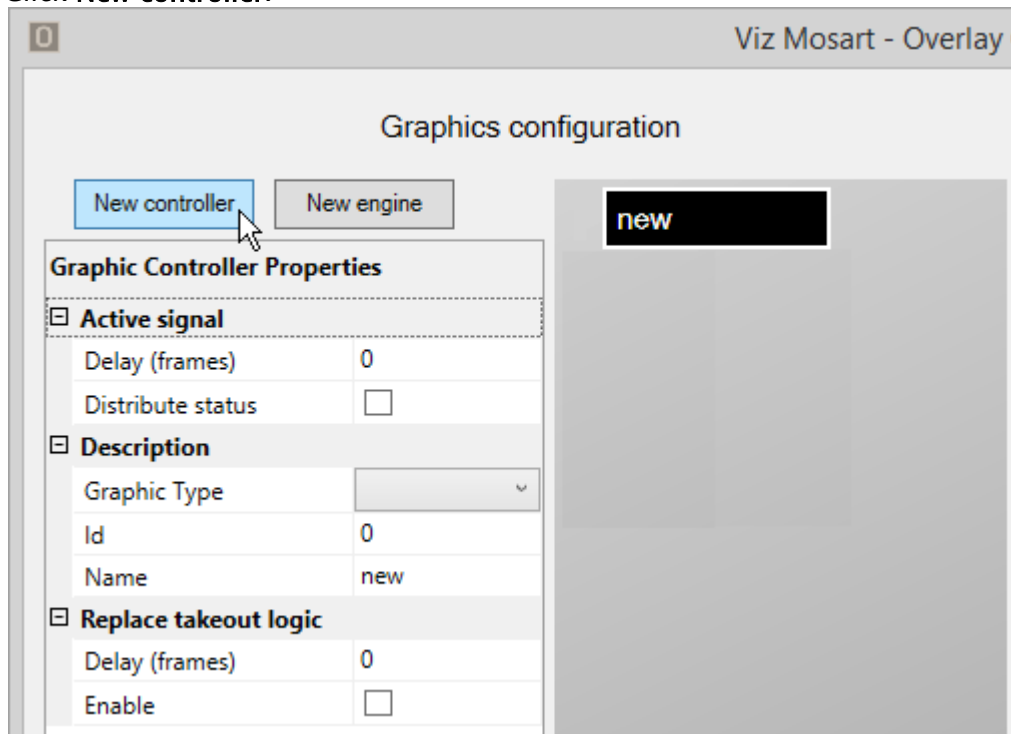
- **Type:** Displays available controller types.
- **Engine**
 - **Engine activation:** Activates the engine by checking the checkbox.
 - **Handler name:** Sets the name of the engine handler.
 - **Host:** Sets the name of the host.
 - **Port:** Sets the port number.
 - **Default Scene:** Sets the default scene.
 - **Destination:** Sets the destination by selecting one from the dropdown list.
 - **Preview engine:** Sets the order of the preview engines.
- **Old static configuration:** This part of the form is considered obsolete.

i Obsolete Configuration Notification: The red **CONVERT** button in the **Obsolete Config** tab is only intended for use by those who have been working with the old configuration and want to convert the old configuration and replace it with the current configuration.

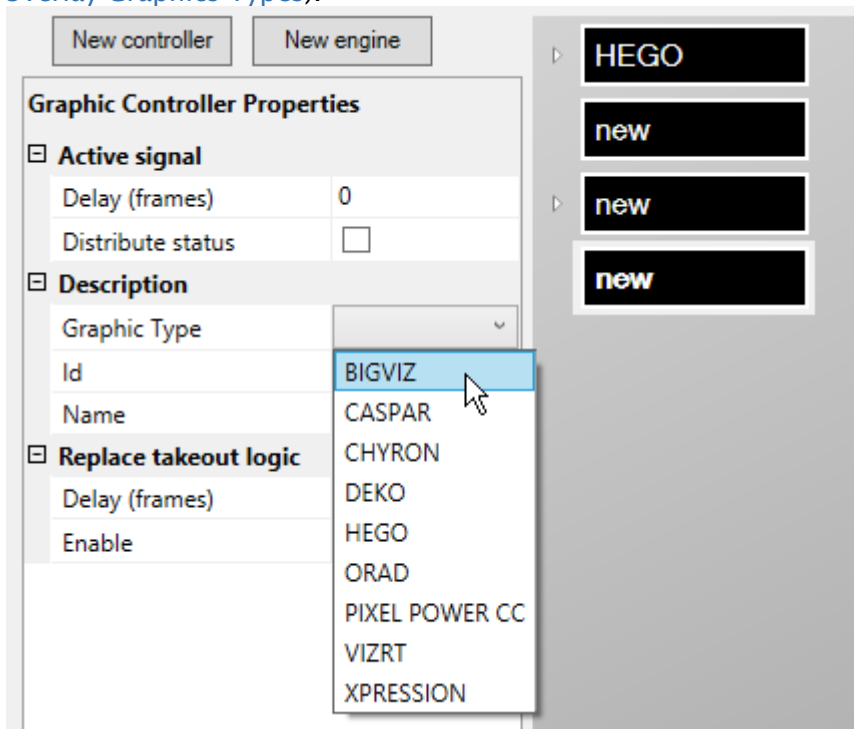
6.7.4 Add Overlay Graphics Controllers and Engines

To Add a New Controller

1. Open the [Overlay Graphics Configuration](#) window by going to **Settings > Properties**.
2. Click **New controller**.



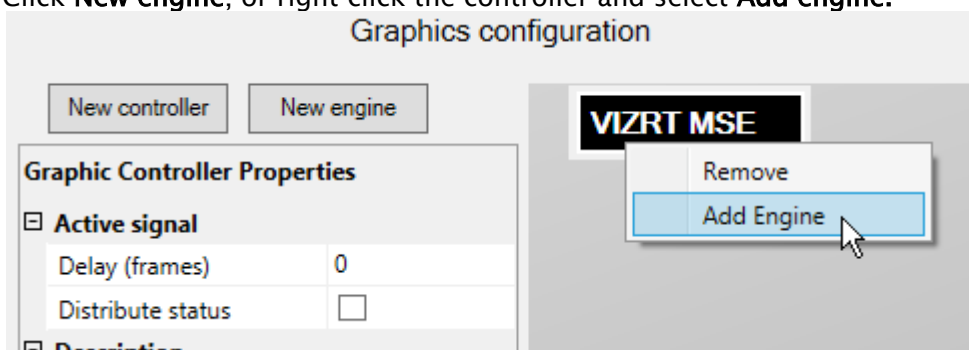
3. In the **Graphic Type** drop-down, select the type of graphics system you are using (see [Overlay Graphics Types](#)).



4. Set the Controller properties according to the selected **Graphic Type**. See corresponding section for each 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 Add a New Engine

1. Open the [Overlay Graphics Configuration](#) window by going to **Settings > Properties**.
2. In 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 corresponding section for each graphic type.
5. Optionally, set actions for selected system events using the [Actions Tab \(Event and Action Rules\)](#).

6.7.5 Common Graphic Controller Properties

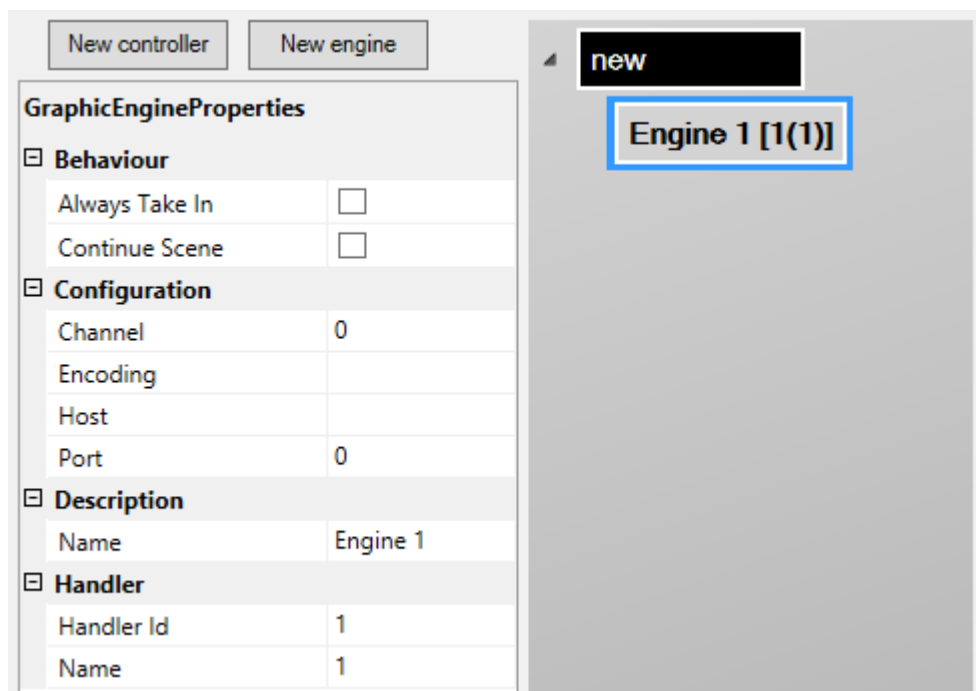
The **Graphic Controller Properties** that are available depend on the Graphic Type of the controller. The controller properties common to most graphics types are described below.

Additional properties for each Graphic Type are described in [Overlay Graphics Types](#).

- **Active signal:** The Active status signal is a signal sent to AvAutomation containing On Air status of a graphics engine.
 - **Distribute status:** Signals the On Air/usage status of an engine to AvAutomation when set to **True**. Status message will be sent to AV Automation log area, Manus console and OverLayGraphics 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 (i.e. **Distribute status** is false).
- AvAutomation 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).
 - Executes direct takes configured on the Switcher preferences page in **Overlay DirectTake control**.
 - 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 and use this 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.

6.7.6 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](#).

- **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 [Actions Tab \(Event and Action Rules\)](#) for other engine settings.

6.7.7 Overlay Graphics Types

The following **Graphics Types** are currently supported.

- VIZRT
- CHYRON
- DEKO
- ORAD
- PIXEL POWER CC
- XPRESSION
- BIGVIZ


VIZRT

Graphic Type = Vizrt (Vizrt via *Media Sequencer* [shortname: MSE])

Vizrt properties		Vizrt engine	
<input type="checkbox"/> Active signal		<input type="checkbox"/> Configuration	
Delay (frames)	0	Channel	0
Distribute status	<input type="checkbox"/>	Encoding	
<input type="checkbox"/> Description		Host	
Graphic Type	VIZRT ▾	Port	0
Id	0	Preview Engine	
Name	VIZRT MSE	<input type="checkbox"/> Description	
<input type="checkbox"/> Media Sequencer		Name	Engine 2
Apply channel name to ne...	<input checked="" type="checkbox"/>	<input type="checkbox"/> Handler	
Host		Handler Id	2
Port	8594	Name	2
<input type="checkbox"/> Media Sequencer Backup			
Host			
Name	Backup		
Port	8594		
<input type="checkbox"/> Playlist			
MOS playlist			
Use concept override	<input type="checkbox"/>		
<input type="checkbox"/> Replace takeout logic			
Delay (frames)	0		
Enable	<input type="checkbox"/>		

- **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**.

CHYRON

Graphic Type = Chyron (Chyron Lyric)

Chyron properties	
Active signal	
Delay (frames)	0
Distribute status	<input type="checkbox"/>
Description	
Graphic Type	CHYRON <input type="button" value="v"/>
Id	0
Name	CHYRON LYRIC
Replace takeout logic	
Delay (frames)	0
Enable	<input type="checkbox"/>

Chyron engine	
Behaviour	
Always Take In	<input type="checkbox"/>
Continue Scene	<input type="checkbox"/>
Configuration	
Channel	0
Encoding	
Host	
Port	0
Description	
Name	Engine 25
Handler	
Handler Id	25
Name	25

- **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](#).

DEKO

Graphic Type = Deko

Deko properties	
Active signal	
Delay (frames)	0
Distribute status	<input type="checkbox"/>
Description	
Graphic Type	DEKO <input type="button" value="v"/>
Id	0
Name	DEKO
Replace takeout logic	
Delay (frames)	0
Enable	<input type="checkbox"/>

Deko engine	
Behaviour	
Always Take In	<input type="checkbox"/>
Continue Scene	<input type="checkbox"/>
Configuration	
Channel	0
Encoding	
Connection	
COM port	
Machine ID	0
Description	
Name	Engine 19
Handler	
Handler Id	19
Name	19

- **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.

ORAD

Graphic Type = Orad (Orad Maestro)

Orad properties		Orad engine	
Active signal		Behaviour	
Delay (frames)	0	Always Take In	<input type="checkbox"/>
Distribute status	<input type="checkbox"/>	Continue Scene	<input type="checkbox"/>
Description		Configuration	
Graphic Type	ORAD	Channel	0
Id	0	Encoding	
Name	ORAD MAESTRO	Host	
Replace takeout logic		Port	0
Delay (frames)	0	Description	
Enable	<input type="checkbox"/>	Name	Engine 24
		Handler	
		Handler Id	24
		Name	24

- **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](#).

PIXEL POWER CC

Graphic Type = Pixel Power CC (Pixel Power Control Center)

Pixel Power Control Center properties	
Active signal	
Delay (frames)	0
Distribute status	<input type="checkbox"/>
Description	
Graphic Type	PIXEL POWER CC ▾
Id	0
Name	PIXEL POWER CC
Replace takeout logic	
Delay (frames)	0
Enable	<input type="checkbox"/>

PixelPower CC engine	
Behaviour	
Always Take In	<input type="checkbox"/>
Continue Scene	<input type="checkbox"/>
Configuration	
Channels	0
Encoding	
Host	
Port	0
Description	
Name	Engine 23
Handler	
Handler Id	23
Name	23

- **Pixel Power Control Center properties:** The properties are mostly the same as described in [Common Graphic Controller Properties](#).
- **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.

XPRESSION

Graphic Type = Xpression (Ross Xpression)

XPression properties	
<input type="checkbox"/> Active signal	
Delay (frames)	0
Distribute status	<input type="checkbox"/>
<input type="checkbox"/> Description	
Graphic Type	XPRESSION ▾
Id	1
Name	ROSS XPRESSION
<input type="checkbox"/> Replace takeout logic	
Delay (frames)	0
Enable	<input type="checkbox"/>

XPression engine	
<input type="checkbox"/> Behaviour	
Always Take In	<input type="checkbox"/>
Continue Scene	<input type="checkbox"/>
<input type="checkbox"/> Configuration	
Channel	0
Encoding	
Host	
Port	0
<input type="checkbox"/> Description	
Name	Engine 22
<input type="checkbox"/> Handler	
Handler Id	22
Name	22

- **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](#).

BIGVIZ

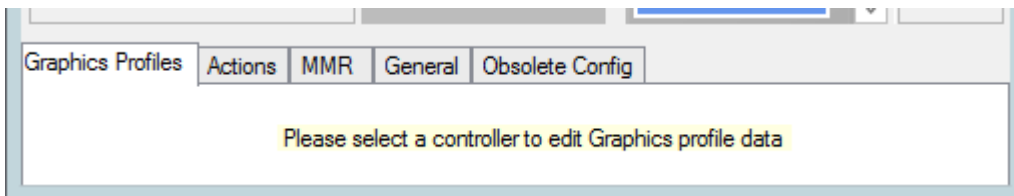
Graphic Type BigViz is a BBC developed system that integrates with ENPS, Vizrt graphics, IBIS Playout, and Autoscript.

BigViz properties	
Active signal	
Delay (frames)	0
Distribute status	<input type="checkbox"/>
Connection	
Encoding	UTF-8
Host	
Port	0
Connection Backup	
Host	
Name	
Port	0
Description	
Graphic Type	BIGVIZ
Id	0
Name	BIGVIZ
Replace takeout logic	
Delay (frames)	0
Enable	<input type="checkbox"/>

BigViz engine	
Behaviour	
Always Take In	<input type="checkbox"/>
Continue Scene	<input type="checkbox"/>
Configuration	
Channel	0
Client ID	0
Encoding	
Host	
Port	0
Description	
Name	Engine 18
Handler	
Handler Id	18
Name	18

- **BigViz properties:** Controller connection properties are not used. All connectivity to BigViz is configured as part of the 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**.

6.7.8 Overlay Graphics Configuration Property Tabs



The tabs functions are:

- [Graphics Profiles Tab](#)
- [Actions Tab \(Event and Action Rules\)](#)
- [MMR Tab](#)
- [General Tab](#)

- [Obsolete Config Tab](#)

Graphics Profiles Tab

The selection of graphics profile changes 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 open, go to **Settings > Properties > Graphics Profiles** tab. You must then select the controller (black) that you want to configure.

The list of Graphics Profiles changes dynamically depending on which controller is selected. When the graphics profile is changed in the Overlay Graphics Interface or the Viz Mosart GUI, it looks through all the controllers. If a controller has that particular graphics profile, it changes the show path, VCP concept (Viz only) and default scene on wall values.

The Overlay Graphics Interface application must be restarted for any changes to take effect.

The screenshot shows the 'Graphics configuration' window with the 'Mosart graphic' section. The 'VIZRT' controller is selected. Below the controller list, the 'Graphics Profiles' tab is active, showing a table of profiles. A red arrow points from the 'VIZRT' controller to the 'Graphics Profiles' tab, and another red arrow points from the 'Graphics Profiles' tab to the table.

	Display Name	ShowPath	VCP concept	Default scene on wall	Default
	BBC24		BBC24	London	<input checked="" type="checkbox"/>
	BBC WORLD		BBCWORLD	World	<input type="checkbox"/>
	BBC FOOD		BBCFOOD	Kitchen	<input type="checkbox"/>
*					<input type="checkbox"/>

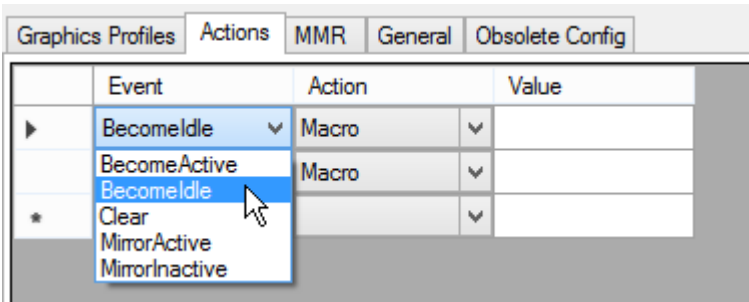
- **Display Name:** Name of the Graphics Profile that is displayed to the user. The Graphics Profiles are displayed as menu options both in the Mosart GUI and in Overlay Graphics.
- **ShowPath:** Path to the files the graphics engine uses. Absolute path where graphics are located within the graphics 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.

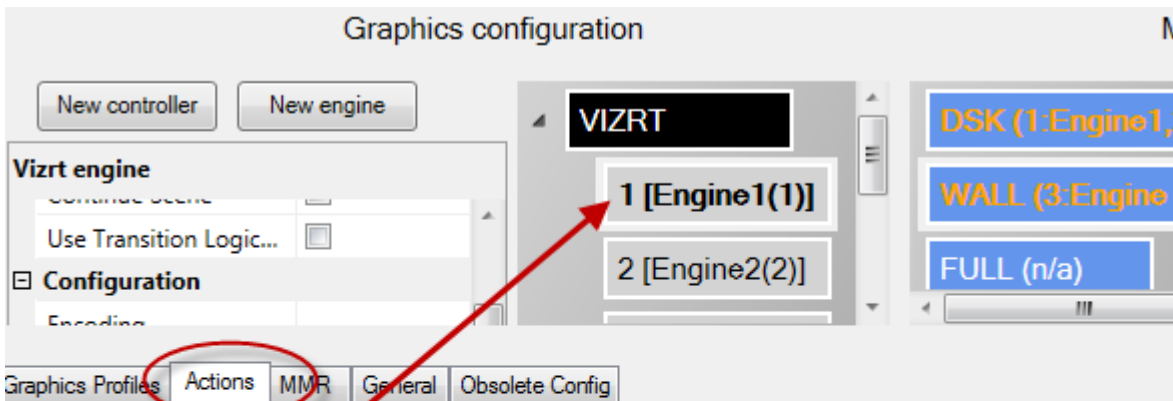


Actions Tab (Event and Action Rules)

To open go to **Settings > Properties > Actions** tab.



In the Actions tab, any number of event-action rules can be set up. The engine you want to configure these actions for must be selected. The rules list changes dynamically depending on which engine is selected.



Event	Action	Value
BecomeActive	Macro	E1
BecomeIdle	Macro	E2
Clear	TakeNamed...	LOGO OFF
*		

Three parameters are defined in order 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](#).
- **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

To open, go to **Settings > Properties > MMR** tab.

Engine	Destination
Engine 1 <input checked="" type="checkbox"/>	DSK
Engine 2 <input checked="" type="checkbox"/>	WALL
Engine 3 <input type="checkbox"/>	
Engine 4 <input type="checkbox"/>	
Engine 5 <input type="checkbox"/>	
Engine 6 <input type="checkbox"/>	
Engine 7 <input type="checkbox"/>	
Engine 8 <input type="checkbox"/>	

- **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 [AV Automation Devices 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.

General Tab

To open, go to **Settings > Properties > General** tab.



- **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 [AV Automation Devices Graphics](#)).

Obsolete Config Tab

To open, go to **Settings > Properties > Obsolete Config** tab.

Engine	Handler name	Host	Port	Default Scene	Destination	Preview engine
Engine 1 <input checked="" type="checkbox"/>	CGA	localhost	10003		DSK	1
Engine 2 <input type="checkbox"/>	CGB	localhost	10003		WALL	2
Engine 3 <input type="checkbox"/>	CGC	localhost	10003			3
Engine 4 <input type="checkbox"/>	CGD	localhost	10003			4

This tab is used to convert overlay graphics configuration from Viz Mosart 3.4 or earlier versions.

IMPORTANT! If you have configuration based on Viz Mosart 3.5 or later, then *DO NOT* use the Obsolete Config tab. If you have a configuration based on Viz Mosart 3.4 or earlier, then it is 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. To convert old configurations to the new standard, click the **CONVERT** button. This overwrites any existing standard configuration.

The fields in this tab are:

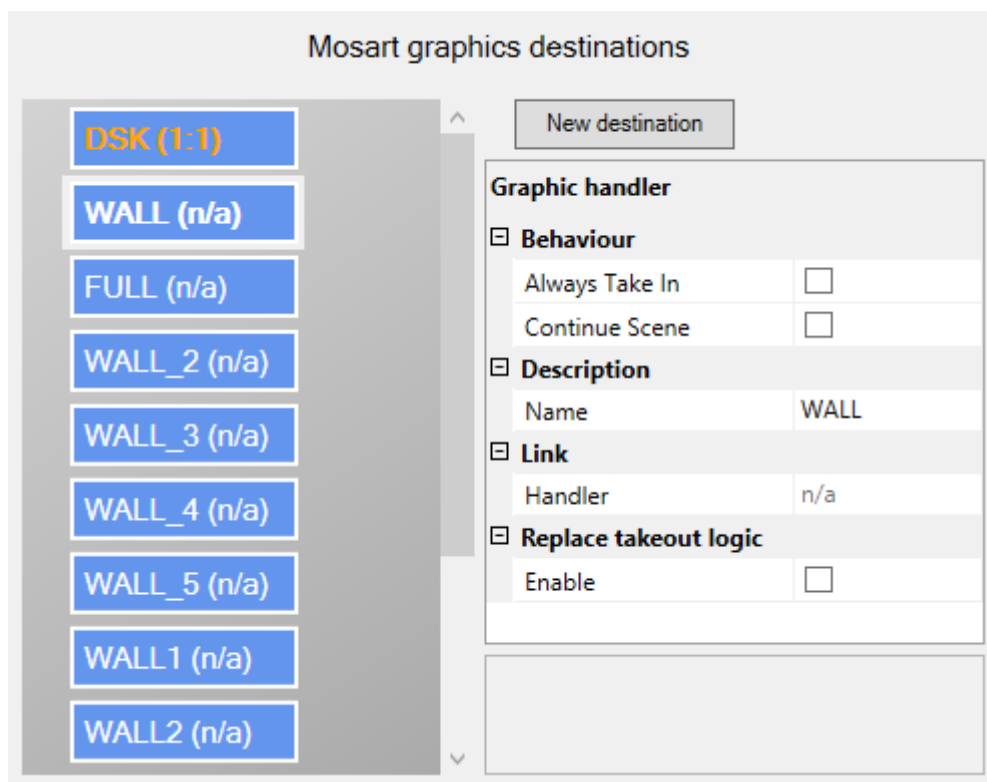
- **Type:** The type of graphics used
- **Engine 1-8 (check box):** Check the graphics engines to be active. When converting to new configuration, only active engines are converted.
- **Handler Name:** Default CGA, CGB etc.
- **Host:** Hostname or IP-address of graphics engine.
- **Port:** TCP/IP port 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.

6.7.9 Adding a Mosart Graphics Destination

This section presents configuration of overlay graphics for various system setups.

- [Mosart Media Router \(MMR\) and Mirrored Graphics](#)

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

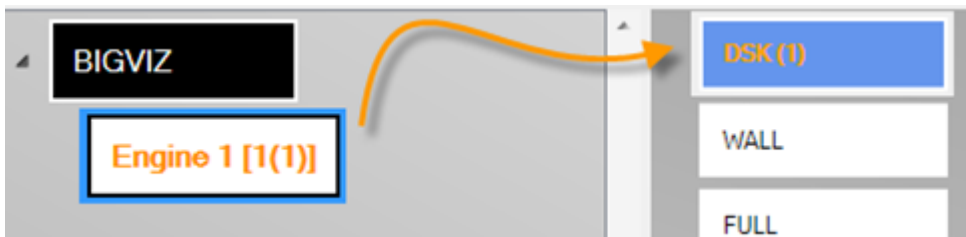
The properties displayed depend on the graphic handler used. For some handlers (e.g. Viz Engine) the full set of or possible properties will be shown. For others, a simplified set of properties will show: **Description** and **Link**.

- **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

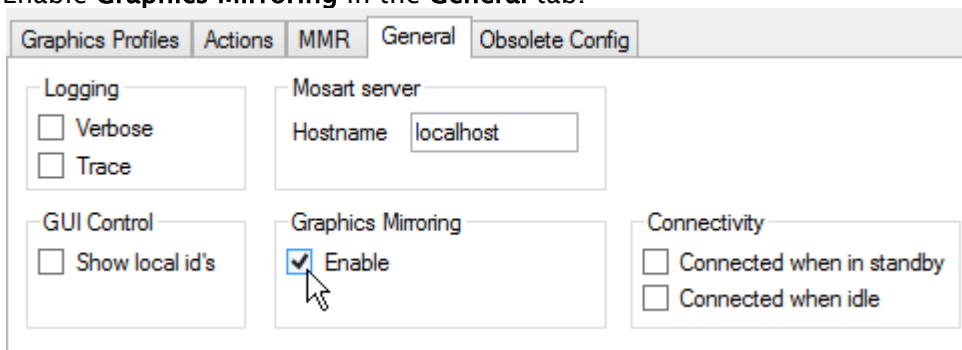
- Mark an engine.
- **Right-click** to display the context menu context menu and select **Remove**.

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.

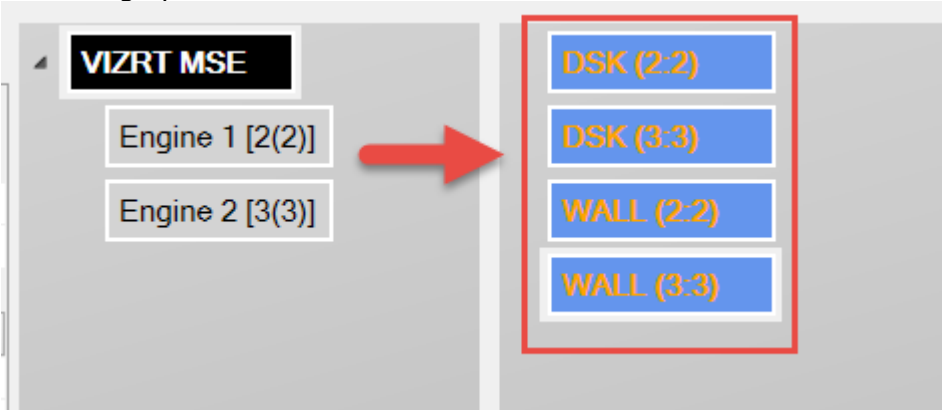
⚠ For using Mosart Media Router (MMR) in conjunction with Mirrored Graphics, see [Mosart Media Router \(MMR\) and Mirrored Graphics](#) below.

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.



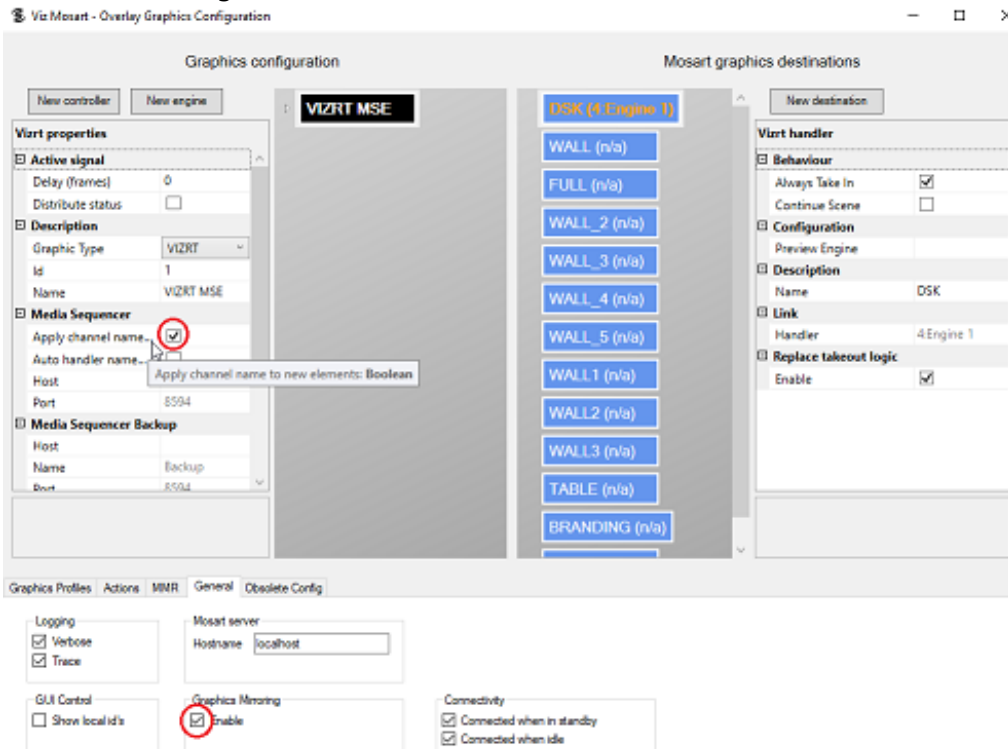
- (Optional): Configure actions to be triggered when changing the active mirrored graphics engine.
Do this from the [Actions Tab \(Event and Action Rules\)](#).

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.

- 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 Audio Player

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

The Audio Player supports the formats: mp3, wav, wma, aac, m4a, mp4, aiff, avi.

 **Note:**

- A broadcast sound card is required for audio output. If you are running the Audio Player on a Windows Server, also install Windows Audio features on this machine, otherwise the various audio formats will not work.
- The Audio Player replaces the discontinued *Soundfile Player*.

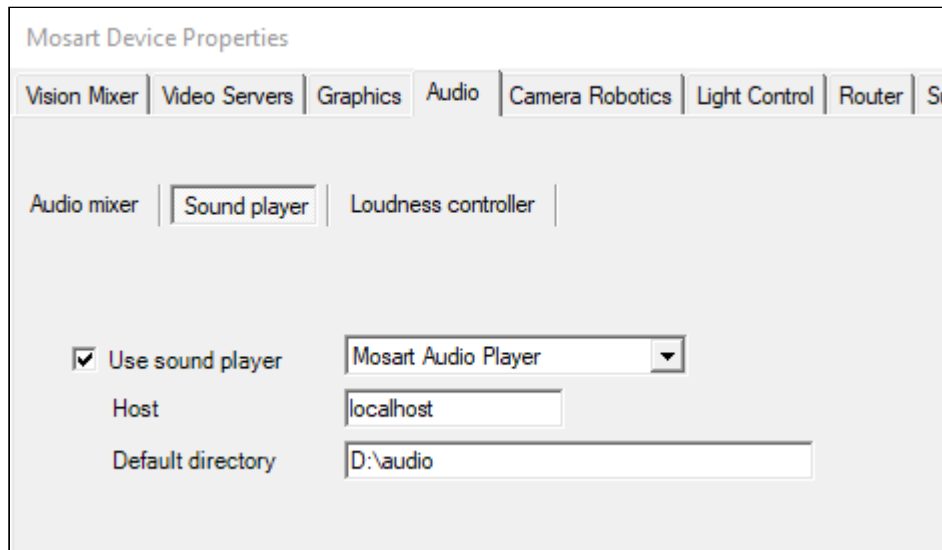
- [Setting Up the Audio Player](#)
- [Audio Player Settings](#)
- [Audio Player Configuration File](#)

7.1 Setting Up The Audio Player



1. Install Audio Player, then start it.
2. Open AV Automation.
3. Go to **Devices > Preferences > Audio > Sound player**.
4. Check **Use sound player** and choose *Mosart Audio Player*.
5. Under **Host** write the IP address to the computer where Audio Player is running.

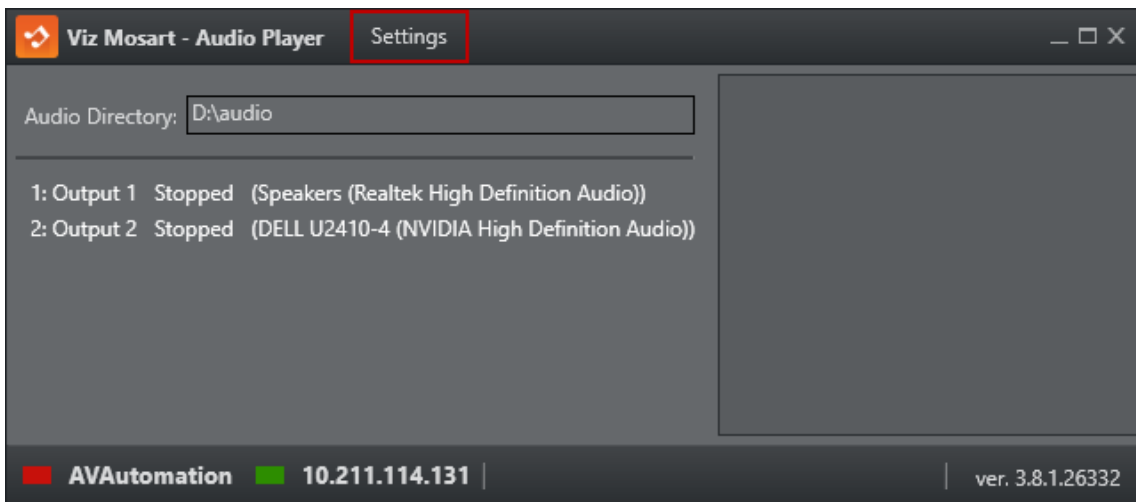
- Under **Default directory**, write the folder path where the audio files are stored.



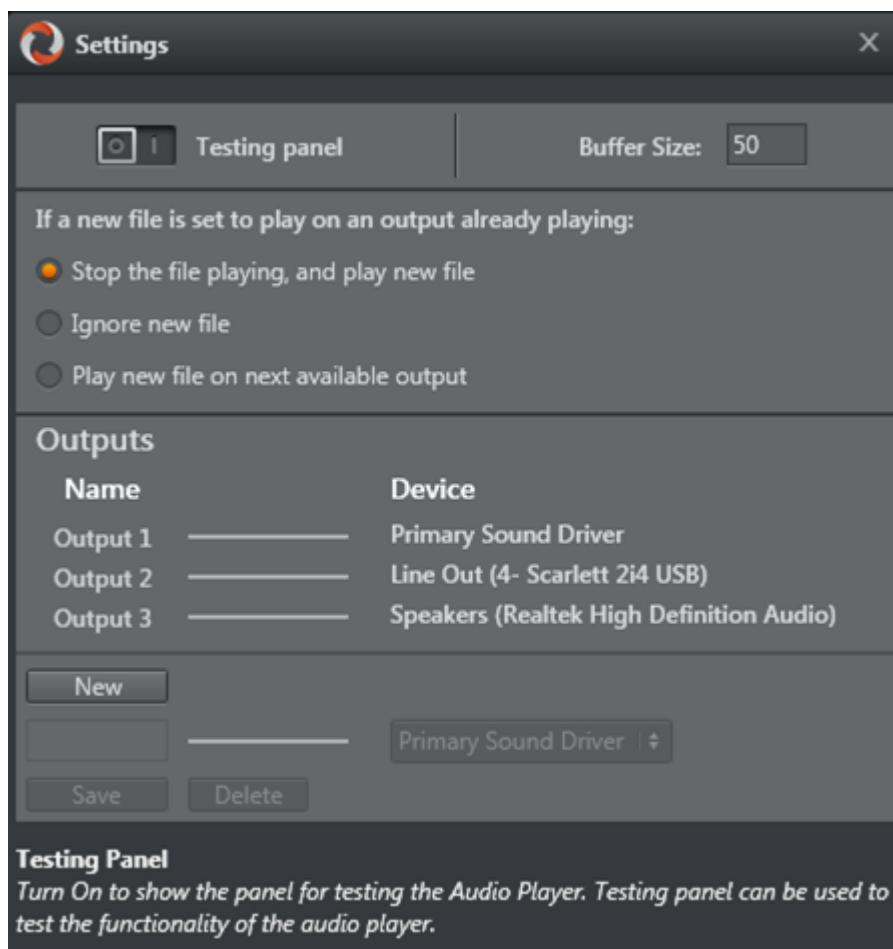
- Restart AV Automation and check that the connection is green.

7.2 Audio Player Settings

- Click **Settings** in the Viz Mosart Audio Player to open the settings dialog:

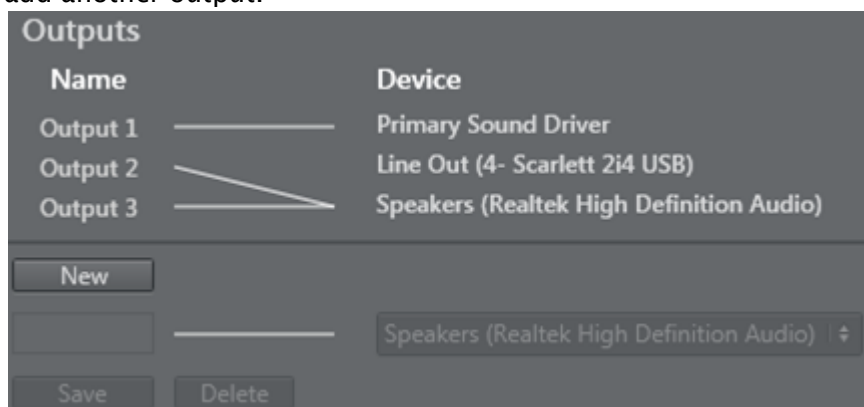


- Testing Panel:** Shows the panel for testing the Audio Player. The testing panel is used to test the functionality of the audio player.
- Buffer Size:** Determines the size of the playback buffer (Higher = smoother playback, Lower = lower response time).



- **File already playing:** Controls what Audio Player does if a new file is played when there is already a file playing.
 - Stop the file playing, and play new file
- **Output editor:** The list on the left shows all the outputs, and the list on the right shows all available Playback devices on the computer. The line between them shows which playback device the output uses to play the audio file. You can click on the outputs to map them to another playback device, rename the output or delete the output. Click the **New** button to

add another output.



- **Tooltip box:** Shows a tooltip for the setting you are hovering over with your mouse.

Testing Panel

Turn On to show the panel for testing the Audio Player. Testing panel can be used to test the functionality of the audio player.

7.3 Audio Player Configuration File

The Audio Player Configuration file `Mosart Audio Player.exe.config` is found in `%ProgramFiles(x86)%\Mosart Medialab\Mosart Audio Player\`.

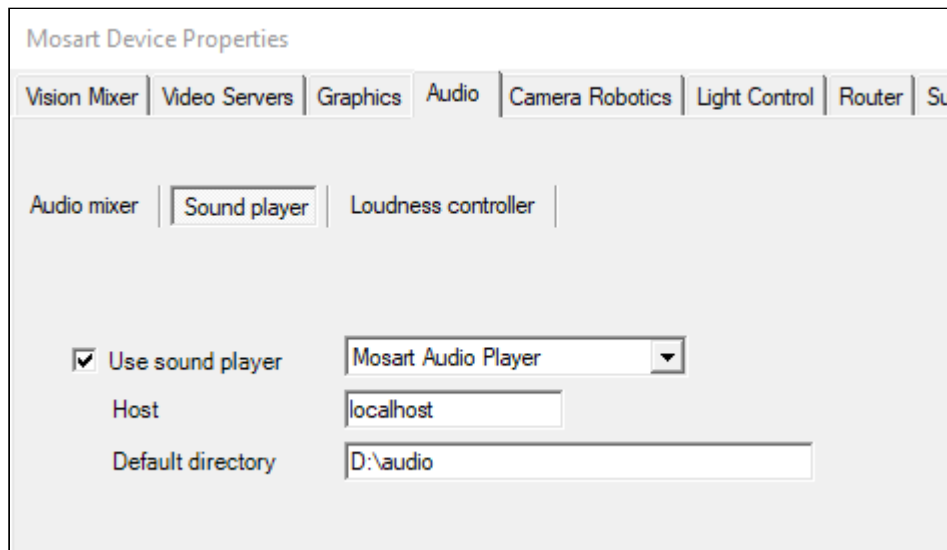
Change the `baseAddress` attribute to use another port:

```
<add baseAddress="http://localhost:8084/LiveSoundService.svc/" />
```

7.4 How To Set Up Audio Player



1. Install Audio Player, then start it.
2. Open AV Automation.
3. Go to **Devices > Preferences > Audio > Sound player**.
4. Check **Use sound player** and choose *Mosart Audio Player*.
5. Under **Host** write the IP address to the computer where Audio Player is running.
6. Under **Default directory**, write the folder path where the audio files are stored.



- Restart AV Automation and check that the connection is green.

7.5 Audio Player Configuration File

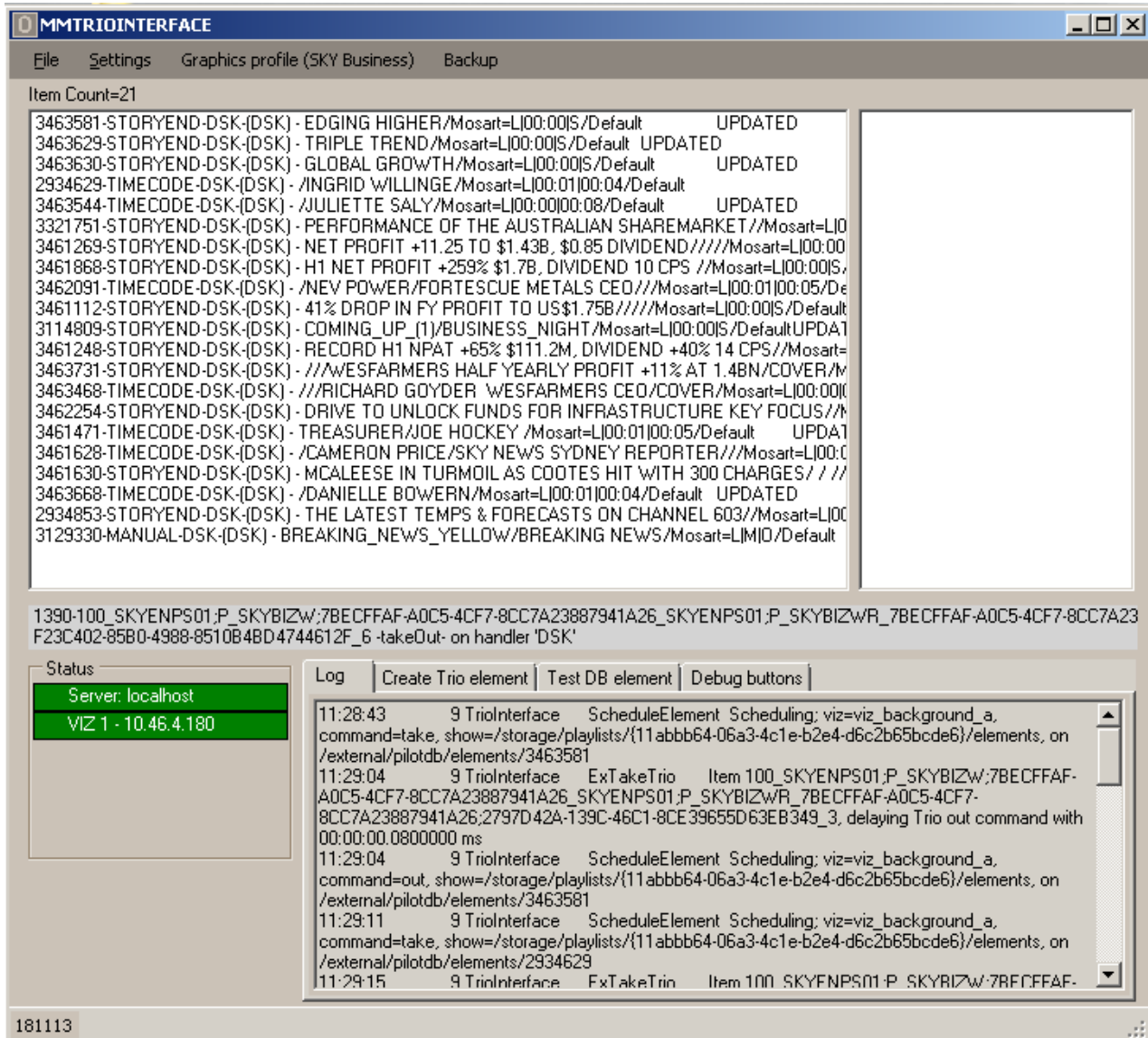
The Audio Player Configuration file `Mosart Audio Player.exe.config` is found in `%ProgramFiles(x86)%\Mosart Medialab\Mosart Audio Player\`.

Change the `baseAddress` attribute to use another port:

```
<add baseAddress="http://localhost:8084/LiveSoundService.svc/" />
```

8 Trio Interface

IMPORTANT! The [Overlay Graphics Interface](#) is the recommended interface for controlling and monitoring overlay graphics for all Viz Mosart approved graphic engines. Trio Interface is no longer recommended, and is kept for backwards compatibility only.



Trio Interface is for controlling and monitoring overlay graphics for Viz Engines. It can be used with either Viz Trio or Viz Pilot, and requires the use of the Media Sequencer to issue commands to the Viz Engine.

This section contains the following topics:

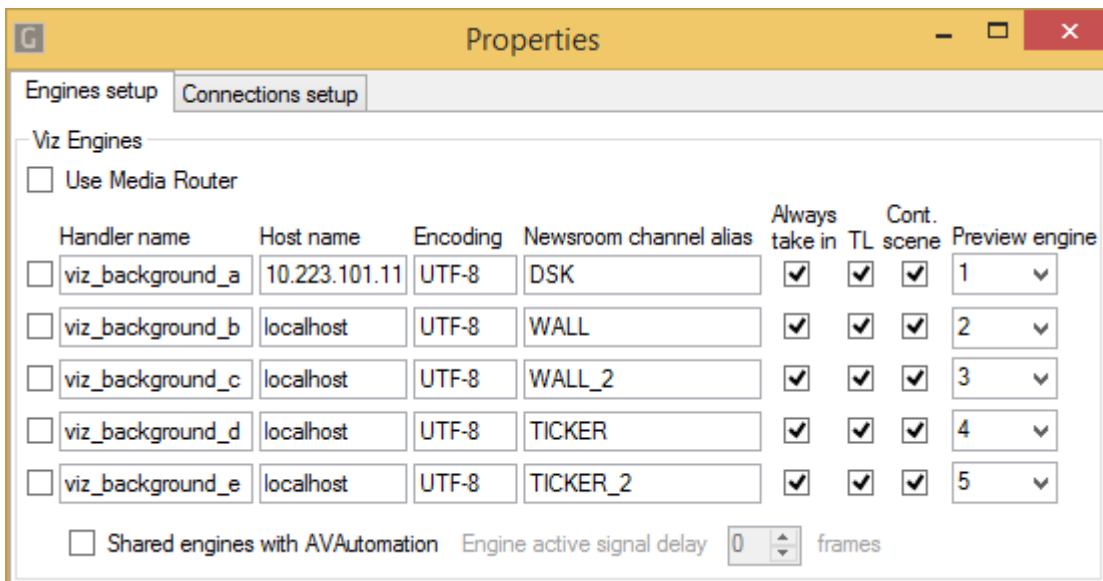
- [Trio Interface Configuration](#)
- [Media Sequencer Redundancy](#)

8.1 Trio Interface Configuration

The Trio Interface Configuration window is used to access the configuration options for **Trio Interface**. To open, go to **Trio Interface > Settings > Properties**. The Properties window contains the [Engines Setup Tab](#) and [Connections Setup Tab](#).

8.1.1 Engines Setup Tab

Go to **Trio Interface > Settings > Properties > Engines setup** tab.



- **Viz Engines**

- **Use Media Router:** Uses Media router when selected.
- **Viz Engines:** Activates or deactivates connection to Viz Engines configured in the corresponding row.
- **Handler name:** Internal Media Sequencer handler name.
- **Host name:** IP address or hostname of Viz Engine.
- **Encoding:** Font encoding on the Viz Engine. Use UTF-8.
- **Newsroom channel alias:** Alias to be used from the NCS to direct the lower third or video wall element to the correct engine.
- **Shared engines with AVAutomation:** Enable the check box if the Trio Interface and the AV Automation are sharing play-out on the engines. If enabled, the Trio Interface will signal the AV Automation when there are no Viz Mosart controlled lower thirds on-air, allowing for any cued full screen graphics controlled through the AV Automation to re-cue. The engine configuration must be the same in both applications.
- **Always take in:** Forces a take-in command to be sent.
- **TL:** Specifies that Transition Logic is used.
- **Cont. scene:** Use Continue for takeout of scene based graphics.
- **Preview engine:** Indicates which Vizrt engine to be used for preview of overlays.

- **Shared engines with AVAutomation:** This functionality assumes that the graphic configuration in AV Automation is equal to in Viz Trio. Viz Trio will protect the engine from being used by AV Automation when CG is active in Viz Trio.

• **Shows**

Shows				
	Display name	Show path	VCP concept	Default
*				<input type="checkbox"/>

- **Display name:** The name of the show shown in the drop down list in the Viz Mosart GUI. Please note that when using Concept Override, this name must be identical to the Viz Pilot concept.
- **Show path:** Determines the Viz Trio path for the show.

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

- **VCP concept:** Name of the Viz Pilot concept.
- **Default:** Selects the row the default show for the gallery.

• **Logic:** Transition Logic Configuration.

Logic

Use Continue for take out of scene based graphics Use Transition Logic aware take in/out commands

Take out delay frames

- **Use Continue for take out of scene based graphics:** Enable if all scene based graphics has an out animation that can be triggered with a continue command to the Viz Engine. Disable to force all take outs on scene based graphics to be taken out with a cut.
- **Use Transition Logic aware take in/out commands:** Activates the mode where Viz Mosart reads out the Transition Logic state of all items and keeps track of whether an item replaces another or is played back to back when checked.
- **Take out delay:** Delay before takeout. Default: *Empty* (ss:ff).

• **Playlist**

Playlist

Use MOS VCP Playlist

Playlist name Use Concept Override

- **Use MOS VCP Playlist:** Enable the Playlist check box when using Vizrt MOS items from the newsroom system.
- **Playlist name:** Viz Pilot Playlist on the Media Sequencer that Viz Mosart will populate with the Vizrt MOS items.

⚠ **Note:** The playlist name used here cannot be the same as used in **AV Automation Devices - Graphics > Vizrt Settings > Miscellaneous > Mos Playlist**.

- **Use Concept override:** Select this check box if your scenes have been prepared for the Concept Override functionality in Viz Pilot 5.2 and later.

8.1.2 Connections Setup Tab

Go to **Trio Interface > Settings > Properties > Connections setup** tab.

- **Mosart Server**

- **Hostname:** Hostname or IP address of the computer running the Manus Administrator application. Default: *localhost*.

- **Media Sequencer Engine**

- **Hostname:** Hostname or IP address of the primary Media Sequencer.
- **Backup:** Hostname or IP address of the backup Media Sequencer.
- **Use MSE backup:** Starts using the backup Media Sequencer instead of the primary when checked.
- **Start MSE if not running:** Attempts to start from the location described under (only valid if the Media Sequencer is running as a console application on the same machine) if the Media Sequencer is not running when Trio Interface is started.
- **Path:** Determines the path to *scheduler.exe* in the Media Sequencer program files folder (only if Media Sequencer is running as a console application on the same machine).

- **MSE Database**

- **Configure MSE DB settings:** Updates the database settings in the Media Sequencer with the details below if enabled.
- **Datasource:** TNS name or connection string of the Oracle database.
- **User:** Username on the Oracle database.
- **Password:** Password on the Oracle database.

· **Logging**

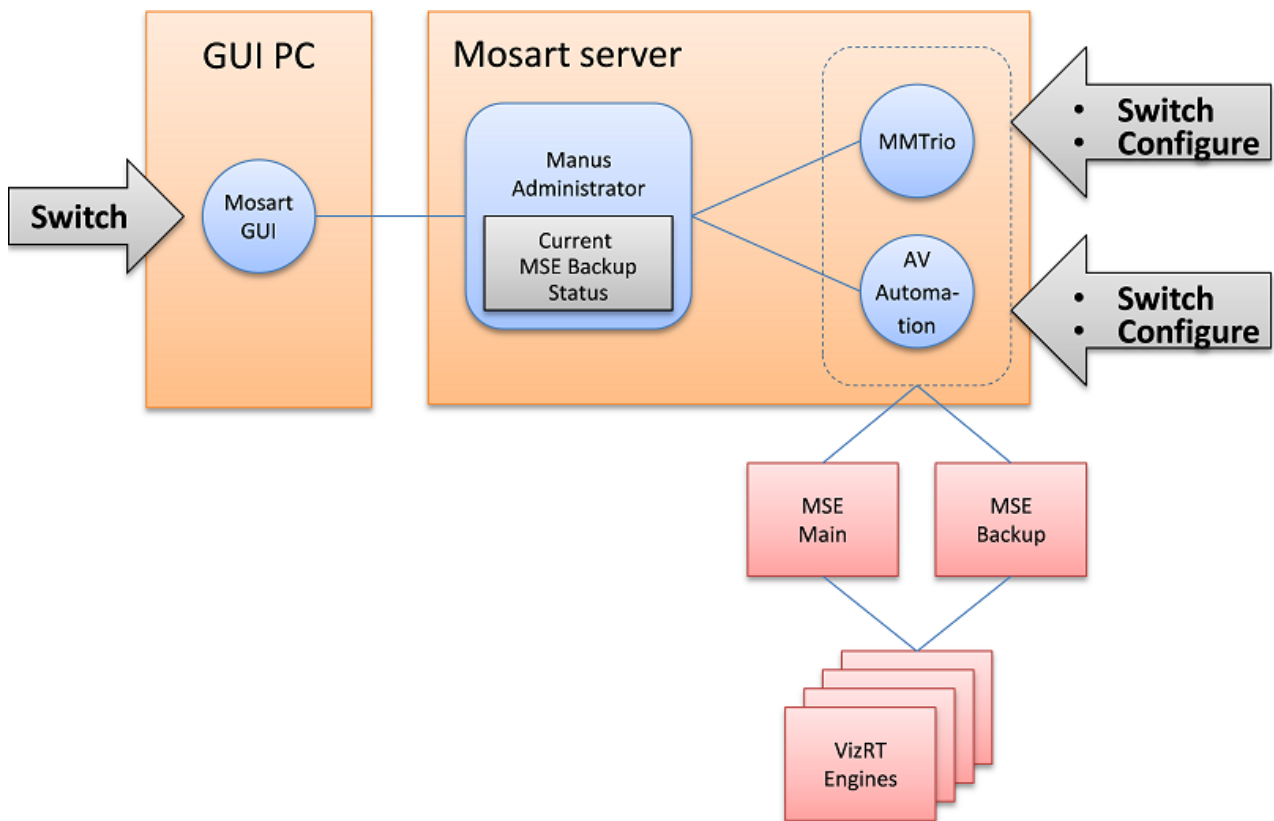


- **Enable:** Sends logging details to the logfile if enabled.
- **Verbose:** Activates verbose logging (increasing the details sent to the log) if enabled.

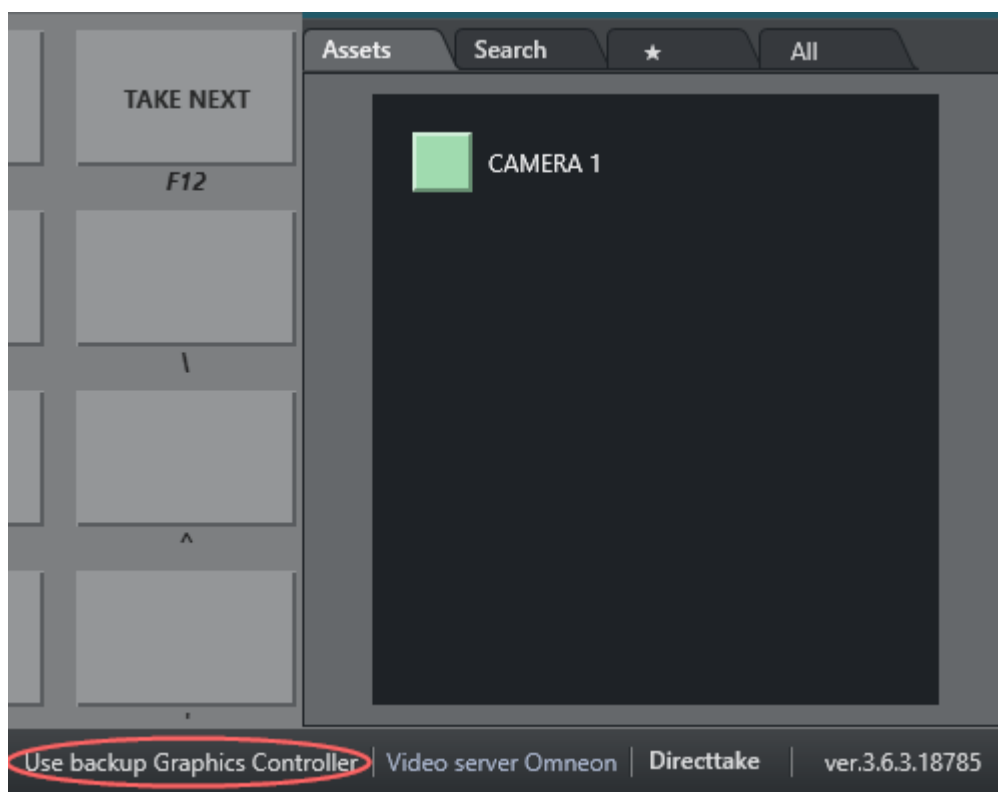
8.2 Media Sequencer Redundancy

This section describes the operation of the main/backup switching of the Media Sequencer. It is possible to select the Main or Backup Media Sequencer, from within Viz Mosart, however there may be a delay of several seconds after switching from one to the other, before the Media Sequencer responds to Viz Mosart commands.

The diagrams below show the configuration and operations possibilities of the various components in the Media Sequencer backup concept. The gray arrows show which operations are available in each Viz Mosart module.



8.2.1 Switching Media Sequencer from the GUI



Switching between main and backup Media Sequencer is normally done in the Viz Mosart GUI. In the bottom right corner, you will find the text: *Use backup Graphic Controller*. This text is clickable and switches Viz Mosart to the backup Media Sequencer. Subsequently, the text changes to *Use main Graphics Controller* on a yellow background, to visibly alert you to the exceptional state.

During the switching process, the alarm indicators in AV Automation momentarily goes red, but goes back to green when the switching process is finished.

Still, the Media Sequencer needs some additional 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.

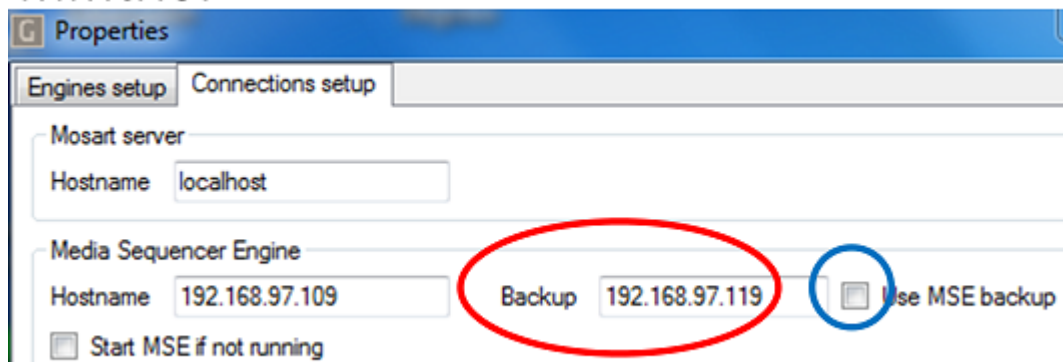
8.2.2 Configuration and Test

Use the fields marked in **red** to configure the use of two Media Sequencers; one main and one backup.

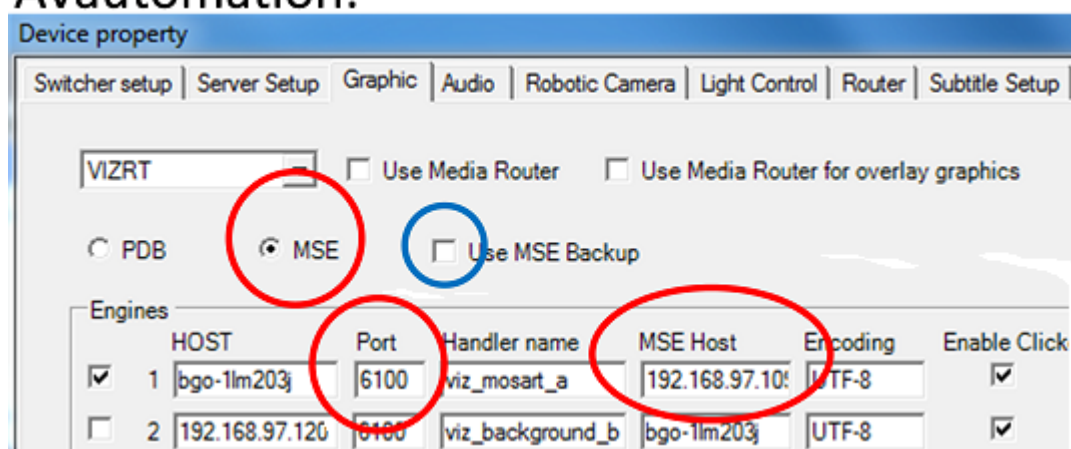
⚠ Note: In AV Automation, in the field *MSE Host*, specify the two IP addresses (one for each Media Sequencer) separated with a semicolon (;). In Trio Interface, there are separate input fields for these addresses.

Use the check box *Use MSE backup* marked in **blue** if you want Viz Mosart to start on the backup Media Sequencer after a restart.

MMtrio:



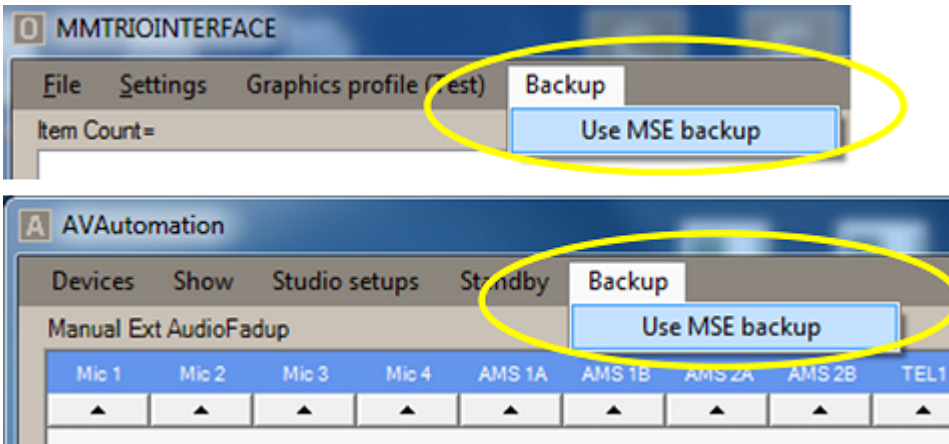
AVautomation:



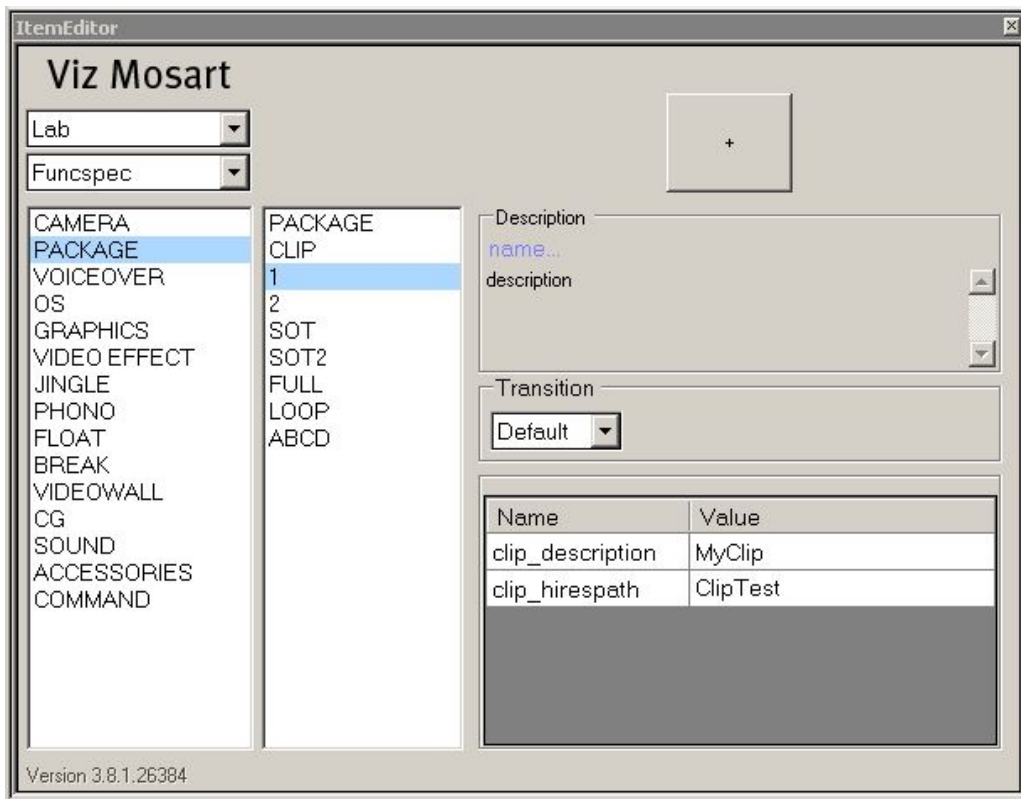
8.2.3 Test and Debug

Both Trio Interface and AV Automation have a separate menu item that enables you to switch each of them between the main and backup Media Sequencer. This switching possibility is just for testing and debugging purposes. It is recommended to use the main switch in the GUI instead.

Note: The yellow background of the GUI Media Sequencer indicator only follows the setting of the AV Automation Media Sequencer selector. The corresponding selector in Trio Interface does not affect this indicator.



9 ActiveX



ActiveX is used to insert Viz Mosart template information into the Newsroom System script. For ActiveX to function correctly, some parameters are required to be set that link ActiveX content to your Viz Mosart Server installation. ActiveX can be linked to the Viz Mosart Server either via the Mosart [Viz Mosart Template Database](#), or directly to the Viz Mosart Server.

This section contains:


- [Setting up the Registry for Viz Mosart ActiveX](#)
- [Connecting the ActiveX to Viz Mosart Server](#)
- [ActiveX Notes](#)

9.1 Setting Up The Registry For Viz Mosart ActiveX

To enable ActiveX features between Viz Mosart and the NRCS, first make configuration in the Viz Mosart machine's registry.

- [Working with Viz Mosart Registry Keys](#)
 - [Ascertaining Machine Architecture](#)
 - [Modifying Registry Keys](#)
- [Viz Mosart ActiveX Registry Properties](#)
 - [Mosart ActiveX Key](#)
 - [Studios Sub-key](#)

- [Configuring the Newsroom System](#)
 - [AP ENPS](#)
 - [Avid iNEWS](#)

 This section assumes knowledge of, and access to the Windows Registry. Normally this means you must run with elevated (Administrator) privileges. Always exercise extreme care when editing the Windows registry.

It is recommended to either

- Create a Windows restore point (from the **Start menu** > Type *create a restore point* > **Create a restore point** > **Create** or
- Create a backup of the registry with the `regedit.exe` application with **File** > **Export** before you change any registry keys.

9.1.1 Working with Viz Mosart Registry Keys

Before continuing, see the [Warning](#) above.

Ascertaining Machine Architecture

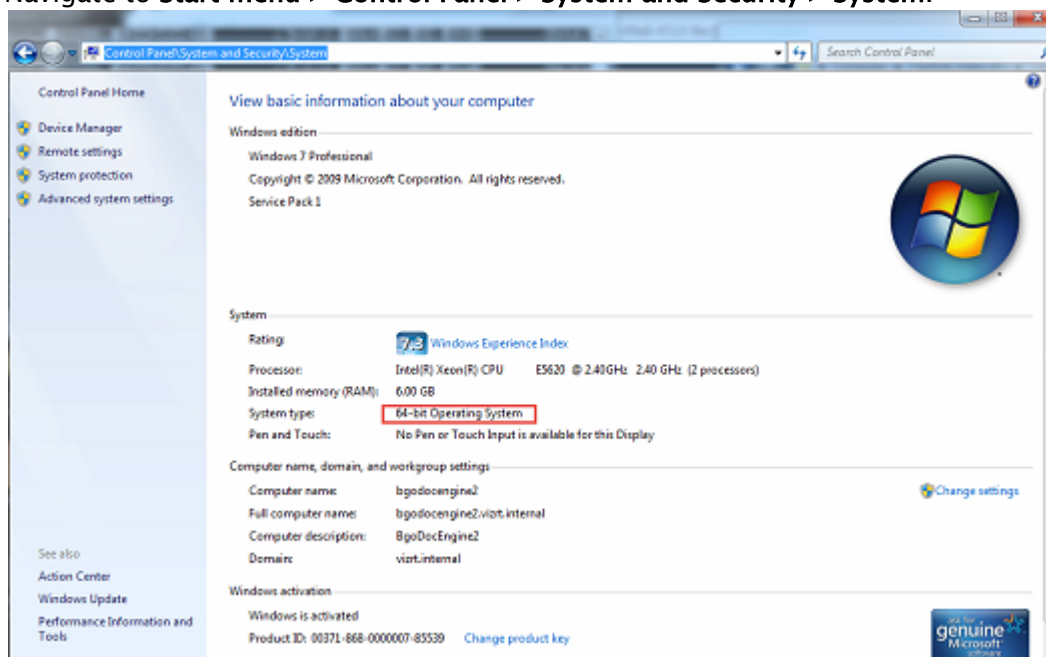
Whilst the majority of machines in commercial service are 64-bit, you can rapidly verify this basic architectural property.

To Verify 32 or 64 bit environment

- **Windows 10**
View the architecture with **Start menu** > **Settings** > **System** > **About this PC**.

- **Windows 7**

Navigate to **Start menu > Control Panel > System and Security > System**.



- Alternatively, run a script or the command line. The example below is a Powershell:

```

show architecture
1  if ((gwm i win32_operatingsystem | select
2      osarchitecture).osarchitecture -eq "64-bit") {
3      #64 bit logic here
4      Write "This PC is running a 64-bit OS"
5  } else {
6      #32 bit logic here
7      Write "This PC is running a 32-bit OS"
  }

```

Modifying Registry Keys

To Browse or Change Registry Keys

- From the Windows **Start menu** type regedit.
 - Registry settings for the Viz Mosart ActiveX on 32-bit machines are found under the registry key:
HKEY_LOCAL_MACHINE/SOFTWARE/Mosart Medialab/Mosart ActiveX.
 - Registry settings for the Viz Mosart ActiveX on 64-bit machines (x64) are found under the registry key:
HKEY_LOCAL_MACHINE/SOFTWARE/Wow6432Node/Mosart Medialab/Mosart ActiveX.

9.1.2 Viz Mosart ActiveX Registry Properties

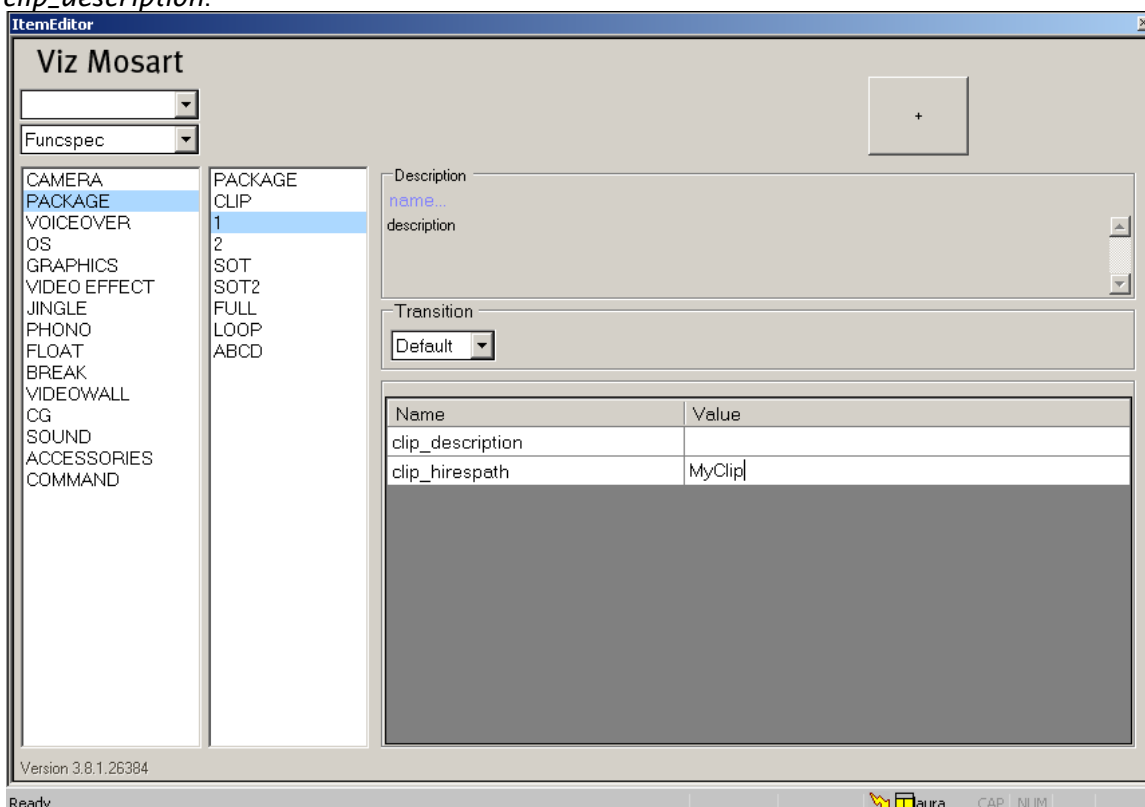
First navigate to the required keys, as described in [Modifying Registry Keys](#) above. This section lists the keys and their values.

Mosart ActiveX Key

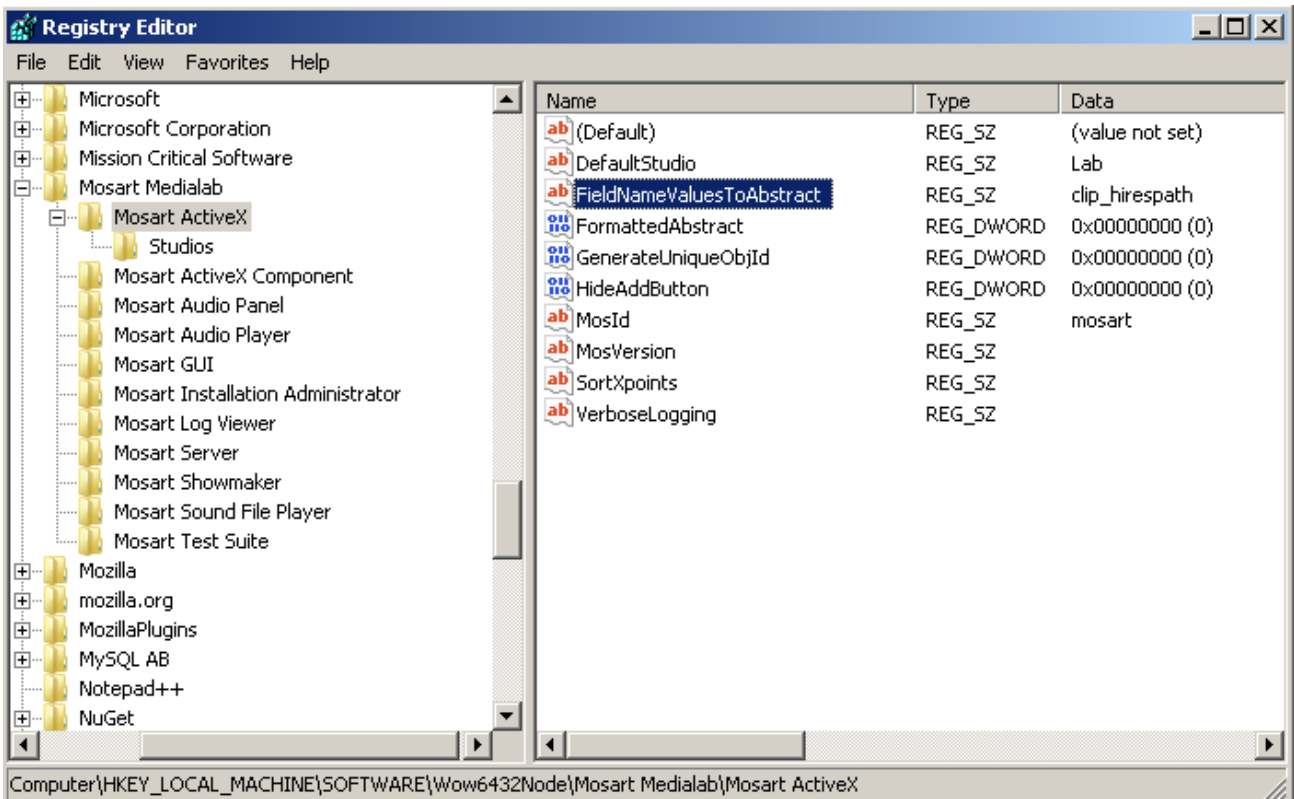
- **DefaultStudio:** *Optional.* The entry of the studio setup in the [Studios](#) sub-key. A blank value selects the (Default) entry.
- **FieldNameValuesToAbstract:** *Optional.* A list of newsroom tags separated by colon or semicolon used to create the content of the mosAbstract tag of a MOS object generated with Mosart ActiveX. See also [Newsroom Tags](#).

Example:

Here, the **template type=PACKAGE** and **variant=1** with two **newsroom tags**, *clip_hirespath* and *clip_description*:



In the registry, **FieldNameValuesToAbstract** is set to *clip_hirespath*:



In the MOS object generated by Viz Mosart ActiveX, the tag `mosAbstract` contains the **clip ID**:

MOS object generated with Mosart ActiveX

```

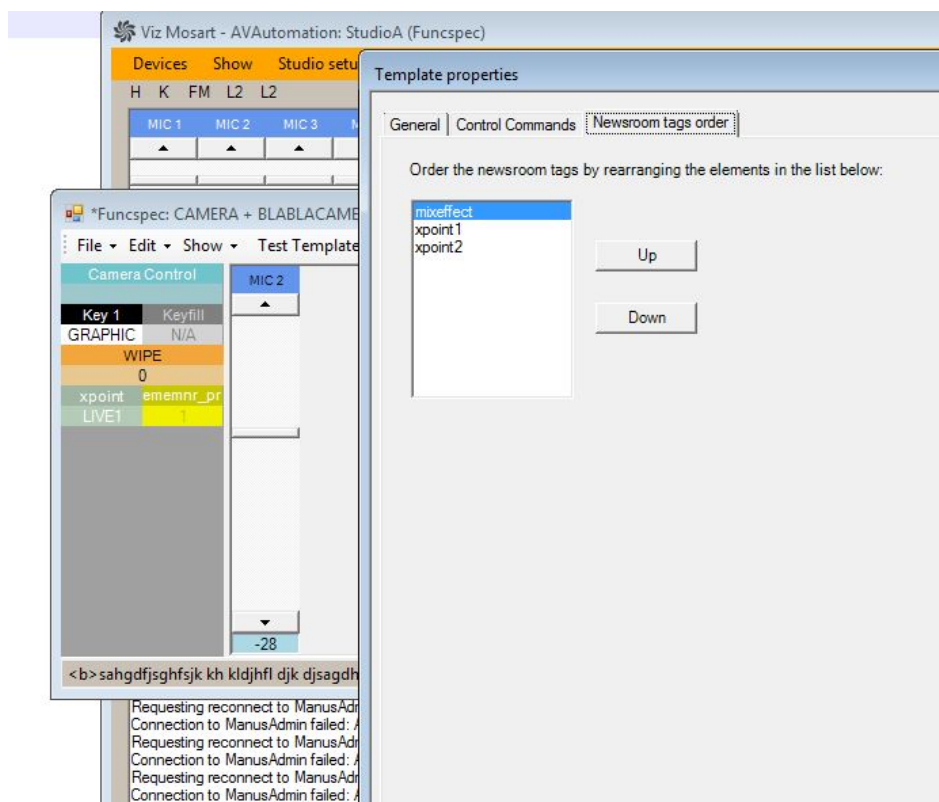
<mos>
  <ncsItem>
    <item>
      <itemID>0</itemID>
      <objID>PACKAGE;1</objID>
      <mosID>mosart</mosID>
      <mosPlugInID>Mosart.ActiveX</mosPlugInID>
      <mosItemBrowserProgID>Mosart.ActiveX</mosItemBrowserProgID>
      <mosItemEditorProgID>Mosart.ActiveX</mosItemEditorProgID>
      <mosAbstract>MyClip</mosAbstract>
      <mosExternalMetadata>
        <mosScope>PLAYLIST</mosScope>
        <mosSchema>http://www.mosartmedialab.no/schema/mositem.dtd</mosSchema>
        <mosPayload>
          <mosarttemplate>
            <type name="PACKAGE" category="">
              <variants value="1" fieldtype="LIST">
                <variant name="1">
                  <fields>
                    <field name="clip_description" default=""
fieldtype="TEXT" keylist="" />
                    <field name="clip_hirespath" value="MyClip"
default="" fieldtype="TEXT" keylist="" />
                  </fields>
                </variant>
              </variants>
            </type>
          </mosarttemplate>
        </mosPayload>
      </mosExternalMetadata>
    </item>
  </ncsItem>
</mos>

```

- If **FieldNameValuesToAbstract**=*clip_hirespath;clip_description* in the registry, and **clip_description** is set to "Test" in Viz Mosart ActiveX for the same template taken as example above, then **mosAbstract** is built using the value of **clip_hirespath** and **clip_description** separated by space (<mosAbstract>MyClip Test</mosAbstract>).
- If **FieldNameValuesToAbstract** is set to an invalid value (for example, a naming error in the newsroom tag name), then **mosAbstract** tag is set to <template type> <template variant> followed by the values of newsroom tags found on the template separated by space. Following the example above, then *mosAbstract* is <mosAbstract>PACKAGE 1 MyClip Test</mosAbstract>.

Value Descriptions

- **FormattedAbstract:** Setting this value to 1 inserts HTML codes and newlines in the MOS Abstract field of the MOS item. Use value 0 to have a plain format of the text from MOS Abstract field. *Optional.* Valid values: 0 or 1.
- **GenerateUniqueObjId:** Enables (when set to 1) or disables (when set to 0 or blank) generation of a unique objID for a MOS object. Set this to 1 for OpenMedia which identifies a MOS object by objID and itemID. *Optional.* Valid values: 0 or 1.
- **HideAddButton:** (Optional) Force use of only Drag&Drop functionality. When MOS-based NCSs are used, setting to 1 hides the **Add** button in the Viz Mosart ActiveX and forces the user to use Drag&Drop.
Default is 0 (false). Valid values: 0 or 1.
- **MosId:** This is the default MOS ID for the Viz Mosart ActiveX. For some NCSs like ENPS, no value is required. For other NCSs like Avid iNews or Octopus, the value must be the MOS ID as mapped in the respective NCS. See below section how to configure MOS ID in some of the NCSs.
- **MosVersion:** Forces the Viz Mosart ActiveX to use the given version of the MOS protocol. A version higher than 2.8 enables the **Add/Apply** button in the Viz Mosart ActiveX. Should be set to 2.8.3 for all MOS-based newsroom systems, such as ENPS and OpenMedia.
- **SortXpoints:** For newsroom tags with a crosspoint value, specify the ordering of these crosspoints as they appear in the drop-down menus.
By default, Viz Mosart ActiveX sorts the newsroom tags *alphabetically* (both in database and channeltemplates.xml).
When enabled (value 1), the crosspoints are sorted based on the order in the avconfig.xml when they were created. *Optional.*
Read more about newsroom tags in Additional Template Functionality, section Newsroom Tags Order .

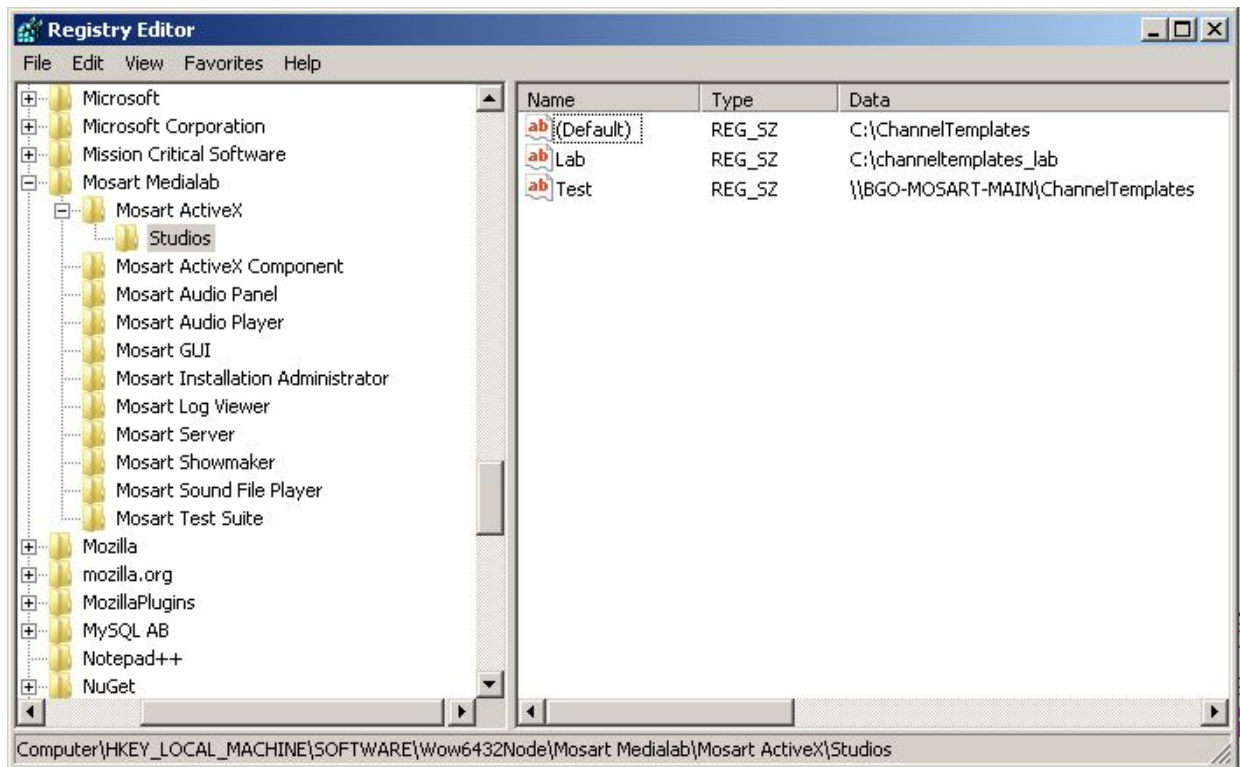


- **VerboseLogging:** Viz Mosart ActiveX logs using Microsoft trace mechanisms. The logging is not written in the Viz Mosart main logs. Use Microsoft Sysinternals DebugView to monitor the logging. When set to 0 or empty, only log events of type *Information*, *Error* and *Warning* are logged. Setting *VerboseLogging* to 1 enables more elaborate logging. *Optional*. Valid values: 0 or 1.

Studios Sub-key

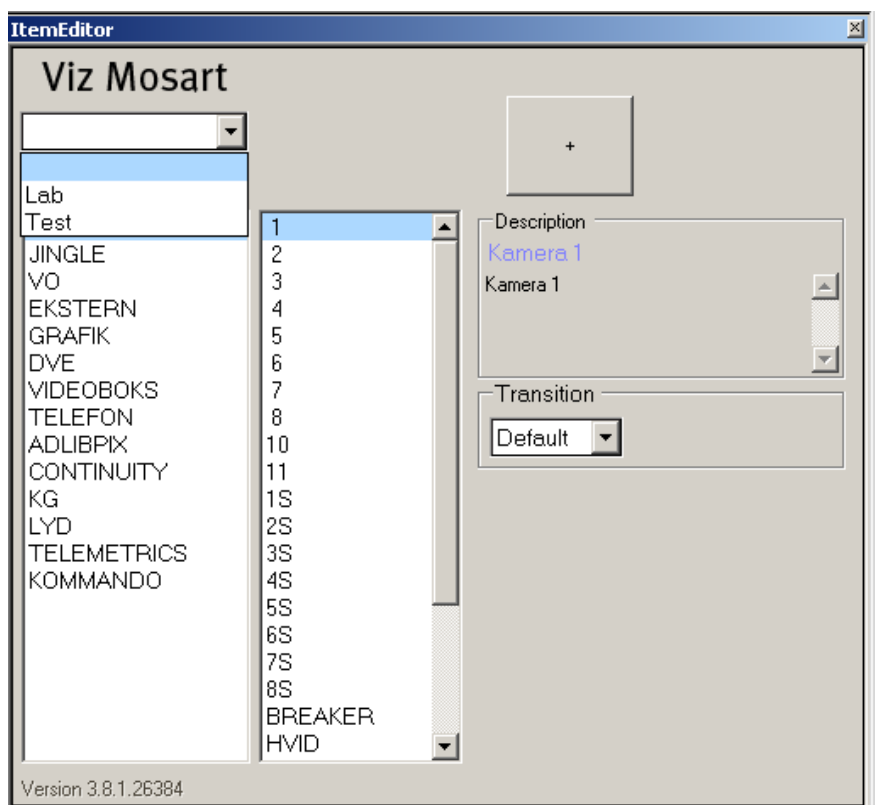
Setting this sub-key enables configuration of a system with multiple control rooms.

- Under **Mosart ActiveX** key select sub-key *Studios*.
A control room is defined by files residing in a specific folder (typically `c:/channeltemplates`) or in a database, as shown below.



The **DefaultStudio** option on the Mosart ActiveX key tells which control room to use by default or set this as blank and configure the (Default) option on the Studios key:

On the Viz Mosart ActiveX page, two drop-down boxes can be found. The first one is a list of control rooms as configured in the **Studios** sub-key from registry:



In this example, three sample control rooms are configured in **Studios** sub-key:

- blank item (default control room)
- Lab
- Test.

These items all be shown in the first drop-down box. Since DefaultStudio registry key was set to blank, Viz Mosart ActiveX opens with blank item selected in the first drop-down box and thus pointing to the control room whose configuration files can be found in the Viz Mosart folder C:\ChannelTemplates.

9.1.3 Configuring the Newsroom System

Additional settings on the NRCS side. Here we describe configuration of:

- [AP ENPS](#)
- [Avid iNEWS](#)

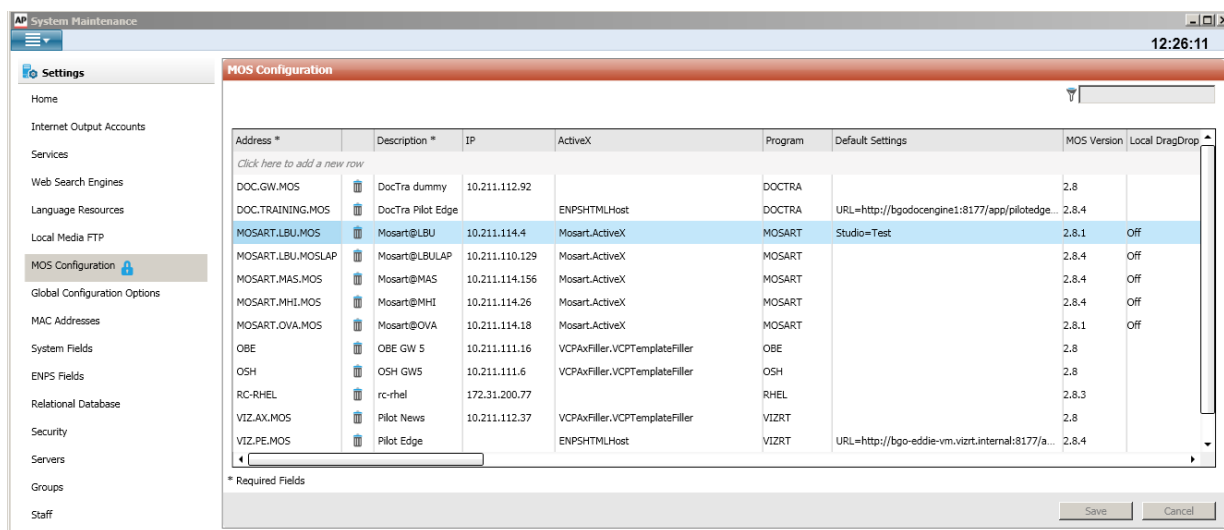
AP ENPS

Configuring ENPS for Viz Mosart ActiveX

The following configuration applies to AP ENPS version 8.0+.

- Each Viz Mosart MOS device (the MOS identification of a Viz Mosart MOS connection- see **mosID** property from [Manus Administrator Configuration/ Settings Editor - MOS](#)) needs the

following properties set in the MOS Configuration section from *ENPS System Maintenance* application to be able to use Viz Mosart ActiveX in ENPS client:



Column name	Value
Description	Gives the Mosart ActiveX name in ENPS client.
Active X	Mosart.ActiveX.
Default Settings	Blank or Studio=Studio name where “Studio name” is one of the studios defined in the Studios registry key above. If blank, the value of the DefaultStudio key attribute is used to open Viz Mosart ActiveX with the given studio name selected in the first drop-down box . If Studio option is given, Viz Mosart ActiveX opens with the given studio name selected in the first drop-down box (in our example is "Test"). Note that DefaultStudio setting from registry is ignored.
Local DragDrop	Off.

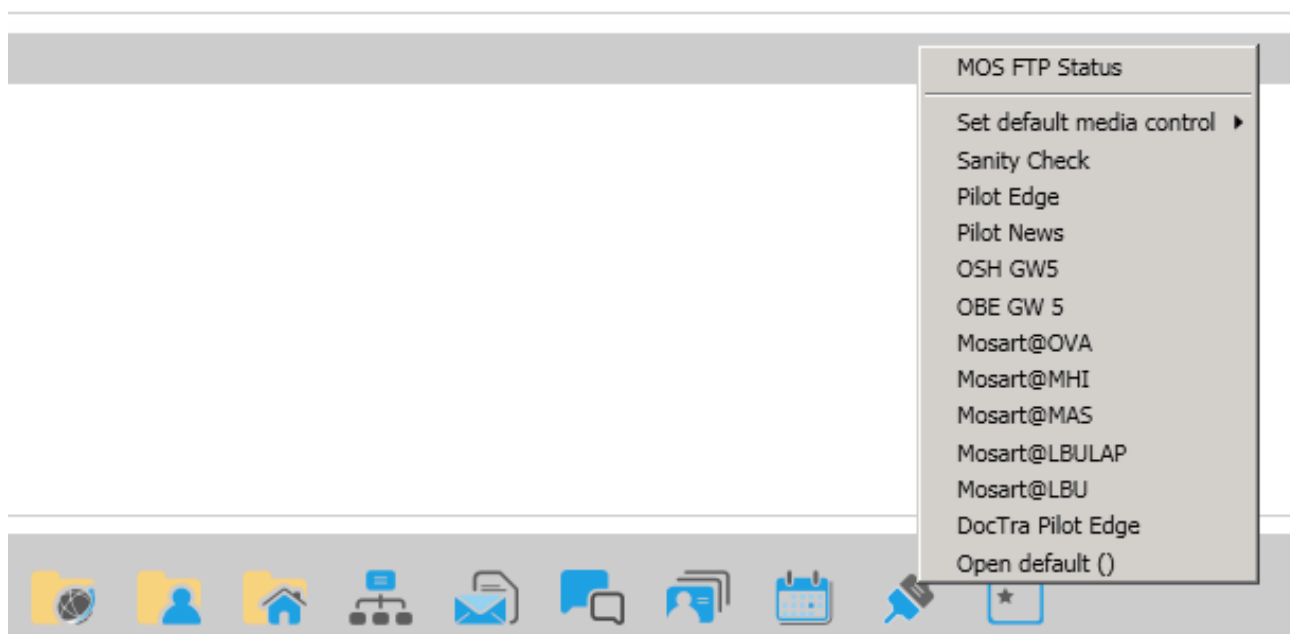
To open Mosart ActiveX from ENPS

- To open Mosart ActiveX from ENPS client:

Right-click on the MOS icon (bottom right) > **Select the Mosart ActiveX** (the name configured in MOS Configuration, in this example *Mosart@LBU*).

If several Viz Mosart MOS devices are configured, any will open Viz Mosart ActiveX.

See the *Viz Mosart User Guide*, section *Viz Mosart ActiveX* under *Mosart User Interface*, for further details of operator actions.



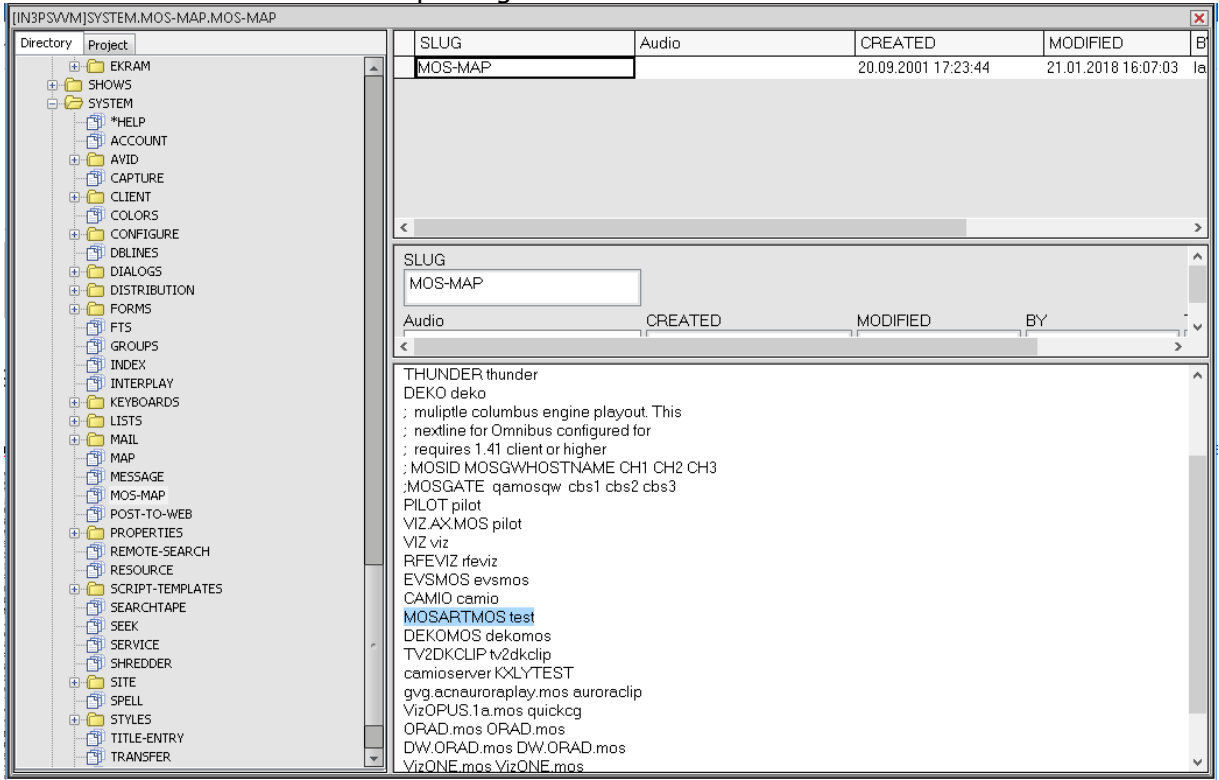
Avid iNEWS

Always consult with the iNEWS administrator before modifying these configurations.

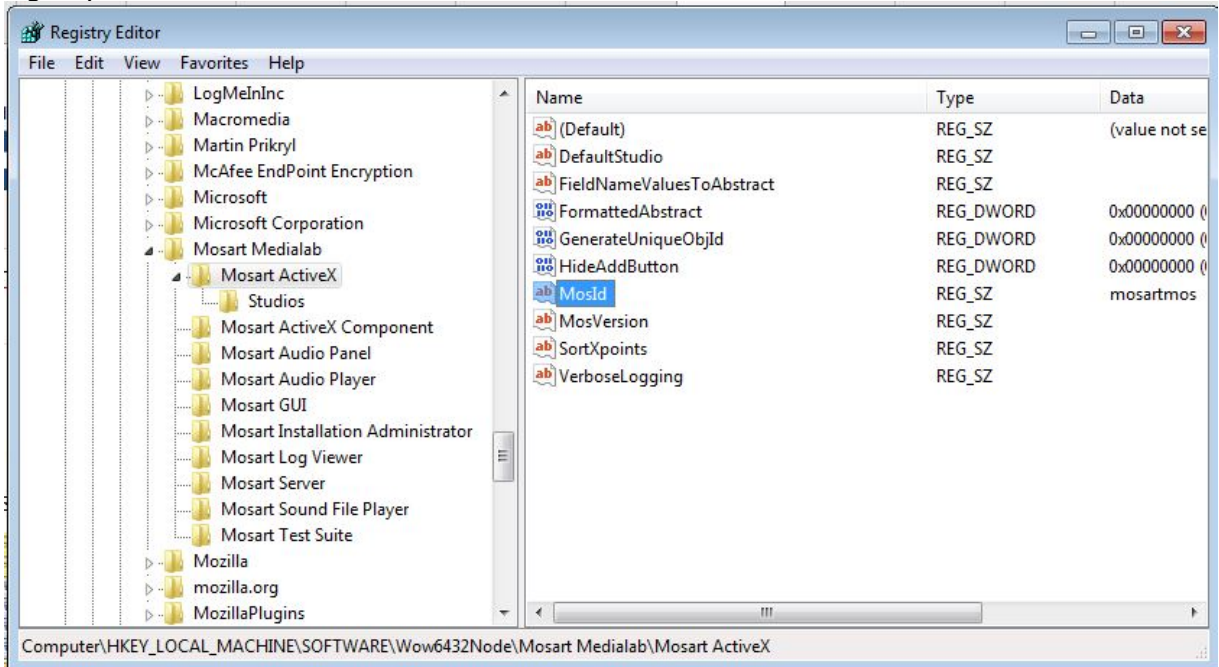
Configuring iNews for Viz Mosart ActiveX

1. Set the MosId in the registry needs to be set to the same value defined in the SYSTEM.MOS-MAP rundown which can be opened from iNEWS client as any rundown.

A user must have administrator privileges to access the SYSTEM.MOS-MAP rundown :



- The first value (MOSARTMOS) is the MOS ID that is configured in the Viz Mosart ActiveX registry. The value is case-insensitive.



3. The second value set in iNEWS SYSTEM.MOS-MAP (**test**) is used when adding the MOS object into a story script as a grommet command:

The screenshot shows the Mosart Item Editor interface. At the top, there are several input fields for metadata: PAGE, MOS ACTIVE, DIRECTTAKE, DT, Mosart Type, and Mosart Vari. Below these are fields for Audio, TAL, SLUG (containing the text 'Test'), CAM, SHOT, FORMAT, GRAPHICS, and R. Further down, there are fields for MODIFIED: (18.01.2018 15:04:00), MOD BY: (laura), EDITOR, WRITER, VID-ID, CH, STATUS, and TAPE #. The main area of the interface is a large text editor containing the text '*test CAMERA 1'. At the bottom right, there are buttons for CAP and NUM.

To open Mosart ActiveX from iNews

- To open Viz Mosart ActiveX from iNEWS client, go to **Tools > Plugins > Mosart Item Editor**.

See the *Viz Mosart User Guide*, section *Viz Mosart ActiveX* under *Mosart User Interface*, for further details of operator actions.

9.2 Connecting The ActiveX To Viz Mosart Server

There are two ways to link the ActiveX to your Viz Mosart Server installation:

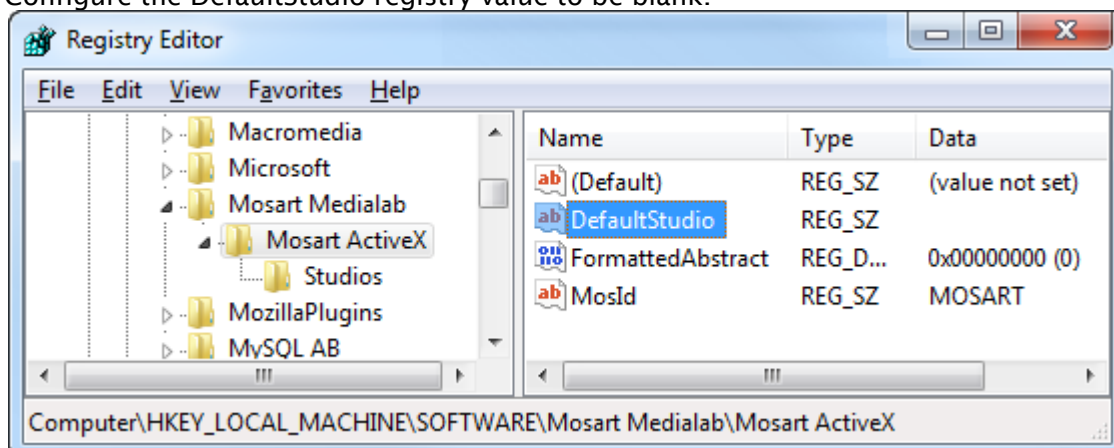
- [Connecting the ActiveX Directly to Viz Mosart Server via folder sharing](#)
- [Connecting the ActiveX to Viz Mosart Server via Template Database](#)

9.2.1 Connecting the ActiveX Directly to Viz Mosart Server via folder sharing

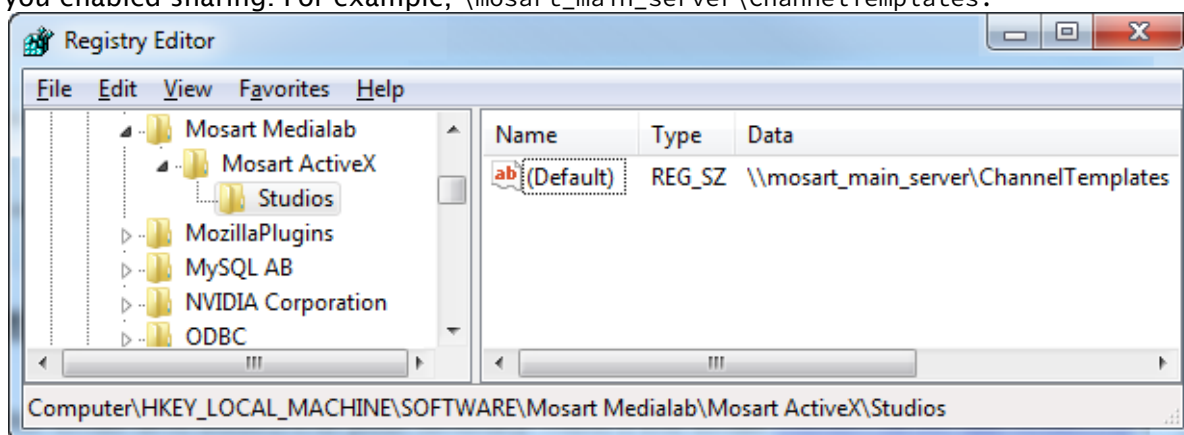
This method is for connecting the ActiveX directly to Viz Mosart Server.

1. Enable folder sharing of *C:\ChannelTemplates* on the Main Viz Mosart Server. Give read only access to this folder to the appropriate users.

2. Configure the DefaultStudio registry value to be blank.



3. Edit the (Default) entry in the Studios key and insert the path to the ChannelTemplates folder you enabled sharing. For example, `\\mosart_main_server\ChannelTemplates`.



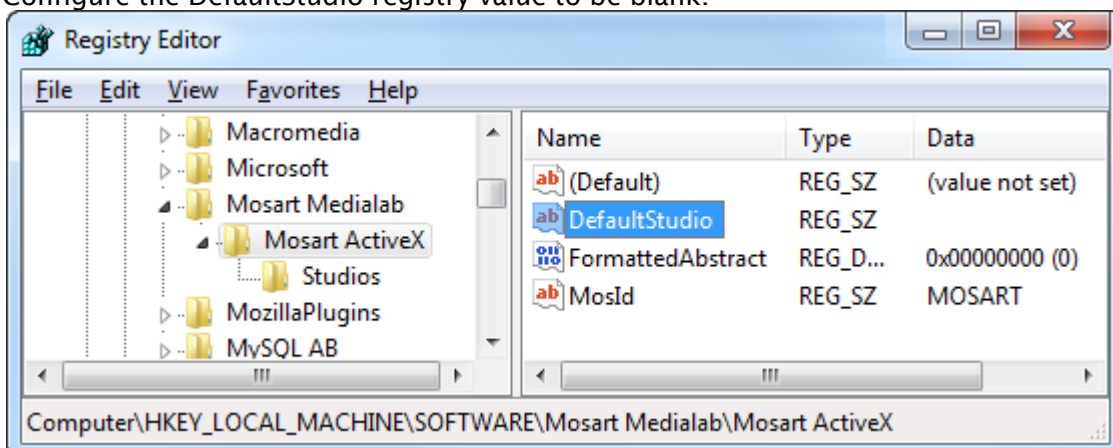
4. Export the now complete ActiveX registry entry and deploy this file using your deployment tools for other workstations that require the ActiveX.

9.2.2 Connecting the ActiveX to Viz Mosart Server via Template Database

This method is for connecting the ActiveX to Template Database.

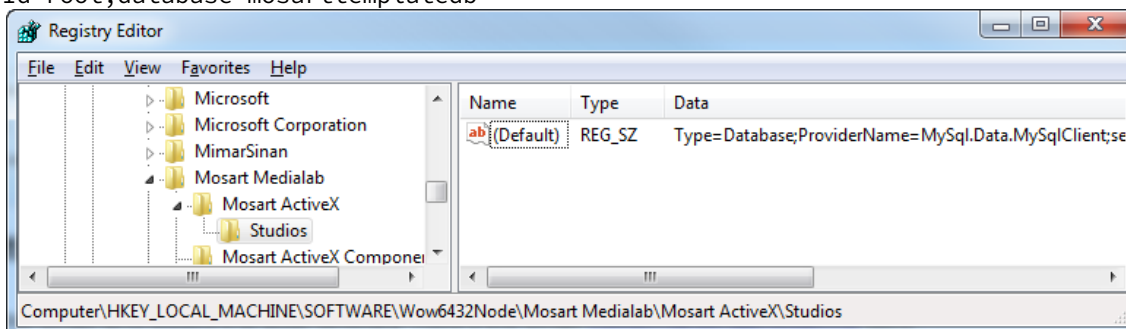
1. A read only database user is recommended for access from the ActiveX.

2. Configure the DefaultStudio registry value to be blank.



3. Edit the (Default) entry in the Studios key and insert the connection string for Template Database.

- Type=Database;ProviderName=MySql.Data.MySqlClient;server=localhost;User Id=root;database=mosarttemplatedb



The connection string of the database contains the following fields:

Name	Value
Type	Database.
ProviderName	The qualified name for the .NET Component used for database access. Currently on MySql.Data.MySqlClient is provided in the installer.
Server	IP-address or hostname of the database server.
User Id	User ID of the database
Password	Optional. The password for the User ID to the database.

Name	Value
Database	The SQL database name on the database server.
Gallery	Optional. Name of gallery to be addressed within the database. Also extracts private galleries for that gallery.

ADO.Net Driver

For the Active X component to connect to a template database, an ADO.NET driver is needed to be installed on each client. The driver can be downloaded from <https://dev.mysql.com/downloads/connector/net/>.

Caching ChannelTemplates files

The Viz Mosart ActiveX stores a cached version of the ChannelTemplate files (*channeltemplates.xml*, *newsroomsettings.xml* and *avconfig.xml*) in the system's TEMP folder: *%localappdata%/Temp/Mosart Medialab/ChannelTemplates* (Note the space character!). This is done for both type of connections described above.

Changes are only downloaded if the timestamps on these files differ.

Deleting the cache folder forces Viz Mosart ActiveX to get the latest configuration files from database or from the shared folder.

9.3 ActiveX Notes

ActiveX stores a cached version of ChannelTemplates in the system *%Temp* folder. Changes are only copied if the cached ChannelTemplates time stamp differs to that of the Viz Mosart Server or Template Database.

A force refresh of ActiveX can be done by opening and closing the ActiveX twice within the host application.

10 AV Automation

AV Automation (*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 Mosart perspective, are treated as clips.

10.1 Overview

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.


All device drivers are stored in AV Automation, offering the administrator a single point of configuration.

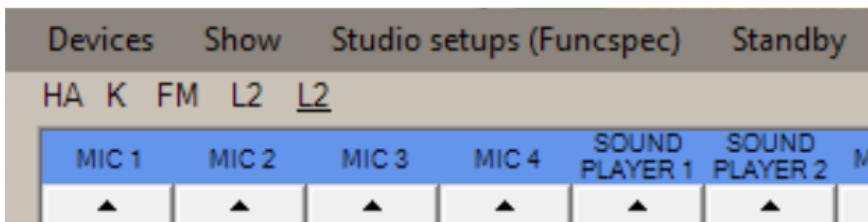
On the fly commands can then be issued directly to each device through the Viz Mosart GUI, or as pre-programmed operations driven by the rundown submitted through the Newsroom System.

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

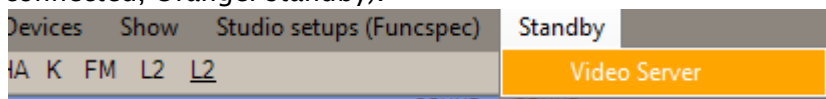
AV Automation is the device handler of Viz Mosart. This application communicates with all connected devices such as cameras, video servers and vision mixers. As well as presenting tools for configuring these devices, it includes [Template Editor](#), for defining and customizing templates.

10.2 Menu Description

 Viz Mosart - AVAutomation: NameOfGallery (Funcspec)



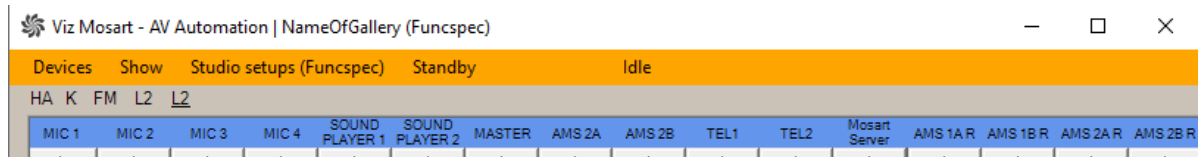
- **Devices:**
 - **Preferences:** Opens [AV Automation Device Properties](#).
 - **Template editor:** Opens [Template Editor](#).
 - **A/V Setup:** Opens [Audio and Video Setup](#).
 - **Record:** See [Recording](#).
 - **Change password:** Creates or changes the [Other Template Functionality](#).
- **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**.
- **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.



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

10.3 Logging And Status

Below the fader and switch representations are three monitoring panels.

10.3.1 Logging Panel

To the left is a Logging panel with timestamped event details. Clicking in this panel toggles between three display modes.

- Basic (default) - log window without color coding
- Color - log window with color coding
- Disabled - log window will not show anything

This section presents the following AV Automation topics:

- [Recording](#)
- [Audio and Video Setup](#)
- [Template Editor](#)
- [Mosart Instrumentation Panel](#)
- [Newsroom Tags](#)

See Also

- [AV Automation Device Properties](#)

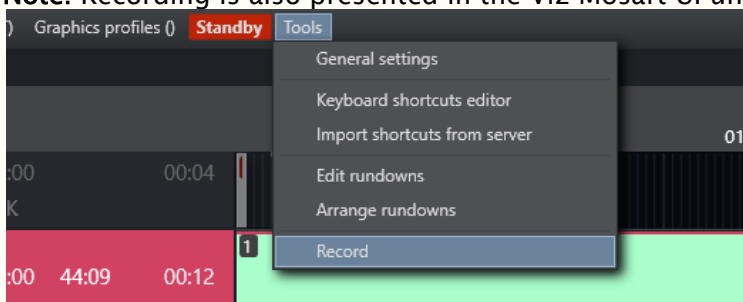
10.4 Recording

Viz Mosart can instruct a connected video server to initiate recording.

The video server must first be configured, as explained in [To add a recording port in AV Automation Devices Video Servers](#), [To Add Recording Ports](#).

- The configured video servers are displayed in AV Automation under **Devices > Record**.

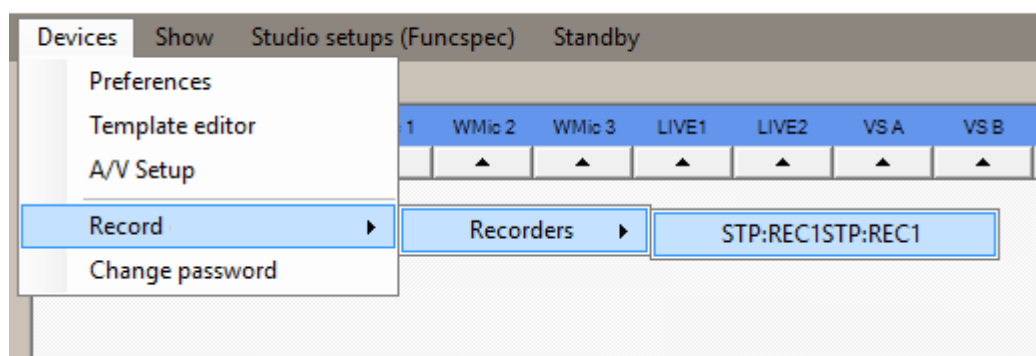
Note: Recording is also presented in the Viz Mosart UI under **Tools > Record**



10.4.1 To Start a Recording

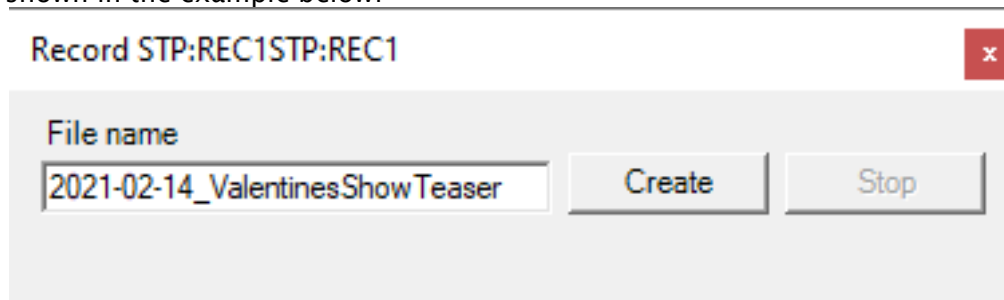
1. In AV Automation, select **Devices > Record > Recorders > [Name of recording port]**

Viz Mosart - AVAutomation: NameOfGallery (Funcspec)

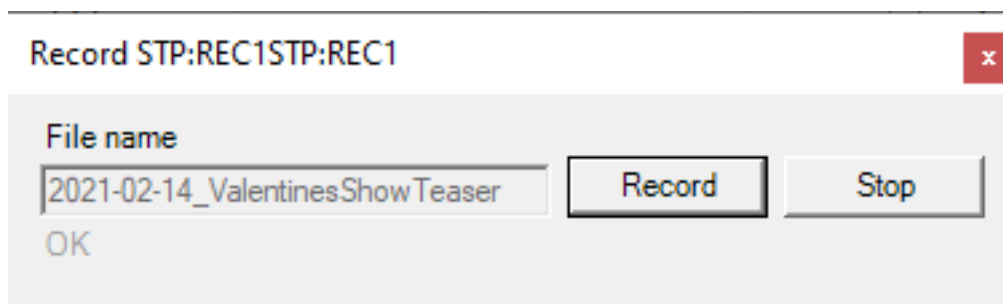


Note: If several recording ports have been configured, the Recorders menu will have several menu items.

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**.



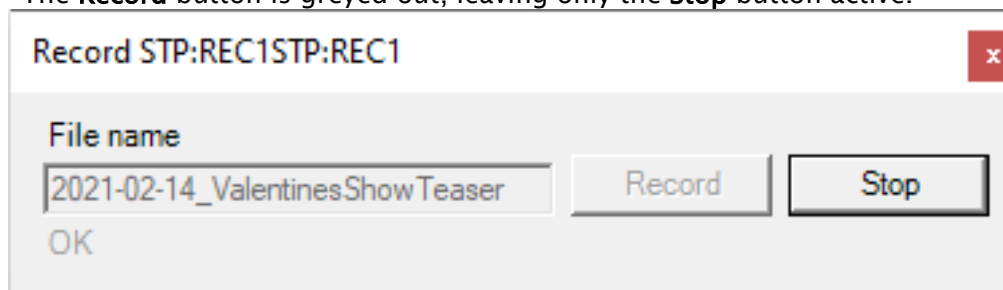
- The commands that are sent in the background to the video server, are presented in AV Automation's Information panel.

17:59:14.96	RECORD(CreateClip) failed: 2021-02-14_ValentinesShowTeaser@STP:REC1 (8...
17:59:15.00	Executing cmd RECORD Prepare Rec1 2021-02-14_ValentinesShowTeaser

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.

18:00:38.18	RECORD(RecordChannel) failed: @STP:REC1 (2 ms): 403 Unit/port STP:REC1...
18:00:38.18	Executing cmd RECORD Start Rec1 2021-02-14_ValentinesShowTeaser

Recordings are stored on the video server's system. Exactly where, is dependent on the video server type.

10.4.2 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.

18:04:07.02	Executing cmd RECORD Stop Rec1 2021-02-14ValentinesShowTeaser
-------------	---

10.4.3 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
AirSpeed MultiStream	2018-05

10.5 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. A simple XML editor utility is used to configure the settings in AV Automation.

- To open the configurations editor, from AVAutomation, 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](#)
- [Audio Configurations](#)
- [Vision Mixer Effects Configurations](#)
 - [Effect Number and Studio Setup](#)
 - [EMEM](#)
 - [Settings](#)
- [Router Source Configurations](#)
- [Router Destination Configurations](#)

10.5.1 Video Configurations

To edit Video Crosspoint Configuration Settings, open the [Audio and Video Setup](#) XML Editor and select **Edit > Video Config**.

	X-point name	Physical input	Audiolink	Audiolink Ch 2	Input type	Hide From User
▶	LIVE1		EXT1		camera	<input type="checkbox"/>
	LIVE2		EXT2		camera	<input type="checkbox"/>
	LIVE3				camera	<input type="checkbox"/>
	LIVE4				camera	<input type="checkbox"/>
	CLIP1				videosever	<input type="checkbox"/>

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

- **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 videosever 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.
 - **videosever:** 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.

10.5.2 Audio Configurations

To edit Audio Mixer configuration settings, open the [Audio and Video Setup XML editor](#) and select **Edit > Audio Config**.

Channel Name	Physical input	Mixer name	Input type	Initial level	Initial standby	Delay address	Loudness channel
MIC 1	1		mic	-90	<input type="checkbox"/>		
MIC 2	2		mic		<input type="checkbox"/>		
EXT1	21		external_sou...		<input type="checkbox"/>		
EXT2	23		external_sou...		<input type="checkbox"/>		
TLF	7		tlf		<input type="checkbox"/>		

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

- **Channel Name:** Name to use for the audio fader in the Viz Mosart system.
- **Physical input:** For audio mixers that use fader numbers or MIDI assigned faders, this value 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.
- **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 audio mixer boards only) Gives the signal address for the delay function.
- **Loudness channel:** Defines the channel number for loudness level control.

10.5.3 Vision Mixer Effects Configurations

Video mixer effects can also be predefined so that the user does not have to recreate the effect in every template that refers to it. The **Effects** table is referenced during the execution of a template that has specified an **Effect name** or **Effect Nr**.

To edit Vision Mixer Effects configurations, open the [Audio and Video Setup XML editor](#) and select **Edit > Effects**.

Effect name	Effect Nr	Emem Nr Preview	Emem Nr Program	Emem Nr Cleanup	Duration	RCC	Next cue delay	Fader delay	Audio file	GPO Preview	G P
SP WIPE ...	1		1		10	-1	20				
B mix dela...	2		2		10	-1	20				

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.

Effect name	Effect Nr	Studio Setup
Non Additive Fade	33	T3
Edge LtoR(H)	1	T2
Curl	1	T2
Flash	3	
Circle	4	

The pair of values "Effect Nr" and "Studio Setup" is already in use. See section "Audio and Video Setup" of the documentation.

EMEM

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.
- **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.

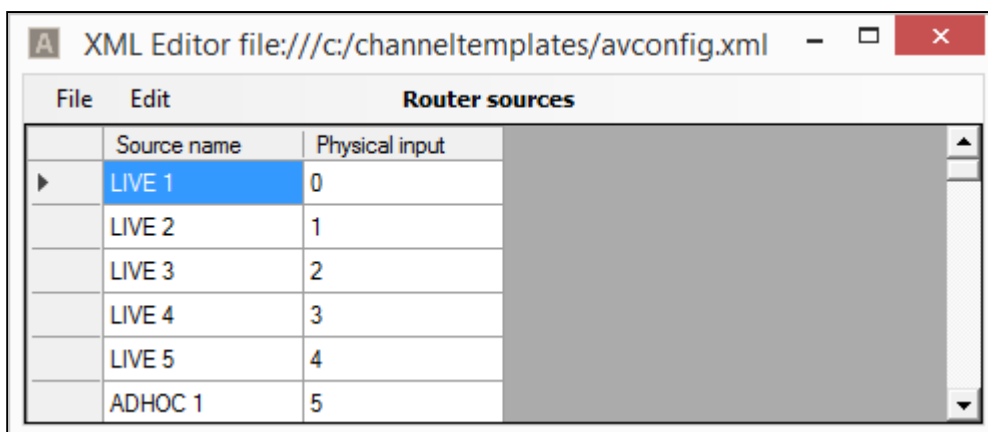
⚠ Macro Preview Action, Macro Program Action: *Load Project* could affect the performance of the switcher and should thus be used with caution during a running show.

- **Macro Program:** Refer to the **Text field** details in **Recall program** in the [Template Device Functions - Macro Recall](#) section.

10.5.4 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, open the [Audio and Video Setup](#) XML editor and select **Edit > Router sources**.



	Source name	Physical input
▶	LIVE 1	0
	LIVE 2	1
	LIVE 3	2
	LIVE 4	3
	LIVE 5	4
	ADHOC 1	5

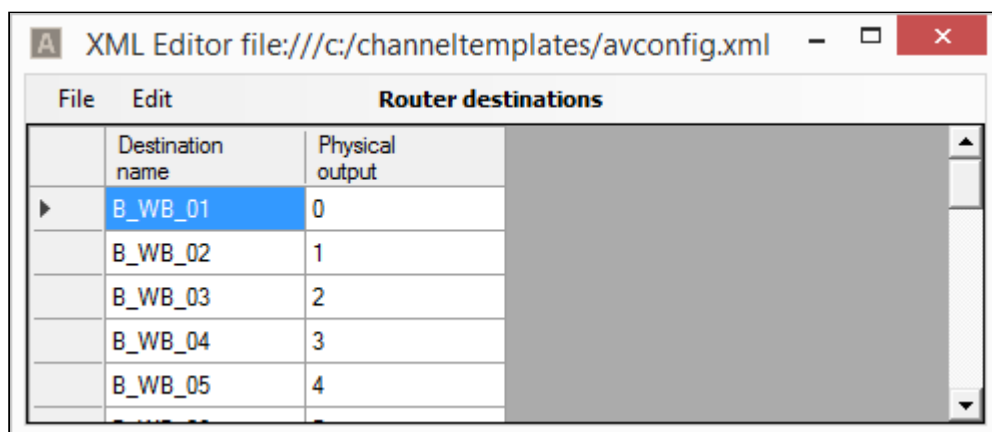
Router switches are predefined as part of a Viz Mosart template. Keyboard shortcuts can also be programmed for a router switch to take place.

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

10.5.5 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, open the [Audio and Video Setup](#) XML editor and select **Edit > Router destinations**.



The screenshot shows a window titled 'XML Editor file:///c:/channeltemplates/avconfig.xml'. Inside, there is a table with the following data:

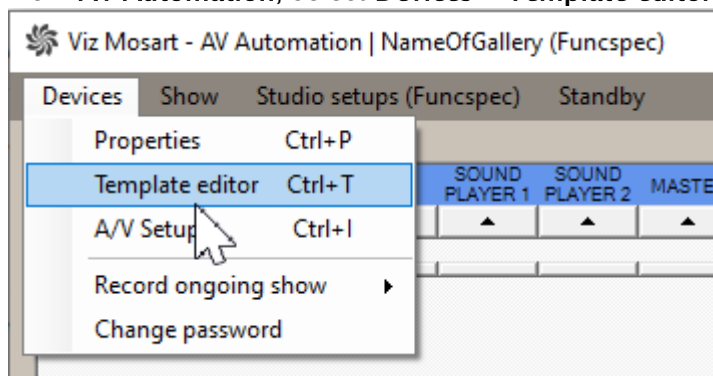
	Destination name	Physical output
▶	B_WB_01	0
	B_WB_02	1
	B_WB_03	2
	B_WB_04	3
	B_WB_05	4

- **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

10.6 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 window is displayed:



See also: [Template Device Functions](#).

Several detailed template topics are presented in this section:

- [Working with Templates](#)
- [Template Examples](#)
- [Template Device Functions](#)
- [Working with Template Sets](#)
- [Other Template Functionality](#)

10.6.1 Working with Templates

This section contains the following topics:

- [Template Operations](#)
- [Template Properties](#)
- [Control Commands in Templates](#)
- [Destination Panel](#)
- [File-based Template Sets](#)

See also: [Linking Device Properties and Newsroom Tags](#).

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

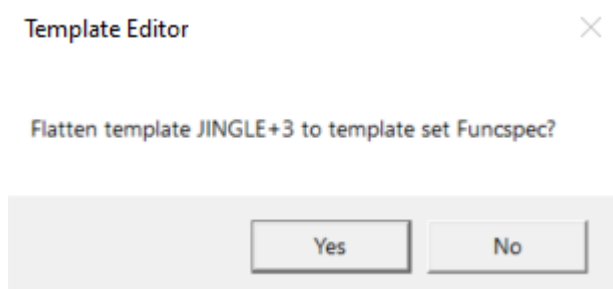
1. Select **File > Remove template**.
2. Confirm delete in the dialog box.

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**.
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.

Template Properties

This section explains the configurable fields on the **General** tab of the **Template Properties** menu, clarifying the function of a *Sequence*.

- [Configuration Descriptions](#)
- [Sequences](#)
 - [Recalling a Sequence from Template](#)
 - [Recall a Sequence from Directtake Template](#)
 - [Controlling a Sequence](#)
 - [Template Database Sequences](#)

Template properties

General | Control Commands | Newsroom tags order | Destination panel

Name: [] Type: BREAK Variant: START Description: [] Recall Nr (Direct take): [] Disable in Rehearsal Mode

Default NCS variant Use state variants

Send to NCS

Preload Pretake []

Take at planned story in-time

Fixed Duration [00:00:00]

Story Duration

Autotake next OffSet [0]

Default TC in [00:00:00]

Default TC out [00:00:00]

Sub Sequence

Sequence [] Loop

Reset State [2] Reset State After [2] Stories

Last state [] Single State []

Variants: ENDE, START, START VERTICAL


up Down

Hide from user

Cancel OK

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 <i>Viz Mosart User Guide</i> , section <i>Viz Mosart User Interface</i> under <i>Rundown Window</i> .
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.

Item	Description
Recall Nr (Direct take)	<p>(Applies only for templates in the Direct takes template set). Values from 0-999 apply, and are the value recalled from the numeric keypad in the Viz Mosart client application.</p> <div style="border: 1px solid #f9c77f; padding: 10px; margin-top: 10px;"> <p> Note: Avoid starting numbers with zero (0). For example, use 56 instead of 056.</p> </div>
Disable in Rehearsal mode	(Applies only for templates in the Direct takes template set). The DirectTake will not be executed when in Rehearsal Mode
Default TC in	(Applies only for templates in the Direct takes template set).
Default TC out	(Applies only for templates in the Direct takes template set).
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	<p>When selected, enable the preview functionality for the Accessory template. The Accessory template must have time code 00:00 for this to function. <i>Only valid for Accessory templates.</i></p>
Take at planned story in-time	<p>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.</p>
Fixed Duration	<p>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.</p>

Item	Description
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 <i>primary</i> 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 OLD and invisible Working with Templates KEPT BECAUSE I AM FULL OF INLINE COMMENTS for explanation of state variants.
Control Commands (tab)	If the template should have <i>control commands</i> attached, configure here. Please refer to Control Commands in Templates for further instructions.

Sequences


A sequence is a saved story (including primary and secondary items with relative timing information).

Sequences are created in Viz Mosart GUI, cp. Viz Mosart User Guide, Operation, 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 directtake template, all sequences, regardless of template set, are eligible, see [Recall a Sequence from Directtake Template](#)


3. If required, select **Loop**.

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 Directtake Template

To recall a sequence from Directtake template

- Sequences may be recalled from directtake templates just as from other templates. However, when taking a directtake template, a sequence will be recalled only if the template set selected *at that time* has a sequence with the name chosen. (as this sequence will be recalled).

 Any take out logic might not work as expected if the directtake is not fired at the beginning of the current main template. When using directtakes 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.

Template Database Sequences

 A template database workflow does *not* support sequences.

Control Commands in Templates

A Mosart control command can be triggered from a template, executed when the template is Taken, at Continue points or when the template is Taken out.

- [Working with Control Commands](#)
 - [Controlling Video Servers](#)
 - [Adding a Control Command](#)
 - [Command On Take](#)
 - [Continue Points](#)

- [Command on Take Out](#)
- [Command Values and Parameters](#)
 - [Parameters with Placeholders](#)



- A *DirectTake* template is a special Mosart template designed for immediate execution, and is not normally part of a rundown.
- A Mosart template can trigger *multiple* control commands.
- A (not-templated) control command can trigger a Mosart template, often used for a *DirectTake* template.

Working with Control Commands

Controlling Video Servers

This is done via the Control Command page within the **Template properties**.

Template properties

General | **Control Commands** | Newsroom tags order | Destination panel

Command On Take

Control commands Take

	Command	Value	Parameter
▶	VIDEO_PORT	PLAY_PAUSE	D
*			

Continue Points

Continue Points

	Autotake	Command	Value	Parameter	Description
*					

Command On Take Out

Control command Take Out

Hide from user

Cancel OK

The figure above shows a *single* control command to be executed when the template is taken. Multiple control commands can be assigned here.

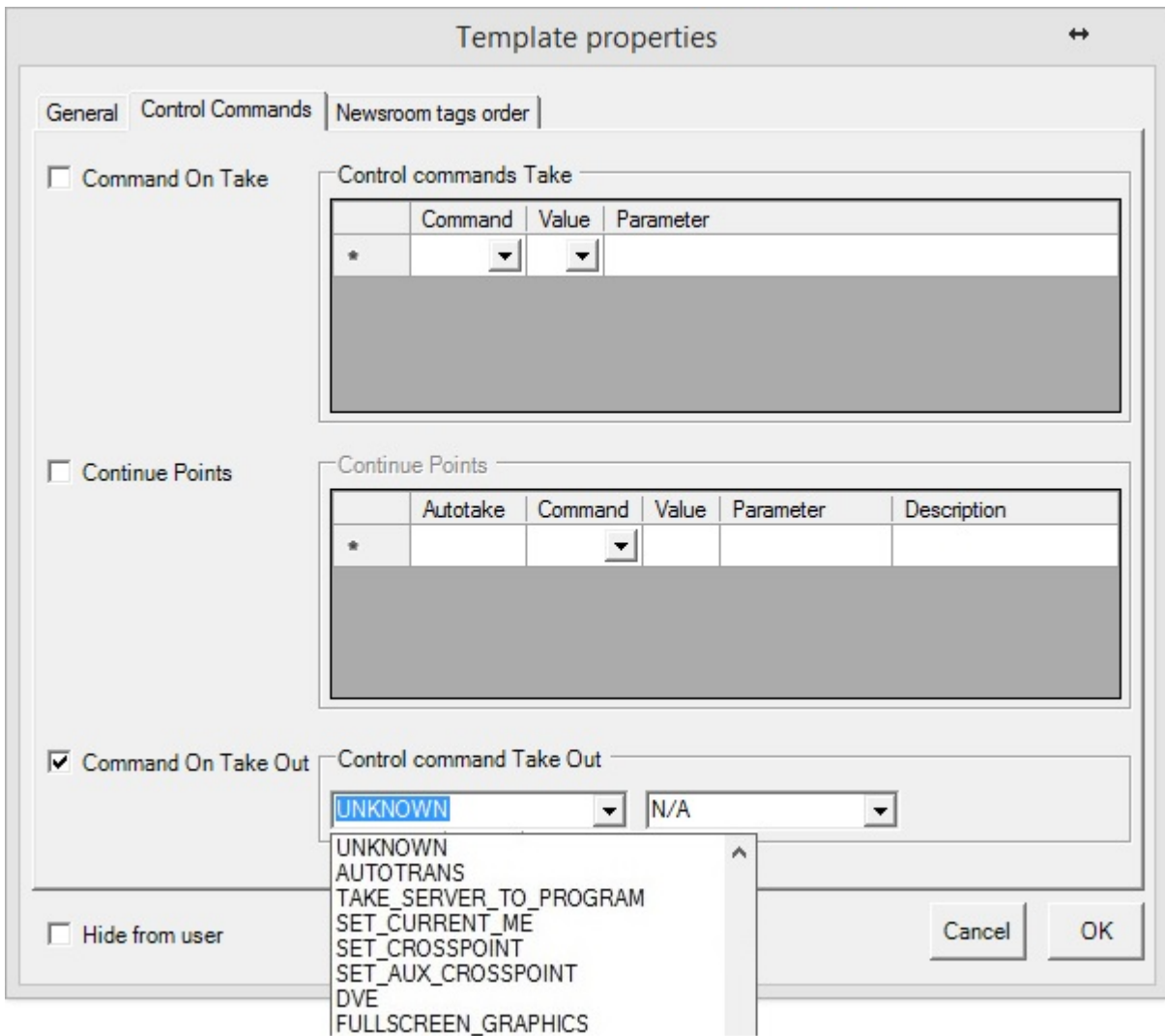
It is also possible to assign control commands to be executed at continue points and when the template is taken out.

The latter has limitation since the Parameter attribute is not available. In this context it may be useful to trigger a *DirectTake* 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 [Command On Take](#), [Continue Points](#) and [Command on Take Out](#).
- Refer to the table [Command Values and Parameters](#) for command details.



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 setting a *Newsroom* tag as a parameter in the **OnTake**-commands, the user can set the parameter from the NCS. When the user is using the **ActiveX** to add a template to the rundown, a text box is visible, requesting this value.

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 control command *continue-point* is then automatically taken after that time.


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 not 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
UNKNOWN	N/A		Default fallback. Does nothing.
AUTOTRANS	PP, ME1, ME2, ME3, ME4	MixEffect (MIX OR WIPE) + Transitionrate	For example: MIX 33, WIPE 10.  Note: The <i>Effect</i> cannot be used.
	AUX, Default	(*)	

COMMAND	Value	Parameter	Description
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.
SET_CURRENT_ME	PP, ME1, ME2, ME3, ME4 AUX Default	(*)	
SET_CROSSPOINT	N/A	(*)	
SET_AUX_CROSSPOINT	N/A	(*)	
DVE	FORWARD REVERSE	Recall forward/reverse mem	n/a
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: <i>AUTO+1</i> : The number of continue counts from the graphics + 1 <i>AUTO-2</i> : The number of continue counts from the graphics - 2

COMMAND	Value	Parameter	Description
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
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 graphics
	MACRO	Engine no : macro	For example: 4:macrohere
	PRETAKE_NEXT_OVERLAY	Render (engine no)	

COMMAND	Value	Parameter	Description
	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.)	For more details, see Control Command Key OVERLAY_TO_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'.
WEATHER	N/A	(*)	
AUDIO	FADE_MANUAL		Toggle fade manual
	FADE_OUT_KEEPS, FADE_DOWN_SECONDARY_AUDIO, FADE_UP_SECONDARY_AUDIO	Fader rate	
	SET_LEVEL_2_ONAIR, SET_LEVEL_2_PREVIEW, FREEZE_AUDIO	-	

COMMAND	Value	Parameter	Description
DIRECTTAKE	#	-	Uses the value selected in the drop-down
LIGHT	N/A	(*)	
AUTOTAKE	N/A	(*)	
PLAY_STORY	N/A	(*)	
STUDIOSETTUP	#	-	
VIDEOWALLMODE	N/A	(*)	
GRAPHICSPROFILE	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
MARKER	N/A	(*)	
VIDEO_SERVER_GOTO	N/A	(*)	
TRANSITION_TYPE	CUT, MIX	Rate	
	EFFECT	Effect no	

COMMAND	Value	Parameter	Description
	TOGGLE	(*)	
RECORD	PREPARE, START, STOP	Split parameters using, - first: clipname, - second: recorder (optional), - third: port name (default <i>Rec</i>), - fourth: group name (default <i>Rec</i>)	For example: clip33,recorder1,Rec,Rec

COMMAND	Value	Parameter	Description						
DEVICE_PROPERTY	CAMERA CONTROL	See next column.	<p>Note: This is for <i>camera robotics</i>.</p> <p>The 'DEVICE_PROPERTY' control command with value 'CAMERA CONTROL' is only used as a way to set speed (Camerobot) and presenter (Camerobot/FxMotion) from a template.</p> <p>Details for Camerobot Presenter:</p> <p>ANCHOR=pos1:presenter1[, pos2:presenter2]...</p> <p>The position and presenter is given as a pair divided by ':'.</p> <p>Multiple position-presenter pairs are possible, separated by ','.</p> <p>Examples:</p> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>ANCHOR=2:David</td> <td>position1: 2, presenter1: David</td> </tr> <tr> <td>ANCHOR=2:David, 3:Mark</td> <td>position1: 2, presenter1: David, position2: 3, presenter2: Mark</td> </tr> </tbody> </table>	Parameter	Meaning	ANCHOR=2:David	position1: 2, presenter1: David	ANCHOR=2:David, 3:Mark	position1: 2, presenter1: David, position2: 3, presenter2: Mark
Parameter	Meaning								
ANCHOR=2:David	position1: 2, presenter1: David								
ANCHOR=2:David, 3:Mark	position1: 2, presenter1: David, position2: 3, presenter2: Mark								

COMMAND	Value	Parameter	Description
	AUDIO	demands <fields> structure as parameter with parameter1=AUDIO, parameter2=key, parameter3=value OR connectionstring style	Note: This is for an <i>audio mixer</i> .
CROSSOVERCOMMAND	N/A	(*)	
ACCESSORIES	TAKE_NEXT	(*)	
SET_VIDEOSERVER_SALVO	N/A	(*)	Not implemented for templates
SWITCH_VIDEOSERVER_MIRRORING	N/A	(*)	Not implemented for templates
SWITCH_GRAPHICS_MIRRORING	TOGGLE, ACTIVATE, DE-ACTIVATE	-	
ENABLE_GRAPHICS_MIRRORING	N/A	(*)	
HOLD_VIDEO_TRANSITION	N/A	-	
HOLD_AUDIO_TRANSITION	N/A	-	

COMMAND	Value	Parameter	Description
NCS	START_STATUS, STOP_STATUS	RUNDOWN STORY ITEM	Specific for Open Media
TAKE_CONTINUE_POINT	N/A	(*)	
RUNDOWN_NCS_RESYNC	N/A	(*)	
RELEASE_BACKGROUND	N/A	(*)	
VIDEO_PORT	N/A	Must be a comma separated list in the format: command,port,parameter	For example: PLAY_PAUSE,A,PLAY. Refer to the section Video Port Control Commands in the <i>Viz Mosart User Guide</i> .

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 NCS, 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 NCS.

In NCS, 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 NCS):

```

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
    
```

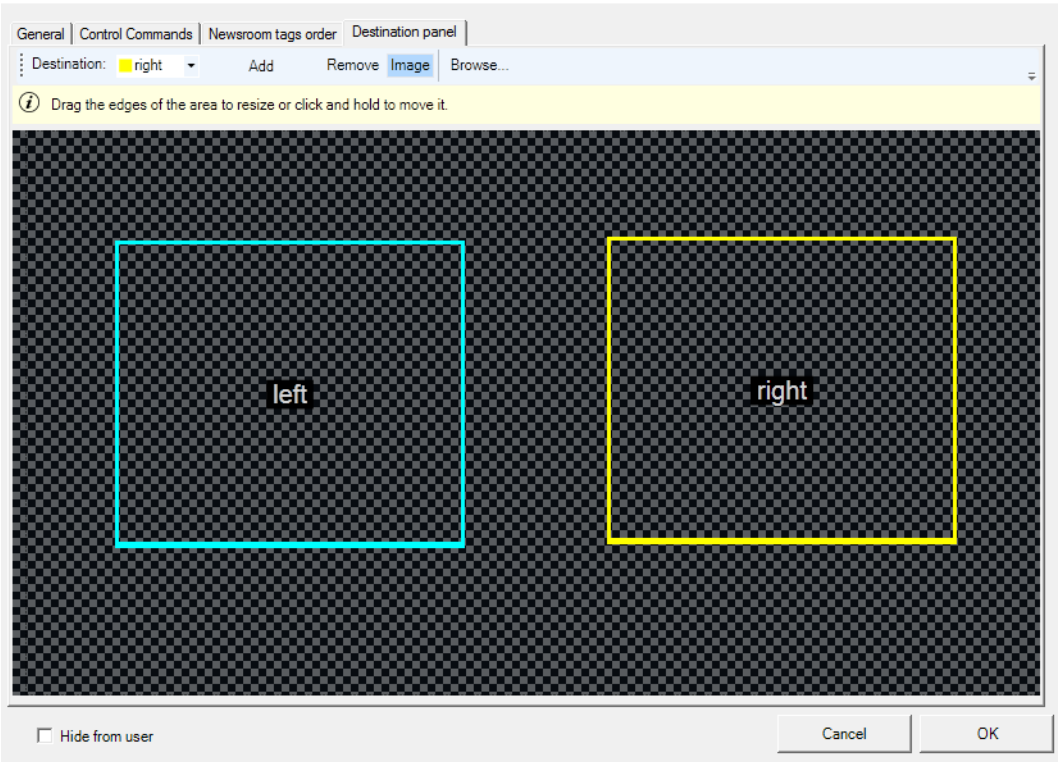
When the template is executed, a clip with name *Clip1_Test* is prepared for recording.

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.

Template properties



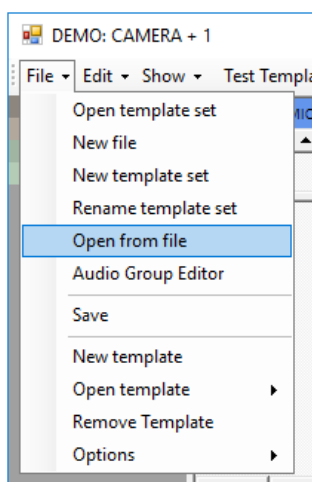
See also Program Window.

File-based Template Sets

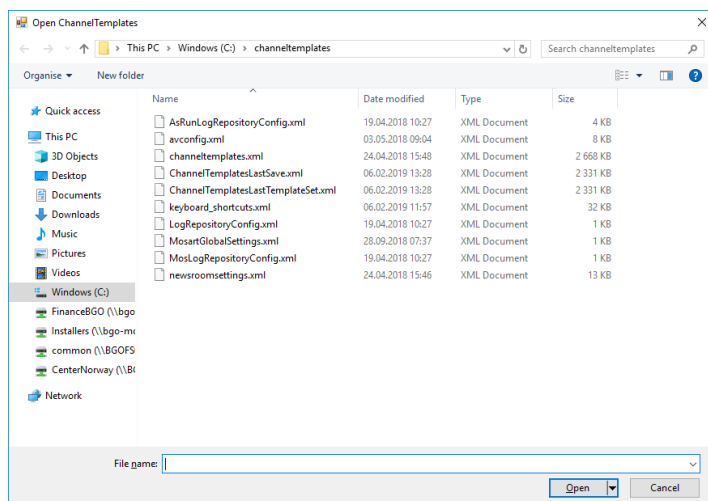
This method can be used *without* the Template Database, for example at installation / show design when trying out several solutions/designs.

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 from File**



2. Select the preferred XML-file from the list, and click **Open**.



10.6.2 Template Examples

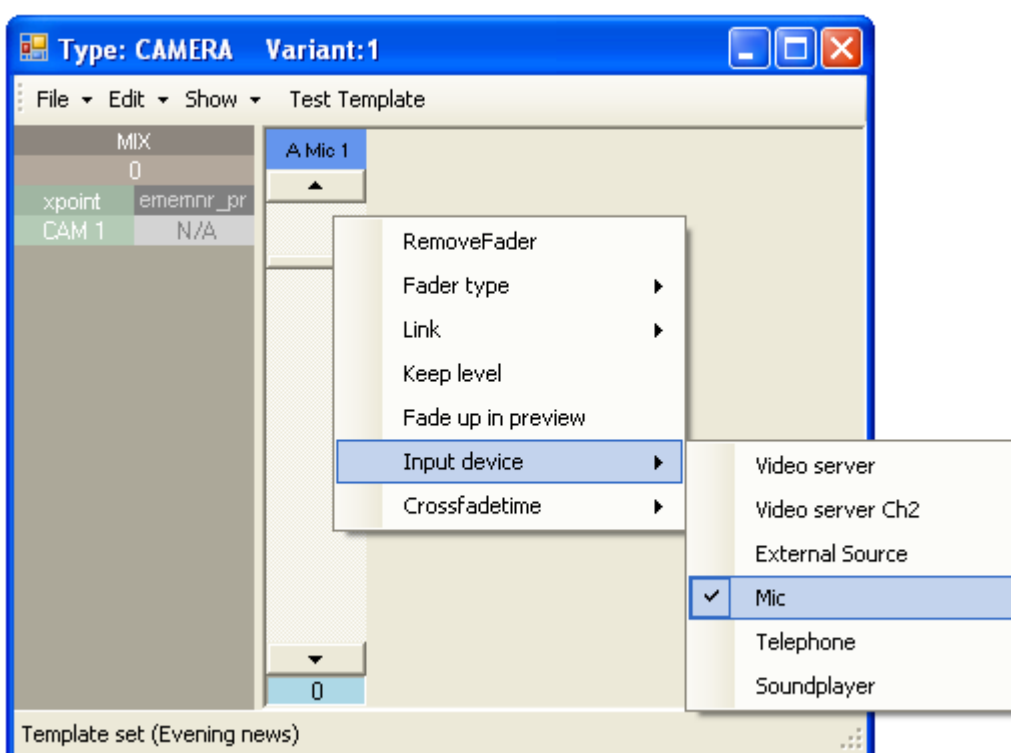
This section contains the following examples:

- [Studio](#)
- [Video Clip with Full Sound](#)
- [Voice-over](#)

- [Live External Source](#)
- [DVE](#)
- [Full Screen Graphics](#)
- [Accessory Templates](#)

Studio

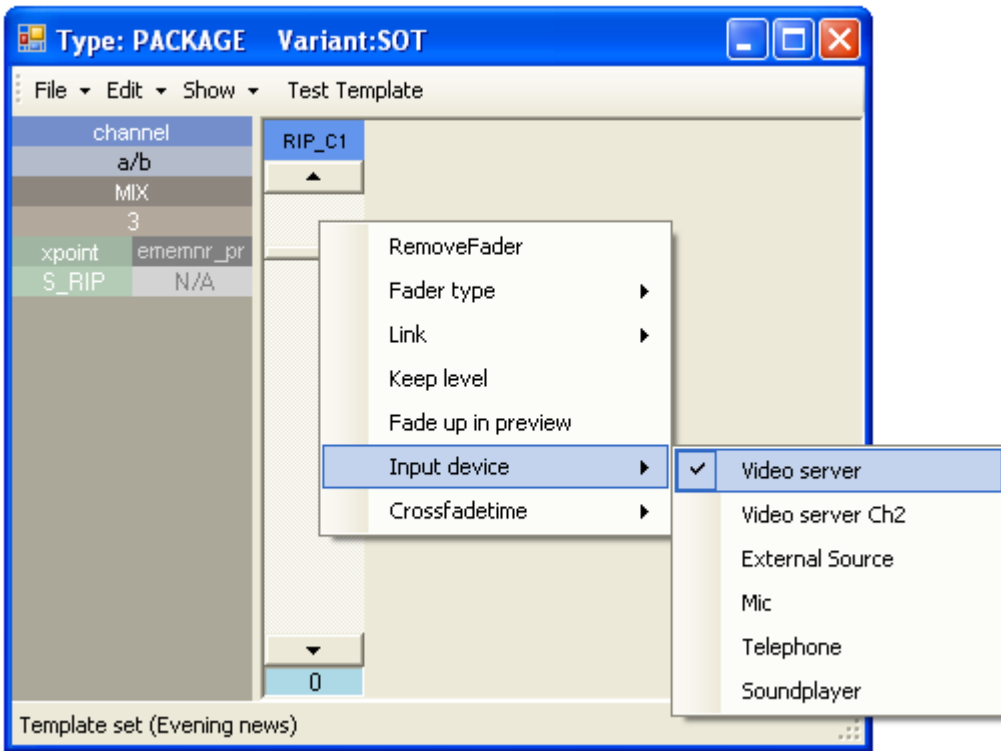
The template for a typical studio camera is based on the CAMERA type. The example below defines a standard camera 1 variant. An entry named CAM 1 in the video setup is selected at the PP bus cross point. Transition duration is 0 frames (hard cut) and E-mem recall is disabled. There is only one studio 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.



For our standard camera 2 variant, the only difference in the above setup is the PP bus cross point which is called CAM 2.

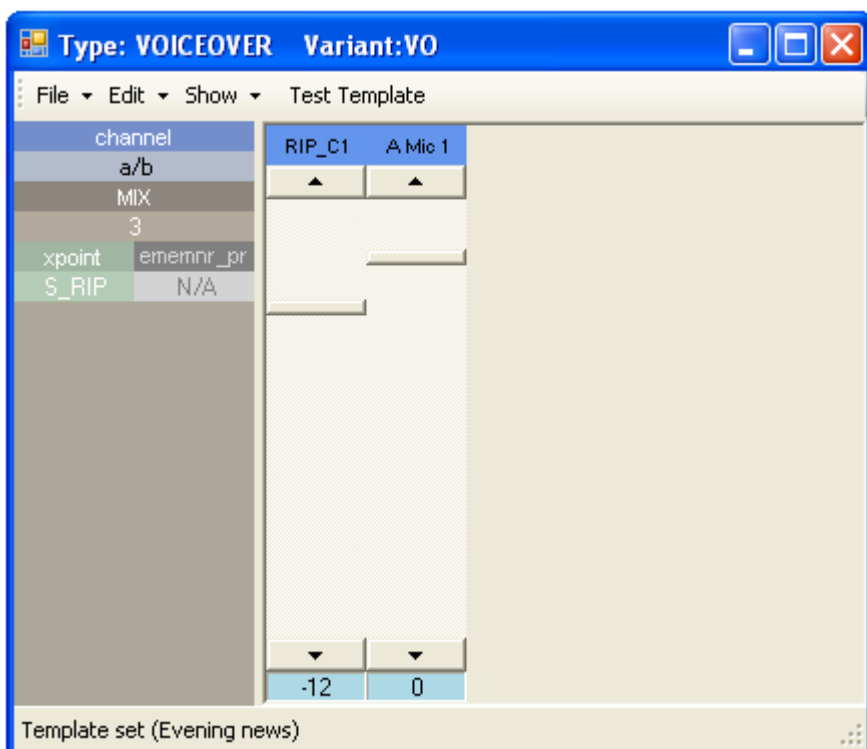
Video Clip with Full Sound

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.



Voice-over

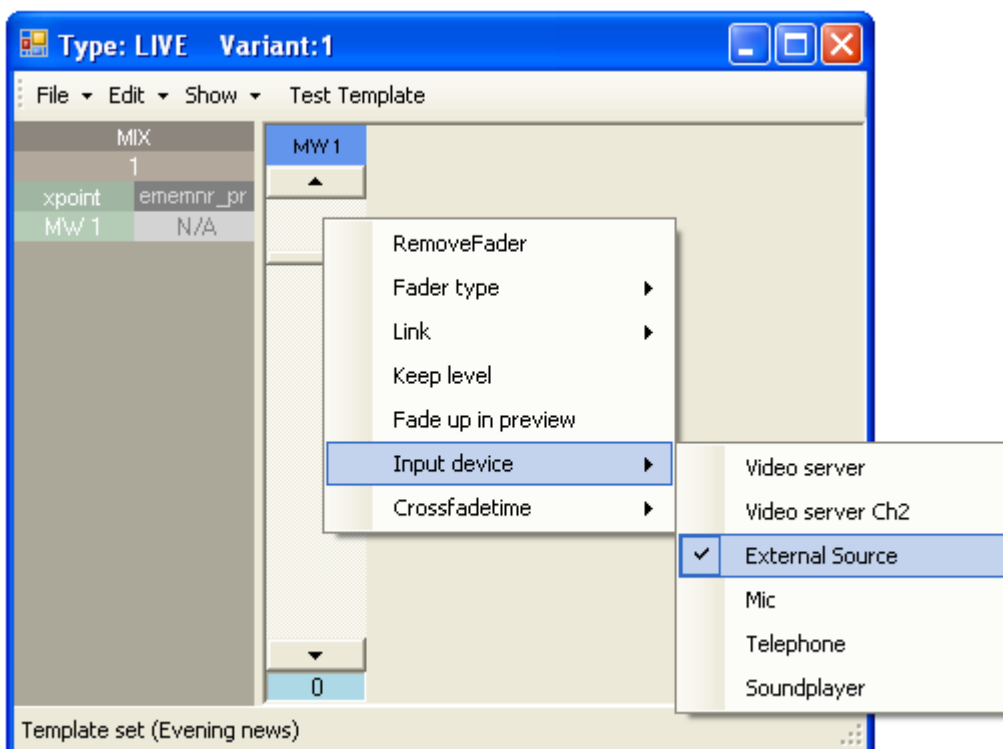
In this example, the variant VO is defined. Notice the difference from the two last examples and the two fader setups. A Mic 1 is microphone input type and RIP_C1 is video server type.



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.

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.



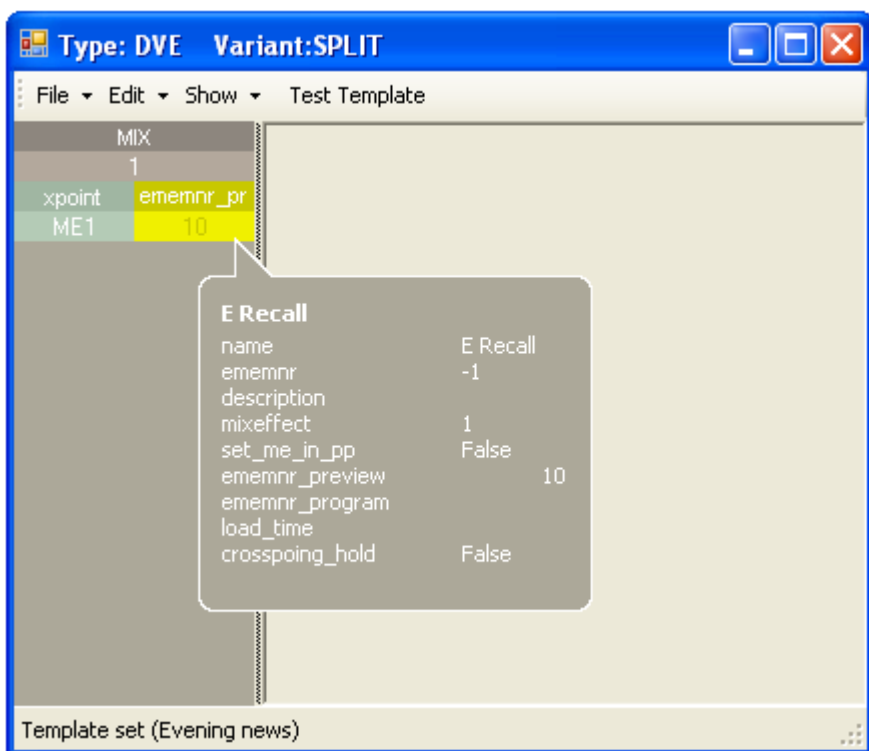
Additional variants 2 to 9 would typically use MW 2 to MW 9 as sources for both video and audio.

DVE

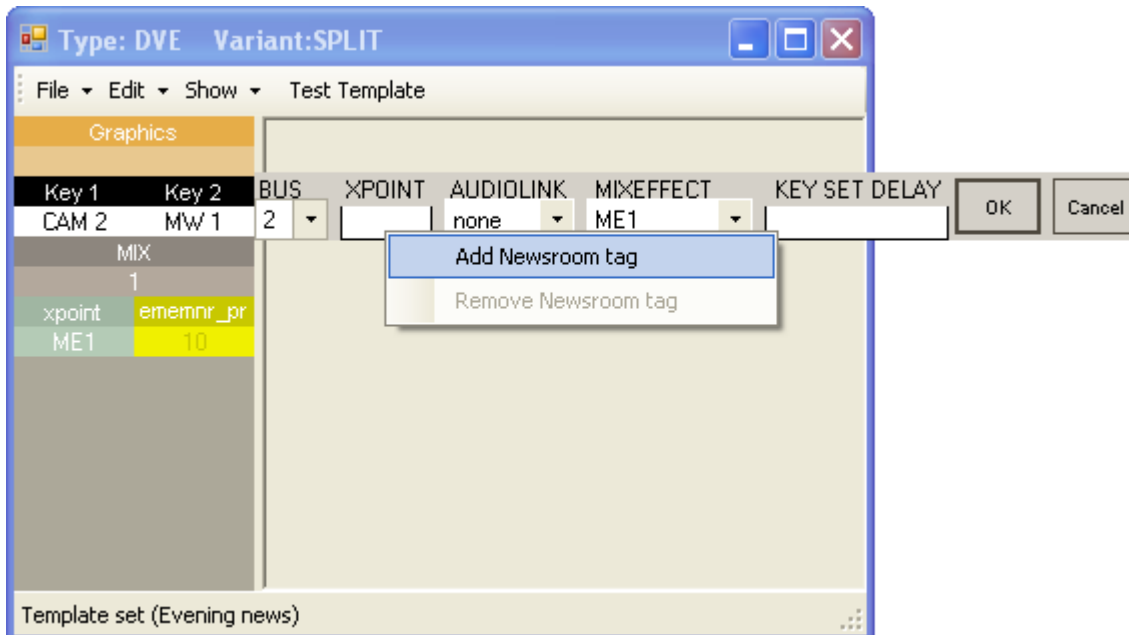
This example recalls a predefined E-mem (emem number 10) defining a split screen DVE on M/E 1 on the vision mixer.

1. E-mem recall definition

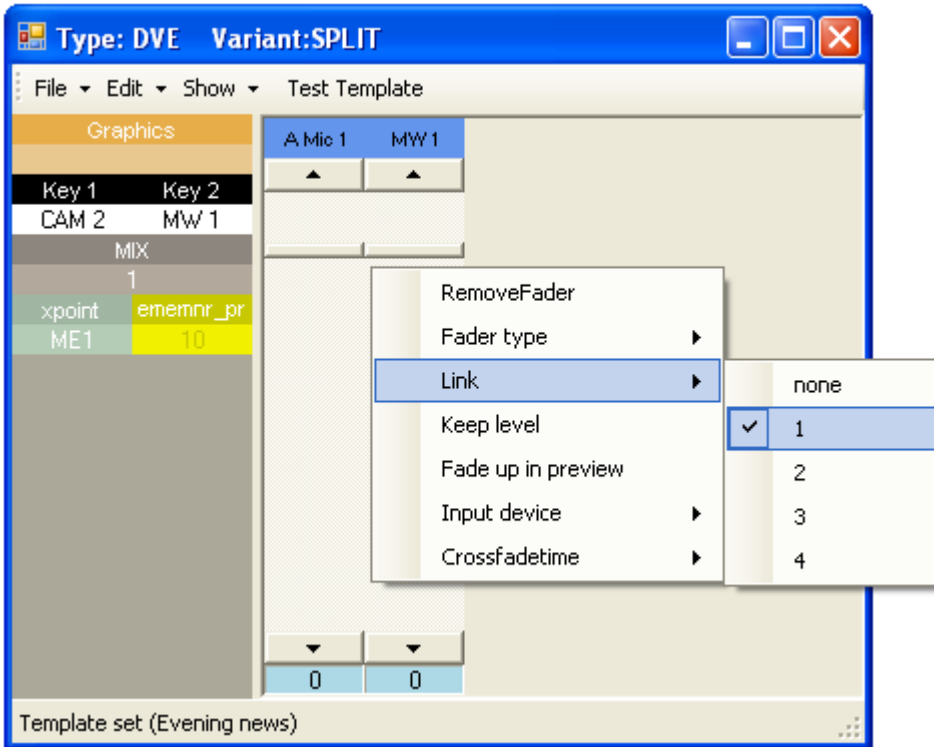
The vision mixer effect uses key 1 and key 2 on M/E 1 as left and right split windows. Default crosspoints are chosen, CAM 2 and MW 1, but the crosspoints can be set from the newsroom system, because a newsroom tag has been added for both keys. Right click over the XPOINT window to add a newsroom tag. In addition, the template defines a recall of a graphics background.



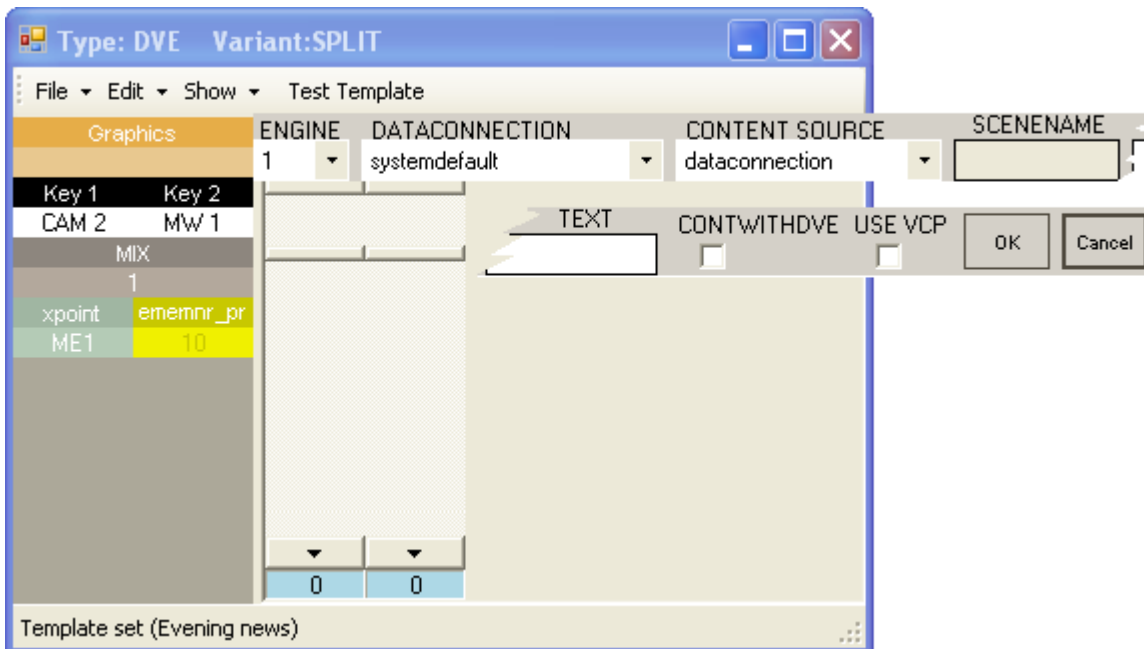
2. Key bus delegation setup



3. Audio linking



4. Graphics engine data element setup



Pictures 1-4, above, show the key steps to configure this template.

Picture 1 shows that 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.

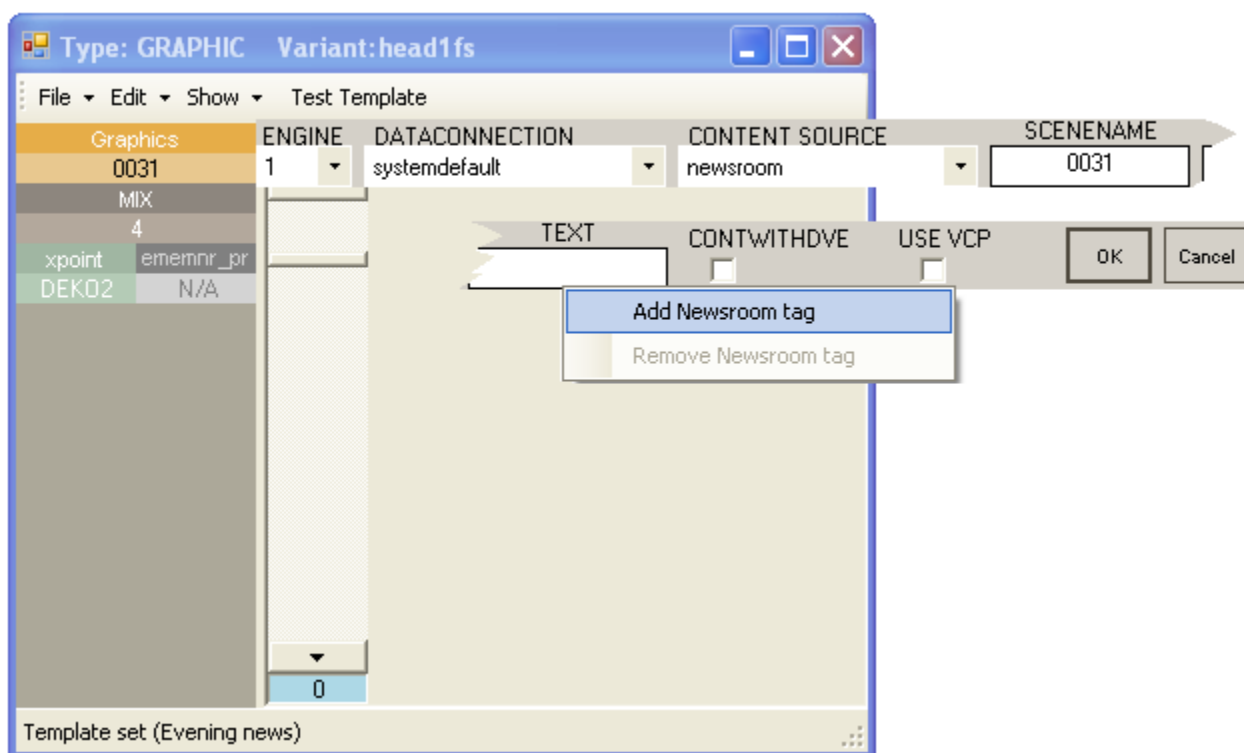
In picture 2, the second key bus delegation is shown. As a default we assign CAM 2 to key bus 1 and MW 1 to key bus 2 on the ME1 mixer effect bus. This assignment can be overridden from the NCS story element by using the newsroom tag. The audio link is set to 1 for both the key 2 delegation and the second fader (MW 1) as shown in picture 3. When using the newsroom tag to set key 2 to a different source (i.e. MW 3), both video and audio delegation will follow, thus key 2 source is set to MW 3 and the second audio fader is set to MW 3.

For setup of the Graphics controller, refer to the next section.

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.



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.

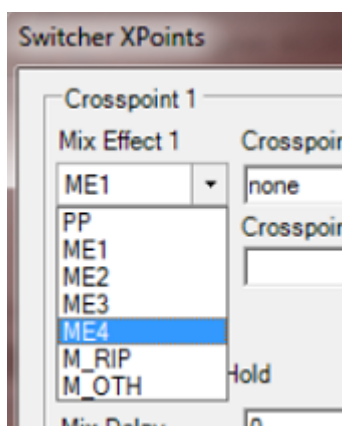
Creating the Wall Accessory Template

The wall accessory template is created as a standard template.

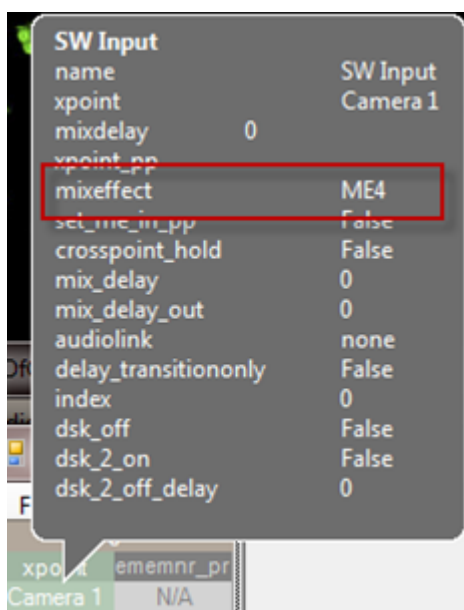
- You can create a new accessory template with a variant name of for instance "wall2".

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.



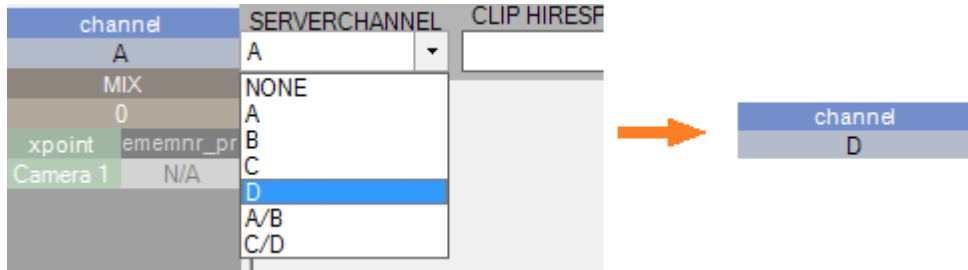
The ME-step will show as a mix effect in the Template Editor.



Video Server Port

If a video clip is wall taken; another video server should be used.

- This can be changed in the accessory.



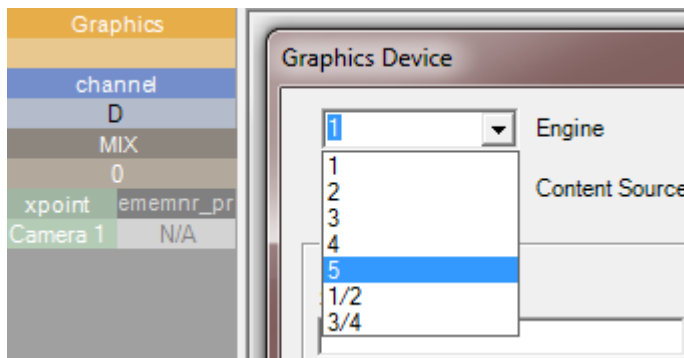
Make sure the video clips are available through this port.

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.



10.6.3 Template Device Functions

This section lists a selection of device properties:

- [Working with Device Functions](#)
- [Video Switcher Crosspoint](#)
- [Video Switcher Transition](#)
- [Video Switcher Register/Timeline Recall](#)
- [Macro Recall](#)
- [Video Switcher Key Bus Delegation \(Keyfill\)](#)

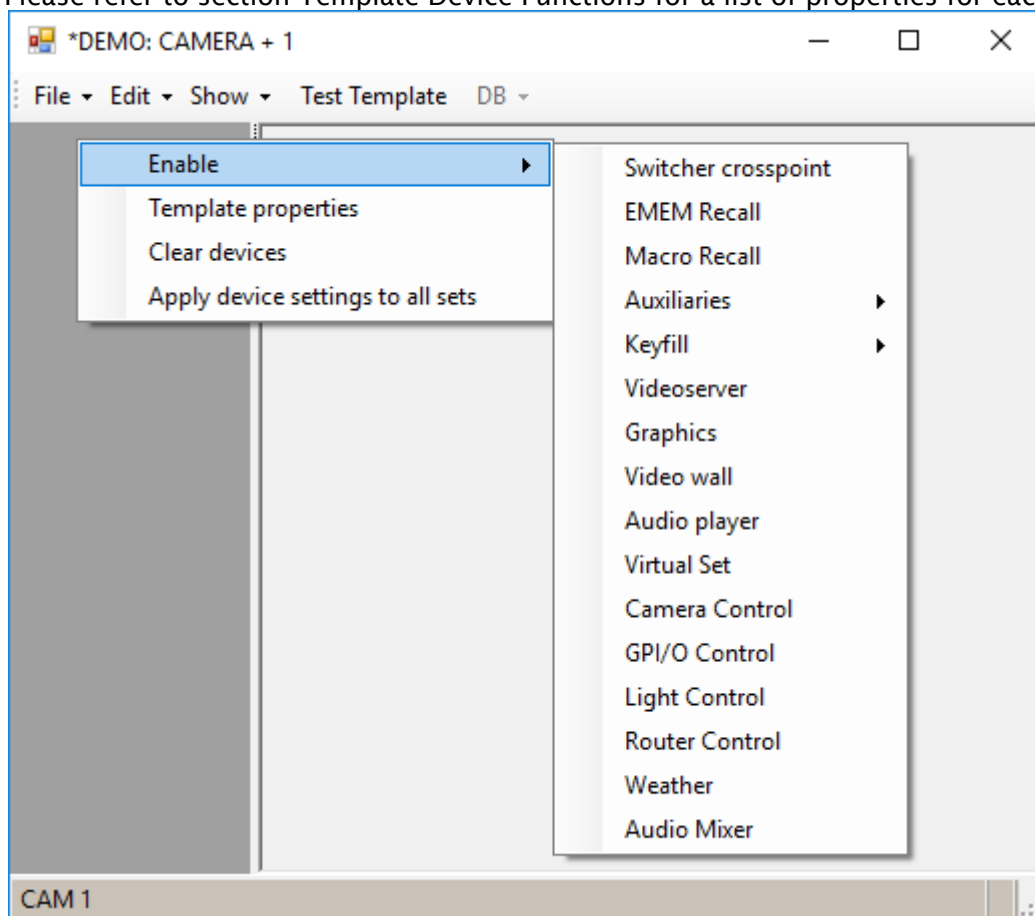
- [Video Switcher Auxiliary Bus Delegation](#)
- [Graphics](#)
- [Robotic Camera Control](#)
- [Router Control](#)
- [Light Control](#)
- [GPI/O](#)
- [Video Wall Register Recall](#)
- [Video Server](#)
- [Audio Player](#)
- [Virtual Set](#)
- [Audio Settings](#)
- [Testing a Template](#)

Working with Device Functions

To Enable Device Functions

1. Right-click in the device Function area.
2. Select the device to configure from the **Enable** menu.
This opens the edit box for the selected device.

3. Please refer to section [Template Device Functions](#) for a list of properties for each device.



1. Right-click over the top half of a device Function area.
2. Select **Device properties**. Please refer to the [Template Device Functions](#) for a list of properties for each device.

Note: Depending on the Protocol chosen, the caption **EMEM Recall** may alternatively be:

- **Emem Recall**
- **Dmem Recall**
- **Snapshot Recall**
- **MemoryRecall**

(**Macro Recall** may alternatively be named **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 NCS

- 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 NCS.

See also: [Newsroom Tags](#).

Video Switcher Crosspoint

The video switcher crosspoint enables switching on the video switcher. The A or B bus assignment of the mixer effect (PP/MEEx) is automatically handled by the automation.

Switcher XPoints

Crosspoint 1

Mix Effect 1	Crosspoint	Audio link
PP	none	none
	Crosspoint PP	

Set ME in PP

Cross Point Hold

Mix Delay: 0 Transition Only

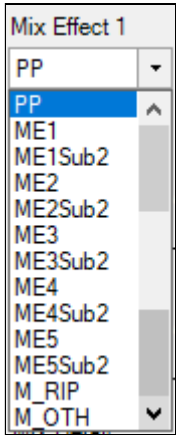
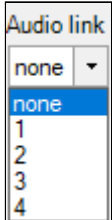
Mix Delay Out: 0

Down stream keyer

<input type="checkbox"/> DSK Off	Off delay
<input type="checkbox"/> DSK 2 On	0
<input type="checkbox"/> DSK 3 On	0
<input type="checkbox"/> DSK 4 On	0
<input type="checkbox"/> DSK 5 On	0
<input type="checkbox"/> DSK 6 On	0
<input type="checkbox"/> DSK 7 On	0
<input type="checkbox"/> DSK 8 On	0

OK Cancel Undo

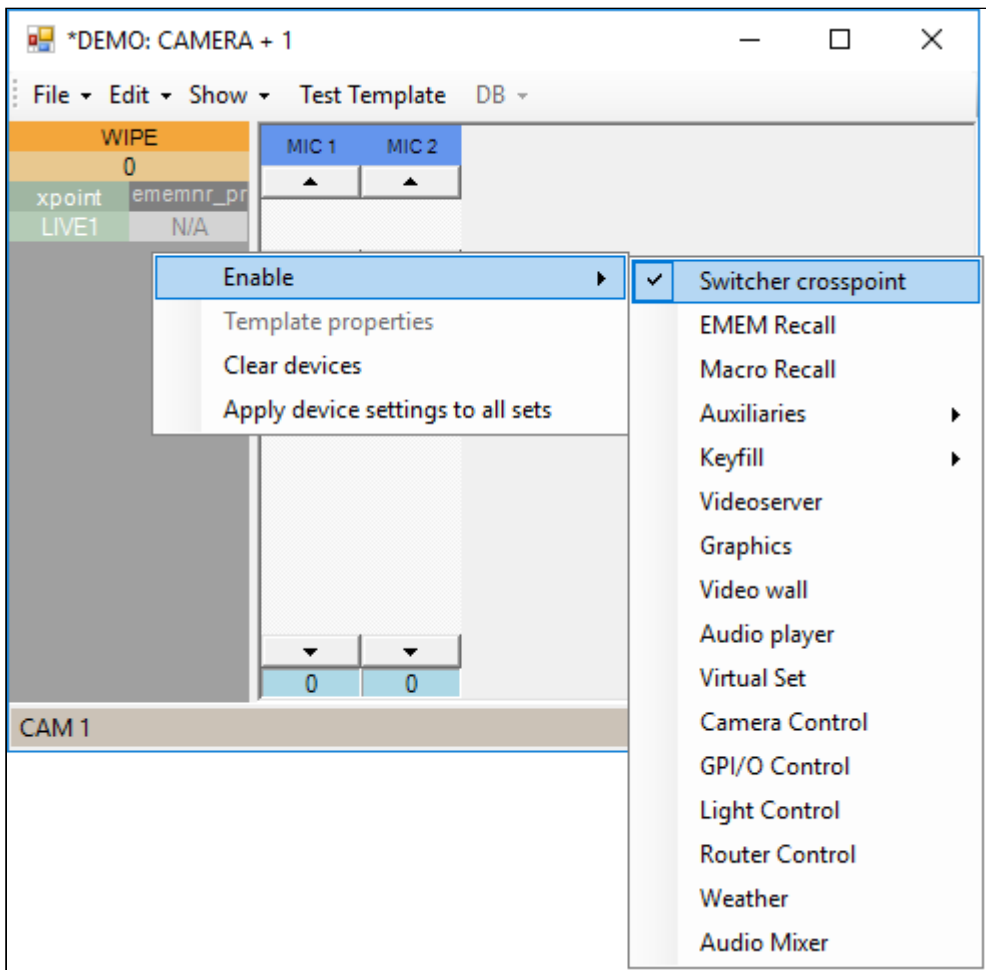
Guidelines for Video Switcher Crosspoint Configuration

Item	Description
Mix Effect	<p>Selects which Mix Effect (ME) bank on your vision mixer that you are currently handling.</p> <p>Note : Newsroom tag available.</p> 
Crosspoint	<p>Crosspoint on the video switcher. Default value: none</p> <ol style="list-style-type: none"> 1. Click OK to display a drop-down list of available x-points. 2. Select from the list. <div style="border: 1px solid #f0e68c; padding: 5px; margin-top: 10px;"> <p>⚠ Note: Newsroom tag available.</p> </div>
Audio link	<p>Default value: <i>none</i></p> 
Crosspoint PP	<p>If another Mix Effect (ME) than <i>PP</i> is selected, then you can select a crosspoint for program by writing the crosspoint name in the field.</p>
Set ME in PP	<p>If another Mix Effect (ME) than <i>PP</i> is selected, check this box to make that ME the crosspoint for PP. (Checking this box disables XPOINT PP).</p>

Item	Description
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.
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), will be used. <input type="checkbox"/> DSK Off
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. <div style="border: 1px solid black; padding: 5px; width: fit-content;"> <input type="checkbox"/> DSK 2 On <input type="text" value="0"/> <input type="checkbox"/> DSK 3 On <input type="text" value="0"/> <input type="checkbox"/> DSK 4 On <input type="text" value="0"/> <input type="checkbox"/> DSK 5 On <input type="text" value="0"/> <input type="checkbox"/> DSK 6 On <input type="text" value="0"/> <input type="checkbox"/> DSK 7 On <input type="text" value="0"/> <input type="checkbox"/> DSK 8 On <input type="text" value="0"/> </div>
Off delay	Delay in frames to set the DSK 2-8 off.

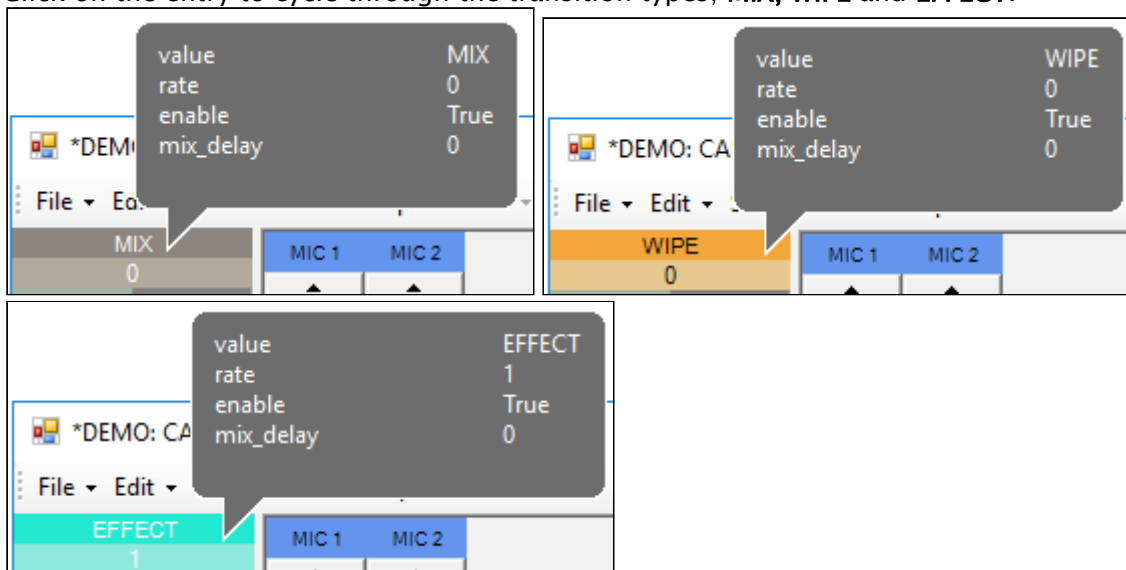
Video Switcher Transition

The transition device is only available when video **Switcher crosspoint** is enabled.

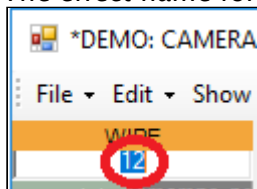


To Define Video Switcher Transitions

1. Click on the entry to cycle through the transition types; **MIX**, **WIPE** and **EFFECT**.



2. Click directly on the value.
3. Enter
 - a. The default duration for the Mix/Wipe transition, or
 - b. The effect name 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*, enter a value in frames in the same property window.
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**.

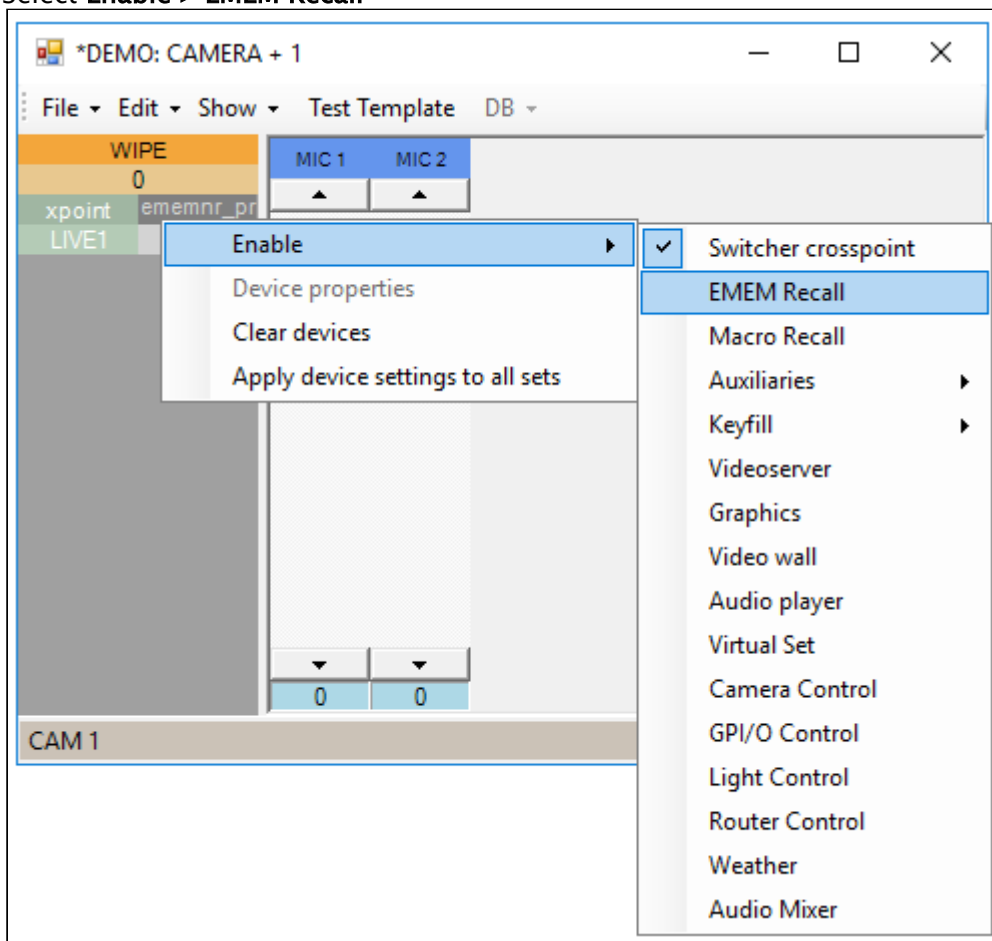
Video Switcher Register/Timeline Recall

Video Switcher Register/Timeline Recall (**EMEM Recall**) will recall a register and/or timeline in the video switcher.

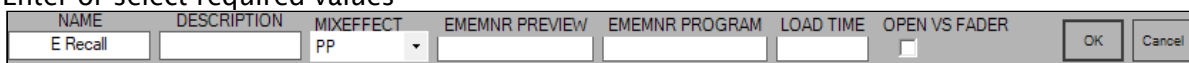
To Open the Video Switcher Register/Timeline Recall

1. Open **Template Editor**.
2. Right-click over **ememnr_pr**

3. Select Enable > EMEM Recall



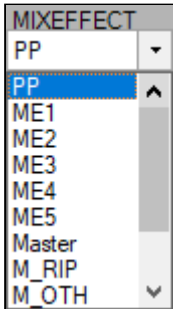
4. Enter or select required values



Note: If a Register recall is used in pair with the video switcher key bus delegation and/or auxiliary bus delegation, 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.

Item	Description
MIXEFFECT	<p>Select the MIXEFFECT (ME) from where the Register/Timeline should be recalled.</p> <p>Drop-down menu values are <i>PP</i> (Program), <i>ME1</i>, <i>ME2</i>, <i>ME3</i>, <i>ME4</i>, and <i>Master</i>. Default value: <i>Master</i>. (Depending on the Protocol chosen, there may be additional values).</p>  <p>To dynamically choose <i>ME1</i> or <i>ME2</i>, values <i>M_RIP</i> and <i>M_OTH</i> are available.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>Note: Newsroom tag available</p> </div>
EEMNRR PREVIEW	Register/Timeline to be recalled when the template is cued in preview.
EEMNRR PROGRAM	Register/Timeline to be recalled when the template is taken to program.
LOAD TIME	Additional delay time (in frames) from when the mixer register is activated (in Program or Preview), till the rest of the template will be 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.

Macro Recall

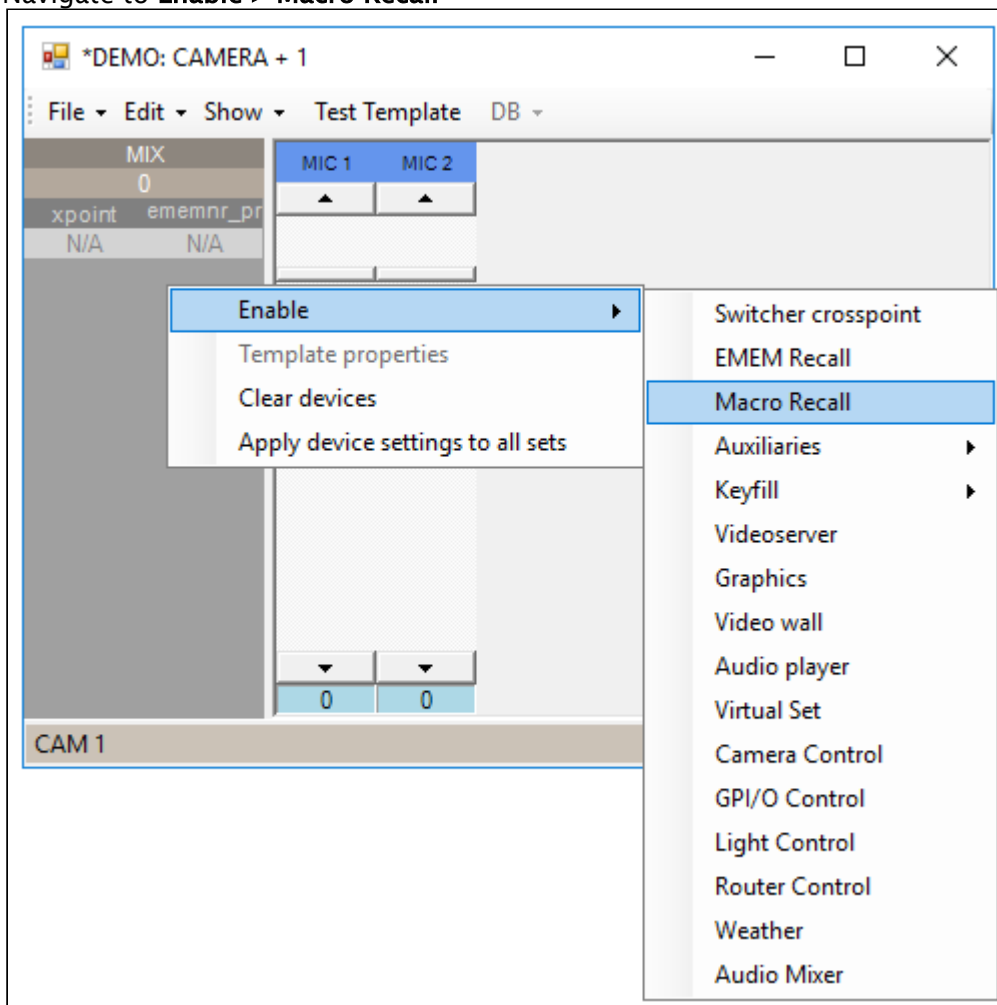
Macro Recall is available when one of the following vision mixer protocols was selected in the configuration menu [AV Automation Devices Vision Mixer](#).

- Carbonite
- GV CPL

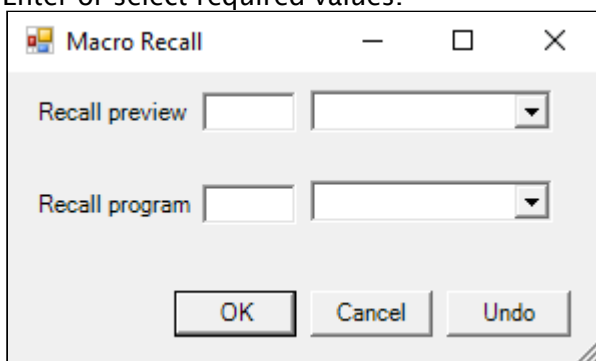
- GV DD35
- KAHUNA (Kahuna/Kula)
- SONY SERIAL TALLY
- vizrt-ip (NewTek\Vizrt Group IP Series Mixers (e.g. TriCaster, VMC1))

To Open a Macro Recall

1. Open **Template Editor**.
2. Navigate to **Enable > Macro Recall**



3. Enter or select required values.



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
Carbonite	One bank character and two macro characters.	Recall (and Recall and take) only. Take does nothing.
GV CPL	Macro number (integer) in the range 1-999.	Recall (and Recall and take) only. Take just takes macro already recalled.

Mixer	Text field	Remark										
GV DD35	Macro number (integer) in the (byte) range 1-255.	Recall (and Recall and take) only. Take does nothing.										
Kahuna/Kula	The meaning and format of the text fields depend on the selected drop-down option:											
	<table border="1"> <thead> <tr> <th>Drop-down option</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Recall</td> <td>Two-digit project number and three-digit macro number in the format <i>pp.mmm</i>.</td> </tr> <tr> <td>Take</td> <td>As for Recall above, or three-digit macro number (<i>mmm</i>).</td> </tr> <tr> <td>Recall and Take</td> <td>As for Take above (if <i>mmm</i> only, the Recall is omitted).</td> </tr> <tr> <td>Load project</td> <td>Project number in the range 0-100 (100 meaning no project).</td> </tr> </tbody> </table>	Drop-down option	Description	Recall	Two-digit project number and three-digit macro number in the format <i>pp.mmm</i> .	Take	As for Recall above, or three-digit macro number (<i>mmm</i>).	Recall and Take	As for Take above (if <i>mmm</i> only, the Recall is omitted).	Load project	Project number in the range 0-100 (100 meaning no project).	
Drop-down option	Description											
Recall	Two-digit project number and three-digit macro number in the format <i>pp.mmm</i> .											
Take	As for Recall above, or three-digit macro number (<i>mmm</i>).											
Recall and Take	As for Take above (if <i>mmm</i> only, the Recall is omitted).											
Load project	Project number in the range 0-100 (100 meaning no project).											
SONY SERIAL TALLY	Macro number (integer) in the range 1-250.	Recall (and Recall and take) only. Take just takes macro already recalled.										
vizrt-ip	Plays a macro having the name provided	Take only										

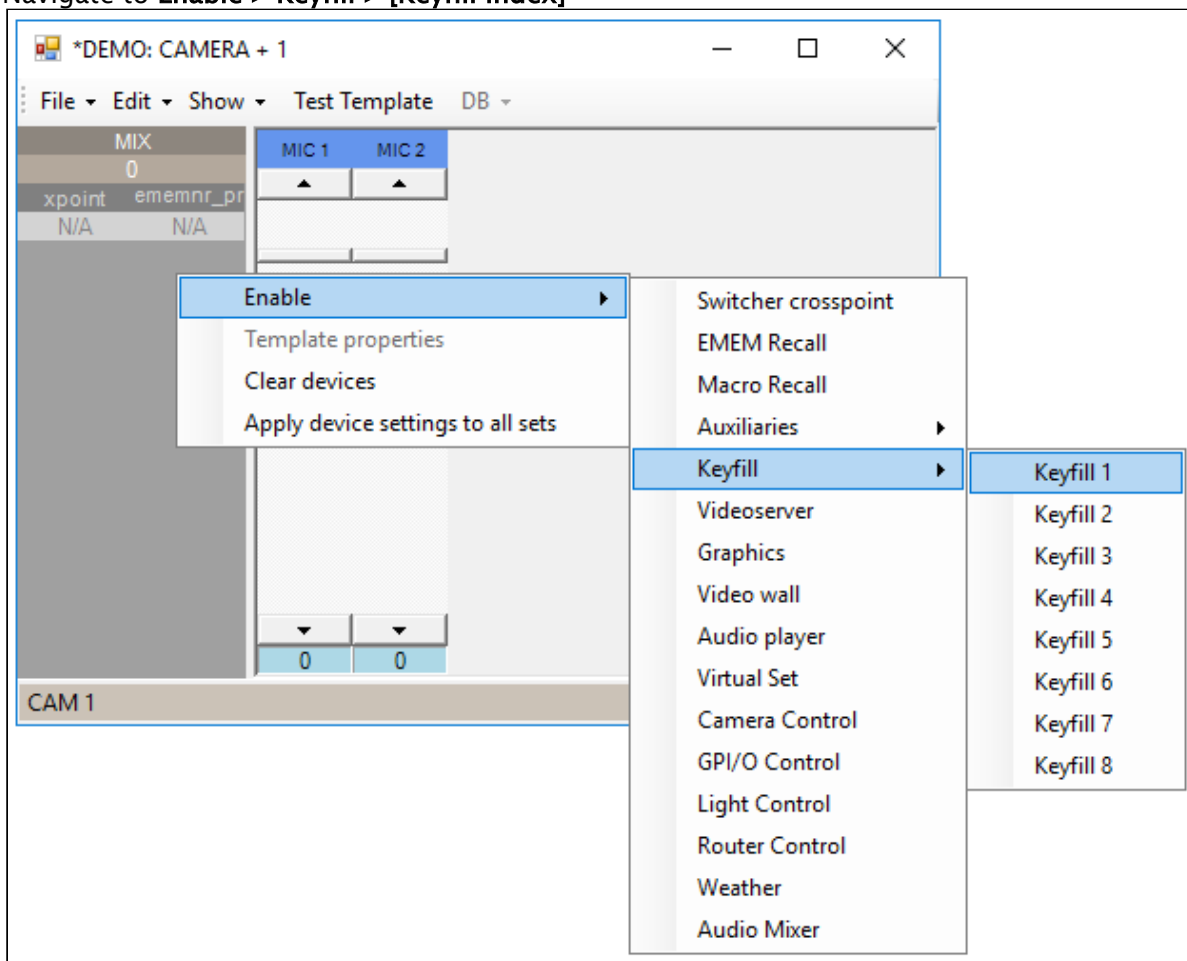
Video Switcher Key Bus Delegation (Keyfill)

Video switcher key bus delegation enables routing of internal video switcher signals to the keyers on the different mixer effects on the switcher.

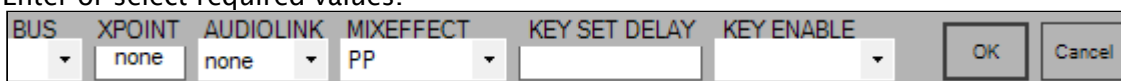
To Enable Keyfill by Bus Delegation

1. Open **Template Editor**.

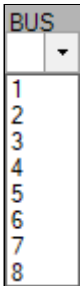
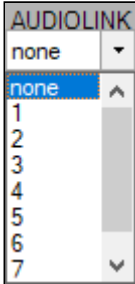
2. Navigate to **Enable > Keyfill > [Keyfill index]**

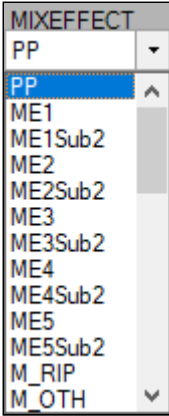
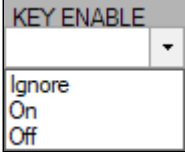


3. Enter or select required values.



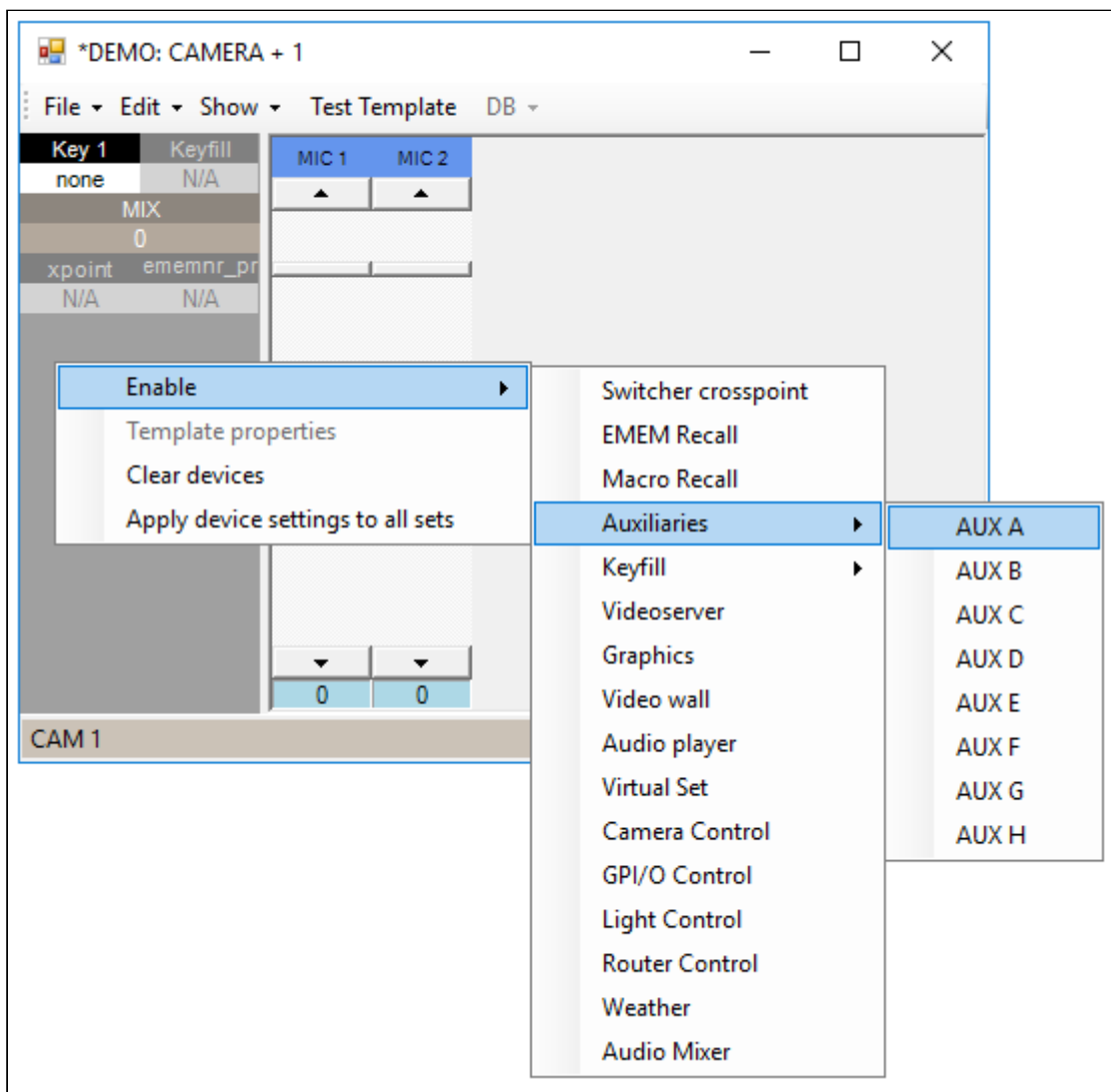
Guidelines for Keyfill Configuration

Item	Description
<p>BUS</p>	<p>Selects the key bus on the selected ME.</p>  <p>Note: Newsroom tag available.</p>
<p>XPOINT</p>	<p>Crosspoint on the video switcher. Default value: <i>none</i></p> <p>Note: Newsroom tag available.</p>
<p>AUDIOLINK</p>	<p>Assigns a link group to this delegation. Please refer to Audio and Video Setup (previously <i>Linking video and audio sources</i>) for the use of this feature.</p> 

Item	Description
<p>MIXEFFECT</p>	<p>Select the ME where the key crosspoint should be set.</p>  <p>Note: Newsroom tag available.</p>
<p>KEY SET DELAY</p>	<p>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.</p>
<p>KEY ENABLE</p>	<p>Key enable drop-down menu.</p>  <p>Note: Newsroom tag available.</p>

Video Switcher Auxiliary Bus Delegation


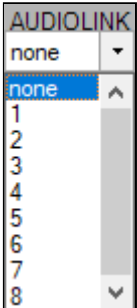
Auxiliary bus delegation enables routing of internal video switcher signals to the auxiliary outputs of your video switcher.



- Type or select relevant values.

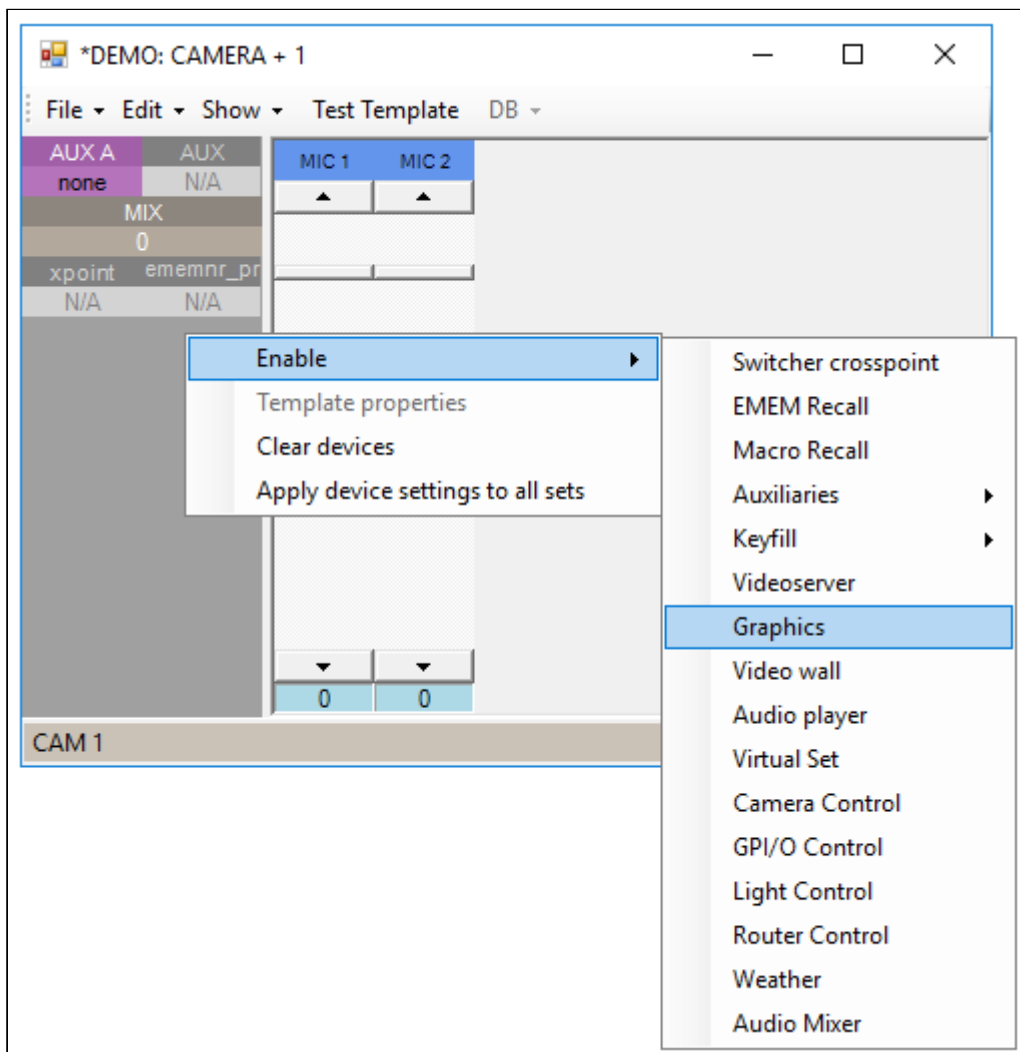
BUS	XPOINT	AUDIOLINK	SET DELAY	TRANS DUR	OK	Cancel
▼	none	none ▼				

Guidelines for Auxiliary Bus Delegation Configuration

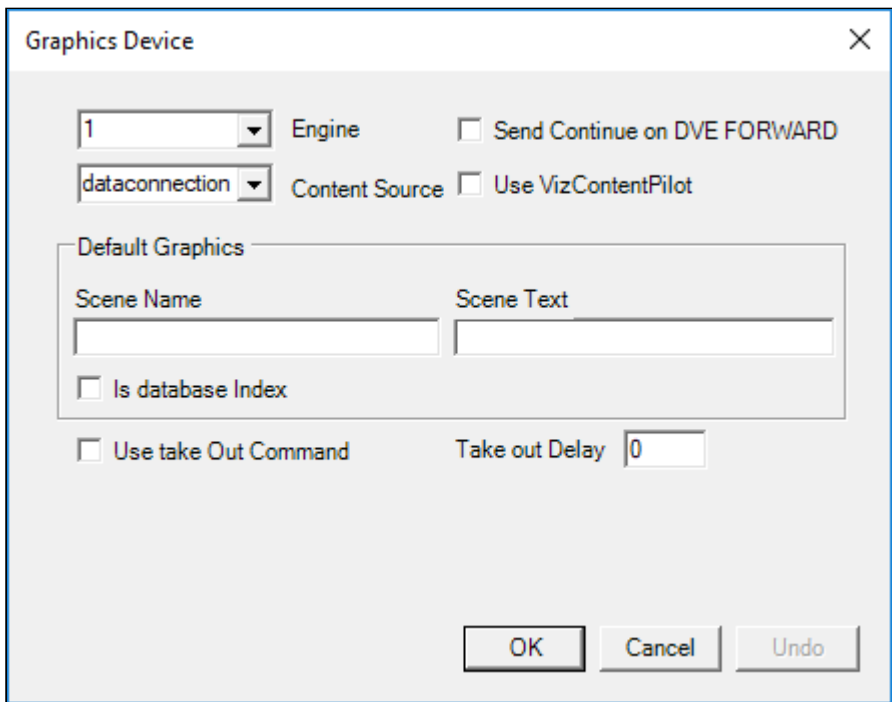
Item	Description
<p>BUS</p>	<p>Selects the auxiliary to assign a new source.</p>  <p>Note: Newsroom tag available.</p>
<p>XPOINT</p>	<p>Crosspoint on the video switcher.</p> <p>Note: Newsroom tag available.</p>
<p>AUDIOLINK</p>	<p>Assigns a link group to this delegation. Please refer to Audio and Video Setup for the use of this feature.</p> 
<p>SET DELAY</p>	<p>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</p>
<p>TRANS DUR</p>	<p>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</p>

Graphics

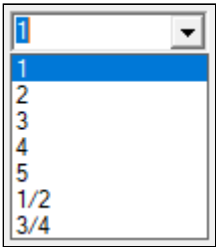
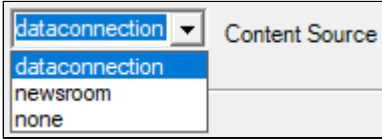
The Graphics device enables recall of graphic elements.



- Graphic elements are cued (loaded) in Preview and played (started) when taken to Program.



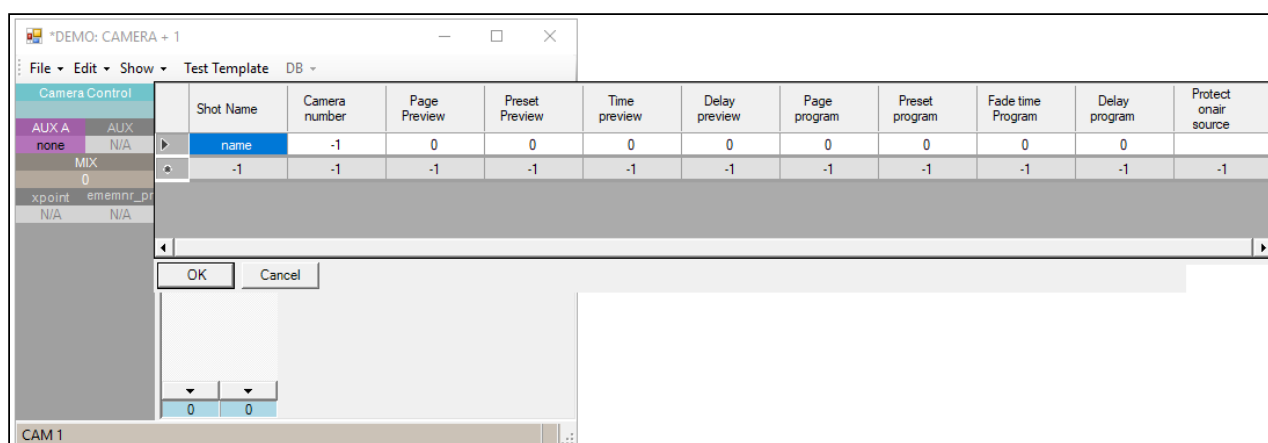
Guidelines for Graphics Configuration

<p>Engine</p>	 <p>Selects the engine to operate on when using this template based on your configuration.</p>
<p>Content Source</p>	 <ul style="list-style-type: none"> · dataconnection: Uses content from the attached graphics database. · newsroom: Uses content from the attached NCS. · none: Uses content written into the template.
<p>Send Continue on DVE FORWARD</p>	<p>If checked, a continue command is sent together with a DVE Forward command.</p>

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 NCS. Note: Newsroom tag available.
Scene Text	The default scene text will be used if not overridden by the NCS. Note: Newsroom tag available.
Is database Index	
Use take Out Command	If checked, 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.

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 [Standard Robotic Camera Control Properties](#) 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](#)
- [Telemetry Properties](#)
- [Vinten 200 Properties](#)
- [Electric Friends Robotic Camera](#)

Guidelines for Standard Robotic Camera Control Configuration

Shot Name	Name of the shot.
Camera number	<ul style="list-style-type: none"> · -1 : Shot is sent to all cameras that are not in standby. · >=0 : Shot is sent to the camera with the same number.
Page Preview	For recalling a shot/move when the template is cued in Preview.
Preset Preview	For recalling a shot/move when the template is cued in Preview.
Time preview	Time to shot's position in Preview.
Delay preview	Delay of the move when cued in Preview (frames). <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;"> <p>⚠ Note: <i>Not recommended for use! Will be deprecated in future versions.</i></p> </div>
Page program	For recalling a shot/move when the template is aired in Program. See <i>Page Preview</i> for an explanation.
Preset program	For recalling a shot/move when the template is aired in Program. See <i>Preset Preview</i> for an explanation.
Fade time Program	Time to shot's position in Program. See <i>Time preview</i> .
Delay program	Delay of move in Program (frames). <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;"> <p>⚠ Note: Only available for Radamec robotics systems</p> </div>

Protect onair source	Protected video switcher cross-point. If the video switcher cross-point is On Air, the shot/move recall will be ignored. This protection may, by configuration for some robots, be confined to Preview and/or Program, and to Cuts and/or Moves.
-----------------------------	--

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 NCS and GUI.
- **Time preview/ Fade time Program:** 0 ~ cut (fastest move) to position, ?0 ~ programmed move speed (adjustable from NCS 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 NCS/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 NCS 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.
- **Delay preview:** Not in use.

- **Delay program:** Not in use.

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 (e.g.: *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 (e.g.: *prepare:ShotName* and *prepareplay:ShotName*).

- **Fade time Program** [Optional]: Duration for move in frames. The time is rounded up to the nearest second.

Shot Name	Camera number	Page Preview	Preset Preview	Time preview	Delay preview	Page program	Preset program	Fade time Program	Delay program	Protect on air source
506 EF_stat_06	2	Brottscentralen:Cam 2	-1	-1	-1	Brottscentralen:Cam 2	PREPARE 506 EF_stat_06	-1	-1	-1
-1	-1									

Camera number (As set in AVA)	Showname : Camera (Camera name as set In EF webserver)	Shot name (PVW)	Not used	Showname : Camera Camera name as set In EF webserver	Shot attribute : shot name (PP)	Not used	Set xpoint (unchanged)
		Time (Speed) of movement (PWV) in frames			Time (speed) of movement (PP) in frames		

- [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](#)
- [Telemetry Properties](#)
- [Vinten 200 Properties](#)
- [Electric Friends Robotic Camera](#)

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.

output_program	Source preview	Dest preview	Source program	Dest program	Source Audiolink	Dest Audiolink	matrix	level	program_delay	salvo_preview	salvo_program
MIX							0	0	0	-	-
0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
xpoint											
ememnr_pr											
N/A											

- 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. <div style="border: 1px solid #ccc; padding: 5px; background-color: #fff9c4;"> <p>⚠ Note: Newsroom tag available.</p> </div>
Dest preview	Destination of routing when a template is cued in Preview.
Source program	Source to route when a template is aired in Program.




Item	Description
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.
level	<ul style="list-style-type: none"> · 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.

Light Control

Light	PREVIEW COMMAND	PREVIEW VALU	PROGRAM COMMAND	PROGRAM VALU
Camera Control	AllOff	-	AllOff	-
Graphics	Fire		Fire	
AUX A	Go		Go	
RUTA	GoOff		GoOff	
Key 1	Load		Load	
RUTA	NoteOff		NoteOff	
MIX	NoteOn		NoteOn	
N/A	Reset		Reset	
xpoint	Restore		Restore	
ememnr pr	Resume		Resume	
	Set		Set	
	Stop		Stop	
	Custom		Custom	

⚠ Note: The commands to use are driver-dependent and may vary.

Guidelines for Light Control Configuration

Option	Description
PREVIEW COMMAND	<p>Command to use for the light mixer cue to be recalled when template enters preview.</p> <p> Note: Newsroom tag available.</p>
PREVIEW VALUE	<p>Light mixer cue to use when template enters preview.</p> <p>Note: Newsroom tag available.</p>
PROGRAM COMMAND	<p>Command to use for the light mixer cue to be recalled when a template is taken On Air.</p> <p> Note: Newsroom tag available.</p>
PROGRAM VALUE	<p>Light mixer cue to use when a template is aired in program.</p> <p> Note: Newsroom tag available.</p>

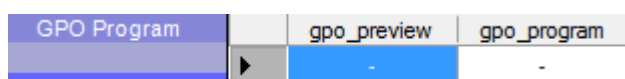
Preview Command and Program Command options are available from the drop-down list:

Command	Parameter count
AllOff	<none>
Fire	1
Go	cue number
GoOff	cue number
Load	cue number
NoteOff	numeric note # (Std. MIDI command)

Command	Parameter count
NoteOn	numeric note # (Std. MIDI command)
Reset	<none>
Restore	<none>
Resume	variable
Set	4 or 9
Stop	cue number
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 (e.g. abc\x20def will send seven bytes as a SysEx message).
TimedGo	- (unsupported)

GPI/O

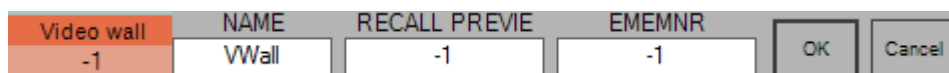
The GPI/O editor is used for sending GPOs to external equipment.



- The editor supports multiple GPO sends.
Add a new line in the editor for multiple GPO sends.
 - **gpo_preview**: GPO to send when a template is cued in Preview.
 - **gpo_program**: GPO to send when a template is aired in Program.

Video Wall Register Recall

This device is only available for Videowall template types.

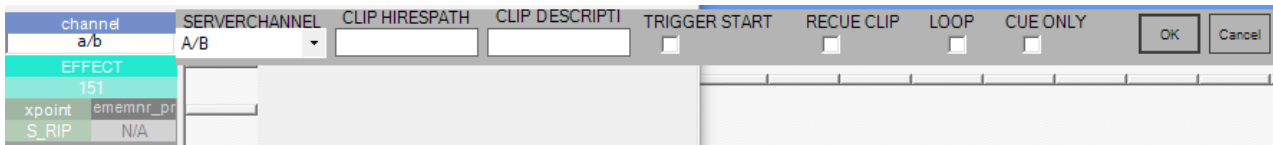


Setting	Description
NAME	Internal name of the shot.
RECALL PREVIE	For WATCHOUT, see notes below. For the other brands, this is the number of the recall to be taken when the template comes in Preview.
EMEMNR	For WATCHOUT, see notes below. For the other brands, this is the number of the preset to take when the template comes 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 'Watchout Connection String' in the [Viz Mosart Administrator Guide](#) .
- In most cases, RECALL PREVIE contains load "show", and EMEMNR contains run "timeline".

Video Server

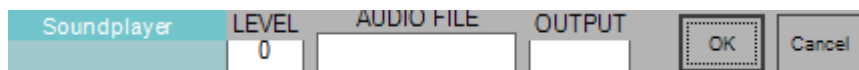


Guidelines for Video Server Configuration

Setting	Description
SERVERCHANNEL	<p>The Mosart video port / port group where the clip is to be played out /recorded.</p> <p>For recording the corresponding video port has to be part of the “Rec” port group.</p> <p>Port groups are created as part of the video server configuration in menu AvAutomation > Devices > Preferences. Configurations are described in AV Automation Devices Video Servers, section <i>Working with Mosart Port Configurations</i>. Notably, the defined virtual ports are among the available options.</p> <p>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 <i>Rec</i> and <i>P</i> groups, if defined), the <i>group</i> as such is not available, but the <i>port</i> is.</p>
CLIP HIRESPTH	<p>The (default) clip ID or name.</p> <div style="border: 1px solid #f0e68c; padding: 5px; margin-top: 10px;"> <p> Note: Newsroom tag available, meaning that a value from the NCS will override the value specified here.</p> </div>
CLIP DESCRIPT	<p>You can enter a Clip description, however a clip description from the NCS or video server overrides this value.</p> <div style="border: 1px solid #f0e68c; padding: 5px; margin-top: 10px;"> <p> Note: Newsroom tag available.</p> </div>
TRIGGER START	<p>Enables the Trigger start function from control commands, a shortcut or continue point could be set to PLAY the Clip.</p>
RECUE CLIP	<p>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.</p> <p>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 <i>only</i> used for AdlibPix templates.</p>
LOOP	<p>Sets the server port to loop the clip. Beware! Not all video servers can loop.</p>

Setting	Description
CUE ONLY	Cues only the clip when the template is in Preview. The template will <i>not</i> start playing the clip when the template is taken.

Audio Player



- **LEVEL:** Enter the value for the audio fader to be set to when taken to Program.
- **AUDIOFILE:** Enter the default audio file name here, this value is replaced by a value from the NCS.

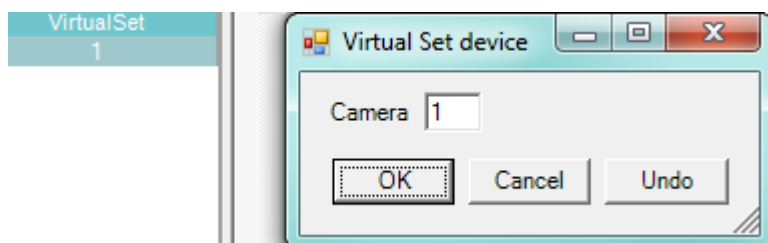
⚠ Note: Newsroom tag available.

- **OUTPUT:** 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 NCS.

⚠ Note: Newsroom tag available.

Virtual Set

You can specify 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.

Audio Settings

- 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 Add Audio Faders

1. Add audio faders to the template by right-clicking in an empty part 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.

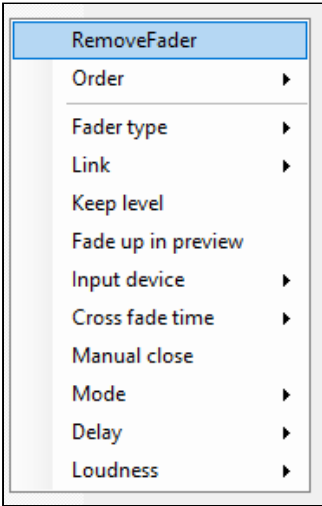
To Remove a Fader

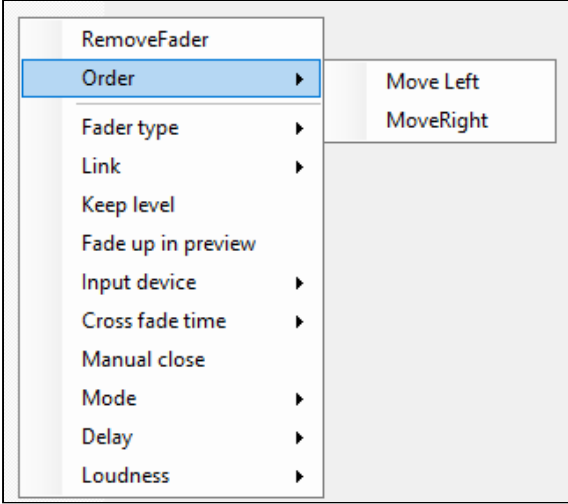
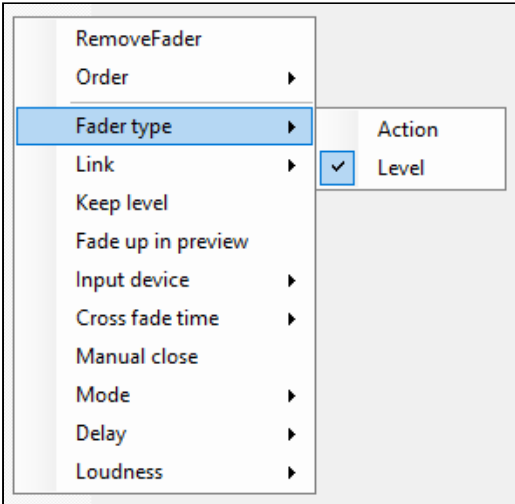
- Right-click over the fader and select **Remove fader**.

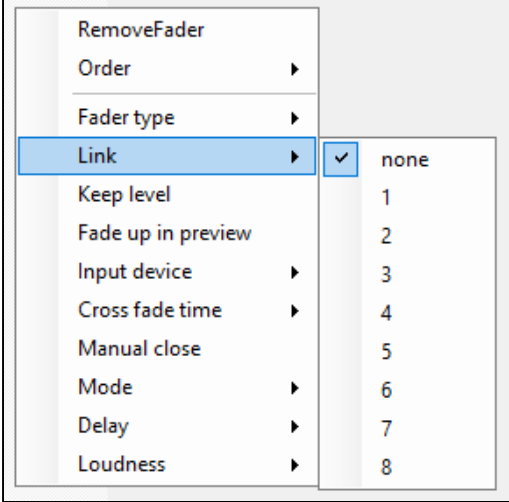
To Configure a Fader

- Right-click over the fader to open the context menu and follow the *Guidelines* below.

Guidelines for Audio Properties Configuration

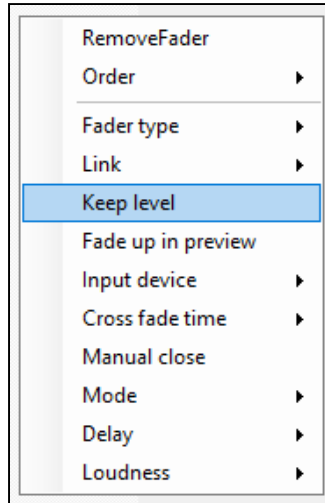
Remove Fader	<p>Removes the fader from the template.</p> 
---------------------	--

<p>Order</p>	<p>Re-arranging the order:</p> <ul style="list-style-type: none"> · <i>Move Left</i> · <i>Move Right</i> 
<p>Fader type</p>	<p>Action: When a fader control is set as an action fader, the controls define whether the fader is enabled or disabled.</p> <p>Level: For normal operation.</p> <ul style="list-style-type: none"> · <i>Action</i> · <i>Level</i> 

Link	<p>Assigns a video/audio link group to this fader.</p> <ul style="list-style-type: none">· <i>none</i>· 1· 2· 3· 4· 5· 6· 7· 8  <p>The screenshot shows a software interface with a dropdown menu. The menu is open, showing the following options: 'RemoveFader', 'Order', 'Fader type', 'Link', 'Keep level', 'Fade up in preview', 'Input device', 'Cross fade time', 'Manual close', 'Mode', 'Delay', and 'Loudness'. The 'Link' option is highlighted in blue. To the right of the 'Link' option, there is a small blue square containing a white checkmark, followed by a list of options: 'none', '1', '2', '3', '4', '5', '6', '7', and '8'.</p>
-------------	---

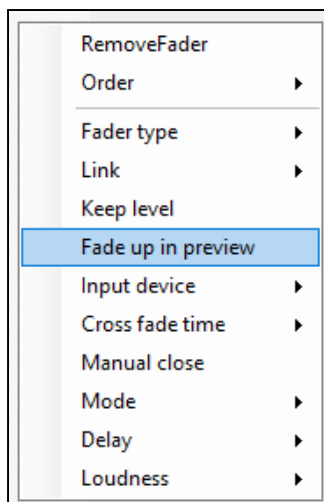
Keep level

Keeps the fader level until manually changed.
Enabled to let the automation open the audio fader, leaving it open until the operator closes it with the **Fade audio** function in the Viz Mosart client, or closes it from another template.



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	✗

Input device

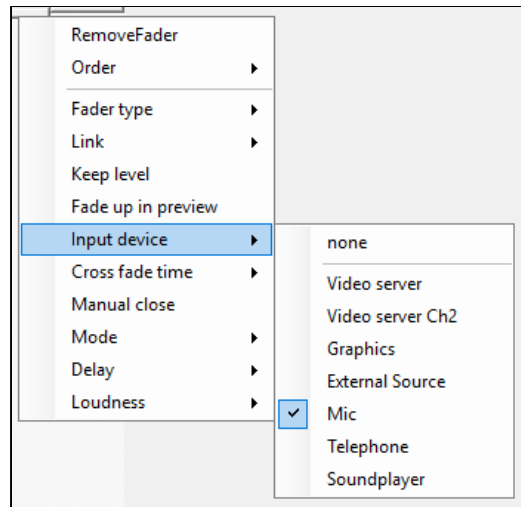
Sets the type of the audio fader.

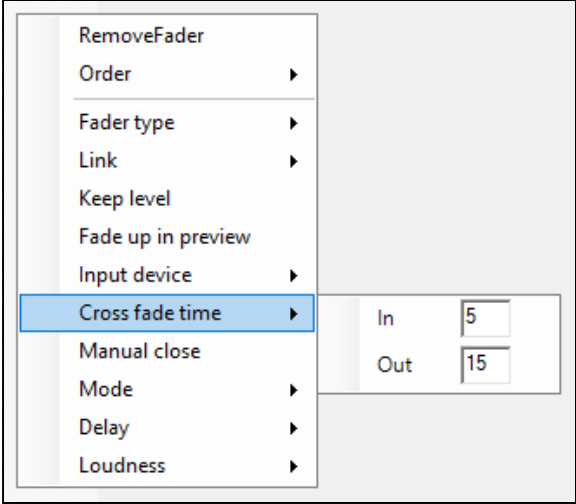
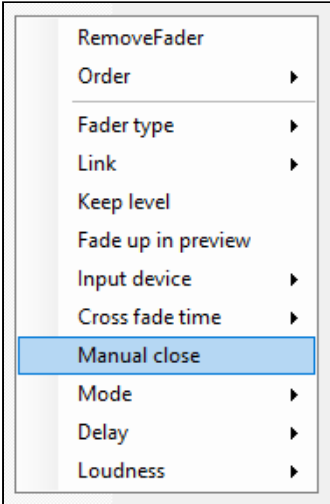
The setting is used in combination with primary template type and On Air status to identify which fader should be treated as secondary audio.

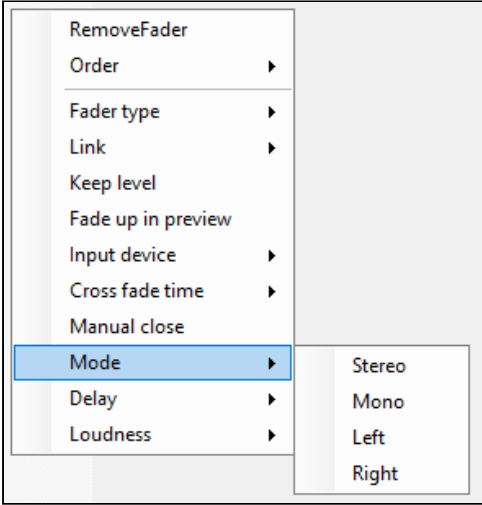
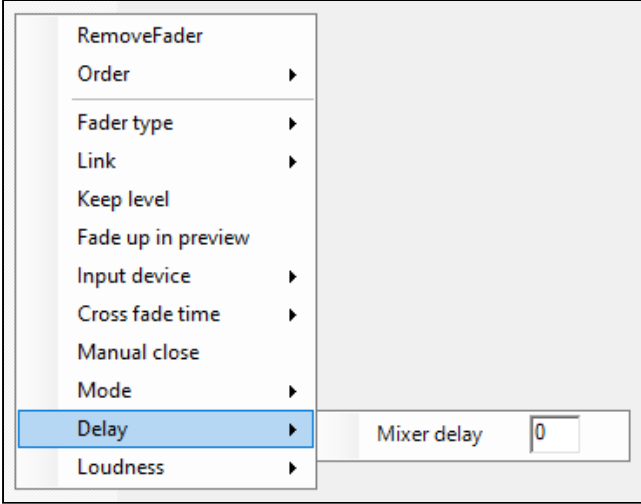
This special handling is applied to the types below:

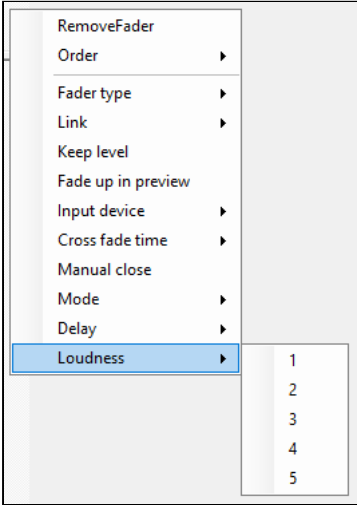
- **none:** No special treatment.

- **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.



<p>Crossfade time</p>	<p>Sets the in and out crossfade time for the fader.</p> <ul style="list-style-type: none"> • <i>In</i> • <i>Out</i> 
<p>Manual close</p>	<p>Select to ignore sending Close-level commands to audio mixer.</p> <p>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.</p> 

<p>Mode</p>	<p>Only for Studer Vista and Lawo!</p> <p>Sends the chosen value to the mixer board for the audio fader.</p> <ul style="list-style-type: none"> • <i>Stereo</i> • <i>Mono</i> • <i>Left</i> • <i>Right</i> 
<p>Delay</p>	<p>Only for Studer Vista and Lawo!</p> <p>Gives a delay value for that audio fader that will be sent to the audio board.</p> <ul style="list-style-type: none"> • <i>Mixer delay</i>. Default: 0 

Loudness	<p>Sends chosen recall value to send to the Junger audio processor.</p> <ul style="list-style-type: none"> · 1 · 2 · 3 · 4 · 5 
-----------------	---

Testing a Template

You can test the behavior of your template by simulating a Preview then Take.

To Test a Template

- Select **Test template** from the main menu.
This cues the template in Preview then five seconds later take it to Program with the selected transition.

10.6.4 Working with Template Sets

This section contains the following topics:

- [Template Set Types](#)
 - [Standalone](#)
 - [Hierarchical](#)
- [Template Set Operations](#)
- [Template Set Properties](#)

Template Set Types

A Viz Mosart template and its variants are arranged in a *set*. There are two approaches to working with template sets.

- [Standalone](#)
- [Hierarchical](#)

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.

Hierarchical

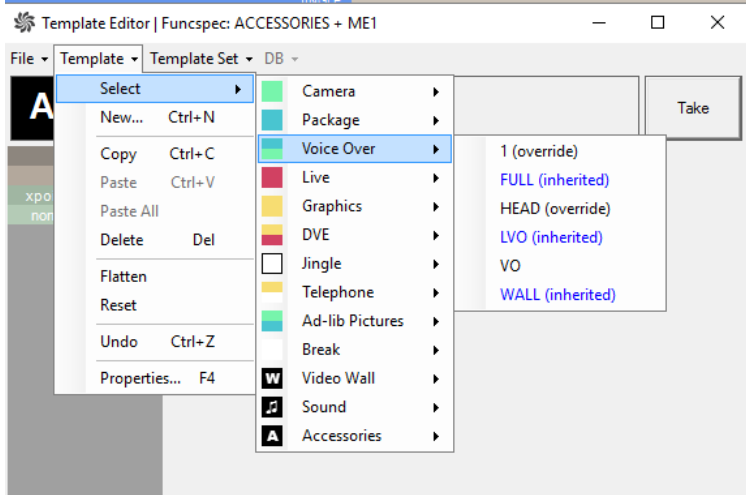
An alternative approach to standalone is *hierarchical* template sets. This method also enables a new template set to be created, based on the logic of an existing template set. However, to begin with, *no* templates are copied from the existing template set.

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 [hierarchical 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.

Template Set Operations

To work with a template set, from the AV Automation utility, select **Devices > Template editor**, to open the editor.

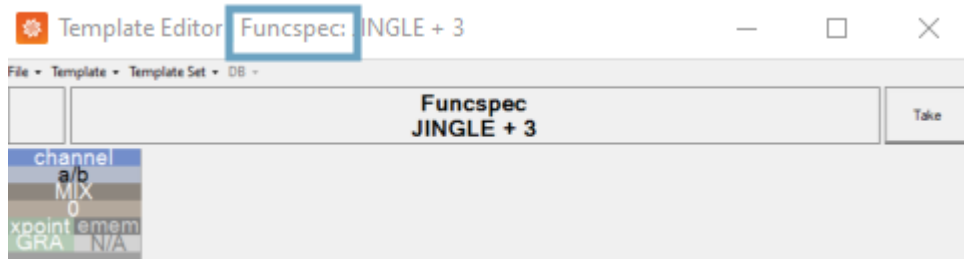
From here you have several operations:

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**.

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

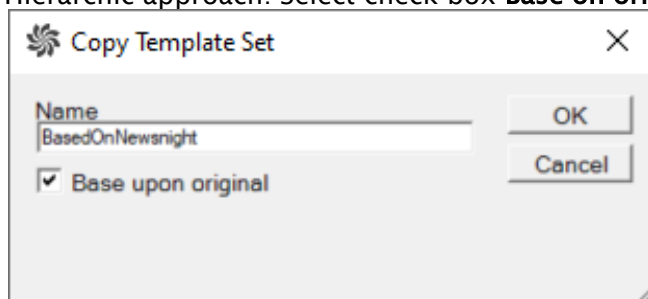
1. Select **Template set > Delete**.
2. In the dialog box, click **Yes** to confirm delete.

⚠ 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 [To Copy a Template Set](#) 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**.



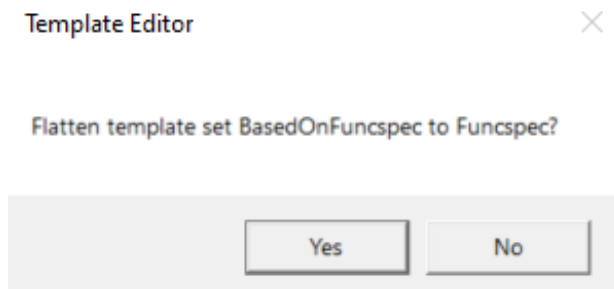
Note: If the **Base on original** check-box is *not* selected a physical copy of the entire template set including all nested templates, is created.

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.



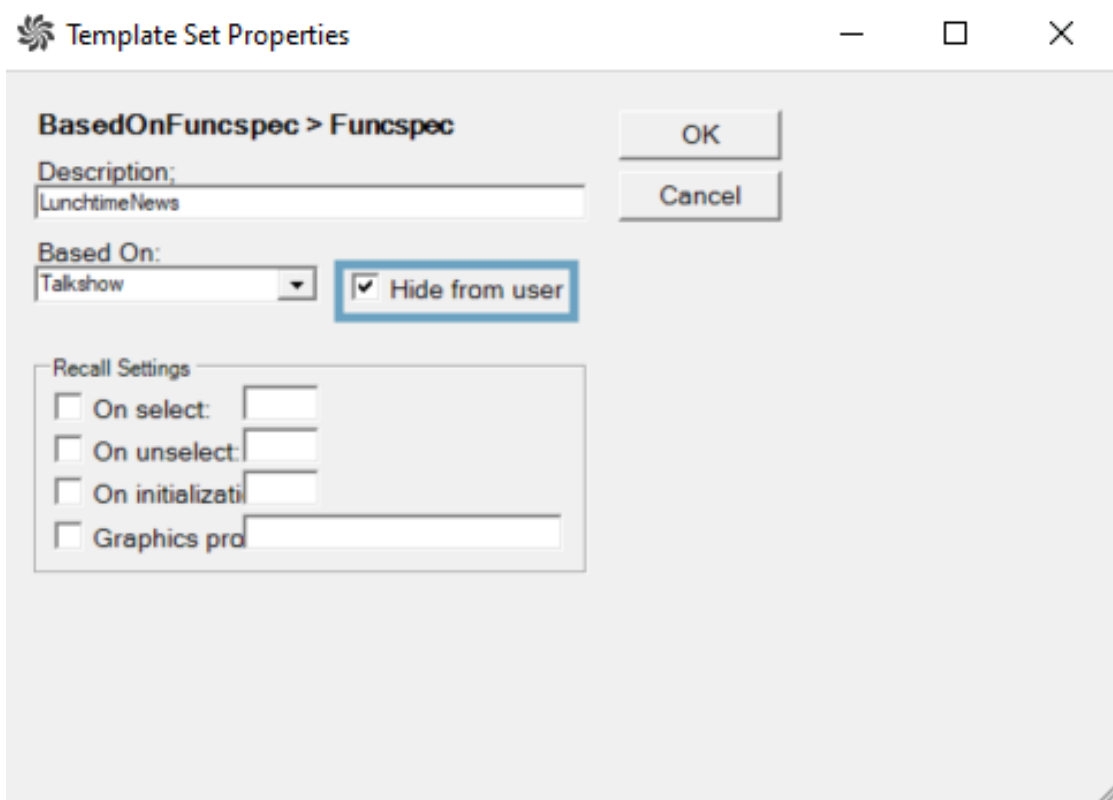
3. Click **OK**.
All templates in the selected set are now inherited from the base set.

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** .

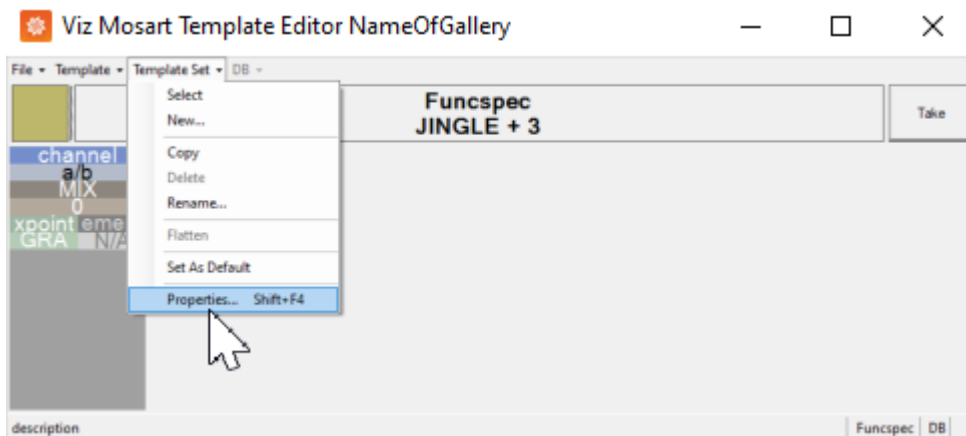
2. Select check-box **Hide from user**.



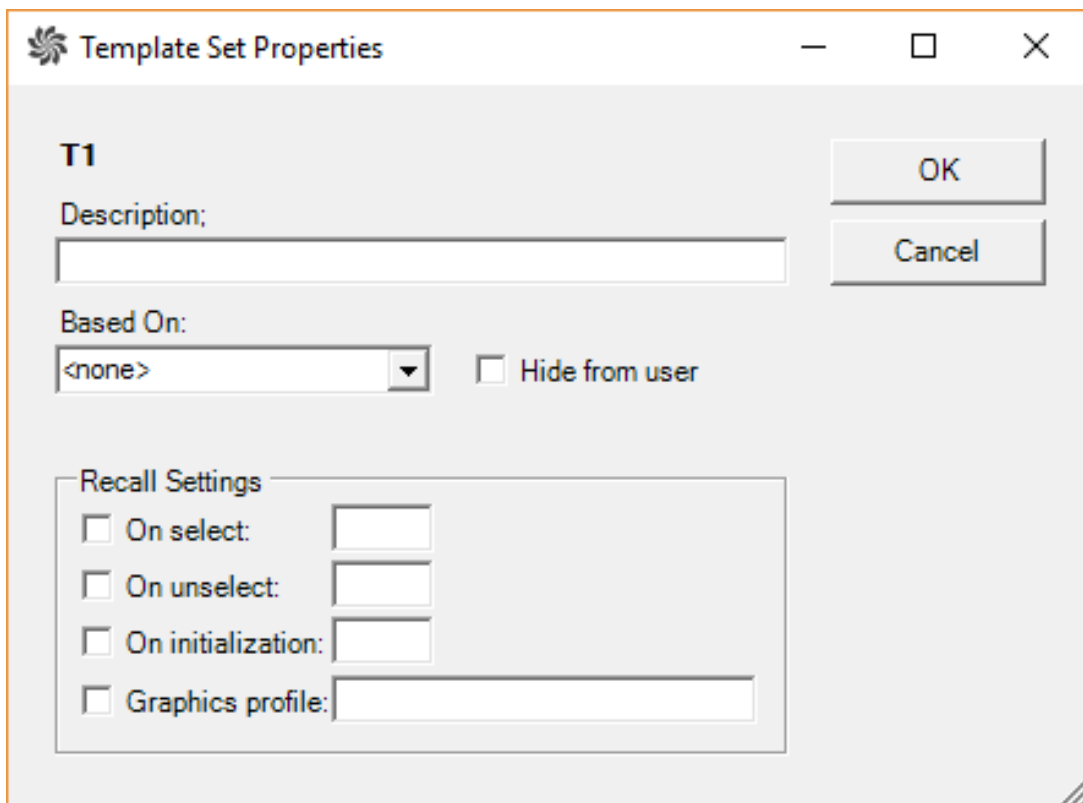
3. Click **OK**.

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.

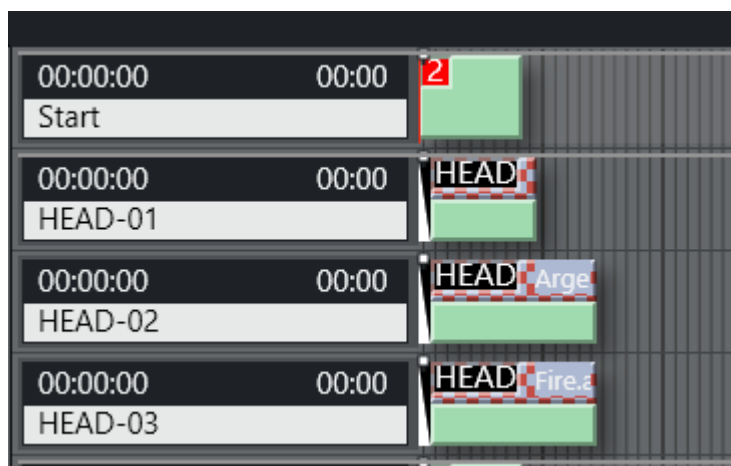
Item	Description
Graphics profile	(When used) Profile that will be applied to the new template set.

10.6.5 Other Template Functionality

- [State Variance](#)
 - [Example of Using State Variants](#)
 - [To Configure Template State Variance](#)
 - [Example - Reset State After](#)
- [Dynamic ME Allocation](#)
- [Template Editor Password](#)
 - [To Enable or Change a Template Editor Password](#)
 - [To Reset a Template Editor Password](#)

State Variance

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, it is 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.

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 Template State Variance

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 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.

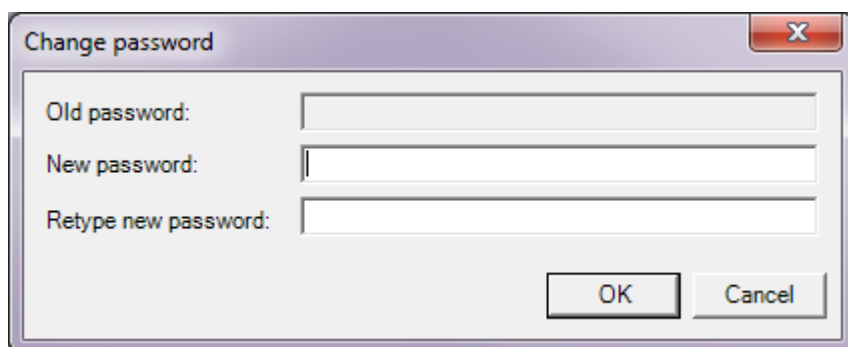
Template Editor Password

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

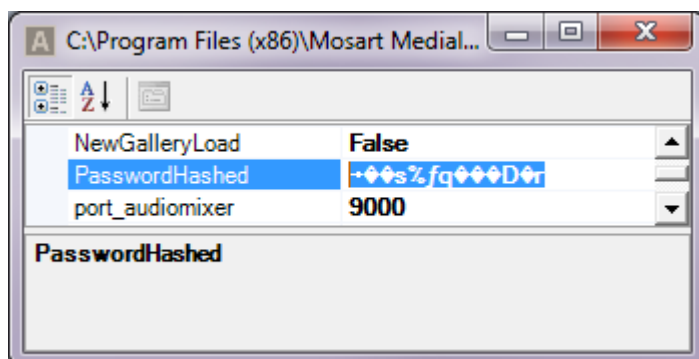
1. Set focus on AV Automation.
2. Select **Devices > Change Password**.
3. In the **Change password** dialog box, enter a new password or change an old one.



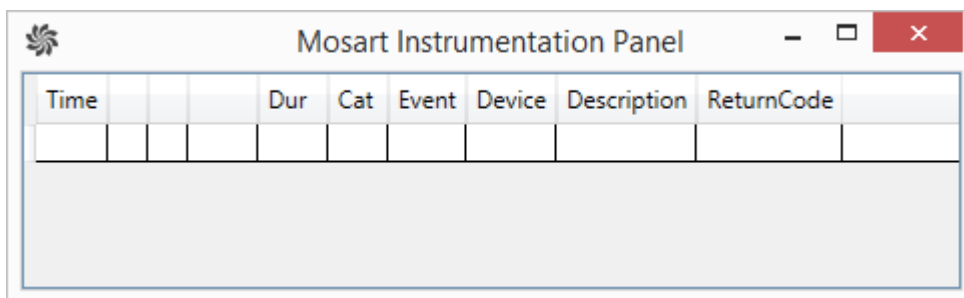
To Reset a Template Editor Password

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.



10.7 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.

10.8 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.

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 *automaticall*y 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.

- (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:

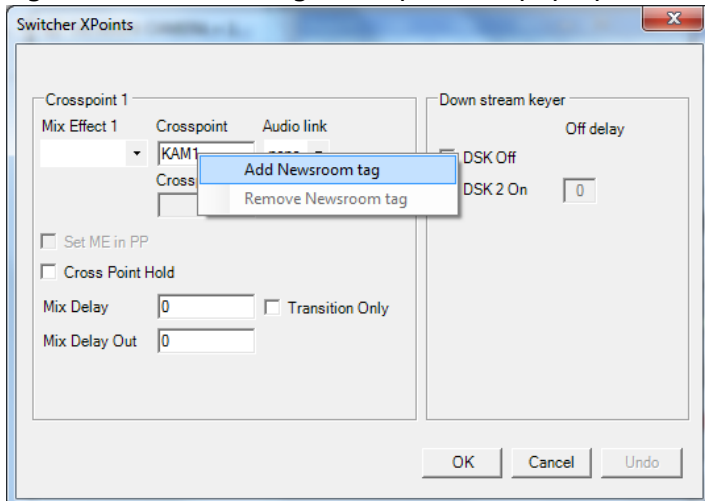
- [Adding a Newsroom tag to your Viz Mosart Template](#)
 - [To Add a Newsroom tag](#)
- [Newsroom Tags Order](#)
 - [To Set newsroom tag order](#)
- [Advanced Features for Specific NRCSs](#)
 - [To Use a Column in ENPS as a Newsroom Tag](#)
 - [Inheriting Newsroom Tags \(applies to MOS-based NRCS s only\)](#)
 - [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](#)

10.8.1 Adding a Newsroom tag to your Viz Mosart Template

To Add a Newsroom tag

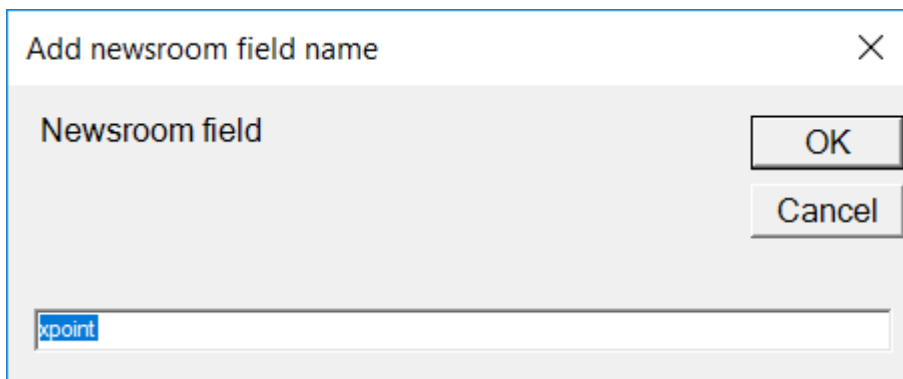
- Refer to section [Working with Device Functions](#), which lists the properties of the device where you want to add the Newsroom tag (e.g. the 'Switcher XPoints' as shown below).

- 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:

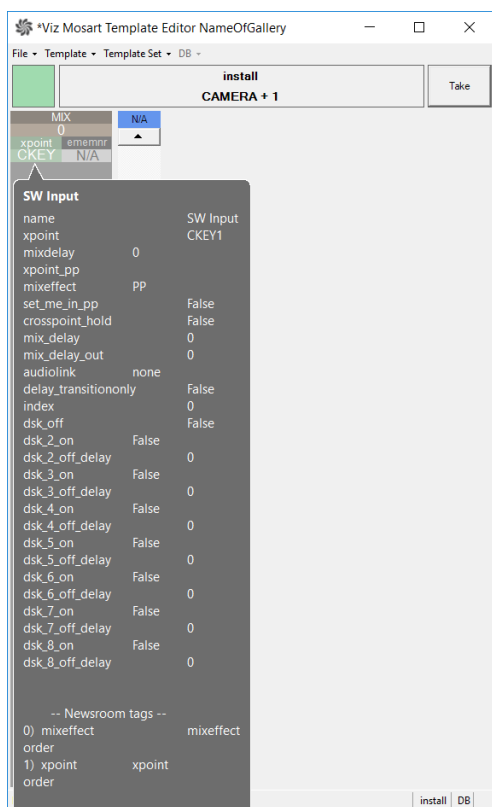


Please note that a default name may be given.

- 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.

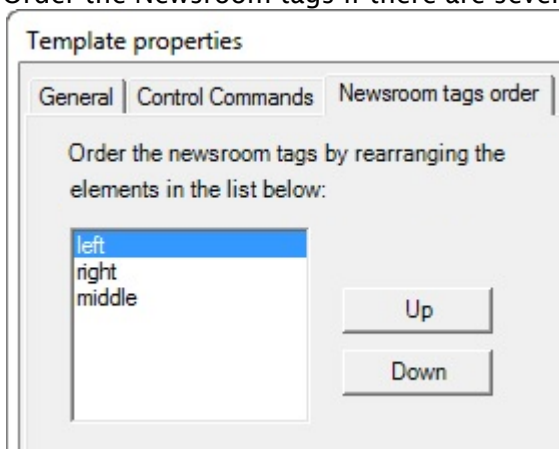


10.8.2 Newsroom Tags Order

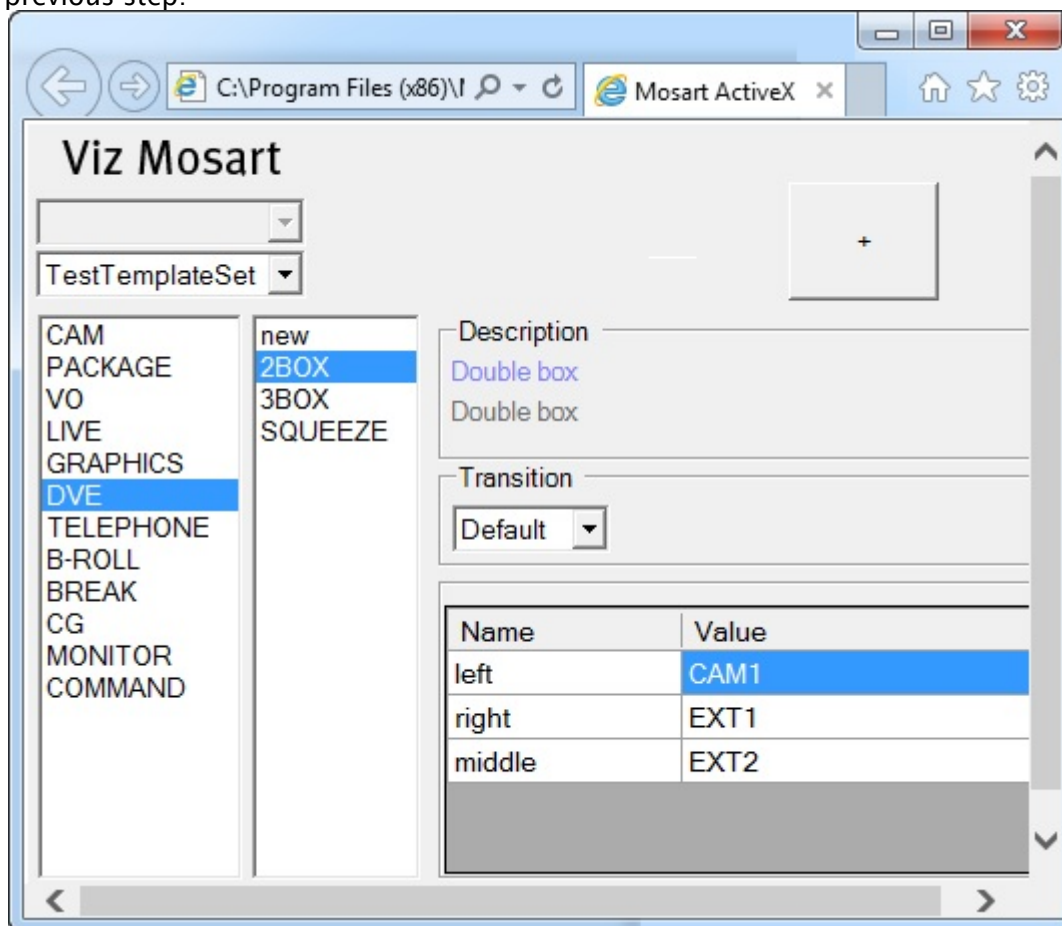
If a template has more than one newsroom tag, the order in which the tags are shown in Viz Mosart’s ActiveX, may be changed in the Template editor.

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**).



4. To verify, open the Activex in the NRCS, and observe that the tag order is as set in the previous step:



10.8.3 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 namings are 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.

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**
- Add the new ENPS field as, for example, an item_variable.
 - 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).
 - 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, 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}'.

Inherit with default value

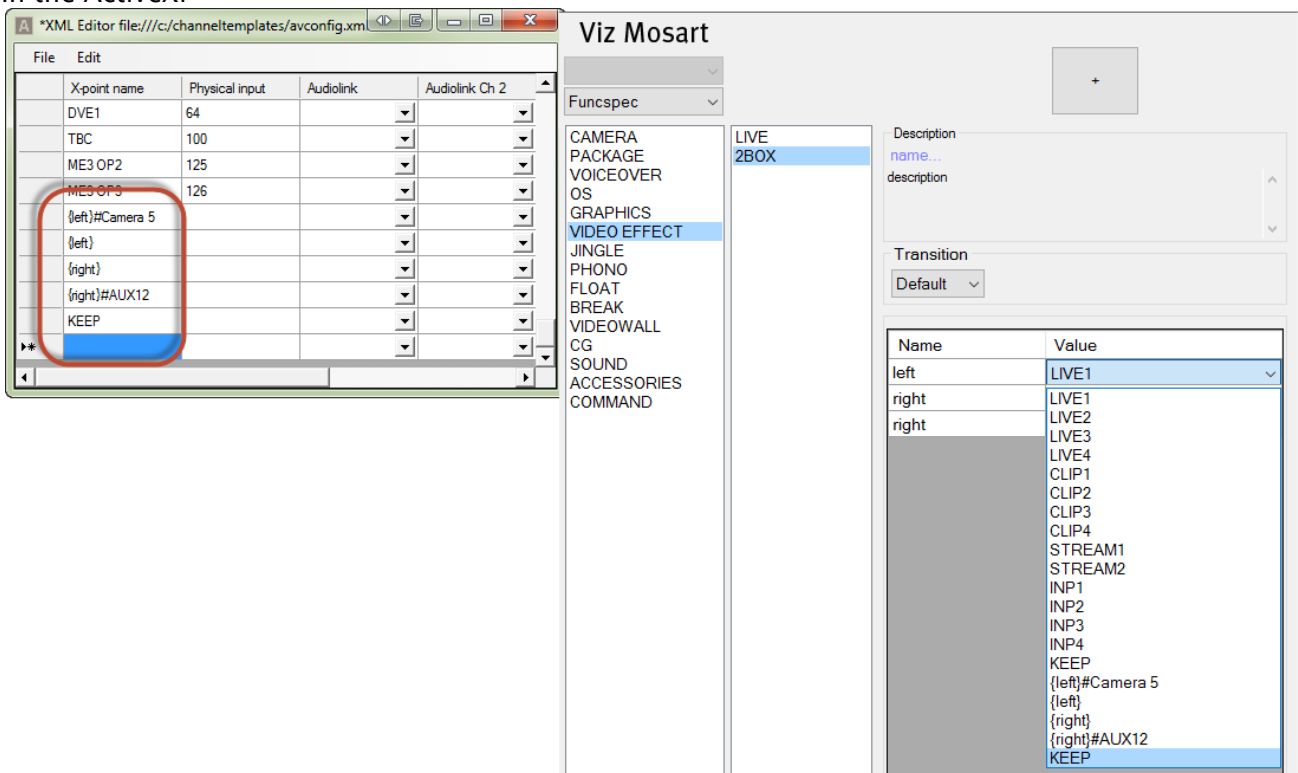
Optionally, a default value may be given, to be used when the tag inherited from has no value (in the story), or does not exist at all. The general syntax (for the value of the inheriting tag) is '{name} #default', where *name* is the tag inherited from (as above), and *default* is the default value.

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:



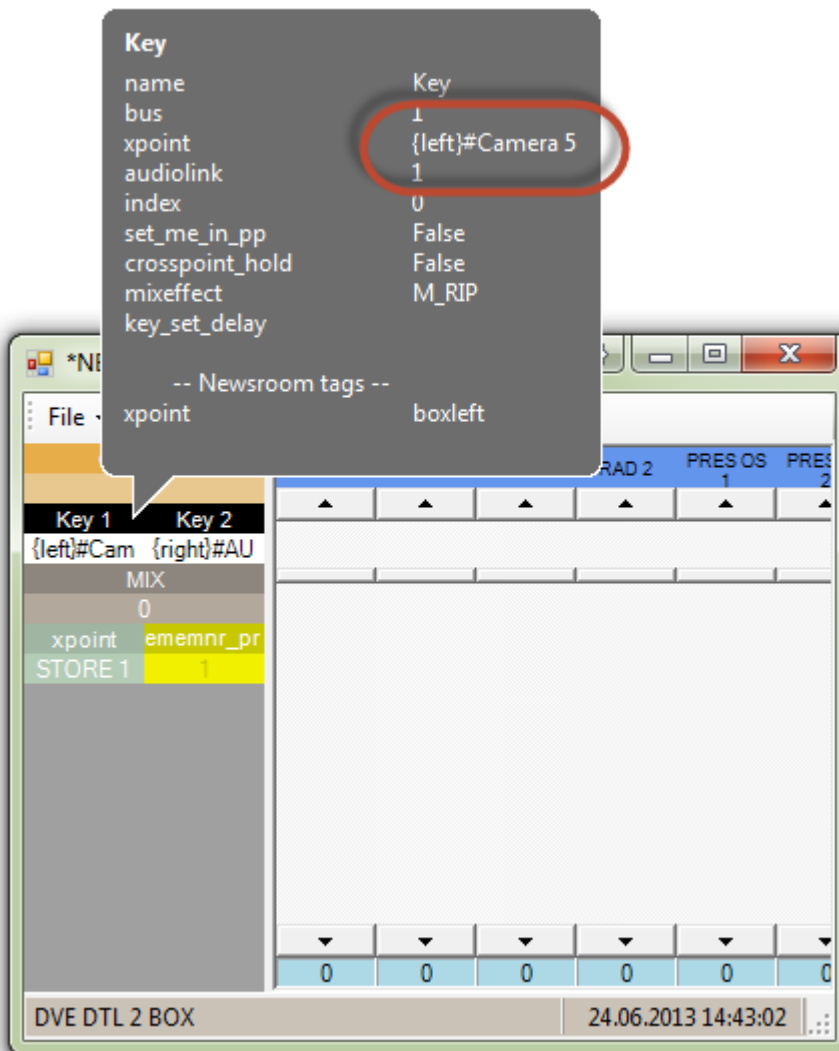
Explanation:

- {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
- 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

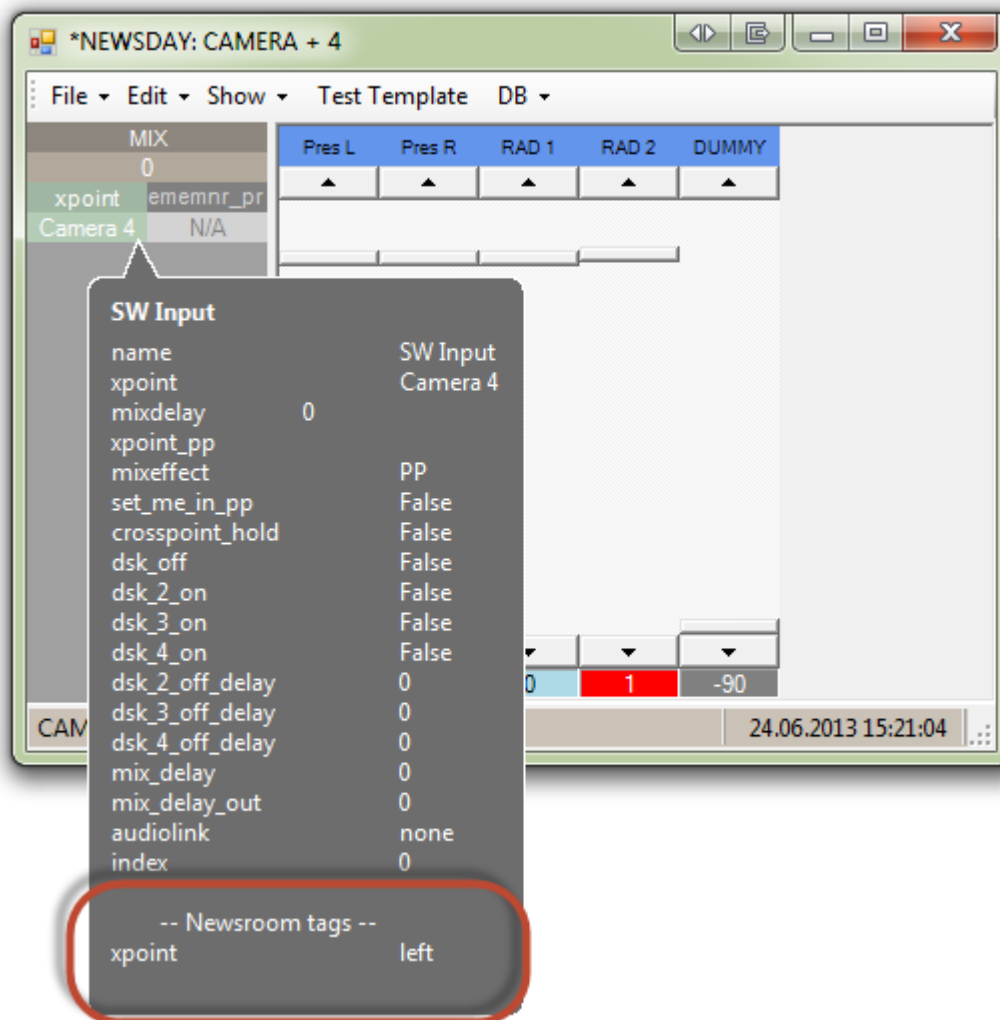
This means that Key 1 will be either the current value of a Newsroom tag named left within the same story or "Camera 5" if no such Newsroom tag exists.

Timeline Example

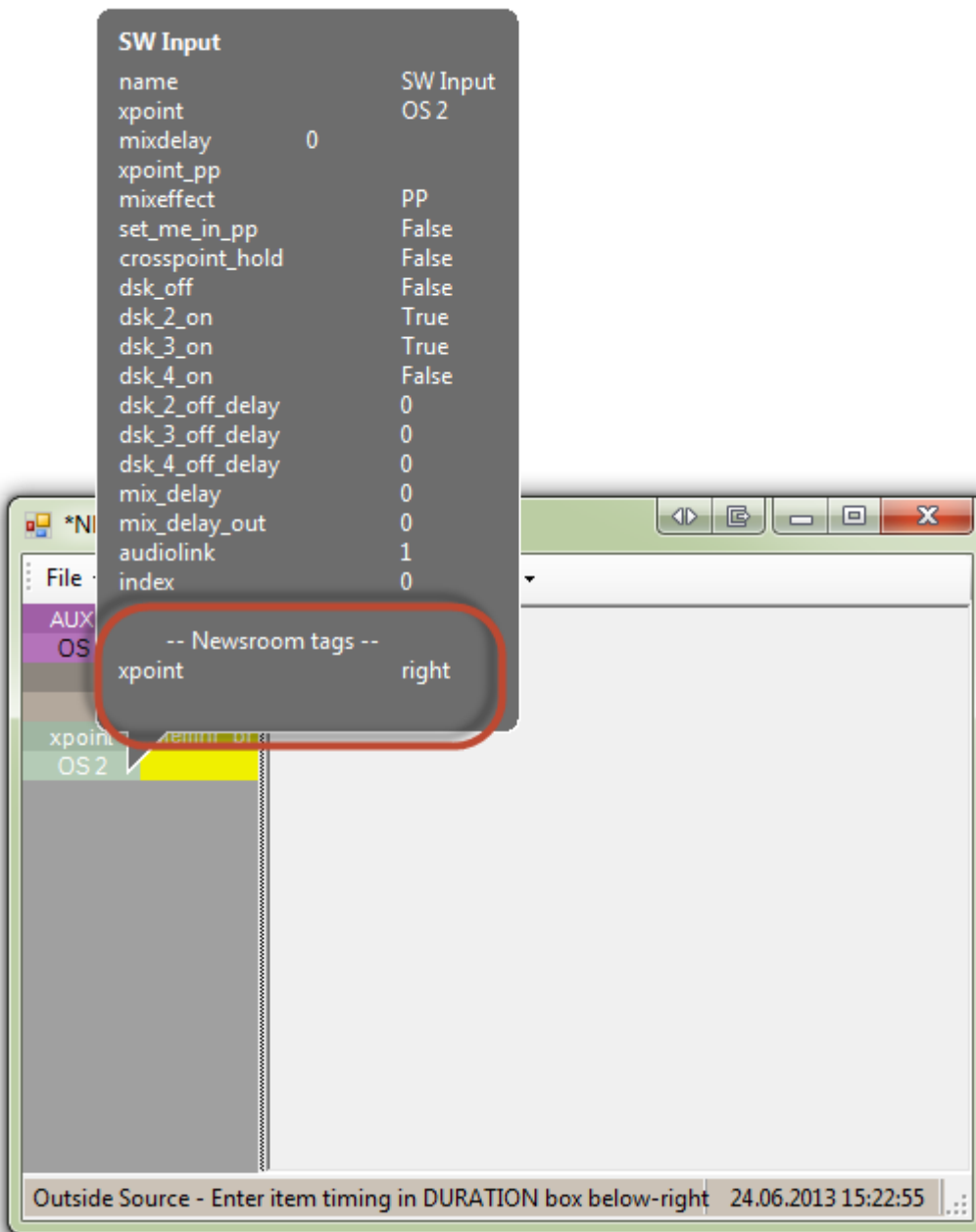
<p>Camera+4: left = Camera 4</p>	<p>LIVE+2: right = OS 2</p>	<p>DVE+DTL2BOX: left = {left} right={right}</p>
---	--	--

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"

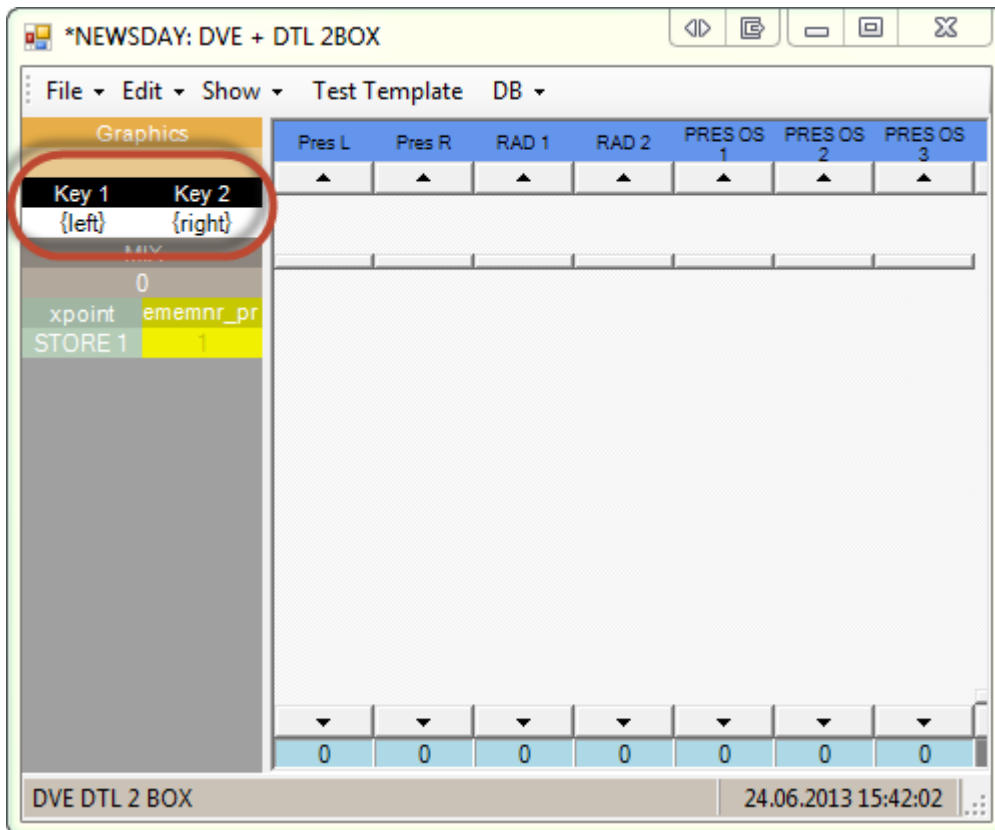


2. Live+2: Where xpoint is assigned to OS 2 with Newsroom tag = "right"



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.



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

11 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 devices must be configured within Viz Mosart.

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

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

This section presents the following device properties:

- [AV Automation Devices Vision Mixer](#)
- [AV Automation Devices Video Servers](#)
- [AV Automation Devices Graphics](#)
- [AV Automation Devices Audio](#)
- [AV Automation Devices Camera Robotics](#)
- [AV Automation Devices Light Control](#)
- [AV Automation Devices Router](#)
- [AV Automation Devices Subtitles](#)
- [AV Automation Devices GPIO](#)
- [AV Automation Devices Tally](#)
- [AV Automation Devices Virtual Set](#)
- [AV Automation Devices Weather](#)

- [AV Automation Devices Video Wall](#)
- [AV Automation Devices Integrated Engine](#)
- [AV Automation Devices General](#)

11.1 AV Automation Devices Vision Mixer

This dialog menu is accessed from the AV Automation menu bar with **Devices > Preferences > Mosart Device Properties.**

The screenshot shows the 'Mosart Device Properties' dialog box with the 'Vision Mixer' tab selected. The settings are as follows:

- Protocol: Carbonite (dropdown)
- Host videoswitcher: 192.168.0.123
- Port videoswitcher: 7788
- Auxillary videowall 1: 4
- Default Camera mix on vw: 4
- Mix Effect Videowall 1: Me 4 (dropdown)
- Min. Cue delay: 25 frames
- Min. Clip Mix delay: 10 frames
- FrameRate: 25 frames per second
- Virtual ME: PP (dropdown)
- Physical ME: ME1 (dropdown)
- ME2: ME2 (dropdown)
- ME3: ME3 (dropdown)
- ME4: ME4 (dropdown)
- Ini - EMEM: 99
- Reset dmem vw 1: 49
- DVE Forward EMEM: 97
- DVE Reverse EMEM: 98
- Effect Memory: PP (dropdown)
- DSKNr: 1 (dropdown)
- AUX Clicked preview: 1
- Use DSK 1 on/off
- Use DSK 2 on/off
- Use DSK 3 on/off
- Use DSK 4 on/off
- Control DSK from Overlay Interface
- Overlay DirectTake control:
 - Active:
 - Inactive:
- AsRun log cross points in stand by

Settings that are required for Viz Mosart to effectively control the Video Mixer, are outlined in the list below:

- **Protocol (check box):** Check the box to enable video switcher control.
- **Protocol:** Select a protocol supported by your switcher:
 - 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
- 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
- Carbonite
- 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. The GVG CPL driver uses UDP protocol and it has an additional client port property used for receiving callbacks.
- **Auxiliary videowall 1:** This is the auxiliary output from the video switcher to the video wall.
- **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.
-

11.2 AV Automation Devices 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.

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

- **AVAutomation > Devices > Preferences.**
From the Mosart Device Properties menu, select the **Video Servers** tab, to display the

configuration menu below.



- The *left* side presents details of the physical servers, 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. 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 [Working with Mosart Port Configuration](#).

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

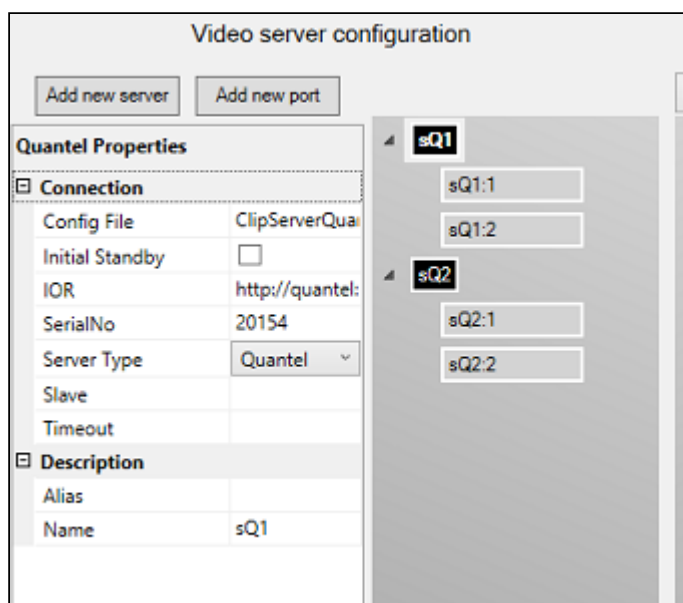
- [Working with Video Server Configurations](#)
 - [Video Server Properties](#)
 - [Common Settings](#)
 - [AirSpeed MultiStream](#)
 - [Grass Valley K2](#)
 - [Nexio](#)
 - [MVCP](#)
 - [Omneon](#)
 - [Quantel](#)
 - [VDCP](#)
 - [Viz Engine](#)

- Vizrt-ips
 - Video Server Port Properties
- Working with Servers and Ports
 - To Add a Server
 - To Remove a Server
 - To Add a Port
 - To Remove a Port
- Mosart Port Configurations
 - Virtual Server Ports
 - Working with Mosart Port Configurations
 - To Add a Virtual Port Group
 - To Remove a Virtual Port Group
 - To Add a Virtual Port
 - To Remove a Virtual Port
 - Working with Salvos
 - To Connect a Server Port to a Virtual Port
 - To Clear Server Links in a Virtual Port Node
 - To Add a Salvo
 - To Remove a Salvo
 - To Modify a Salvo

11.2.1 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.



⚠ Note: Only compatible combinations of MediaAdmin and AVA drivers can be defined. Please contact Vizrt Support for further details.

The **Name** property links the selected video server to a MediaAdmin properties line.

For an explanation of the various configuration properties, please refer to the specific video server model, as presented in [Video Server Properties](#) below.

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.

⚠ EVSXedio: This value 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, for display in **AV Automation**.

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

AirSpace Properties	
☐ Connection	
Initial Standby	<input type="checkbox"/>
Server	
Server Type	AirSpace ▾
☐ Description	
Alias	Alias
Name	Name

AirSpeed MultiStream

AirSpeed MultiStream Properties	
[-] Connection	
BasePort	
Initial Standby	<input type="checkbox"/>
Server	
Server Type	AirSpeedM ▾
[-] Description	
Alias	Alias
Name	Name

- **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 need to be open for Mosart to be able to connect

Manus admin uses a separate port when it connects: 49171 (although a range is OK. Use 49168-49172)

Grass Valley K2 Properties	
[-] Connection	
Clip Bin	
Domain	
Initial Standby	<input type="checkbox"/>
Pass	
Server	
Server Type	GrassValle ▾
Suite	
User	
Volume	
[-] Description	
Alias	Alias
Name	Name

- **Clip Bin:** Defines the location on the server that holds the clips.
- **Domain:** Define the name of the Domain the user is part of.
- **Pass:** Defines the password for logging into the server.
- **Suite:** The suite name on the **K2** video server that plays out the clips.

- **User:** Username to log into the server.
- **Volume:** The volume letter where the clips are stored on the **K2** server.

After configuration, restart MediaAdministrator and AVautomation, and ensure that the GUI is connected to the K2 server(s).

Verify that all lights regarding videoservers go green.

✔ **Tip:** If you do not get green lights, and you see the following message 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 in Grass Valley.
 More info here: http://www.gvgdevelopers.com/concrete/apis/appserver_api/windows/

Nexio

HarrisNexio Properties	
☐ Connection	
Initial Standby	<input type="checkbox"/>
Port	1
Server	
Server Type	HarrisNexi ▾
☐ Description	
Alias	Alias
Name	Name

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

MVCP

MVCP Properties	
☐ Connection	
Initial Standby	<input type="checkbox"/>
Port	5558
Server	
Server Type	MVCP ▾
☐ Description	
Alias	Alias
Name	Name

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

Omneon

Omneon Properties	
☐ Connection	
Directory	
ExtList	
Initial Standby	<input type="checkbox"/>
Server	
Server Type	Omneon ▾
☐ Description	
Alias	Alias
Name	Name

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

Quantel Properties	
☐ Connection	
Config File	
Initial Standby	<input type="checkbox"/>
IOR	
SerialNo	
Server Type	Quantel ▾
Slave	
Timeout	
☐ Description	
Alias	Alias
Name	Name

- **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** payout 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.

VDCP

VDCP Properties	
[-] Connection	
COM Port	1
Initial Standby	<input type="checkbox"/>
Server Type	VDCP ▾
[-] Description	
Alias	Alias
Name	Name

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

VDCP(tcp) Properties	
[-] Connection	
Port	1
Initial Standby	<input type="checkbox"/>
Server	
Server Type	VDCPtcp ▾
[-] Description	
Alias	Alias
Name	Name

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

Viz Engine

VizEngine Properties	
☐ Connection	
Port	1
Initial Standby	<input type="checkbox"/>
Server	
Server Type	VizEngine ▾
☐ Description	
Alias	Alias
Name	Name

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

Vizrt-ips

vizrt-ips Properties	
☐ Connection	
Directory	D:\Sessions\De
Initial Standby	<input type="checkbox"/>
Port	5951
Server	192.168.1.1
Server Type	vizrt-ips ▾
☐ Description	
Alias	
Name	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

Server Port	
<input checked="" type="checkbox"/> AvConfig	
Alias	
Audio Channel 1	SVR A ▾
Audio Channel 2	(empty) ▾
CrossPoint	SVR A ▾
Video port	1
<input checked="" type="checkbox"/> Description	
Name	Quantel:1

- **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.
- **Name**

11.2.2 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 the **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**.

11.2.3 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.
 - **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 direct take template number that should be taken when switching to this virtual server salvo.
 - **On switch from:** The direct take template number that should be taken when switching from this virtual server salvo.
 - **On mirror switch to backup:** The direct take template number that should be taken when switching to a backup server using this virtual server salvo.
 - **On mirror switch from backup:** The direct take template number 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 the **Media Router** is not used, this value can be left to the default 0.

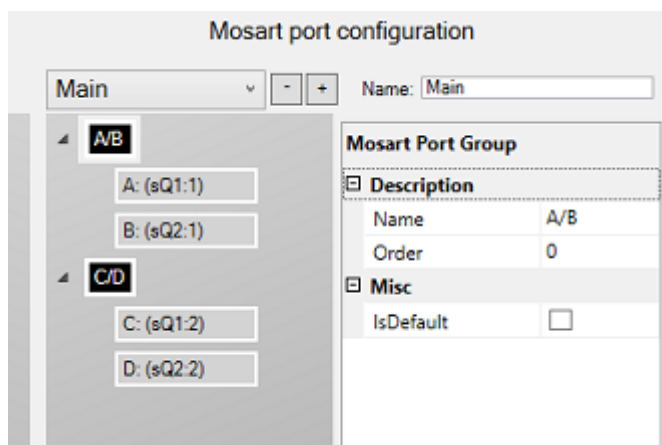
Mosart Port	
AvConfig	
Audio Channel 1	(empty) v
Audio Channel 2	(empty) v
CrossPoint	(empty) v
Description	
Name	A
Server ports	
Main	Quantel:1
Mirror	n/a

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*
- **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 [Working with Salvos](#) below).



To Add a Virtual Port Group

1. 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.

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).

To connect two server ports to one virtual port, causing mirroring:

1. Select a server port and drag it to the virtual port that you use to represent the port in the selected salvo.
2. Select another server port and drag it to the same virtual port.

⚠ Note: Only two ports can be mirrored.

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.

11.3 AV Automation Devices 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
- Shows Tab
- Deko Settings
- XPression Settings
- Orad Settings
- Pixel Power Settings
- Chyron Settings

11.3.1 Vizrt Settings

Vision Mixer | Video Servers | Graphics | Audio | Camera Robotics | Light Control | Router | Subtitles | GPI/O | Tally | Virtual Set | W

General | Shows

Use MSE Backup

Engines

	Host	Port	Handler name	MSE Host	Encoding	Clicked Preview	Auto Preview	Preview engine
<input checked="" type="checkbox"/> 1	10.211.114.5:6100	6100	viz_background_a	10.211.114.5	UTF-8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1
<input type="checkbox"/> 2		6100	viz_background_b	localhost	UTF-8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2
<input type="checkbox"/> 3		6100	viz_background_c	localhost	UTF-8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	3
<input type="checkbox"/> 4		6100	viz_background_d	localhost	UTF-8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	4
<input type="checkbox"/> 5		6100	viz_background_e	localhost	UTF-8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	5

Database

Configure MSE database settings

User:
 Password:
 Datasource:
 Schema:

Viz Content Pilot

Connect to VCP client host
 Port:

Miscellaneous

Mos Playlist:
 Path Externals:
 Local Media Sequencer Engine Start MSE if not running


 Use Concept Override
 Translate Cue for Transition Logic elements:
 No translate Take Take out
 Take out full screen graphics Apply Channel Name to Elements

General Tab


- **Use MSE Backup:** Uses Media Sequencer backup engine when starting the system when checked.

- **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.

- **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 ";"

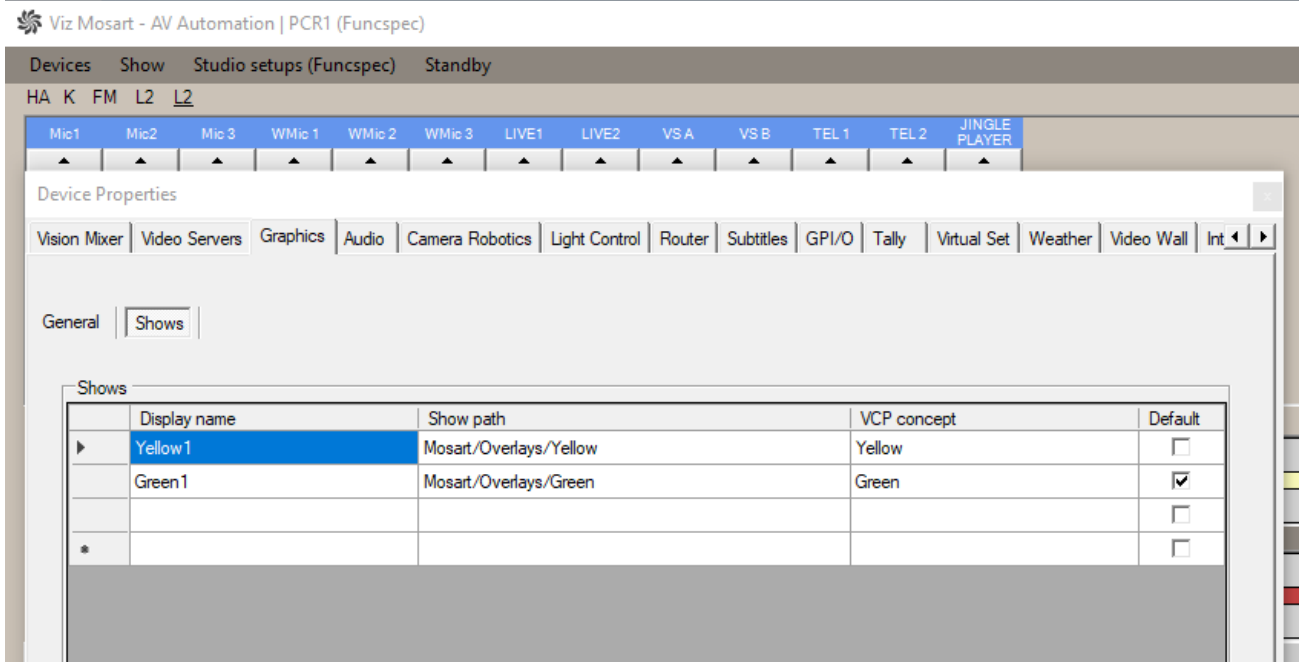
- **Encoding:** Font encoding for the Viz Engine. Default: UTF-8.
 - **Clicked Preview:** Enables the “click on full screen graphic element” to be taken on the preview engine.
 - **Auto Preview:** Enable to automatically take all full screen graphic elements in the preview engine.
 - **Preview engine:** 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**
 - **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.
Please refer to the ‘Shows’ fields in the [Engines Setup Tab](#).
-  **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

i **Info:** The playlist name used here cannot be the same as that used in [Trio Interface](#) > Engines Setup Tab > Playlist > Playlist name.

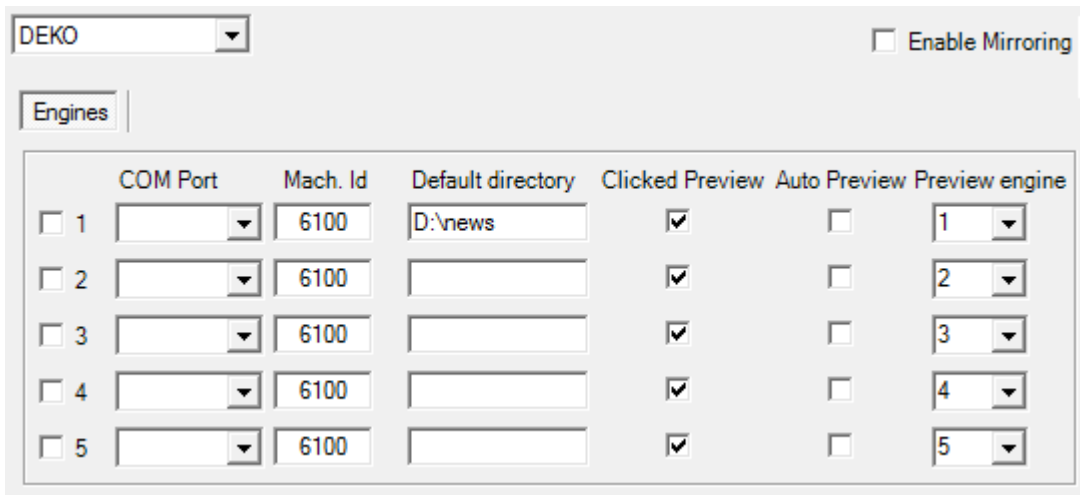
- **Path Externals:** Internal Media Sequencer path to the location for handling database elements. Default: external/pilotdb/elements
- **Local Media Sequencer Engine:** Path to the launcher.exe in the Media Sequencer program files folder (only if Media Sequencer is running as a console application 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 Sequencer 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.

Shows Tab



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

11.3.2 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.

- **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.

11.3.3 XPression Settings

XPRESSION Enable Mirroring

Engines

	Host	Channel	Channel list	Default directory	Clicked Preview	Auto Preview	Preview engine
<input type="checkbox"/> 1		6100	viz_background_a	D:\news	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1
<input type="checkbox"/> 2		6100	viz_background_b		<input checked="" type="checkbox"/>	<input type="checkbox"/>	2
<input type="checkbox"/> 3		6100	viz_background_c		<input checked="" type="checkbox"/>	<input type="checkbox"/>	3
<input type="checkbox"/> 4		6100	viz_background_d		<input checked="" type="checkbox"/>	<input type="checkbox"/>	4
<input type="checkbox"/> 5		6100	viz_background_e		<input checked="" type="checkbox"/>	<input type="checkbox"/>	5

- **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.

11.3.4 Orad Settings

ORAD Enable Mirroring

Engines

	Host	Channel	Channel list	Default directory	Clicked Preview	Auto Preview	Preview engine
<input type="checkbox"/> 1		6100	viz_background_a	D:\news	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1
<input type="checkbox"/> 2		6100	viz_background_b		<input checked="" type="checkbox"/>	<input type="checkbox"/>	2
<input type="checkbox"/> 3		6100	viz_background_c		<input checked="" type="checkbox"/>	<input type="checkbox"/>	3
<input type="checkbox"/> 4		6100	viz_background_d		<input checked="" type="checkbox"/>	<input type="checkbox"/>	4
<input type="checkbox"/> 5		6100	viz_background_e		<input checked="" type="checkbox"/>	<input type="checkbox"/>	5

- **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.

11.3.5 Pixel Power Settings

PIXEL POWER

Engines

	Host	Port	System (roChannel)	Default directory	Clicked Preview	Auto Preview	Preview engine
<input type="checkbox"/>	1	6100	viz_background_a	D:\news	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1
<input type="checkbox"/>	2	6100	viz_background_b		<input checked="" type="checkbox"/>	<input type="checkbox"/>	2
<input type="checkbox"/>	3	6100	viz_background_c		<input checked="" type="checkbox"/>	<input type="checkbox"/>	3
<input type="checkbox"/>	4	6100	viz_background_d		<input checked="" type="checkbox"/>	<input type="checkbox"/>	4
<input type="checkbox"/>	5	6100	viz_background_e		<input checked="" type="checkbox"/>	<input type="checkbox"/>	5

- **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.

11.3.6 Chyron Settings

CHYRON Enable Mirroring

Engines

	Host	Channel	Channel list	Default directory	Clicked Preview	Auto Preview	Preview engine
<input type="checkbox"/> 1	<input type="text"/>	6100	viz_background_a	D:\news	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1
<input type="checkbox"/> 2	<input type="text"/>	6100	viz_background_b	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2
<input type="checkbox"/> 3	<input type="text"/>	6100	viz_background_c	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	3
<input type="checkbox"/> 4	<input type="text"/>	6100	viz_background_d	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	4
<input type="checkbox"/> 5	<input type="text"/>	6100	viz_background_e	<input type="text"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	5

- **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.

11.4 AV Automation Devices Audio

Viz Mosart is capable of connecting a variety of Audio Mixers available from several different manufacturers.

Select the correct Audio Mixer from the menu for your broadcast environment. AV Automation handles the following:

- [Audio Mixer](#), a backup MIDI Audio Mixer.
- [Loudness Control](#) from Junger.

11.4.1 Audio Mixer

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

Audio mixer	Protocol	Supported hardware	Interface
Calrec	RAP		RS-422, TCP/IP (Artemis)
DHD	DHD RM4200-D v2.02		TCP/IP
EUPHONIX	RAP		COM
LAWO RMNOPL	RemoteMNOPL	Lawo mc ²	TCP/IP
LAWO ZIRKON	RAS, Lawo Zirkon subset		RS-422 or TCP/IP
MIDIMIXER	Sanford.Multimedia		MIDI

Audio mixer	Protocol	Supported hardware	Interface
SSL	Light Broadcast Automation protocol, SSL Broadcast Automation Control Specification - v1.0		RS-422 or TCP/IP
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 ip series 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).
- **Backup hostname and port (Lawo):** Hostname (or IP address) and port to the backup audio mixer console.
- **Set user labels on console:** When selected, user labels on the Lawo or Vista console are overwritten.
- **MidiCh (Yamaha):** MIDI channel assigned to the audio mixer.
- **Midi Output port (Yamaha):** System MIDI 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.
- **Use sound player:** Enables connection to the Audio Player.
- **Host:** Hostname or IP address to the computer running the Audio Player.
- **Default directory:** Directory for the audio files (on the Audio Player's PC).

AV Automation Audio Mixer Compatibility Table

CAL REC	D H D	LAWO RMNO PL	LAWO ZIR KON	S S L	STAG ETEC	STUD ER 3000	Wheat Stone	YAM AHA O2	STUDE R VISTA	EUP HON IX	vizrt-ips
Serial	x	x	x								
TCP/IP	x	x	x								x
MidiCh	x										
Midiport	x										
Hostname	x	x	x	x	x	x	x	x	x		
Port	x	x	x	x	x	x	x	x	x	x	
Backuphost name	x										
BackupPort	x										
Faderinputoffset	x										
Useemmerprotocol	x										
Comptaudio mixer	x	x	x	x	x	x					

11.4.2 Loudness Control

Audio mixer | Sound player | Loudness controller

Use loudness controller

JÜNGER T*AP

Triggers/levels

- 1
- 2
- 3
- 4
- 5

Channels	Host:port
1-4	
5-8	
9-12	
13-16	
17-20	
21-24	

- **Use loudness controller:** Check the box to enable loudness control.
- **Loudness controller type:** JÜNGER 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.

See Also

- [Audio Player](#)

11.5 AV Automation Devices Camera Robotics

11.5.1 Configuring a Robotic Camera

Use Robotic Cameras

	Robotic Camera	Host	Port	Cameras
<input checked="" type="checkbox"/>	VINTEN 200	10.209.101.102	5000	1-5
<input type="checkbox"/>	TELEMETRICS			

- **Use Robotic Cameras:** Check to enable the robotic camera control functionality.
- **Robotic Camera:** CAMEROBOT, CINNEO, FX-MOTION, PANASONIC, SHOTOKU TR-T, SHOTOKU TR-T IP, TECHNODOLLY, TELEMETRICS, VINTEN 200, ELECTRIC FRIENDS.
- **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.
- **Com port:** For serial communication with the RCC server.

⚠ Note: When using PANASONIC 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. List members are separated by commas, or by hyphens indicating ranges.

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

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*, *CAMERA CONTROL*.

The sequence is:

1. The Camerobot system supports shot variations dependent on current anchor information from NCS.
2. Viz Mosart retrieves *presenter* and *pos* tuples from OpenMedia and communicates them to Camerobot in the value **MATRIX_SET_PRESENTER**.
3. OpenMedia transmits the name of the presenter with the story via Viz Mosart to Camerobot.

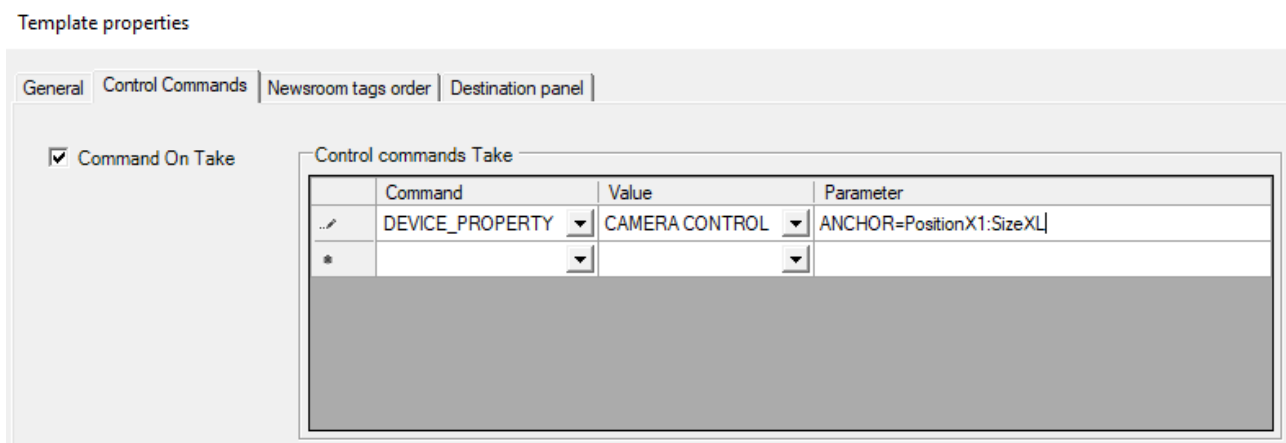
4. Camerobot uses a lookup table to assign position data to individual presenter names.
5. If Time =1 (>0), Viz Mosart sends MATRIX_MOVE_MPOS command to Camerobot, while if Time = 0 Viz Mosart sends MATRIX_CUT_MPOS command Camerobot.
6. The value received from OpenMedia, would have (see **Examples** below) the following syntax: [pos1:]presenter1[, [pos2:]presenter2] [, [pos3:]presenter3]...

The position and presenter is given as a tuple divided by ':'.
 Multiple position-presenter tuples are possible separated with ','.
 If no position is given, '1' is used as default position.

Important Setting

When creating the camera templates for this, the **ANCHOR=** is necessary for the functionality to work. The image below displays the configuration from a customer using this feature.

In template properties; use Command: DEVICE_PROPERTY → Value: CAMERA CONTROL → Parameter: ANCHOR=xxx



Examples

- David
 - Presenter1=David, Position1=1 (default)
- 2:David
 - Presenter1=David, Position1 = 2
- David,2:Mark
 - Presenter1=David, Position1=1 (default)
 - Presenter2=Mark, Position2=2
- 2:David,3:Mark
 - Presenter1=David, Position1=2
 - Presenter2=Mark, Position2=3

⚠ Note: The position is a *string* and does not have to be a number, as in the examples above.

MATRIX_SET_PRESENTER

Presenter(s) can be set whenever suitable. The presenter set influences the move selection (if a move can be found).

Warning: Recalculation of suitable moves will only be done if the robot has moved at least once after setting *presenters*.

It is possible to set several presenters at once (for different presenter-positions each). Just repeat the *presenter* tag.

```
<cmd id="#int">
  <type>MATRIX_SET_PRESENTER</type>
  <presenter pos="#modpos">#pres_string</presenter>
  [begin optional]
  <presenter pos="#modpos">#pres_string</presenter>
  ...
  [end optional]
</cmd>
```

type: Type of the command. Here it is SET_PRESENTER.

presenter: Name and position of the presenter.

pres_string: A string that identifies presenter.

pos: A string that defines the presenter position (*#modpos*).

This allows different behaviour, if several presenters are 'in use'.

Response from Interface:

```
<ack id="#int">
  <presenter pos="#modpos">SYS_ACK|SYS_POS_UNKNOWN|SYS_PRESENTER_UNKNOWN</presenter>
  [begin optional]
  <presenter pos="#modpos">SYS_ACK|SYS_POS_UNKNOWN|SYS_PRESENTER_UNKNOWN</presenter>
  ...
  [end optional]
</ack>
```

Responses are generated in the following order:

- **SYS_PRESENTER_UNKNOWN:** The presenter is unknown. No loading will be done.
- **SYS_POS_UNKNOWN:** The presenter-position is unknown. No loading will be done.
- **SYS_ACK:** The command was successful.

11.6 AV Automation Devices Light Control

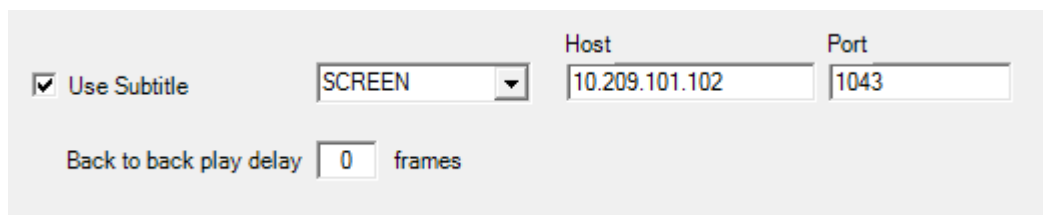
- **Light Control:** Enables the Light control through MIDI Show Control.
- **Midiport:** System MIDI port to communicate with the Light board.
- **Midi Show Control (MSCID):** The MIDI Show Control ID that is used for communicating with the Light board. Default: 0.
- **Custom command Prefix:** This 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.

11.7 AV Automation Devices Router

- **Use Router:** Check to enable router control and select protocol from the list.
- **Router type:** 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 (SW-P-08 TCP/IP and Vikinx TCP/IP).
- **Main Port:** IP port of the main router (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.

11.8 AV Automation Devices Subtitles



Use Subtitle SCREEN Host: 10.209.101.102 Port: 1043
 Back to back play delay: 0 frames

- **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.

11.9 AV Automation Devices GPIO

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

Adding GPI/O control to a device is performed in Viz Mosart AvAutomation.

- [DirectTakes via GPI](#)
 - [Syntax](#)
 - [Configuring](#)
 - [Verifying](#)
 - [To Trigger DirectTakes Templates via a GPI](#)
 - [Syntax for GPI 5-12](#)
- [Sample Configuration \(W&T Web-IO\)](#)
 - [GPI 0-4](#)
 - [GPI 5-11](#)
 - [Triggering a Mosart Template](#)
 - [Hardware Configuration](#)

	Host	Port	Password
<input type="checkbox"/> GPO (1-12)	10.209.101.102	49153	
<input type="checkbox"/> GPO (13-24)	10.209.101.102	49153	
<input type="checkbox"/> GPO (25-36)		49153	
<input type="checkbox"/> GPO (37 - 48)		49153	
<input type="checkbox"/> GPI 0 (Reload rundown)			
<input type="checkbox"/> GPI 1 (Start/continue rundown)			
<input type="checkbox"/> GPI 2 (Start rundown from top)			
<input type="checkbox"/> GPI 3 (Rehearsal mode OFF)			
<input type="checkbox"/> GPI 4 (Rehearsal mode ON)			
<input type="checkbox"/> GPI 5 (Fire Template)			
<input type="checkbox"/> GPI 6 (Fire Template)			
<input type="checkbox"/> GPI 7 (Fire Template)			
<input type="checkbox"/> GPI 8 (Fire Template)			
<input type="checkbox"/> GPI 9 (Fire Template)			
<input type="checkbox"/> GPI 10 (Fire Template)			
<input type="checkbox"/> GPI 11 (Fire Template)			
Min Time Between GPIs (ms)	300		

- **GPO (1-12):** Enables a WebIO box for the first 12 GPO when checked.
- **GPO (13-24):** Enables a WebIO box for the second 12 GPO when checked.
- **GPO (25-36):** Enables a WebIO box for the third 12 GPO when checked.
- **GPO (37-48):** Enables a WebIO box for the fourth 12 GPO when checked.
- **Host:** Hostname or IP address to the WebIO GPI/O box.
- **Port:** Port to communicate with the HTTP WebIO protocol.
- **Password:** Defines the password, if needed.
- **GPI 0 (check box):** Enables initialize Viz Mosart rundown when the external pulse is received.
- **GPI 1 (check box):** Enables start/continue the Viz Mosart timeline from an external pulse.
- **GPI 2 (check box):** Enables starting Viz Mosart rundown on the first story on the external pulse.
- **GPI 3 (check box):** Enables rehearsal mode off.
- **GPI 4 (check box):** Enables rehearsal mode on.
- **GPI 5-11 (check boxes/custom):** Enables firing of the template or command given in the text box.

11.9.1 DirectTakes via GPI

As taking a template using GPI would add the corresponding template to the timeline, which is not always wanted, you can instead use a DirectTake control command.

Syntax

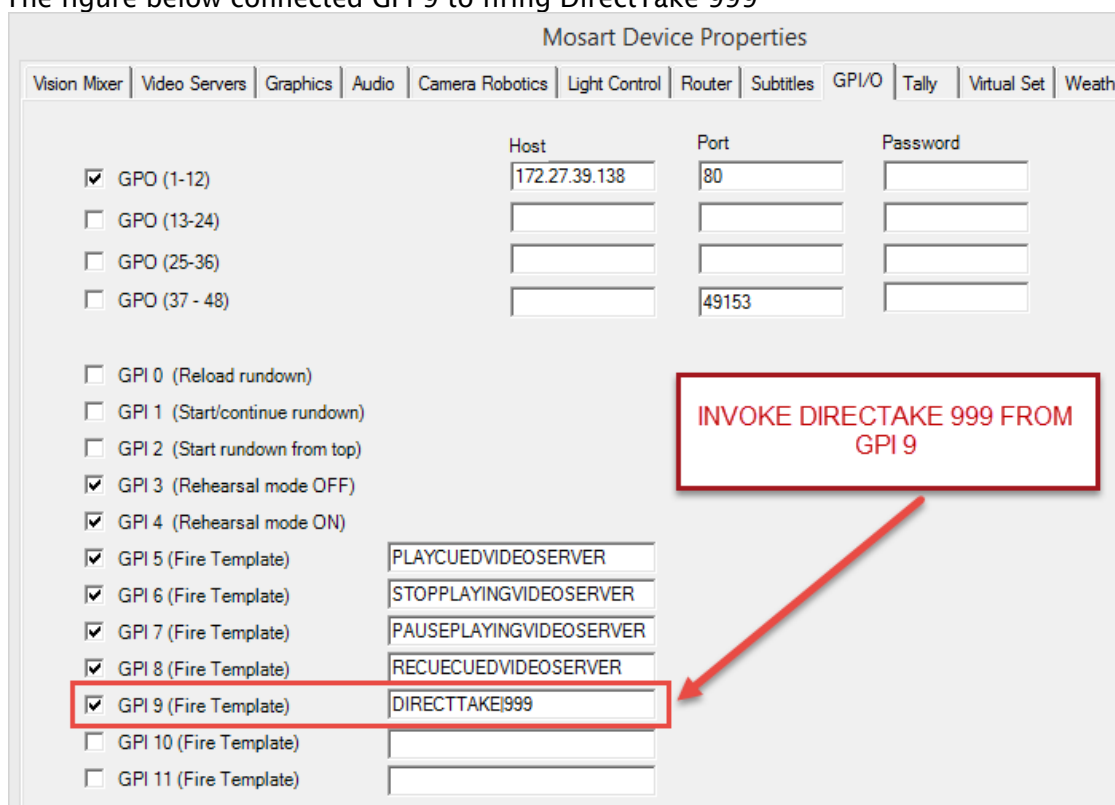
DirectTake|<directtake number>

Example: DirectTake|99

Triggers DirectTake number 99

Configuring

The figure below connected GPI 9 to firing DirectTake 999



Verifying

- Assign a DirectTake to a GPI between 5 and 11
- Restart AvA
- Trigger this GPI

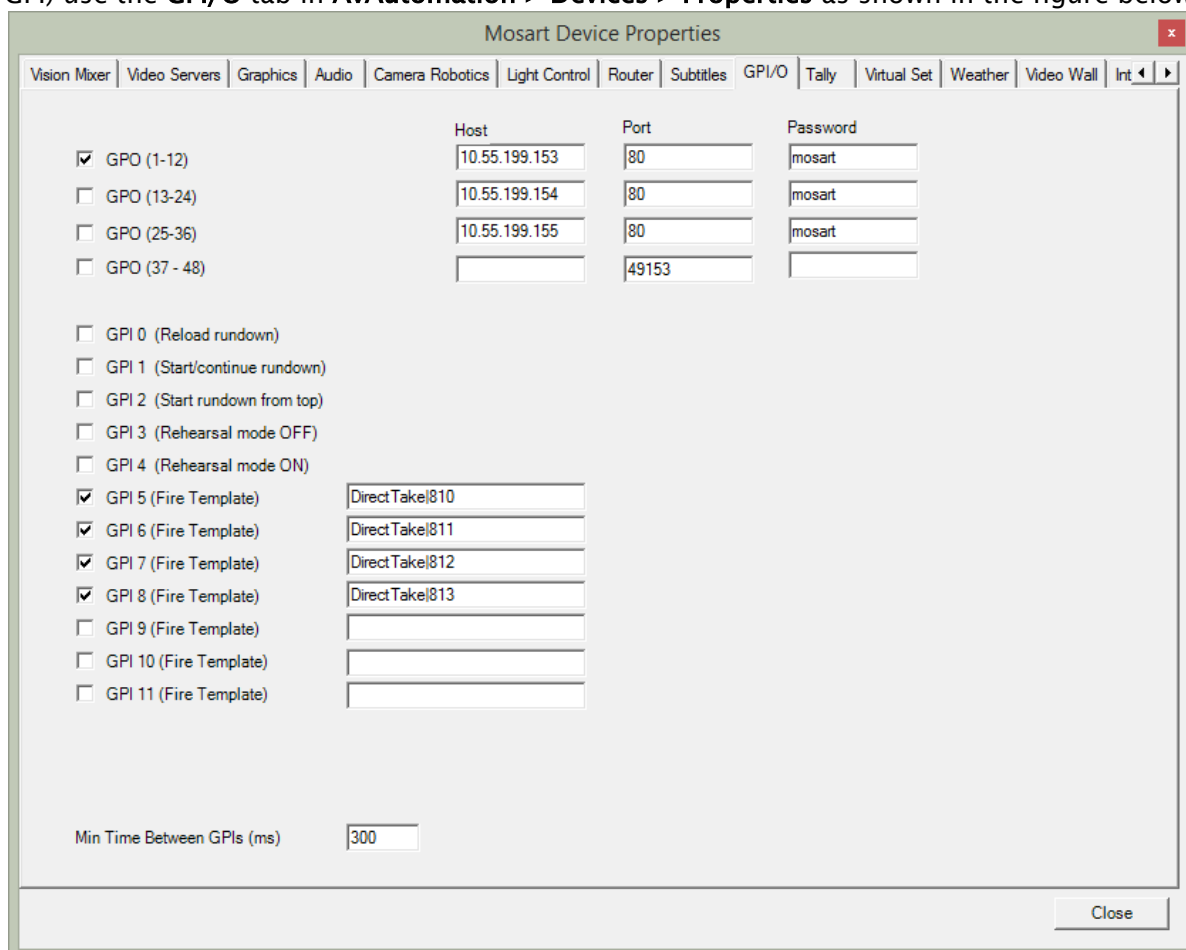
This can be done from within AvAutomation by pressing **SHIFT+CTRL+<no>** where <no> is the GPI number + 1.

For example, **SHIFT+CTRL+0** will invoke GPI 9

- Verify that the corresponding GPI is executed
- Directtakes are shown in the AvAutomation Log window (lower left) when taken.

To Trigger DirectTakes Templates via a GPI

- To trigger a DirectTake (which can then further trigger multiple control commands from a GPI) use the **GPI/O** tab in **AvAutomation > Devices > Properties** as shown in the figure below.



Syntax for GPI 5-12

DirectTake|<directtake number>

Note: A GPI/GPO device must first be configured. The configuration of GPI/GPO devices is done in [AV Automation Devices GPIO](#).

✓ Testing GPI signals without the use of GPI

If a GPI/GPO device is configured, as shown in the figure above, any GPI signals 0-9 may be simulated using the keyboard sequence **CTRL+SHIFT+<number>** inside AvAutomation to trigger a corresponding GPI.

This is an easy way to test the GPI configuration.

For example: CTRL+SHIFT+6 will trigger GPI 5

11.9.2 Sample Configuration (W&T Web-IO)

GPI is enabled by attaching a device capable of sending GPI signals to Viz Mosart. The configuration below shows configuration of a W&T Web-IO device capable of 24 GPI/GPO connections. This configuration uses at a maximum 24 GPO signals. A maximum of 12 GPI signals can be used to control Viz Mosart.

GPI 0-4

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

The signals are as follows:

GPI 0 - Reload rundown, Will trigger a rundown reload event. I.e. the timeline will be stopped and the current rundown will be reloaded from the NCS cache.

GPI 1 - Start/continue rundown. Will trigger the TAKE-NEXT event and will start the timeline or advance to the next timeline element

GPI 2 - Start rundown from top. Will start the rundown from the first story

GPI 3 - Disable rehearsal mode

GPI 4 - Enables rehearsal mode

GPI 5-11

The next 7 GPI signals may be custom assigned to either a Mosart Control Command or to fire an arbitrary template within the currently active template set.

Triggering Mosart Control Commands

To trigger a Mosart Control Command enter a valid command in the text field associated with the GPI signal. Supported commands:

ClearForegroundGraphics - Removes all CGs

SkipNextStory - Skips next story **item**

UnskipNextStory - Unskips next story **item**

TakeLastGraphicsOut - Removes last CG

TakeManualGraphicsOut - 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

Triggering a Mosart Template

This can be used to trigger a Mosart template within the currently active template set

Syntax: [TemplateType] | [Variant]

- Example: Camera|1

Triggering a DirectTake

This can be used to trigger a DirectTake

Syntax: DirectTake|<directtake number>


- Example: DirectTake|99
- Triggers DirectTake number 99

Hardware Configuration

For this example, you must configure the in- and out-ports as required for the GPIO system as per their own specifications.

The Web-IO physical device can be configured via a browser pointed to the box' IP address.

- To get into the configuration menus select **Config > Login > Administrator login** from the start page.

 **Note:** Configuration changes will only be saved after clicking the **Logout** button.

Mosart IAT/SAT

- For testing from Mosart, the new configuration is either directly loaded, or inspected and used for setting the inport 0 manually. Inspection and comparison may be necessary should the load fail. If necessary reset the box to factory settings before using the configuration below.
- You only need to set the port minimum pulse filter to 100ms and the input type to positive pulse (three separate settings). These settings are found under the **inport** nodes in the menu tree.
- If you use port 0 it becomes very easy to test Mosart, as this is hardwired to the **Reload Rundown** function.

Preparation of test circuit

- This requires two pieces of wire, ~20cm, stripped (and preferably tinned) at the end.
- To set up the box for generating an input test pulse, interconnect the terminal strip ground connector for power supply ground (the outermost VCC connector) with the outermost GND connector, across the box. This establishes a common ground between power supply side and the input logic circuits.

⚠ Note: It is not possible to simply take the output of one output port and feeding it straight back to the input side, as the outputs are meant for relay circuits (“current drivers” - very low voltage). The 12 in 12 out version requires an external relay; The inputs requires 8V referred to GND to detect a pulse.

Testing GPIO

- To generate the input pulse, connect a piece of wire from the other VCC connector and briefly touch the connector for Inport 0 (the pulse must be >100ms).
- A green light beside the connector will briefly light up if successfully triggered. Given that AV Automation has GPIO enabled and is also enabled for Input 0, any loaded rundown will reload, clearly visible on the Mosart GUI.
- Configure GPIO with the file containing the working configuration (a [sample file is available internally](#) at Vizrt)

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.

11.10 AV Automation Devices Tally

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

11.11 AV Automation Devices Virtual Set

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

11.12 AV Automation Devices Weather

<input checked="" type="checkbox"/>	VIZ WEATHER			
MSE Hostname	MSE Port	Host Viz/Weather	Default concept	
localhost	8594	127.0.0.1	1	

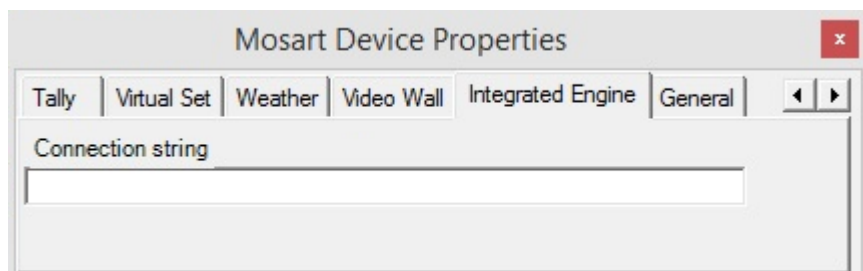
- **Use Weather:** Check to enable control of a weather control.
- **Weather controls:** VIZ WEATHER or WEATHER ONE
- **MSE Hostname (Viz Weather):** Hostname or IP address to the Media Sequencer controlling the Weather Control engine.
- **Hostname (Weather One):** Hostname or IP address of the Borealis 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.

11.13 AV Automation Devices Video Wall

<input checked="" type="checkbox"/>	Video wall type	SPYDER
Server address:	localhost	

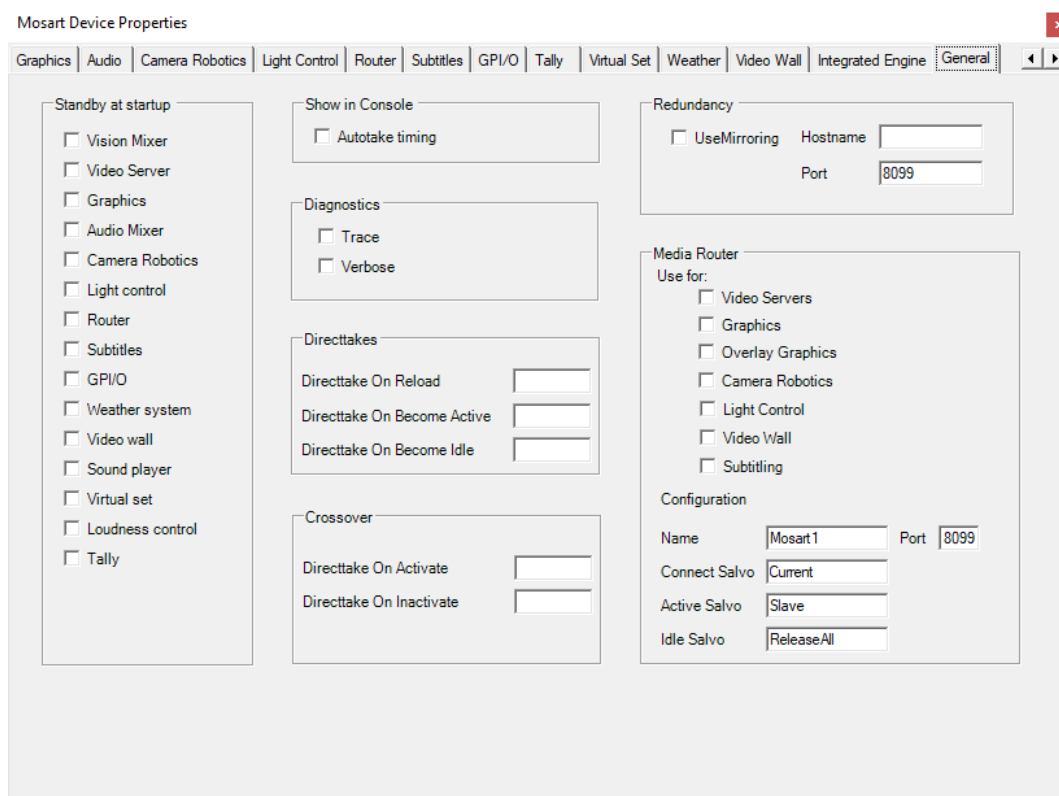
- **Video wall type:** Check to enable video wall control.
- **Video wall controls:** ENCORE, PANDORA, SPYDER, WATCHOUT (Version 5.2, DISPLAY CLUSTER PROTOCOL).
- **Port (Encore):** Serial port.
- **Bit rate (Encore):** Serial bit rate.
- **Server address (Pandora, Spyder, Watchout):** Hostname or IP 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.

11.14 AV Automation Devices Integrated Engine



- **Connection string:** For details, see the Viz Opus documentation.

11.15 AV Automation Devices 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 number of the directtake template to be fired upon re-loading the rundown.
 - **Directtake On Become Active:** The number of the directtake template to be fired when the server becomes active from idle, when switching between main and backup Viz Mosart servers.
 - **Directtake On Become Idle:** The number of the directtake 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 direct take number to be taken when this server goes active in crossover switch.
 - **Directtake on Inactivate:** The direct take number 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.
- **Media Router**
 - **Use for:** Select which controls should be used by the Media Router. (Also see [MMR Tab](#)).
 - **Name:** Media router's hostname or IP address.
 - **Port:** Media router's connection port number.
 - **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.

12 Viz Mosart Template Database

The Viz Mosart Template Database (TDB) is used to share template information among the following clients:

- Between galleries using Mosart automation. I.e. templates to be shared between multiple Mosart servers/galleries.
- Between Mosart main and backup servers in a redundancy setup.
- To NCS clients using Mosart ActiveX
- To NCS clients using custom ActiveX

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. The only main requirement to the database is that it is MySQL compliant. The template database may also be virtualized.

12.1 Mosart Template Definition

A Mosart template as stored in the template database has the following properties:


Property	Description
Gallery	The Mosart gallery for which this template is implemented. Note that a special global gallery is used for templates shared between all galleries. This is by default named 'SHARED'
Template set	The Mosart template set which the template belongs to. A template set is either shared or local. See definition below.
Template type	The Mosart template type. I.e. like CAMERA, CLIP, DVE, ...
Template variant	The Mosart template variant. I.e. like which camera (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 which is used to control systems using Mosart automation. This is the main part of the template and has to be implemented in every gallery.

12.1.1 Shared Template Set

A template set defined as shared is replicated among all galleries. This means that the template set shall be present in all galleries using the template database. A dedicated virtual gallery named “SHARED” is defined to be the owner of all shared template sets.

Local or Gallery-specific Template Set

A local template is said to be gallery specific meaning that it is totally independent upon the other galleries. Local template sets are owned by the corresponding gallery and not visible elsewhere.

 **Note:** The Mosart special template set *Directtakes* shall be defined as a shared template set.

Information Stored in the Template Database

Two of the main Mosart server application stores information in the template database:

- AvAutomation stores templates and AvConfig.
- ManusAdministrator stores NCS aliases as configured in the newsroomsettings dialog. Note that all other information in newsroomsettings.xml is **not** stored in the database.

Both applications will extract its information from the database and store the information locally in channeltemplates.xml, AvConfig.xml and newsroomsettings.xml respectively. Any existing files will be overwritten. Therefore prior connecting to the database the first time a backup of those files should be done.

This section continues with:

- [Installing the Viz Mosart Template Database](#)
- [Creating and Upgrading Mosart templatedb](#)
- [Configuring Viz Mosart Server](#)
- [Backup and Recovery](#)
- [Database Maintenance](#)
- [Viz Mosart Template Database Specification](#)

12.2 Installing The Viz Mosart Template Database

Both installing and creating the Viz Mosart Template database is outlined in this section.

12.2.1 Prerequisites

- A MySQL or equivalent database server (MariaDB, AWS Aurora MySQL) need to be installed and accessible from the Mosart server. The database server may either be dedicated for the Mosart automation servers or an existing in-house or cloud based server.

- A database user with credentials sufficient for creating and altering databases. This is mandatory for AvAutomation which is responsible for automatic upgrading of the template database.
- A local mysql client or tool that makes it possible to connect and run sql scripts towards the database.

MySQL Database Servers and Clients

- [Wampserver](#) - which is 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.
This solution is recommended when installing a dedicated template database for the Mosart automation servers. It allows remote access to the database via http.
- [MySQL server](#) - Use the MySQL Installer to install the native MySQL database server. Normally also install the [MySQL Workbench](#) as a client for managing and configuring the database server.
- [MySQL Workbench](#) - which 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

Verify the connection to the template database from the Mosart Server using a MySql client before configuring the Mosart Server. Typical issues here are related to firewall and credentials.

12.2.2 MySQL Users

Four types of users may be created for using the 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 AvAutomation
Database use (read/write)	Need to be able to do CRUD operations on a single database	Minimum rights for ManusAdmin
Database use (read only)	Only need to be able to do 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 for the Mosart database and the other for the Mosart ActiveX which only requires rights to perform *read* operations.

12.2.3 Initial Mosart Template Database

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. Currently version 1.0.5:

```
InitialMosartTemplateDatabase_1.0.5.sql
```

The sql file containing the initial Mosart Template Database shall be used to create the initial database on the database server.

12.2.4 Creating the Initial Template Database

Creating the initial template database can be done in one of the following 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 *Batch Files* directory as part of the Mosart Server installation.

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

12.3 Creating And Upgrading Mosart Templatedb

The database used for Template Database is named 'mosarttemplatedb'. The following procedure expects the Viz Mosart Server Suite has already been installed. This is due to running the batch file *RunDbScripts.cmd* found in the following location: *%ProgramFiles%\Mosart Medialab\Mosart Server\BatchFiles*.

Before continuing, ensure Viz Mosart has been installed (refer to Viz Mosart Administrator Guide, section on Installation) along with WampServer (refer to [Installing WampServer](#)).

This section contains the following topics:

- [Verify WampServer is running](#)
- [Running Database Scripts](#)
- [Running Scripts Automatically](#)
- [Running Scripts Manually](#)

12.3.1 Verify WampServer is running

Verify WampServer is running by performing these steps:

1. Check the WampServer icon is present in the system tray.
 - If not, start WampServer from the **Start** menu it should then appear green in the system tray.
 - If the WampServer icon is not green, click the icon and select **Start All Services**.

12.3.2 Running Database Scripts

The Viz Mosart Database can be created automatically through the *Db00Create.sql* script found in the following location, 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.

12.3.3 Running Scripts Automatically

Scripts are run through a batch file, *RunDbScripts.cmd*, found in the following location:

- %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. Additionally, the reference to the tool itself should also be verified.

The following steps should be performed:

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-e
xtra-file=mysql.ini < %%f
pause
```

2. Check if the stated folder (`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 has replaced root.
7. Save the file if you have changed it.

12.3.4 Running Scripts Manually


Run the scripts manually by following these steps:

1. Start phpMyAdmin, refer to Starting phpMyAdmin for more information.
2. Create the database, `mosarttemplatedb`, if not already done.
3. Upgrade the database using the scripts from the following location:
 - %ProgramFiles%\Mosart Medialab\Mosart Server\Sql
 - %ProgramFiles(x86)%\Mosart Medialab\Mosart Server\Sql

12.4 Configuring Viz Mosart Server


Both AvAutomation and ManusAdministrator can be configured to use the Viz Mosart Template Database.

- AvAutomation: For storing templates and AvConfig.
- ManusAdministrator: 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:

- AvAutomation: *channeltemplates.xml* and *AvConfig.xml*.
- ManusAdministrator: *newsroomsettings.xml*.

 Any existing files will be overwritten. Before connecting to the database for the first time, always back-up these files.

- [Connecting AvAutomation to the Mosart Template Database](#)
 - [Manual Verification of a AvAutomation Database Connection](#)
 - [Manual Import of channeltemplates.xml to the Template Database](#)
 - [Manual Import of AvConfig.xml to the Template Database](#)
 - [Manual Import of newsroomsettings.xml Content to the Template Database](#)
- [Viz Mosart Template Database and Viz Mosart ActiveX](#)

12.4.1 Connecting AvAutomation to the Mosart Template Database

Connecting AvAutomation 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.
- **ThisGallery**: The name of the gallery. Should be unique among galleries but identical for Mosart main and backup servers.
- **TemplateDbConnectionString**: Contains the database connection string to the database:

```
server=hostname;User Id=username;Password=password;database=mosarttemplatedb
```

- **TemplateDbDefaultInserter**: Identifies what entity is 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**: Set to *true* (default) if the AvAutomation shall automatically synchronize with the database so that any changes in the database are read

immediately by AvAutomation.

Recommend set to *true* unless intensive show design with frequent template changes occurs.

⚠ Only *idle* servers will update automatically. *Live* servers requires a manual operation, like opening the template editor, clicking on the status bar or restarting AvAutomation.

- **TemplateDbPollIntervalSeconds**: Number of seconds between polling the database for updates. Set to a higher value than the default (5 seconds) if multiple frequent template changes occurs.
- **TemplateDbEnableLocking** (deprecated): Leave default value (*true*).
- **TemplateDbProviderName**: Leave the default value for *MySQL* (MySQL.Data.MySqlClient).

For initial connection open the settings dialog and set the properties *UseTemplateDb*, *ThisGallery*, *TemplateDbConnectionString* and *TemplateDbDefaultInserter* then restart AvAutomation. 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.

Manual Verification of a AvAutomation Database Connection

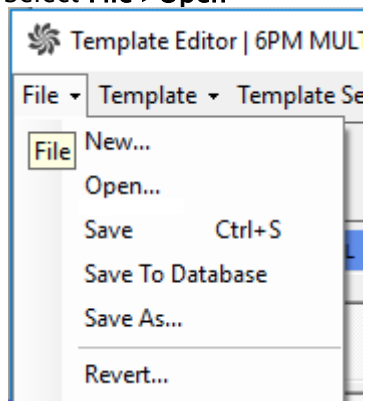
1. Do the initial connection steps as outlined in the former section.
2. Using any MySQL database client, verify the content of the *ga_gallery* table. This table shall now contain a single row where *ga_name=ThisGallery*.

Manual Import of channeltemplates.xml to the Template Database

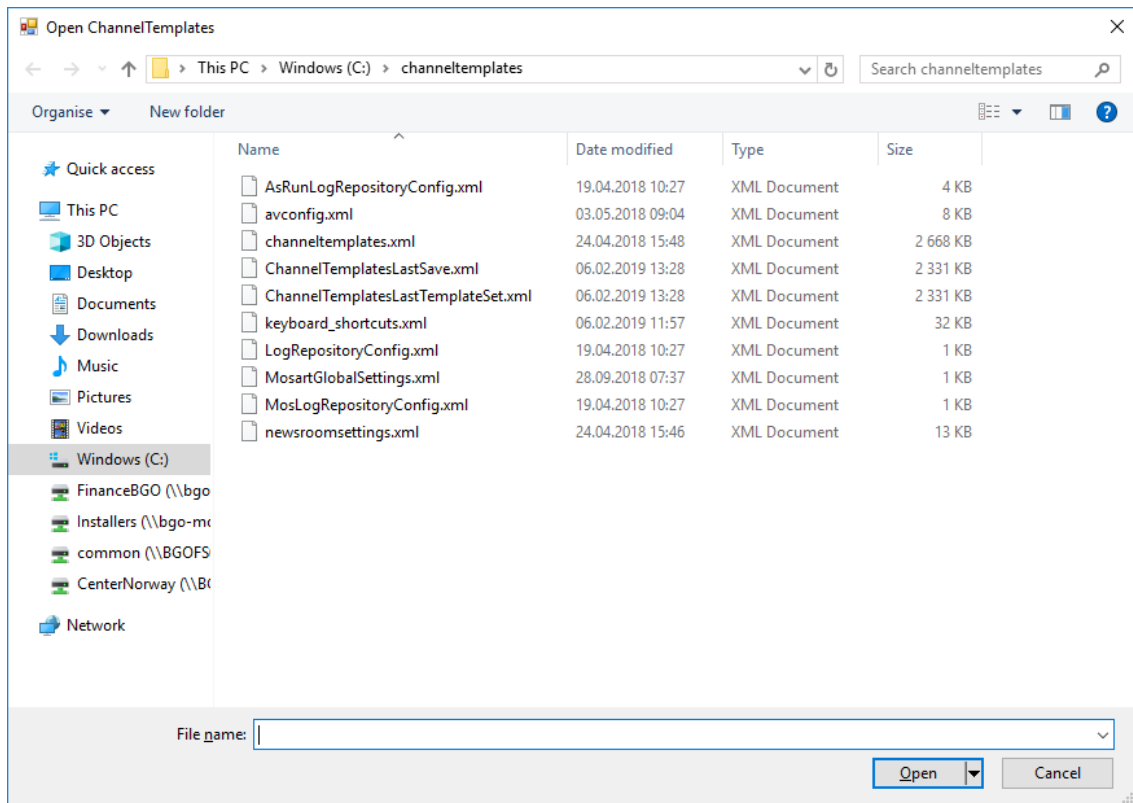
You can use a ready-made template set, stored as an XML-file. and manually import to the database. This is useful to populate the template database with the content from a *channeltemplates.xml* file. Either initially or as backup / transfer operation.

Import *channeltemplates.xml* file via the Template Editor in AvAutomation. 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**.



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 AvAutomation. 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.

Connecting ManusAdministrator to the Mosart Template Database

The only information stored in the database from ManusAdministrator 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 ManusAdministrator (unlike *ThisGallery* for AvAutomation)

Connecting ManusAdministrator 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)

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 ManusAdministrator, 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*.

12.4.2 Viz Mosart Template Database and Viz Mosart ActiveX

The Viz Mosart ActiveX is used to insert Viz Mosart template information into the NRCS script.

- 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.

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.

For more information, see [Connecting the ActiveX to Viz Mosart Server](#).

12.5 Backup And Recovery

Since the Viz Mosart Template Database is stored in a MySQL database, any backup and recovering strategies offered by the database vendors apply. Also 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.

12.5.1 Using mysqldump for backups

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
```

12.5.2 Using mysql for restore

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
```

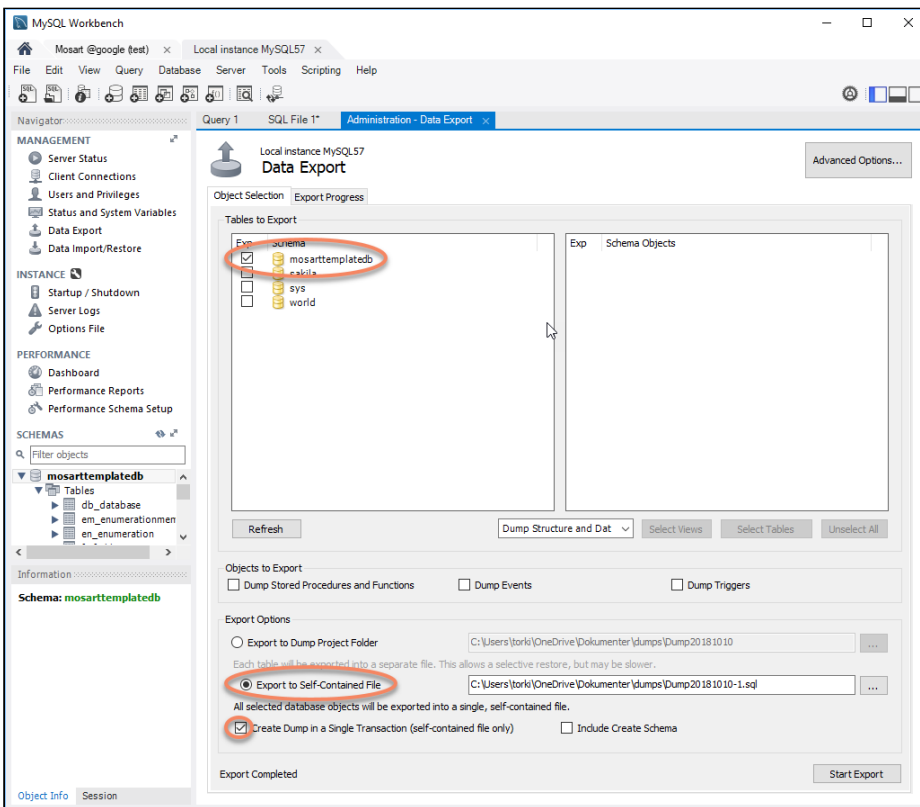
12.5.3 Backup and restore from MySQL clients

The backup and restore operations using the *mysqldump* and *mysql* tools is also available in all recommended MySQL client applications. When using a client application ensure the following:

- Database schema = mosarttemplatedb.
- Export to a single file (i.e. the entire database should be saved on 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

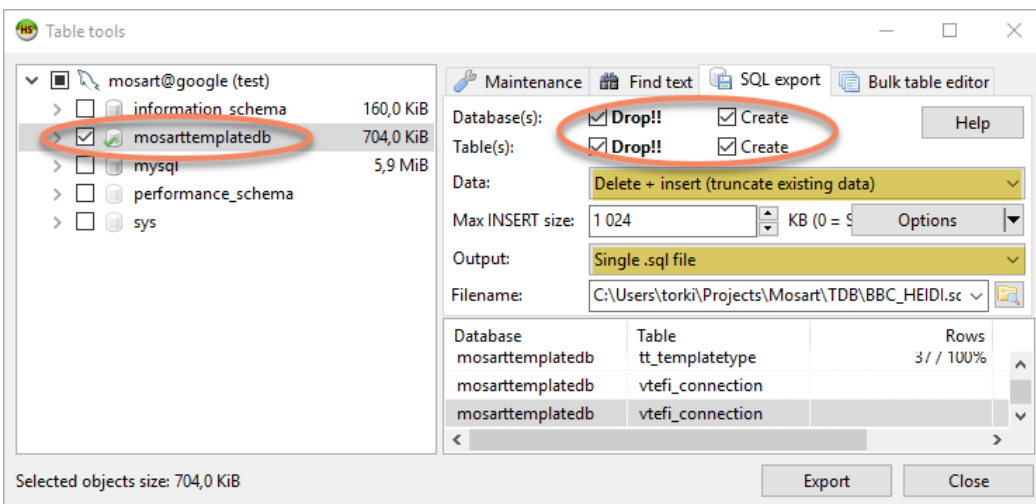
In MySQL Workbench the export dialog box is obtained from the **Server > Data Export** menu option. The figure below highlights the recommended options:



Restore from a single .sql file in MySQL Workbench is done via the **Server > Data Import** dialog. Simply select the .sql file as created during backup.

Example, backup from HeidiSQL

In HeidiSQL the export dialog is obtained from the **Tools > Export database as SQL** menu option.



In HeidiSQL ensure the following:

- Drop and Create is selected for both Database(s) and Table(s)
- Data: is set to "Delete + insert (truncate existing data)"

- Output: is set to "Single .sql file"

Restore from a single .sql file in HeidiSQL is done via the **File > Run SQL file...** menu option.

12.6 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 \[set1\] \[set2\] \[type\] \[template\]](#)

12.6.1 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.

Version	Date	Version	Description
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.
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.

12.6.2 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.


12.6.3 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

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 repository (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;UserId=user;password=password;database=mosarttemplatedb
- *Database2*: gallery=StudioE;server=146.148.9.188;UserId=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
<i>DiffTool1</i> : Visual Studio Code	Name=VSCode;Program=C:\Program Files\Microsoft VS Code\Code.exe;Args=--diff {0} {1}
<i>DiffTool2</i> : Beyond Compare	Name=BComp;Program=C:\Program Files\Beyond Compare 4\BCompare.exe;Args={0} {1}
<i>DiffTool3</i> : 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
<i>EditTool1</i> : Visual Studio Code	Name=VSCode;Program=C:\Program Files\Microsoft VS Code\Code.exe;Args={0}
<i>EditTool2</i> : XmlMarker	Name=XmlMarker;Program=C:\Program Files (x86)\XML Marker\xmlmarker.exe;Args={0}
<i>EditTool3</i> : 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 deprecated 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
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.

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).

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).
- **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 templateset / *channeltemplates.xml* file. Default: *C:\channeltemplates\ChannelTemplates.xml*.

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.

Save

Saves any changes to the template database and the current *channeltemplates.xml* file.

SaveAs [filename]

Saves the current templatesets to the given filename.

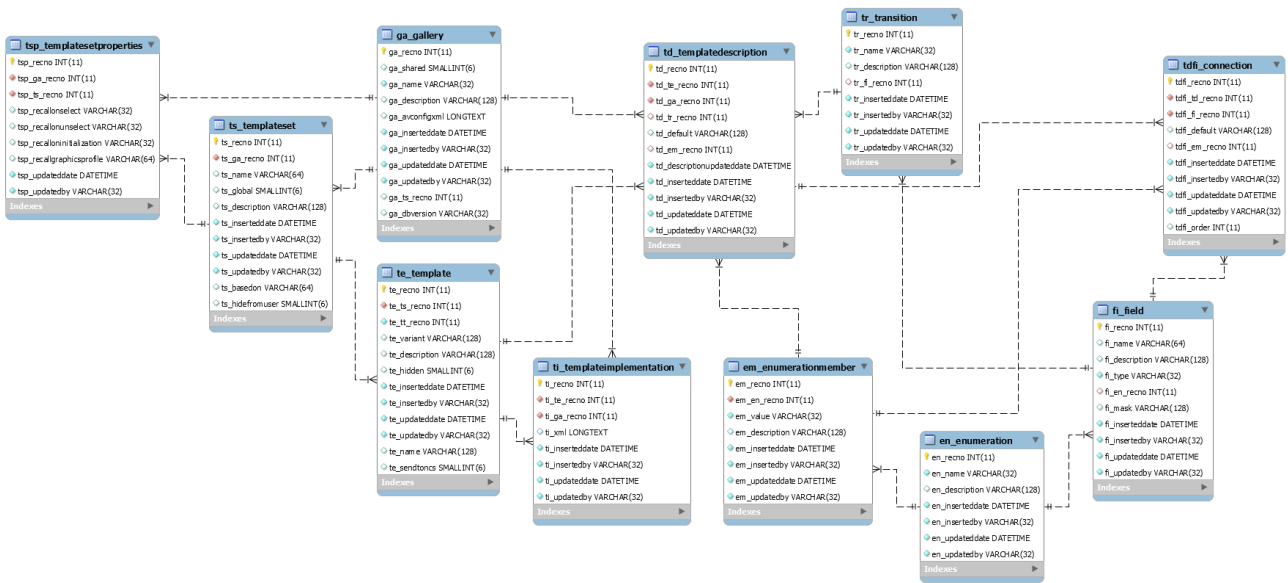
12.7 Viz Mosart Template Database Specification

This section provides database specifications for the Viz Mosart Template Database:

- [Introductions and Notations](#)
- [Viz Mosart Database Tables](#)

12.7.1 Entity Relationships

The following figure shows the relationships between the most important tables within the database.



12.7.2 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.

Table name	Prefix	Short description
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.
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:

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).

Type	Comment
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.

12.7.3 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](#)

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.

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 <i>en_enumeration</i> table.
em_value	varchar(32)	No	The value of the enumeration member. Normally presented to the users.

Name	Type	Allow Null	Description
em_description	varchar(128)	Yes	Optional enumeration member description.

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.

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 users 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'.

Name	Type	Allow Null	Description
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.


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	Allow Null	Description
ga_recno	integer	No	Primary key.
ga_shared	smallint	Yes	Set to true (1) for the <i>shared</i> gallery. The <i>shared</i> gallery may be used by clients to obtain all shared template sets within the database. Only one gallery shall be used as the <i>shared</i> gallery. By default, a virtual gallery named SHARED is used for this purpose.
ga_name	varchar(32)	No	The gallery name.

Name	Type	Allow Null	Description
ga_description	varchar(128)	Yes	Optional gallery description.
ga_avconfigxml	longtext	Yes	The XML formatted contents of the <i>avconfig.xml</i> file used in this gallery.
ga_ts_recno	integer	Yes	Optional foreign key referencing the 'default' template set of the gallery.
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.

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.

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 <i>ga_recno</i> in the <i>ga_gallery</i> table.
sq_ts_recno	integer	No	Foreign key referencing the corresponding template set <i>ts_recno</i> in the <i>ts_templateset</i> table.
sq_name	varchar(32)	No	Name of sequence.

Name	Type	Allow Null	Description
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.

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	Allow Null	Description
ta_re cno	inte ger	No	Primary key.
ta_n ame	varc har(32)	No	The template type alias name. Should be unique, as this is presented to the users.
ta_d escr iption	varc har(128)	Ye s	Optional template type alias description.
ta_tt _rec no	inte ger	No	Foreign key referencing the corresponding Mosart template type in the <i>tt_templatetype</i> table.
ta_d efaul t	sma llint	Ye s	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.

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 <i>te_recno</i> in the <i>te_template</i> table.
td_ga_recno	integer	No	Foreign key referencing the corresponding gallery <i>ga_recno</i> in the <i>ga_gallery</i> table
td_tr_recno	integer	Yes	Optional foreign key referencing a transition <i>tr_recno</i> in the <i>tr_transition</i> 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 <i>em_recno</i> in the <i>em_enumerationmember</i> table.
td_descriptionupdate ddate	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.

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 <i>ts_templateset</i> table.
te_tt_recno	integer	No	Foreign key referencing the corresponding template type in the <i>tt_templatetype</i> 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.

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 <i>te_recno</i> in the <i>te_template</i> table.
ti_ga_recno	integer	No	Foreign key referencing the corresponding gallery <i>ga_recno</i> in the <i>ga_gallery</i> 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

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	<p>Foreign key referencing the corresponding gallery in the <i>ga_gallery</i> table.</p> <p>Template sets belonging to the shared gallery (<i>ga_shared=1</i>) are by definition <i>shared</i> template sets. Shared template sets will be present in all galleries.</p> <p>Template sets belonging to any other gallery are by definition <i>private</i> template sets owned by the corresponding gallery.</p>

Name	Type	Allow Null	Description
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).

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	

Id: tt_recno	Name: tt_name	Description
0	CAMERA	
1	PACKAGE	
2	VOICEOVER	
3	LIVE	
4	FULLSCREENGRAPHICS	
5	DVE	
6	JINGLE	
7	TELEPHONEINTERVIEW	
8	ADLIBPIX	
9	BREAK	
10	MACRO	
17	VIDEOWALL	
25	CLIP	
100	LOWERTHIRDS	
110	OVERSHOULDERGRAPHICS	
215	PLAYSOUND	
216	SETSOUNDLEVEL	

Id: tt_recno	Name: tt_name	Description
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	
999	IGNORE	

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.

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.

Name	Type	Allow Null	Description
tr_description	varchar(128)	Yes	Optional transition description.
tr_fi_recno	integer	Yes	Optional foreign key referencing an input field, <i>fi_recno</i> in the <i>fi_field</i> table.

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 <i>ga_recno</i> in the <i>ga_gallery</i> table.
tsp_ts_recno	integer	No	Foreign key referencing the corresponding template set <i>ts_recno</i> in the <i>ts_templateset</i> table.
tsp_recallonselect	varchar(32)	Yes	Optional directtake to be recalled when the template is selected in Viz Mosart automation.
tsp_recallonunselect	varchar(32)	Yes	Optional directtake to be recalled when the template is unselected in Viz Mosart automation.
tsp_recalloninitialization	varchar(32)	Yes	Optional directtake to be recalled when a rundown is reloaded.
tsp_recallgraphicsprofile	varchar(32)	Yes	Optional directtake to be recalled when a graphics profile (or graphics concept) is selected in Viz Mosart automation.

13 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](#)
-

13.1 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.

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.

Viz Mosart Logs are stored at the default Log Directory:

- *C:\MMLogs* (configurable)
The technical log stream is located at the root of this directory, while the other two log streams have their own corresponding subdirectories.

This section contains the following topics:

- [Log File Structure](#)
- [Log Viewer](#)
- [Technical Log Stream](#)
- [AsRun Log Stream](#)
- [MOS Log Stream](#)
- [Log Adapters](#)
- [Log Configuration](#)
- [Custom-built AsRunLog Adapters](#)
- [Log Properties](#)

13.1.1 Log File Structure

All Viz Mosart logs are stored as XML files.

Filenames

Log filenames are configurable, but typically follow this 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-second"

Current and Previous Files

There are two sets of log files:

- The current log file, which is active.
 - 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.
- Previous log files.
 - When a new log file starts, the current active log is merged into a single XML. The figure below shows the default setup with a list of log files belonging to the technical log and two directories containing similar log files for the AsRun and MOS logs respectively.

Navn	Størrelse...	Type	Endret dato
Log.20101011-101246.log	4 kB		11.10.2010 10:29
Log.20101011-101246.xml	1 kB		11.10.2010 10:12
Log.20101007-101245.xml	249 kB	XML-dokument	08.10.2010 10:12
Log.20101006-101245.xml	1,235 kB	XML-dokument	07.10.2010 10:12
Log.20101005-101244.xml	1,078		06.10.2010 10:12
AsRunLog		Filmappe	11.10.2010 10:13
MosLog		Filmappe	11.10.2010 10:12

13.1.2 Log Viewer

Viz Mosart contains a dedicated application for viewing log files. Log Viewer is located in the Viz Mosart Server installation directory and is available as a separate installation package.

✔ **Tip:** To improve filtering and visualization, the XML log files can be opened in Microsoft Excel.

13.1.3 Technical Log Stream

This log contains ordinary log events produced by all running [Viz Mosart Applications](#).

Note: This 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 shown in the figure below:

- **AsRunLogger:** See [AsRun Log Stream](#).
- **AudioMixer:** Events to/from the audio mixer.
- **AudioPlayer:** Events from the Audio player (will be changed to also include messages from AVAutomation 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 local log service from the Viz Mosart GUI. Extended set of local GUI events. Logged to local GUI PC, except pressed keys, which are sent to the server log.
- **INewsWebServices:** Events associated with iNews web service.
- **InstantCollect:** Event used to identify a user triggered log marker. Note that this event type is obsolete in version 3.x.
- **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](#).
- **MediaAssetManagement:** Currently only used for Amadeus component.
- **MediaRouter:** Media router events.
- **Mimic:** Events associated with Mimic functionality (version 3.8).
- **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 Soundfile Player (currently also for messages from AVAutomation to the new Audio Player - will be changed to AudioPlayer).
- **SpeakNoticer:** Events associated with the Asio Speak Notifier (for Mimic) (version 3.8).
- **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 (version 3.7).
- **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:** Currently only used for BrainstormVS component.
- **WeatherControl:** Events to/from the weather control.

Verbose Properties

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:**
 - A comma separated list of log event types to ignore during verbose logging.
 - Default: <empty> (i.e. no events are ignored).
- **VerbosePassEvents:**
 - A comma separated list of log event types that are passed during verbose logging.
 - Default: <empty> (i.e. all events are passed).

Example: Setting `VerbosePassEvents = "VideoServer,AudioMixer"` ensures that only verbose events to/from the video servers and audio mixer are passed to the Log Service.

13.1.4 AsRun Log Stream

This log stream contains events connected to content that has been on-air, typically this includes all video and graphic content. The Log Service may be configured to extract information from the AsRun Log for input into media asset management or presentation automation systems. All events in the AsRun Log stream are classified according to the event types as shown in the figure 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.

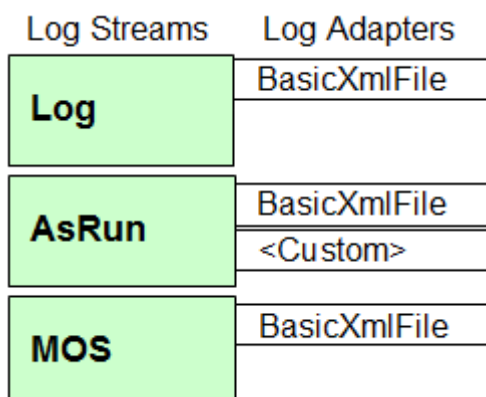
Note: Customer specific AsRun Log Adapters have their own syntax.

13.1.5 MOS Log Stream

This log stream contains events communicated between a MOS based Newsroom System and [Media Administrator](#).

13.1.6 Log Adapters

Each log stream may be processed simultaneously by a set of log adapters as indicated by the figure below:



In the figure the three log streams are configured as follows:

- The [Technical Log Stream](#) has one adapter:
 - BasicXmlFile: default adapter that creates the XML based log files.
- The [AsRun Log Stream](#) has two adapters:
 - BasicXmlFile: Adapter as for the technical log stream.
 - Custom adapter: Processes the AsRun log stream according to site specific requirements. These adapters are custom built, and not included in the Viz Mosart distribution.
- The [MOS Log Stream](#) has one adapter:
 - BasicXmlFile: Adapter that creates the XML based log files.

13.1.7 Log Configuration

Each log has its own configuration file making it possible to adjust properties such as filenames and how often new log files are generated.

Overview

The log configuration files are **located** in the following directory:

```
%ProgramFiles%\Mosart Medialab\Mosart Server\ConfigurationFiles
```

The log configuration files are **named** as follows:

- *LogRepositoryConfig.xml*: Technical log.
- *AsRunLogRepositoryConfig.xml*: AsRun log

✔ **Tip:** For details about customer specific AsRun log files, see [Custom-built AsRunLog Adapters](#).

- *MosLogRepositoryConfig.xml*: MOS log, a log configuration file consists of three **segments**:
 - Adapters: Configuration of one or more log adapters
 - FilterBank: Configuration of optional log filters
 - Properties: Configuration of log properties

Example (AsRun log)

A sample log configuration file for the AsRun log stream is shown below:

```
<?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>
```

Configuration of Log Adapters

A log stream may have one, or several, log adapters. These are configured in the Adapters section as shown in the log filter sample earlier. If the Adapter section is omitted, or empty, one single log adapter of the default type is created.

The default log adapter type is configurable in the property 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. Used mainly for logging purposes.
- **configuration:** Optional configuration file for the log adapter. If no configuration file is given the log adapter shall inherit the values in the current configuration file.
- **assembly:** Optional .Net assembly (.dll) containing the log adapter. I.e. opens for dynamic log adapters that may be loaded at run-time.

Configuration of Log Filters

A log adapter may have a set of log filters that remove events before being 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 the following three attributes:

- **event:** Which event the filter applies to. Events are listed in sections [Technical Log Stream](#) and [AsRun Log Stream](#). 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 (i.e. forwarded to the log adapter).
 - **Break:** The event is treated as a break event.

The following shows some examples for filtering the 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>

```

13.1.8 Custom-built AsRunLog Adapters

These adapters are custom-built to suit a certain customer's needs. They are configured by the standard *AsRunLogRepositoryConfig.xml* as earlier described, but also include two more configuration files that may contain elements that override this standard configuration file.

- *xxxAsRunLogConfig.xml*, (Overrides AsRunlogRepositoryConfig)
- *xxxSchedulerConfig.xml*, (Overrides xxxAsRunLogConfig)

 **Note:** xxx is here substituted for the actual Broadcaster's name.

An example of the *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" />
<!-- These values may be overridden in SchedulerConfig -->
<item name="RepositoryPath" value="C:\MMLLOGS\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>

```

An example of the *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:\MMLLOGS\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>

```

13.1.9 Log 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.
- **MaxFileTimePeriodHours:** Maximum number of hours before a new logfile is created. Default: 24.
- **UseExtendedLogging:** Logs verbosely. See example in [Grass Valley K2](#)
- **UseLocalTime:** If set to true, all timestamps are converted from UTC to local time before being passed to the log adapter. Default: False.

13.2 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](#)

13.2.1 Server File Structure

The Viz Mosart Server stores files in different locations on the workstation as indicated by the list below:

Location	Definition
<i>C:\Program Files (x86)\Mosart Medialab</i>	Mosart applications. Various sub-directories for components such as Mosart GUI, Mosart Server and more. Either <i>\Program Files</i> or <i>\Program Files (x86)</i> .
<i>C:\channelTemplates</i>	XML-files defining the channel templates. Normally contains at a minimum <i>channeltemplates.xml</i> , <i>newsroomsettings.xml</i> , <i>AVConfig.xml</i> .

Location	Definition
<i>C:\manus</i>	The Run-down (playlist) files fetched from the Newsroom systems.
<i>C:\MMLogs</i>	Default location for Log-files. Rotated.
<i>C:\ProgramData\Mosart Medialab\ConfigurationFiles</i>	System-wide configuration settings (XML).
<i>C:\Users\<USERNAMEHERE>\AppData\Local\Mosart_Medialab</i>	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.

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%/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.

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. One for each Viz Mosart installer.

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.

13.2.2 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.

13.2.3 File Purging

The [Manus Administrator Repository](#) and the [Log Directory](#), have a built in mechanism for purging files. The content of these directories grows unless a maintenance procedure is established. This can be configured in the [Manus Administrator](#) program by writing `settings`, and then setting the value `ManusExpirationTime` to any number of days.

The Log system is preset to a 60 days retention period, but this can be changed, see [Log Properties](#).

13.3 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](#)

13.3.1 Rebooting and Restarting – General Notes

All software system must be restarted occasionally. This applies to both the operating and Viz Mosart system.

13.3.2 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.

13.3.3 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.

13.3.4 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.

13.3.5 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).

13.3.6 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).

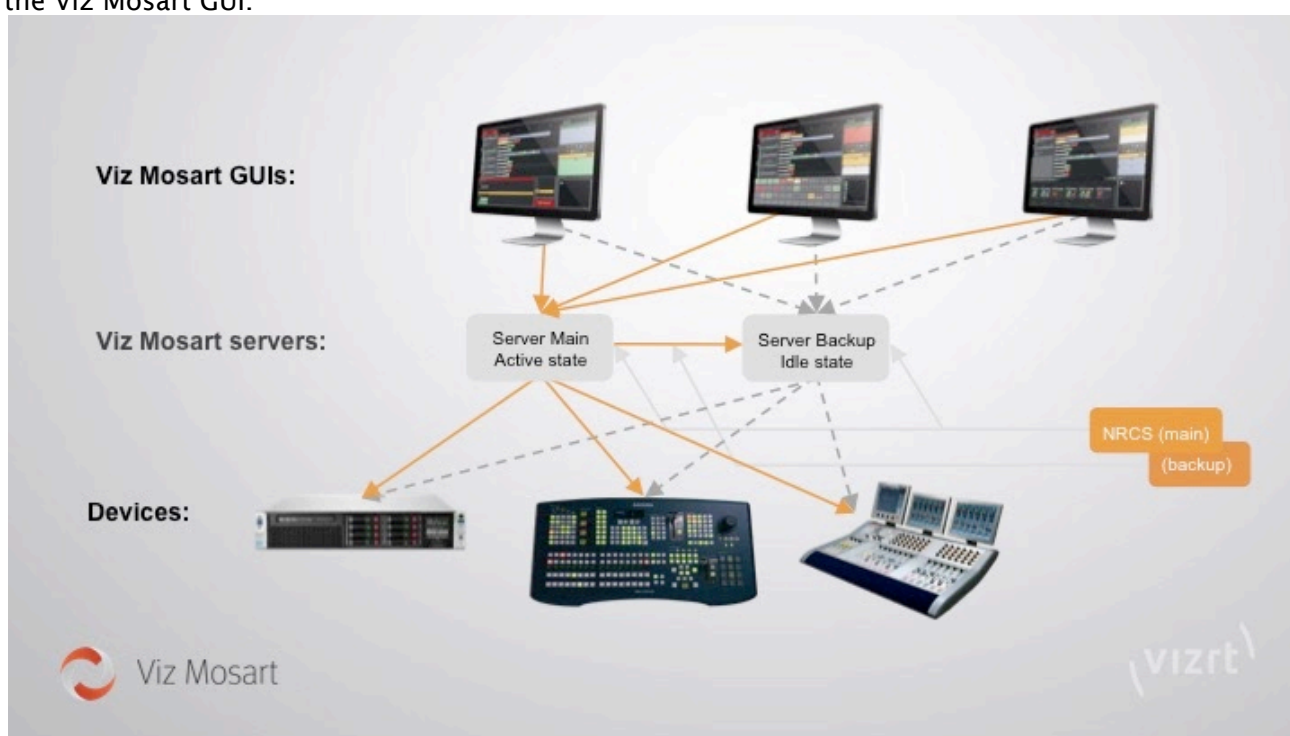
13.4 Redundancy

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 setups possible.

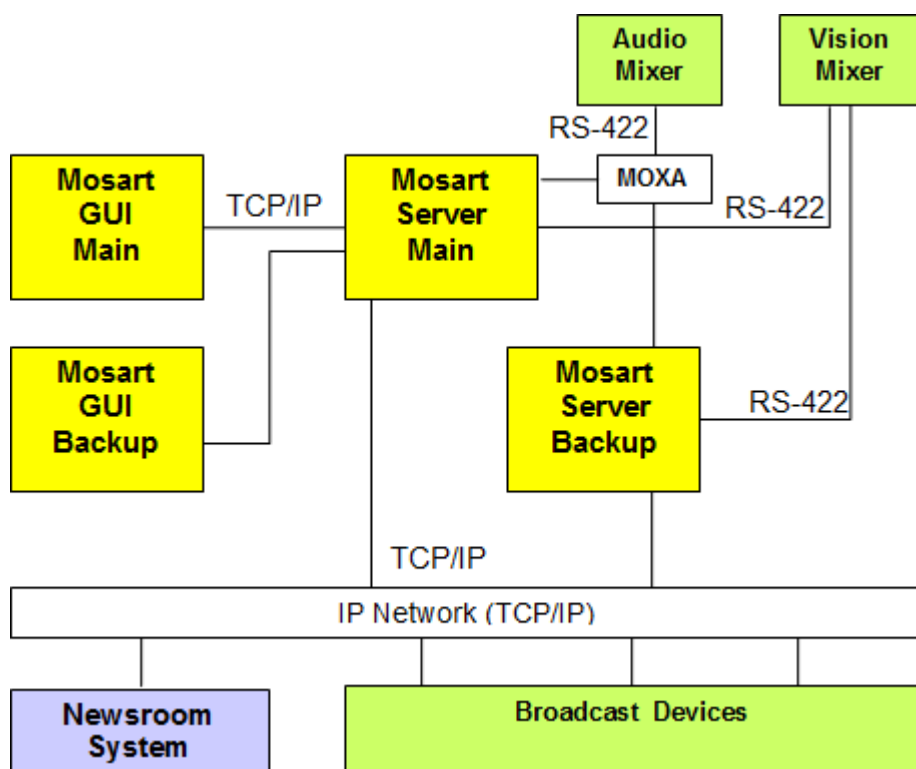
- [Automation Server Mirroring](#)

- 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 GUI PC configuration with dual redundant connections (Network Teaming) to broadcast devices where possible. The recommended redundancy setup is to run a Main and Backup Viz Mosart Server with two workstations running the Viz Mosart GUI.



A schematic example of a recommended redundancy setup is shown in the figure below (depending on the characteristics of the connected devices):



The system in the figure above has the following characteristics:

- Dual redundant system.
- All Viz Mosart applications and services part of the Viz Mosart Server suite are interconnected and running on the same server.
- Both Main and Backup Viz Mosart GUI applications are configured to initially use the Main Viz Mosart Server. Only one click is needed in order to connect to the backup server.
- All external equipment and systems are shared between the main and backup servers.
- The vision mixer is in this example 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 to enable 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.

13.4.1 Automation Server Mirroring

The automation servers work in an active/idle setup where the active server is pushing the various states to the idle server. This allows the idle server to take over the operation as any given time seamlessly. Both servers will work independently from each other in the case of a server failure.

13.4.2 Connected GUIs

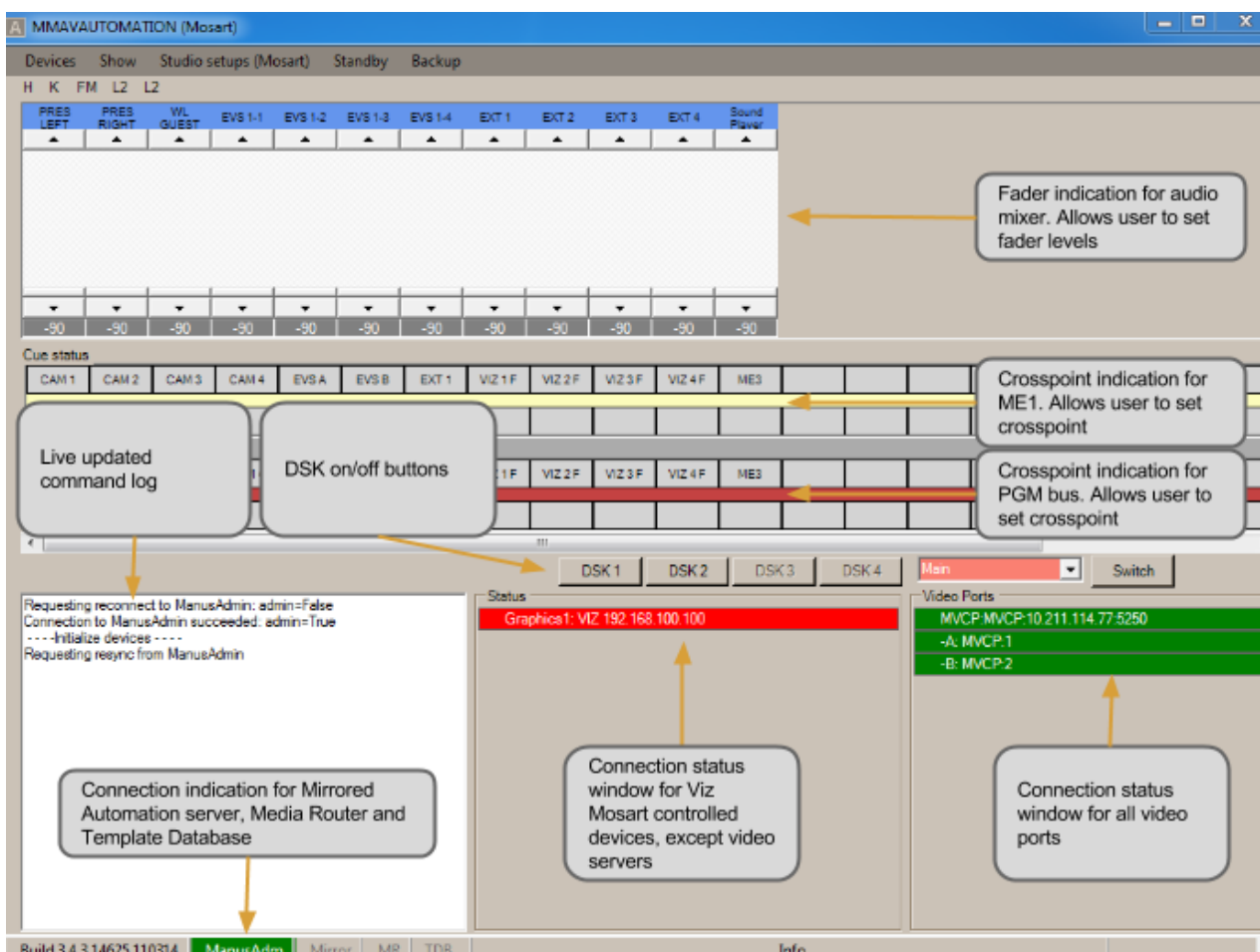
Viz Mosart supports an unlimited number of user interface clients. In the case of a server switch-over, all GUI clients follow this switch-over.

13.4.3 Graphical Interfaces

Viz Mosart has two main interfaces that expose redundancy features of the system to operators:

13.4.4 AV Automation – The Engineering Interface

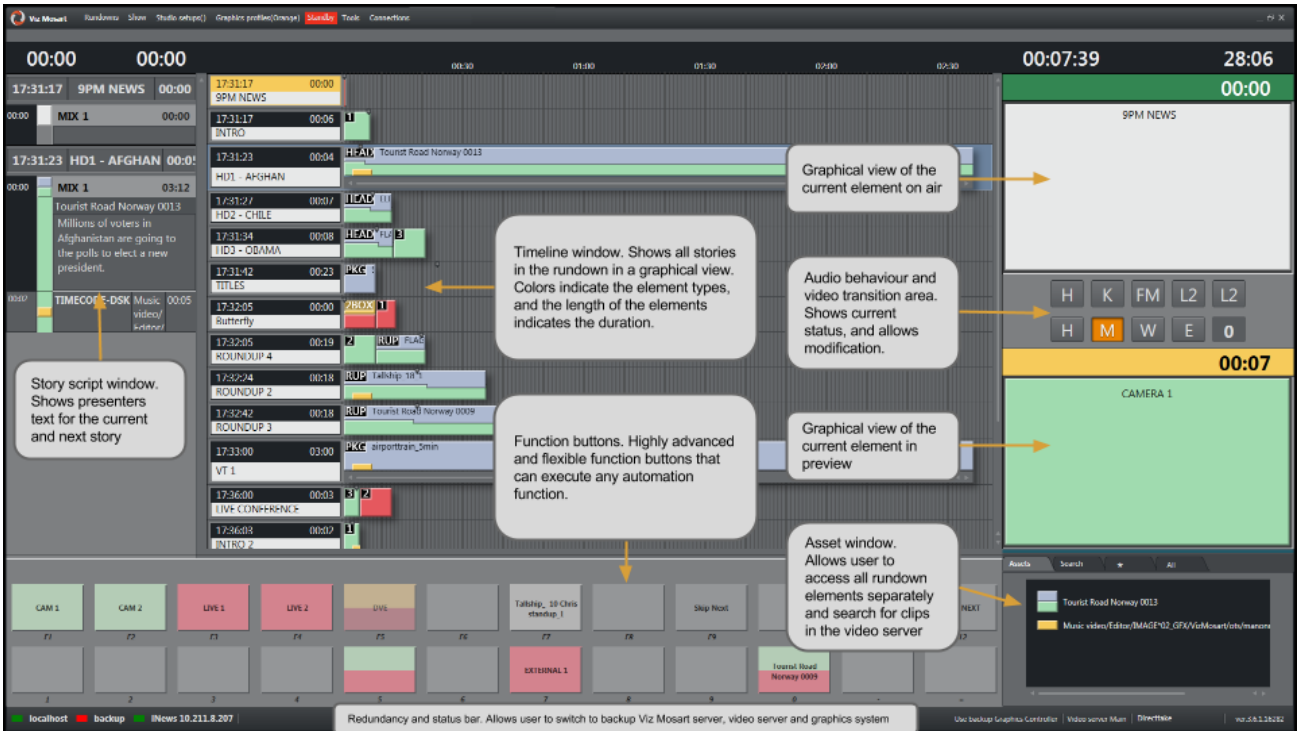
AV Automation provides connection statuses of the connected devices and enables the user to access several functions connected to system redundancy. AV Automation runs on each of the two automation servers.



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.



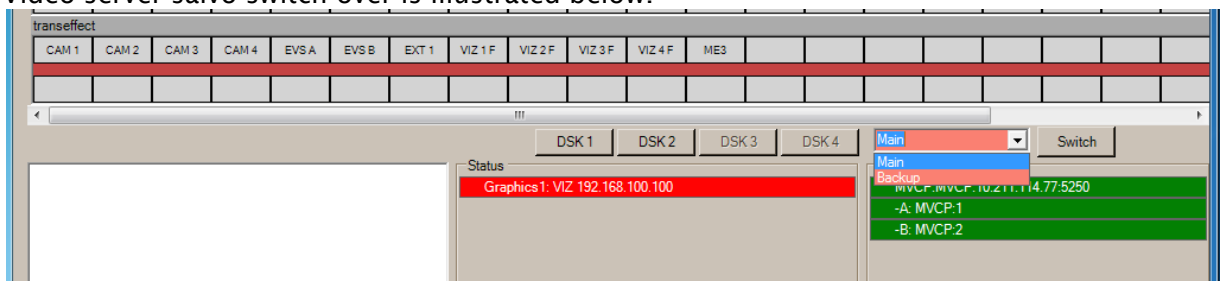
13.4.5 Device-specific Redundancy

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.

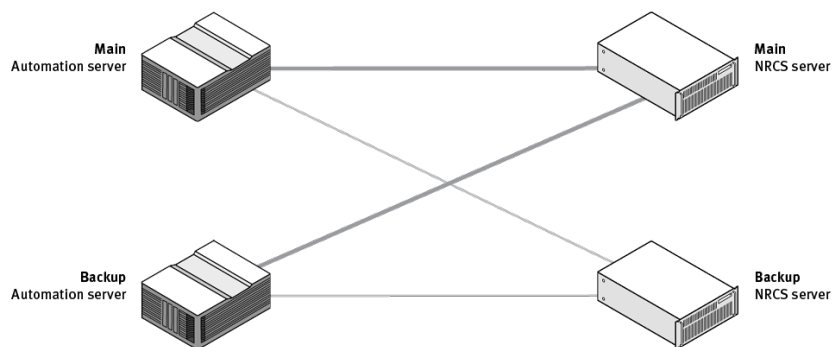


Video Switcher Backup

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.

Newsroom System Switch-over

Viz Mosart supports dual NRCS servers and will follow the backup switch-over mechanism of the newsroom system.



See Also

- [Media Sequencer Redundancy](#)

14 General Configuration Files

This section contains:

- [Named Overlay Graphics](#)
-

14.1 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](#)
- [How to obtain valid overlay graphics as a named overlay candidate](#)

14.1.1 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.

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>
```

14.1.2 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:

- **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.

14.1.3 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.

14.1.4 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".

14.1.5 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:

Command On Take		Control commands Take	
	Command	Value	Parameter
▶	OVERLAY_GRAPHICS	TAKE_NAMED_OVERLAY	LIVE ON
	OVERLAY_GRAPHICS	TAKE_NAMED_OVERLAY	LOGO BUG ON

14.1.6 How to obtain 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](#).

15 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](#)
-

15.1 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 Settings**
For example Name, Type, Server, Config (the name of the associated file that contains **Custom Settings**)
- **Custom Settings**
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 auto-generated when configured in the **Video Server** GUI. In AVAutomation, you can also enter additional device parameters to match specific control room parameters.

15.1.1 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

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

15.1.2 Standard Connection String Properties

Irrespective of device type, several values are standard parameters.

Viz Mosart Video Server Connection String

```
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 AVAutomation, the **Name** in AVAutomation should be the same as here.
- **Type:** Specifies which driver to be used for connecting to the server. See the table below for details about which Video Server Drivers that are currently supported.
- **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 this MAM system (or video server), also when *Enable dynamic configuration* (see [Media Administrator Properties Editor](#), Configuration) is *True*, and this MAM system (or video server) is not part of the salvo selected in *AV Automation* (see [AV Automation Devices Video Servers](#), Virtual Server Ports). (Normally, a MAM system will not be part of any salvo, so this setting is mostly used for MAM systems which should always be connected, regardless of the selected salvo.) Default *False*.

Video Server Drivers

- **AirSpace:** AirSpace video driver.
- **AirSpeed:** AirSpeed video driver.
- **AirSpeedSearch:** As AirSpeed driver, but contains search functionality only. Used by Media Administrator to obtain clip status from an AirSpeed video server.
- **AirSpeedMultiStream:** AirSpeed Multi Stream driver, also Airspeed 5000.
- **AirSpeedMultiStreamSearch:** As AirSpeed Multi Stream driver, but contains search functionality only. Used by Media Administrator to obtain clip status from an AirSpeed Multi Stream or 5000 video server.
- **Amadeus:** Amadeus MAM driver.
- **EVS:** EVS driver.
- **GrassValleyK2:** Grass Valley K2 driver.
- **OPUS:** Omnibus MAM driver using OPUS interchange.
- **Omneon:** Omneon video driver.
- **OmneonSearch:** As Omneon driver, but contains search functionality only. Used by Media Administrator to obtain clip status from an Omneon video server.
- **Quantel:** Quantel video driver.
- **ScreenLLSearch:** Screen LinguaLib search driver.

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 (Default=True).
- **SupportSearchUsingRefID (boolean):** Activates searching using RefID (Default=False).
- **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 general heartbeat interval. Default: 10000 msec.
- **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 msec.
- **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](#).
- **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 templateset.
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:

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>

```

15.1.3 Device-specific Connection String Parameters

AirSpeed Classic Connection String

Applies to Models


- AirSpeed
- AirSpace

Connection String Properties

- **Type:** AirSpeed and AirSpace.
- **Name:** 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.

Optional

- **BasePort*:** The lowest port number to connect to (Default=59451).
- **Player*:** The player number (1-6). Used for setting the TCP port (Default=1).
- **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.

Examples

AV Automation:

```
Name=HAR; Type=AirSpeed; Server=192.168.1.1; Port=6000;
Config=ClipServerMultiStream.xml
```

Media Administrator:

```
Type=AMS; Name=AirSpace; Server=10.211.114.104; BasePort=59451; Player=2;
ChangeCheckInterval=5; ReceiveTimeout=5; HeartbeatInterval=20;
Config=ClipServerMultiStream.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).
- **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=False).
- **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 data 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 String

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

Applies to Models

- AirSpeed Multi Stream 1.8
- AirSpeed 5000

Connection String Properties

- **Type:**
 - AV Automation: AirSpeedMultiStream
 - Media Administrator: AirSpeedMultiStreamSearch
- **Name:** See [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, cp. the 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. Default *false*. Must be *true* if MPEG-2 HD clips are involved.
 - **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=True).
- **Media Administrator only**
 - **ChangeCheckInterval:** Time in seconds how often to poll AirSpeed for clip changes (Default=1).
 - **HeartbeatInterval:** Time in seconds to send out a heartbeat in case of no communication detected from AirSpeed (Default=30).
 - **Player*:** Specifies the *port* to use. Valid values are 1-6. Default=1. See the Note below for how the actual port number is calculated.
 - **ReceiveTimeout:** Time in seconds to wait for response from AirSpeed (Default=10).

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

Examples

AV Automation:

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

Media Administrator:

```
Name=AMS; Type=AirSpeedMultiStreamSearch; Server=192.168.1.1; BasePort:59451;
Player=2; ChangeCheckInterval=5; ReceiveTimeout=5; HeartbeatInterval=20;
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 data 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).

Amadeus Connection String

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

Connection String Properties

- **Name:** Amadeus ID for database
- **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).
- **PurgeCacheInterval:** Use hh:mm:ss format (Default=00:00:00).
- **PurgeCacheAge:** Use hh:mm:ss format (Default=00:05:00).

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*:** 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).
- **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=HAR; 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](#).

JupiterWebService Connection String

JupiterWebService is the BBC’s own 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:** 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*:** Set to "SVT" to enable SVT custom search on *objSlug*.
- **SetNameAsRefId*:** Set clip name as ref ID in the media object (Default=False) for re-use in other clip servers.
- **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 (Default="") from the MVCP interface.
- **ClipNamePattern:** See [General Configuration File Properties](#).

 **Note:** The asterisk (*) above denotes custom search 'SVT': SVT has file names with 36 characters. The Nexio only supports 32 characters. SVT exposes 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=MVCP1; Type=MVCP; Server=192.165.4.1; Port=5250
```

Media Administrator:

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

Omnibus OPUS Connection String

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

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

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:** Specifies 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.

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.xml
```

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).
- **ClipNamesValidNumber (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 (Default=Title). Used if SupportSearchUsingRefID is set to true.
- **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](#).

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.

Omneon Connection String

Applies to Models

- Models with Omneon Spectrum

Note: Viz Mosart does not work properly with Omneon Spectrum versions earlier than 4.6. 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:** ID of this connection.
- **Server:** Hostname or IP-address of the server.
- **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 (Default=\$1). Used when replacing object slug (objSlug).
- **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).
- **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 cuing a new clip. This causes cuing to take longer than usual.
- **ClipNamePattern:** see [General Configuration File Properties](#).

Search Properties: See [General Search Configuration File Properties](#).

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 [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:** The bin of the file repository on the K2 (Default=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 .
- **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.

- **LoadDatabase:** Whether to maintain a 'cache'/'database' of the server clips. Default *false*. For AV Automation, keep the default *false*. For Media Administrator, use *true*.
- **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:** The Grass Valley suite. 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.) And if there is more than one K2 server, this applies to each of them, and *all* the values should be unique. For further info wrt. the redundancy setup, refer to [Failover with K2 Servers](#).
- **User:** The username as part of the user credentials.
- **Volume:** The volume of the file repository on the K2. (Default=V:)

Examples

AV Automation:

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

```
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 connection 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. If neither is specified, 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. If neither is specified, 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 (e.g., if a clip added on the GV server does not turn solid blue in GUI.)

- **UseVideoFormatFromServer (Boolean, default true):** Set to false if clip durations (e.g. 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:

- **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.

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.
- **Name:** Defines a unique name for the server.
- **Server:** Defines the IP address of the server.
- **LoadDatabase:** If true, loads all clip metadata into local cache (Default=False). 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).
- **LocalMtpclipAddress:** 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 <code>LogOutput=...\linx.log</code>
0x00000003	Function entry and exit points, with parameters and return code	linx.log
0x00000010	Connections	linxConnect.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 (Default=16). Increase if the error code 'Too many retries' is encountered, e.g. when synchronizing the database.
- **SharedConnectTimeout:** Maximum time in milliseconds to execute connection request on a shared connection (Default=500). Consider increasing if the error code 'Too many retries' is encountered, e.g. when synchronizing the database.

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:** Defines the unique name for the server.
- **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.

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:** Defines a unique name for the server.
- **Server:** Defines the IP address for the Control center.
- **Config:** Defines the configuration file containing further details.

Note that the parameters mentioned so far are insufficient to connect to the database. For this, the parameters DataSourceName, User, and Password are required. The defaults are probably sufficient. If not, these parameters are more conveniently set in the configuration file, 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=False).
- **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=NOTPLAYABLE-).
- **PrefixCheckItemReturnOnlyItemLowResPresent:** The prefix added to the Description of a low-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).

ScreenLL Connection String

Screen Connection String, used by Media Administrator.

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

- **Name:** Not important, used for logging purposes only.
- **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**

Orad Using OCIP Connection String

This connection string is used for Orad video servers controlled by OCIP (Orad Control Interface Protocol).

```
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 [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.

 **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 MediaAdmin). 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 *growing* parameter set to *True* will be ignored. When set, this implies that growing clips are processed as non-existent and are unavailable. Default False. 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.
- **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

Device Prerequisites

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

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.

Applies to Models

- Dalet BRIO
- Vector MultiPlay VServer

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.

Connection String Properties – VDCP

- **Type:** VDCP (serial) or *VDCPtcp* (Ethernet).
- **Name:** ID of this connection.
- **Server:** Hostname or IP-address of the server (for Type=VDCPtcp only).
- **Port:** Serial port (Type=VDCP, Default=COM1) or server side TCP port to connect to (Type=VDCPtcp, Default=52z05).
- **LoadDatabase:** If true, builds internal database of clips present on server and enables clip monitoring (Default=False).
- **Config:** Name of the configuration file. *ClipServerVDCP.xml* is the default name of the file used by VDCP Connection String.

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
```

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
```

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=False).
- **IgnoreClipExistence (boolean):** Enable to ignore requesting clip presence on the server.
- **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.
- **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.
- **UseAddedClipsCommand (boolean):** Enable to retrieve a list of new files added to the video server.
- **NextPingDelay (milliseconds):** Interval in milliseconds between heartbeats sent from AV Automation to the VDCP server to keep the connection alive. Default = 10 seconds). e.g.:

```
<item name="NextPingDelay" value="10000" />
```

 Turn off this ping using the *DisableHeartbeat* setting described below.
 To modify the ping interval for Media Administrator see [Media Administrator Properties Editor > Ping Delay](#).
- **DisableHeartbeat (boolean):** A command request (heartbeat) is sent from AV Automation and Media Administrator to the VDCP server to keep the connection alive.
 - Turn off the heartbeats from both AV Automation AND Media Administrator by setting `DisableHeartbeat=true` in `ClipServerVDCP.xml`. e.g. `<item name="DisableHeartbeat" value="true" />`
 - Turn off the heartbeats from one application (AV Automation OR Media Administrator), by setting `DisableHeartbeat` in the application's connection string and removing (commenting out) `DisableHeartbeat` from `ClipServerVDCP.xml`.
 Example connection string:

```
Type=VDCPtcp;Name=Server1;Port=10001;DisableHeartbeat=true;Config=ClipServerVDCP.xml
```

 See also *NextPingDelay*, described above.
- **ClipNamePattern:** see [General Configuration File Properties](#).

Viz Engine Connection String

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

Connection String Properties

- **Type:** VizEngine.
- **Name:** 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):
 - 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 Viz One and Media Administrator for [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. Should be equal the name of the video server in AV Automation.
Type	VizOne	Type of connection. Must be <i>VizOne</i> 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. <i>admin</i> as username and password may be used for testing, but should <i>not</i> 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
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.

Naming	Examples	Description
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 <i>Configure Media Service</i> 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 .

Example (Media Service)

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

Example (Viz One)

```
Name=VizOne;Type=VizOne;UserName=username;Password=password;Server=10.211.112.213;PublishingPoints=vizengine-1;ValidDuration=26;AutoTransfer=true;PublishingPointAsObjId=false;Config=ClipServerVizOne.xml
```

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

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* (e.g. in Video clip Server 1) that/those of the video server(s).

Applies to 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 [Viz Mosart Video Server Connection String](#) 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](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](#).
- **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.xml;DbConnectionString="Persist Security Info=True;Initial Catalog=HarrisDBTest;DataSource=BGO-OFREENGINE\SQLEXPRESS;Failover Partner=aut-db-srv03.fs-pn.vizrt.net\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 changed, follow the standard procedure:

- Copy the file to `c:\channel\templates`, 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.
- **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.
WhereColumn	The WHERE clause column to be searched for the (variable) search criterion.
AdditionalWhereColumns	Additional WHERE clause columns being searched for <i>constant</i> 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

Attribute	Value(s)	Description
type	String/Int	The DB column data type. For Microsoft SQL Server, <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> Default: The value is passed as is. (And this <i>is</i> 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	
type	String/Int	The DB column data type. For Microsoft SQL Server, <ul style="list-style-type: none"> String is used for char and varchar columns. Int is used for int columns. 	Only String has been tested.

Attribute	Value(s)	Description	Comment
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): <ul style="list-style-type: none"> · 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 <i>not</i> 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:

Attribute	Description	Comment
name	The property name	<p>Only these have been implemented:</p> <ul style="list-style-type: none"> · 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.

Sub-element	Description	Comment
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.
OperationExpression	An OperationExpression is used to combine other (simpler) expressions by performing an operation, which means applying an <i>operator</i> to the values of its <i>operands</i> (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.

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).

Operator	Expression value
Plus	<p>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. Anyway, a</p> <pre><ConstantExpression value=0 /></pre> <p>would be more convenient.)</p>

15.2 Video Wall Connection Strings

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

 **Note:** For general information on connection strings, see [Video Server, MAM and Database Connection Strings](#).

15.2.1 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];BusyNonCommandTimeout=[busyNonCommandTimeout];BusyTimeout=[busyTimeout];GetStatusThreshold=[getStatusThreshold];LoadTimeout=[loadTimeout];PingInterval=[pingInterval];PingTimeout=[pingTimeout];Port=[port];ReconnectInterval=[reconnectInterval];ResetAnyway=[resetAnyway];ResetTimeout=[resetTimeout];RunTimeout=[runTimeout];SequenceIndividualCommands=[sequenceIndividualCommands];SequenceRecallAndMixes=[sequenceRecallAndMixes];Server=[server];UseGetStatus=[useGetStatus]
```

Example:

```
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

String	Type	Description	Default
GetStatusThreshold	integer	The highest acceptable value of the third parameter of the Reply feedback to the getStatus command General health status of the cluster: <ul style="list-style-type: none"> · 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
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

String	Type	Description	Default
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

16 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 `C:\channeltemplates`, and modify them there. This ensures that configurations remain valid after system upgrades and downgrades.

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: you might 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](#)

Information about [Video Server Configuration Files](#) is found in the previous section (Device Connection Strings).

16.1 Robotic Camera Configuration Files

This section contains the Robotic Camera types:

- [Cambotics Configuration File](#)
- [Cinneo Configuration File](#)
- [Technodolly Configuration File](#)
- [Panasonic Configuration File](#)
- [Electric Friends Configuration File](#)

16.1.1 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 [AV Automation Devices 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*.

16.1.2 Cinneo Configuration File

The Cinneo configuration file is named *DeviceConfig.xml*.

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

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

16.1.3 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*.

16.1.4 Panasonic Configuration File


The Panasonic configuration file 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.

16.1.5 Electric Friends Configuration File

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

 **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>}}

```

16.2 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](#)

16.2.1 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>
```

- **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 rold 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 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.

16.2.2 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"/>
    <item name="ConnectedWhenIdle" value="false" />
    <item name="ConnectedWhenInStandBy" value="false" />
  </Properties>
</DeviceConfig>

```

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

- **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 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.
- **AskPageInfo:** Defines whether Viz Mosart should ask Pixel Power for page info when creating a page.
- **AskPageImage:** Defines whether Viz Mosart should ask Pixel Power for page image when creating a page.
- **AskFieldInfo:** Defines whether Viz Mosart should ask 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.

16.2.3 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

- **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)
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`

16.2.4 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 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

16.3 Subtitling Configuration Files

16.3.1 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>
```

- **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 (i.e. is not available even if it matches the *AvailabilityLanguageCode*). Other configuration items are client specific, or for Vizrt, and will not be further described in this document.

16.4 Video Router Configuration Files

16.4.1 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.

For details on the NV9000 protocol, refer to the product documentation written by GrassValley (www.grassvalley.com/support).

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

The following settings are available:

- **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 [AV Automation Devices Router](#). Select the router protocol MIRANDA NV9000, set the IP address of the router, and the Port number (use 9193).

16.5 Audio Mixer Configuration Files

16.5.1 Calrec TCP/IP Configuration File

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

```

<?xml version="1.0" encoding="utf-8" ?>
  <DeviceConfig name="AudioMixerCalrec">
    <Properties>
      <!-- If set to true, controller stays connected when in idle. Note that in a
      redundancy setup, on both main and backup Viz Mosart servers this option must be set
      to false. -->
      <item name="ConnectedWhenIdle" value="false"/>

      <!-- If set to true, AvAutomation waits for Ack after sending command to the
      driver.
      Note that this configuration will overwrite the general configuration with same
      name from AvAutomation settings (Ctrl+Shift+S).
      If commented, the general configuration from AvAutomation will be taken into
      consideration.-->
      <item name="AudioWaitForAck" value="true"/>

      <!-- Max wait time for receiving Ack -->
      <item name="AckTime" value="199"/>

      <!-- Wait time before trying to reconnect -->
      <item name="ConWait" value="1000"/>

      <!-- Interval to set fader levels on the audio driver. If set to 0, the
      operation will not be re-scheduled.
      Note that this configuration will overwrite the general configuration with same
      name from AvAutomation settings (Ctrl+Shift+S).
      If commented, the general configuration from AvAutomation will be taken into
      consideration.-->
      <item name="AudioPollInterval" value="0"/>

      <!-- Interval to send an heartbeat to the driver (number of counts until next
      heartbeat will be sent). Set this to -1 to disable sending the heartbeat.
      In this case, AvAutomation will not be able to detect when audio driver
      connection goes down, for example. -->
      <item name="HeartbeatInterval" value="1000"/>
    </Properties>
  </DeviceConfig>

```

16.5.2 SSL Configuration File

The SSL configuration file is the same as used for [Calrec TCP/IP Configuration File](#).