Viz Mosart Administrator Guide

Product Version 3.8.1

January 22, 2018

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.

Last Updated

January 22, 2018

Contents

1 Introduction

This document is a reference guide for use during installation and configuration of Viz Mosart. 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.

This section contains the following topics:

- Related Documents
- Customer Feedback and Suggestions
- Customer Support Requests

1.1 Related Documents

- *Viz Mosart User's Guide*
- *Viz Mosart Media Router Guide*

1.2 Customer Feedback and Suggestions

We encourage suggestions and feedback about our products and documentation.

To give feedback and, or suggestions, please identify your local Vizrt customer support team at www.vizrt.com

1. Click on **Contact** (top of page).
2. The Vizrt office which is nearest to your location will be shown, or select from the list of Vizrt offices.
3. Click on the Contact button for the office you want.
4. Complete the required details in the window that opens.

Note: If this message is for Customer Support, and there is a Support Contract in place, then click on the 'For support requests, please visit our support portal' link in the message window.

A Vizrt representative will contact you as soon as possible.

1.3 Customer Support Requests

Support Requests are supported if you have a valid Service Agreement in operation. Customers who do not have a Service Agreement and would like to set up a Service Agreement should contact their regional sales representative (see Customer Feedback and Suggestions).

When submitting a Support Request, relevant and correct information should be given to Vizrt Support, to make sure that Vizrt Support can give the quickest and best solution to your Support Request.

1.3.1 Before Submitting a Support Request

Before a Support Request is submitted make sure that you:

Read:

- The relevant User Guide or Guides
- The release notes

and Check:

- That the system is configured correctly. Always keep track of all changes, and roll back to previous configuration file versions and test this if the system has been reconfigured recently.
- That you have the specified hardware, tested and recommended versions.

Always refer to your Vizrt Service Level Agreement document.

Submitting a Support Request

When completing a Support Request, add as much information as possible.

Content of a Support Request

The report must contain the following topics:

- **Problem description:** Include a step-by-step description of what the problem is and how to reproduce it. Specify your workflow. Use simple English. Refer to the Naming Convention for Components when describing the effected parts of the system.
- **Expected behavior:** Describe what you expected to happen.
- **Actual behavior:** Describe what actually happened.
- **Screen shots, illustrations and videos:** Use these to augment the message.
- **Software configuration:** Add exact versions of software (-build) used.
- **System locale:** Specify the Region and Language settings of the system, and also the Time Zone setting on the servers and PCs, as this might be different from local time (some global stations using UTC/GMT setting).
- **System log files:** Send the system log files (see Take Snapshot).
- **Crash log files:** Send the error report and crash log files.
- **System Config file**: Send the system configuration files.
- **Hardware configuration:** Add exact versions of hardware used

.

Optional:

- **System setup:** Describe differences in the installation, if any, from the recommended setup.
- **Windows event log files:** Add these log files if deemed necessary or requested by Vizrt.

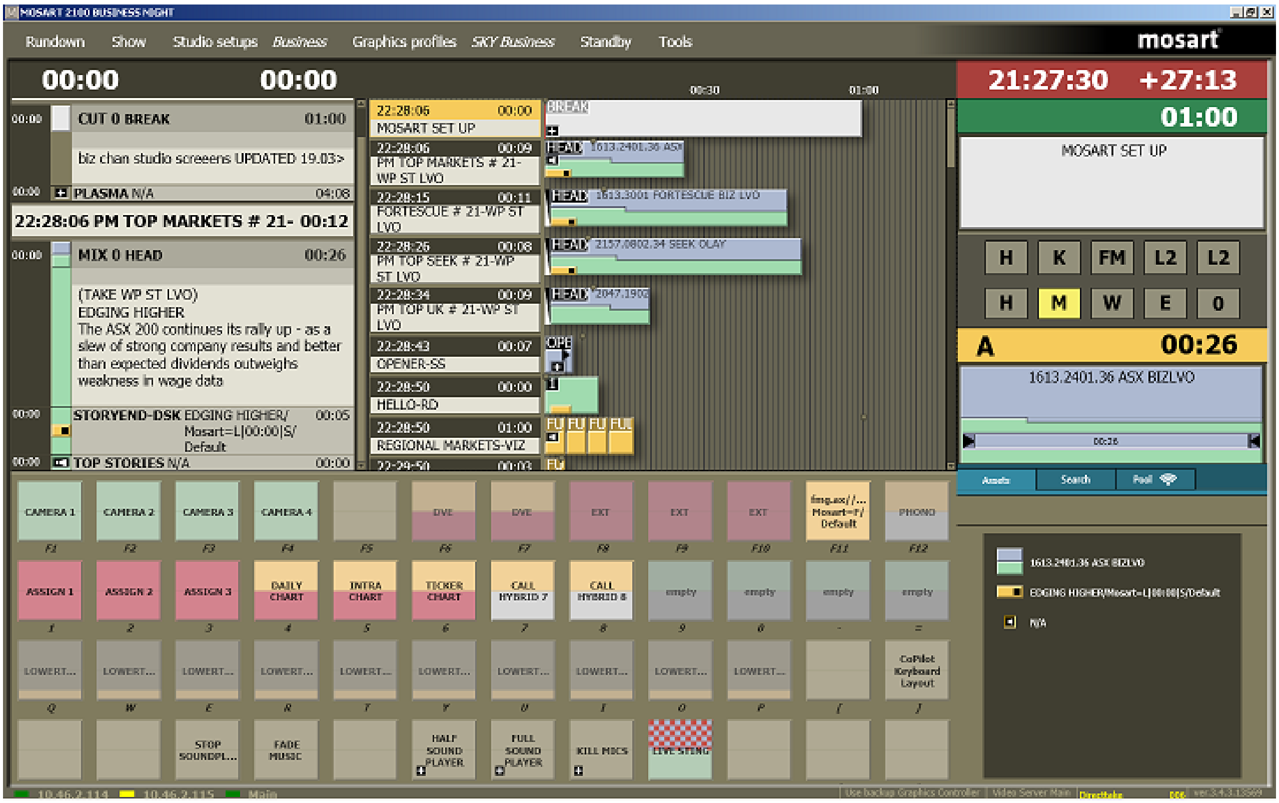
- **System Network:** Add a description of how the network, bandwidth, routers, and switches are configured.

  Always refer to your Vizrt Service Level Agreement document.

To submit a Support Request:

1. On the www.vizrt.com page, click on Support.
2. Click on **Report a case**.
3. Click on **LOG IN** to login to the Customer and Partner portal.
4. At the top of the Case Management page, click on **Report a Case**.
5. In the online form complete the required minimum information (shown by a red asterisk) and click **SAVE**.
6. In the saved Support Case that opens, complete the various text boxes and upload any required documents, files, etc. (see Submitting a Support Request).
   To **track the status** of open support tickets, login to the Customer and Partner portal. Add information or communicate about the cases directly with the support team.

2 About Viz Mosart



The ASX 200 continues its rally up - as a slew of strong company results and better than expected dividends outweighs weakness in wage data

Viz Mosart is a broadcast solution designed for Newsroom environments and 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 there is a standard naming convention for the following applications and components that form Viz Mosart:

- **AVA**: AV Automation Application controlling attached broadcast equipment
- **GUI**: Viz Mosart GUI Main user control application
- **KVM**: Keyboard, Video, Mouse. Application allowing multiple computers sharing the same monitor, keyboard and mouse

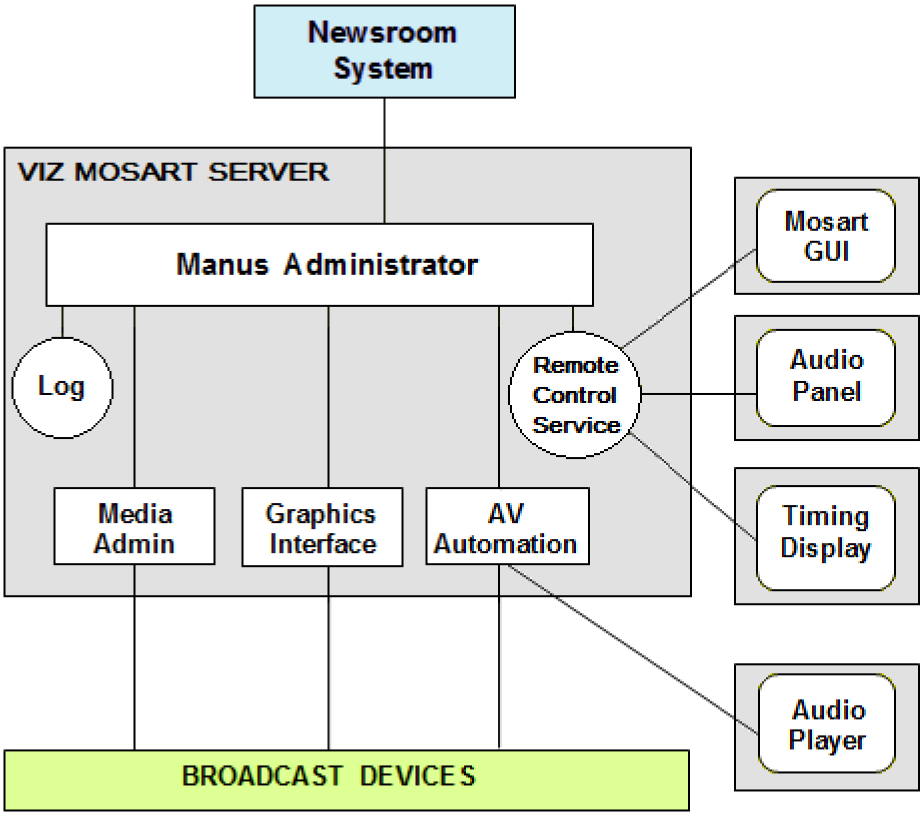
- **Manus Admin**: Manus Administrator Application controlling the Viz Mosart Rundown
- **Media Admin**: Media Administrator Media Administrator. Application for monitoring media objects (clips)
- **MIA**: Viz Mosart Installation Administrator Application that installs the various Viz Mosart components
- **Media Router (MMR)**: Application that shares broadcast devices between control rooms. See Media Router and Mosart Template Database
- **NCS, NRCS**: Newsroom Control System
- **Overlay Graphics Interface**: Application that controls non-Vizrt overlay graphic 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 mentioned 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, like the Timing Display, Audio Panel, and so on. You can have more than one Viz Mosart GUI running at workstations connected to the Mosart server. Normally there will 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. This allows 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

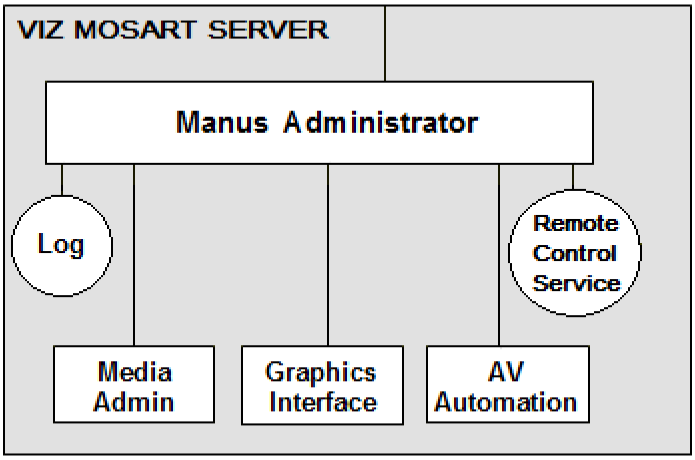
  - **Viz Mosart Client Applications**

- 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
  - Viz Mosart Installation Administrator
  - 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 independent of one another on the same workstation.

Viz Mosart Server supports multiple GUI workstations connected to it at any time. The control room will have 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.

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 *MMConsoleAdmin_2007.exe* (FTP, for iNEWS) or MMConsoleAdmin_MOS.exe (for MOS workflow Newsroom systems)
	Mosart Media Administrator Executable: *MMMediaAdministrator.exe*
	Mosart AV Automation Executable: *MMAVAutomation.exe*
	Mosart Overlay Graphics Executable: *MMOverLayGraphicsInterface.exe*

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 for greater flexibility to handle a variety of broadcast devices and productions.

A typical Viz Mosart Server installation will have the following applications and services.

- Manus Administrator (console application): Handles the connection to the Newsroom system.
- Media Administrator (console application): Application for monitoring media objects (clips) referenced in the current rundown. Also 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 (Wndows service): Service used for generating Viz Mosart command logs. Automatically installed with Viz Mosart Server and Viz Mosart GUI.
- Remote Control Service (Windows service): The service used for all external PCs, like GUI and Timing Display, to connect to Viz Mosart Server. Automatically installed with Viz Mosart Server.

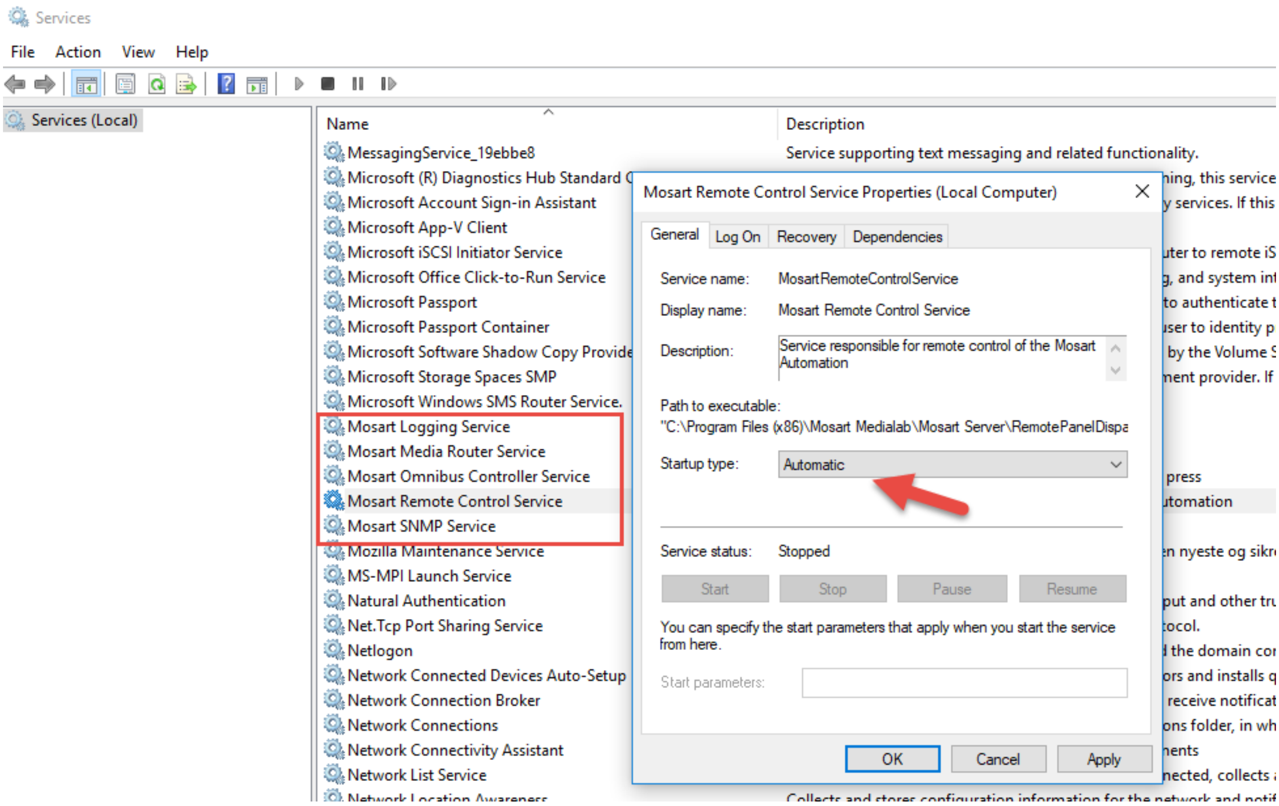
Normally all of these services should be running at all times at the Mosart server. To make sure the required applications and services are started and running you could start them from a convenience batch-script looking something like this:

StartMosartServicesAndGUI.cmd

```
@echo off

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

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

The status of the console applications (*Manus Admin* and *Media Admin*) can of course very easily be inspected at the Mosart server console. Same goes for the Windows applications *AvAutomation* and *OverlayGraphics*.

The RCS (Remote Control Service) and Log Service run as a Windows services and are therefore "hidden" from easy visual inspection. It's recommended to make these services autostart upon Windows startup. There are many ways to do this depending on your preference. An easy way is to use the Windows services application (Start buttoon > Services), locate the required services in the services window, right-click the service and make sure it's properties are set to atuostart as indicated in the screenshot below:



Service supporting text messaging and related functionality.

MessagingService_19ebbe8
Microsoft (R) Diagnostics Hub Standard C
Microsoft Account Sign-in Assistant
Microsoft App-V Client
Microsoft iSCSI Initiator Service
Microsoft Office Click-to-Run Service
Microsoft Passport
Microsoft Passport Container
Microsoft Software Shadow Copy Provide
Microsoft Storage Spaces SMP
Microsoft Windows SMS Router Service.
Mosart Logging Service
Mosart Media Router Service
Mosart Omnibus Controller Service
Mosart Remote Control Service
Mosart SNMP Service
Mozilla Maintenance Service
MS-MPI Launch Service
Natural Authentication
Net.Tcp Port Sharing Service
Netlogon
Network Connected Devices Auto-Setup
Network Connection Broker
Network Connections
Network Connectivity Assistant
Network List Service
Network Location Awareness

Mosart Remote Control Service Properties (Local Computer)

Service name: MosartRemoteControlService

Display name: Mosart Remote Control Service

Description: Service responsible for remote control of the Mosart Automation

Path to executable:
"C:\Program Files (x86)\Mosart Medialab\Mosart Server\RemotePanelDispa

Startup type: Automatic

Service status: Stopped

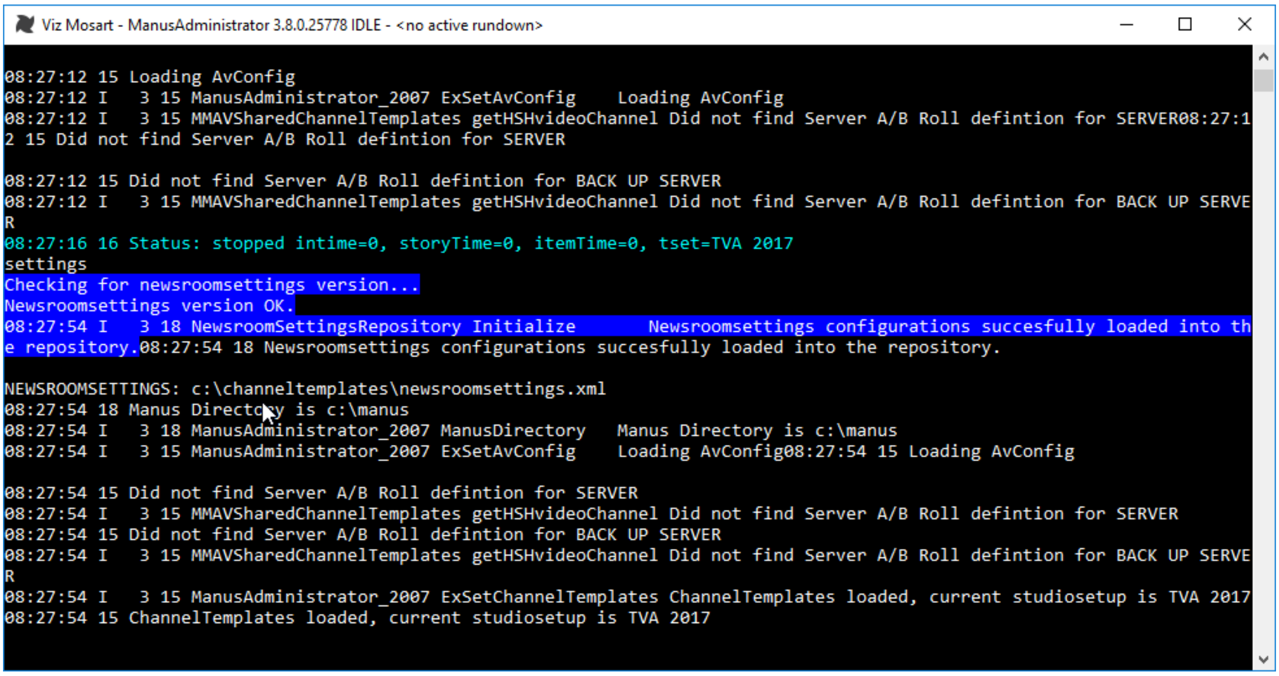
You can specify the start parameters that apply when you start the service from here.

Start parameters:

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 Logging service and RCS is running, the other Mosart service 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 System, and Manus Administrator is the center application in Viz Mosart. It controls the current rundown in the Viz Mosart GUI and receives rundowns from the attached Newsroom System. It issues events according to the different story items in the rundown, and is dynamically updated when changes are made in the NCS. Manus Administrator receives information from the NCS but does not send updates back, in order to avoid the systems fighting over who is in control of the rundown.

Manus Administrator handles the connection to the Newsroom system, and also the execution of commands at the right time according to GUI input and template configurations.

Manus 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

See also Manus Administrator Configuration.

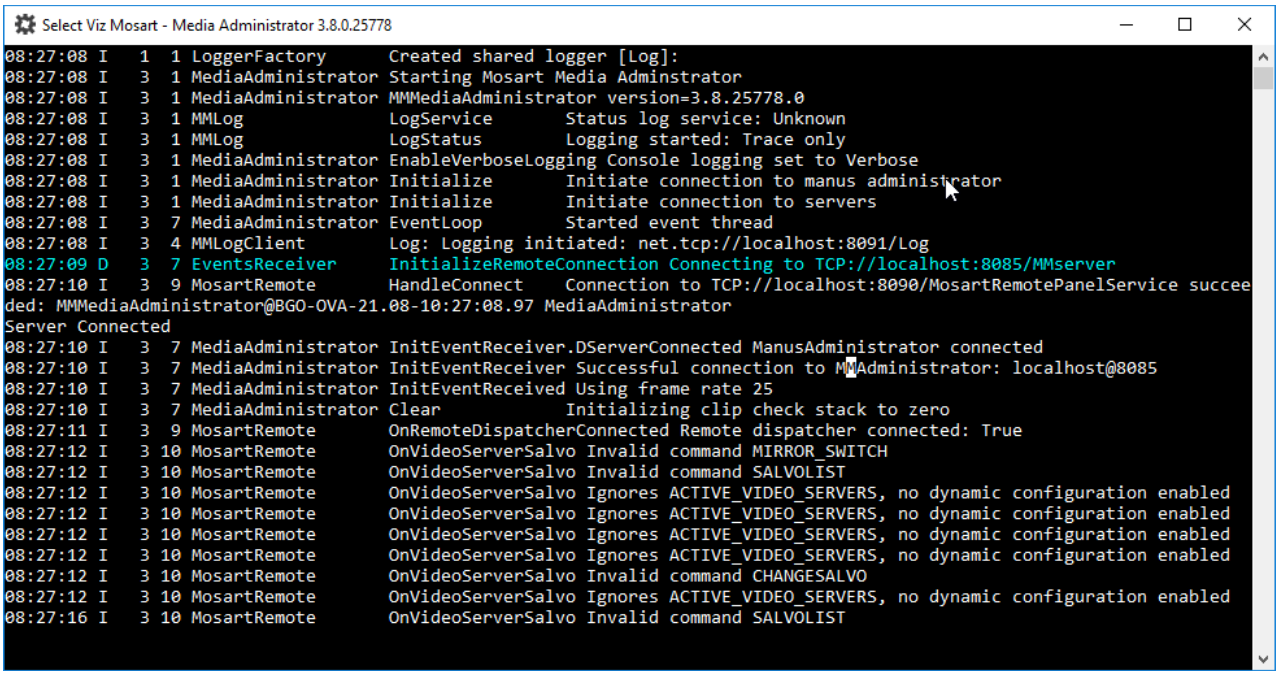
Variants

The variants of Manus Administrator suit different broadcast environments, and have slightly different settings. 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 has been known by several names: MMConsoleAdmin, ManusAdmin, ManusAdministrator or just Manus.

2.3.3 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 Adminstrator
08:27:08 I   3  1 MediaAdministrator MMMediaAdministrator 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 succee
ded: MMMediaAdministrator@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 MMAdministrator: 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
```

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

Media Administrator is mainly used for monitoring and searching video clips on the attached video servers.

- 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 clip properties like clip duration. The clip status for video servers is reflected 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. The length of the horizontal bar in the Viz Mosart GUI reflects the clip length. For other media objects, the Viz Mosart GUI may present the clip status in various ways.
- Makes it possible to **search** for clips on the video server. This functionality is used by the Viz Mosart GUI media search window, making it possible to add clips to the rundown without use of a newsroom system. Other media objects may also be searched using this application such as 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.

See also Media Administrator Configuration.

2.3.4 Overlay Graphics Interface

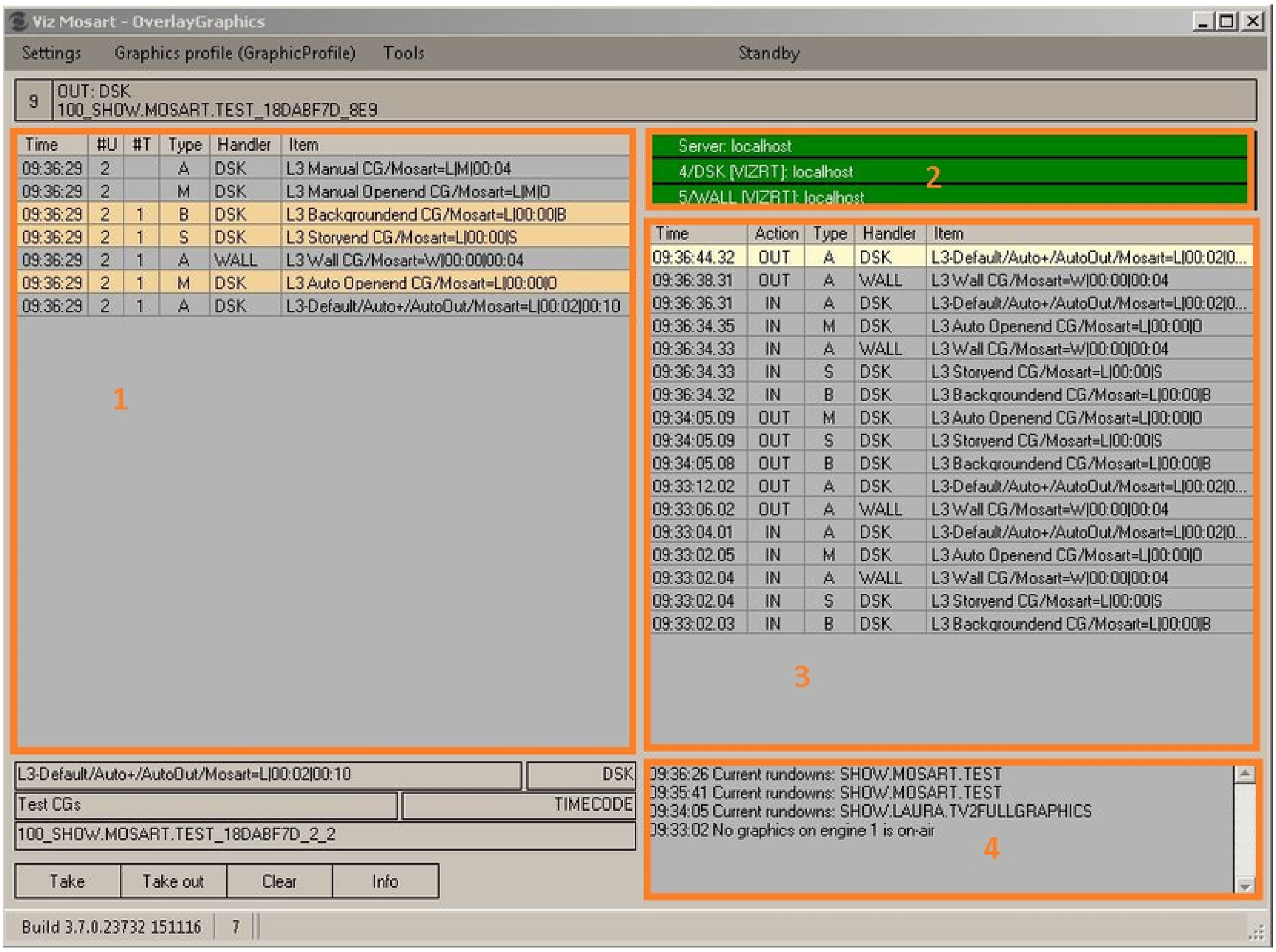
An overlay graphics interface is used for controlling and monitoring overlay graphics for graphic engines.

The two overlay graphics interfaces are:

- Overlay Graphics Interface (recommended).
- Trio Interface (to be deprecated, kept for backwards compatibility only).

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

Overlay Graphics Interface


Viz Mosart - OverlayGraphics
Settings
Graphics profile (GraphicProfile)
Tools
Standby
1
2
3
4
Take
Take out
Clear
Info

9 OUT: DSK
100_SHOW.MOSART.TEST_18DABF7D_8E9

Time	#U	#T	Type	Handler	Item
09:36:29	2		A	DSK	L3 Manual CG/Mosart=L|M|00:04
09:36:29	2		M	DSK	L3 Manual Openend CG/Mosart=L|M|O
09:36:29	2	1	B	DSK	L3 Backgroundend CG/Mosart=L|00:00|B
09:36:29	2	1	S	DSK	L3 Storyend CG/Mosart=L|00:00|S
09:36:29	2	1	A	WALL	L3 Wall CG/Mosart=W|00:00|00:04
09:36:29	2	1	M	DSK	L3 Auto Openend CG/Mosart=L|00:00|O
09:36:29	2	1	A	DSK	L3-Default/Auto+/AutoOut/Mosart=L|00:02|00:10

Server: localhost
4/DSK [VIZRT]: localhost
5/WALL [VIZRT]: localhost

Time	Action	Type	Handler	Item
09:36:44.32	OUT	A	DSK	L3-Default/Auto+/AutoOut/Mosart=L|00:02|0...
09:36:38.31	OUT	A	WALL	L3 Wall CG/Mosart=W|00:00|00:04
09:36:36.31	IN	A	DSK	L3-Default/Auto+/AutoOut/Mosart=L|00:02|0...
09:36:34.35	IN	M	DSK	L3 Auto Openend CG/Mosart=L|00:00|O
09:36:34.33	IN	A	WALL	L3 Wall CG/Mosart=W|00:00|00:04
09:36:34.33	IN	S	DSK	L3 Storyend CG/Mosart=L|00:00|S
09:36:34.32	IN	B	DSK	L3 Backgroundend CG/Mosart=L|00:00|B
09:34:05.09	OUT	M	DSK	L3 Auto Openend CG/Mosart=L|00:00|O
09:34:05.09	OUT	S	DSK	L3 Storyend CG/Mosart=L|00:00|S
09:34:05.08	OUT	B	DSK	L3 Backgroundend CG/Mosart=L|00:00|B
09:33:12.02	OUT	A	DSK	L3-Default/Auto+/AutoOut/Mosart=L|00:02|0...
09:33:06.02	OUT	A	WALL	L3 Wall CG/Mosart=W|00:00|00:04
09:33:04.01	IN	A	DSK	L3-Default/Auto+/AutoOut/Mosart=L|00:02|0...
09:33:02.05	IN	M	DSK	L3 Auto Openend CG/Mosart=L|00:00|O
09:33:02.04	IN	A	WALL	L3 Wall CG/Mosart=W|00:00|00:04
09:33:02.04	IN	S	DSK	L3 Storyend CG/Mosart=L|00:00|S
09:33:02.03	IN	B	DSK	L3 Backgroundend CG/Mosart=L|00:00|B

L3-Default/Auto+/AutoOut/Mosart=L|00:02|00:10 DSK
Test CGs TIMECODE
100_SHOW.MOSART.TEST_18DABF7D_2_2

```
09:36:26 Current rundowns: SHOW.MOSART.TEST
09:35:41 Current rundowns: SHOW.MOSART.TEST
09:34:05 Current rundowns: SHOW.LAURA.TV2FULLGRAPHICS
09:33:02 No graphics on engine 1 is on-air
```

Build 3.7.0.23732 151116 7

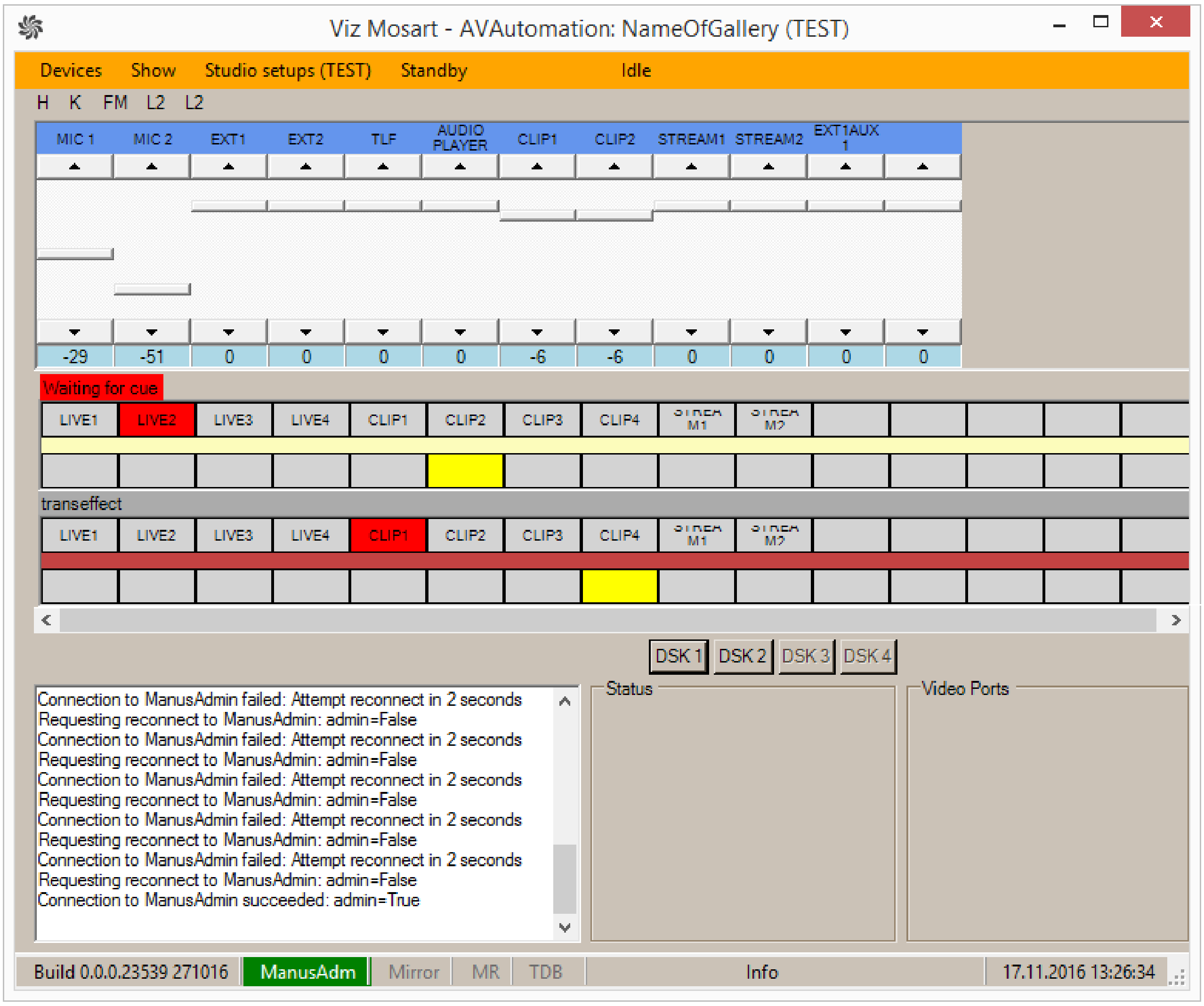
The Overlay Graphics Interface is used for controlling and monitoring overlay graphics for all Viz Mosart approved graphic engines. Configure it using the Overlay Graphics Configuration. For a list of supported graphics devices, see Overlay Graphics Types.

Trio Interface

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

For details, see Trio Interface.

2.3.5 AV Automation



```
Connection to ManusAdmin failed: Attempt reconnect in 2 seconds
Requesting reconnect to ManusAdmin: admin=False
Connection to ManusAdmin failed: Attempt reconnect in 2 seconds
Requesting reconnect to ManusAdmin: admin=False
Connection to ManusAdmin failed: Attempt reconnect in 2 seconds
Requesting reconnect to ManusAdmin: admin=False
Connection to ManusAdmin failed: Attempt reconnect in 2 seconds
Requesting reconnect to ManusAdmin: admin=False
Connection to ManusAdmin failed: Attempt reconnect in 2 seconds
Requesting reconnect to ManusAdmin: admin=False
Connection to ManusAdmin succeeded: admin=True
```

AV Automation (MMAVAutomation) is where all Viz Mosart connected broadcast devices (except Overlay Graphics engines) are controlled. 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 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.

See also AV Automation and AV Automation Device Properties.

2.3.6 Log Service and Log Viewer

Log Service

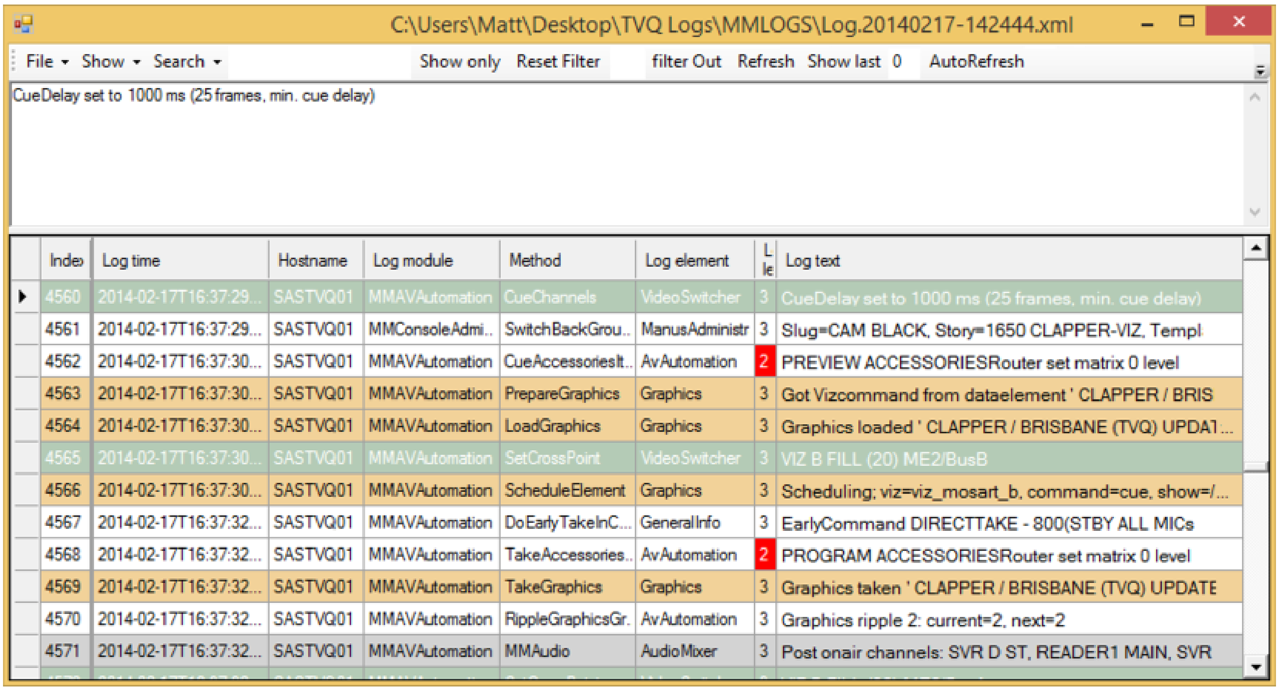
Service used for generating Viz Mosart command logs. Automatically installed with Viz Mosart Server and Viz Mosart GUI.

Log Viewer

Logviewer

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

Log entries are color coded to allow for quicker command visualization in the file. Microsoft Excel may also be used for displaying Viz Mosart log files (XML), and gives the possibility of several layers of filtering and also plotting graphs.



CueDelay set to 1000 ms (25 frames, min. cue delay)

Inde>	Log time	Hostname	Log module	Method	Log element	L le	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) UPDAT...
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

2.3.7 Remote Control Service

Remote Control Service (**RCS**, sometimes refered to as RCPS) is the service that allows all external PCs, like Viz Mosart GUI and Timing Display, to connect to Viz Mosart Server. Automatically installed with Viz Mosart Server.

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.

The **additional** Viz Mosart Client applications are:

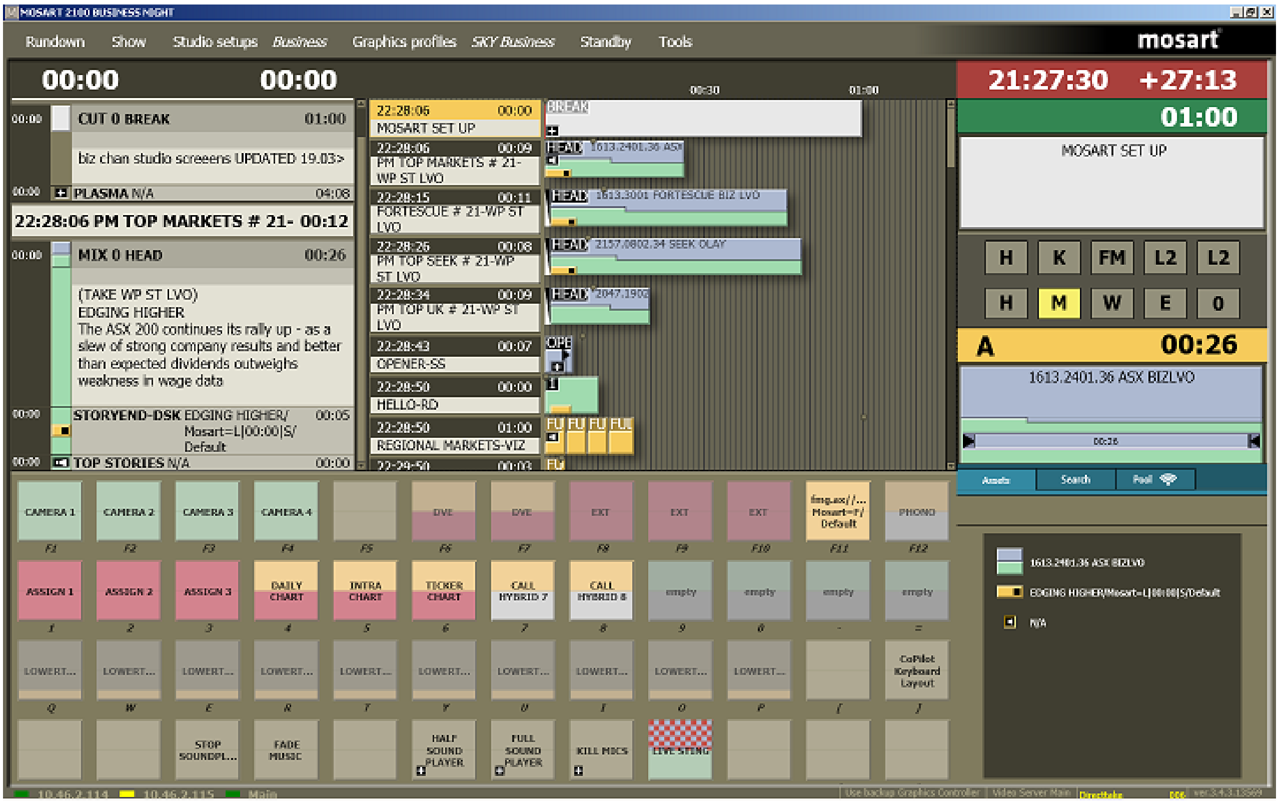
- ActiveX
- iNews Timer

- 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 be running the same Viz Mosart version with a standard OS across all Viz Mosart client machines.

2.4.1 Viz Mosart GUI

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



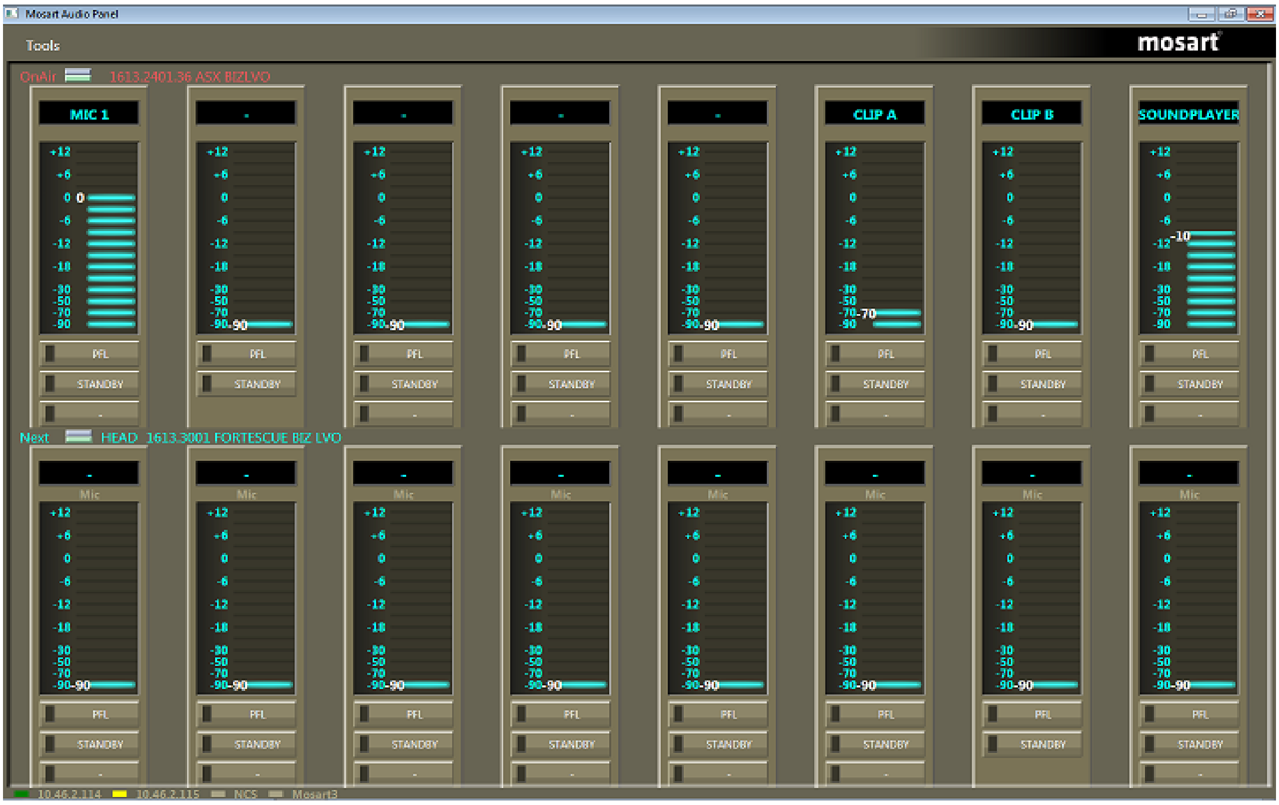
biz chan studio screeens UPDATED 19.03>

(TAKE WP ST LVO) EDGING HIGHER The ASX 200 continues its rally up - as a slew of strong company results and better than expected dividends outweighs weakness in wage data

Alternate configurations may also host a fader panel and a Timing Display, it is preferable each Viz Mosart Client application run on a standalone networked PC.

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

2.4.2 Audio Panel



Audio Panel (Server)

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

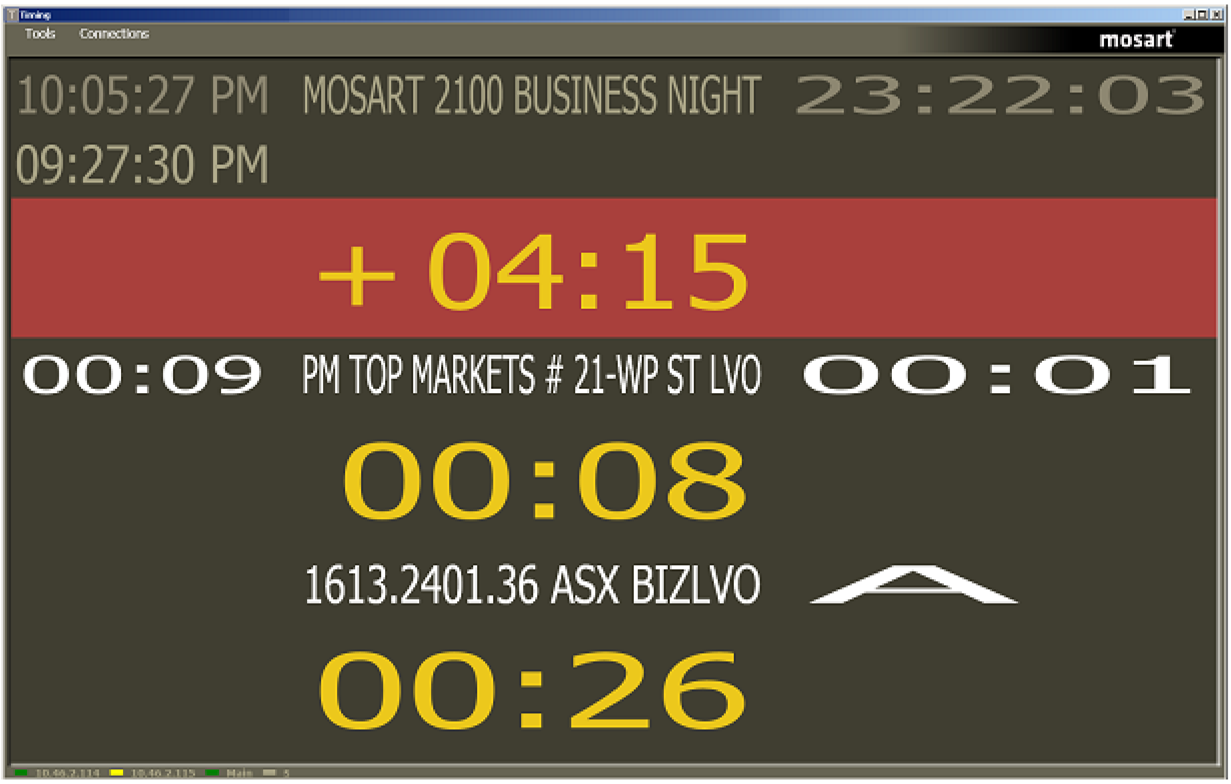
Audio Panel (Client)

Mosart Audio Panel

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

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

2.4.3 Timing Display



10:05:27 PM MOSART 2100 BUSINESS NIGHT 23:22:03

09:27:30 PM

+04:15

00:09 PM TOP MARKETS # 21-WP ST LVO 00:01

00:08

1613.2401.36 ASX BIZLVO A

00:26

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

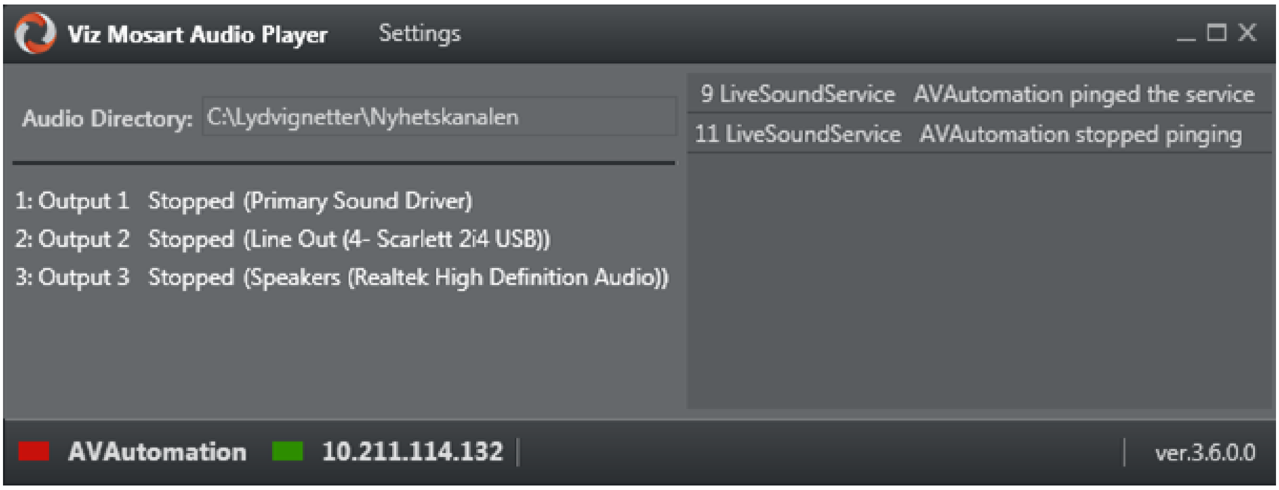
The timing display is synchronized to the current rundown in the Viz Mosart GUI, and you can run as many customized Timing Display applications as you need for the production.

The Timing Display is described in detail in the *Viz Mosart User’s 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 to play out audio files that are used on a regular basis, such as for openers and audio-beds.

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.



Audio Directory: C:\Lydvignetter\Nyhetskanalen

```
9 LiveSoundService   AVAutomation pinged the service
11 LiveSoundService  AVAutomation stopped pinging
```

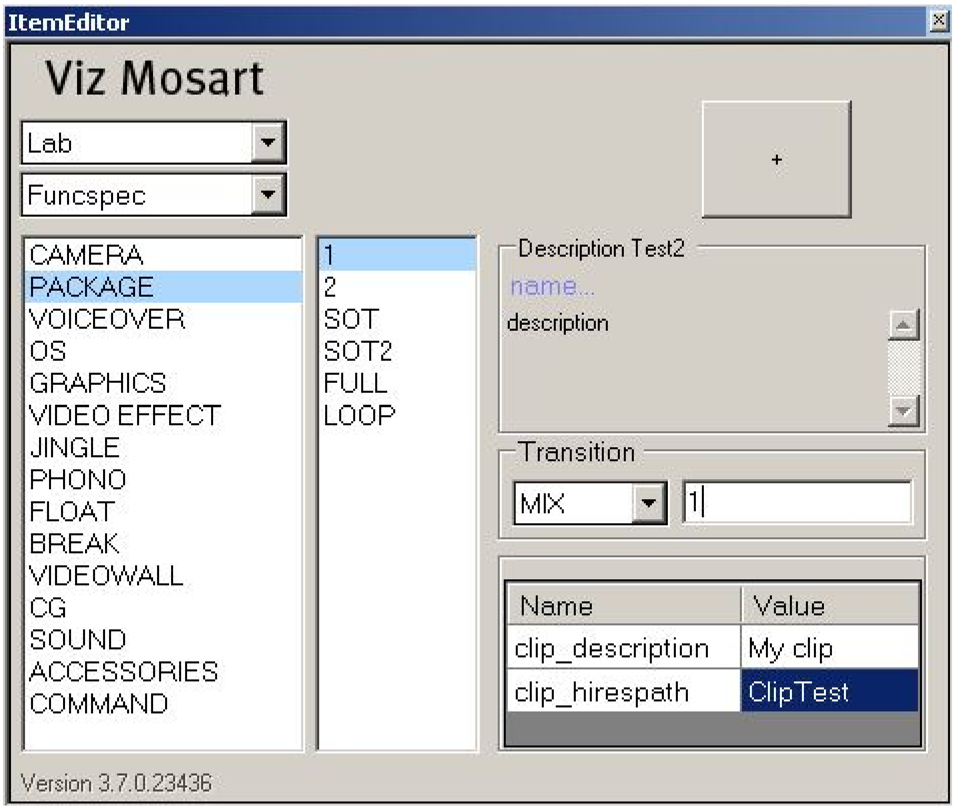
```
1: Output 1   Stopped (Primary Sound Driver)
2: Output 2   Stopped (Line Out (4- Scarlett 2i4 USB))
3: Output 3   Stopped (Speakers (Realtek High Definition Audio))
```

🟥 AVAutomation 🟩 10.211.114.132 | ver.3.6.0.0

Note: The Audio Player replaces the discontinued Soundfile Player.

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

2.4.5 ActiveX



Lab

Funcspec

CAMERA
PACKAGE
VOICEOVER
OS
GRAPHICS
VIDEO EFFECT
JINGLE
PHONO
FLOAT
BREAK
VIDEOWALL
CG
SOUND
ACCESSORIES
COMMAND

1
2
SOT
SOT2
FULL
LOOP

name...
description

Transition: MIX 1

Name	Value
clip_description	My clip
clip_hirespath	ClipTest

Version 3.7.0.23436

The **ActiveX** is used to insert Viz Mosart template information into the Newsroom System script. It is 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 gives users the ability to 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 Vision Mixer Setup.

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

2.4.6 iNews Timer

The iNews Timer allows for accurate timing of the production in iNews. It is used to synchronize the current on-air story in the Viz Mosart GUI to the correct corresponding entry in iNews.

2.5 Other Viz Mosart Applications

This section contains the following topics:

- Media Router and Mosart Template Database
- Omnibus Controller
- Viz Mosart Installation Administrator
- 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 is for Viz Mosart installations running Omnibus equipment. This is a service that allows communication with Omnibus systems.

2.5.3 Viz Mosart Installation Administrator

Viz Mosart has its own installation assistant, known as the Viz Mosart Installation Administrator (MIA).

2.5.4 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, will be able to use this service.

2.5.5 Test Suite

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

The Test Suite has its own GUI interface, and contains the following tools:

- **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. These applications can be manually installed by running standard Microsoft installers (.MSI files), or you can use the Viz Mosart Installation Administrator (MIA).

This section contains the following topics:

- Prerequisites
- Viz Mosart Installation Files
- Version Numbering
- Manual Installation
- Viz Mosart Installation Administrator
- Upgrading Viz Mosart

3.1 Prerequisites

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

This section contains:

- Server Prerequisites
- Microsoft .NET Framework 4.5
    - To determine whether Microsoft .NET Framework 4.5 is installed
    - Installing Microsoft .NET Framework 4.5
- Microsoft Visual C++ 2012 Redistributable Package (x86)
    - Installing Microsoft Visual C++ 2012 Redistributable Package
- Connections

3.1.1 Server Prerequisites

The following are the prerequisites for all machines running Viz Mosart software

Prerequisite	Target	Description
Microsoft Windows Server 2008 R2 or later	Viz Mosart Server	Operating system
Microsoft Windows 7 Professional or later	Viz Mosart Client	Operating system
Microsoft .NET Framework 4	All	The Viz Mosart installers are currently dependent on *Microsoft . NET Framework 4.5*, and Microsoft . NET Framework 3.5.Most of the Viz Mosart installers will abort the installation process if .NET 4.5 is not detected.

Prerequisite	Target	Description
vcredist_x86.exe	Viz Mosart Server	*Microsoft Visual C++ 2012 Redistributable Package (x86)*
Windows Folder Structure	All	The following folder must be available, containing Viz Mosart installers, before installation: C:\mosart\installers
Windows Firewall	All	For some versions of Windows, it may be necessary to open ports. Either completely disable Windows firewall, or alternatively open for Ping and ports 8080-8099. If Mosart Template Database is used, port 3306 also must be opened.
Windows Audio	Viz Mosart Server	If MIDI devices are connected via direct conversion serial adapters, Windows Audio feature must be enabled on the server.Note: If using Windows Remote Desktop, audio must be set to play on the server, not the remote PC, as the Audio configuration otherwise will be disturbed.
Windows Server Type	Viz Mosart Server	It may be favorable to configure the Windows Server running the Viz Mosart Server software to “Application Server”, as this automatically pre-configures several features and also automatically sets a detailed .NET configuration.
Server Specifications	All	For more details, see the *Viz Mosart Recommended Computer Platform* manual.

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

3.1.2 Microsoft .NET Framework 4.5

Microsoft .NET 4.5 must be installed on the system before installing any Viz Mosart Applications.

To determine whether Microsoft .NET Framework 4.5 is installed

1. Navigate to the directory below:

```
%SystemRoot%\Microsoft.NET\Framework
```

2. The presence of a v4.5.xxxxx folder is basically sufficient, however you may perform the next step in order to investigate the .NET details further.
3. Open the v4.5.xxxxx folder and perform the following:
   - Right-click any .dll file, and then click **Properties**.
   - Click the **Details** tab.
   - Make sure that the **File Version** is 4.5.xxxxx, or later.
   - The **Product Name** should be Microsoft .NET Framework.

   Note: If presence of .NET 4.5 is not detected, installation is mandatory.

Installing Microsoft .NET Framework 4.5

Obtain the Microsoft .NET Framework 4.5 (or later) installer from the Microsoft Download Center.

Run the installer and follow the on screen prompts.

The installer may ask you to restart the machine on completion. Please restart the machine when prompted.

3.1.3 Microsoft Visual C++ 2012 Redistributable Package (x86)

The *Microsoft Visual C++ 2012 Redistributable Package (x86)* must be installed on the system before installing the Viz Mosart Server software.

Note: This is a 32-bit program.

IMPORTANT! Do *not* use the 2013 version of the Microsoft Visual C++ Redistributable Package.

Installing Microsoft Visual C++ 2012 Redistributable Package

Obtain the Visual C++ 2012 Redistributable Package (x86) installer from the Microsoft Download Center.

On clicking download, you will be prompted to select the installer you require. Please ensure that you select the *32-bit (x86) *version, as follows:

- `VSU_4\vcredist_x86.exe`
  Run the installer and follow the on screen prompts.

The installer may ask you to restart the machine on completion. Please restart the machine when prompted.

3.1.4 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 a software package split into various server and client specific applications, connecting to broadcast devices and controlling the Viz Mosart rundown.

The table below shows a list of all available Viz Mosart installers:

Installer	Machine Role	Description
MosartActiveXInstaller	NCS Client	Installer containing the *ActiveX*, used for displaying Viz Mosart Templates in the NCS client.
MosartAudioPanelInstaller	Audio Client (GUI or standalone PC)	Installer containing the *Audio Panel*, an application for controlling audio faders, both on air and in preview.
MosartAudioPlayerInstaller	Viz Mosart Server or Audio Client	Installer containing the *Audio Player* , typically installed on a computer, or Viz Mosart Server, attached to the newsroom audio mixer.Note: Audio Player is a replacement for the discontinued Soundfile Player.
MosartGUIInstaller	Viz Mosart GUI Client	Installer containing the *Viz Mosart GUI* application, which is the Viz Mosart user interface.
MosartINEWSTimerInstaller	iNews Client	Installer containing the *iNews Timer* , an application used to create accurate timing of the production in iNews.
MosartInstallationAdministratorInstaller	All Viz Mosart PCs	Installer containing the Viz Mosart Installation Administrator, an installation assistant for all Viz Mosart software.
MosartLogInstaller	Viz Mosart Server	Installer containing the *Log Viewer*, an application used to decipher Viz Mosart logs.
MosartMediaRouterAdminInstaller	Viz Mosart Database Server	Installer containing the Media Router Administrator.For more details, see Media Router and Mosart Template Database Admin Guide.

Installer	Machine Role	Description
MosartMediaRouterInstaller	Viz Mosart Database Server	Installer containing the Media Router.For more details, see Media Router and Mosart Template Database Admin Guide.
MosartOminibusControllerInstaller	Viz Mosart Server	Installer containing the *Omnibus Controller*, an application used for Omnibus Server control.
MosartServerInstaller	Viz Mosart Server	Installer containing the various Viz Mosart Server components; Manus Administrator, Media Administrator, AV Automation, and Overlay Graphics Interface.
MosartSNMPServiceInstaller	Viz Mosart Server	Installer containing the *SNMP Service*, used for monitoring the Viz Mosart Server suite via SNMP.
MosartTestSuite	Viz Mosart Server	Installer containing the *Test Suite*, test utilities used when installing Viz Mosart.
MosartTimingDisplay2Installer	Timing Client	Installer containing the *Timing Display*, 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 will also carry 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\MosartServerInstaller.3.6.1.12345.msi`

Here we can see that the computer has a folder for the Viz Mosart 3.6.1.12345 release, containing associated installers.
Every significant code change of Viz Mosart will result in a change to the Viz Mosart Version Number. This will be increasing 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 Viz Mosart (and upgrading to new Viz Mosart versions), it is recommended to use the Viz Mosart Installation Administrator. It is however, possible to run the Viz Mosart Installation Files manually from Windows Explorer.

3.4.1 To perform a manual installation

1. Download all relevant Viz Mosart Installation Files to the preferred location.

   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\3.7.0.24523** for Mosart 3.7.0 build 24523
2. Double-click an MSI installation file, and follow the prompts to complete the installation.
3. Repeat the previous step for all relevant installation files.

3.5 Viz Mosart Installation Administrator

Viz Mosart has its own installation assistant, known as the Viz Mosart Installation Administrator (MIA).

Note: The Viz Mosart Installation Administrator is beta software. Some functionality in this software is not yet fully implemented.

This section contains the following topics:

- Installing the Viz Mosart Installation Administrator
- Viz Mosart Installation Administrator Interface
- Semi-Automated Installation
- Settings
- Take Snapshot
- Replace UserConfig
- Start Mosart Application
- SysInternals Suite
- Backup Files
- Download Installers
- Stop Services
- Uninstall Mosart
- Install Mosart
- Start Services

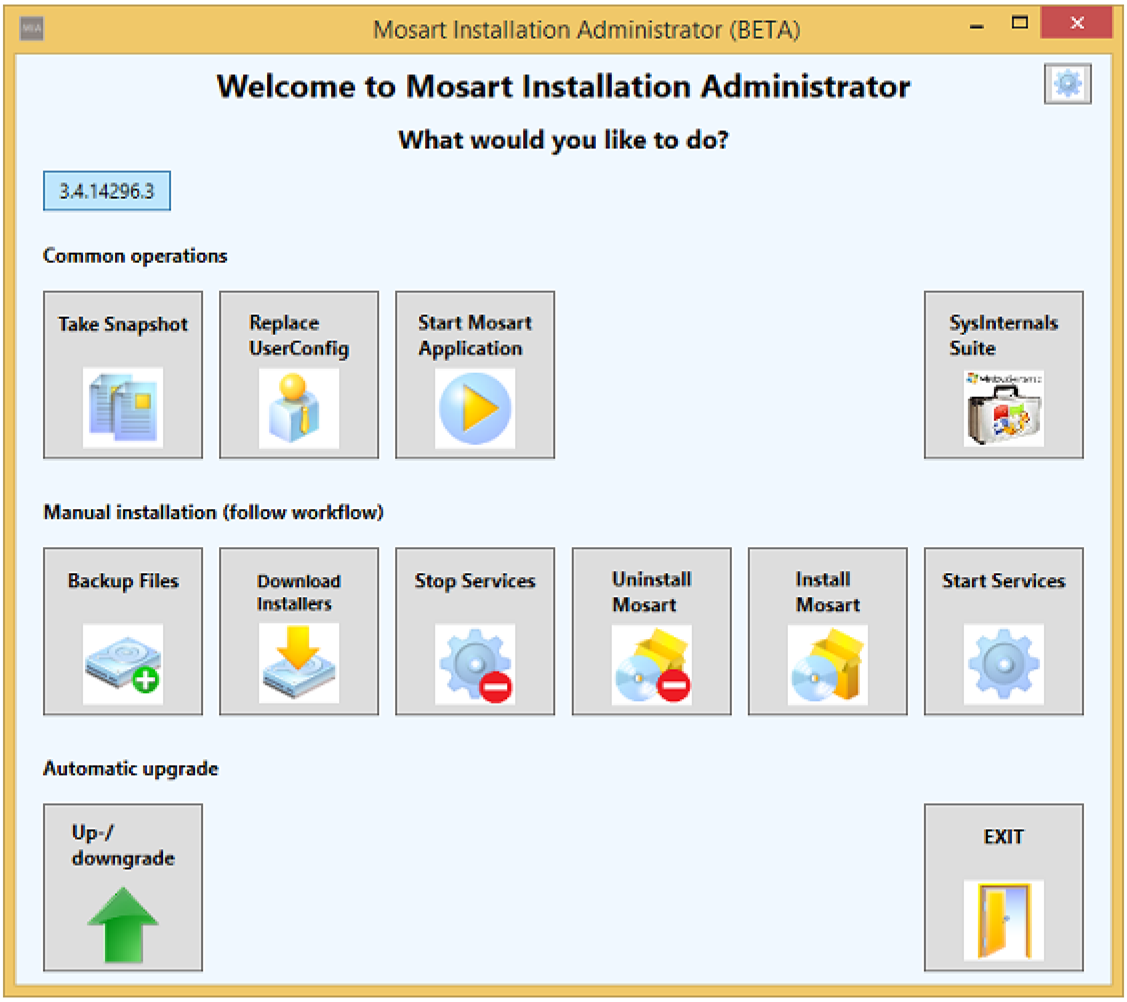
3.5.1 Installing the Viz Mosart Installation Administrator

Before the various Viz Mosart Applications can be installed using the Viz Mosart Installation Administrator (MIA), the installation assistant must first be installed itself.

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 the installation. Repeat this installation process on all PCs where you plan to install any of the Viz Mosart Applications.

3.5.2 Viz Mosart Installation Administrator Interface

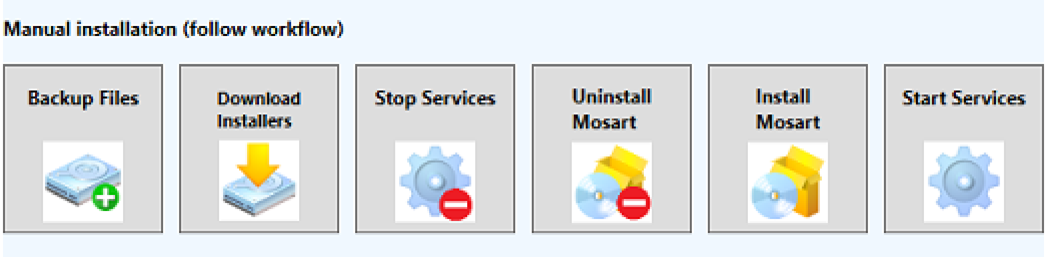


The MIA is a very straightforward tool to install Viz Mosart, both for the first time and later on when performing an upgrade. The various options are described below.

PLEASE NOTE

- The "Download Installers" section in MIA is deprecated and will be removed in future versions. Instead you should use an FTP client such as FileZilla or the embedded ftp client in recent Windows Explorer to download the .MSI installers from `ftp://download.vizrt.comproducts/VizMosart/Latest Version` using your customer credentials supplied by your Vizrt representative. Download all MSI installers in the directory using ftp to the local disk path `c:\Mosart\Installers\VERSION`
  So assuming you are downloading 3.8.0.25837 you should download all installers to `c:\Mosart\Installers\3.8.0.25837`
- The sysinternals suite download in MIA is also deprecated and will be removed in future version. The sysinternals-suite is a free and valuable suite of tools. If you would like to get more info and possibly download it visit https://docs.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite.

3.5.3 Semi-Automated Installation



By using the steps in the manual installation workflow, the .Viz Mosart Installation Administrator v3.8 can be used to run a semi-automated installation process. The buttons are placed in the order the user would normally use when upgrading an existing Viz Mosart installation.

New Installs

If Viz Mosart has not been installed before, and installer files are in the proper installation location, the user can start from the **Install Mosart** button.

Upgrades

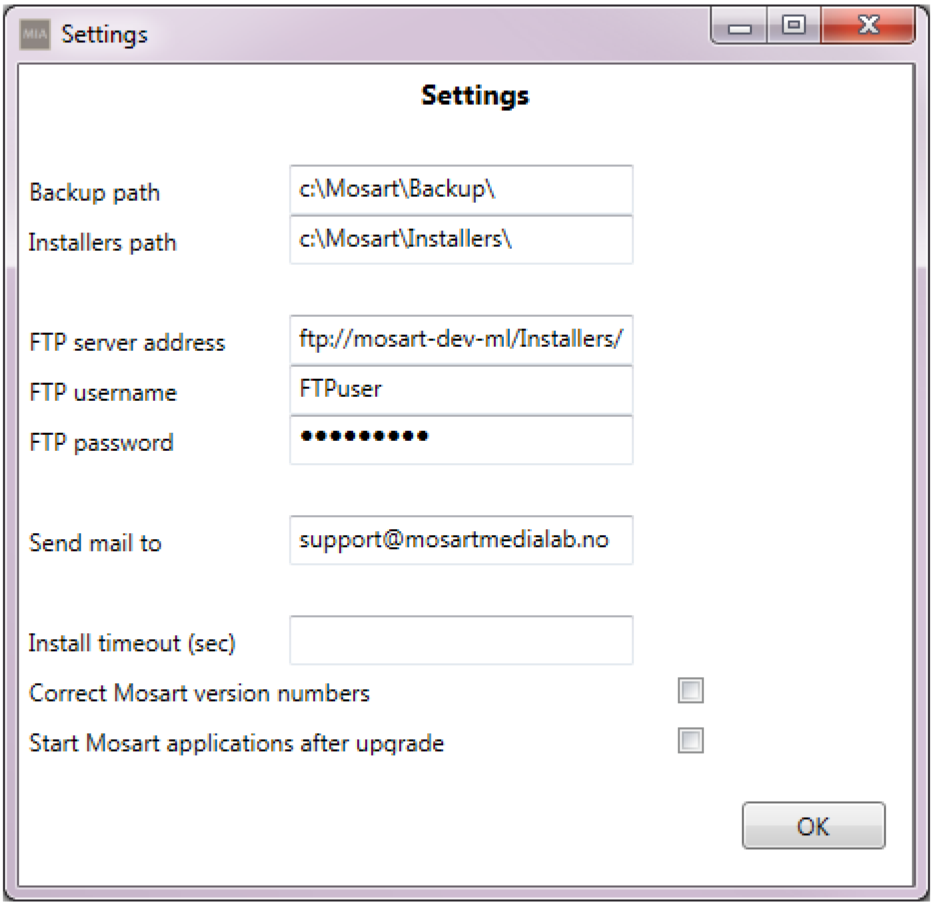
The easy to follow six step process, running from left to right, is used when performing a semi-automated upgrade. Alternatively, use the one-step operation by simply pressing the automatic upgrade button found on the lower left of the main window.

Note: If Viz Mosart has previously been manually installed on the system you are using, without 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, making MIA unable to locate previous manually installed Viz Mosart Applications.

3.5.4 Settings

The Settings are accessed by clicking on the gear icon, located at the top right of the Viz Mosart Installation Administrator Interface.

The following figure shows the default settings.



Settings

Backup path: c:\Mosart\Backup\

Installers path: c:\Mosart\Installers\

FTP server address: ftp://mosart-dev-ml/Installers/

FTP username: FTPuser

FTP password: ●●●●●●●●●

Send mail to: support@mosartmedialab.no

Install timeout (sec):

Correct Mosart version numbers □

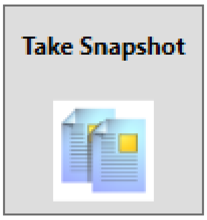
Start Mosart applications after upgrade □

Note: All paths not found in the .Viz Mosart Installation Administrator v3.8 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.
- **FTP server address:** Defines the FTP server address. Normally **ftp.vizrt.com**
- **FTP username:** Defines the FTP username.
- **FTP password:** Defines the FTP password.
- **Send mail to:** Defines the e-mail address where error reports should be sent when troubleshooting.
- **Install timeout (sec):** Viz Mosart Services will try to be started/stopped for the defined number of seconds.

3.5.5 Take Snapshot



A snapshot is a current copy of Viz Mosart logs and user configuration files, and is helpful for Viz Mosart Support to assist in diagnosing any issues you may encounter.

The user must specify a date and time, which must be in the past, for the snapshot.

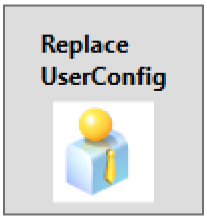
Files gathered in this operation are as follows:

- *C:\MMLogs*
- *C:\MMLogs\MosLog*
- *C:\MMLogs\AsRunLog*
- *%appdata%\..\Local\Mosart_Medialab\.\user.config*

Once complete, the MIA will copy the files to C:\Mosart\Backup\ and compress the files in .ZIP format.

An e-mail interface will appear, with the .ZIP attached, for the user to send the file to Viz Mosart Support.

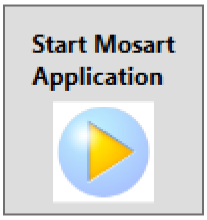
3.5.6 Replace UserConfig



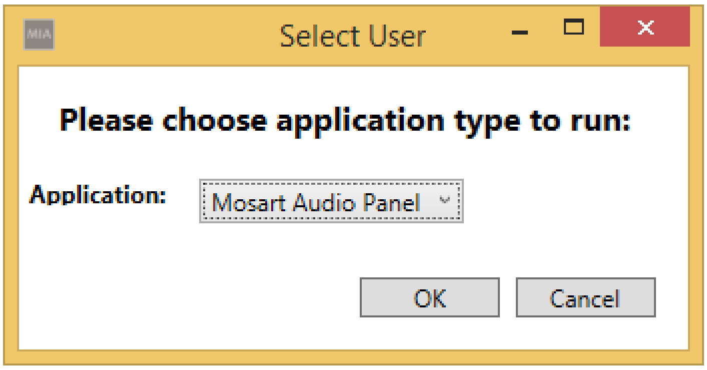
This is intended to replace the current version of the user.config files with the selected version of user.config files.

Note: This must not be done without backing up files first.

3.5.7 Start Mosart Application



The .Viz Mosart Installation Administrator v3.8 provides an easy launch of any installed Viz Mosart application. This feature only allows for one application to be started at a time.



Please choose application type to run:

Application: Mosart Audio Panel

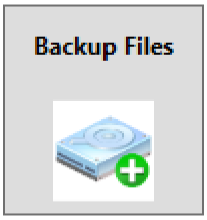
3.5.8 SysInternals Suite

This button opens the system default Internet browser at the download site for the Microsoft SysInternal Suite of Utilities.

These utilities are a collection of selected utilities containing troubleshooting tools for installed software on the Windows platform.

Note: These tools are useful when assisting Viz Mosart Support in troubleshooting software issues. If possible, please have the SysInternal Suite already installed on the Viz Mosart Server.

3.5.9 Backup Files



This option will backup all user.config files from folders with current version numbers and all files located in `C:\channeltemplates`.

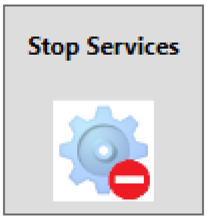
The files will be copied to the specified backup path in the with the Viz Mosart version/username as folder name.

3.5.10 Download Installers

Note: The download feature of the MIA is not yet implemented.

The installers must be copied to the correct repository (see Settings).

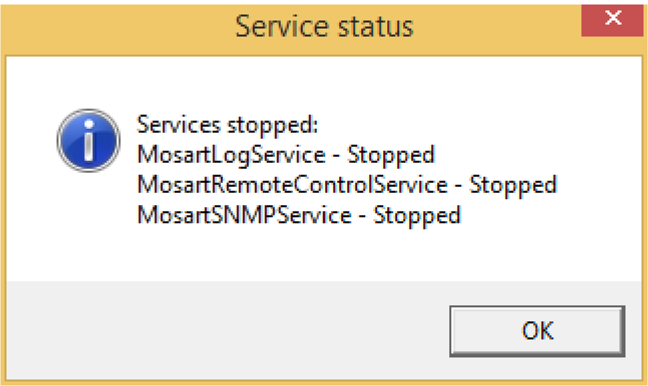
3.5.11 Stop Services



This button will attempt to stop all Viz Mosart services installed with a timeout of 6 seconds.

Tip: The timeout length can be adjusted in the installation assistant’s Settings.

A dialog box will appear with the current service status.

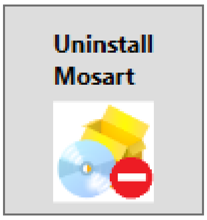


Service status

Services stopped:
MosartLogService - Stopped
MosartRemoteControlService - Stopped
MosartSNMPService - Stopped

Repeat if necessary, alternatively perform a manual stop within Windows Services.

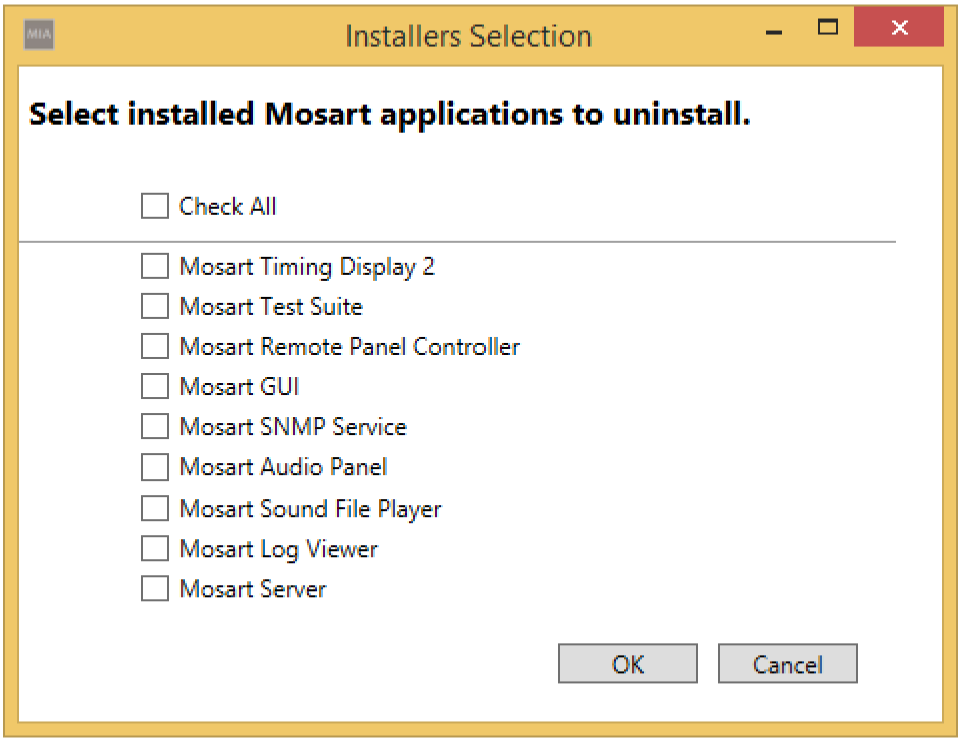
3.5.12 Uninstall Mosart



When pressing this button, the .Viz Mosart Installation Administrator v3.8 will find and display all installed Viz Mosart Applications.

The user may select one or all of the applications to uninstall.

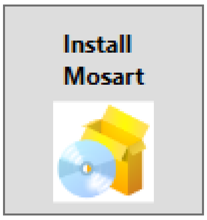
The MIA does not delete desktop shortcuts. Existing shortcuts can then be reused on completion of an upgrade.



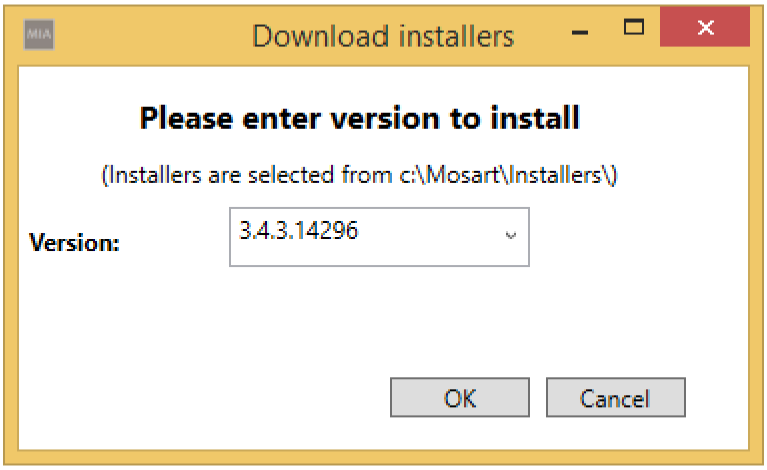
Select installed Mosart applications to uninstall.

- □ Check All
- □ Mosart Timing Display 2
- □ Mosart Test Suite
- □ Mosart Remote Panel Controller
- □ Mosart GUI
- □ Mosart SNMP Service
- □ Mosart Audio Panel
- □ Mosart Sound File Player
- □ Mosart Log Viewer
- □ Mosart Server

3.5.13 Install Mosart



When pressing this button, the .Viz Mosart Installation Administrator v3.8 will display all available Viz Mosart installers found in the Viz Mosart installation path, `C:\Mosart\Installers`.



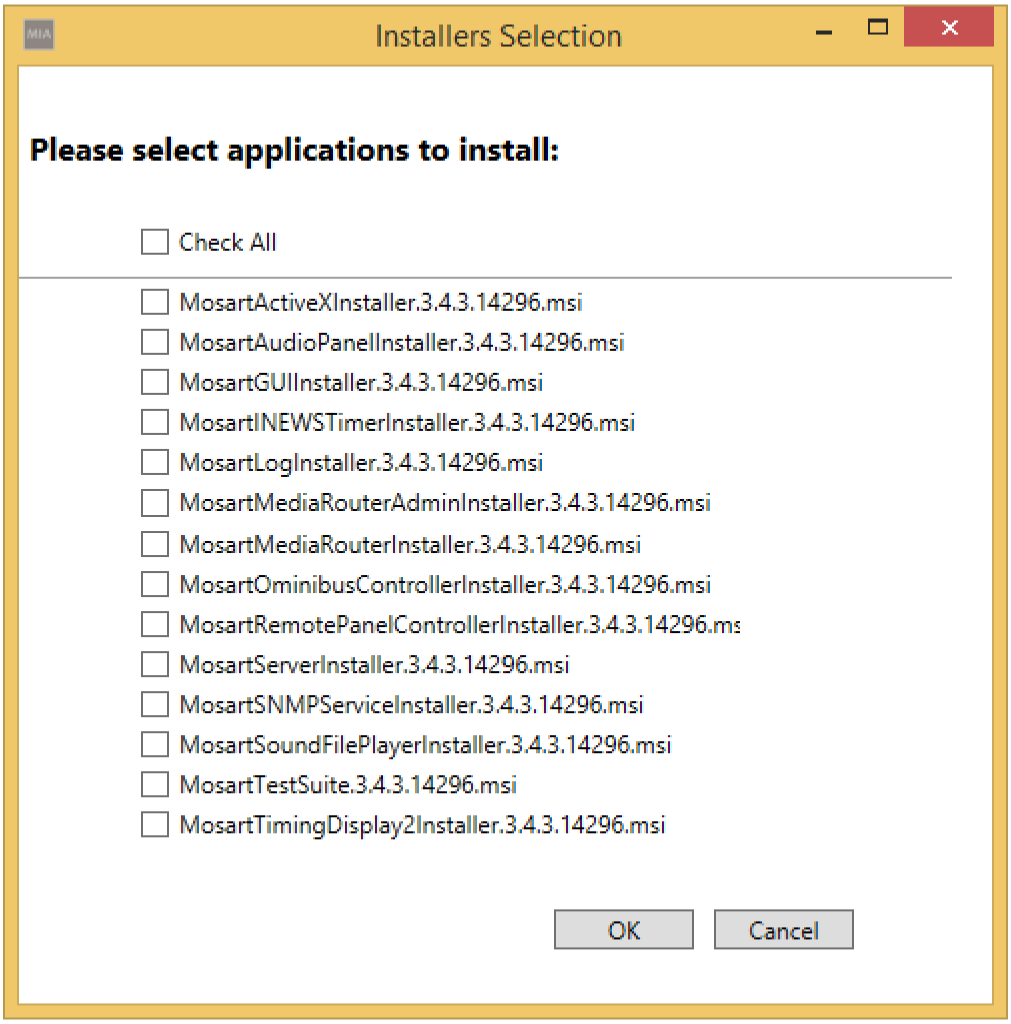
Please enter version to install

(Installers are selected from c:\Mosart\Installers\)

Version: 3.4.3.14296

Select the Viz Mosart version you want to install, and click **OK**. The next screen will show applications found in the installation folder. The user may select one or all of the installers to run.

Tip: For details on how to locate/change your installation folder path, see Settings.



Installers Selection

Please select applications to install:

☐ Check All

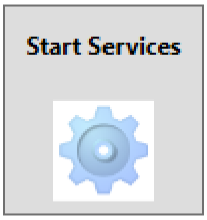
☐ MosartActiveXInstaller.3.4.3.14296.msi
☐ MosartAudioPanelInstaller.3.4.3.14296.msi
☐ MosartGUIInstaller.3.4.3.14296.msi
☐ MosartINEWSTimerInstaller.3.4.3.14296.msi
☐ MosartLogInstaller.3.4.3.14296.msi
☐ MosartMediaRouterAdminInstaller.3.4.3.14296.msi
☐ MosartMediaRouterInstaller.3.4.3.14296.msi
☐ MosartOminibusControllerInstaller.3.4.3.14296.msi
☐ MosartRemotePanelControllerInstaller.3.4.3.14296.ms
☐ MosartServerInstaller.3.4.3.14296.msi
☐ MosartSNMPServiceInstaller.3.4.3.14296.msi
☐ MosartSoundFilePlayerInstaller.3.4.3.14296.msi
☐ MosartTestSuite.3.4.3.14296.msi
☐ MosartTimingDisplay2Installer.3.4.3.14296.msi

The .Viz Mosart Installation Administrator v3.8 will then perform a silent installation, where user interaction is unnecessary, of the selected applications.

The installation assistant does not create desktop shortcuts, this is so that existing shortcuts can be reused on completion of an 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.

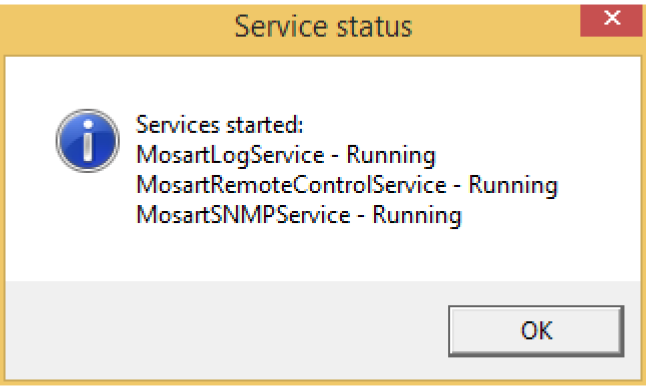
3.5.14 Start Services



This button will attempt to start all the Viz Mosart services installed, with a default timeout of 6 seconds.

Tip: The timeout length can be adjusted in the installation assistant’s Settings.

A dialog box will appear with current service status.



Services started:
MosartLogService - Running
MosartRemoteControlService - Running
MosartSNMPService - Running

Repeat if necessary, alternatively perform a manual start within Windows Services.

3.6 Upgrading Viz Mosart

Viz Mosart can be upgraded using the Viz Mosart Installation Administrator in one of two ways:

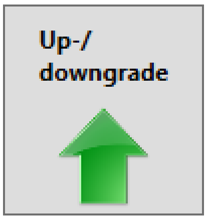
- Semi-automatic mode (see Semi-Automated Installation), or
- Fully automatic mode (see Upgrading)

3.6.1 Prerequisites

To upgrade Viz Mosart, the following prerequisites must be ensured:

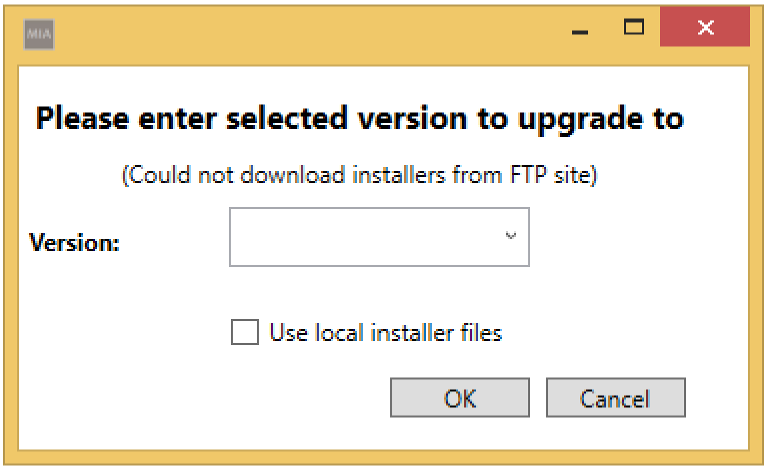
1. Make sure that the following directory structure is present on the server or workstation to upgrade:
   - `C:\Mosart`
   - `C:\Mosart\Installers`
   - `C:\Mosart\Backup`
2. Copy the new installer set, using the version name, to the following directory:
   - `C:\Mosart\Installers`
3. Perform a backup of the current configuration using the Viz Mosart Installation Administrator.

3.6.2 Upgrading



An automatic upgrade with the Viz Mosart Installation Administrator allows the user to seamlessly upgrade an existing Viz Mosart installation. The existing installation must have been installed with the installation assistant in order for the fully automatic upgrade to work.

The automatic upgrade will display all newer versions found in `C:\Mosart\Installers`.



Please enter selected version to upgrade to

(Could not download installers from FTP site)

Version:

□ Use local installer files

Select the version you want to install, and click **OK**.

The following actions will take place:

- Some critical user-data will be backed up - see Backup Files
- Services will be stopped - see Stop Services
- Uninstall all found Viz Mosart Applications
- Mosart Applications will be installed (replacing uninstalled applications) - see Install Mosart
- Services will be started - see Start Services

Note: For more information on installing Viz Mosart on a clean system, see Semi-Automated Installation.

3.6.3 Rollback

If you encounter an issue and require a software rollback, please perform the following steps:

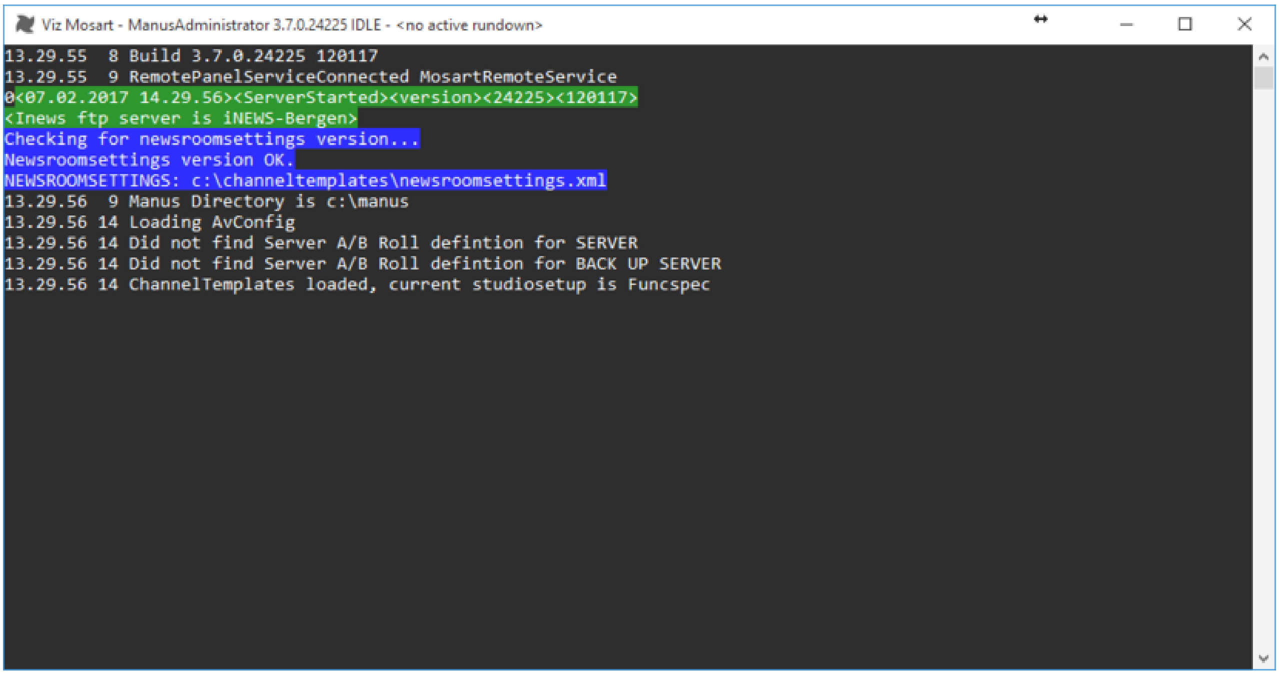
1. Uninstall the current version of Viz Mosart using the Viz Mosart Installation Administrator.
2. Install the previous version of Viz Mosart using the Viz Mosart Installation Administrator.

Note: If any configuration or template changes are made, a reversal of the *C:\ChannelTemplate* directory will cover all necessities.

Keep it safe

Remember to always have a **full and verified backup** of all critical applications and datafiles before changing, upgrading or installing.

4 Manus Administrator Configuration


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 defintion for SERVER
13.29.56 14 Did not find Server A/B Roll defintion for BACK UP SERVER
13.29.56 14 ChannelTemplates loaded, current studiosetup is Funcspec
```

Manus Administrator controls the current rundown in the Viz Mosart GUI and receives rundowns from the attached Newsroom System. It runs as a console application and should be kept running at all times on your Mosart server.

Note: Manus Administrator must be restarted for any changes made to the configuration settings to be applied.

There are two versions of the Manus Administrator, which one you use depends on your 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.

4.1 Configuration Editors

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

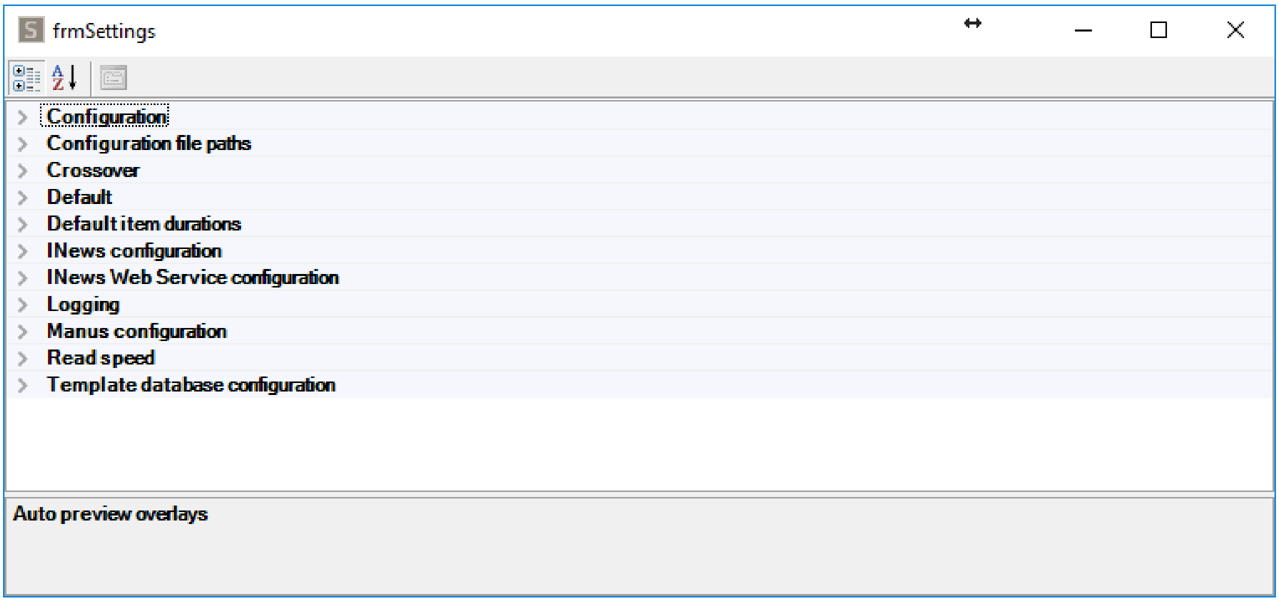
- To open the **frmSettings XML Editor**, type `settings` in the console followed by <*ENTER*>. For details see:
  - Settings Editor - iNews (for iNews FTP Newsroom connections)

  - Settings Editor - MOS (for MOS Newsroom connections)
- 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 will open in a new window. If you make changes to the configuration, you will need to restart Manus Administrator for the changes to be applied.

4.2 Settings Editor - iNews

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



frmSettings

- Configuration
- Configuration file paths
- Crossover
- Default
- Default item durations
- INews configuration
- INews Web Service configuration
- Logging
- Manus configuration
- Read speed
- Template database configuration

Auto preview overlays

Open sub-sections by clicking the section headers, for example Configuration.

- **Templates allowing graphic pretake**: List of template types that will allow pretake of overlay graphics elements. Default: PACKAGE, VOICEOVER
- **Transitions on Accessories**: If enabled, allow transition effects on accessory templates. If disabled, any transition effects will be removed from accessory templates when translating newsroom xml into Mosart xml. This is how it has been, historically.
- **Auto preview overlays**: List of graphics destinations supporting preview of overlay graphics. I.e. DSK will enable preview of all overlay graphics with handler=DSK. Default: Empty = no preview
- **Auto Take Offset**: Value of an offset for the autotake function. Default: 0
- **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
- **Force clip editorial time**: Always show Editorial time in GUI. Default: False

- **Crossover Auto Take On Switch Offset**: Offset in milliseconds to pre-take an autotake when the next story item is after a crossover switch. Positive value will trigger the autotake before the given duration of the story item. Negative value will trigger the autotake after the given duration of the story item.
- **ConnectionString**: The connection string of the crossover, example: controller=IP address, client=crossover. Default: “”
- **Crossover Set Next On Switch Delay**: Delay of sending setnext story from the server running the show when taking control. Default: 500 (milliseconds)
- **Break**: The minimum length given to the BREAK template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 100 (milliseconds)
- **Camera**: The minimum length given to the CAMERA template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 125 (milliseconds)
- **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 will be given to the element. Default: 125 (milliseconds)
- **DVE**: The minimum length given to the DVE template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 125 (milliseconds)
- **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 will be given to the element. Default: 125 (milliseconds)
- **Default Handler Name**: For all graphics that have no handler name, a default one will be assigned, and this can be configured using DefaultHandlerName setting in Manus Administrator. Default: DSK
- **Item**: The minimum length given to any template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 100 (milliseconds)
- **Live**: The minimum length given to the LIVE template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 125 (milliseconds)
- **Lower-third**: The minimum length given to the Lowerthird template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 100 (milliseconds)
- **Default Manus**: This value selects the default rundown to use. If the setting UseDefaultManus is set to True, the system will load this rundown on startup.
- **Telephoneinterview**: The minimum length given to the PHONO template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 125 (milliseconds)
- **Frame rate**: Frame rate of the system. Valid rates 25, 29.97, 50 and 60.
- **FTP Port**: iNews FTP port
- **IdleOnStartUp**: *Deprecated (Startup in idle mode)*
- **IgnoreSendCueStatusToNCSFForOfflineClips**: Enable this flag to prevent the NCS for receiving CUED or READY statuses of offline video clips. Default: False
- **IgnoreUpdatesIfNoChanges**: If set, will ignore story update from NCS if the updated story is assumed to be equal to the current version of the story. Default: False (no NCS story updates are ignored)
- **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
- **Server**: Hostname or IP address where the iNews ftp server is running.

- **Password**: Password of a valid iNews FTP account. Normally password to a valid iNews user with access rights to the iNews FTP server.
- **User Name**: User name of a valid iNews FTP account. Normally name of a valid iNews user with access rights to the iNews FTP server.
- **Web Service Connection**: Connection string. Example: WebServiceServer=localhost; iNewsServer=10.211.112.104; iNewsUsername=mosart; iNewsPassword=mosart; SendUpdatesStatusForAllItems=true; ClearStatusWhenRundownReloaded=true
- **Working Directory**: Initial directory in iNews. Viz Mosart will give access to all rundowns stored hierarchically within this directory. Default: SHOW
- **Time to delay initial devices while waiting for NCS to add story**: Delay of displaying the loading rundown when rundowns are being published to Viz Mosart. Default:500 (milliseconds)
- **Keep story status on updates from NCS**: Enable this to keep played out stories gray if the story is updated in the NCS. Default: False
- **Log level**: Sets the detail level of logging to the log file: 0=normal, 1=warnings, 2=errors, 3=info, 4=detailed
- **Manus Directory**: The path to the folder containing copies of the internal Viz Mosart rundown. Default: c:\manus
- **Manus expiration time**: Time in days to keep Manus Administrator files. I.e. files older than ManusExpirationTime will be deleted when 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. I.e. "TEST*;DEMO??.xml" will keep 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
- **MetaData creator priority**: Select between Newsroom editorial time, MAM duration or full video clip length. Separate IDs with commas. Override the priority of where the timing information of a video file is taken from. 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 will insert NCS as the first and GLOBAL as last entries. Default: Empty
- **Minimum clip length**: This value sets the minimum visible length of an offline clip in a story in the GUI, as soon as Viz Mosart receives clip information the visible length will be updated. Default: 5 (ss:ff)
- **Path for MMLog**: The path where the Viz Mosart log is stored. Default: MosartLog
- **MSMQ Log limit**: Value to identify when the application should dump the log queue to file. Default: 4023
- **MyPort**: The port used for other Viz Mosart applications to connect to the Manus Administrator. Default: 8085
- **Network exclude**: Manus Administrator will automatically enter 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 will automatically enter 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 will follow primary events mix delay. For example, a lower third element will be delayed the same mixdelay as the package it is supposed to be keyed on. Default: False

- **Pause Timing On First Break**: When enabled, any first story with a single break template will not start the rundown timing (Elapsed rundown duration). The timing will start when the next story is taken
- **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.
- **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 prompter text to determine duration of announcement and thus the length of the green camera bar and the green part of the voiceover bar in the Viz Mosart GUI. Default: 145 (words per minute)

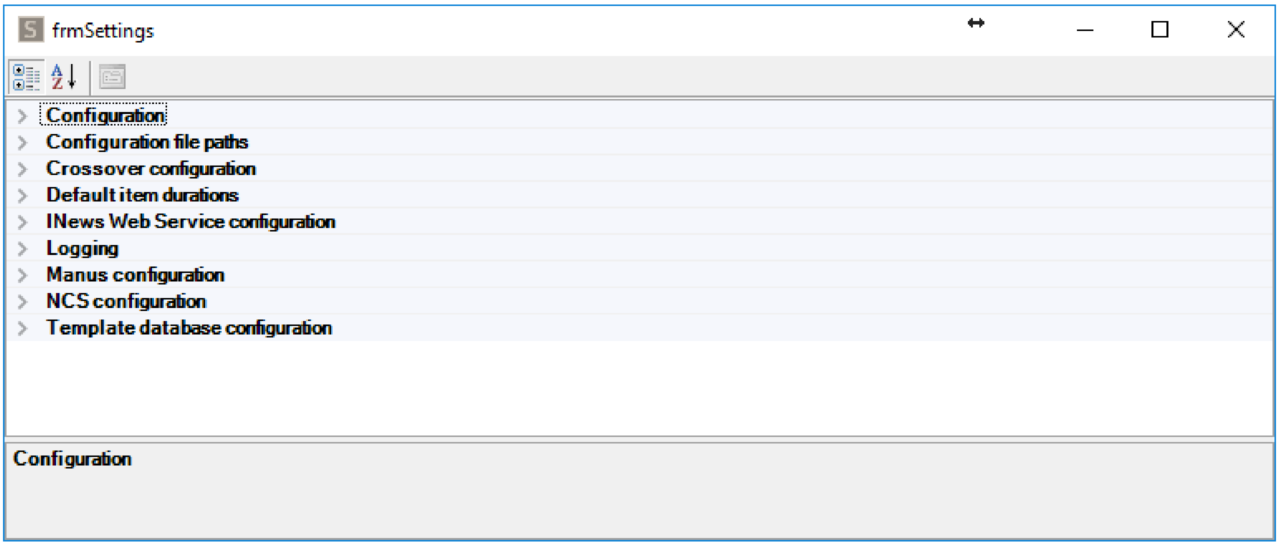
> **Note:** Note: This is a fallback readrate which is only used when the NCS System fails to provide a ReadRate of its own. This function can be tested by changing any readspeed=”xx” to readspeed=”” on an item in a local Manus file.

- **Use character for read speed word**: If True, each *character* will count as a word for the purpose of calculating the spoken duration of prompter text, cf. the setting 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 (the default), words are separated by space characters.
- **Refresh Media On NCS Update**: Setting to true will trigger clip refresh for all clips that belong to a story being updated, normally via story updates from NCS. This ensures that all clip information is synchronized. Default: False
- **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**: User defined descriptive name of server that is displayed in the Timing Display
- **Should Upgrade**: For internal use only. Should not be manually changed. Default: False
- **Story Compare Ignore Attributes**: Used for debugging purposes only. Default: Empty
- **ConnectionString**: The connection string for the Template DB. Forexample, 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.
- **TestManus**: Rundown to be used for maintenance purposes.
- **Trace internally**: Enables or disables internal tracing to console, for debugging only. Default: False
- **Use the default selected manus**: When enabled, will automatically initialize the rundown given in the DefaultManus setting. Default: True
- **UseItemStatusToNCS**: Enable this to send online offline CUE/PLAY/STOP status to the NCS. Default:False
- **In use**: Enables or disables the logging of events from the Manus Administrator to the logfile. Default: True
- **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 will be executed. Default:False. When True, enables the takeout functionality for overlay graphics.

- **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
- **Use verbose logging**: Enables or disables verbose logging. If UseLogging is set, verbose increases the details sent to the log. Default: False
- **Ignore verbose events filter**: Semicolon based list of events to ignore in the log when using verbose logging. Default: Empty (log all events)
- **Pass verbose events filter**: Semicolon based list of events to log when using verbose logging. Default: Empty (log all events)
- **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 Settings Editor - MOS

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 Web Service configuration
- Logging
- Manus configuration
- NCS configuration
- Template database configuration

Configuration

- **Templates allowing graphic pretake**: List of template types that will allow pretake of overlay graphics elements. Default: PACKAGE, VOICEOVER
- **Auto preview overlays**: List of graphics destinations supporting preview of overlay graphics. I.e. DSK will enable preview of all overlay graphics with handler=DSK. Default: Empty (no preview)
- **Auto take offset**: Value of an offset for the autotake function. Default: 0 (milliseconds)
- **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
- **Force clip editorial time**: Always show Editorial time in GUI. Default: False
- **ConnectionString** [CrossoverClientConnectionString]: The connection string of the crossover, example: controller=IP address, client=crossover. Default: empty
- **Crossover Set Next On Switch Delay**: Delay of sending setnext story from the server running the show when taking control. Default: 500 (milliseconds)

- **Break**: The minimum length given to the BREAK template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 100 (milliseconds)
- **Camera**: The minimum length given to the CAMERA template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 125 (milliseconds)
- **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 will be given to the element. Default: 125 (milliseconds)
- **DVE**: The minimum length given to the DVE template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 125 (milliseconds)
- **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 will be given to the element. Default: 125 (milliseconds)
- **Default Handler Name**: For all graphics that have no handler name, a default one will be assigned, and this can be configured using DefaultHandlerName setting in Manus Administrator. Default: DSK
- **Item**: The minimum length given to any template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 100 (milliseconds)
- **Live**: The minimum length given to the LIVE template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 125 (milliseconds)
- **Lower-third**: The minimum length given to the Lowerthird template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 100 (milliseconds)
- **Default Manus**: This value selects the default rundown to use. The special READYTOAIR value selects and uses all rundowns indicated by the newsroom system as ready to air. Default: READYTOAIR
- **Telephoneinterview**: The minimum length given to the PHONO template type in the rundown. If no time is given in the NCS, then this time will be given to the element. Default: 125 (milliseconds)
- **ReadyToAir by default**: If the value is true, all MOS active rundowns will be assumed to be ready to air. Default: False
- **Enabled**: Enables sending templates to NCS through MOS communication. Default: True
- **Forced Mos protocol version**: Protocol version to use in NCS communication. Default (blank) setting will detect the version from the NCS, otherwise use "2.8.3", and so on. Default: blank
- **Frame rate**: Frame rate of the system. Valid rates: 25, 29.97, 50 and 60.
- **GrupedByType**: 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. Default: True
- **IdleOnStartUp**: *Deprecated - Startup in idle mode*
- **Ignore initial syncronization**: Will ignore all roReq on startup when synchronizing with NCS. Default: False
- **Ignore Send Cue Status To NCS For Offline Clips**: Enable this flag to prevent the NCS for receiving CUED or READY statuses of offline video clips. Default: False
- **Ignore sending item status filter**: List of 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 set, will ignore story update from NCS if the updated story is assumed to be equal to the current version of the story. Default: False (no NCS story updates are ignored)

- **Web Service Connection**: Example: WebServiceServer=localhost;iNewsServer=10.211.112.104; iNewsUsername=mosart;iNewsPassword=mosart;SendUpdatesStatusForAllItems=true; ClearStatusWhenRundownReloaded=true
- **Item status unavailable values**: List of 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**: Enable this to keep played out stories gray if the story is updated in the NCS. Default: False
- **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
- **Manus directory**: The path to the folder containing copies of the internal Viz Mosart rundown. Default: c:\manus
- **Manus expiration time**: Time in days to keep Manus Administrator files. Files older than the ManusExpirationTime will be deleted when 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. Default: Empty (no files to keep) i.e. "TEST*;DEMO??.xml" will keep all files starting with “TEST” and all DEMO??.xml files where ‘?’ denotes a wild character. Note ‘;’ (semicolon) is used to separate the file search patterns.
- **MergedClips**: Select Merged to send all PACKAGE and VOICEOVER templates as collection called CLIPS to the NCS using the newt or updt commands in the Manus Administrator console. Default: True
- **MetaData creator priority**: Select between Newsroom editorial time, MAM duration or full video clip length. Separate IDs with commas. Override the priority of where the timing information of a video file is taken from. 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 will insert NCS as the first and GLOBAL as last entries. Default: Empty
- **Minimum clip length**: This value sets the minimum visible length of an offline clip in a story in the GUI. As soon as Viz Mosart receives clip information, the visible length will be updated. Default: 5 (ss:ff)
- **Minimum Time Between Story Update**: If no update is received from the NCS within the given minimum time, Viz Mosart will force a story update. Default: 1000 (milliseconds)
- **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.
- **Path for MMLog**: The path where the Viz Mosart log is stored. Default: MosartLog
- **mosID**: MOS identification of this instance of the Manus Administrator. Generic value is mosart. <galleryID>.<stationID>.mos.
- **Lower Port**: The MOS protocol communicates on three ports; lower, upper and top. MosUpperPort is MosLowerPort + 1 and MosTopPort is MosLowerPort + 2. Default: 10540
- **Reply with connection MosId**: If set, any MOS message sent to the NCS will use the connection MosId insteaDefault:d 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.
- **Server BackUp**: IP address or hostname of the backup newsroom system’s MOS gateway. See Notes below.
- **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 time between received heartbeats exceeds value, connection is displayed lost. Value=0 means no timeout. Default: 0 (seconds)
- **2. NCS to Mosart Upper port**: NCS heartbeat sent to Viz Mosart. If time between received heartbeats exceeds value, connection is displayed lost. Value=0 means no timeout. Default: 0 (seconds)
- **3. NCS to Mosart High port**: NCS heartbeat sent to Viz Mosart. If time between received heartbeats exceeds value, connection is displayed lost. Value=0 means no timeout. Default: 0 (seconds)
- **4. Mosart to NCS**: Viz Mosart to NCS timeout. If time from heartbeat is sent to response is received exceeds value, connection is displayed lost. Value=0 means no timeout. Default: 0 (seconds)
- **MSMQ Log limit**: Value to identify when the application should dump the log queue to file. Default: 4023
- **NCSId**: Same form as mosID, but this value is the ID for the newsroom system. See Notes below.
- **NCSId Backup**: Same form as mosID, but this value is the ID for the backup newsroom system. See Notes below.
- **NCS Time Zone**: Used when time zone is not indicated by the NCS Server. Other values can be +/-hh[:mm], where hh and mm are two digits hours and optional minutes. Default: Z (UTC)
- **NCS Type**: Generic: No special handling of native NCS commands. Generic, DaletPlus, ENPS, NcPower, Octopus, Open Media and MOSInews. Default: Generic
- **Network Exclude**: Manus Administrator will automatically enter 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 will automatically enter 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 will follow primary events mix delay. For example, a lower-third element will be delayed the same mixdelay as the package it is supposed to be keyed on. Default: False
- **Pause Timing On First Break**: When enabled any first story with a single break template will not start the rundown timing (Elapsed rundown duration). The timing will start when the next story is taken
- **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 prompter text to determine duration of announcement and thus the length of the green camera bar and the green part of the voiceover bar in the Viz Mosart GUI. Default: 145 (words per minute)

Note: Note: This is a fallback readrate 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* will count as a word for the purpose of calculating the spoken duration of prompter text, cf. the setting 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 (the default), words are separated by space characters.
- **Refresh Media On NCS Update**: Setting to true will trigger clip refresh for all clips that belong to a story being updated, normally via story updates from NCS. This ensures that all clip information is synchronized. Default: False
- **Reset AutoTake on 'Clear Loop'**: Enable this to automatically disable the autotake mode when using the clear loop function from the GUI. Default: True
- **SendAllTemplateSets**: Enable to send templates from all template sets to the NCS. Disable to send only the default template set. Default: False
- **Server Description**: The description of the server is used for display only. Will be displayed in Timing Display.
- **Should Upgrade**: Internal use only. Should not be manually changed. Default: False
- **Story Compare Ignore Attributes**: Used for debugging purposes only. Default: Empty
- **Story Scope**: Specifies the story scope. Default: Empty (Single). Either:
  - Single = A single story uses one row in the Viz Mosart GUI
  - Grouped = A single story may span over a sequence of rows in the Viz Mosart GUI
    Currently only supported for ENPS where in “Grouped” mode ENPS stories with same story name but different story segment is treated as the same story in Viz Mosart.
- **ConnectionString**: The connection string for the Template DB. E.g. for MySQL: server=<hostname>;User Id=<user>; Password=<password>;database=mosarttemplatedb.
- **Default inserter**: The name to be used for the …_insertedby and …_updatedby columns.
- **Provider name**: The provider name for the Template DB. E.g. MySql.Data.MySqlClient for MySql.
- **Template feedback to NCS**: Default: Enabled, Grouped, Merged
- **Trace**: Enables or disables internal tracing to console for debugging only. Default: False
- **Use NCS backup server**: Enables the NCS backup configuration. Default: False
- **Use Default Manus**: When enabled will automatically initialize the rundown given in the DefaultManus setting. Default: True
- **UseItemStatusToNCS**: When enabled, Viz Mosart will send roItemStat/roElementStat messages to the NCS when clip updates are received from the Media Administrator. If the clip is available on the play-out server, READY is sent. Otherwise NOT READY is sent. Default: False
- **In use**: Enables or disables the logging of events from the Manus Administrator to the log file. Default: True
- **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 will be executed. Default:False.
  When True, enables the takeout functionality for overlay graphics.
- **Use template database**: When True enabes 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

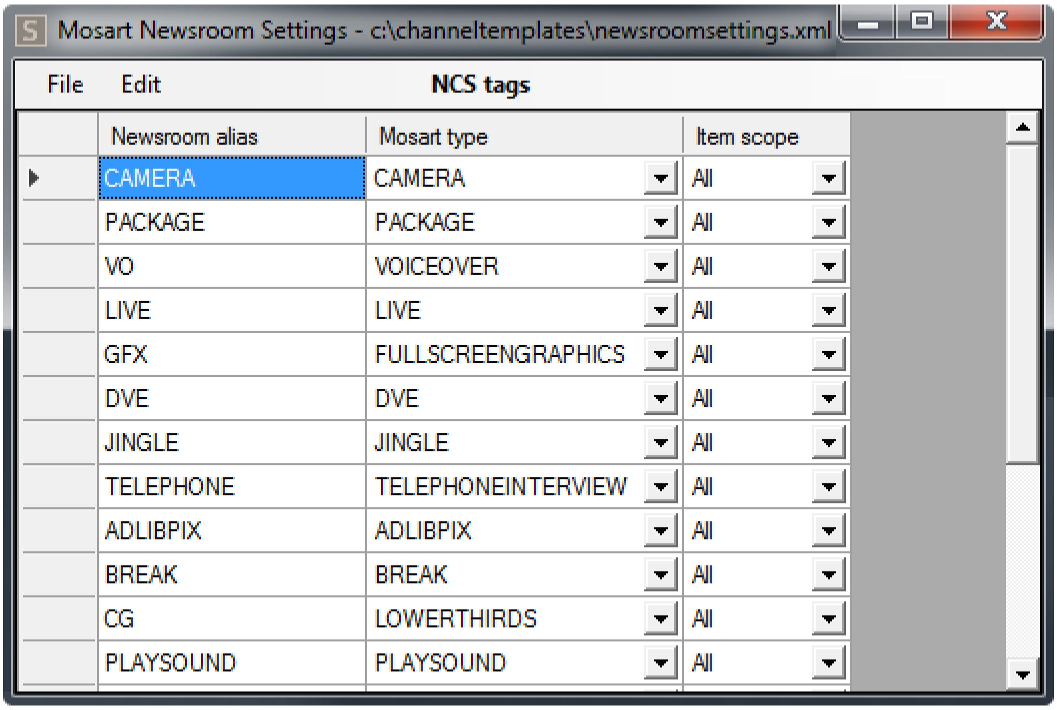
- **Use verbose logging**: Enables or disables verbose logging. If UseLogging is set, verbose increases the details sent to the log. Default: False

- **Ignore verbose events filter**: Semicolon based list of events to ignore in the log when using verbose logging. Default: Empty (log all events)
- **Pass verbose events filter**: Semicolon based list of events to log when using verbose logging. Default: Empty (log all events)
- **wtcLevel**: Sets the detail level of logging to the console. 0=normal, 1=warnings, 2=errors, 3=info, 4=detailed
- **Newsroomsettings**: The path to the XML file containing the mappings from newsroom system commands to Viz Mosart templates. Default: c:\channeltemplates\newsroomsettings.xml

4.3.1 Notes

- If there is more than one main newsroom system, the settings 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 both as main and backup, and if one system is backup for more than one main, it must be listed as backup for each of them.
- 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.
- Only one space between list values, and all values are case sensitive.

4.4 Newsroom Settings Editor


Mosart Newsroom Settings - c:\channeltemplates\newsroomsettings.xml
File
Edit
NCS tags

Newsroom alias	Mosart type	Item scope
CAMERA	CAMERA	All
PACKAGE	PACKAGE	All
VO	VOICEOVER	All
LIVE	LIVE	All
GFX	FULLSCREENGRAPHICS	All
DVE	DVE	All
JINGLE	JINGLE	All
TELEPHONE	TELEPHONEINTERVIEW	All
ADLIBPIX	ADLIBPIX	All
BREAK	BREAK	All
CG	LOWERTHIRDS	All
PLAYSOUND	PLAYSOUND	All

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

This section contains the following topics:

4.4.1 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.

4.4.2 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 will be ignored in Viz Mosart.
- **NCS breakline stories**: List of story names that automatically will be converted to a Viz Mosart break line story when story is tagged as break.
- **NCS accessory stories**: Mosart perceive 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 should be kept on-air for the duration of the parent Viz Mosart template.

- **Lower third keep while story**: Lower third variants that should be kept on-air for the duration of the parent story.
- **Lower third none auto out**: Lower third variants that should be 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 handler that will automatically take the first element in the next story on PACKAGE or VOICEOVER types.
- **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 will be translated to Viz Mosart commands. A value of (*****) will translate all commands in the form (** COMMAND ***) as the * (asterisk) value is interpreted as one or more asterisks.
- **Story External Metadata**: The newsroom system can send special story values in a container called the mosExternalMetadata (MEM), which can be translated to special Viz Mosart content. For detail, see Story External Metadata below.
- **Graphic destination Letters**: Graphic Destination Letters is used to identify output and behavior for graphics elements. For details, see Graphic Destination Letters below.
  Only applies for Vizrt VCP MOS items and use of 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.
- **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 VizMosart.
- **MOS Id mapping**: Enables Viz Mosart to recognize MOS elements from NCS and trigger specific Viz Mosart actions when encountered. Not supposed to 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**: Various keywords specified during Viz Mosart installation for installation-specific behavior of Viz Mosart. Not supposed to be changed by the user.

4.4.3 Story External Metadata

The newsroom system can send special story values in a container called the mosExternalMetadata (MEM). These values can be translated to special Viz Mosart content. This enables you to set the transition type (MIX,WIPE,EFFECT) of the first item in the story from the NRCS rundown. This also

gives a good overview in the newsroom on which effects are used between the stories. Items IDs are:
mostagname: XML-tag name in the MEM
mos_value: Value in the mostagname from the NCS
Mosart Action:

- *template_type*: use the 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 etc. Or, you can use Mosart type number, example: 0 for Camera, 1 for Package etc. If the value from NCS is empty, no item will be created.
  Example:
  ```
  <mostag mostagname="var-1" mosart_action="template_type"
  format_type="String" />
  ```
  A new Mosart item of type given in the NCS column "var-1" will be created.
- *item_variable*: use the value from NCS to set a field on the first primary item found in the story. Non-empty values will 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 will be 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 will be applied to the first primary item in the story.
  - If *action_value* contains the template type of a 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, then the duration will apply to all items of that type (eg. 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 comma, then the duration will apply to those specific items in the story (eg. 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. The effect transition is given in the format <effect_name><separator><input>, where separator can be whitespace, bar (|), comma(,), semicolon(;) or colon (:). The <effect_name> can be EFFECT, MIX, WIPE or their translations as configured in the Keywords section from Newsroomsettings. For example, if EFFECT is mapped as EFFEKT in Keywords section, we can add as following effects to the 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 will NOT overwrite existing fields.
- *story_duration*: use the value from the 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*: will execute 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*: will execute 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*: will translate 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 *endfrase* 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 Mosart item with the value given 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="" fieldty
  pe="TEXT" keylist="" />
      <field name="clip_hirespath" value="" default="" fieldtype
  ="TEXT" keylist="" />
      <field name="metadata_lookuppath" value="" default="" fiel
  dtype="TEXT" keylist="" />
  </fields>
  ```

  and the following mappings are given in newsroom settings:

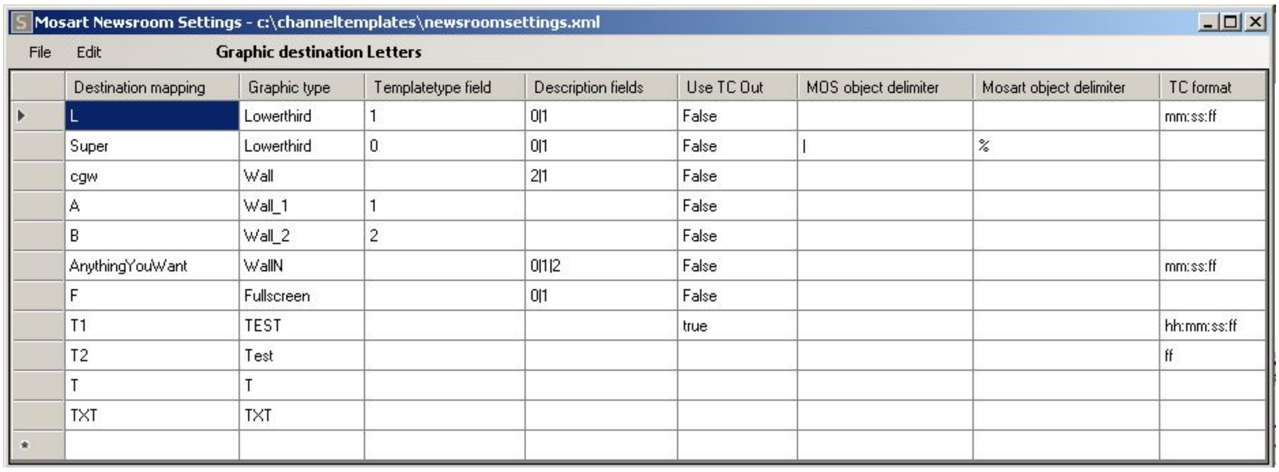
  ```
  <mostag mostagname="ClipDescrCol" mosart_action="field_value_1
  " action_value="clip_description" format_type="String" />
  <mostag mostagname="ClipIdCol" mosart_action="field_value_2"
  action_value="clip_hirespath=MyClip" mos_value="test" format_t
  ype="String" />
  ```

  then the field "clip_description" will be set to the value given in NCS in the column with id "ClipDescrCol" and the field "clip_hirespath" will be set to the value "MyClip" if the value from column "ClipIdCol" in NCS is "test".

Notice that 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 Mosart story. This is used to store specific info going to the AsRunLog

4.4.4 Graphic Destination Letters



Mosart Newsroom Settings - c:\channeltemplates\newsroomsettings.xml
File
Edit
Graphic destination Letters

	Destination mapping	Graphic type	Templatetype field	Description fields	Use TC Out	MOS object delimiter	Mosart object delimiter	TC format
▶	L	Lowerthird	1	0|1	False			mm:ss:ff
	Super	Lowerthird	0	0|1	False	|	%	
	cgw	Wall		2|1	False			
	A	Wall_1	1		False			
	B	Wall_2	2		False			
	AnythingYouWant	WallN		0|1|2	False			mm:ss:ff
	F	Fullscreen		0|1	False			
	T1	TEST			true			hh:mm:ss:ff
	T2	Test						ff
	T	T						
	TXT	TXT						
*								

Graphic Destination Letters 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 has 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 will be 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 will be 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 will be shown in Viz Mosart GUI separated by the character given in *outputtxtdelimiter* setting. The indexes are 0-based and will be 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 graphic starting time, end time or duration. Default is *mm:ss*. Example of formats: *hh:mm, hh:mm:ss, mm:ss:ff, mm:ss, ss:ff, ff*.

  Example mapping:

```
<graphicdestinationletters>
  <graphicdestinationletter destinationmapping="L" type="Lowerthird"
templatetypefield="1" descriptionfields="0|1" use_tc_out="False" inpu
ttxtdelimiter="" outputtxtdelimiter="" mask="mm:ss:ff"/>
  <graphicdestinationletter destinationmapping="Super" type="Lowerthi
rd" templatetypefield="0" descriptionfields="0|1" use_tc_out="False"
inputtxtdelimiter="|" outputtxtdelimiter="%" mask=""/>
  <graphicdestinationletter destinationmapping="cgw" type="Wall" temp
latetypefield="" descriptionfields="2|1" use_tc_out="False" inputtxtd
elimiter="" outputtxtdelimiter=""/>
  <graphicdestinationletter destinationmapping="A" type="Wall_1" temp
latetypefield="1" use_tc_out="False" inputtxtdelimiter="" outputtxtde
limiter=""/>
  <graphicdestinationletter destinationmapping="B" type="Wall_2" temp
latetypefield="2" use_tc_out="False" inputtxtdelimiter="" outputtxtde
limiter=""/>
  <graphicdestinationletter destinationmapping="AnythingYouWant" type
="WallN" templatetypefield="" descriptionfields="0|1|2" use_tc_out="F
alse" inputtxtdelimiter="" outputtxtdelimiter="" mask="mm:ss:ff"/>
  <graphicdestinationletter destinationmapping="F" type="Fullscreen"
templatetypefield="" descriptionfields="0|1" use_tc_out="False" input
txtdelimiter="" outputtxtdelimiter=""/>
  <graphicdestinationletter destinationmapping="T1" type="TEST" use_t
c_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>
```

To support a **configurable number of walls** there are two ways of doing it:

1. Having a general rule in the Graphics Destination Mappings where the graphics type must be "WallN". The "destionationmapping" attribute can be set to any string of 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 Graphics objects have to contain a Mosart string command ending into a number (this will give 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. So, given these examples, the destination for the first graphic will be *WALL_1*, and then *WALL_2*, *WALL_3* and *WALL_4*.

2. The Mosart string destination mapping does not end into 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 graphic type. So for the Mosart string examples given earlier, the configuration can be:

   ```
   <graphicdestinationletter destinationmapping="B" type="Wall_2" t
   emplatetypefield="" descriptionfields="-1" use_tc_out="False" in
   puttxtdelimiter="" outputtxtdelimiter=""  mask="mm:ss"/>
   <graphicdestinationletter destinationmapping="Z" type="Wall_3" t
   emplatetypefield="" descriptionfields="-1" use_tc_out="False" in
   puttxtdelimiter="" 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="W
   all_5"templatetypefield="" descriptionfields="-1" use_tc_out="Fa
   lse" inputtxtdelimiter="" outputtxtdelimiter=""   mask="mm:ss"/>
   ```

   and this means that overlays with `Mosart=B|00:00|00:05` will be redirected to *WALL_2*, those with `Mosart=Z|00:00|00:05` to *WALL_3* and so on.

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

```
<graphicdestinationletter destinationmapping="T" type="TEST" template
typefield="" descriptionfields="-1" use_tc_out="False" inputtxtdelimi
ter="" outputtxtdelimiter="" mask="mm:ss"/>
```

and a Vizrt graphic with Mosart string "`T|00:00|00:06`", it will send 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="TIMECO
DE-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="tr
ue" use_graphics_id="true" graphics_id="121" handler_name="DSK" graph
ics_out_on="TIMECODE" 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" defau
lt="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" defaul
t="00:00" value="00:00"/>
  </fields>
</item>
```

The graphic `slug` is used, for example, to visualize the graphics in the Viz Mosart GUI Assets window, or in the Overlay Graphics interface. The slug is taken from the `MosAbstract` found in the Viz Mos object and the fields delimited by the **inputtxtdelimiter** are replaced with **outputtxtdelimiter**.

There are several types of Viz Mos graphic objects that Viz Mosart can receive 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 as explained above, from MosAbstract).

Otherwise, the slug is set to the description created as explained above from data nodes entries where only the data nodes from the positions specified in the **descriptionfields** will be 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="t
rue" 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 destinationmapp
ing="L" type="Lowerthird" templatetypefield="" descriptionfields="" i
nputtxtdelimiter="" outputtxtdelimiter="|" /> </graphicdestinationlet
ters>
```

Note that **inputtxtdelimiter** is empty, so the default ‘/’ will be used.

The result is the following Mosart item (notice 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="TIMECO
DE-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="tr
ue" use_graphics_id="true" graphics_id="121" handler_name="DSK" graph
ics_out_on="TIMECODE" 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" defau
lt="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" defaul
t="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="%"
/>
```

The result is as follows, where you can see that the delimiters are now ‘%’ instead of ‘|’. Notice 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="" i
ntimeline="true" date_0="" accessory="False" static="false" endfrase=
"" mosid="PILOT" objid="121" ismoselement="true" use_graphics_id="tru
e" graphics_id="121" handler_name="DSK" graphics_out_on="TIMECODE" de
scription="(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.

And similarly, 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 as explained above, from MosAbstract).

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 destinationmapp
ing="S" type="Lowerthird" templatetypefield="" descriptionfields="1|3
" inputtxtdelimiter="|" outputtxtdelimiter="%" /> </graphicdestinatio
nletters>"
```

The result is as follows:

```
<item type="100" slug="Super Tema + info%1:Tema" source="1" index="10
0___2_2" idref="2" templatetype="TIMECODE-DSK" status="1" error="0" i
n="0" dur="100" pin="0" pdur="100" rdur="0" externaleffect="" intimel
ine="true" date_0="" accessory="False" static="false" endfrase="" mos
id="VIZ.NPRO.MOS" objid="1529718" ismoselement="true" use_graphics_id
```

```
="true" graphics_id="1529718" handler_name="DSK" graphics_out_on="TIM
ECODE" 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" defau
lt="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 the requirements for metadata to a Vizrt MOS item 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 lowerthirds to ignore “Replace TakeOut Logic” meaning that when taking and taking out the item, ignore the replace logic if the value is set on the item
- F is used only for lowerthirds to force “Replace TakeOut Logic” for graphics coming from different templates with same layers (when used, it will practically ignore the “O” state for a layer). Not to be confused with destination letter “F” used often 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 shows examples of the Mosart= part of the Vizrt data element description as in the value cg_gordon_brown/prime minister/Mosart=L|00:02|00:05. The data element will be played out as a lower third, automatically in at 2 seconds with 5 seconds duration.

L|00:02|00:05	Lower third, automatically in at 2 seconds with 5 seconds duration
L|00:02|S	Lower third, automatically in at 2 seconds with duration equal to the length of the story (taken out at the switch from one story to the next)
L|00:00|O	Lower third, automatically in at 0 seconds, never taken out (except for when replaced by another item or manually from the operator)
L|M|00:05	Lower third, manual play-out, duration 5 seconds
L|00:00|00:10|X	Lower third, automatically in at 0 seconds with 10 seconds duration and Replace TakeOut Logic is ignored
W|00:00|B	Video wall, automatically in at 0 seconds, out when switching from one story element to another
F	Full screen graphics (note that this letter is usually used for full screen graphics, but any combination of alphanumeric characters can be used to identify a fullscreengraphic)

If no fields are given, the element will default as a full screen graphics element.

VCP template requirements

The Mosart field is supported embedded in either of two ways: 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

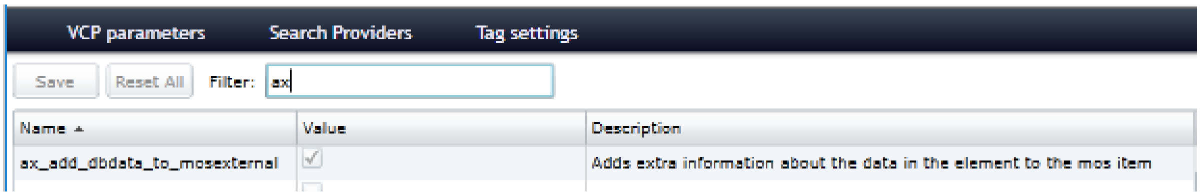
The following VB script can be used in a Vizrt Template Wizard template and will return a properly formatted Mosart field string:

```
'' 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 Pilot database.

If you have a Pilot version from 5.7 this setting is available on the Pilot Data Server settings page found on http://[hostname of PDS]:8177/settings#Params

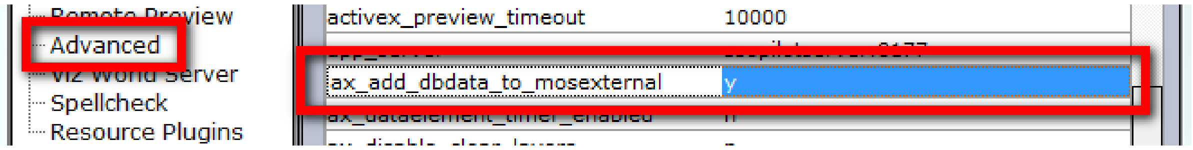


Filter: ax

Name ▲	Value	Description
ax_add_dbdata_to_mosexternal	☑	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 Pilot version older than 5.7 this is easily achieved by opening the Preferences editor in Viz Content Pilot (Options Preferences Advanced), 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

4.4.5 MOS ID Mapping

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

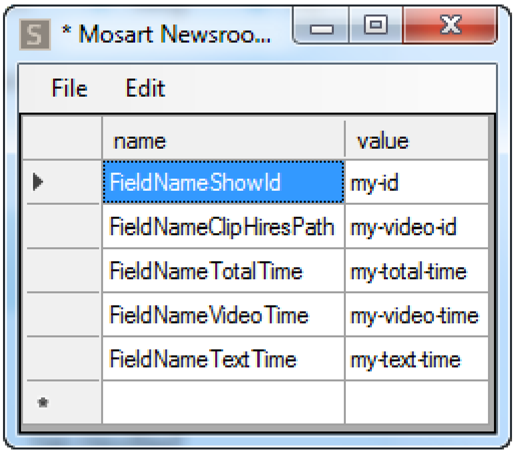
```
<mosids>
    <tag name="" value="" fieldmapping="" appendContent=""
embeddedMosartItem="" keywords="" />
</mosids>
```

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 will exclude video clip information from Vizrt Video MOS objects, and only use 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 will exclude 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 will be ignored. e.g.
    `<tag name="MOSART" value="" appendContent="false" fieldMapping="false" keywords="IgnoreTemplateSet=true"/>`

4.4.6 Overriding Default Field Names When Parsing iNEWS Stories

When parsing the iNEWS NSML the system will extract various field values using default field names. These defaults can be overridden by setting new values in *Newsroom Settings > Edit > Properties*.



name	value
FieldNameShowId	my-id
FieldNameClipHiresPath	my-video-id
FieldNameTotalTime	my-total-time
FieldNameVideoTime	my-video-time
FieldNameTextTime	my-text-time

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 will be 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 will be calculated using this value is added to video time.

4.5 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.

4.5.1 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

4.5.2 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

4.5.3 Target Fields

Target fields is a list of the fields used when controlling a video server or graphics element(s).

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 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 will be forwarded to graphics systems supporting concepts for graphics when cueing and playing an item.
- **channel**: The channel to play the item, ie. DSK, WALL etc.
- **take**: *MANUAL* or *AUTO*. If MANUAL then *tc_in* will be 0 and only *tc_dur* will be used.
- **outBehaviour**:
    - *TIMECODE* = the element will be taken out relative to the current primary element using tc_dur
    - *BACKGROUNDEND* = the element will be taken out when switching primary story element
    - *STORYEND* = the element will be taken out when switching from one story to the next
    - *OPENEND* = the element will be taken out when reloading the rundown or by manually taking the element out
- **auto_continue**: true/false (p.d. only for Orad, to send PlayWithoutPause command)

4.5.4 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**: if true, overwrites if a field already exists. Default is false.

4.5.5 Example - Video Server Item

This example 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>    <abstr
act>Flug über die Alpen fertig2 1:00</abstract>    <objDur>1500<
/objDur>    <objTB>25</objTB>    <objSlug>Flug über die Alpen
fertig2</objSlug> </mos>
```

Note: The first part of the `objID` is the Quantel clip Id for the zone in the second part

Viz Mosart MOS mapping:

```
<mosmap id="QUANTEL" elementtype="CLIP" idasindex="false">
<fieldmapping fieldname="clip_hirespath" mospath="//objID" regex="^\c
+" />     <fieldmapping fieldname="clip_description" mospath="
//objSlug" />     <fieldmapping fieldname="clip_mark_in" fieldtype="s
tatic" value="00:00:00:00" valuetype="TIMECODE" />     <fieldmapping
fieldname="clip_dur" fieldtype="FIELDS" mospath="//objDur" valuetype=
"TEXT" /> </mosmap>
```

Target Viz Mosart XML:

```
<item>   <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> </mosmap>
```

4.5.6 Example - Graphics Item

This example 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> </storyItem>
```

Note: The first part of the `objID` is the Quantel clip Id for the zone in the second part

Viz Mosart MOS mapping:

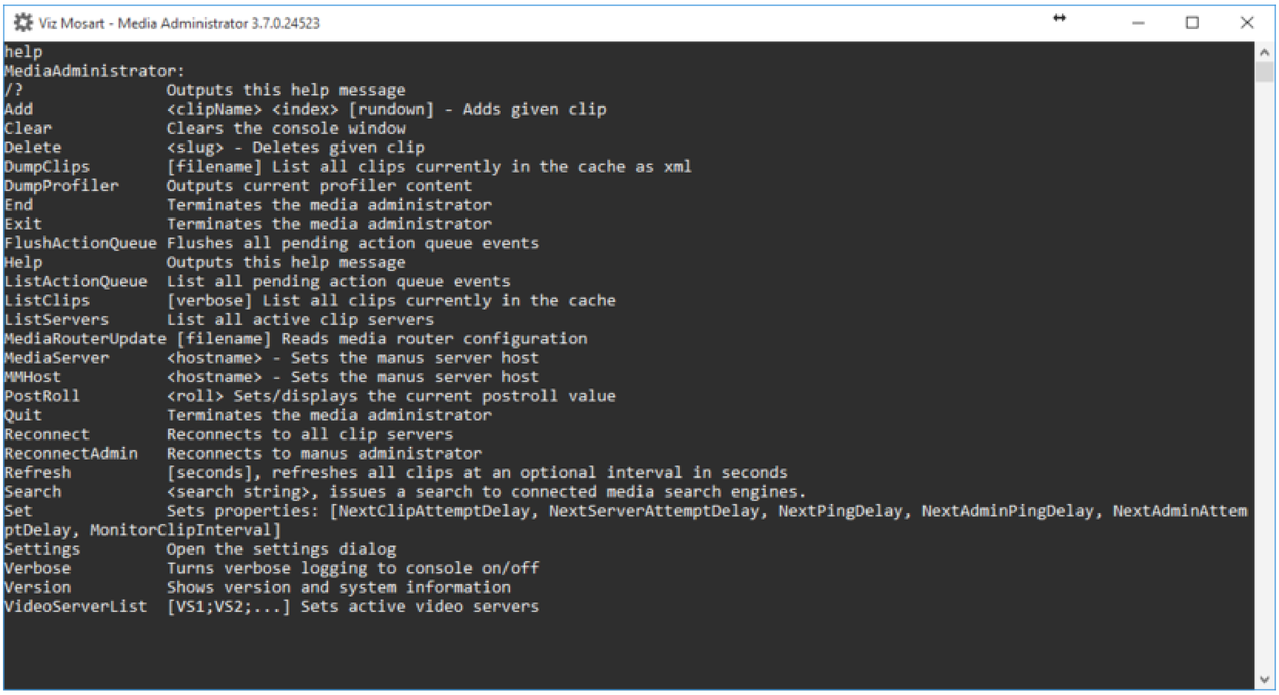
```
<mosmap id="XPRESSION" elementtype="GRAPHICS">     <fieldmapping
fieldname="graphics_id" mospath="(//itemID | //objID)" separator="-"
fieldtype="REPLACE" value="{STORYID}-{MOSPATH}" />     <fieldmapping
fieldname="graphics_description" mospath="//mosAbstract" />
<fieldmapping fieldname="tc_in" fieldtype="FRAMES" mospath="
//itcTimeIn" valuetype="TEXT" />     <fieldmapping fieldname="tc_dur'
 fieldtype="FRAMES" mospath="//itcTimeDur[text()!='0']" valuetype="TE
XT" />     <fieldmapping fieldname="outBehaviour" mospath="
//macroOut" />     <fieldmapping fieldname="channel" fieldtype="STATI
C" mospath="//gfxtype[text()='CG']" value="DSK" valuetype="TEXT"
mustexist="true" />     <fieldmapping fieldname="channel" fieldtype='
STATIC" mospath="//gfxtype[text()='OTS']" value="WALL" valuetype="TEX
T" mustexist="true" />     <fieldmapping fieldname="channel"
fieldtype="STATIC" mospath="//gfxtype[text()='FS']" value="FULL"
valuetype="TEXT" mustexist="true" />     <fieldmapping fieldname="cha
nnel" fieldtype="STATIC" mospath="//gfxtype[text()='OTHER']" value="F
ULL" 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>
```

5 Media Administrator Configuration



```
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
MMHost           <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, NextAdminAttem
ptDelay, 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
```

Media Administrator handles the database connection to your video servers. It is responsible for reporting clip status for all video server elements in the current rundown, and searching the video server for clips within the Viz Mosart GUI. The Media Administrator runs a console application aand should be kept running at all times when it's services are required.

To open the Settings XML editor, type `settings` in Manus Administrator followed by **<ENTER>**.

Note: The application must be *restarted* for any changes made 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, and responds to free text commands. The following table outlines available commands:

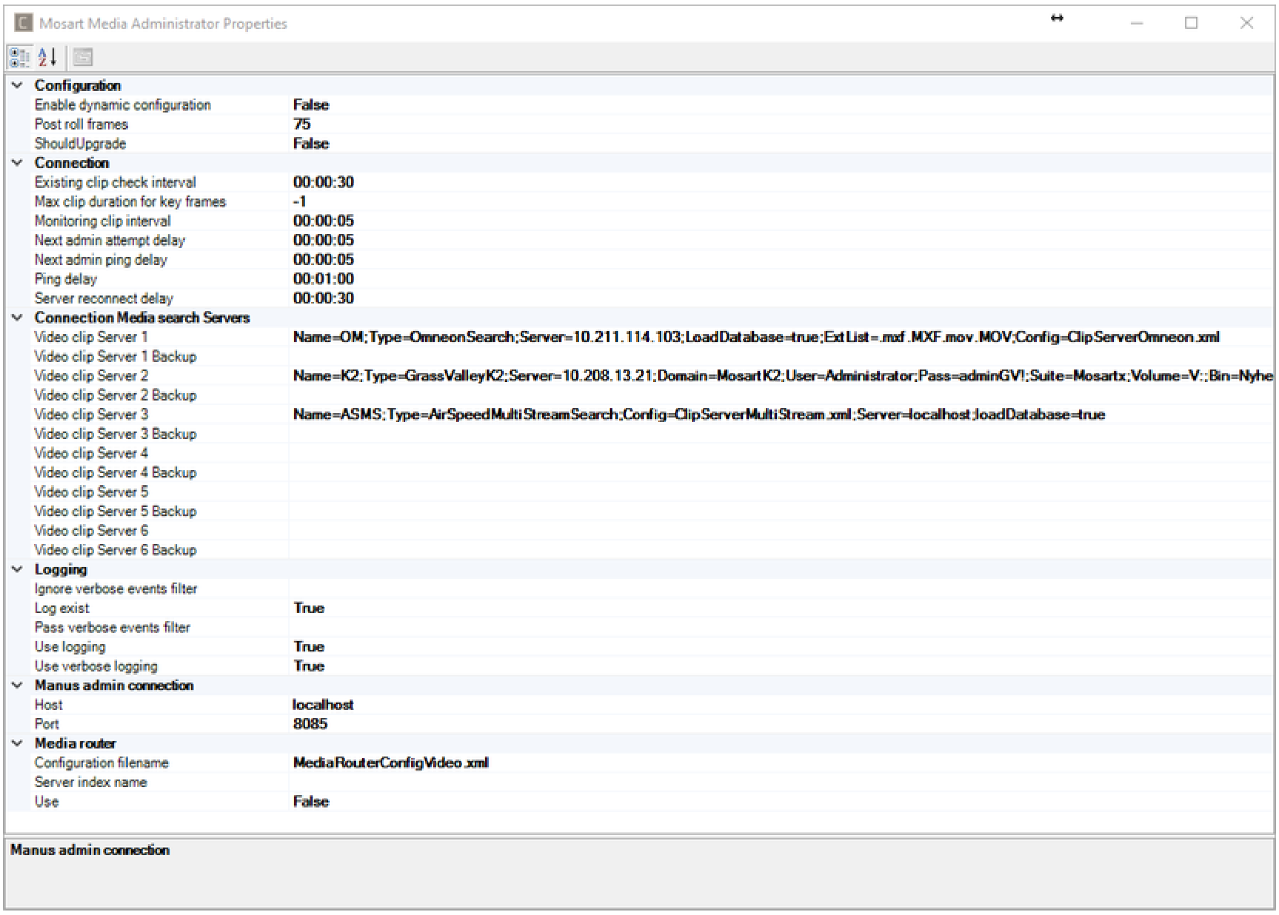
- **add <clipname> <index>**: To manually add a clip to the list of clips currently monitored. This function is typically used while testing the installation.
- **delete <index>**: Deletes a monitored clip with the given index.
- **dumpclips <filename>**: Outputs a list of all monitored clips in an XML format. Outputs to console if a filename is not given.
- **dumpprofiiler**: For testing only. Shows timing information.
- **exit**: Exits the Media Administrator.

- **flushactionqueue**: Flushes the internal action queue. I.e. all pending actions are executed.
- **help**: Outputs a help messages showing 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**: Reconnect 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. I.e. a request is made to the server for all clips. This function is 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. This function is 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. Currently 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. I.e. if connecting 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. Pinging is used 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 changes are made, 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;ExtList=.mxf.MXF.mov.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=adminGV!;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**: Enable this to filter the active server configuration based on the active servers in AV Automation. Default: False
- **Post roll frames**: Value that will be subtracted 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 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**: Use this for disabling 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 Attemt 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 <n>, for details see Video Server and MAM Connection Strings. At most 6 media servers may be connected. In most cases only one server connection is necessary. The connection string may vary according to the type of server.
  Note that 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 that case, the property will be set from the last server checked (i.e the lowest numbered server configured) where the clip was found present.
- **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**: If true, the presence of the Viz Mosart Log Service is verified at startup. Default: True
- **Pass verbose events filter**: Semicolon based list of events to log when using verbose logging. Default: <none>
- **Use logging**: If true, all logging shall be passed to the Log Service. Default: True
- **Use verbose logging**: If true, verbose logging is enabled. Default: False

5.2.5 Manus admin connection

- **Host**: 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**: Corresponding TCP/IP (.NET Remoting) port used to establish connection to the Manus Administrator application. Default: 8085

5.2.6 Media router

- **Configuration filename**: Configuration file for Media Router. Default: MediaRouterConfigVideo.xml
- **Server index name**: *Deprecated*
- **Use**: If true, the Media Router is to be used. Default: False

6 Overlay Graphics Interface

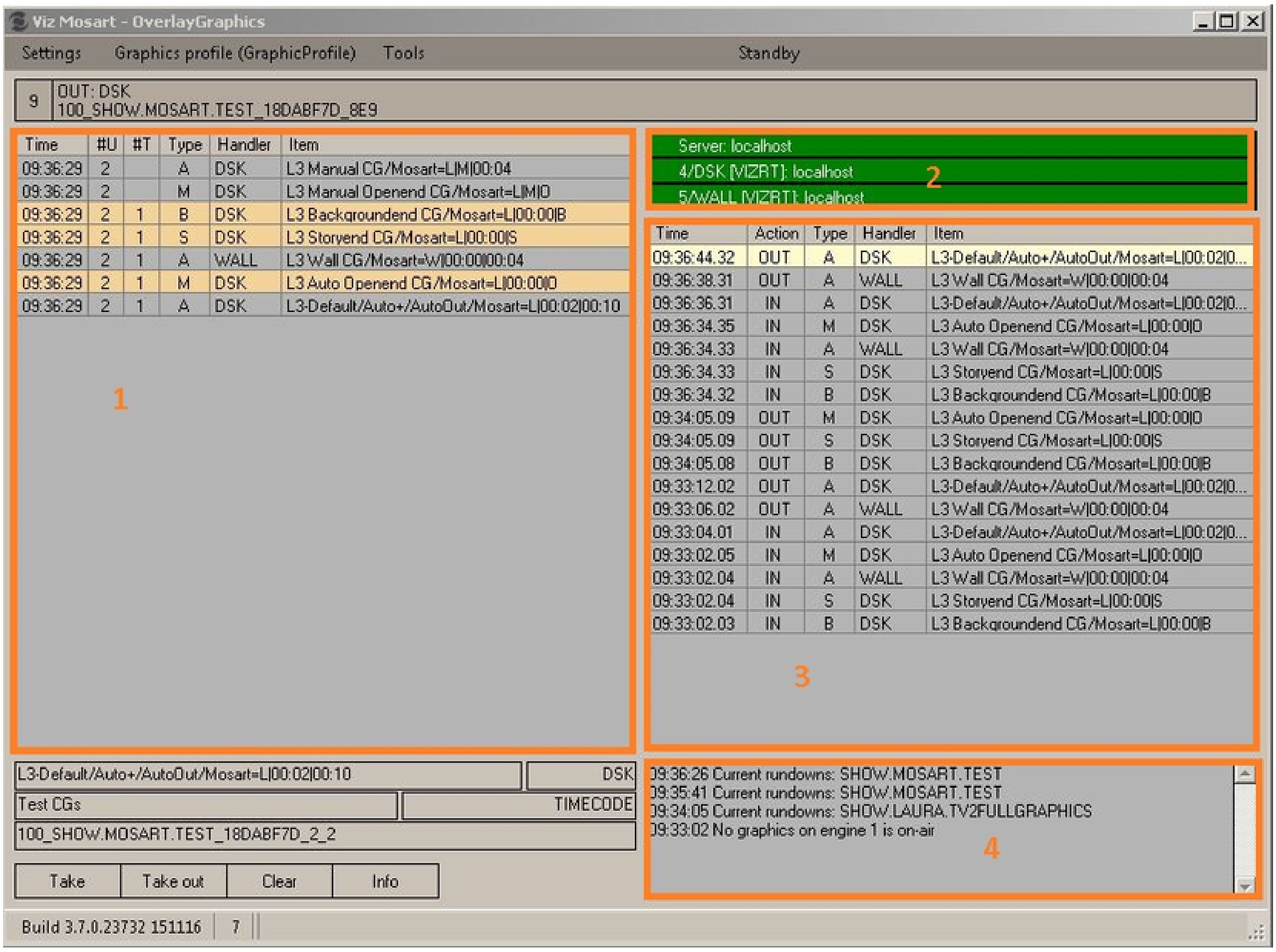
The Overlay Graphics Interface is used for controlling and monitoring overlay graphics for all Viz Mosart approved graphic engines. Configuration is done using the Overlay Graphics Configuration. For a list of supported graphics devices, see Overlay Graphics Types.

Start Overlay Graphics by double-clicking the application icon:

or navigate in Explorer to the installation directory and start the executable *MMOverLayGraphicsInterface.exe*

6.1 Using Overlay Graphics Interface

Overlay Graphics is *only* used for overlay graphics. Full-frame graphics are controlled through the AV Automation application.



9 OUT: DSK
100_SHOW.MOSART.TEST_18DABF7D_8E9

Time	#U	#T	Type	Handler	Item
09:36:29	2		A	DSK	L3 Manual CG/Mosart=L|M|00:04
09:36:29	2		M	DSK	L3 Manual Openend CG/Mosart=L|M|O
09:36:29	2	1	B	DSK	L3 Backgroundend CG/Mosart=L|00:00|B
09:36:29	2	1	S	DSK	L3 Storyend CG/Mosart=L|00:00|S
09:36:29	2	1	A	WALL	L3 Wall CG/Mosart=W|00:00|00:04
09:36:29	2	1	M	DSK	L3 Auto Openend CG/Mosart=L|00:00|O
09:36:29	2	1	A	DSK	L3-Default/Auto+/AutoOut/Mosart=L|00:02|00:10

Server: localhost
4/DSK [VIZRT]: localhost
5/WALL [VIZRT]: localhost

Time	Action	Type	Handler	Item
09:36:44.32	OUT	A	DSK	L3-Default/Auto+/AutoOut/Mosart=L|00:02|0...
09:36:38.31	OUT	A	WALL	L3 Wall CG/Mosart=W|00:00|00:04
09:36:36.31	IN	A	DSK	L3-Default/Auto+/AutoOut/Mosart=L|00:02|0...
09:36:34.35	IN	M	DSK	L3 Auto Openend CG/Mosart=L|00:00|O
09:36:34.33	IN	A	WALL	L3 Wall CG/Mosart=W|00:00|00:04
09:36:34.33	IN	S	DSK	L3 Storyend CG/Mosart=L|00:00|S
09:36:34.32	IN	B	DSK	L3 Backgroundend CG/Mosart=L|00:00|B
09:34:05.09	OUT	M	DSK	L3 Auto Openend CG/Mosart=L|00:00|O
09:34:05.09	OUT	S	DSK	L3 Storyend CG/Mosart=L|00:00|S
09:34:05.08	OUT	B	DSK	L3 Backgroundend CG/Mosart=L|00:00|B
09:33:12.02	OUT	A	DSK	L3-Default/Auto+/AutoOut/Mosart=L|00:02|0...
09:33:06.02	OUT	A	WALL	L3 Wall CG/Mosart=W|00:00|00:04
09:33:04.01	IN	A	DSK	L3-Default/Auto+/AutoOut/Mosart=L|00:02|0...
09:33:02.05	IN	M	DSK	L3 Auto Openend CG/Mosart=L|00:00|O
09:33:02.04	IN	A	WALL	L3 Wall CG/Mosart=W|00:00|00:04
09:33:02.04	IN	S	DSK	L3 Storyend CG/Mosart=L|00:00|S
09:33:02.03	IN	B	DSK	L3 Backgroundend CG/Mosart=L|00:00|B

L3-Default/Auto+/AutoOut/Mosart=L|00:02|00:10 DSK
Test CGs TIMECODE
100_SHOW.MOSART.TEST_18DABF7D_2_2

```
09:36:26 Current rundowns: SHOW.MOSART.TEST
09:35:41 Current rundowns: SHOW.MOSART.TEST
09:34:05 Current rundowns: SHOW.LAURA.TV2FULLGRAPHICS
09:33:02 No graphics on engine 1 is on-air
```

The Overlay Graphics user interface consists of four main areas (referenced as [1] to [4] in the text) as well as a top main menu, graphic status line under the main menu and an info line at the bottom of the main window.

On the top you have the Main Menu giving access to various functions grouped into *Settings*, *Graphic Profile* and *Tools.*

Just under the menu there is the status line which gives information about the graphic that is on-air or off-air.

[1] in figure above: On the left section of the screen you can see a table of overlay graphics (CGs) present in the current rundown. The significance of each column label is:

- **Time**: the time when the graphic is added in Overlay Graphics
- **U**: The number of updates for the selected graphic
- **T**: The number of takes for the selected graphic
- **Type**: The graphic type which can be:
  - **A**= Auto Out (a Manual CG can also be Auto Out)
  - **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 will be suffixed with an "L"

Note:*Locator* is a special graphics item that will be **linked** to a video server *crosspoint*. When taking the crosspoint from a Mosart template, either as a switcher crosspoint or keyed crosspoint (in a DVE box), the graphics will also be (re-)taken.

When an overlay graphic from the list is selected you can see more information about it in graphic details area below the list (from left to right): the graphic title, the handler name, the graphic description, the graphic type and the graphic id.

Under the graphic details area there are buttons which describe actions that can be performed directly from Overlay Graphics for a selected graphic from the list:

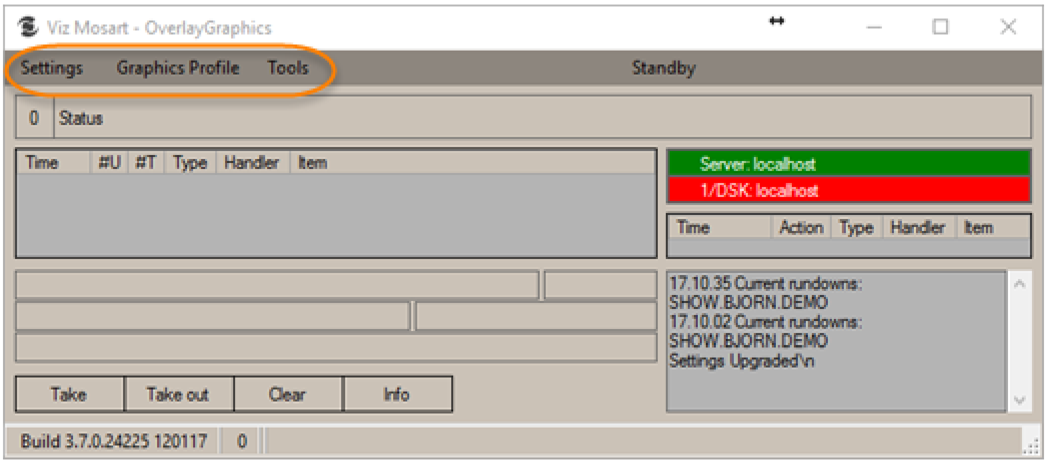
- **Take**: takes graphic on-air
- **Take out**: takes graphic off-air
- **Clear**: takes out all taken graphics
- **Info**: opens an editor window where the graphic MOS item can be changed and updated.
  [2] in figure above: On the top right area you can see the status of the Mosart server and the status of various graphic engines configured.

[3] in figure above: The middle right area is a log of all actions performed on the overlay graphics either using the Multi GUI or the buttons from bottom left side of Overlay Graphics. Double clicking in this area will clear it.

[4] in figure above: On the bottom right you have the event log area which contains messages, warnings and errors from the Overlay Graphics application. Double-clicking in this area will clear it.

The info line at the bottom displays information about the application version, the number of CGs in the table and whether the MMR is used in the configurations.

6.1.1 Main Menu



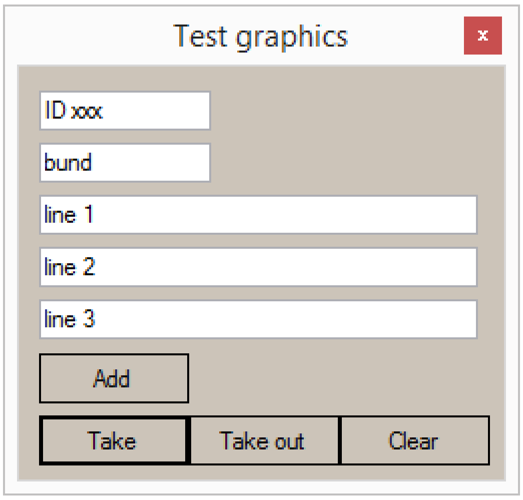
17.10.35 Current rundowns:
SHOW.BJORN.DEMO
17.10.02 Current rundowns:
SHOW.BJORN.DEMO
Settings Upgraded\n

Build 3.7.0.24225 120117 | 0

- **Settings**:

  - **Properties**: Opens Overlay Graphics Configuration
- **Graphics Profile**: Select the graphics profile to use
- **Tools**:
  - **Editor**: Open theTest Graphics Window
- **Standby**: Toggle the graphics engines in or out of standby. The same action can also be performed from MultiGUI's menu.

6.1.2 Test Graphics Window



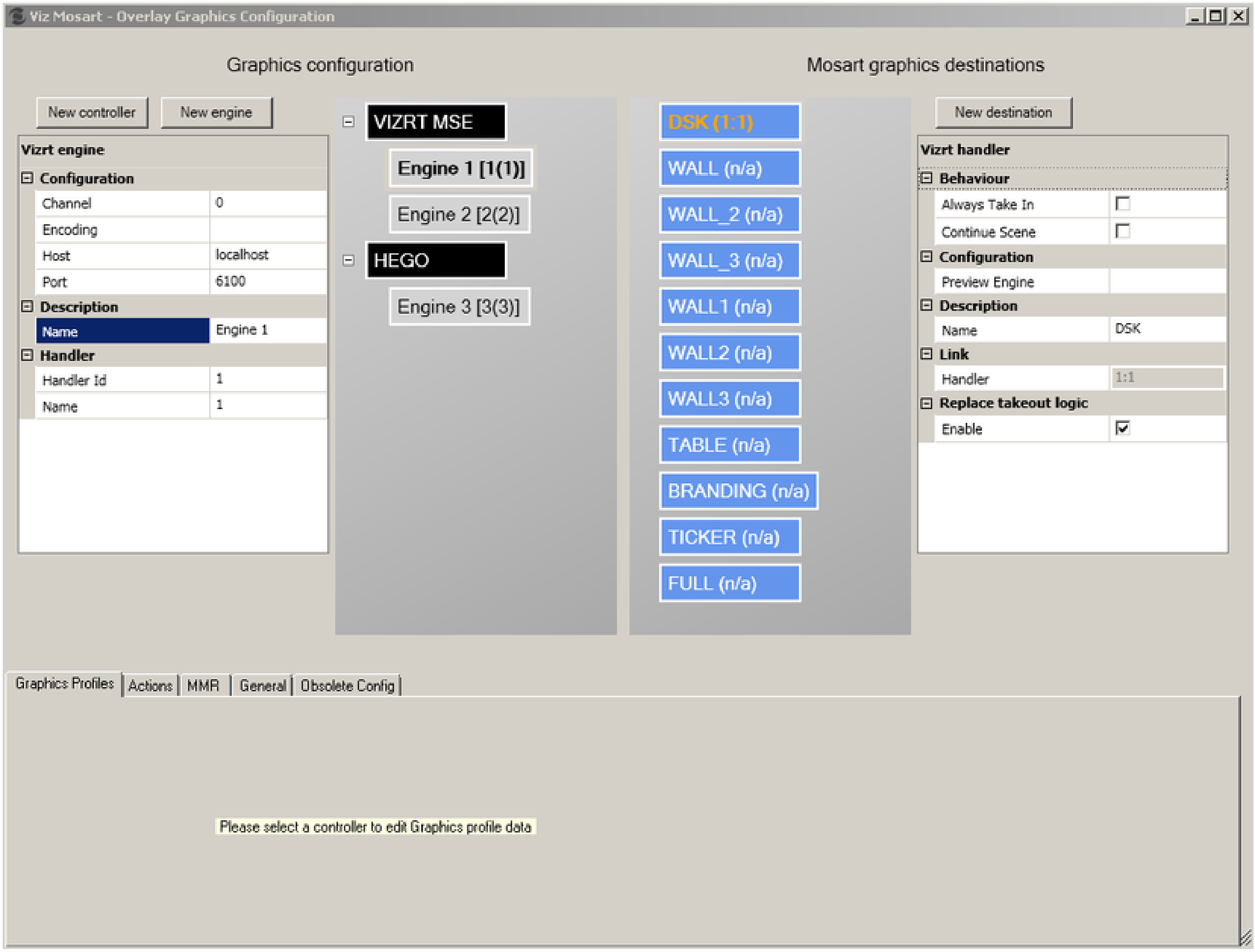
You can use this editor to test graphics.

7 Overlay Graphics Configuration

7.1 Overlay Graphics Configuration Window

The Overlay Graphics Configuration window is used to access the configuration options for Overlay Graphics Interface .

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



Vizrt engine

⊟ Configuration

Channel	0
Encoding	
Host	localhost
Port	6100

⊟ Description

Name	Engine 1

⊟ Handler

Handler Id	1
Name	1

Vizrt handler

⊟ Behaviour

Always Take In	☐
Continue Scene	☐

⊟ Configuration

Preview Engine	

⊟ Description

Name	DSK

⊟ Link

Handler	1:1

⊟ Replace takeout logic

Enable	☑

Please select a controller to edit Graphics profile data

On the left hand side are the Engines and Controllers, while on the right hand side are the Destinations. The Overlay Graphics Configuration - Property Tabs are available at the bottom of the window.

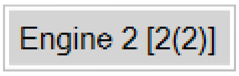
7.2 Controller

The controller defines the type of the graphic system that is going to be used. For each controller some common or specific properties can be configured. Details for each property is explained further. The connection to the graphic system is not going to be provided at this level, but for some graphic types, connection to some specific components used by the graphics system must be provided. For example the connection to Media Sequencer for Vizrt graphics systems.

See:

- Add Overlay Graphics Controllers and Engines
- Common Graphic Controller Properties
- Overlay Graphics Types
- Overlay Graphics Configuration - Property Tabs

7.3 Engine


Engine 2 [2(2)]

For each controller one or more Engines can be added. An Engine defines the connection to the graphics system.

See:

- Add Overlay Graphics Controllers and Engines
- Common Graphic Controller Properties
- Overlay Graphics Configuration - Property Tabs
- Mirrored Graphics Playout

7.4 Destination

DSK (1:1)

The destination defines the output channel where the graphics are going to be played out. Multiple engines can be assigned to:

- same destination (Mirrored Graphics Playout),
- to different destinations with different names,
- or to different destinations with 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.

The first of these will use the same graphics concept, the latter separate concepts for each output.

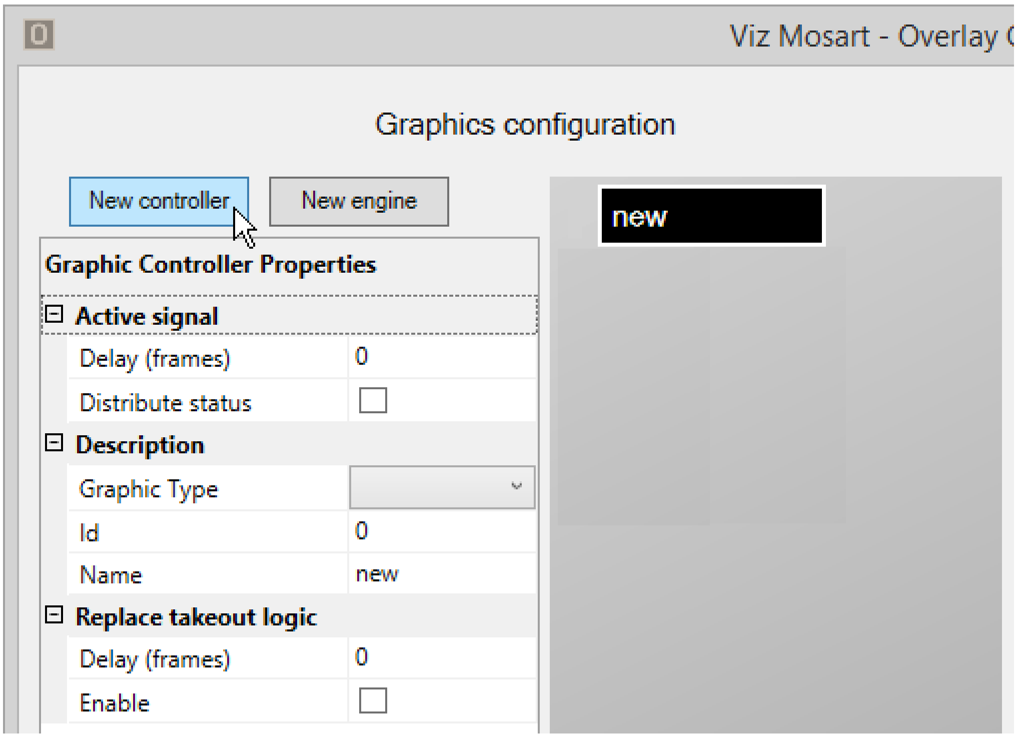
See:

- Add Mosart Graphics Destination
- Mirrored Graphics Playout

7.5 Add Overlay Graphics Controllers and Engines

7.5.1 To Add a New Controller

1. Open the Overlay Graphics Configuration window by going to *Settings* > *Properties*
2. Click the **New controller** button



Graphics configuration

Graphic Controller Properties

⊟ Active signal

Delay (frames)	0
Distribute status	□

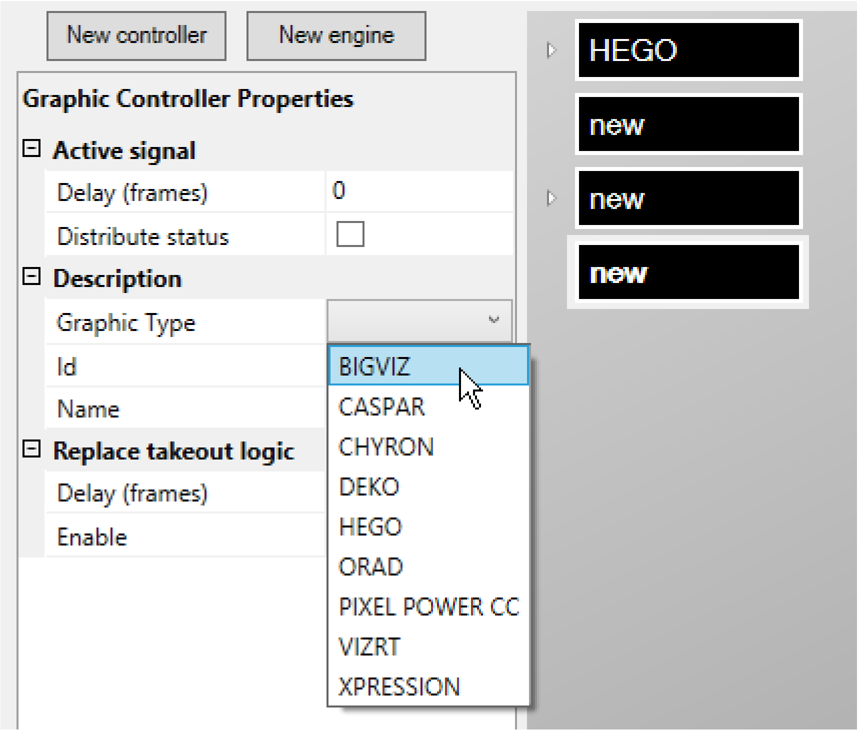
⊟ Description

Graphic Type	
Id	0
Name	new

⊟ Replace takeout logic

Delay (frames)	0
Enable	□

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



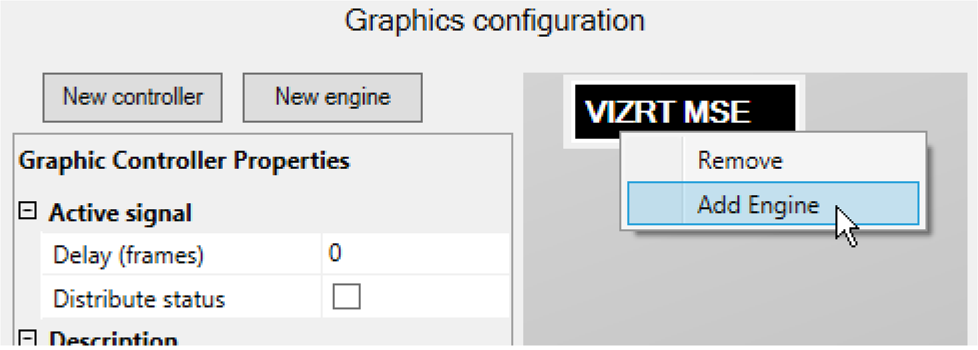
Graphic Controller Properties

⊟ Active signal
Delay (frames) 0
Distribute status □
⊟ Description
Graphic Type
Id
Name
⊟ Replace takeout logic
Delay (frames)
Enable

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.

7.5.2 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 that you want to use. (See how in To Add a New Engine, above).
3. Click the **New engine** button, or right-click the controller and select **Add engine.**



Graphic Controller Properties

⊟ Active signal
Delay (frames) 0
Distribute status □
⊟ Description

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).

7.6 Common Graphic Controller Properties

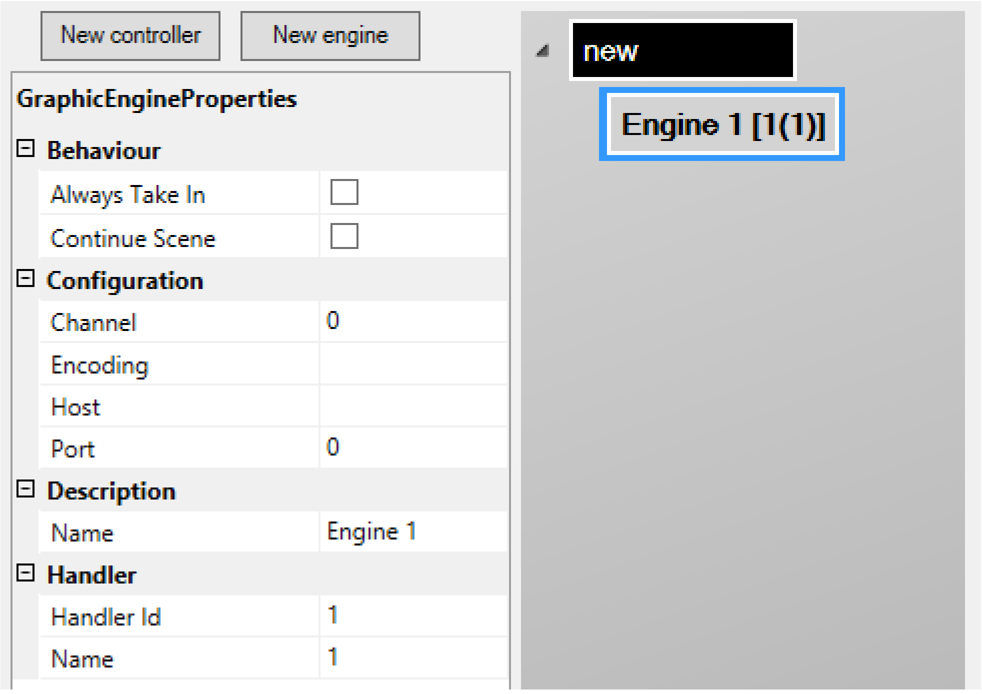
The **Graphic Controller Properties** that are available, depend on the Graphic Type of the controller. The controller properties common to most graphics types are described below.

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**: When true, signals the on-airs/usage status of an engine to AvAutomation. 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 will disable the functionality, i.e. *Distribute status* is false. AvAutomation will perform 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 the section "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**: Select the brand and type of graphics
  - **Id**: Unique number id identifying the graphics controller. Shall be unique among all defined controllers.
  - **Name**: Optional, defines the name of the server.
- **Replace takeout logic:**
  - Enabling takeout logic will read 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. 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: Also see the Graphics Profiles Tab for other controller settings.

7.7 Common Graphic Engine Properties



GraphicEngineProperties

Behaviour	
Always Take In	□
Continue Scene	□
Configuration	
Channel	0
Encoding	
Host	
Port	0
Description	
Name	Engine 1
Handler	
Handler Id	1
Name	1

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.

- **Behaviour**:
  - **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-16 or UTF-8. 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 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: Also see the Actions Tab (Event and Action Rules) for other engine settings.

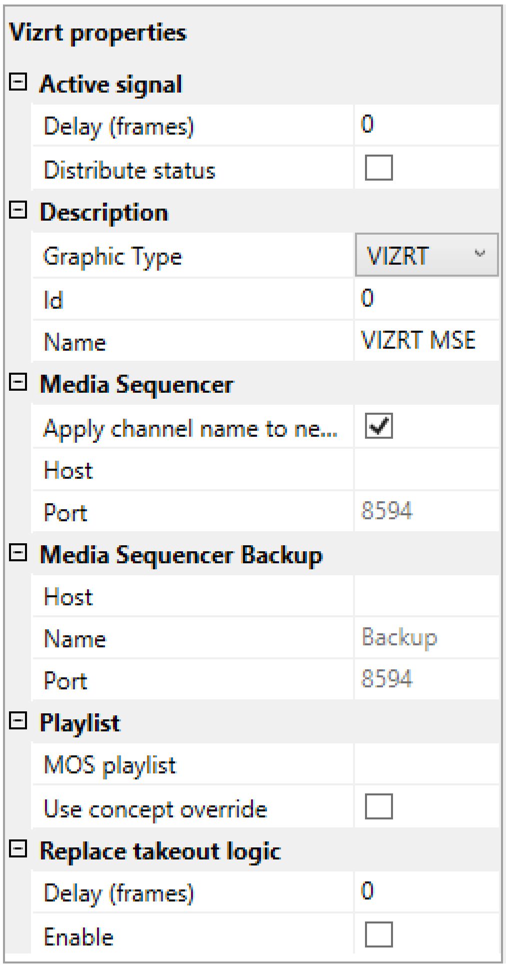
7.8 Overlay Graphics Types

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

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

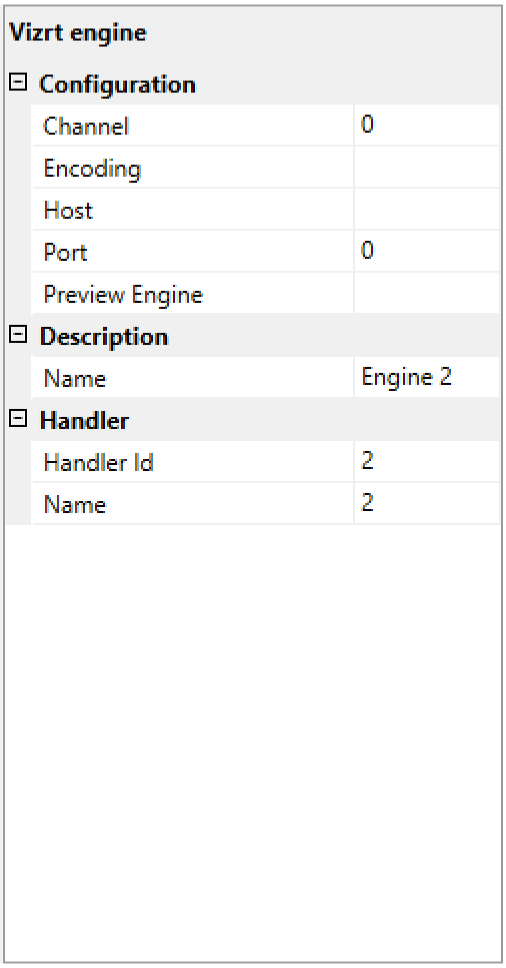
7.8.1 VIZRT

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



Vizrt properties

⊟ **Active signal**	
Delay (frames)	0
Distribute status	□
⊟ **Description**	
Graphic Type	VIZRT
Id	0
Name	VIZRT MSE
⊟ **Media Sequencer**	
Apply channel name to ne...	☑
Host	
Port	8594
⊟ **Media Sequencer Backup**	
Host	
Name	Backup
Port	8594
⊟ **Playlist**	
MOS playlist	
Use concept override	□
⊟ **Replace takeout logic**	
Delay (frames)	0
Enable	□



Vizrt engine

⊟ **Configuration**	
Channel	0
Encoding	
Host	
Port	0
Preview Engine	
⊟ **Description**	
Name	Engine 2
⊟ **Handler**	
Handler Id	2
Name	2

Vizrt properties:

Properties In addition to Common Graphic Controller Properties:

- **Media Sequencer**:

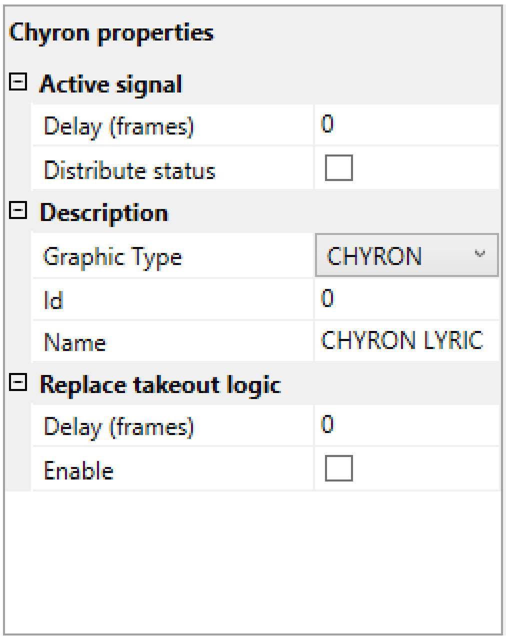
  - **Apply channel name to new elements**: Assigns automatically channel name to new graphics elements.
  - **Host**: hostname or IP address to the Media Sequencer
  - **Port**: IP-port to the Media Sequencer. Normally *8594* (Media Sequencer TreeTalk protocol) or *8580* (REST protocol).
- **Media Sequencer Backup**: For an optional backup MSE. Same properties as for Media Sequencer. Note that the "Apply channel name to new elements" property is common for both main and backup MSE's.
- **Playlist**:
  - **MOS playlist**: Name of playlist in MSE where graphics elements are stored.
  - **Use concept override**: Enables use of *concept override*. If concept override is enabled the currently selected graphics concept will override any concept defined within the graphice element. Use the Graphics Profile tab to define the various graphics concepts.

    Vizrt engine:

- Configuration:
  - **Host**: Hostname or IP address of the Viz Engine.
  - **Port**: Tcp/ip port of the Viz Engine.

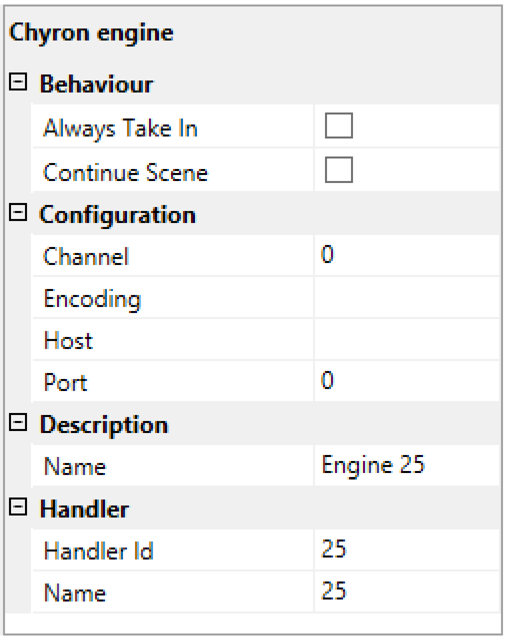
7.8.2 CHYRON

Graphic Type = Chyron (Chyron Lyric)



Chyron properties

⊟ **Active signal**	
Delay (frames)	0
Distribute status	□
⊟ **Description**	
Graphic Type	CHYRON
Id	0
Name	CHYRON LYRIC
⊟ **Replace takeout logic**	
Delay (frames)	0
Enable	□



Chyron engine

⊟ **Behaviour**	
Always Take In	□
Continue Scene	□
⊟ **Configuration**	
Channel	0
Encoding	
Host	
Port	0
⊟ **Description**	
Name	Engine 25
⊟ **Handler**	
Handler Id	25
Name	25

Chyron properties:

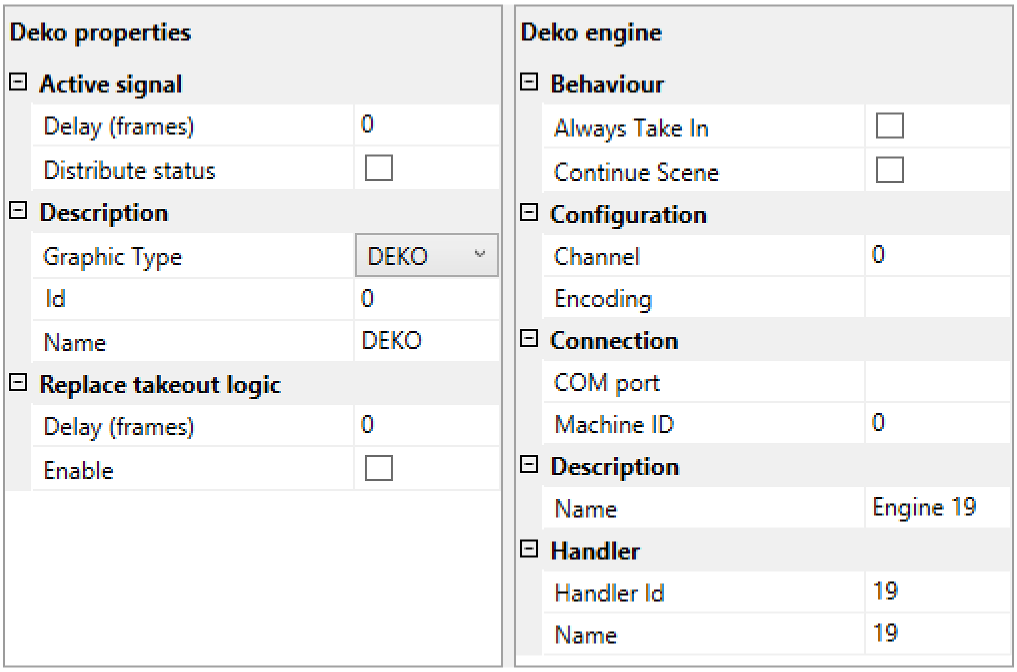
The properties are the same as described in Common Graphic Controller Properties.

Chyron engine:

The properties are the same as described in Common Graphic Engine Properties.

7.8.3 DEKO

Graphic Type = Deko



Deko properties

⊟ **Active signal**	
Delay (frames)	0
Distribute status	□
⊟ **Description**	
Graphic Type	DEKO
Id	0
Name	DEKO
⊟ **Replace takeout logic**	
Delay (frames)	0
Enable	□

Deko engine

⊟ **Behaviour**	
Always Take In	□
Continue Scene	□
⊟ **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.

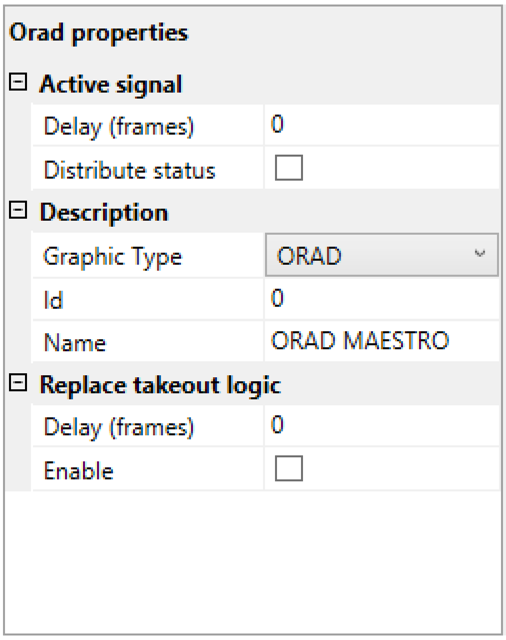
Deko engine:

In addition to Common Graphic Engine Properties, are the following:

- **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 allows to specify different field or layer (TypeDeko) when more than one TypeDeko is connected to the Deko host. Set equal to 1 if there is only one.

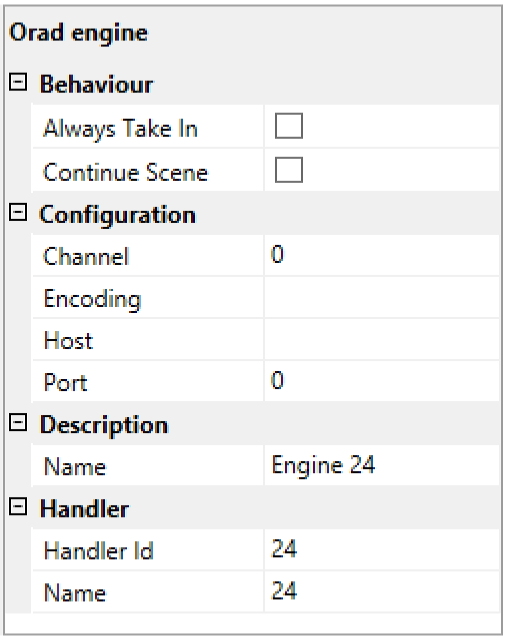
7.8.4 ORAD

Graphic Type = Orad (Orad Maestro)



Orad properties

⊟ Active signal	
Delay (frames)	0
Distribute status	□
⊟ Description	
Graphic Type	ORAD
Id	0
Name	ORAD MAESTRO
⊟ Replace takeout logic	
Delay (frames)	0
Enable	□



Orad engine

⊟ Behaviour	
Always Take In	□
Continue Scene	□
⊟ Configuration	
Channel	0
Encoding	
Host	
Port	0
⊟ Description	
Name	Engine 24
⊟ Handler	
Handler Id	24
Name	24

Orad properties:

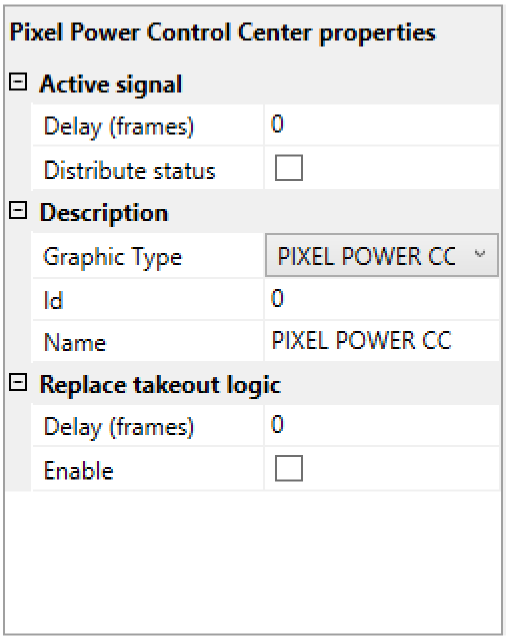
The properties are the same as described in Common Graphic Controller Properties.

Orad engine:

The properties are the same as described in Common Graphic Engine Properties.

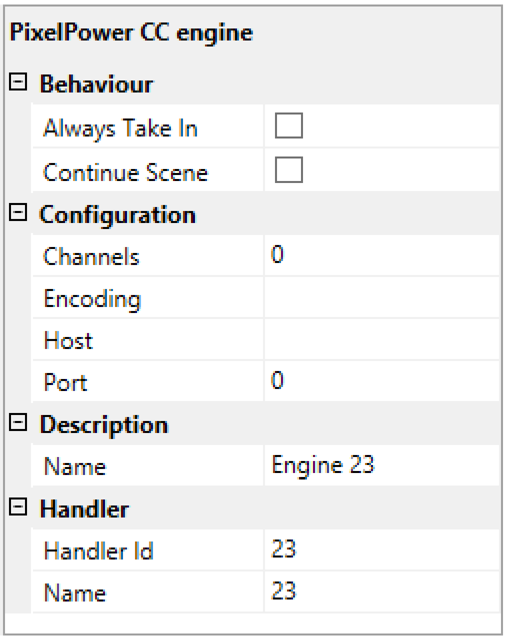
7.8.5 PIXEL POWER CC

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



Pixel Power Control Center properties

⊟ Active signal	
Delay (frames)	0
Distribute status	□
⊟ Description	
Graphic Type	PIXEL POWER CC
Id	0
Name	PIXEL POWER CC
⊟ Replace takeout logic	
Delay (frames)	0
Enable	□



PixelPower CC engine

⊟ Behaviour	
Always Take In	□
Continue Scene	□
⊟ 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:

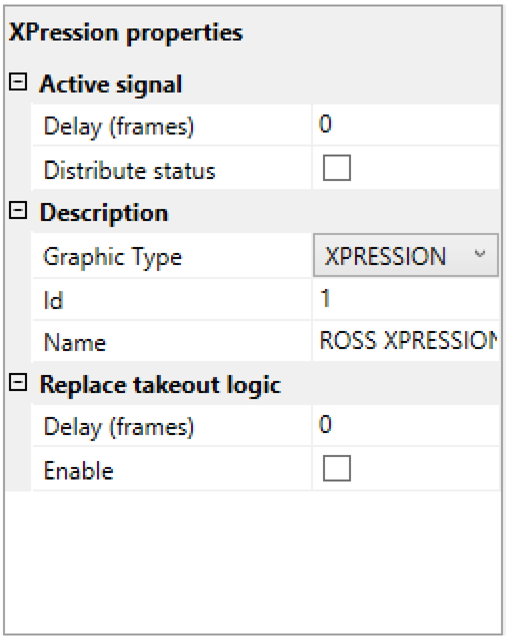
The properties are the same as described in Common Graphic Engine Properties.

Configuration:

- **Channels**: Use to specify the Pixel Power channels associated with the Pixel Power engine. Specify as a comma separated list of integers.

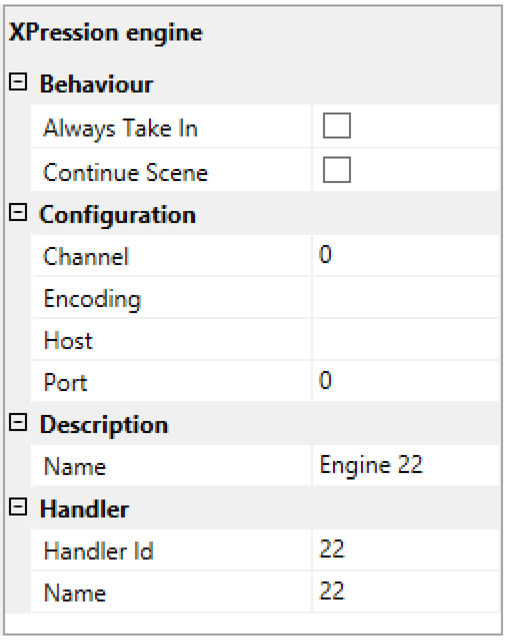
7.8.6 XPRESSION

Graphic Type = Xpression (Ross Xpression)



XPression properties

⊟ **Active signal**	
Delay (frames)	0
Distribute status	□
⊟ **Description**	
Graphic Type	XPRESSION
Id	1
Name	ROSS XPRESSIO
⊟ **Replace takeout logic**	
Delay (frames)	0
Enable	□



XPression engine

⊟ **Behaviour**	
Always Take In	□
Continue Scene	□
⊟ **Configuration**	
Channel	0
Encoding	
Host	
Port	0
⊟ **Description**	
Name	Engine 22
⊟ **Handler**	
Handler Id	22
Name	22

XPression properties:

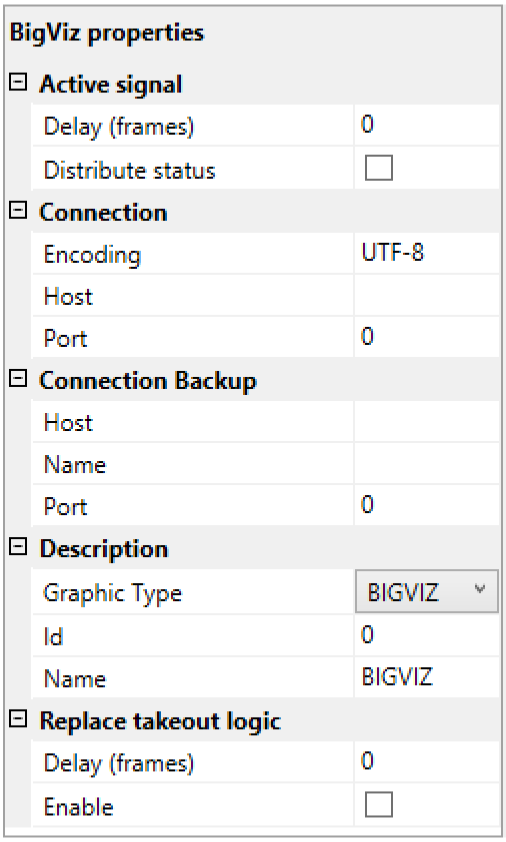
The properties are the same as described in Common Graphic Controller Properties.

XPression engine:

The properties are the same as described in Common Graphic Engine Properties.

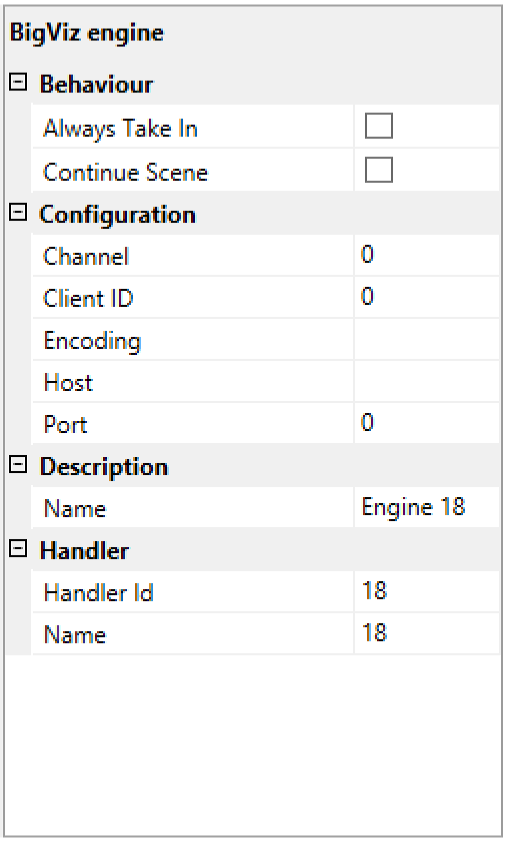
7.8.7 BIGVIZ

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	□
⊟ **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	□



BigViz engine

⊟ **Behaviour**	
Always Take In	□
Continue Scene	□
⊟ **Configuration**	
Channel	0
Client ID	0
Encoding	
Host	
Port	0
⊟ **Description**	
Name	Engine 18
⊟ **Handler**	
Handler Id	18
Name	18

BigViz properties:

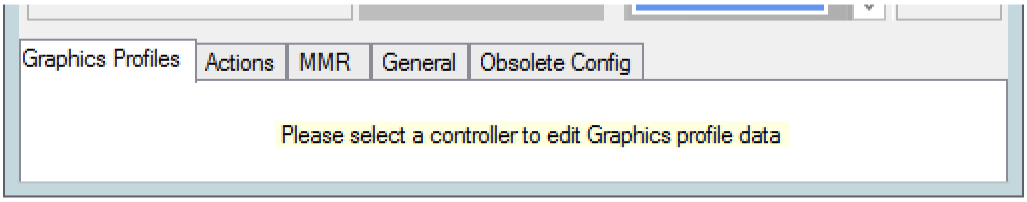
Controller connection properties are not used. All connectivity to BigViz is configured as part of the Engine configuration.

BigViz engine:

The following properties are used. 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.

7.9 Overlay Graphics Configuration - Property Tabs



Please select a controller to edit Graphics profile data

The tabs functions are:

- Graphics Profiles Tab
- Actions Tab (Event and Action Rules)
- MMR Tab
- General Tab
- Obsolete Config Tab

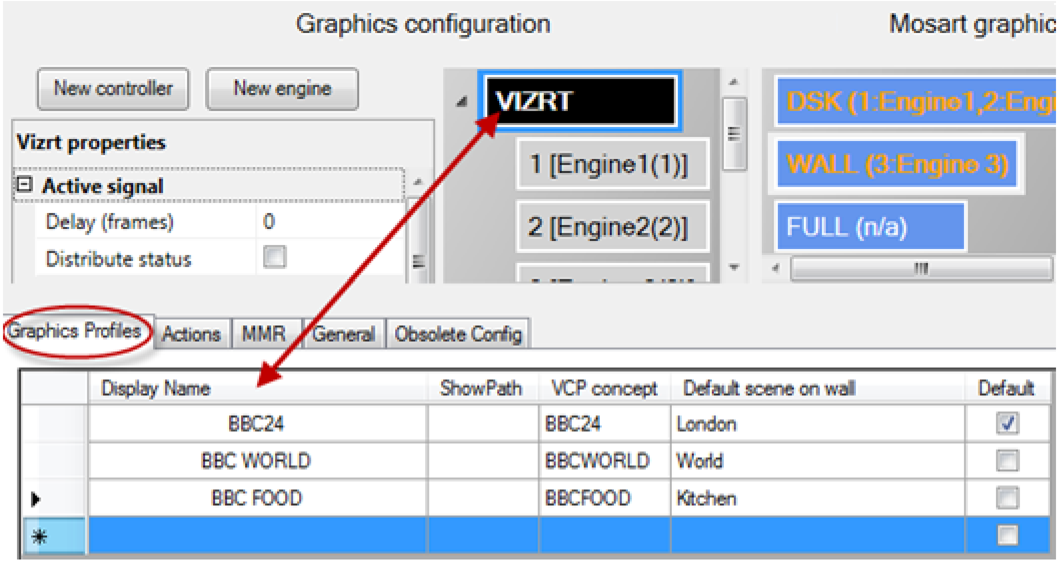
7.9.1 Graphics Profiles Tab

The selection of graphics profile will change the look of all graphics used for the specific show. Viz Mosart will take care of which concept the graphics devices will 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 Overlay Graphics Interface or Viz Mosart GUI it will look through all the controllers. If a controller has that particular graphics profile, it will change 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.



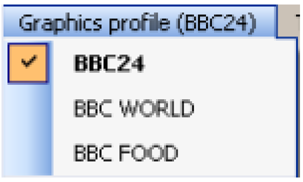
Delay (frames) 0

Distribute status □

	Display Name	ShowPath	VCP concept	Default scene on wall	Default
	BBC24		BBC24	London	☑
	BBC WORLD		BBCWORLD	World	□
▶	BBC FOOD		BBCFOOD	Kitchen	□
*					□

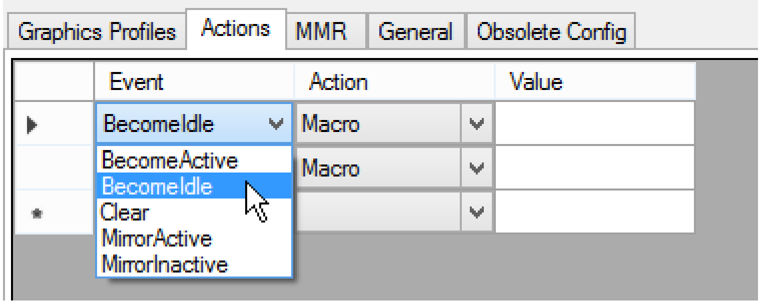
- **Display Name**: Name of the Graphics Profile that will be displayed to the user. The Graphics Profiles are displayed as menu options both in the Mosart GUI and in Overlay Graphics.
- **ShowPath**: Path to the files the graphics engine uses. Absolute path where graphics are located within the graphics engines. Dependent upon graphics type.
- **VCP concept** (Vizrt only): Name of the Viz Pilot concept that should be used when working with the selected profile.
- **Default scene on wall** Scene name to be used as default for wall graphics. Dependant upon Graphics Type.

- **Default**: Selects the graphics profile to be used 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.



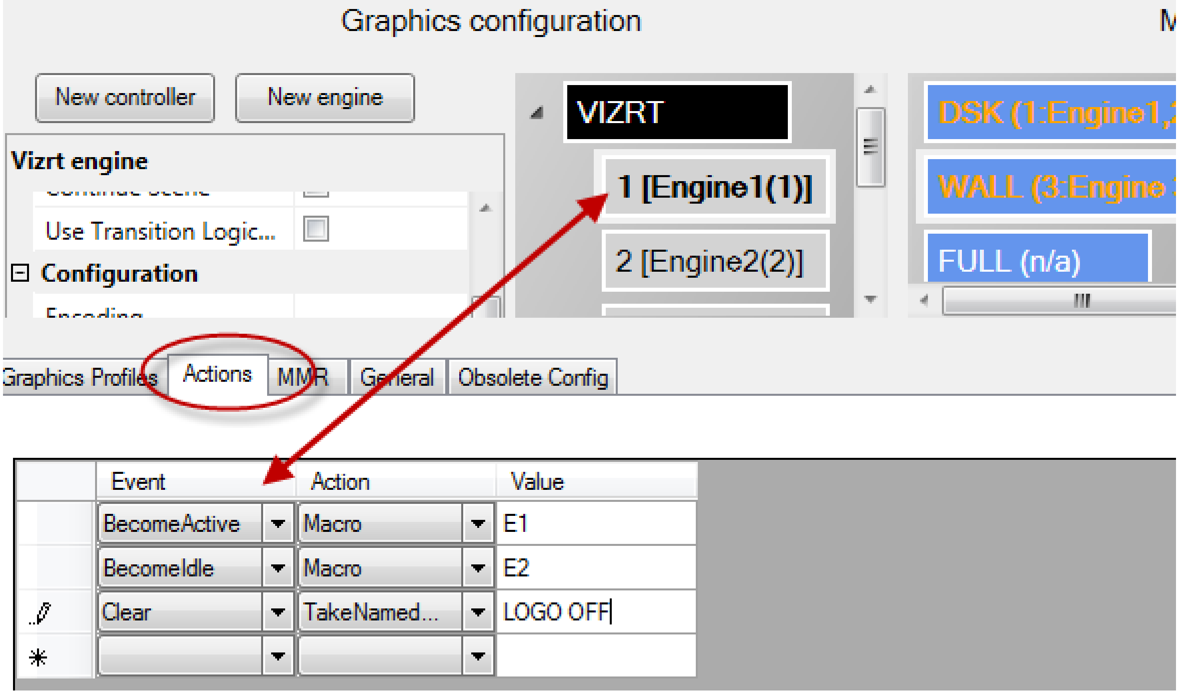
7.9.2 Actions Tab (Event and Action Rules)

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



Event	Action	Value
BecomeIdle	Macro	
	Macro	

In the Actions tab, any number of event-action rules can be setup. 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

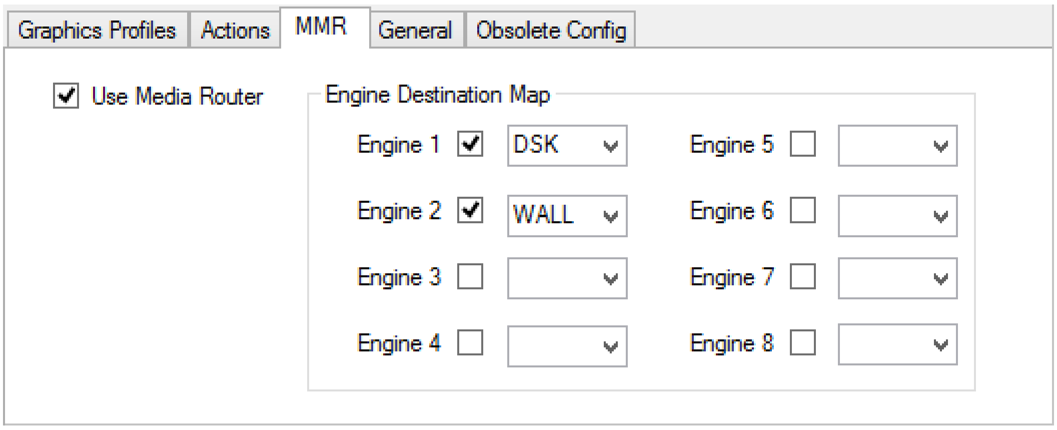
Three parameters are defined in order to set up an event-action rule:

- **Event:** Specifies the event that will trigger an action. Currently the following events are available:
  - **BecomeActive:** Trigger **action** whenever the Overlay Graphics Interface state changes from “Idle” to “Active”.
  - **BecomeIdle:** Trigger **action** whenever the Overlay Graphics Interface state changes from “Active” to “Idle”.
  - **Clear:** Trigger **action** whenever a Viz Mosart Clear event is received to clear all overlay graphics. A clear event is typically received when a rundown is reloaded or deleted.
  - **MirrorActive:** Trigger **action** when the mirroring graphics engine becomes active (for mirroring). Can only be used only when using Mirrored Graphics Playout
  - **MirrorInActive:** Trigger **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. Currently two actions are available:
  - **Macro:** Will invoke a macro on the given engine. The macro to take is specified in the Value parameter.
  - **TakeNamedOverlay:** Will invoke a named (or constant) CG on the given engine. The named CG to take is specified in the Value parameter.
- **Value:** Optional value depending upon Action. Value depending upon selected Action.
  As shown in the figure above, there may be **multiple event-action rules** for a particular 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.

7.9.3 MMR Tab

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



☑ Use Media Router

Engine Destination Map

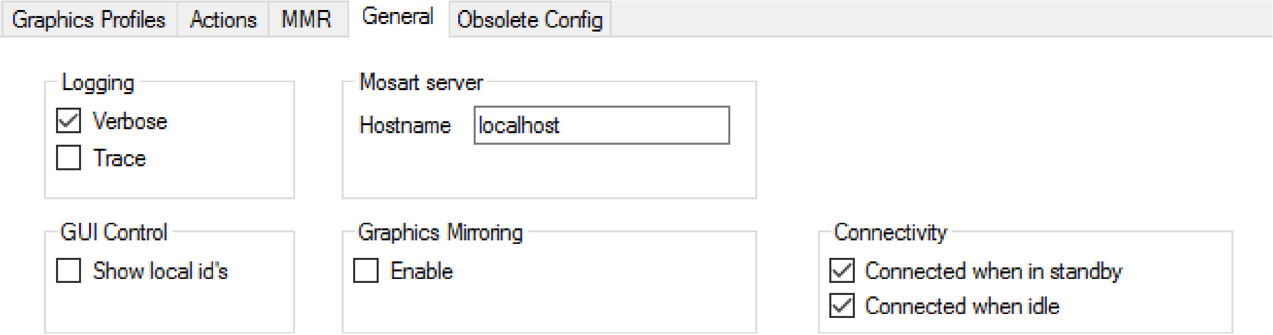
Engine 1 ☑ DSK
Engine 2 ☑ WALL
Engine 3 □
Engine 4 □
Engine 5 □
Engine 6 □
Engine 7 □
Engine 8 □

- **Use Media Router**: In order 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 to a maximum of 8 engines. In the example in the figure above the MMR configuration specifies two overlay graphics engines with number 1 and 2 respectively.

7.9.4 General Tab

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



Logging
☑ Verbose
☐ Trace

Mosart server
Hostname localhost

GUI Control
☐ Show local id's

Graphics Mirroring
☐ Enable

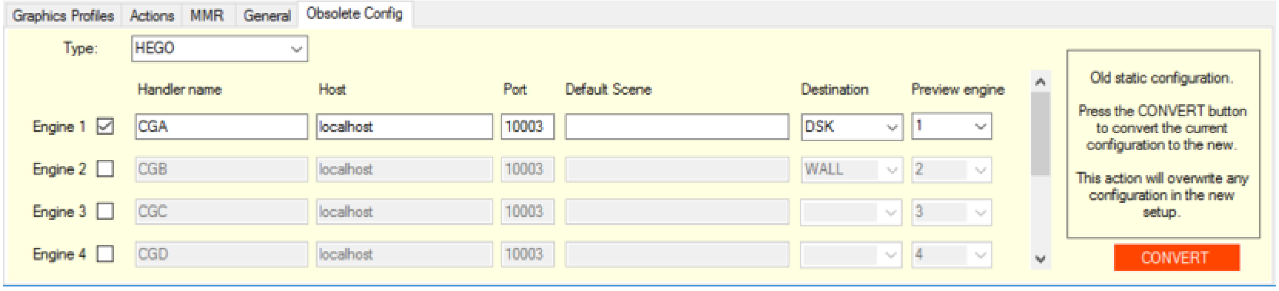
Connectivity
☑ Connected when in standby
☑ Connected when idle

- **Logging**:
  - **Verbose**: If enabled, increases the detail level of information sent to the Viz Mosart log.
  - **Trace**: If enabled, internal tracing is activated (more details are being sent to the debug view). 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**: 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**: If enabled, graphics devices will stay connected when in standby or idle modes.
    - **Connected when idle**: If enabled, Graphics devices will stay connected when put on idle, for example when switching from main to backup (Default = disabled).

Note: A similar parameter is available for full screen graphics, see _GraphicsConnectedWhenIdle _in AV Automation Settings (see AV Automation Devices - Graphics).

7.9.5 Obsolete Config Tab

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



Type: HEGO

	Handler name	Host	Port	Default Scene	Destination	Preview engine
Engine 1 ☑	CGA	localhost	10003		DSK	1
Engine 2 ☐	CGB	localhost	10003		WALL	2
Engine 3 ☐	CGC	localhost	10003			3
Engine 4 ☐	CGD	localhost	10003			4

Old static configuration.

Press the CONVERT button to convert the current configuration to the new.

This action will overwrite any configuration in the new setup.

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 we strongly recommend converting 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.

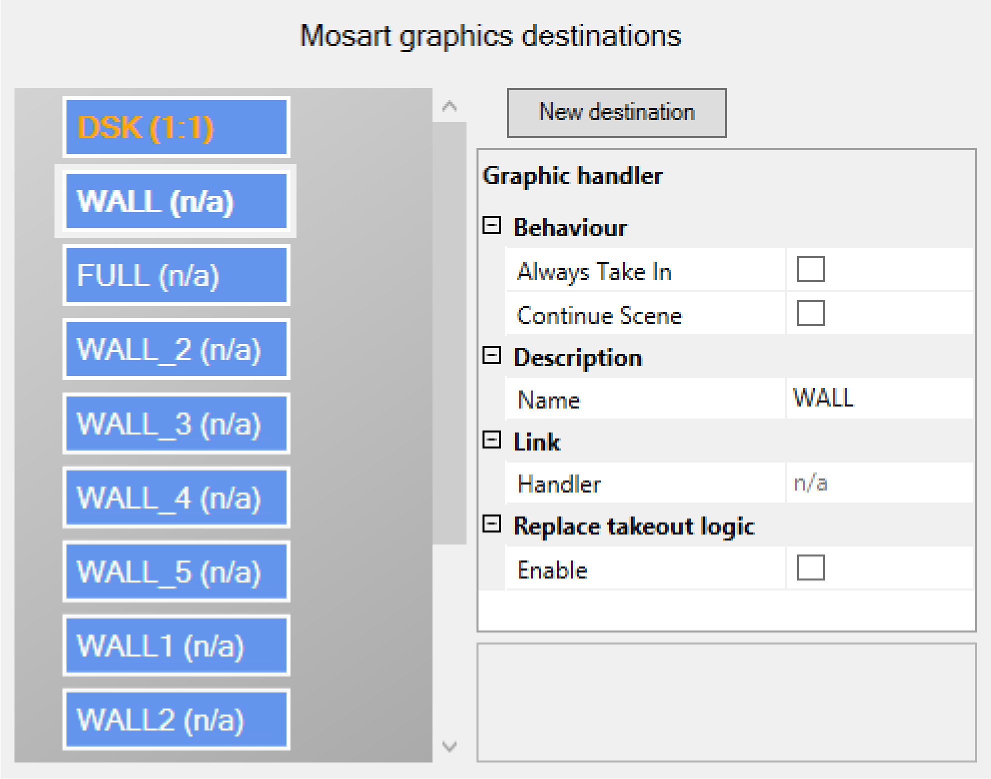
To convert old configurations to the new standard, click the **CONVERT** button. This will overwrite any existing standard configuration.

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 will be 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.

7.10 Add Mosart Graphics Destination

Overlay Graphics Interface provides a set of standard *Mosart graphics destinations *including WALL handlers (WALL, WALL_2 to WALL_5), FULL, and TABLE. It is recommended that you keep these as is.



Graphic handler

⊟ Behaviour

Always Take In	☐
Continue Scene	☐

⊟ Description

Name	WALL

⊟ Link

Handler	n/a

⊟ Replace takeout logic

Enable	☐

To add a new Mosart graphics destination, click the **New destination** button, then fill in the *Graphic Handler Properties*.

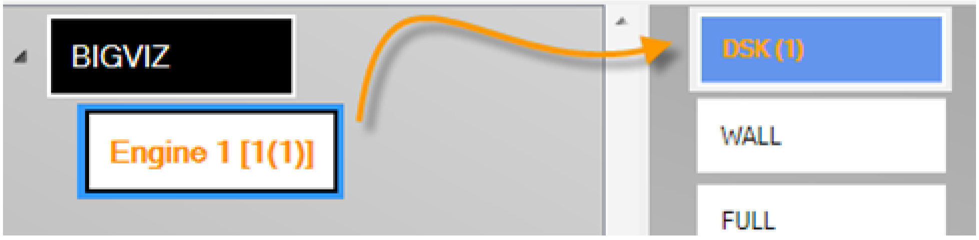
7.10.1 Graphic Handler Properties

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 be shown: *Description* and *Link*.

- **Behaviour**:
  - **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**: Enabling this will read 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.

7.10.2 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.



7.10.3 To remove engine links from a destination

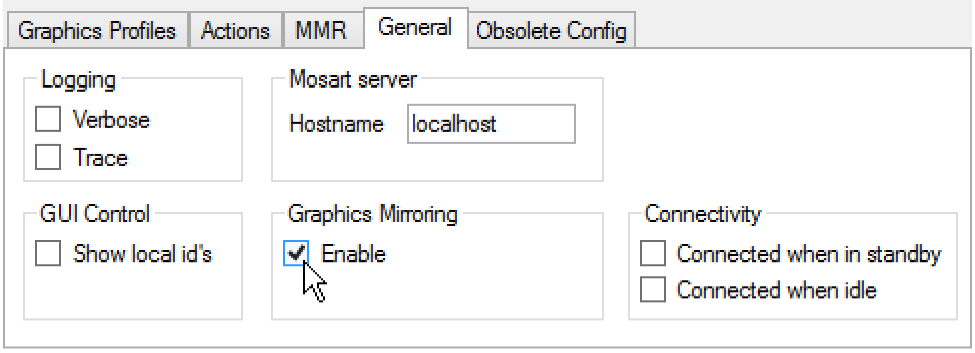
Engine links are removed from a destination using the Remove menu option from the right-click context menu. I.e. select an Engine, then right-click on the engine and from the context-menu select Remove.

7.10.4 Mirrored Graphics Playout

This section describes how to play out the same graphics in multiple graphics engines. This functionality is supported both for overlay graphics and full-screen graphics (AV Automation). This section describes the approach used in Overlay Graphics Interface.

Configure mirrored graphics playout in Overlay Graphics Interface as follows:

1. Enable Mirroring in the General Tab.



Graphics Profiles | Actions | MMR | General | Obsolete Config

Logging
□ Verbose
□ Trace

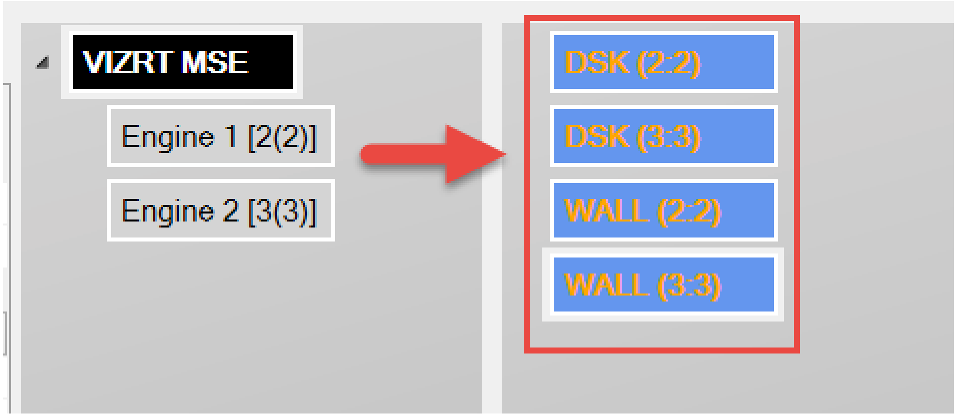
Mosart server
Hostname: localhost

GUI Control
□ Show local id's

Graphics Mirroring
☑ Enable

Connectivity
□ Connected when in standby
□ Connected when idle

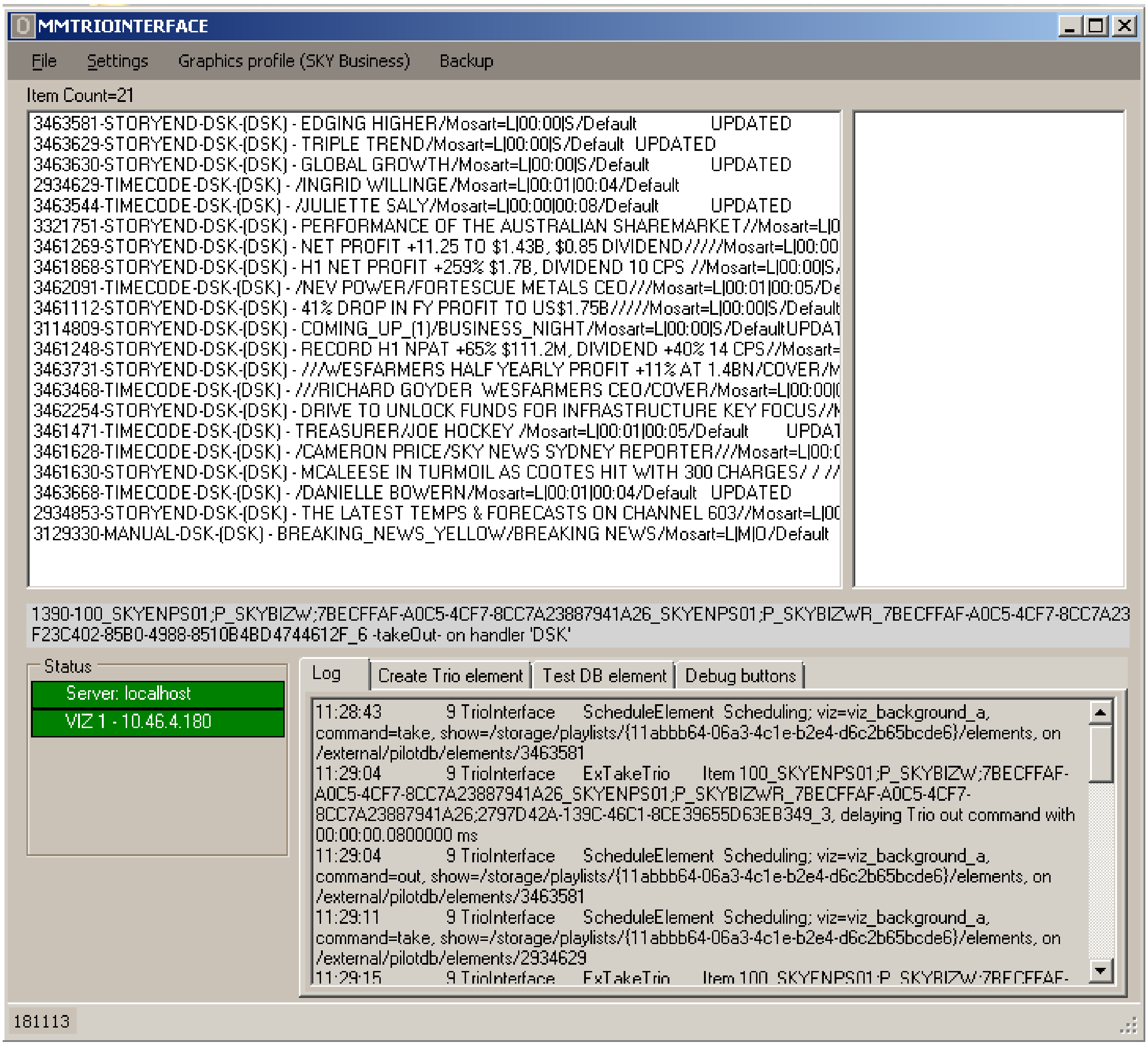
2. Set up multiple graphics engines with the same destination.
   Example: To configure mirrored graphics for a destination create two destinations with the same name and assign a different engine to each destination. In the example below mirrored graphics is enabled for DSK and WALL destinations.



3. Optional configure actions to be triggered when changing the active mirrored graphics engine. Do this from the Actions Tab (Event and Action Rules).

8 Trio Interface

IMPORTANT! 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.



```
3463581-STORYEND-DSK-(DSK) - EDGING HIGHER/Mosart=L|00:00|S/Default          UPDATED
3463629-STORYEND-DSK-(DSK) - TRIPLE TREND/Mosart=L|00:00|S/Default  UPDATED
3463630-STORYEND-DSK-(DSK) - GLOBAL GROWTH/Mosart=L|00:00|S/Default          UPDATED
2934629-TIMECODE-DSK-(DSK) - /INGRID WILLINGE/Mosart=L|00:01|00:04/Default
3463544-TIMECODE-DSK-(DSK) - /JULIETTE SALY/Mosart=L|00:00|00:08/Default         UPDATED
3321751-STORYEND-DSK-(DSK) - PERFORMANCE OF THE AUSTRALIAN SHAREMARKET//Mosart=L|0
3461269-STORYEND-DSK-(DSK) - NET PROFIT +11.25 TO $1.43B, $0.85 DIVIDEND/////Mosart=L|00:00
3461868-STORYEND-DSK-(DSK) - H1 NET PROFIT +259% $1.7B, DIVIDEND 10 CPS //Mosart=L|00:00|S/
3462091-TIMECODE-DSK-(DSK) - /NEV POWER/FORTESCUE METALS CEO///Mosart=L|00:01|00:05/De
3461112-STORYEND-DSK-(DSK) - 41% DROP IN FY PROFIT TO US$1.75B/////Mosart=L|00:00|S/Default
3114809-STORYEND-DSK-(DSK) - COMING_UP_(1)/BUSINESS_NIGHT/Mosart=L|00:00|S/DefaultUPDAT
3461248-STORYEND-DSK-(DSK) - RECORD H1 NPAT +65% $111.2M, DIVIDEND +40% 14 CPS//Mosart=
3463731-STORYEND-DSK-(DSK) - ///WESFARMERS HALF YEARLY PROFIT +11% AT 1.4BN/COVER/M
3463468-TIMECODE-DSK-(DSK) - ///RICHARD GOYDER  WESFARMERS CEO/COVER/Mosart=L|00:00|(
3462254-STORYEND-DSK-(DSK) - DRIVE TO UNLOCK FUNDS FOR INFRASTRUCTURE KEY FOCUS//M
3461471-TIMECODE-DSK-(DSK) - TREASURER/JOE HOCKEY /Mosart=L|00:01|00:05/Default       UPDAT
3461628-TIMECODE-DSK-(DSK) - /CAMERON PRICE/SKY NEWS SYDNEY REPORTER///Mosart=L|00:(
3461630-STORYEND-DSK-(DSK) - MCALEESE IN TURMOIL AS COOTES HIT WITH 300 CHARGES/ / //
3463668-TIMECODE-DSK-(DSK) - /DANIELLE BOWERN/Mosart=L|00:01|00:04/Default  UPDATED
2934853-STORYEND-DSK-(DSK) - THE LATEST TEMPS & FORECASTS ON CHANNEL 603//Mosart=L|0(
3129330-MANUAL-DSK-(DSK) - BREAKING_NEWS_YELLOW/BREAKING NEWS/Mosart=L|M|0/Default
```

```
1390-100_SKYENPS01;P_SKYBIZW;7BECFFAF-A0C5-4CF7-8CC7A23887941A26_SKYENPS01;P_SKYBIZWR_7BECFFAF-A0C5-4CF7-8CC7A23
F23C402-85B0-4988-8510B4BD4744612F_6 -takeOut- on handler 'DSK'
```

```
11:28:43          9 TrioInterface     ScheduleElement  Scheduling; viz=viz_background_a,
command=take, show=/storage/playlists/{11abbb64-06a3-4c1e-b2e4-d6c2b65bcde6}/elements, on
/external/pilotdb/elements/3463581
11:29:04          9 TrioInterface     ExTakeTrio     Item 100_SKYENPS01;P_SKYBIZW;7BECFFAF-
A0C5-4CF7-8CC7A23887941A26_SKYENPS01;P_SKYBIZWR_7BECFFAF-A0C5-4CF7-
8CC7A23887941A26;2797D42A-139C-46C1-8CE39655D63EB349_3, delaying Trio out command with
00:00:00.0800000 ms
11:29:04          9 TrioInterface     ScheduleElement  Scheduling; viz=viz_background_a,
command=out, show=/storage/playlists/{11abbb64-06a3-4c1e-b2e4-d6c2b65bcde6}/elements, on
/external/pilotdb/elements/3463581
11:29:11          9 TrioInterface     ScheduleElement  Scheduling; viz=viz_background_a,
command=take, show=/storage/playlists/{11abbb64-06a3-4c1e-b2e4-d6c2b65bcde6}/elements, on
/external/pilotdb/elements/2934629
11:29:15          9 TrioInterface     ExTakeTrio     Item 100_SKYENPS01;P_SKYBIZW;7BECFFAF-
```

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

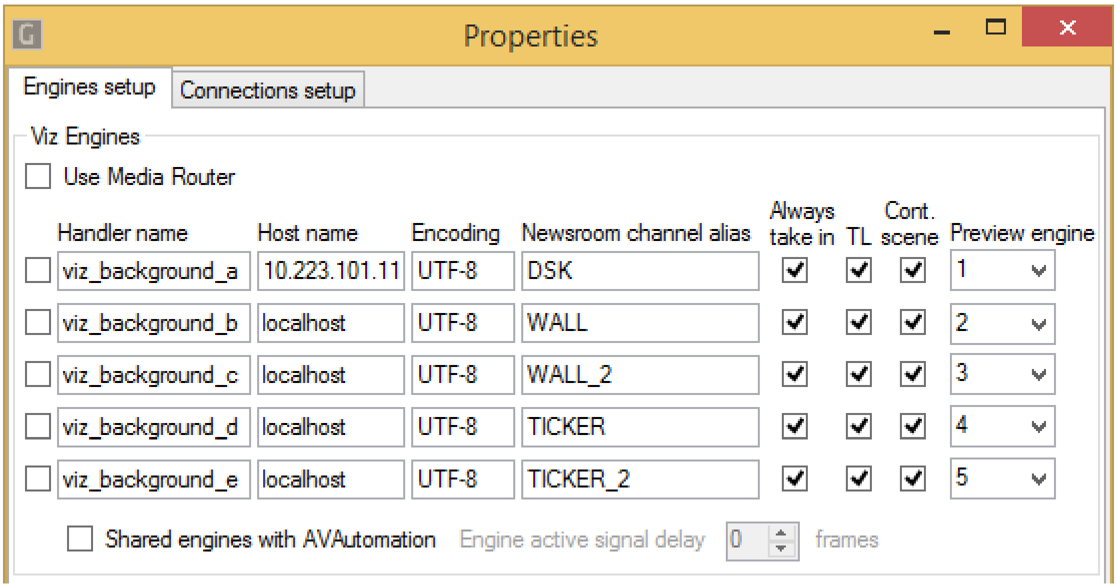
The Trio Interface Configuration window is used to access the configuration options for **Trio Interface**.

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

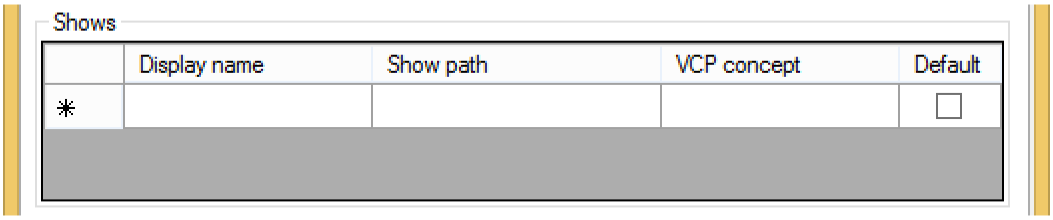
	Handler name	Host name	Encoding	Newsroom channel alias	Always take in	TL	Cont. scene	Preview engine
☐	viz_background_a	10.223.101.11	UTF-8	DSK	☑	☑	☑	1
☐	viz_background_b	localhost	UTF-8	WALL	☑	☑	☑	2
☐	viz_background_c	localhost	UTF-8	WALL_2	☑	☑	☑	3
☐	viz_background_d	localhost	UTF-8	TICKER	☑	☑	☑	4
☐	viz_background_e	localhost	UTF-8	TICKER_2	☑	☑	☑	5

☐ Shared engines with AVAutomation   Engine active signal delay 0 frames

- **Viz Engines**
  - **Use Media Router**: Check for using Media router.
  - **Viz Engines**: Use the check box to activate or deactivate connection to the Viz Engines configured in the corresponding row.
  - **Handler name**: Internal Media Sequencer handler name.
  - **Host name**: IP address or hostname of the 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**: Check to always force a take-in command to be sent.
  - **TL**: Check to specify that Transition Logic shall be used.
  - **Cont. scene**: Use Continue for takeout of scene based graphics.
  - **Preview engine**: Indicate which Vizrt engine to be used for preview of overlays.

- **Shared engines with AVAutomation**: This functionality will assume 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**

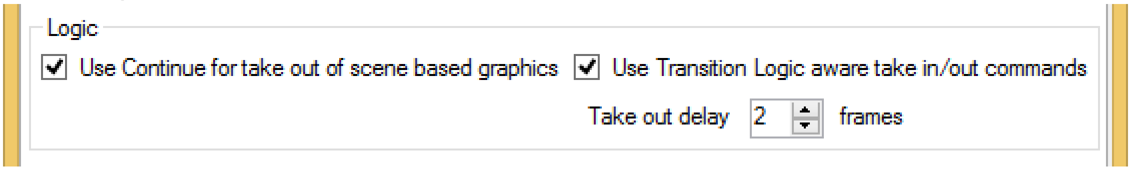


Shows

	Display name	Show path	VCP concept	Default
*				☐

- **Display name**: The name of the show that will be 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**: Viz Trio path for the show.
- **VCP concept**: Name of the Viz Pilot concept.
- **Default**: Enable to make the row the default show for the gallery.

- **Logic**- Transition Logic Configuration



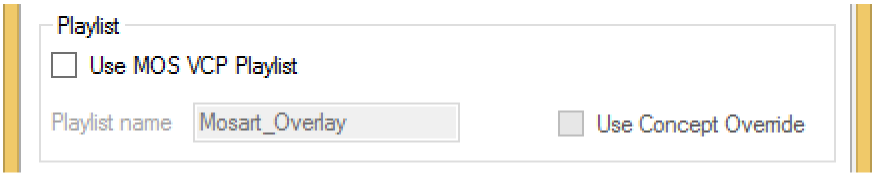
Logic

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

Take out delay 2 frames

- **Use Continue for take out of scene based graphics**: Enable check box 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**: Check to activate 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.
- **Take out delay**: Delay before takeout. Default: Empty (ss:ff)

- **Playlist**



Playlist

☐ Use MOS VCP Playlist

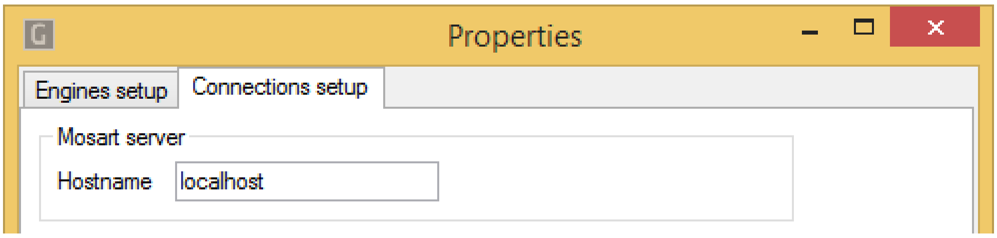
Playlist name Mosart_Overlay ☐ Use Concept Override

- **Use MOS VCP Playlist**: Enable the Playlist check box when using Vizrt MOS items from the newsroom system.
- **Playlist name**: VizPilot 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**: Enable 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**

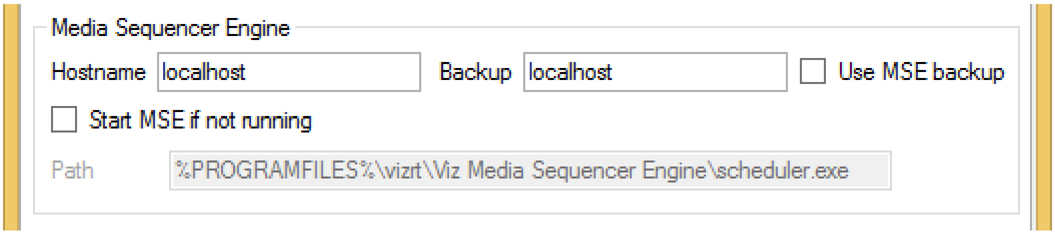


Mosart server

Hostname localhost

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

- **Media Sequencer Engine**



Media Sequencer Engine

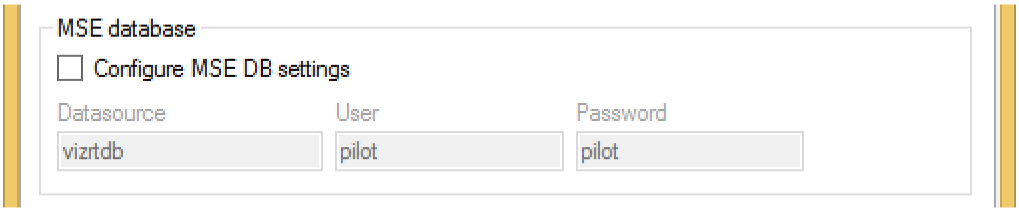
Hostname localhost Backup localhost ☐ Use MSE backup

☐ Start MSE if not running

Path %PROGRAMFILES%\vizrt\Viz Media Sequencer Engine\scheduler.exe

  - **Hostname**: Hostname or IP address of the primary Media Sequencer.
  - **Backup**: Hostname or IP address of the backup Media Sequencer.
  - **Use MSE backup**: Check this box to initially start using the backup Media Sequencer instead of the primary.
  - **Start MSE if not running**: If the Media Sequencer is not running when Trio Interface is started, it will try to start from the location described under (only valid if the Media Sequencer is running as a console application on the same machine).
  - **Path**: Path to the 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**



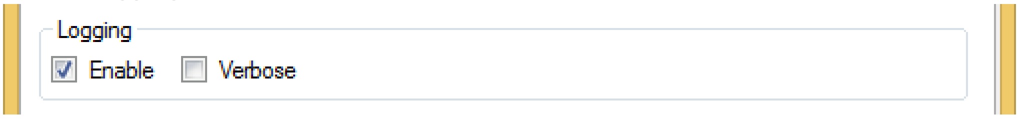
MSE database

☐ Configure MSE DB settings

Datasource	User	Password
vizrtdb	pilot	pilot

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

- **Logging**



Logging

☑ Enable ☐ Verbose

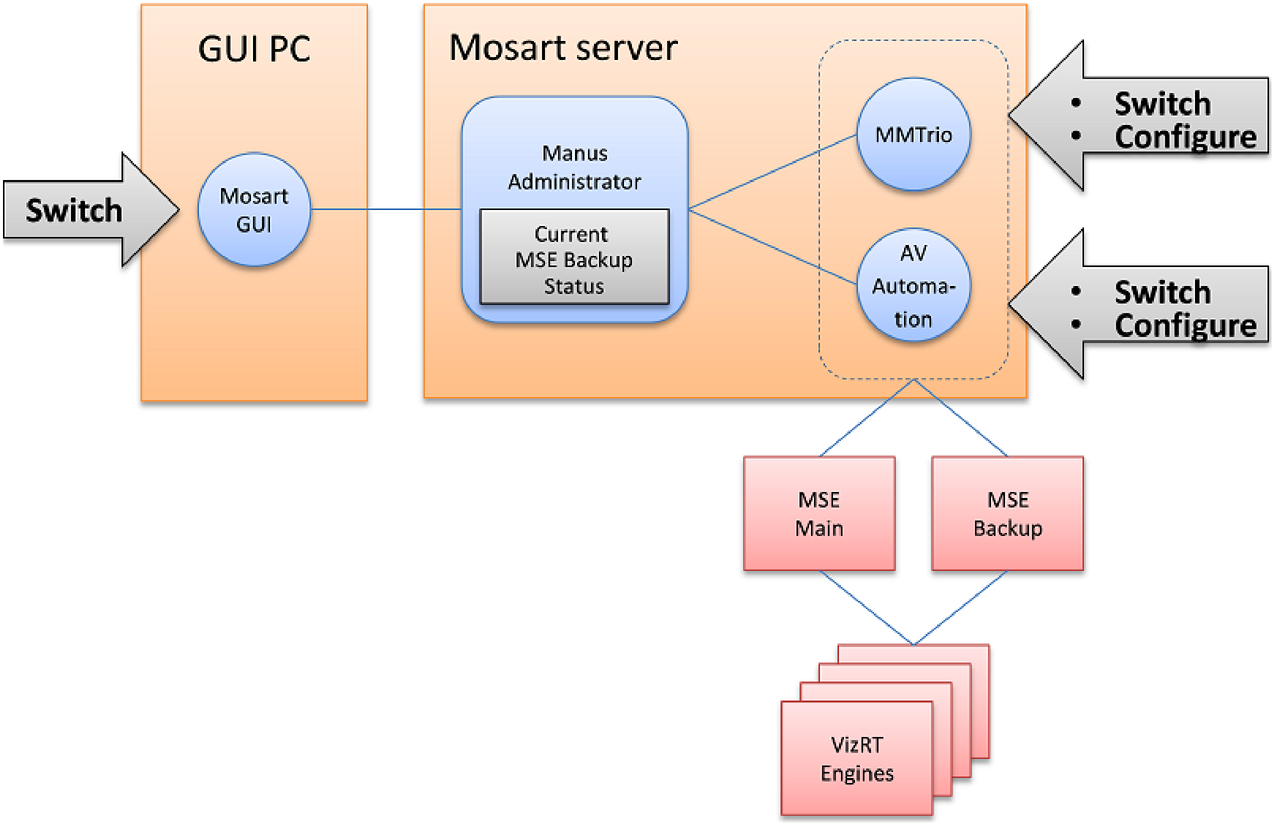
  - **Enable**: If enabled, logging details are sent to the logfile.
  - **Verbose**: If enabled, verbose logging is activated (increasing the details sent to the log).

8.2 Media Sequencer Redundancy

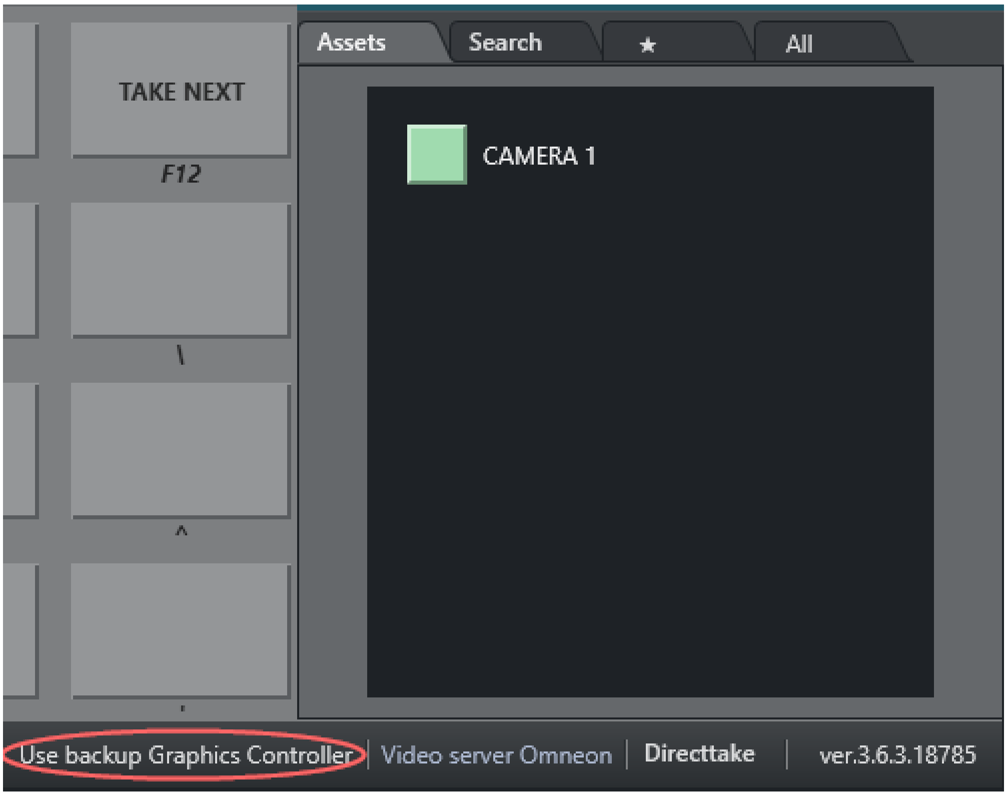
This section describes the operation of the main/backup switching of the Media Sequencer.

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 will momentarily go red, but will go back to green when the switching process is finished.

Still, the Media Sequencer will need some additional seconds to initialize all parts, so it is recommended to wait some time before continuing graphics operations. The exact delay will vary with graphics size and complexity and such.

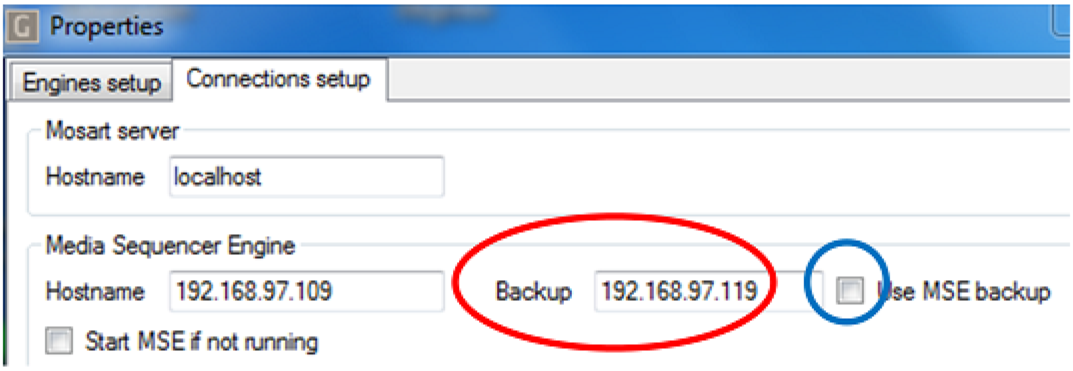
8.2.2 Configuration and Test

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:



Properties

Engines setup | Connections setup

Mosart server

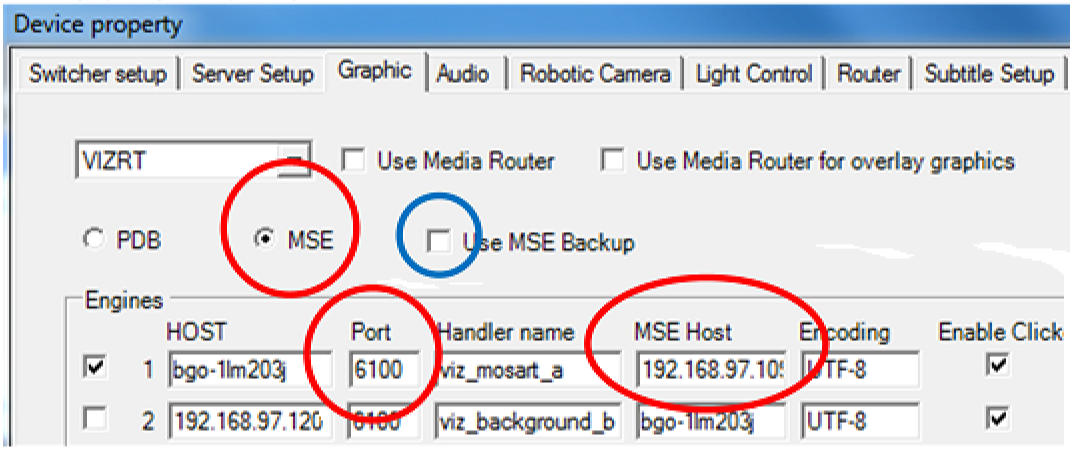
Hostname: localhost

Media Sequencer Engine

Hostname: 192.168.97.109   Backup: 192.168.97.119   □ Use MSE backup

□ Start MSE if not running

AVautomation:



Device property

Switcher setup | Server Setup | Graphic | Audio | Robotic Camera | Light Control | Router | Subtitle Setup

VIZRT   □ Use Media Router   □ Use Media Router for overlay graphics

○ PDB   ● MSE   □ Use MSE Backup

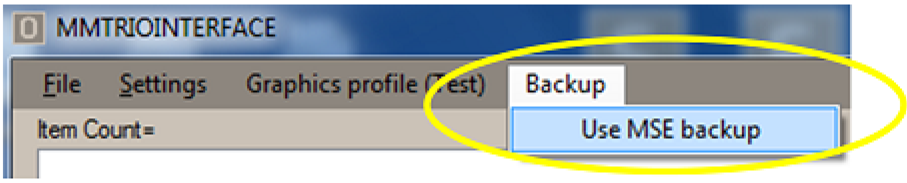
Engines

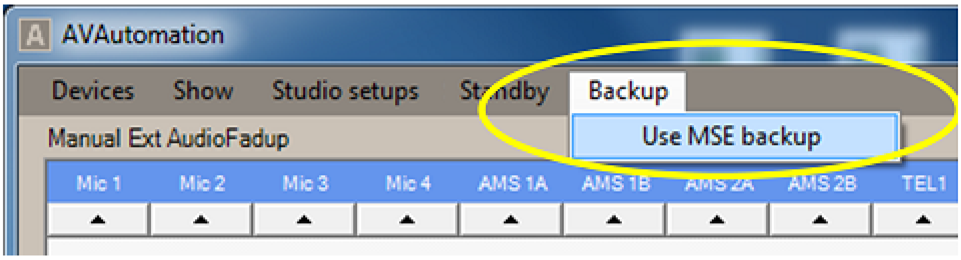
		HOST	Port	Handler name	MSE Host	Encoding	Enable Click
☑	1	bgo-1lm203j	6100	viz_mosart_a	192.168.97.10!	UTF-8	☑
□	2	192.168.97.120	6100	viz_background_b	bgo-1lm203j	UTF-8	☑

8.2.3 Test and Debug

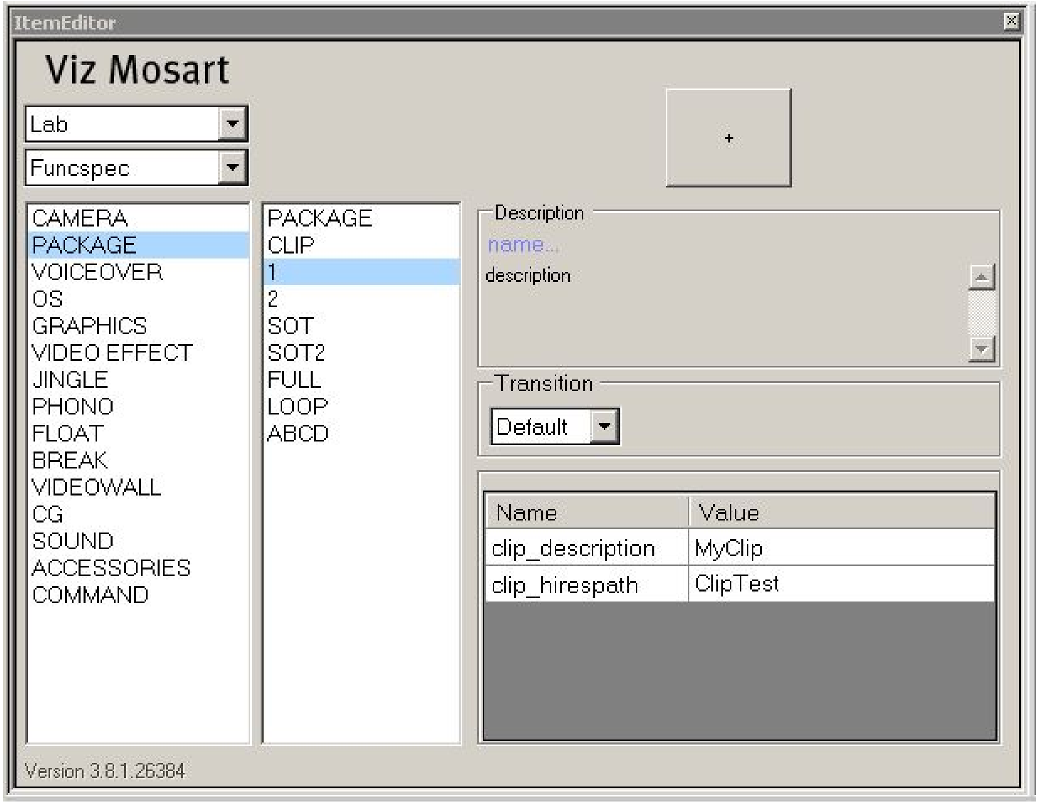
Both Trio Interface and AV Automation have a separate menu item that enables you to switch each of them between the main and backup Media Sequencer. This switching possibility is just for testing and debugging purposes. It is recommended to use the main switch in the GUI instead.

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





9 ActiveX Configuration



Lab

Funcspec

Transition: Default

Name	Value
clip_description	MyClip
clip_hirespath	ClipTest

Version 3.8.1.26384

The **ActiveX** is used to insert Viz Mosart template information into the Newsroom System script.

For the ActiveX to function correctly, some parameters are required to be set that link the ActiveX content to your Viz Mosart Server installation.

The ActiveX can be linked to the Viz Mosart Server either via the 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

Note: This section assumes knowledge of, and access to, the Windows Registry. Normally this means you must run with elevated (Administrator) privileges. Care should be taken when editing the Windows registry. It is recommended to either create a windows restore point (from the Start menu type "`create a restore point`" > Select `Create a restore point` > Choose `Create`) or create a backup of the registry in the **regedit.exe** application with **File > Export** before you change any registry keys.

To browse or change registry keys you can use the "**regedit.exe**" application: From the Windows Start menu type "*regedit*".

Registry settings for the Viz Mosart ActiveX on 32-bits machines are found under the registry key

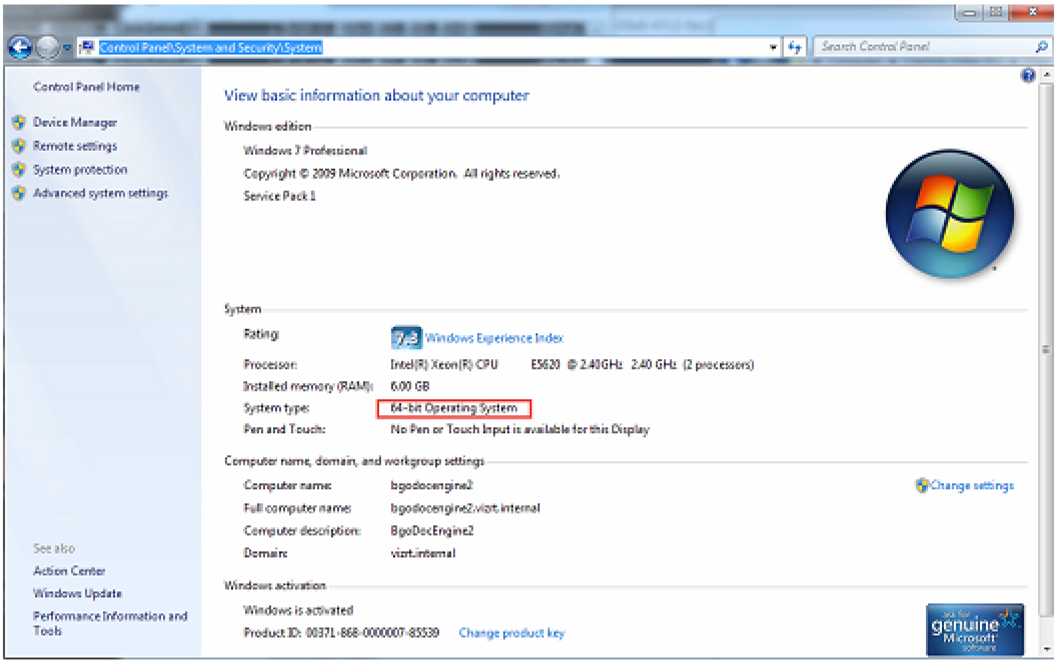
```
HKEY_LOCAL_MACHINE/SOFTWARE/Mosart Medialab/Mosart ActiveX
```

If the Viz Mosart ActiveX runs on an x64 (64-bit) based machine, the settings will be found under the registry key:

```
HKEY_LOCAL_MACHINE/SOFTWARE/Wow6432Node/Mosart Medialab/Mosart ActiveX
```

Most PC's today are 64-bit. To investigate the architecture:

- On Windows 10 you can view the architecture with *Start-menu > Settings > System > About this PC*
- On Windows 7 type PCs: *Start-menu > Control Panel > System and Security > System*



View basic information about your computer

Windows edition

Windows 7 Professional

Copyright © 2009 Microsoft Corporation. All rights reserved.

Service Pack 1

System

Rating: 7.3 Windows Experience Index

Processor: Intel(R) Xeon(R) CPU E5620 @ 2.40GHz 2.40 GHz (2 processors)

Installed memory (RAM): 6.00 GB

System type: 64-bit Operating System

Pen and Touch: No Pen or Touch Input is available for this Display

Computer name, domain, and workgroup settings

Computer name: bgodocengine2

Full computer name: bgodocengine2.vizrt.internal

Computer description: BgoDocEngine2

Domain: vizrt.internal

Windows activation

Windows is activated

Product ID: 00371-868-0000007-85539

If you prefer to get this information with a script or the command line, something like the following Powershell script can be used:

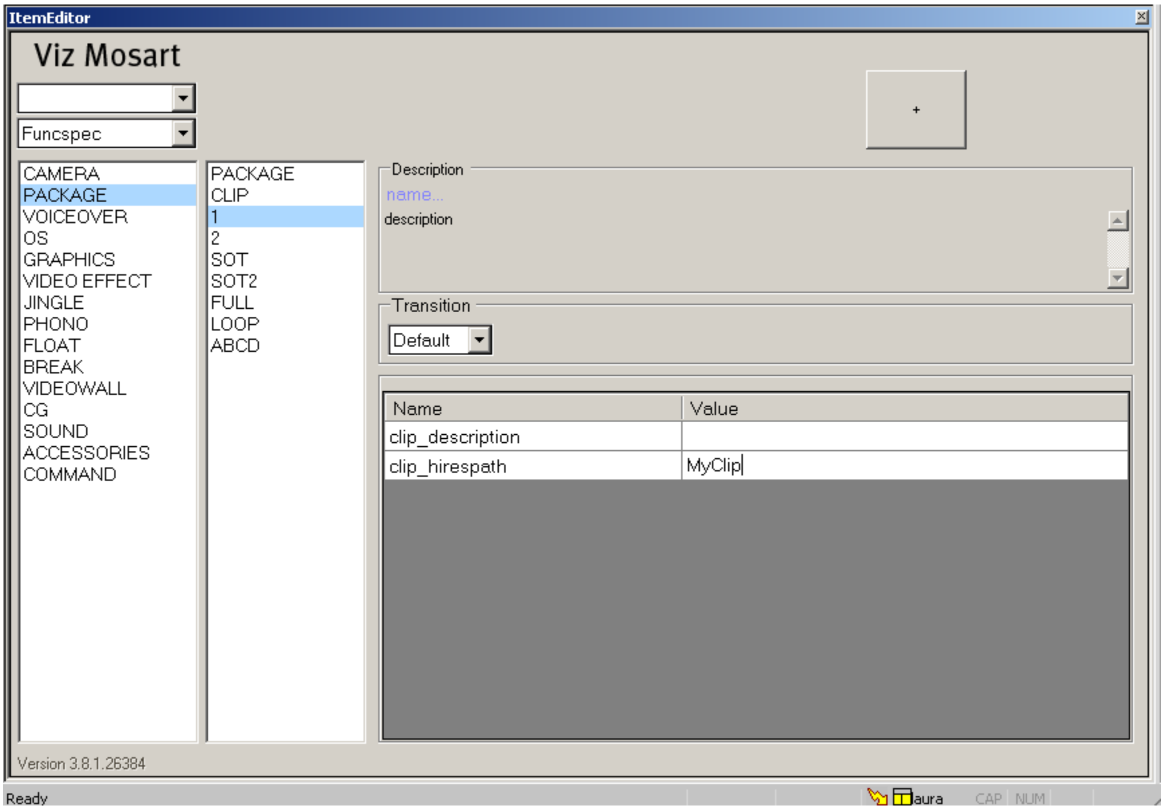
show architecture

```
if ((gwmi win32_operatingsystem | select osarchitecture).
osarchitecture -eq "64-bit")
{
    #64 bit logic here
    Write "This PC is running a 64-bit OS"
}
else
{
    #32 bit logic here
    Write "This PC is running a 32-bit OS"
}
```

9.1.1 Viz Mosart ActiveX Registry Properties

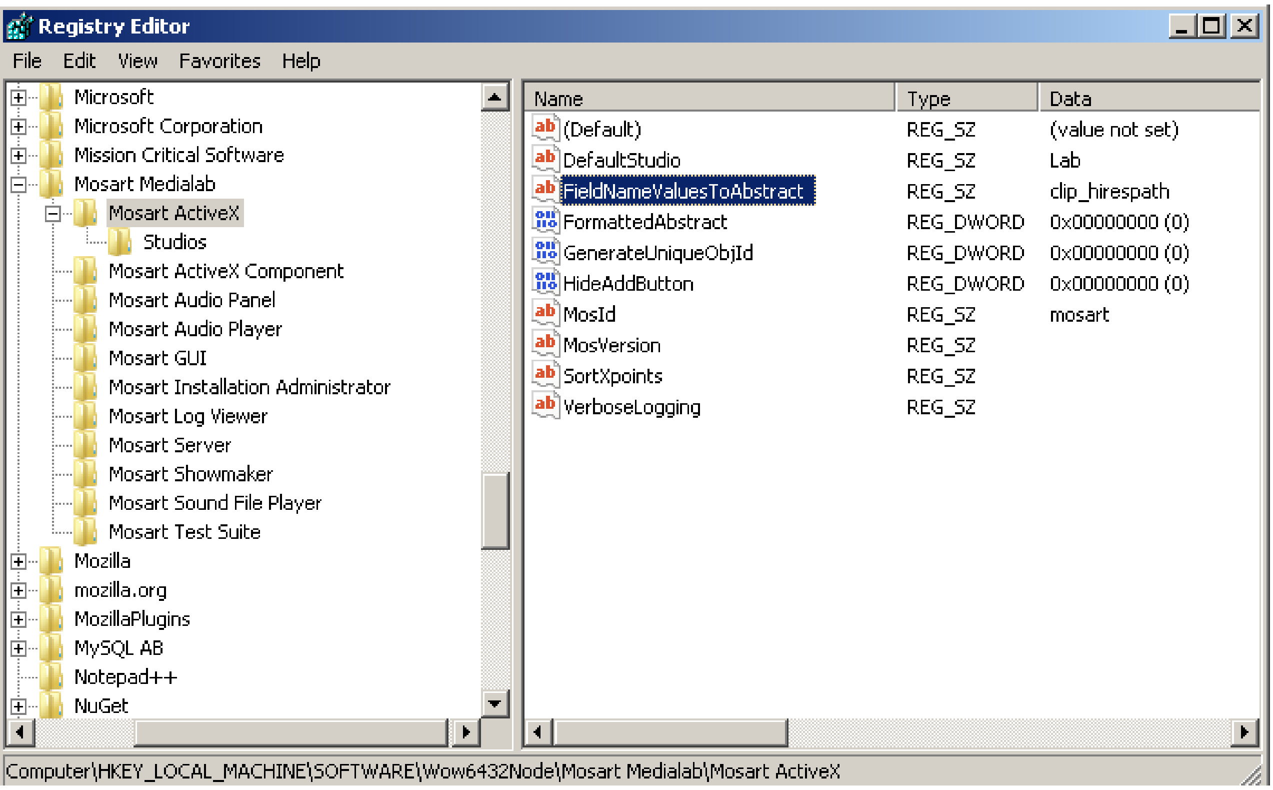
Mosart ActiveX key:

- **DefaultStudio**: *Optional*. The entry of the studio setup in the *Studios* sub-key. A blank value will select 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. Read more about newsroom tags in *Linking Device Properties and Newsroom Tags* section from Building Viz Mosart Templates.
  Example:
  Given the following template type=PACKAGE and variant=1 with two newsroom tags, clip_hirespath and clip_description:



Name	Value
clip_description	
clip_hirespath	MyClip

And *FieldNameValuesToAbstract* is set to *clip_hirespath* in the registry:



Microsoft
Microsoft Corporation
Mission Critical Software
Mosart Medialab
    Mosart ActiveX
        Studios
    Mosart ActiveX Component
    Mosart Audio Panel
    Mosart Audio Player
    Mosart GUI
    Mosart Installation Administrator
    Mosart Log Viewer
    Mosart Server
    Mosart Showmaker
    Mosart Sound File Player
    Mosart Test Suite
Mozilla
mozilla.org
MozillaPlugins
MySQL AB
Notepad++
NuGet

Name	Type	Data
(Default)	REG_SZ	(value not set)
DefaultStudio	REG_SZ	Lab
FieldNameValuesToAbstract	REG_SZ	clip_hirespath
FormattedAbstract	REG_DWORD	0x00000000 (0)
GenerateUniqueObjId	REG_DWORD	0x00000000 (0)
HideAddButton	REG_DWORD	0x00000000 (0)
MosId	REG_SZ	mosart
MosVersion	REG_SZ	
SortXpoints	REG_SZ	
VerboseLogging	REG_SZ	

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Mosart Medialab\Mosart ActiveX

Then *mosAbstract* tag will contain the clip id in the MOS object generated by Viz Mosart ActiveX:

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</mosItemBrowserProgI
D>
            <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="" />
```

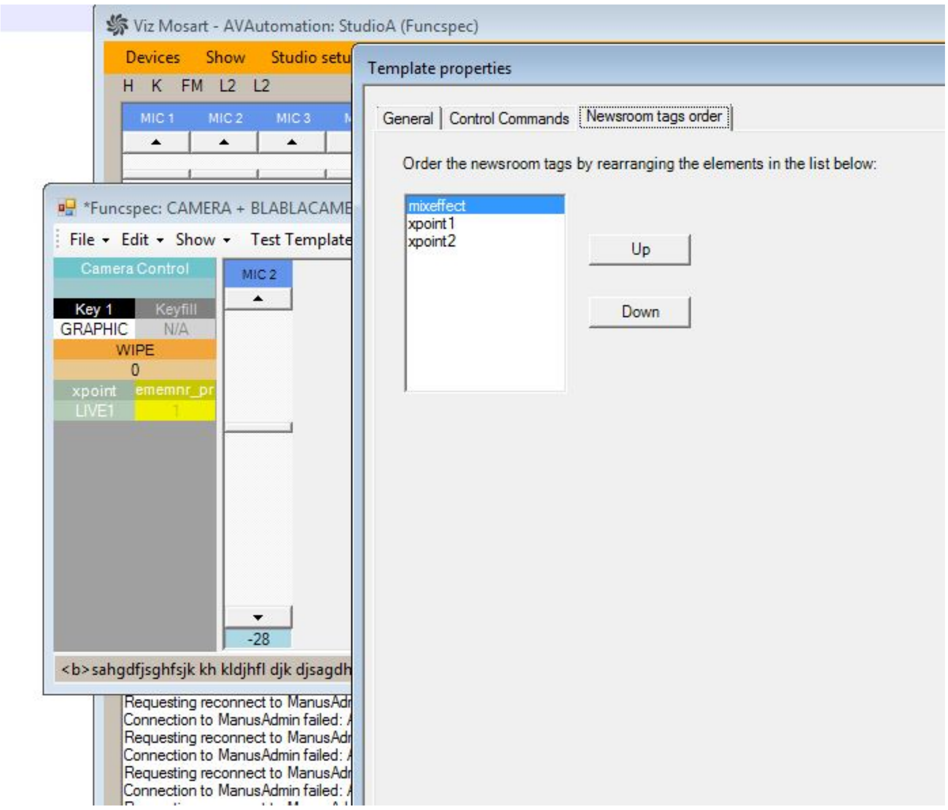
```
                                    </fields>
                                </variant>
                            </variants>
                        </type>
                    </mosarttemplate>
                </mosPayload>
            </mosExternalMetadata>
        </item>
    </ncsItem>
</mos>
```

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* will be built using the value of *clip_hirespath* and *clip_description* separated by space, i.e. <mosAbstract>MyClip Test</mosAbstract>.

If *FieldNameValuesToAbstract* is set to an invalid value (for example, a typo error in the newsroom tag name(s)), then *mosAbstract* tag will be set to <template type> <template variant> followed by the values of newsroom tags found on the template separated by space. Given the example above, then *mosAbstract* will be <mosAbstract>PACKAGE 1 MyClip Test</mosAbstract>.

- **FormattedAbstract**: *Optional*. Valid values: 0 or 1. Setting this value to 1 will insert 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.
- **GenerateUniqueObjId**: *Optional*. Valid values: 0 or 1. Enables (when set to 1) or disables (when set to 0 or blank) generation of an unique objID for a MOS object. Set this to 1 for OpenMedia which identifies a MOS object by objID and itemID.
- **HideAddButton**: *Optional*. Valid values: 0 or 1. When MOS-based NCSs are used, set this setting to 1 for hiding the "Add" button in the Viz Mosart ActiveX and force the user to use Drag&Drop. Default is 0 (false).
- **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**: *Optional*. Setting this value will force the Viz Mosart ActiveX to use the given version of the MOS protocol. A version higher than 2.8 will enable 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. Use *HideAddButton* option in order to force using only Drag&Drop functionality.
- **SortXpoints**: *Optional*. Specifies the ordering of the newsroom tags selection in the Viz Mosart ActiveX. By default, Viz Mosart ActiveX sorts the newsroom tags alphabetically (both in database and channeltemplates.xml). When enabled (value 1), the newsroom tags will be sorted based on the

order that was given in the `channeltemplates.xml` when they were created. Read more about newsroom tags in Additional Template Functionality, section Newsroom Tags Order.



<b>sahgdfjsghfsjk kh kldjhfl djk djsagdh

Order the newsroom tags by rearranging the elements in the list below:

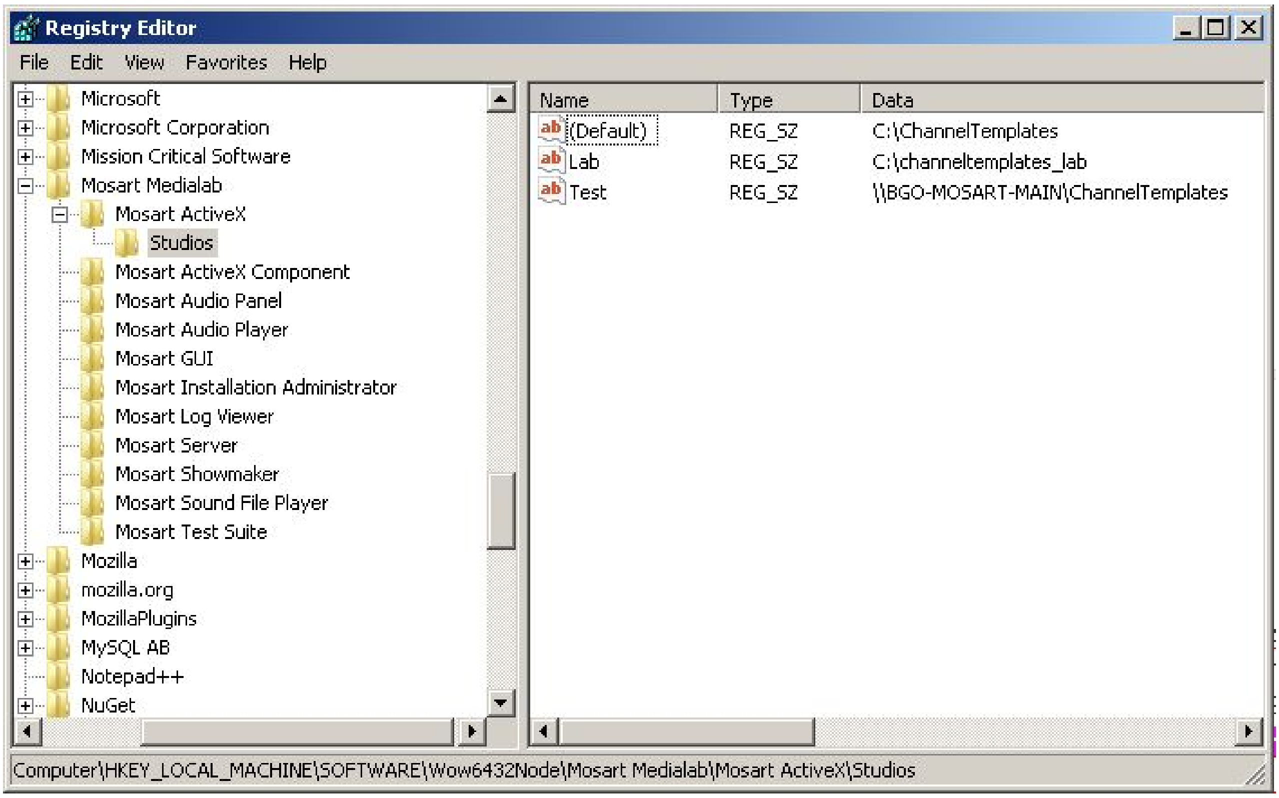
mixeffect
xpoint1
xpoint2

```
Requesting reconnect to ManusAdr
Connection to ManusAdmin failed: A
Requesting reconnect to ManusAdr
Connection to ManusAdmin failed: A
Requesting reconnect to ManusAdr
Connection to ManusAdmin failed: A
```

- **VerboseLogging**: *Optional*. Valid values: 0 or 1. Viz Mosart ActiveX logs using Microsoft trace mechanisms. The logging will not be 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* will be logged. Setting *VerboseLogging* to 1 will enable more elaborate logging.

Studios sub-key

Under **Mosart ActiveX** key a sub-key *Studios* can be found where the system can be configured with multiple control rooms. A control room is defined by files residing in a specific folder (typically *C:/channeltemplates*) or in a database. See the Registry Editor screenshot below . For information on connecting Viz Mosart ActiveX to a control room, please refer to Connecting the ActiveX to Viz Mosart Server.

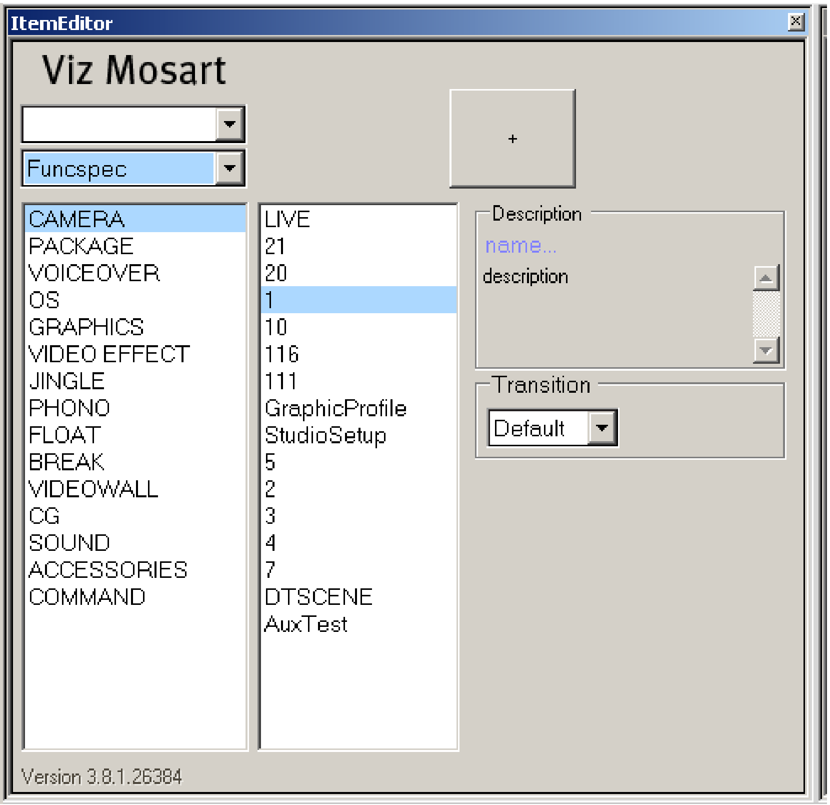


Name	Type	Data
(Default)	REG_SZ	C:\ChannelTemplates
Lab	REG_SZ	C:\channeltemplates_lab
Test	REG_SZ	\\BGO-MOSART-MAIN\ChannelTemplates

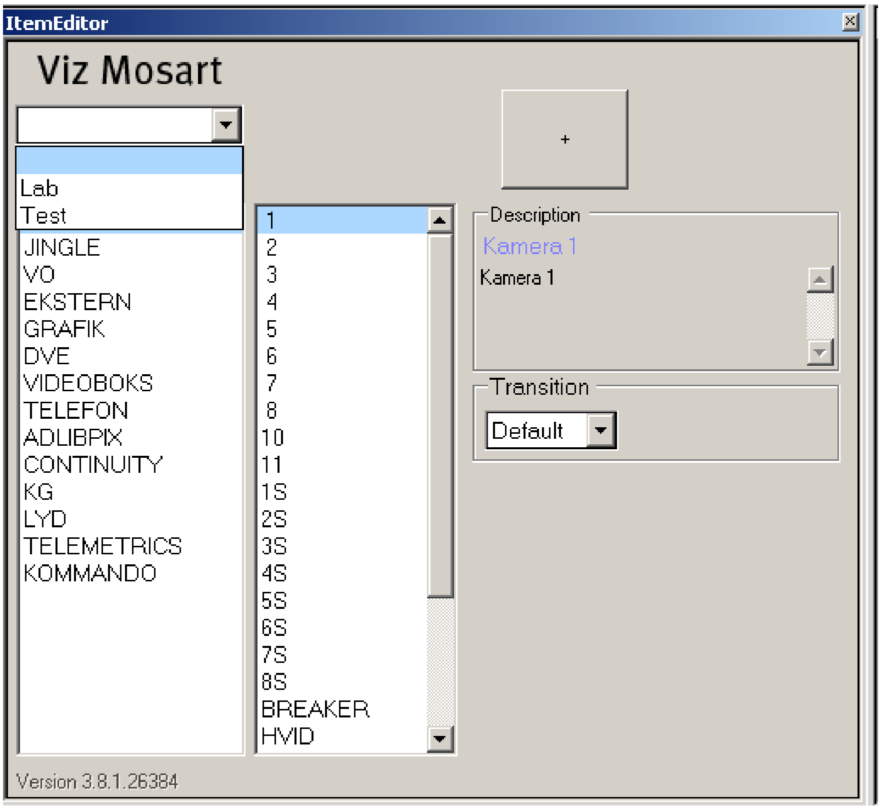
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Mosart Medialab\Mosart ActiveX\Studios

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:



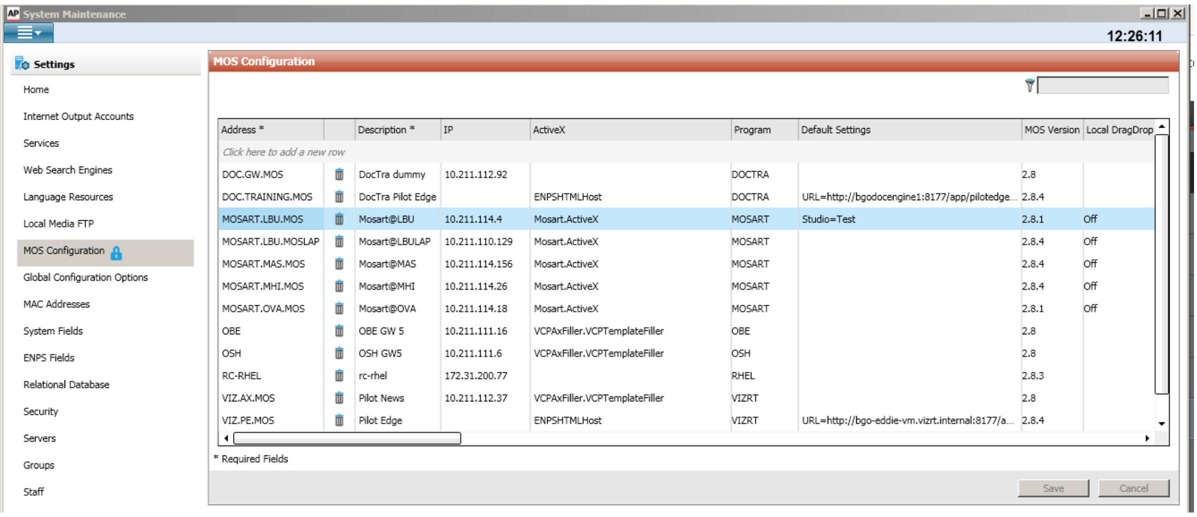
In our example, three control rooms were configured in Studios sub-key: **blank item** (default control room), **Lab** and **Test**. These items all be shown in the first drop-down box. Since **DefaultStudio** registry key was set to blank, Viz Mosart ActiveX will open with blank item selected in the first drop-down box and thus pointing to the control room whose configuration files can be found in Viz Mosart folder "C:\ChannelTemplates":



9.1.2 Configuring AP ENPS

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 in order to be able to use Viz Mosart ActiveX in ENPS client:



MOS Configuration

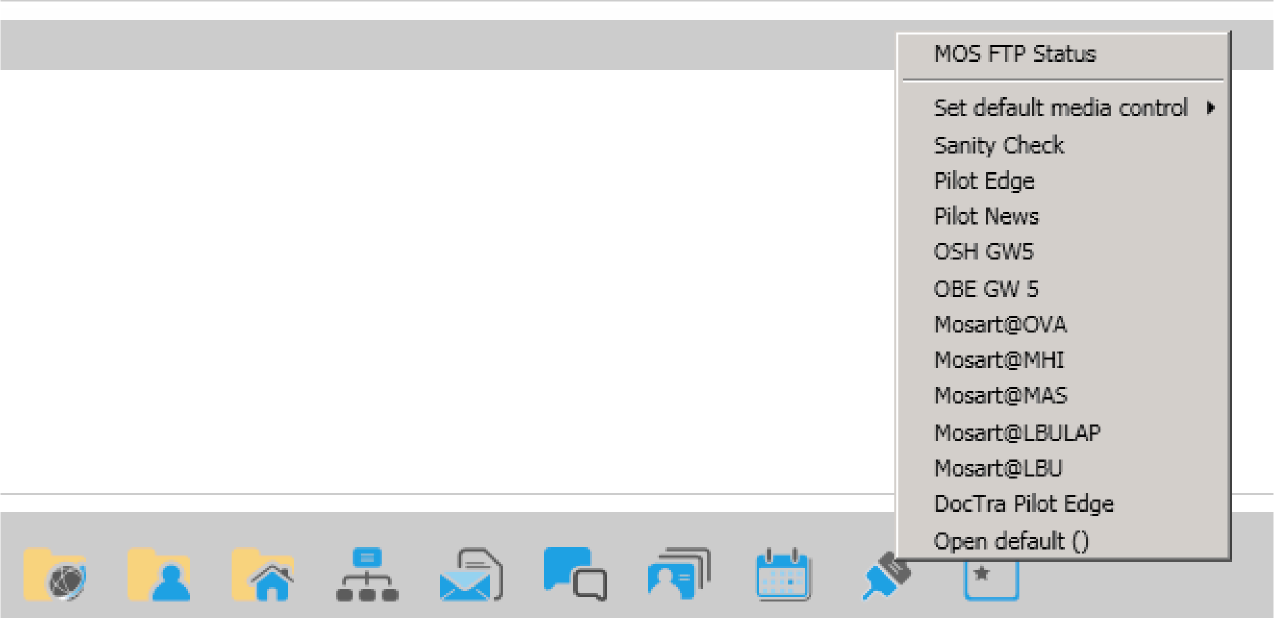
Address *		Description *	IP	ActiveX	Program	Default Settings	MOS Version	Local DragDrop
Click here to add a new row								
DOC.GW.MOS	🗑	DocTra dummy	10.211.112.92		DOCTRA		2.8	
DOC.TRAINING.MOS	🗑	DocTra Pilot Edge		ENPSHTMLHost	DOCTRA	URL=http://bgodocengine1:8177/app/pilotedge...	2.8.4	
MOSART.LBU.MOS	🗑	Mosart@LBU	10.211.114.4	Mosart.ActiveX	MOSART	Studio=Test	2.8.1	Off
MOSART.LBU.MOSLAP	🗑	Mosart@LBULAP	10.211.110.129	Mosart.ActiveX	MOSART		2.8.4	Off
MOSART.MAS.MOS	🗑	Mosart@MAS	10.211.114.156	Mosart.ActiveX	MOSART		2.8.4	Off
MOSART.MHI.MOS	🗑	Mosart@MHI	10.211.114.26	Mosart.ActiveX	MOSART		2.8.4	Off
MOSART.OVA.MOS	🗑	Mosart@OVA	10.211.114.18	Mosart.ActiveX	MOSART		2.8.1	Off
OBE	🗑	OBE GW 5	10.211.111.16	VCPAxFiller.VCPTemplateFiller	OBE		2.8	
OSH	🗑	OSH GW5	10.211.111.6	VCPAxFiller.VCPTemplateFiller	OSH		2.8	
RC-RHEL	🗑	rc-rhel	172.31.200.77		RHEL		2.8.3	
VIZ.AX.MOS	🗑	Pilot News	10.211.112.37	VCPAxFiller.VCPTemplateFiller	VIZRT		2.8	
VIZ.PE.MOS	🗑	Pilot Edge		ENPSHTMLHost	VIZRT	URL=http://bgo-eddie-vm.vizrt.internal:8177/a...	2.8.4	

* Required Fields

Column name	**Value**
Description	Gives the Mosart ActiveX name in ENPS client
ActiveX	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 **DefaultStudion** 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 will open with the given studio name selected in the first drop-down box (in our example will be "Test"). Note that **DefaultStudion** setting from registry will be ignored.
Local DragDrop	Off

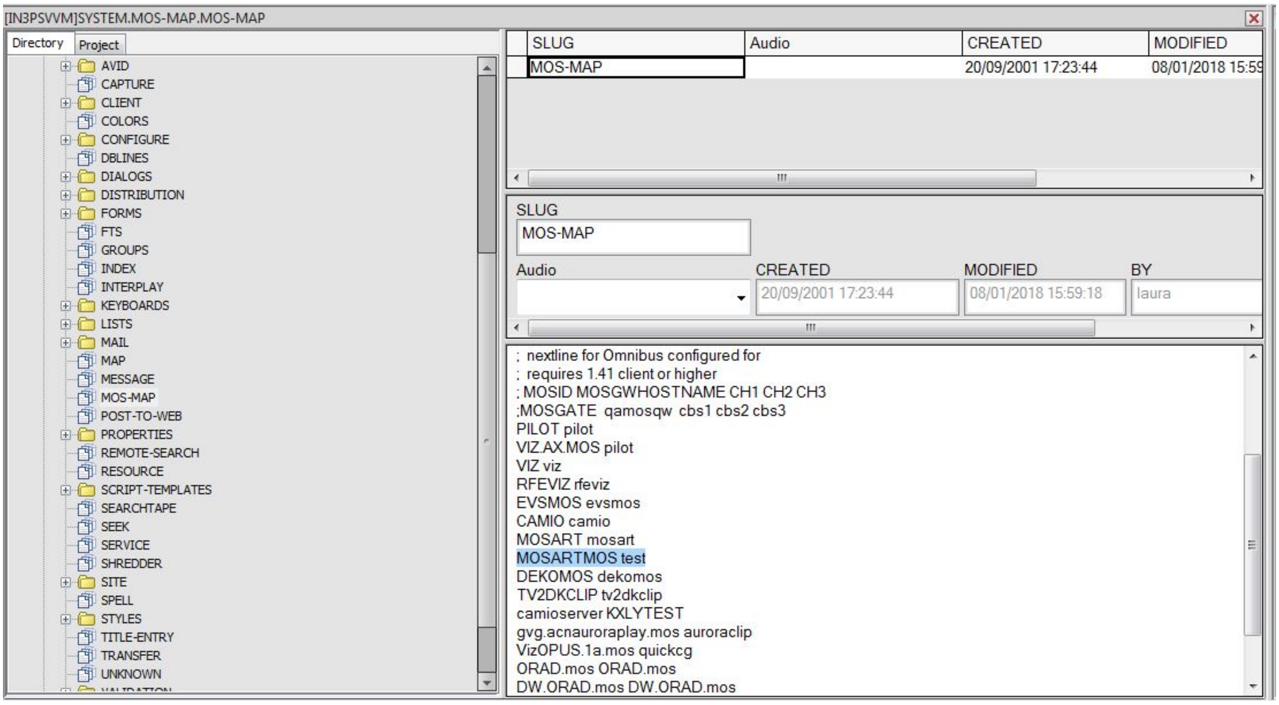
In the registry for Viz Mosart ActiveX, there is no need to provide a value for **MosId** because the value of **Address** column in the *MOS Configuration* from *ENPS System Maintenance* will be used, but it is important to have a value for **MosVersion**.

To open Mosart ActiveX from ENPS client, right click on the MOS icon from bottom right and select the Mosart ActiveX given by the name configured in MOS Configuration (in our example *Mosart@LBU*). If several Viz Mosart MOS devices are configured, any of them would work to open Viz Mosart ActiveX.



9.1.3 Configuring Avid iNEWS

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 (one must have administrator privileges in order to access the SYSTEM.MOS-MAP rundown):



SLUG	Audio	CREATED	MODIFIED
MOS-MAP		20/09/2001 17:23:44	08/01/2018 15:59

SLUG: MOS-MAP

Audio:

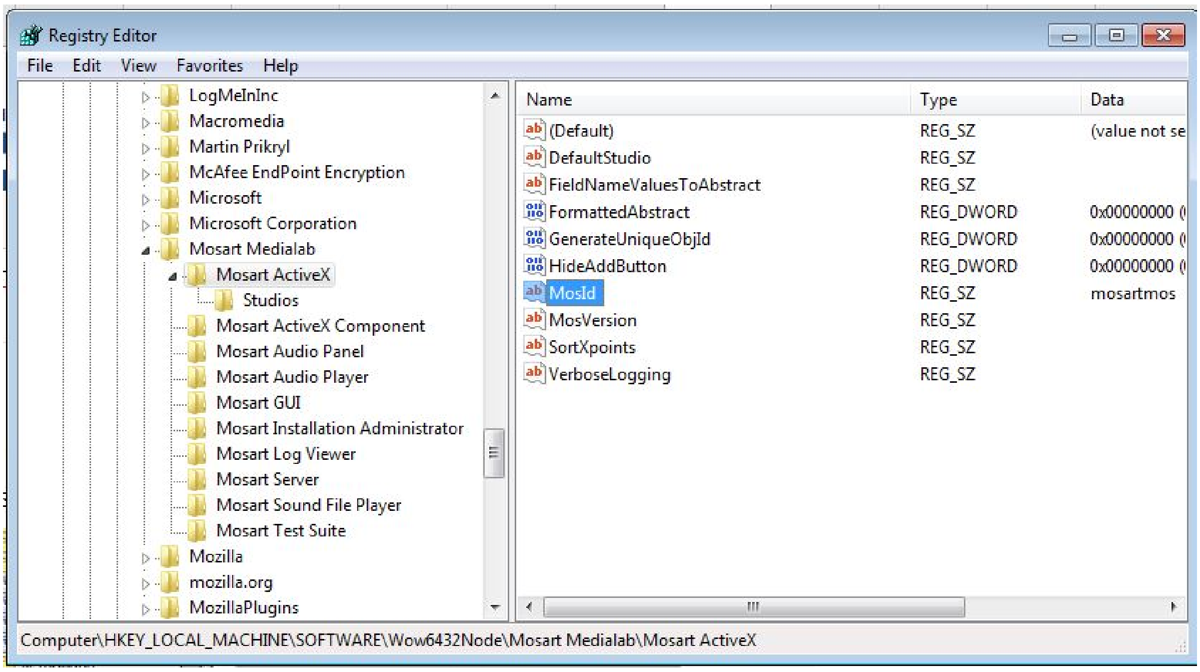
CREATED: 20/09/2001 17:23:44

MODIFIED: 08/01/2018 15:59:18

BY: laura

```
; nextline for Omnibus configured for
; requires 1.41 client or higher
; MOSID MOSGWHOSTNAME CH1 CH2 CH3
;MOSGATE  qamosqw  cbs1 cbs2 cbs3
PILOT pilot
VIZ.AX.MOS pilot
VIZ viz
RFEVIZ rfeviz
EVSMOS evsmos
CAMIO camio
MOSART mosart
MOSARTMOS test
DEKOMOS dekomos
TV2DKCLIP tv2dkclip
camioserver KXLYTEST
gvg.acnauroraplay.mos auroraclip
VizOPUS.1a.mos quickcg
ORAD.mos ORAD.mos
DW.ORAD.mos DW.ORAD.mos
```

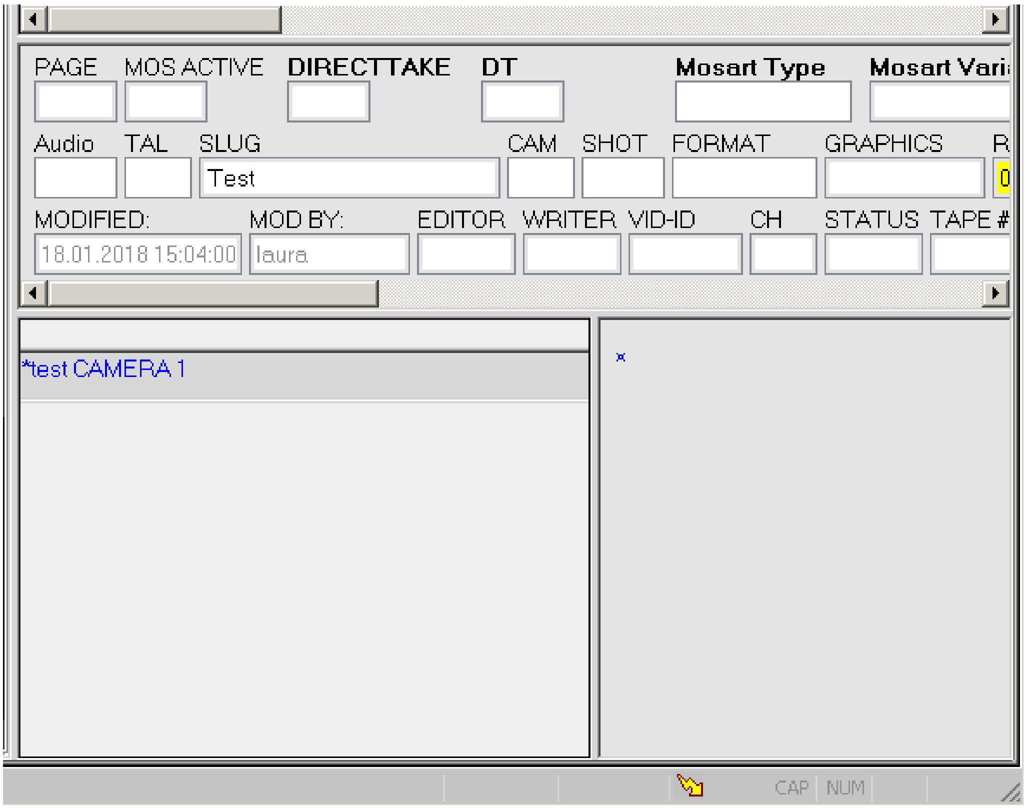
Thus the first value (MOSARTMOS) is the MOS Id that will be configured in the Viz Mosart ActiveX registry. The value is case-insensitive.



Name	Type	Data
(Default)	REG_SZ	(value not se
DefaultStudio	REG_SZ	
FieldNameValuesToAbstract	REG_SZ	
FormattedAbstract	REG_DWORD	0x00000000 (
GenerateUniqueObjId	REG_DWORD	0x00000000 (
HideAddButton	REG_DWORD	0x00000000 (
MosId	REG_SZ	mosartmos
MosVersion	REG_SZ	
SortXpoints	REG_SZ	
VerboseLogging	REG_SZ	

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Mosart Medialab\Mosart ActiveX

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:



SLUG: Test

MODIFIED: 18.01.2018 15:04:00

MOD BY: laura

*test CAMERA 1

To open Viz Mosart ActiveX from iNews client, got to *Tools/Plugins/Mosart Item Editor*.

9.2 Connecting the ActiveX to Viz Mosart Server

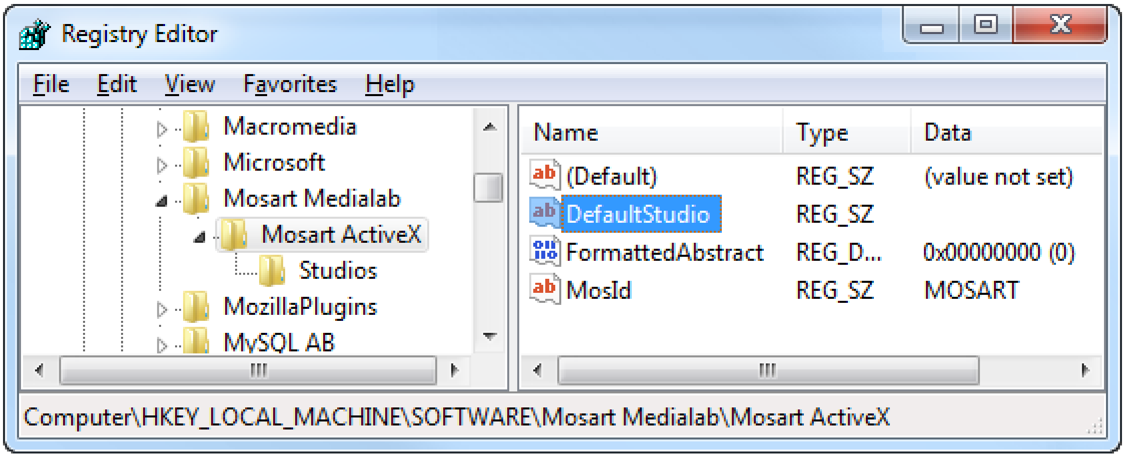
There are two ways to link the ActiveX to your Viz Mosart Server installation:

- 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.

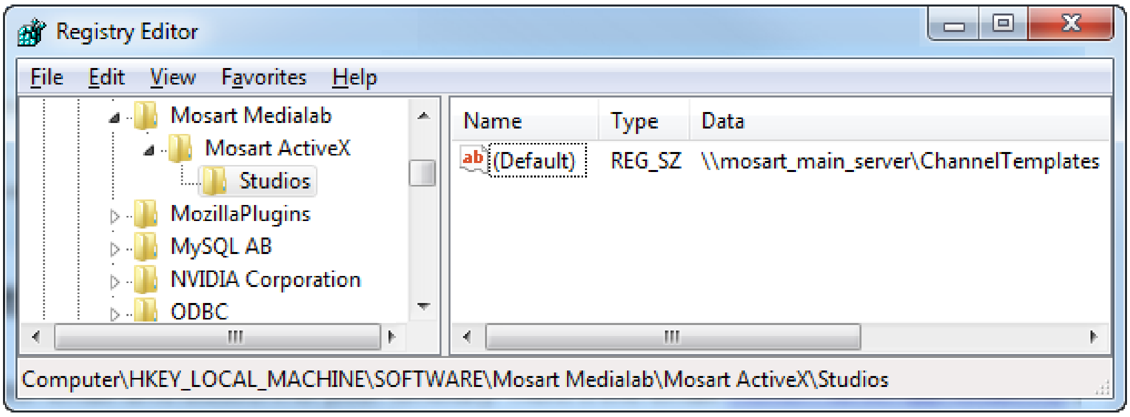
1. Enable folder sharing of C:\ChannelTemplates on the Main Viz Mosart Server.
   - Give read only access to this folder to the appropriate users.
2. Configure the DefaultStudio registry value to be blank.



Name	Type	Data
(Default)	REG_SZ	(value not set)
DefaultStudio	REG_SZ	
FormattedAbstract	REG_D...	0x00000000 (0)
MosId	REG_SZ	MOSART

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Mosart Medialab\Mosart ActiveX

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`



Name	Type	Data
(Default)	REG_SZ	\\mosart_main_server\ChannelTemplates

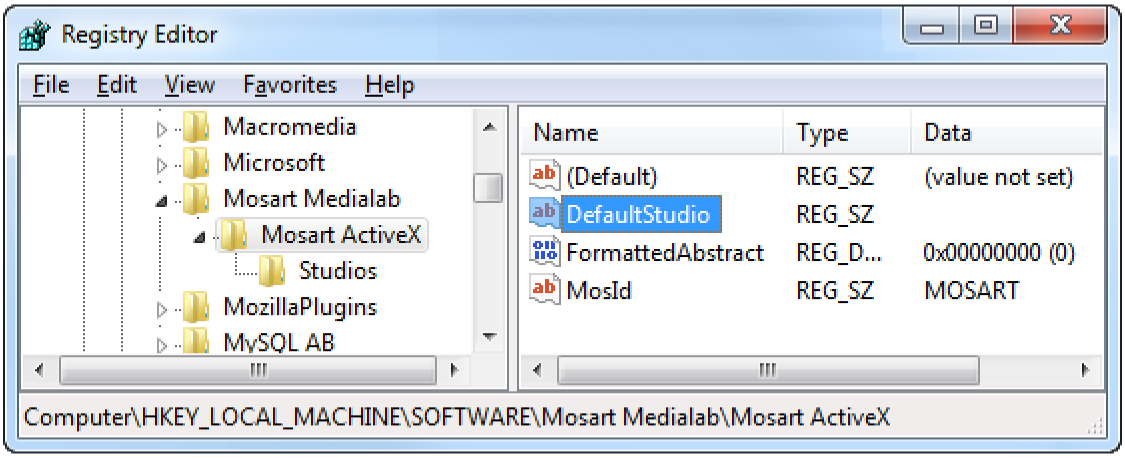
Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Mosart Medialab\Mosart ActiveX\Studios

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.

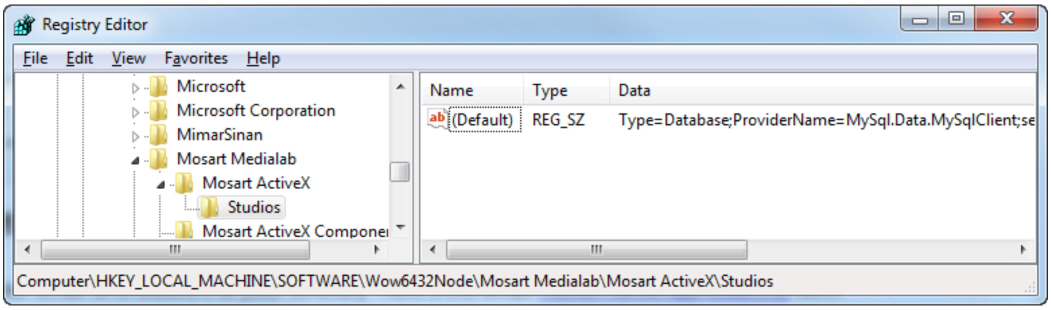
1. A read only database user is recommended for access from the ActiveX.
2. Configure the DefaultStudio registry value to be blank.



Name	Type	Data
(Default)	REG_SZ	(value not set)
DefaultStudio	REG_SZ	
FormattedAbstract	REG_D...	0x00000000 (0)
MosId	REG_SZ	MOSART

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Mosart Medialab\Mosart ActiveX

1. 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`



Name	Type	Data
(Default)	REG_SZ	Type=Database;ProviderName=MySql.Data.MySqlClient;se

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Mosart Medialab\Mosart ActiveX\Studios

The connection string of the database contains the following fields:

Name	Value
Type	Database
ProviderName	The qualified name for the .NET Component used for DB 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.
Database	The SQL database name on the database server.
Gallery	Optional. Name of gallery to be addressed within the database. Will also extract 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 will store 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 will only be downloaded if the timestamps on these files differ.

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

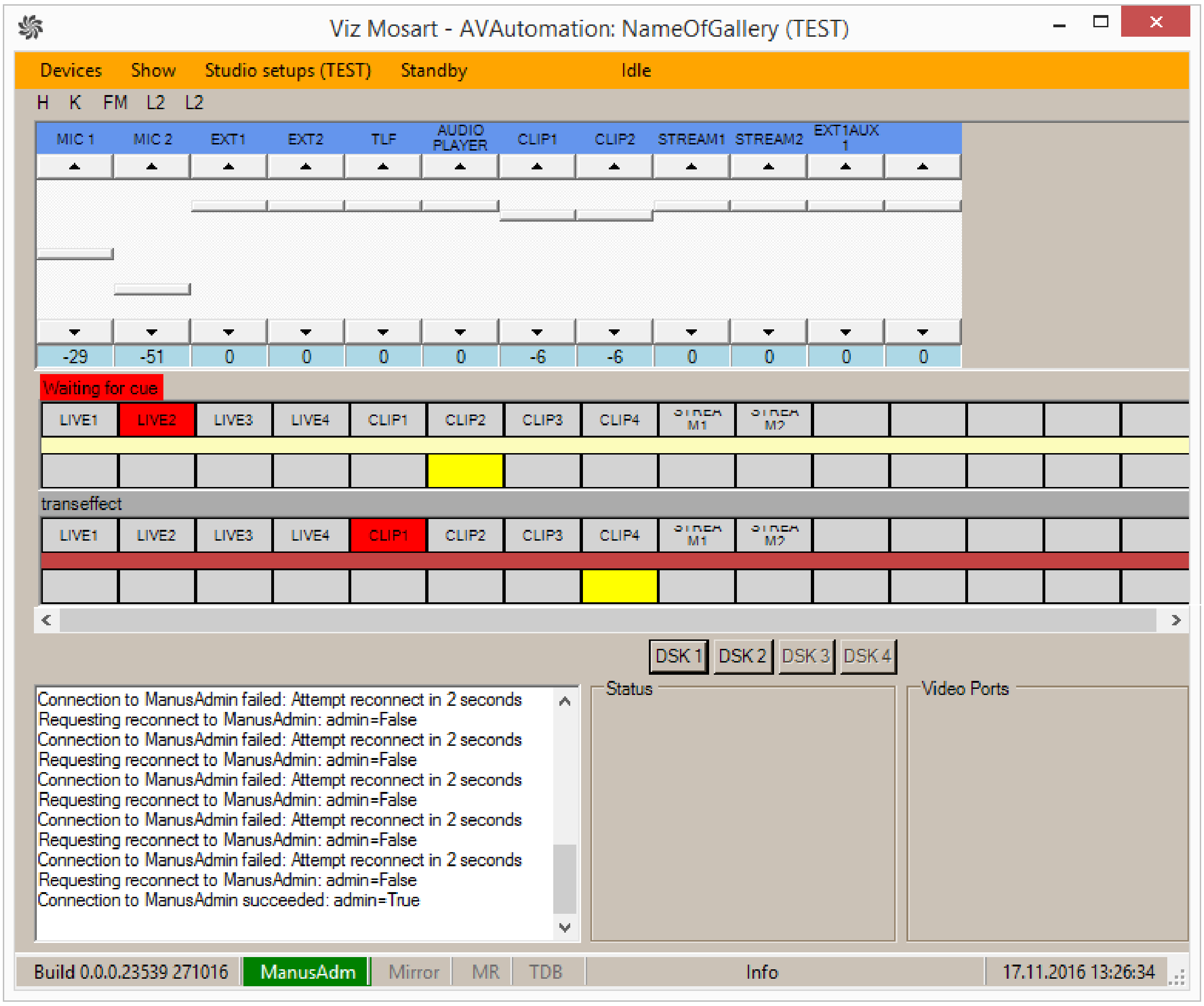
9.3 ActiveX Notes

The ActiveX will store 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 the ActiveX can be done by opening and closing the ActiveX twice within the host application.

10 AV Automation



```
Connection to ManusAdmin failed: Attempt reconnect in 2 seconds
Requesting reconnect to ManusAdmin: admin=False
Connection to ManusAdmin failed: Attempt reconnect in 2 seconds
Requesting reconnect to ManusAdmin: admin=False
Connection to ManusAdmin failed: Attempt reconnect in 2 seconds
Requesting reconnect to ManusAdmin: admin=False
Connection to ManusAdmin failed: Attempt reconnect in 2 seconds
Requesting reconnect to ManusAdmin: admin=False
Connection to ManusAdmin failed: Attempt reconnect in 2 seconds
Requesting reconnect to ManusAdmin: admin=False
Connection to ManusAdmin succeeded: admin=True
```

AV Automation (*MMAVAutomation*) is where all Viz Mosart connected broadcast devices (except Overlay Graphics engines) are controlled. 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 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.

This section contains the following topics:

- Using AV Automation
- Audio and Vision Mixer Setup
- Template Editor Password

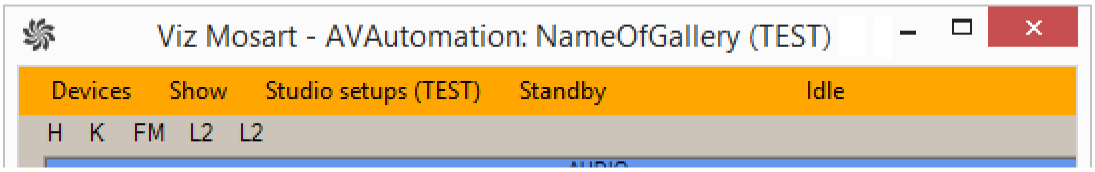
See Also

- AV Automation Device Properties

10.1 Using AV Automation

AV Automation is the device handler for Viz Mosart. This application communicates with all the different devices such as cameras, video servers, vision mixers and also has the tools to configure these devices. The template editor, where you can define and customize the templates, can also be found in this application.

10.1.1 AV Automation Main Menu



- **Devices**:
  - **Preferences**: Open AV Automation Device Properties
  - **Template editor**: Open Template Editor
  - **A/V Setup**: Open Audio and Vision Mixer Setup
  - **Show recording**: Open a dialog to start and stop recordings
  - **Change password**: Create or change the Template Editor Password
- **Show**:
  - **Show MVCP Control**: Deprecated
  - **Instrumentation Panel**: Open the Mosart Instrumentation Panel
- **Studio setups**: Select the template set to use for the current show
- **Standby**: Lists all configured devices and their status
- **Idle**: Enabled if AV Automation is in idle mode

10.2 Audio and Vision Mixer Setup

In Audio and Vision Mixer Setup the user can edit the Audio and Vision Mixer channels and configure their attributes.

Viz Mosart relies on predefined Video **Crosspoints** entered into AV Automation on initial setup and install of Viz Mosart.

The A/V Setup XML editor is used to configure the settings in AV Automation.

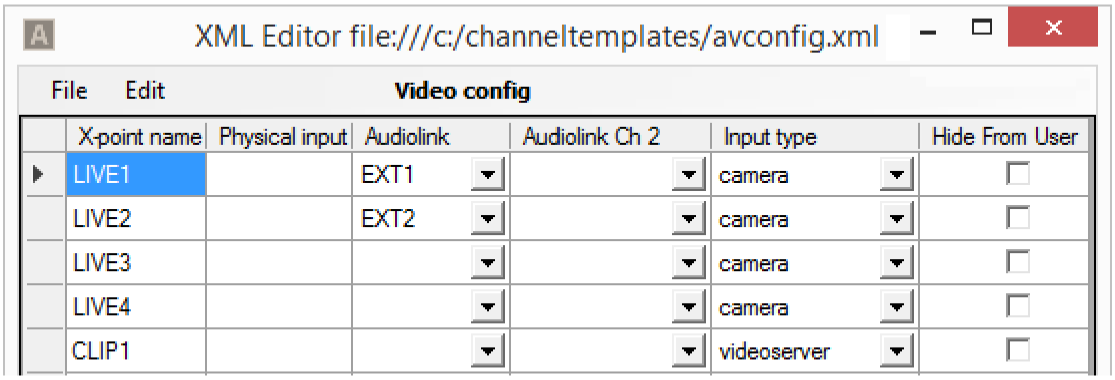
To open, go to *Devices* > *A/V Setup*.

10.2.1 A/V Setup Main Menu

- **File**:
  - Open: to open
  - Save: to save
  - Config: to configure
- **Edit**:

- Video Config
- Audio Config
- Effects: Audio and Vision Mixer Effects Setup
- Router Sources
- Router Destinations

Video Config



XML Editor file:///c:/channeltemplates/avconfig.xml

Video config

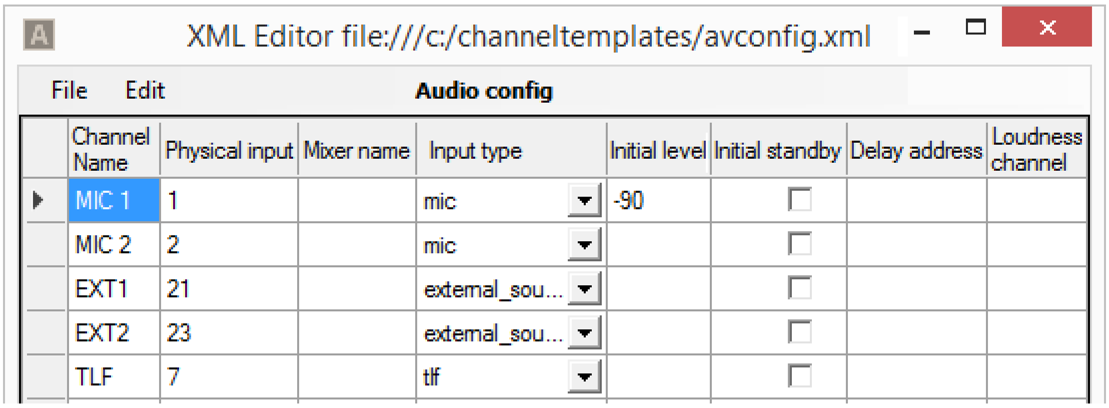
X-point name	Physical input	Audiolink	Audiolink Ch 2	Input type	Hide From User
LIVE1		EXT1		camera	□
LIVE2		EXT2		camera	□
LIVE3				camera	□
LIVE4				camera	□
CLIP1				videoserver	□

To enter the Vision Mixer configuration settings, open the Audio and Vision Mixer Setup XML editor and select *Edit* > *Video Config*. Enter values as needed.

As a minimum, you should enter the X-point name, physical input, and input type (where available).

- **X-point name**: Name to use for the video cross point 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 video source is changed in the newsroom system or Viz Mosart GUI, so the audio faders will follow video crosspoints.
- **Audiolink Ch 2**: Second audio fader link for the video source. The fader name to use when enabling "videoserver ch2" in the template editor. When video source is changed in the newsroom system or Viz Mosart GUI, the audio faders will follow video crosspoints. Should be used for audio mixers with mono-faders which cannot be coupled in the mixer.
- **Input type**:
  - camera: Select to assign this crosspoint as a camera type. This is used by the Audio Panel application.
  - external_source: Select to assign this crosspoint as an external source. This is used by the Audio Panel application.
  - videoserver: Select to assign this crosspoint as a server channel.
  - GRAPHIC_1-5: Select to assign this crosspoint as a graphics engine.
  - ROUTER_RIPPLE_A: Select to use as router source A.
  - ROUTER_RIPPLE_B: Select to use as router source B.
- **Hide from user**: Do not show crosspoint in AV Automation. This will hide the crosspoint from displaying in the crosspoint drop-down menu in templates and the ActiveX application.
- **Comments**: Only for comments on this page. Not visible or used anywhere else in the application.

10.2.2 Audio Config



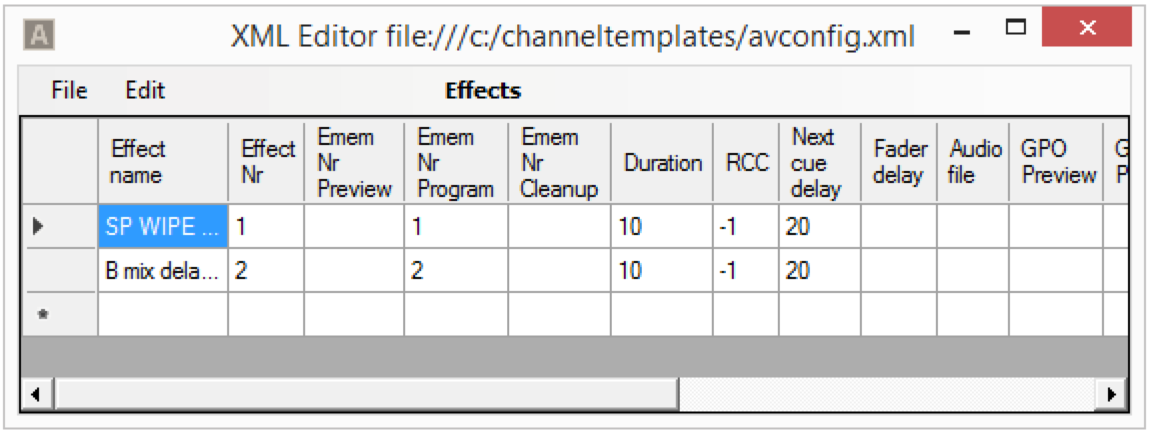
Channel Name	Physical input	Mixer name	Input type	Initial level	Initial standby	Delay address	Loudness channel
MIC 1	1		mic	-90	□		
MIC 2	2		mic		□		
EXT1	21		external_sou...		□		
EXT2	23		external_sou...		□		
TLF	7		tlf		□		

To enter the Audio Mixer configuration settings, open the Audio and Vision Mixer Setup XML editor and select *Edit > Audio Config*. Enter values as needed.

Channel name and physical input should always be entered as a minimum.

- **Channel Name**: Name to use for the audio fader in the Viz Mosart system.
- **Physical input**: For audio mixers that uses fader numbers or MIDI assigned faders, this value must correspond with the fader number on the audio desk.
- **Mixer name**: A fader unique identifier, for audio desks that uses 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**: Enter the desired level for the audio fader to obtain after a reload of the Viz Mosart rundown.
- **Initial standby**: Check the box for the faders that should not be reset at start-up of the AV Automation application.
- **Delay address**: This is to give the signal address for the delay function (only used by various Lawo audio mixer boards).
- **Loudness channel**: Defines the channel number for loudness level control.

10.2.3 Audio and Vision Mixer Effect Setups



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				

Video Mixer effects can also be predefined so the user does not have to recreate the effect in every Template that calls upon it. The Effects table is referenced during execution of a template that has specified an Effect Name or Number.

To **edit** the Vision Mixer Effects configuration, open the Audio and Vision Mixer Setup XML editor and select *Edit* > *Effects*. Enter values as needed.

Effect Number and Studio Setup

- The combination of *Effect Nr* and *Studio Setup* must be unique within the Effects table. That is, 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 which do not have a dedicated effect specified.
  Searching for an effect is done in the following order:

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

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.

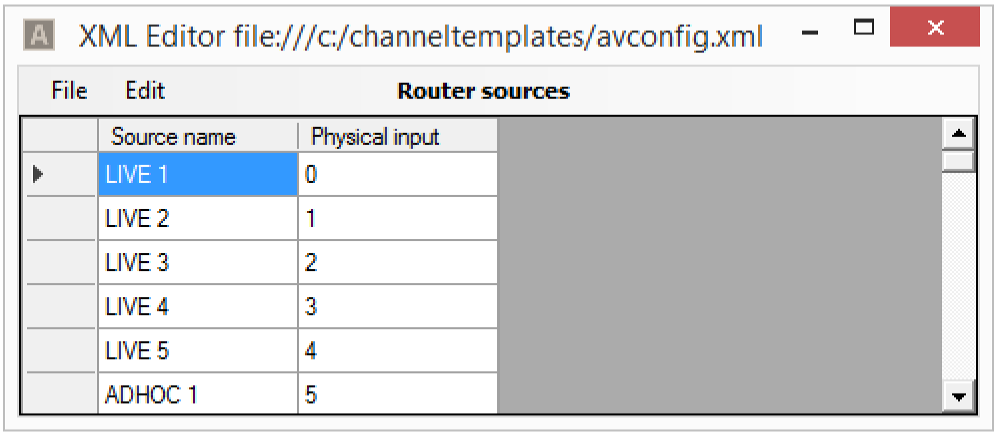
Settings

- **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 template is taken in preview. Mandatory.
- **EMEM Nr Program**: Video mixer preset to recall when template is taken on program. Mandatory.
- **EMEM Nr Cleanup**: This register will be recalled when cuing 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. Will hold the trigging 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 cuing in 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.
- **Fader Delay**: Delay opening the audio effect fader (the Audio Player fader).
- **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**: Macro number to run when entering preview mode (Sony only).

- **Directtake**: Number of directtake templates triggered when performing an Effect transition. Often used to allow control of additional devices.
  For example: A directtake triggering a named overlay graphic, or a directtake with additional commands required by the vision mixer for this effect.
- **Studio Setup**: Specifies the studio setup for which the effect will be run. This attribute is optional. Used in conjunction with *Effect Nr* to uniquely identify effects.
- **Macro Preview Action**:
  - Recall: Macro is prepared (or cued).
  - Take: Macro is taken.
  - Recall and Take: Macro is prepared and taken.
- **Macro Program**: Macro to run when entering program mode (Sony only).
- **Macro Program Action**:
  - Recall: Macro is prepared (or cued).
  - Take: Macro is taken.
  - Recall and Take: Macro is prepared and taken.

10.2.4 Router Sources

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 studio, or anything that needs switching as part of your production.



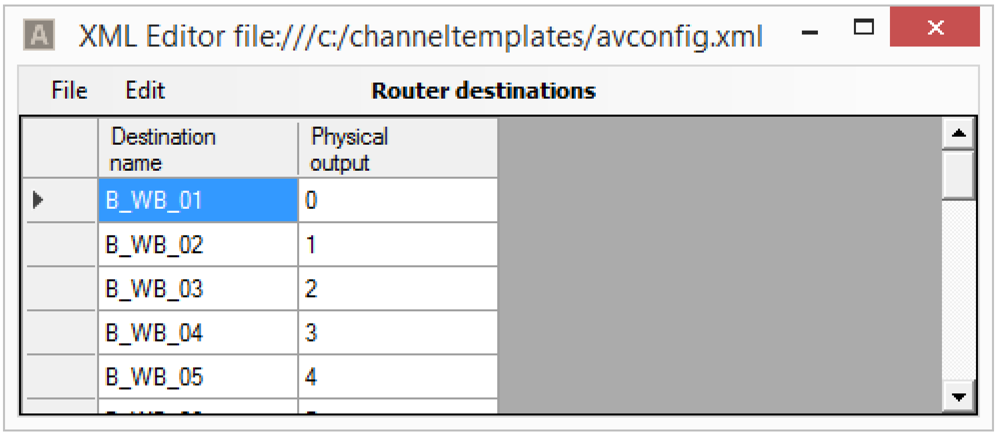
Source name	Physical input
LIVE 1	0
LIVE 2	1
LIVE 3	2
LIVE 4	3
LIVE 5	4
ADHOC 1	5

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 on the router

10.2.5 Router Destinations

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 studio, a transmission output, or anything that needs switching as part of your production.



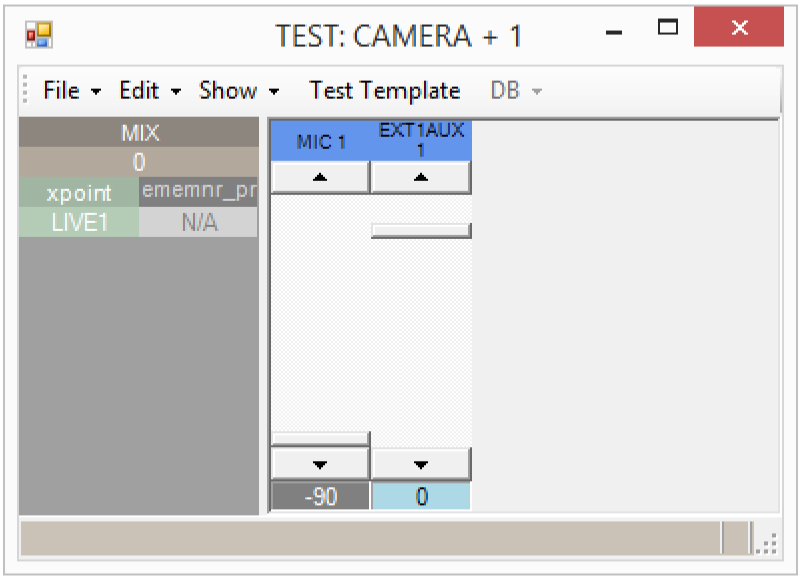
Destination name	Physical output
B_WB_01	0
B_WB_02	1
B_WB_03	2
B_WB_04	3
B_WB_05	4

- **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

10.3 Template Editor

Use the Template Editor to configure Viz Mosart templates. Used in show design.

Open the Template Editor from *AV Automation*. Go to *Main menu* > *Devices* > *Template editor*.



10.4 Template Editor Password

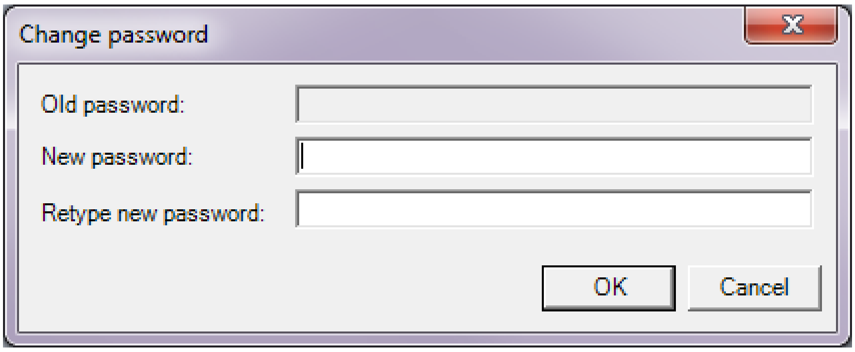
Password protection can be enabled for the Template Editor. If enabled, the user will be prompted for the password when opening the Template Editor.

By default, no protection is enabled, and the user is not asked for a password before opening the Template Editor.

10.4.1 Add or Change Password

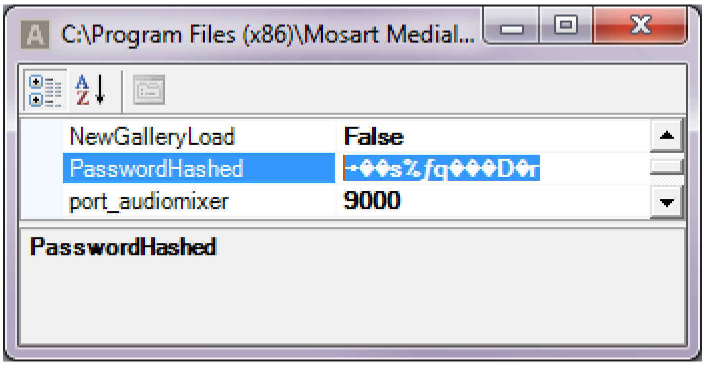
To enable password protection in AV Automation, select *Devices > Change Password*.

In the **Change password** dialog box that opens, a password can be set or an existing one changed.



10.4.2 Reset Password

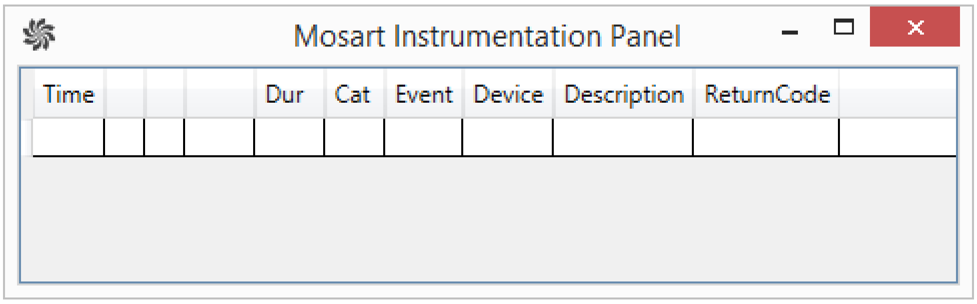
If the password is lost, it can be reset by opening AV Automation Settings (press **CTRL+SHIFT+S** while AV Automation has focus). Find the `PasswordHashed` entry, and remove the value.



NewGalleryLoad	False
PasswordHashed	+��s%fq���D�r
port_audiomixer	9000

PasswordHashed

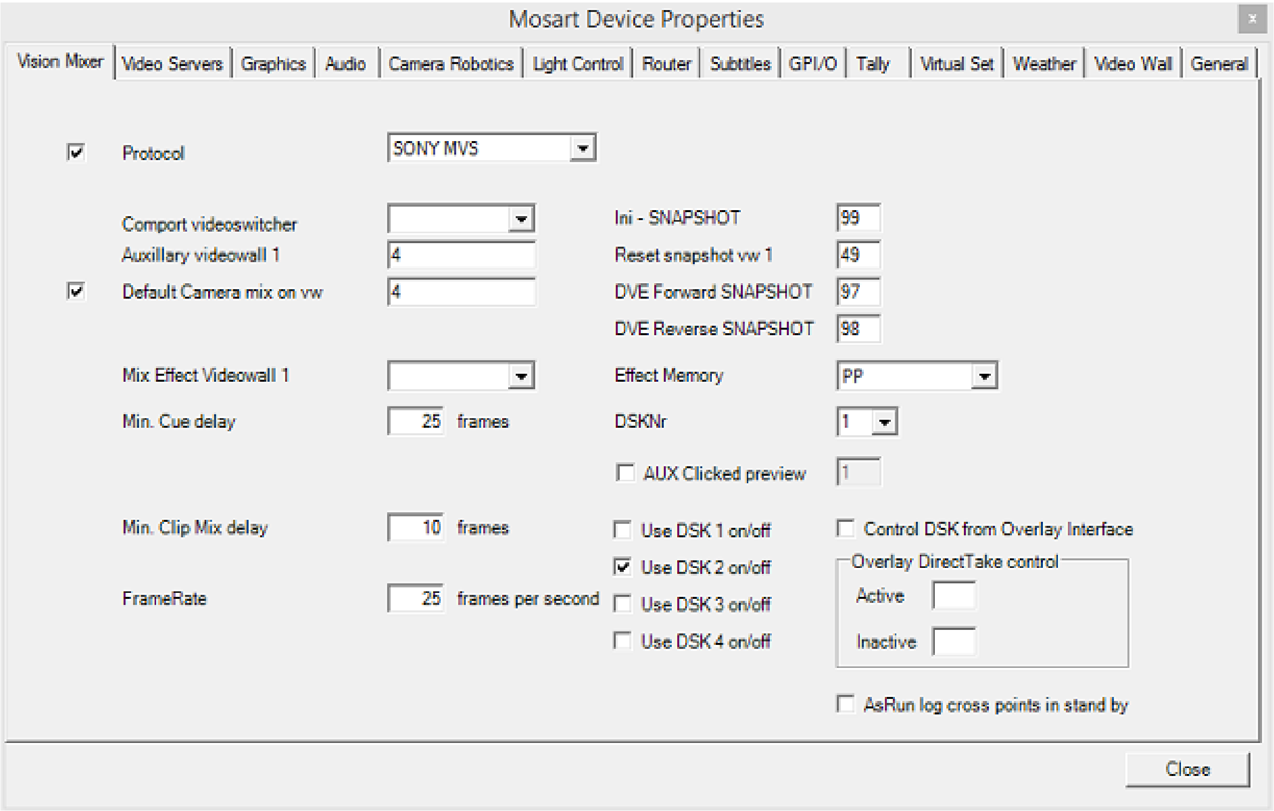
10.5 Mosart Instrumentation Panel



Time				Dur	Cat	Event	Device	Description	ReturnCode

The Mosart Instrumentation Panel displays the all device commands in real time as being sent to the various connected devices. Every command is time stamped making it easy to see when a device command was issued related to the taking of a Mosart template.

11 AV Automation Device Properties



☑ Protocol: SONY MVS

Comport videoswitcher:

Auxillary videowall 1: 4

☑ Default Camera mix on vw: 4

Mix Effect Videowall 1:

Min. Cue delay: 25 frames

Min. Clip Mix delay: 10 frames

FrameRate: 25 frames per second

Ini - SNAPSHOT: 99

Reset snapshot vw 1: 49

DVE Forward SNAPSHOT: 97

DVE Reverse SNAPSHOT: 98

Effect Memory: PP

DSKNr: 1

□ 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

AV Automation sends out commands to all connected broadcast devices. For this to happen, your broadcast devices must be configured for use with Viz Mosart.

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

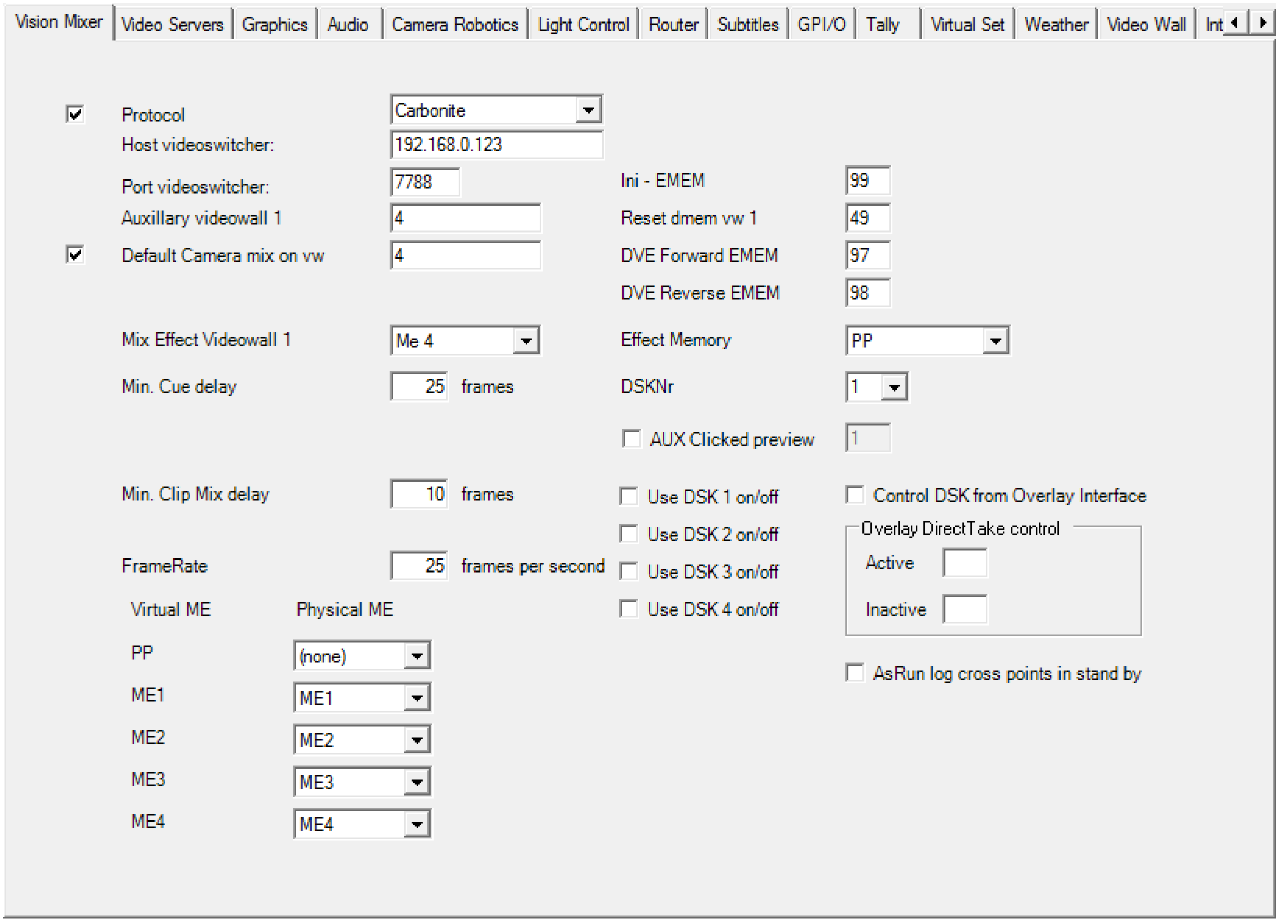
Note: AV Automation must be restarted before any changes take effect.

This section contains the following topics:

- AV Automation Devices - Video 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 - GPI/IO
- 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 - Video Mixer



☑ Protocol: Carbonite
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
Min. Cue delay: 25 frames
Min. Clip Mix delay: 10 frames
FrameRate: 25 frames per second

Virtual ME	Physical ME
PP	(none)
ME1	ME1
ME2	ME2
ME3	ME3
ME4	ME4

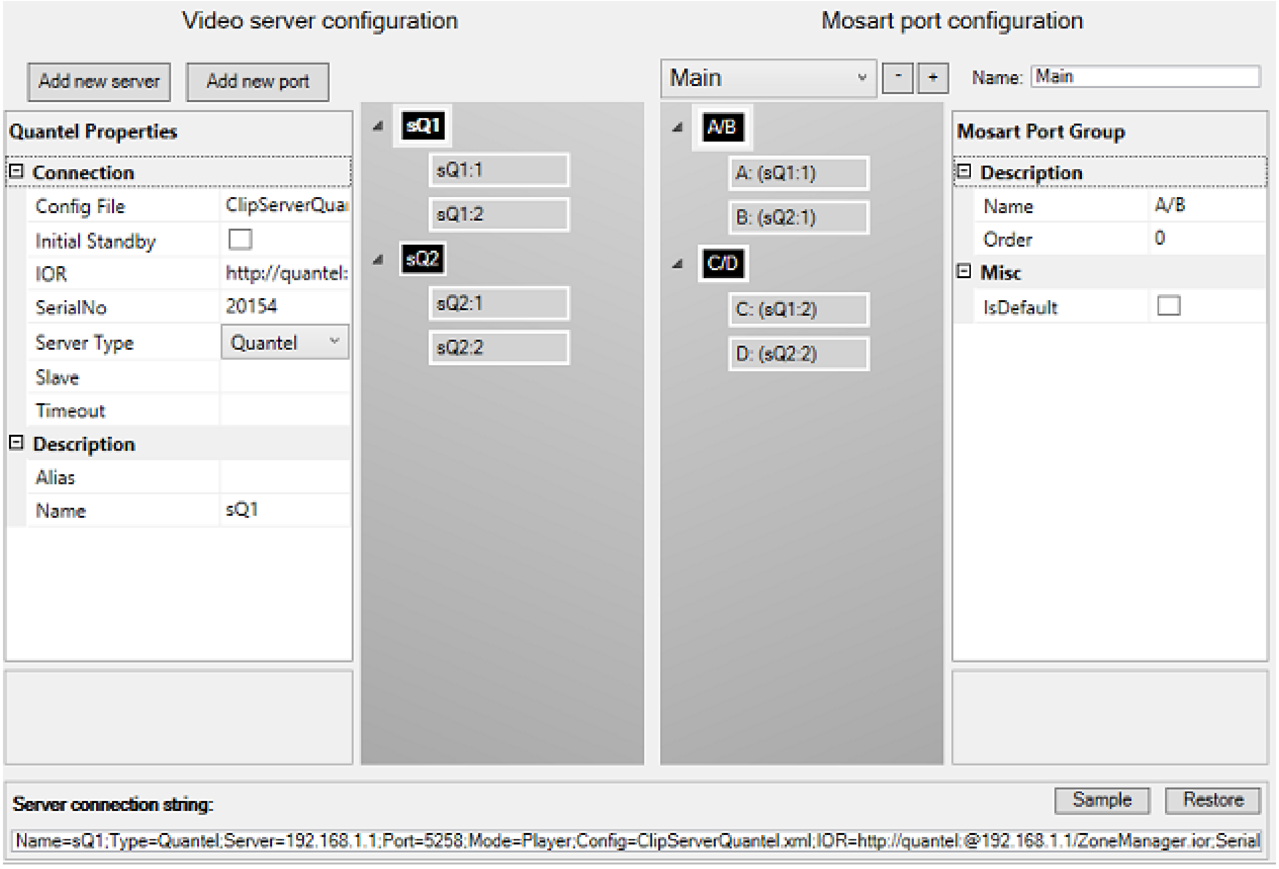
Ini - EMEM: 99
Reset dmem vw 1: 49
DVE Forward EMEM: 97
DVE Reverse EMEM: 98
Effect Memory: PP
DSKNr: 1
□ AUX Clicked preview: 1
□ Use DSK 1 on/off
□ Use DSK 2 on/off
□ Use DSK 3 on/off
□ Use DSK 4 on/off
□ Control DSK from Overlay Interface
Overlay DirectTake control
Active:
Inactive:
□ AsRun log cross points in stand by

Settings are required for Viz Mosart to effectively control the Video Mixer, outlined in the table 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
  - GV KAYENNE PRIMARY
  - GV KAYENNE SECONDARY
  - KAHUNA: Snell Kahuna
  - 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
- **Comport videoswitcher**: The serial port the video switcher is connected to.
- **Host videoswitcher/Port videoswitcher**: Defines the IP and port the video switcher is connected to.

- **Auxiliary videowall 1**: This is the auxiliary output from the video switcher to the video wall.
- **Default Camera mix on vw (check box)**: Check to enable 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.
- **Reset EMEM vw 1**: Register to recall to normalize the ME before entering video wall mode.
- **DVE Forward EMEM**: EMEM to recall for running DVE forward (not supported by Sony).
- **DVE Reverse EMEM**: EMEM to recall for running DVE backward (not supported by Sony).
- **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 1-4.
- **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.
- **Overlay Direct Take control: 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



Name: Main

Quantel Properties

Connection	
Config File	ClipServerQua
Initial Standby	□
IOR	http://quantel:
SerialNo	20154
Server Type	Quantel
Slave	
Timeout	
Description	
Alias	
Name	sQ1

Mosart Port Group

Description	
Name	A/B
Order	0
Misc	
IsDefault	□

Server connection string:

Name=sQ1;Type=Quantel;Server=192.168.1.1;Port=5258;Mode=Player;Config=ClipServerQuantel.xml;IOR=http://quantel:@192.168.1.1/ZoneManager.ior;Serial

Viz Mosart is capable of controlling video servers from many different manufacturers.

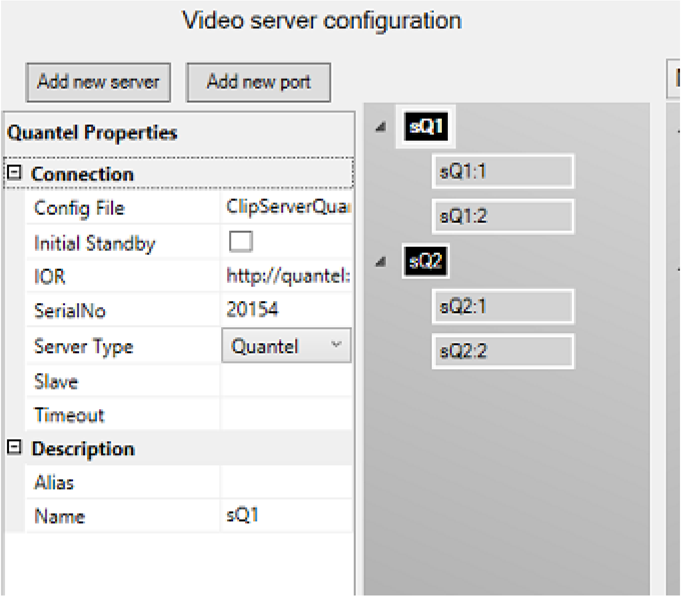
Although the end result is the same, each video server type operates in a slightly different way. It is essential that the **correct settings** are defined in AV Automation for Viz Mosart to effectively and correctly control your video server.

This section contains the following:

- Working with Video Server Configuration
  - To Add a Server
  - To Add a Port
  - To Remove a Port
  - To Remove a Server
- Working with Mosart Port Configuration
  - To Add Virtual Server Groups
  - To Remove a Virtual Ripple Group
  - To Remove a Virtual Port
  - To Connect Server Ports to Virtual Ports
  - To Add Recording Ports
  - About Salvoes
  - To Clear Server Links in a Virtual Port Node
  - To Add a Salvo
  - To Remove a Salvo
  - To Modify a Salvo

- AirSpace, AirSpeed, EVS LinX, EVS Xedio, OradOcip
- AirSpeed MultiStream
- Grass Valley K2
- Nexio
- MVCP
- Omneon
- Quantel
- VDCP
- VDCPtcp
- Viz Engine
- Physical Server Ports
- Virtual Server Ports

11.2.1 Working with Video Server Configuration



Quantel Properties

⊟ Connection	
Config File	ClipServerQua
Initial Standby	□
IOR	http://quantel:
SerialNo	20154
Server Type	Quantel
Slave	
Timeout	
⊟ Description	
Alias	
Name	sQ1

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 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

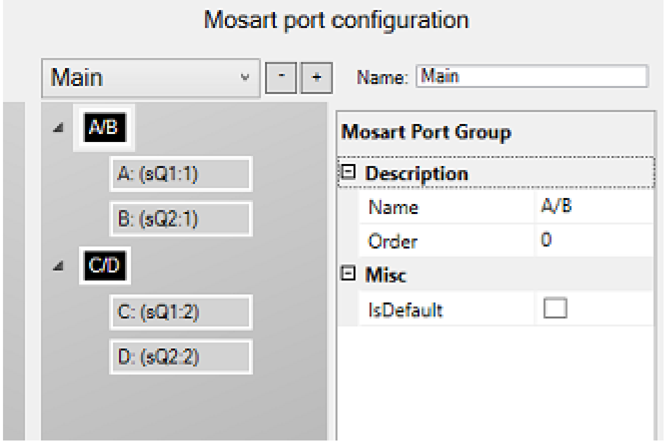
To Remove a Port

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

To Remove a Server

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

11.2.2 Working with Mosart Port Configuration



Name: Main

Name	A/B
Order	0
IsDefault	□

To Add Virtual Server Groups

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 server group. Default group names are *A/B, C/D, E/F* etc.
   - **Add virtual preview group** to add a new green virtual server group node to be used for preview ports. Default name is *P*.
     You can edit or rename the selected group using the Properties Editor on the right.

To Remove a Virtual Ripple Group

1. In the Mosart Port Configuration tree, right-click any *virtual 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 salvoes.

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 salvoes.

To Connect Server Ports to Virtual Ports

To connect server ports to virtual ports:

1. Select the server port you want to connect, and drag it to the virtual port that you will use to represent the port in the selected salvo.
   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 will 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 Add Recording Ports

1. Right-click in the Mosart Port Configuration tree (not on a node) and select *Add virtual recording group. *This will add a new red virtual server group node named *Rec*, which can be used for recording.
   Alternatively, rename an existing group to *Rec*. It will turn red.
2. Right-click the node called *Rec, _add select **Add virtual port**. The new recording port will be called _Rec* by default, do not change the name.

About Salvoes

Salvoes are used to switch between server setups. They are used when the operator needs to switch between server parks, for instance 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.

It is recommended that the default Main and Backup salvoes are not renamed.

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

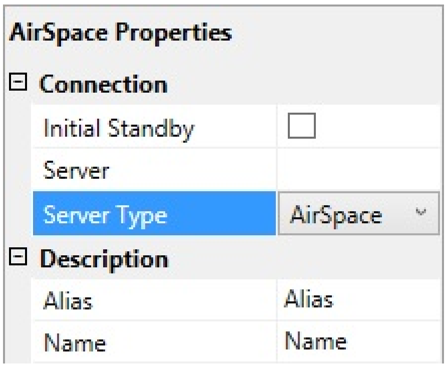
To Remove a Salvo

- 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

11.2.3 AirSpace, AirSpeed, EVS LinX, EVS Xedio, OradOcip

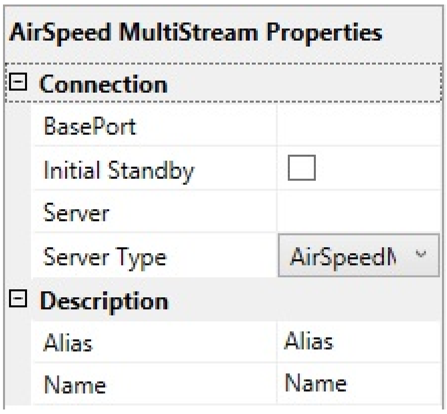


AirSpace Properties

⊟ **Connection**	
Initial Standby	□
Server	
Server Type	AirSpace
⊟ **Description**	
Alias	Alias
Name	Name

- **Initial Standby**: If selected, the server will be forced to start in Standby mode.
- **Server**: Defines the hostname or IP address of the video server.
- **Server Type**: Drop-down list of video server types and protocol.
- **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.

11.2.4 AirSpeed MultiStream

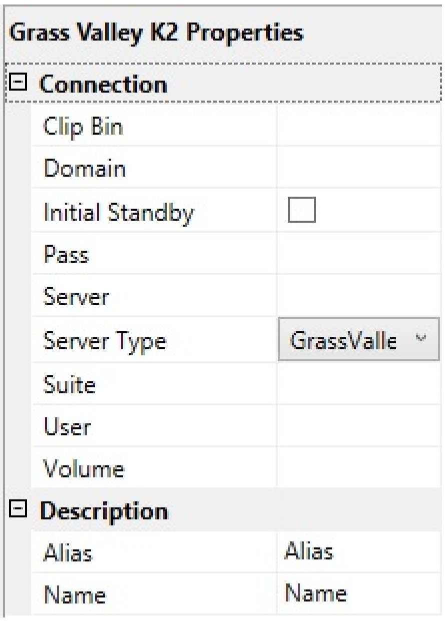


AirSpeed MultiStream Properties

⊟ **Connection**	
BasePort	
Initial Standby	□
Server	
Server Type	AirSpeedN
⊟ **Description**	
Alias	Alias
Name	Name

- **BasePort**: The first port Viz Mosart will use to communicate with the AirSpeed MultiStream server. Default: 59451
- **Initial Standby**: If selected, the server will be forced to start in Standby mode.
- **Server**: Defines the hostname or IP address of the AirSpace video server.
- **Server Type**: Drop-down list of video server types and protocol.
- **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.

11.2.5 Grass Valley K2



Grass Valley K2 Properties

⊟ Connection

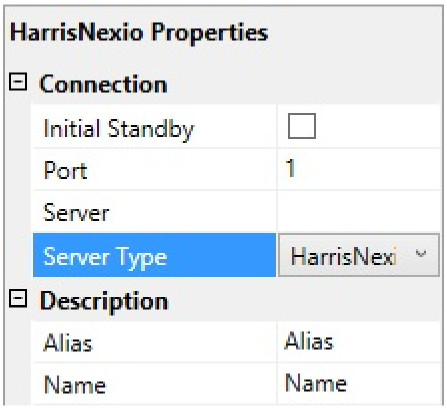
Clip Bin	
Domain	
Initial Standby	□
Pass	
Server	
Server Type	GrassValle
Suite	
User	
Volume	

⊟ 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.
- **Initial Standby**: If selected, the server will be forced to start in Standby mode.
- **Pass**: Defines the password for logging into the server.
- **Server**: Defines the hostname or IP address of the GVG K2 video server.
- **Server Type**: Drop-down list of video server types and protocol.
- **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.
- **Alias**: 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.

11.2.6 Nexio



HarrisNexio Properties

⊟ Connection

Initial Standby	□
Port	1
Server	
Server Type	HarrisNexi

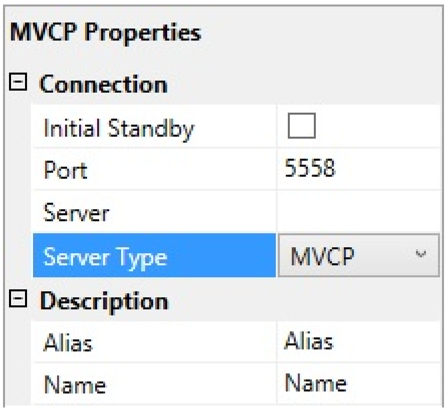
⊟ Description

Alias	Alias
Name	Name

- **Initial Standby**: If selected, the server will be forced to start in Standby mode.
- **Port**: TCP/IP communication port to the Nexio server.

- **Server**: The hostname or IP address of the Nexio video server.
- **Server Type**: Drop-down list of video server types and protocol.
- **Alias**: 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.

11.2.7 MVCP

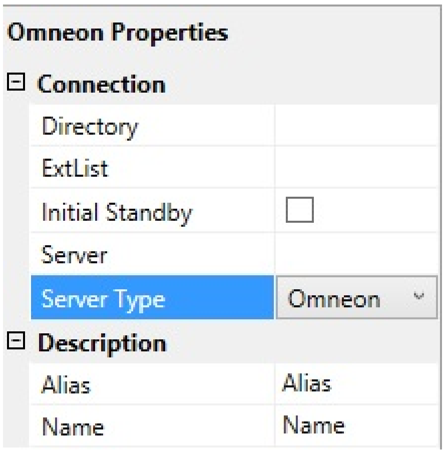


MVCP Properties

⊟ **Connection**	
Initial Standby	□
Port	5558
Server	
Server Type	MVCP
⊟ **Description**	
Alias	Alias
Name	Name

- **Initial Standby**: If selected, the server will be forced to start in Standby mode.
- **Port**: TCP/IP communication port to the MVCP server.
- **Server**: The hostname or IP address of the MVCP video server.
- **Server Type**: Drop-down list of video server types and protocol.
- **Alias**: 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.

11.2.8 Omneon



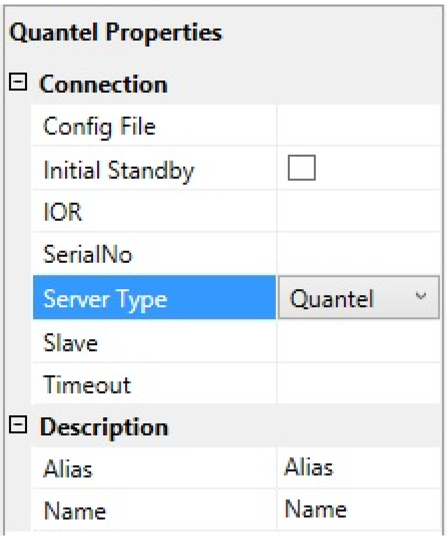
Omneon Properties

⊟ **Connection**	
Directory	
ExtList	
Initial Standby	□
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.
- **Initial Standby**: If selected, the server will be forced to start in Standby mode.
- **Server**: The hostname or IP address of the Omneon video server.
- **Server Type**: Drop-down list of video server types and protocol.
- **Alias**: 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.

11.2.9 Quantel

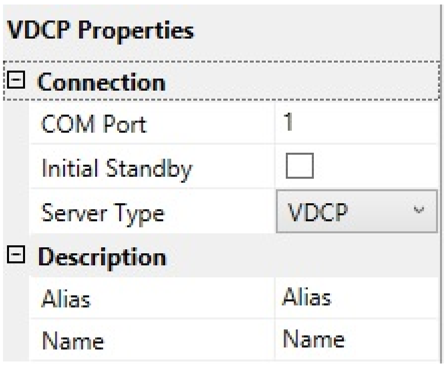


Quantel Properties

Connection	
Config File	
Initial Standby	□
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.
- **Initial Standby**: If selected, the server will be forced to start in Standby mode.
- **IOR**: The HTTP link including hostname or IP address to the IOR resource on the Quantel ISA manager. Example: http://192.168.100.50:2096/ZoneManager.ior
- **SerialNo**: The serial number of the Quantel playout server.
- **Server Type**: Drop-down list of video server types and protocol.
- **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 will be reinitialized. Setting a value here should only be needed for sites experiencing issues with the Quantel connections. Leaving a blank value will use the default timeout
- **Alias**: 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.

11.2.10 VDCP



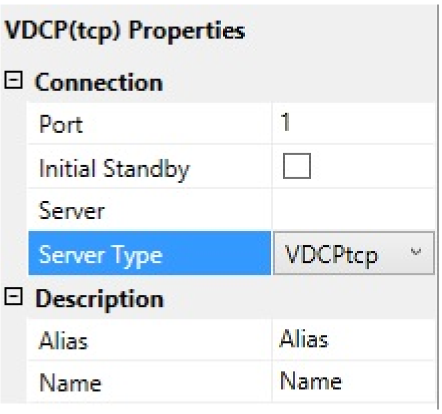
VDCP Properties

Connection	
COM Port	1
Initial Standby	□
Server Type	VDCP
Description	
Alias	Alias
Name	Name

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

- **Initial Standby**: If selected, the server will be forced to start in Standby mode.
- **Server Type**: Drop-down list of video server types and protocol.
- **Alias**: 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.

11.2.11 VDCPtcp



VDCP(tcp) Properties

⊟ **Connection**

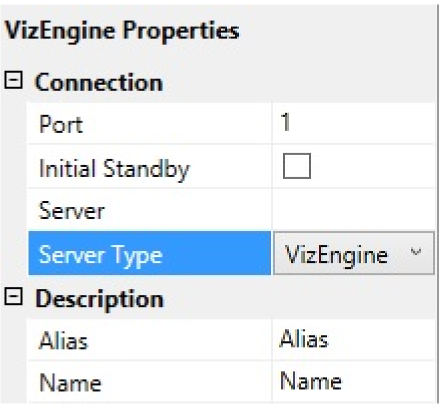
Port	1
Initial Standby	□
Server	
Server Type	VDCPtcp

⊟ **Description**

Alias	Alias
Name	Name

- **Port**: The TCP/IP communication port connected to the VDCP video server.
- **Initial Standby**: If selected, the server will be forced to start in Standby mode.
- **Server**: The hostname or IP address of the VDCP server.
- **Server Type**: Drop-down list of video server types and protocol.
- **Alias**: 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.

11.2.12 Viz Engine



VizEngine Properties

⊟ **Connection**

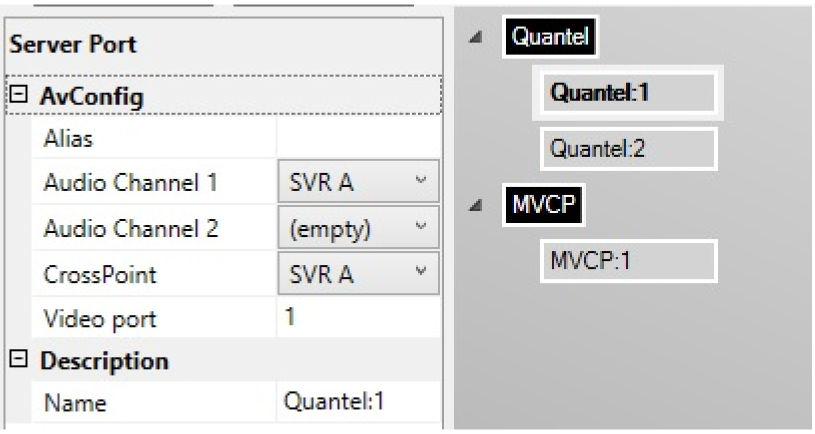
Port	1
Initial Standby	□
Server	
Server Type	VizEngine

⊟ **Description**

Alias	Alias
Name	Name

- **Port**: The TCP/IP communication port connected to the Viz Engine video server.
- **Initial Standby**: If selected, the server will be forced to start in Standby mode.
- **Server**: The hostname or IP address of the Viz Engine video server.
- **Server Type**: Drop-down list of video server types and protocol.
- **Alias**: 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.

11.2.13 Physical Server Ports



Server Port

⊟ AvConfig

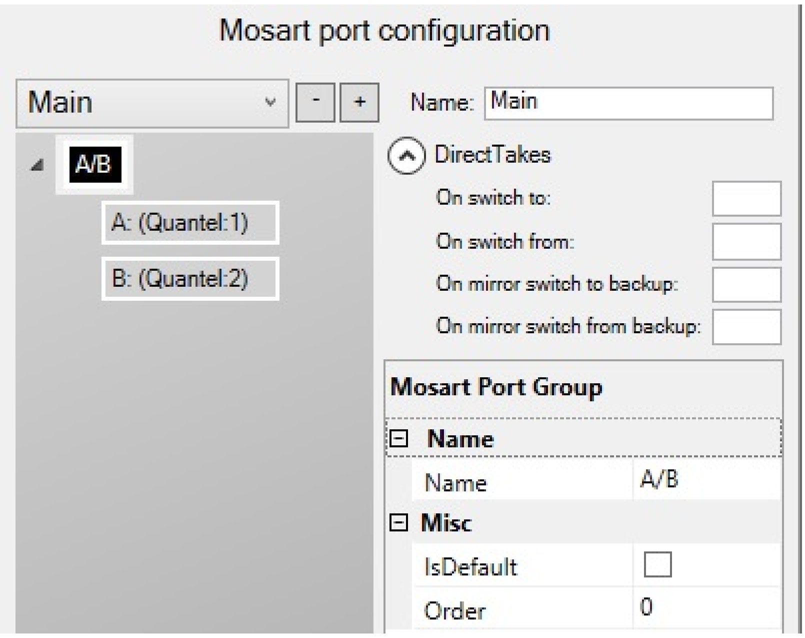
Alias	
Audio Channel 1	SVR A
Audio Channel 2	(empty)
CrossPoint	SVR A
Video port	1

⊟ Description

Name	Quantel:1

- **Alias**: Name of the videoport to display in the Viz Mosart GUI and Timing Display.
- **Audio Channel 1**: The audio channel 1 for the selected videoport.
- **Audio Channel 2**: The audio channel 2for the selected videoport.
- **CrossPoint**: The video crosspoint for the selected videoport.
- **Video port**: The name or number of videoport on the video server for the selected videoport.
- **Name**: Internal name of the videoport, for display in AV Automation of the chosen videoport.

11.2.14 Virtual Server Ports



Mosart port configuration

Main - +

Name: Main

DirectTakes

On switch to:

On switch from:

On mirror switch to backup:

On mirror switch from backup:

Mosart Port Group

⊟ Name

Name	A/B

⊟ Misc

IsDefault	□
Order	0

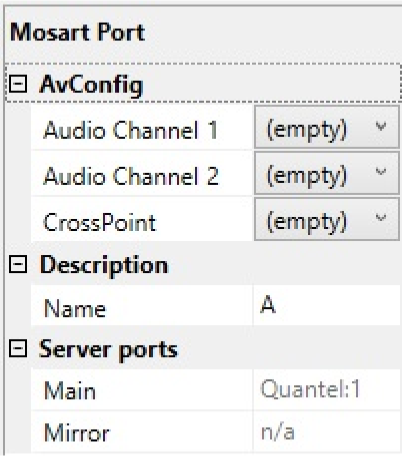
- **Add new salvo**: Use the + icon to add another empty virtual video server salvo.
- **Remove a salvo**: Use the **-** icon to remove/delete a virtual video server salvo.
- **Switch between salvos**: Use the drop-down menu to select another virtual video server salvo.
- **Name**: Defines the name of 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)
Audio Channel 2	(empty)
CrossPoint	(empty)

⊟ Description

Name	A

⊟ Server ports

Main	Quantel:1
Mirror	n/a

Mosart Port

- **Audio Channel 1**: The virtual audio fader 1 for the selected videoport. If nothing is selected here, the audio fader from the physical part will be used. Default: Empty
- **Audio Channel 2**: The virtual audio fader 1 for the selected videoport. If nothing is selected here, the audio fader from the physical part will be used. Default: Empty
- **CrossPoint**: The virtual video crosspoint for the selected videoport. If nothing is selected here, the video crosspoint from the physical part will be 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.

11.3 AV Automation Devices - Graphics

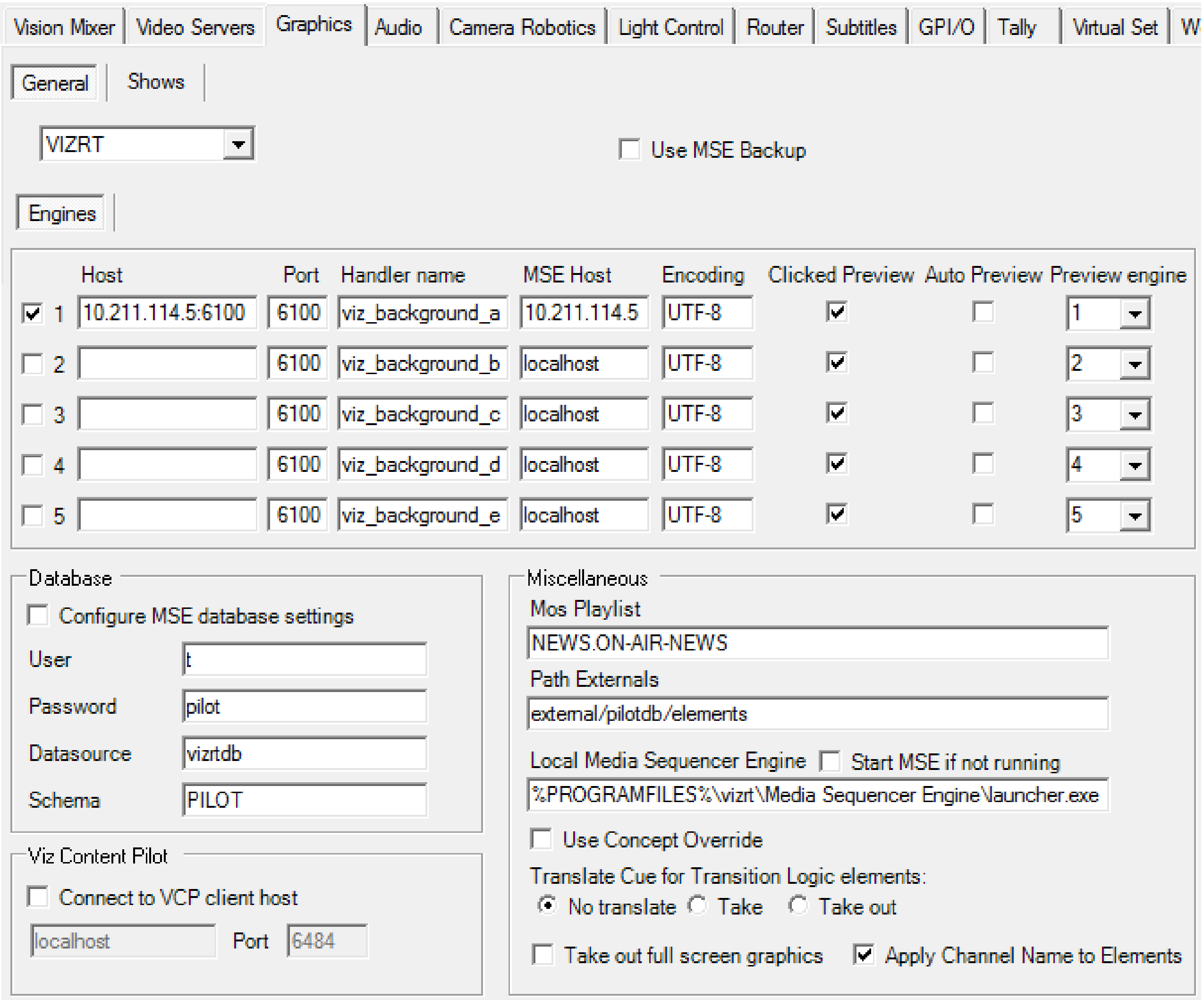
Viz Mosart is capable of connecting to a variety of Graphic Engines available from many different manufacturers. AV Automation handles full frame graphics playout, from a primary template type in the rundown.

Tip: Set **GraphicsConnectedWhenIdle**. When set to true, graphics devices will stay connected when in idle mode.

The properties available depend on the system to be used:

- Vizrt Settings
- Deko Settings
- XPression Settings
- Orad Settings
- Pixel Power Settings
- Chyron Settings

11.3.1 Vizrt Settings



VIZRT

□ Use MSE Backup

Engines

		Host	Port	Handler name	MSE Host	Encoding	Clicked Preview	Auto Preview	Preview engine
☑	1	10.211.114.5:6100	6100	viz_background_a	10.211.114.5	UTF-8	☑	□	1
□	2		6100	viz_background_b	localhost	UTF-8	☑	□	2
□	3		6100	viz_background_c	localhost	UTF-8	☑	□	3
□	4		6100	viz_background_d	localhost	UTF-8	☑	□	4
□	5		6100	viz_background_e	localhost	UTF-8	☑	□	5

Database

□ Configure MSE database settings

User: t

Password: pilot

Datasource: vizrtdb

Schema: PILOT

Viz Content Pilot

□ Connect to VCP client host

localhost Port 6484

Miscellaneous

Mos Playlist: NEWS.ON-AIR-NEWS

Path Externals: external/pilotdb/elements

Local Media Sequencer Engine □ Start MSE if not running

%PROGRAMFILES%\vizrt\Media Sequencer Engine\launcher.exe

□ Use Concept Override

Translate Cue for Transition Logic elements:

● No translate ○ Take ○ Take out

□ Take out full screen graphics ☑ Apply Channel Name to Elements

- **Use MSE Backup**: Check to use Media Sequencer backup engine when starting the system.
- **Engines**: Check to enable connection to each Viz Engine. Viz Mosart supports five graphics engine connections.
- **Host**: Hostname or IP address to the Viz Engine.
- **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.
- **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 Pilot

- **Connect to VCP client host**: Check to enable 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.
- **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

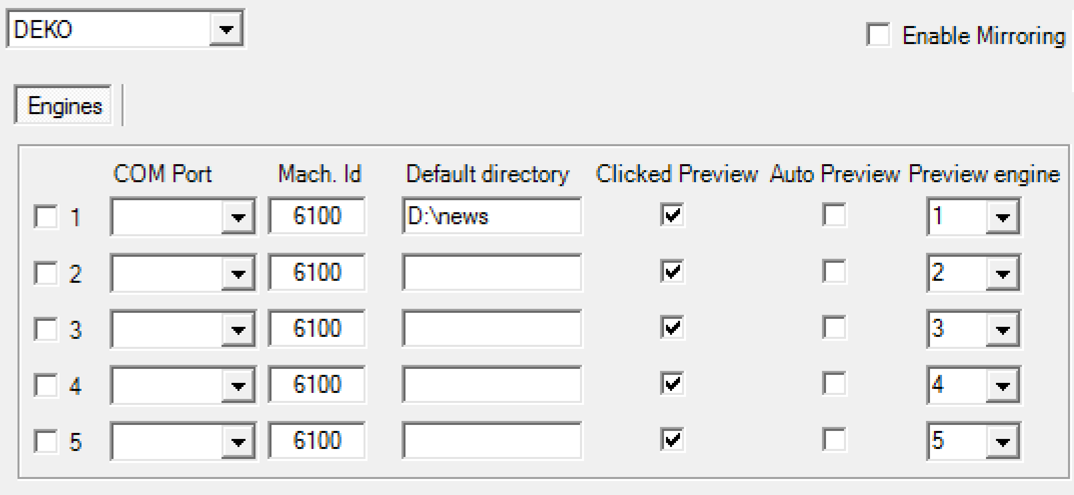
Note

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**: Check to enable the Vizrt Concept Override if your scenes are prepared for this. Viz Pilot 5.2 or later is needed.
- **Translate Cue for Transition Logic elements**: Only applicable when using Transition Logic full screen graphics as the Media Sequnecer ignores a normal cue command for these items.
    - No translate: Send cue command to the Media Sequencer (on Media Sequencer versions below 1.20 no cue will be performed).
    - Take: Use a take command when cuing.
    - Take out: Use a take out command when cuing.

- **Take out full screen graphics**: Check to take out full screen graphics. By default, Viz Mosart will not do a takeout for full screen graphics.
- **Apply Channel Name to Elements**: This will send the Channel name to be displayed with the graphic elements in the GUI.
- **Viz Pilot: Connect to VCP client (check box)**:
- **Shows**: If the full screen graphics concept should change when a new Graphics Profile is selected, then this is configured here. Please refer to the ‘Shows’ fields in the Engines Setup Tab. Take note that the "Display name" of both configurations must be exact.

11.3.2 Deko Settings



DEKO

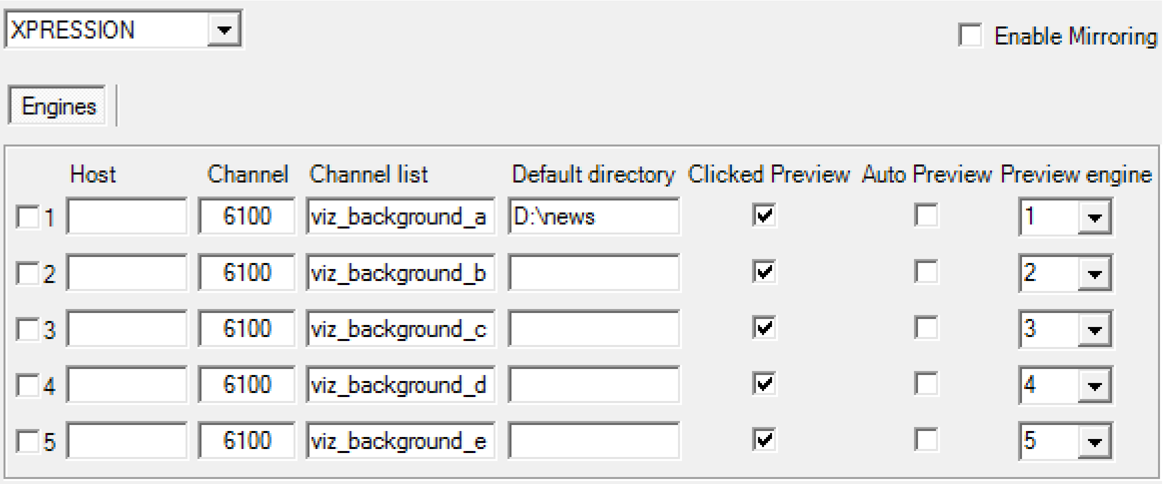
☐ Enable Mirroring

Engines

	COM Port	Mach. Id	Default directory	Clicked Preview	Auto Preview	Preview engine
☐ 1		6100	D:\news	☑	☐	1
☐ 2		6100		☑	☐	2
☐ 3		6100		☑	☐	3
☐ 4		6100		☑	☐	4
☐ 5		6100		☑	☐	5

- **Engines**: Check to enable the connections to the Deko engines. Viz Mosart supports five graphics engine connections.
- **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 ▾

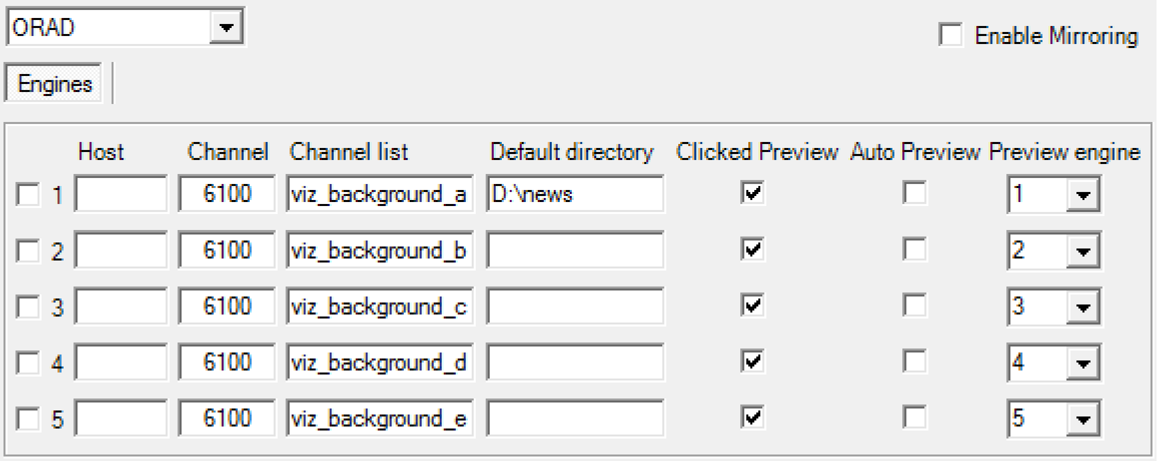
☐ Enable Mirroring

Engines

	Host	Channel	Channel list	Default directory	Clicked Preview	Auto Preview	Preview engine
☐ 1		6100	viz_background_a	D:\news	☑	☐	1
☐ 2		6100	viz_background_b		☑	☐	2
☐ 3		6100	viz_background_c		☑	☐	3
☐ 4		6100	viz_background_d		☑	☐	4
☐ 5		6100	viz_background_e		☑	☐	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

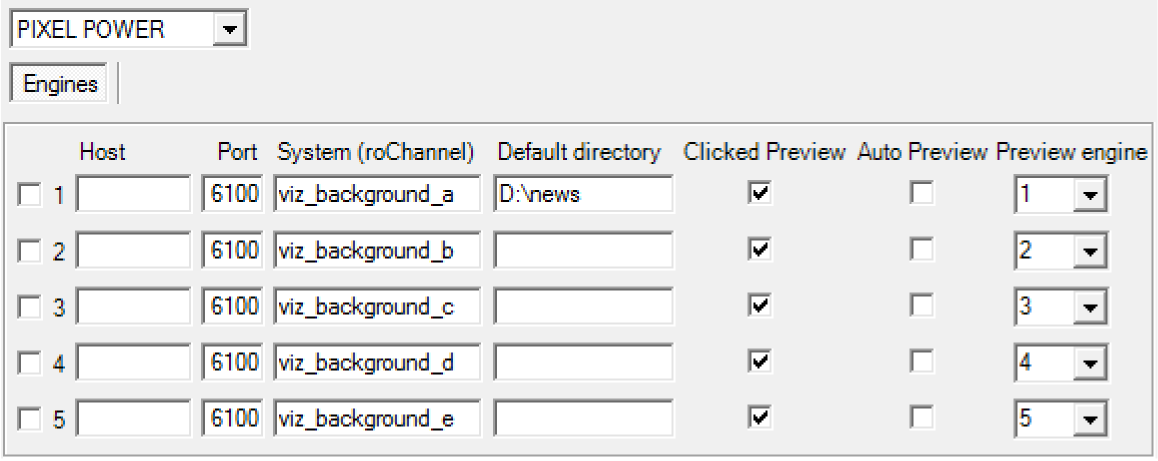
Engines

	Host	Channel	Channel list	Default directory	Clicked Preview	Auto Preview	Preview engine
☐ 1		6100	viz_background_a	D:\news	☑	☐	1
☐ 2		6100	viz_background_b		☑	☐	2
☐ 3		6100	viz_background_c		☑	☐	3
☐ 4		6100	viz_background_d		☑	☐	4
☐ 5		6100	viz_background_e		☑	☐	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



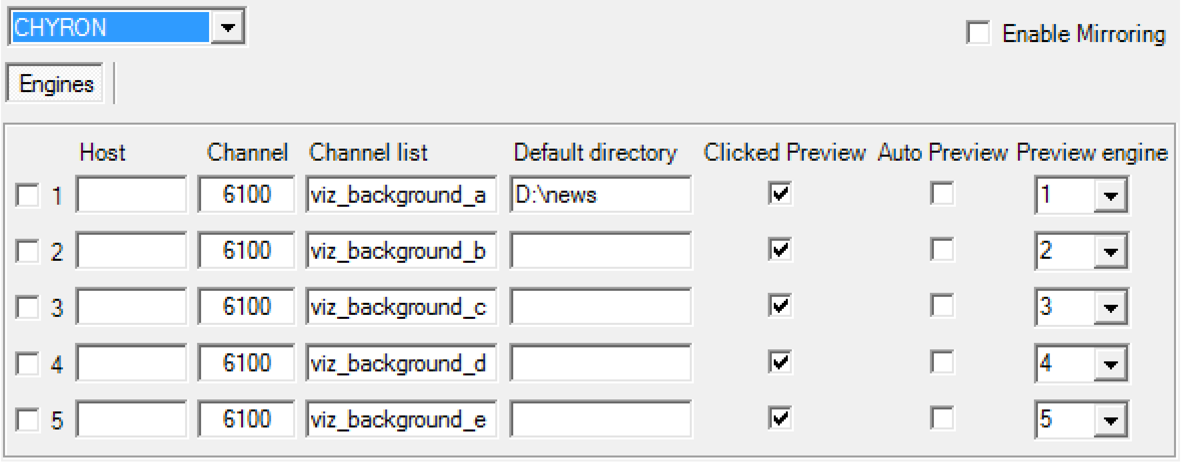
PIXEL POWER

Engines

		Host	Port	System (roChannel)	Default directory	Clicked Preview	Auto Preview	Preview engine
□	1		6100	viz_background_a	D:\news	☑	□	1
□	2		6100	viz_background_b		☑	□	2
□	3		6100	viz_background_c		☑	□	3
□	4		6100	viz_background_d		☑	□	4
□	5		6100	viz_background_e		☑	□	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
□	1		6100	viz_background_a	D:\news	☑	□	1
□	2		6100	viz_background_b		☑	□	2
□	3		6100	viz_background_c		☑	□	3
□	4		6100	viz_background_d		☑	□	4
□	5		6100	viz_background_e		☑	□	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.

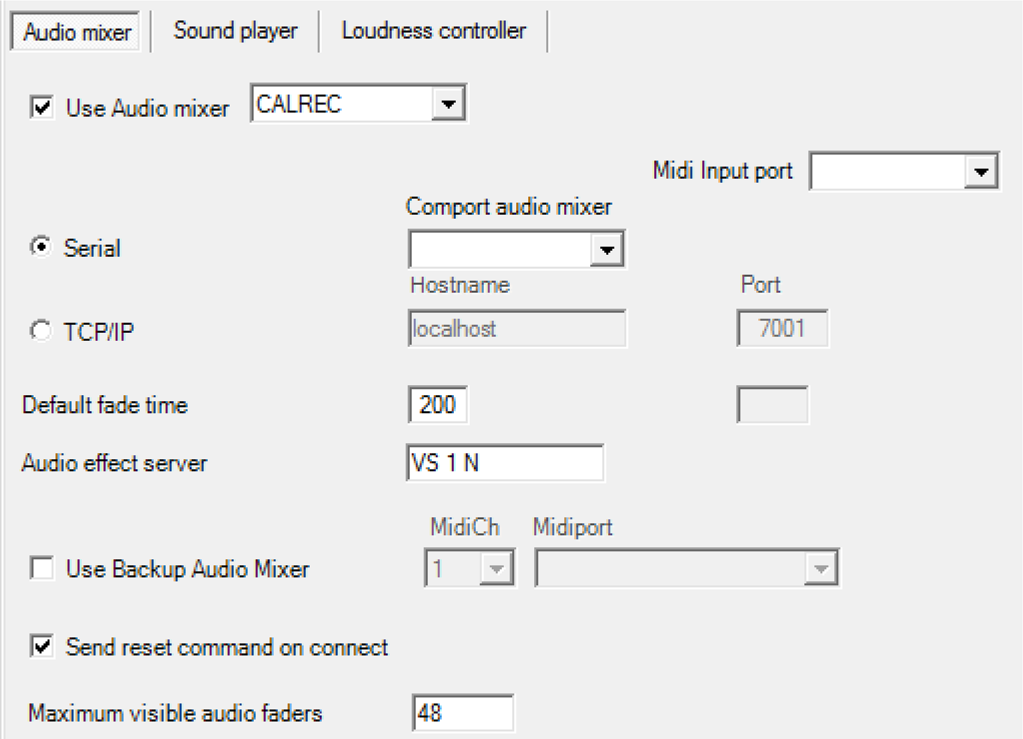
Select the correct Audio Mixer from the menu for your broadcast environment. AV Automation handles the following:

- Audio Mixer, a backup MIDI Audio Mixer
- Loudness Control from Junger

See Also

- Audio Player

11.4.1 Audio Mixer



☑ Use Audio mixer: CALREC

Midi Input port:

● Serial — Comport audio mixer:

○ TCP/IP — Hostname: localhost — Port: 7001

Default fade time: 200

Audio effect server: VS 1 N

□ Use Backup Audio Mixer — MidiCh: 1 — Midiport:

☑ Send reset command on connect

Maximum visible audio faders: 48

- **Use Audio mixer**: Check the box to enable the audio mixer functionality.
- **Audio mixer type**:
  - Calrec: Calrec Serial Protocol
  - DHD: RM4200-D Fa. DHD

- LAWO RMNOPL: RemoteMNOPL protocol
- LAWO ZIRKON: RAS subset protocol
- SSL: Light Broadcast Automation protocol
- STAGETEC: Stagetec RAS protocol
- STUDER_3000: For audio mixers supporting the MONITORA protocol
- WHEATSTONE: Wheatstone Mixer Automation Protocol
- YAMAHA_O2: For audio mixers controlled through MIDI control change and parameter change
- STUDER VISTA: Control of the Studer Vista series via the Ember protocol

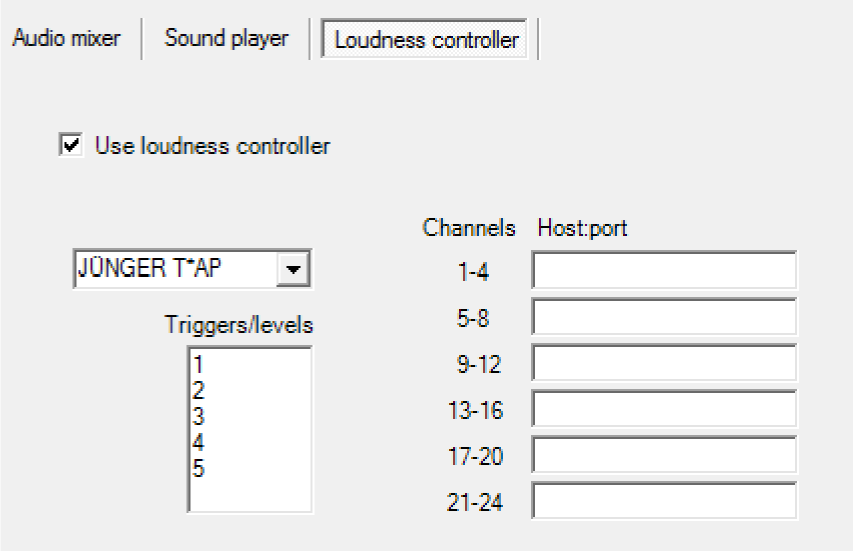
- **Serial (Calrec, Lawo, Studer, etc.)**: Serial port connected to the audio mixer (when Serial radio button is selected).
- **TCP/IP**: Hostname (or IP address) and port to communicate with the audio mixer (when TCP/IP radio button is selected).
- **Backup hostname and port (Lawo)**: Hostname (or IP address) and port to the backup audio mixer console.
- **Set user labels on console**: When selected, user labels on the Lawo 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**: Check box to enable 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	**DHD**	**LAWO RMNOPL**	**LAWO ZIR KON**	**SSL**	**STAG ETEC**	**STUDER 3000**	**Wheatstone**	**YAMAHA O2**	**STUDER VISTA**	**EUPH**
Serial	x	**x**	**x**							
TCP/IP	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**									

CAL REC	DHD	LAWO RMNO	LAWO ZIRKON	SSL	STAG ETEC	STUDER 3000	Wheatstone	YAMAHA O2	STUDER VISTA	EUPH
BackupPort	**x**									
Faderinputoffset	**x**									
Useemberprotocol	**x**									
Comportaudiomixer	**x**	x	x	x	**x**	x				

11.4.2 Loudness Control



☑ Use loudness controller

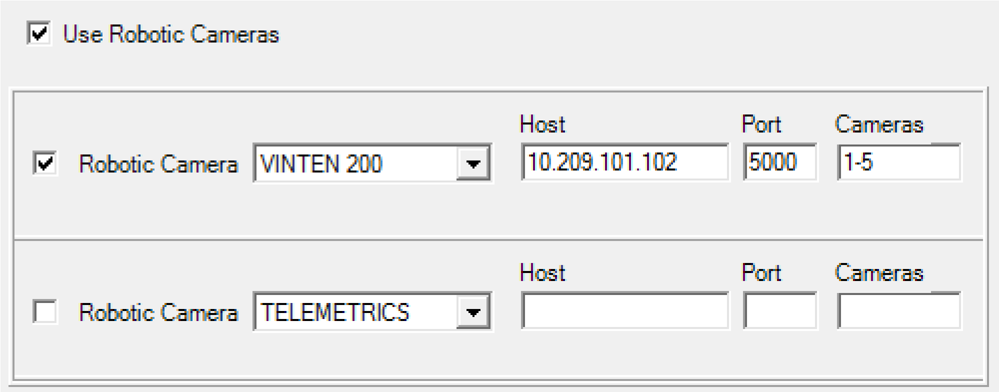
JÜNGER T*AP

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**: JUNGER T*AP
- **Triggers/levels**: Name of triggers or loudness levels, one for each line. These will appear in a tool-tip menu in the Template editor, and must correspond to the configuration in the loudness controller (s).
- **Channels**: Fixed values. For Junger T*AP there are 4 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.

11.5 AV Automation Devices - Camera Robotics



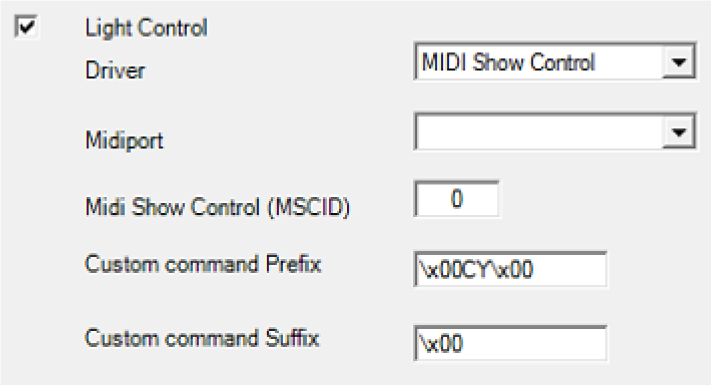
☑ Use Robotic Cameras

☑ Robotic Camera: VINTEN 200 — Host: 10.209.101.102 — Port: 5000 — Cameras: 1-5

☐ Robotic Camera: TELEMETRICS — Host: — Port: — Cameras:

- **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.

11.6 AV Automation Devices - Light Control



☑ Light Control

Driver: MIDI Show Control

Midiport:

Midi Show Control (MSCID): 0

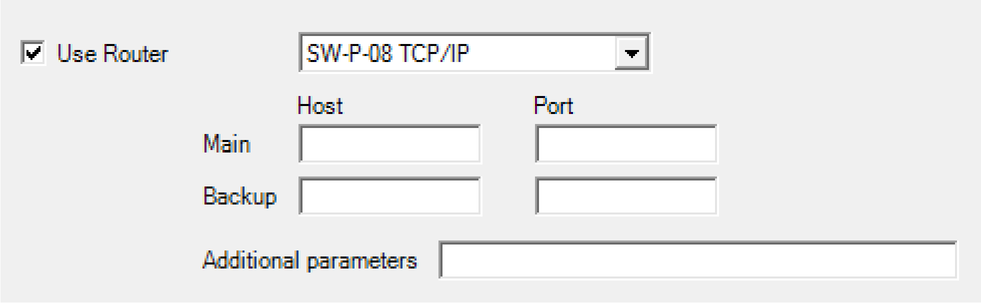
Custom command Prefix: \x00CY\x00

Custom command Suffix: \x00

- **Light Control**: Check to enable 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



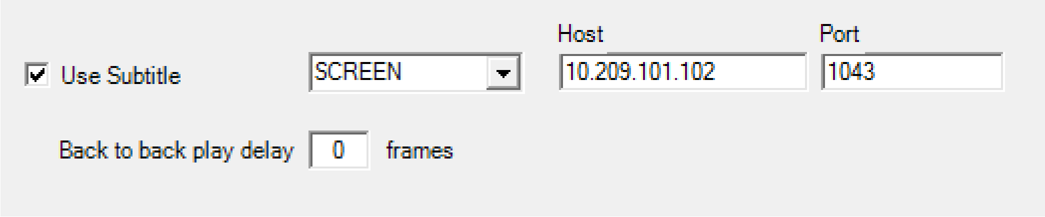
☑ Use Router: SW-P-08 TCP/IP

	Host	Port
Main		
Backup		

Additional parameters:

- **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

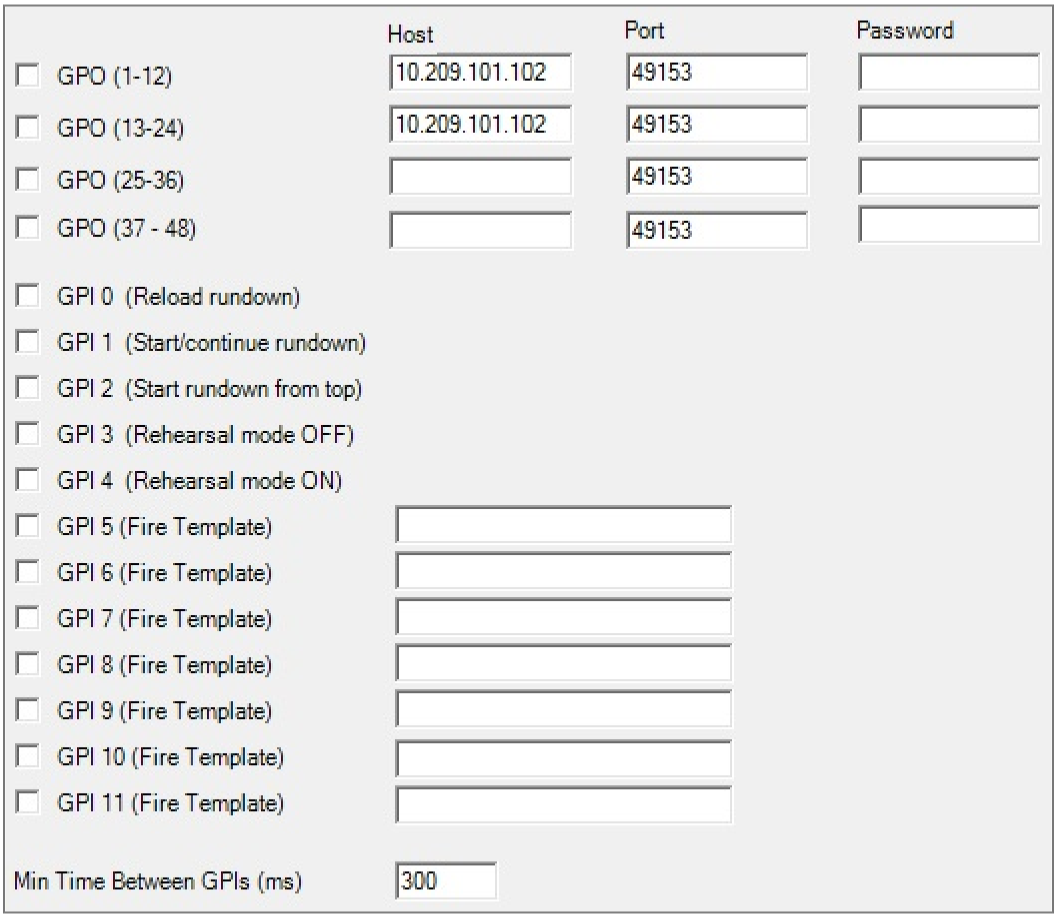
Host: 10.209.101.102

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 - GPI/IO



	Host	Port	Password
□ GPO (1-12)	10.209.101.102	49153	
□ GPO (13-24)	10.209.101.102	49153	
□ GPO (25-36)		49153	
□ GPO (37 - 48)		49153	

□ GPI 0 (Reload rundown)

□ GPI 1 (Start/continue rundown)

□ GPI 2 (Start rundown from top)

□ GPI 3 (Rehearsal mode OFF)

□ GPI 4 (Rehearsal mode ON)

□ GPI 5 (Fire Template)

□ GPI 6 (Fire Template)

□ GPI 7 (Fire Template)

□ GPI 8 (Fire Template)

□ GPI 9 (Fire Template)

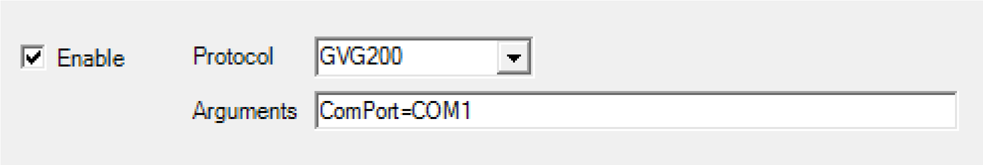
□ GPI 10 (Fire Template)

□ GPI 11 (Fire Template)

Min Time Between GPIs (ms) 300

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

11.10 AV Automation Devices - Tally



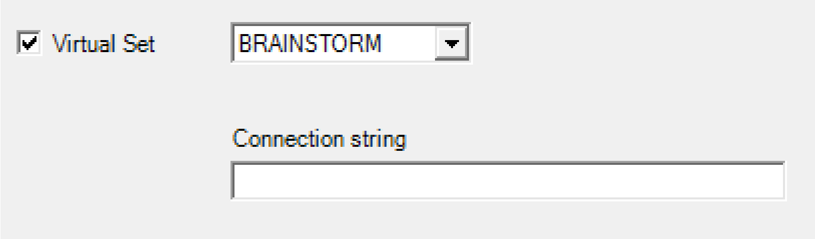
☑ Enable

Protocol: GVG200

Arguments: ComPort=COM1

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

11.11 AV Automation Devices - Virtual Set

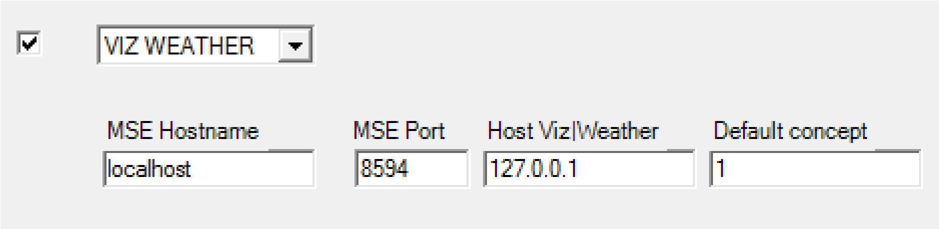


☑ Virtual Set: BRAINSTORM

Connection string:

- **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

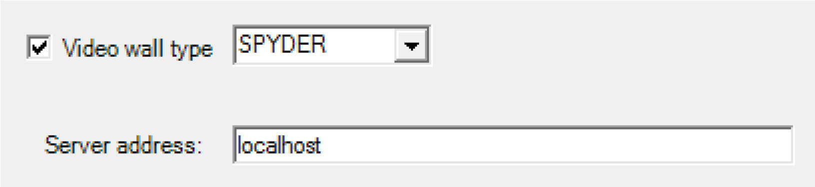


☑ VIZ WEATHER

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

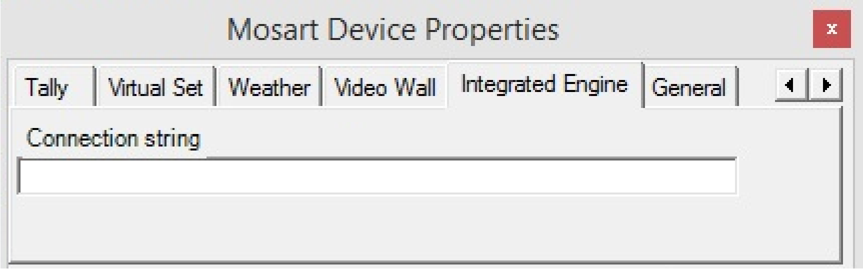


☑ Video wall type: SPYDER

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 Device Connection Strings > Watchout Connection String.
- **Backup (Pandora)**: Hostname or IP address of the backup PB Widget Designer.

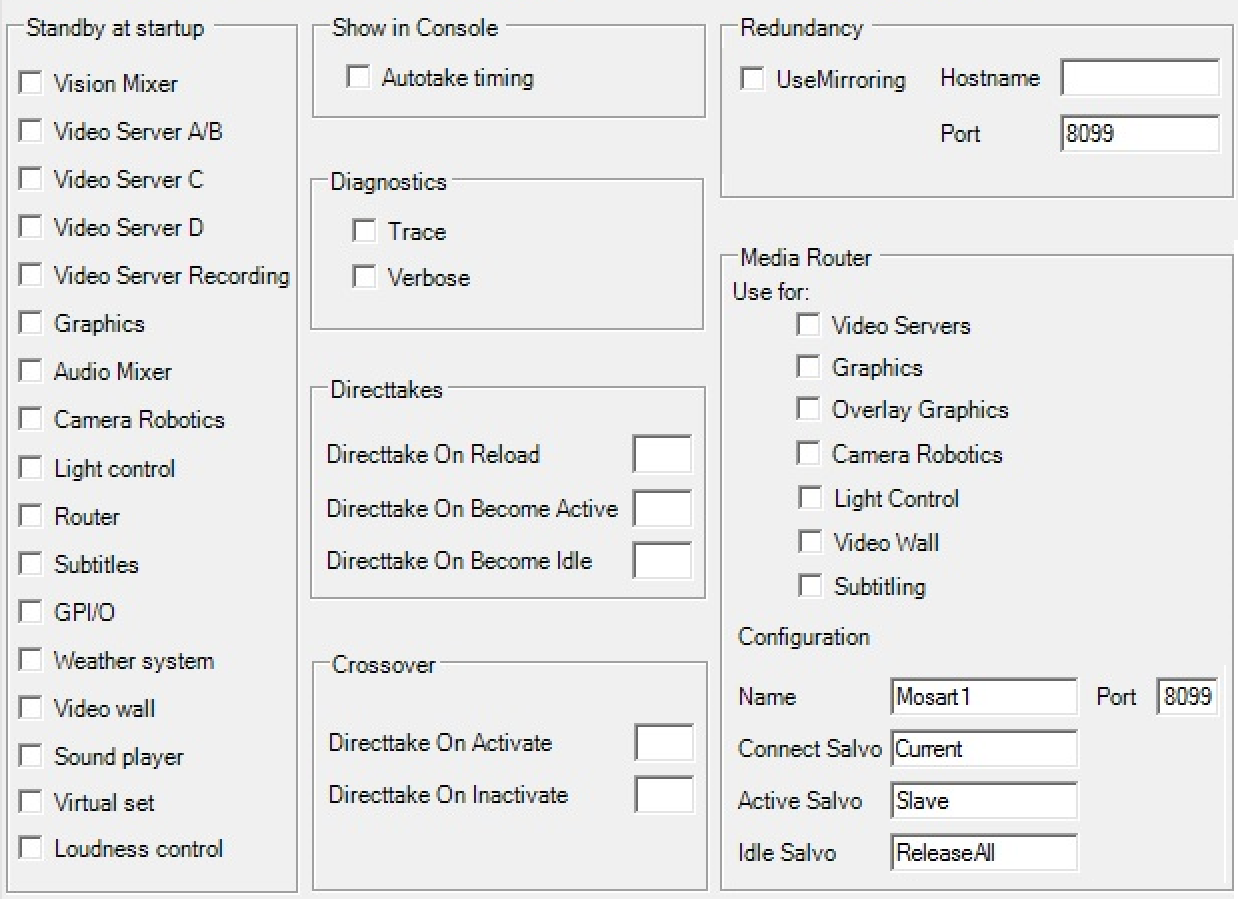
11.14 AV Automation Devices - Integrated Engine



Connection string:

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

11.15 AV Automation Devices - General



Standby at startup

□ Vision Mixer
□ Video Server A/B
□ Video Server C
□ Video Server D
□ Video Server Recording
□ Graphics
□ Audio Mixer
□ Camera Robotics
□ Light control
□ Router
□ Subtitles
□ GPI/O
□ Weather system
□ Video wall
□ Sound player
□ Virtual set
□ Loudness control

Show in Console

□ Autotake timing

Diagnostics

□ Trace
□ Verbose

Directtakes

Directtake On Reload
Directtake On Become Active
Directtake On Become Idle

Crossover

Directtake On Activate
Directtake On Inactivate

Redundancy

□ UseMirroring
Hostname
Port 8099

Media Router

Use for:
□ Video Servers
□ Graphics
□ Overlay Graphics
□ Camera Robotics
□ Light Control
□ Video Wall
□ Subtitling

Configuration

Name Mosart1 Port 8099
Connect Salvo Current
Active Salvo Slave
Idle Salvo ReleaseAll

Standby at startup

- Select the drivers to start in standby mode. They must be taken out of standby manually.
  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 will be 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 Template Database

The Template Database (TDB) is used to synchronise Template information between Viz Mosart Servers, the ActiveX as found in the NCS, and Viz Mosart GUI.

The Template Database allows for the synchronisation of Viz Mosart Template Sets between studios in a large organisation. It relies on a relational database to store and sync data across servers, workstations, and Viz Mosart applications.

Changes to TDB are done via the Template Editor.

Usually TDB can be installed on any computer connected to Viz Mosart Server. It may be a dedicated database server or hosted on another MySQL server available on the network, alternatively the Viz Mosart Database can be virtualised.

Note: If an external MySQL install is used, please note minimum version requirements as outlined in Installing WampServer.

This section contains the following topics:

- Installing the Viz Mosart Database
- Creating and Upgrading Mosart templatedb
- Starting phpMyAdmin
- Creating the Database Manually
- Upgrading mosarttemplatedb Manually
- Change Scripts
- Configuring Template Database
- Maintenance

12.1 Installing the Viz Mosart Database

Installing and creating the Viz Mosart Database is outlined in this section.

This functionality has been developed and tested on MySQL, consequently Viz Mosart recommends MySQL for hosting the Viz Mosart Database.

Setting up the Viz Mosart Database is done as follows:

1. Install MySQL
2. Create the database
3. Configure Viz Mosart to use the database
   Configuring AV Automation to work with the Viz Mosart Database is outlined in Template Database Settings in AV Automation.

This section contains the following topics:

- Installing WampServer
- MySQL Security

12.1.1 Installing WampServer

WampServer is a software package including, among others, MySQL and the management tool phpMyAdmin.

At the time of writing, WampServer 2.1e, includes MySQL 5.5.8 (x86) or MySQL 5.1.53 (x64).

The template Database has been developed and tested on this version, consequently Viz Mosart recommends installing WampServer 2.1e or later.

IMPORTANT! WampServer has to be installed without the following Windows components: HTTP Support for MSMQ (part of Message Queuing).

WampServer is installed by following the steps below, the exact steps may vary on OS used:

1. Using a web browser, navigate to: http://www.wampserver.com/en/download\-wampserver\-32bits/
2. Download the appropriate version of WampServer
3. Run the downloaded file
   If WampServer is installed on a server also running Internet Information Services further action should be taken:

1. Start WampServer from the Start menu. The WampServer icon should then appear in green in the system tray.
2. Click the WampServer icon, select Apache, and then Version, to see the Apache version, it should be similar to 2.2.17
3. Locate the following file, if you are unable to locate it move to step 9 and take extra care: C:\wamp\bin\apache\Apache<version>\conf\httpd.conf.
4. Make a backup copy of the httpd.conf file
5. Click the WampServer icon, select Apache, and then httpd.conf
   A Notepad with the file contents should appear.
6. Locate the line:

   ```
   Listen 80
   ```

7. Make a copy of this line, comment the original line and rename the new line as follows:

   ```
   #Listen 80 Listen 8080
   ```

8. Save and exit.
9. Restart WampServer by clicking the WampServer icon and selecting Restart all services.
   Verify that no error messages are given

12.1.2 MySQL Security

As default MySQL server creates a user with all privileges.

For security purposes, please create a user relevant to your broadcast environment, and disable or set a password for the default root user.

This will also appear as an alert within MyPHPAdmin on this homepage during initial setup

12.2 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’s 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

12.2.1 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

12.2.2 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, however Viz Mosart recommends running scripts automatically.

If problems arise when running scripts automatically, more information on manually running scripts can be found later in this section

12.2.3 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-extra-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 followed by 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.
     Save the file if you have changed it

12.2.4 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`

12.3 Starting phpMyAdmin

Start phpMyAdmin by clicking the green WampServer icon found in the system tray and selecting phpMyAdmin.

Wait for phpMyAdmin to start.

When started, the phpMyAdmin interface should have several elements, among them the following tabs:

- Databases
- Import

12.4 Creating the Database Manually

Create the database using phpMyAdmin and following the steps outlined below:

1. Click the Databases tab
   A list of databases should appear
2. Check if mosarttemplatedb is present in the database list
   If so, continue to step 3, otherwise continue with step 6
3. Click the mosarttemplatedb database.
   The tabs should now change, to show Structure and Import
4. Click the Structure tab.
   A list of tables should appear
5. Check if de_device and em_enumerationmember tables are present, if both are present, the DB has already been created and you should upgrade the database as described in 0
   If one, or both tables are missing, continue to step 6
6. Choose the Import tab.
7. Click the Browse button, a ’Choose File to Upload’ (or similar) dialog box should appear.
8. In the dialog box, choose the following:
   %ProgramFiles%\Mosart Medialab\Mosart Server\Sql\Db00Create.sql
9. Check setting in the Import tab have the following values:
   a. Character set of the file: utf8
   b. Format of imported file: SQL
   c. SQL compatibility mode: NONE
   d. Do not use AUTO_INCREMENT for zero values: Checked.
10. Click the Go button
    Wait for phpMyAdmin to finish.
    If either the Template sharing or the Media Router part of the database had been installed earlier, some parts of the script will generate error messages, these should be ignored.
11. Click the Structure tab
12. Verify both de_device and em_enumerationmember tables are present in the table list.
13. Click the Privileges tab, the exact name may vary on language, a list of users/hosts should appear.
14. Make sure there is at least one suitable user, assigned to mosarttemplatedb, with the following privileges:
    - DELETE
    - INSERT

- SELECT
- UPDATE
  Users may be created by clicking ’Add new user’, please consult the MySQL and/or phpMyAdmin documentation for more information

12.5 Upgrading mosarttemplatedb Manually

The task of upgrading the database consists of running change scripts that have been added since the last upgrade (or installation).

When doing this manually, the installer should keep track of which scripts that have been previously run.

Care has been taken to construct the scripts in such a way if the installer accidentally runs the same script twice, it either has no effect or produces an error message.

To upgrade the Viz Mosart Database, perform the following:

1. Make sure mosarttemplatedb has been selected
   This is done by clicking it in the Databases tab
2. Decide which script is the first that has not been run on a previous installation or upgrade.
   If in doubt, it is better to run a script that has been run before, than to skip a script
3. For each script, for example Db04Change….sql and above, complete the following steps:
   a. Choose the Import tab
   b. Click the Browse button
   c. In the dialogue box, choose the script you wish to run, for example: %ProgramFiles%\Mosart Medialab\Mosart Server\Sql\Db04Change….sql
   d. Make sure the other settings in the Import tab have the following values:
- Character set of the file: utf8
- Format of imported file: SQL
- SQL compatibility mode: NONE
- Do not use AUTO_INCREMENT for zero values: Checked.
   a. Click the Go button

12.6 Change Scripts

The following table shows the existing change scripts.

Script	Former name	Effect
Db01ChangeGa_avconfigxml.sql	Template01Change…	Adds a ga_avconfigxml column to the ga_gallery table
Db02ChangeSHARED.sql	Template02Change…	Adds a ga_gallery row with- ga_recno = 1- ga_shared = 1- ga_name = SHARED(See Notes below)
Db03ChangeGLOBAL.sql	Template03Change…	

Script	Former name	Effect
		Adds a ts_templateset row with:- ts_recno = 1- ts_ga_recno = 1- ts_name = GLOBAL- ts_global = 1 (See Notes below)
Db04ChangeGa_ts_recno.sql	Template04Change…	Adds a ga_ts_recno column to the ga_gallery table
Db05ChangeTs_ga_updating.sql	Template05Change…	Adds a ts_ga_updating column to the ts_templateset table
Db06ChangeRo.sql	MR01Change…	Initializes the ro_role table by inserting 2 rows
Db07ChangeMe.sql	MR02Change…	Initializes the me_message table by inserting 10 rows
Db08ChangeDtt.sql	MR03Change…	Initializes the dtt_devicetypetype, dt_devicetype, pr_protocol, de_device, and po_port tables by inserting 3, 2, 20, 3, and 3 rows, respectively
Db09ChangeTe_name.sql	Template06…	Adds a te_name column to the te_template table
Db10ChangeUs.sql	MR04Change…	Initializes the us_user table by inserting 1 row
Db11ChangePo.sql	MR05Change…	Removes the uniqueness constraint on the row combination (po_de_recno, po_devicename)
Db12ChangeMe.sql	MR06Change…	Makes me_recno primary key of me_message
Db13ChangePoCrosspoint.sql		Adds column po_crosspoint

12.6.1 Notes

Db02ChangeSHARED.sql

If the ga_gallery table already has a row with ga_shared = 1, but some other ga_name than SHARED, the database should be created from scratch (by first dropping it and then creating it according to MySQL Security).

If the ga_gallery table already has a row with ga_shared = 1 and ga_name = SHARED, this script should be skipped. (Running it will produce an error message.)

If the ga_gallery table already has a row with ga_recno = 1, the script should be changed to give ga_recno some unused value.

Db03ChangeGLOBAL.sql

If the shared gallery (i.e., the ga_gallery row with ga_shared = 1) has ga_recno ? 1, the script should be changed to give ts_ga_recno this value.

If the ts_templateset table already has a row with ts_global = 1, but some other ts_ga_recno than the ga_recno of the shared gallery, the database should be created from scratch (by first dropping it and then creating it according to Creating the Database Manually).

If the ts_templateset table already has a row with ts_global = 1, (and ts_ga_recno equal to the ga_recno of the shared gallery), this script should be skipped. (Running it may or may not produce an error message.)

12.7 Configuring Template Database

This section describes the configuration parameters needed to establish connection to the database within Viz Mosart applications that rely on the MySQL backend.

This section contains the following topics:

- Template Database Settings in Manus Administrator
- Template Database Settings in AV Automation
- Template Database and ActiveX

See also:

- Template Editor Password

12.7.1 Template Database Settings in Manus Administrator

The following settings are required for Template Database:

- TemplateDbConnectionString
- TemplateDbDefaultInserter
- TemplateDbProviderName
- TemplateToMosObject
- UseTemplateDb

For details see Settings Editor - iNews and Settings Editor - MOS.

12.7.2 Template Database Settings in AV Automation

Configuration via AV Automation

Connection to the Template Database is via a connection string in AV Automation.

Template Database configuration is currently only available as part of the general configuration parameters within AV Automation and is accessed through the AV Automation Settings XML Editor.

These parameters are:

- **UseTemplateDb**: Default: False. When True enabling the Template DB functionality, cf. below.

- **TemplateDbProviderName**: The provider name for the Template DB. E.g. `MySql.Data.MySqlClient` for MySQL.
- **TemplateDbConnectionString**: The connection string for the Template DB.
  For example, for MySQL: `server=<hostname>;User Id=<user>;Password=<password>;database=mosarttemplatedb`.
- **TemplateDbDefaultInserter**: The name to be used for the _…*insertedby* and _…*updatedby* columns
- **TemplateDbEnableAutoSynchronize**: (boolean) If true, ensures that Mosart Servers targeting the same gallery (*ThisGallery*) are synchronized automatically. If changes are done to a template on one Mosart Server it will automatically be obtained by the other Mosart Servers. This is typically used by Mosart main/backup servers. Recommended value: true.
- **TemplateDbEnableLocking**: If true, ensures that only one Mosart Server gets write access to a template in the template database at a time. That is, it prevents two users from editing the same template at the same time. Recommended value: true.
- **TemplateDbPollIntervalSeconds**: Specifies how often the Mosart Server will poll the database for updates. Used by the synchronization mechanism to check whether a synchronization operation is necessary. If set to 0, the default value of 5 seconds is used. Recommended value: 5 seconds.
- **SharedGallery**: Default: SHARED. This setting SHOULD NOT be changed from its default value, and may indeed not be changed using the Settings dialogue. (It may be inspected, though.) However, it MAY be changed by changing the MMAVAutomation.exe.config file, but this is not recommended.
  SharedGallery is the DB name of the (fictitious) gallery containing the shared template sets (containing the shared templates). If Viz Mosart has been installed to use the Template DB functionality, such a gallery has been inserted into the DB, along with a shared template set ’ GLOBAL’. (The name of this template set MAY be changed in the Template editor.)
- **ThisGallery**: Default: NONSHARED. The name of the DB gallery representing the gallery of this Viz Mosart installation. (This DB gallery may not exist, but will be inserted when saving in A/V Setup or saving to DB in the Template editor.) Main and backup Viz Mosart installations for the same gallery should have the same value for *ThisGallery*. Installations for different galleries should have different values.

AV Automation Actions

When connected to the Template Database, different actions occur at the following times in AV Automation.

On launch of AV Automation:

- avconfig.xml is replaced by the value tied to ThisGallery
  - If ThisGallery does not yet exist, any existing avconfig.xml is used as is, facilitating bringing an existing setup into Template Sharing.
- channeltemplates.xml is replaced
  - Any shared template sets are written as channels elements to the new channeltemplates.xml with a gallery attribute of SharedGallery.
  - Any template sets tied to ThisGallery are written as channels elements to the new channeltemplates.xml with a gallery attribute of ThisGallery.

    - If ThisGallery does not yet exist, any channels elements in the old channeltemplates.xml, except those with a gallery attribute of SharedGallery, are transferred as is to the new one. This avoids naming conflicts and facilitates bringing an existing set-up into Template Sharing.
      When saving in A/V Setup:

- If ThisGallery does not exist in the database, it is inserted
- The new contents of avconfig.xml is stored in the database tied to ThisGallery
- The contents of selected enumerations are stored separately in the database.
    - These are used when storing channeltemplates.xml in the database.
    - New or changed A/V Setup should always be saved before saving to database in the Template editor
      When saving to Template Database in the Template Editor:

- If ThisGallery does not exist in the database, it is inserted.
- The new contents of channeltemplates.xml is stored in database
  Channel elements, Template Sets, with a gallery attribute of SharedGallery are stored as shared Template Sets - for example, tied to SharedGallery. Each template in these sets will have
    - A template description tied to SharedGallery
    - A template description tied to ThisGallery, the two descriptions will be equal
    - A template implementation tied to ThisGallery.
      Any existing descriptions and implementations tied to other galleries are retained.
- Other channels elements, including any new template sets are tied to ThisGallery. Each template in these sets will have:
    - A template description tied to ThisGallery
    - A template implementation tied to ThisGallery

12.7.3 Template Database and ActiveX

The ActiveX is used to insert Viz Mosart template information into the Newsroom System 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, via the Template Database or directly to the Viz Mosart Server.

See Connecting the ActiveX to Viz Mosart Server.

12.8 MySQL mosarttemplatedb Backup

Whenever a configuration change is made, Viz Mosart advises to backup mosarttemplatedb before making any changes.

This is to ensure the database doesn’t have to be unnecessarily rebuilt due to a misconfiguration.

There are built in protection mechanisms inside mosarttemplatedb, but it never hurts to have a backup.

For more information on backing up the database, please refer to documentation for your specific MySQL instance.

13 MOS-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
```

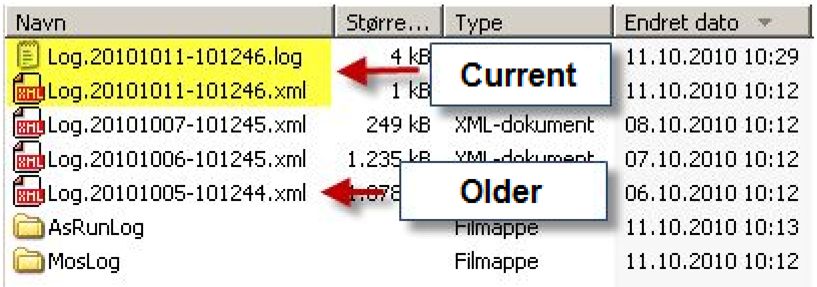
Where:

- `{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 and one .log
  - 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	Større...	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 identity 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:
  - Turn 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”` will ensure 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

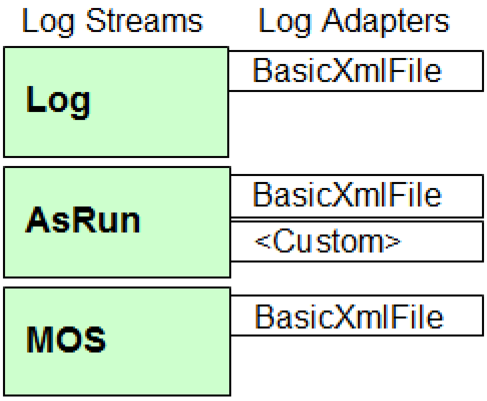
> **Note:** Customer specific AsRun Log Adapters have their own syntax.

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- process 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.

This section contains the following topics:

- Overview
- Example (AsRun log)
- Configuration of Log Adapters
- Configuration of Log Filters

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“ will change 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 will only return 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 will be ignored.
  - Log: The event will be 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 suita 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:\MMLOGS\AsRunLog\xxx“ />
<item name=“FilePrefix“ value=“UPD“ />
<!-- Set to true to use local time codes instead of UTC -->
<item name=“UseLocalTime“ value=“True“ />
<!-- Clean up command for files. Set to 0 if no cleanup -->
<item name=“MaxDaysInRepository“ value=“0“ />
</Properties>
</LogRepositoryConfig>
```

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:\MMLOGS\AsRunLog\xxx</Directory>
<Extension>xml</Extension>
<DayOfWeek>Dim,Lun,Mar,Mer,Jeu,Ven,Sam</DayOfWeek>
<!--<DayOfWeek>Dimanche,Lundi,Mardi,Mercredi,Jeudi,Vendredi,Samedi<
/DayOfWeek> -->
<NoLoggingDuringRehearsal>true</NoLoggingDuringRehearsal>  <!-- If fa
lse, 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 as shown in the table below:

- **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
- **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
- 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
`C:\Program Files (x86)\Mosart Medialab`	Mosart applications. Various sub-directories for components such as Mosart GUI, Mosart Server and more. Either `\Program Files` or `\Program Files (x86)`
`C:\channelTemplates`	`XML-files defining the channel templates. Will normally contain minimum channeltemplates.xml, newsroomsettings.xml, AVConfig.xml`
`C:\manus`	The Run-down (playlist) files fetched from the Newsroom systems.
`C:\MMLogs`	Default location for Log-files. Rotated.
`C:\ProgramData\Mosart Medialab\ConfigurationFiles`	System-wide configuration settings (XML)
`C:\Users\<USERNAMEHERE>\AppData\Local\Mosart_Medialab`	User-specific application configuration files

Note: The list above is an example and may differ for various installations and Windows versions. Refer to the corresponding succeeding subsections 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 will not overwrite any the content mentioned below but new configuration parameters will be 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 will 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 will 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.

Alternatively, you can use the *Backup Files* function in the Viz Mosart Installation Administrator.

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 will grow unless a maintenance procedure is established. This can be configured in the Manus Administrator Configuration 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 now and then. 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 PC's and Servers at least once a month.

Should any ill effects occur periodically 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

The Viz Mosart Applications are regularly updated, and Vizrt strongly recommends to upgrade to the latest versions whenever these are released. This is easily done using the Viz Mosart Installation Administrator.

Fallback to previous versions is equally fast, facilitating - if necessary - rapid test runs in between other transmission schedules.

As long as your operation is running to your expectations - not hampered by the causes for our published fixes, or you do not need the new functionality being introduced now and then, and the servers are running in network isolation - there is no absolute need to do an upgrade of the application.

13.3.5 Windows Updates

The interval for running Windows updates depends on the environment. If the PCs 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.

Note: There might be a ~30 second delay when starting AV Automation. The cause of this phenomenon, which typically happens just after some configuration, is the need for a Windows Update to be run. Consult the proper Microsoft procedure to update Windows via downloaded files if it is 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.

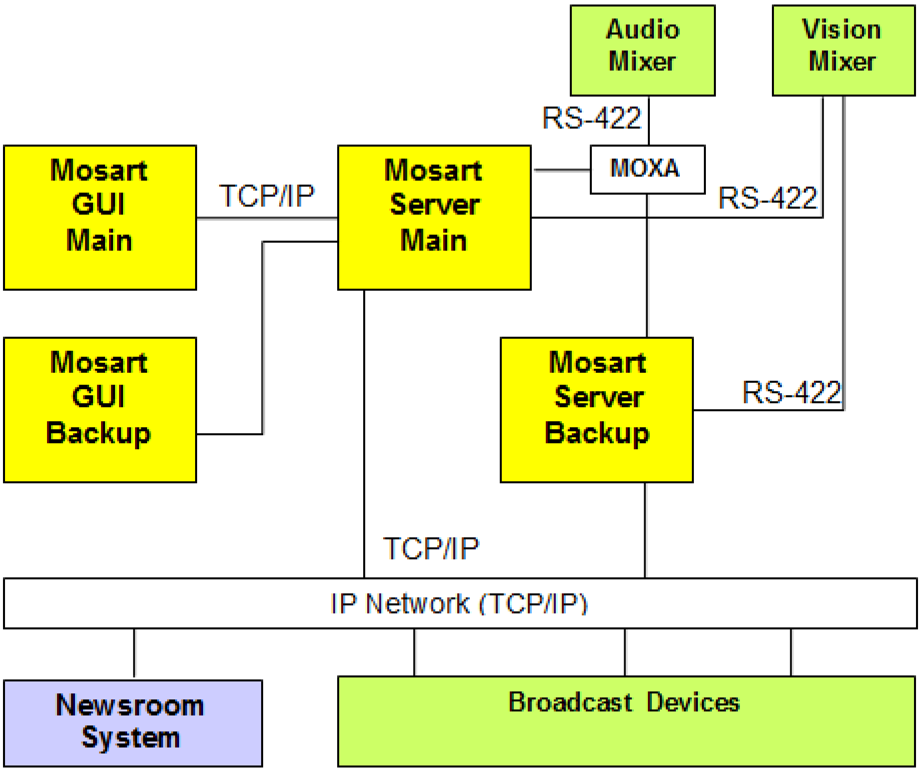
13.4 Redundancy

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.

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.

An 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.

See Also

- Media Sequencer Redundancy

14 General Configuration Files

This section contains:

- NamedOverlayGraphics.xml

14.1 NamedOverlayGraphics.xml

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

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” will take 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=“2
5” 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=“2
5” 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¦regex” />
     <action name=”clear” />
   </actions>
   <content>
   </content>
</item>
```

The following actions are available:

- takeout – Will take out any stored onair 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
  - regex – Takes all graphics with slug matching the given regular expression. I.e. regex = “LIVE.*” will take out all graphics with slug starting with “LIVE”. Regular expressions follows the Microsoft .Net regular expression language.
- clear – Will takeout all onair graphics

Note: Named CG actions will 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_NWATVNT1;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" stati
c="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" defa
ult="00:00" value="00:20" />
     <field name="continuecount" value="-1" fieldtype="TEXT" />
     <field name="tc_in" fieldtype="TIMECODE" inputmask="mm:ss" defau
lt="00:00" value="00:00" />
   </fields>
</item>
```

The required fields are `slug, templatetype, in, dur, use_graphics_id, graphics_id, handler_name, graphics_out_on, description`.

Extracted XML example

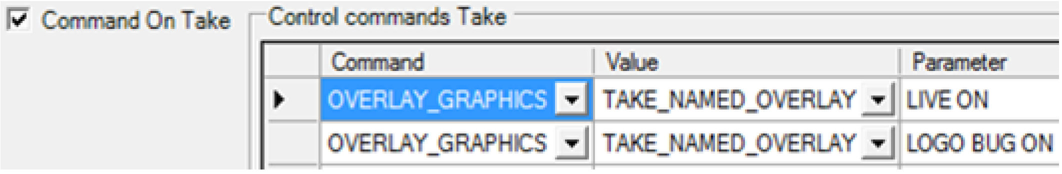
The required fields extracted will look like this, please note that the slug and description has changed:

```
<items>
<!--LIVE ON -->
    <item slug="LIVE ON" in="0" dur="125" use_graphics_id="true"
graphics_id="1863809" handler_name="DSK" graphics_out_on="BACKGROUNDE
ND" 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

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 will 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 and MAM Connection Strings
- Video Wall Connection Strings

15.1 Video Server and MAM Connection Strings

A connection string is used to connect to a particular device. It consists of a series of keywords, in this instance, values with device specific commands. These values are usually in pairs, with each pair separated by a semicolon.

In Viz Mosart, connection strings are used in Media Administrator configuration and, in some cases, AV Automation. The AV Automation connection strings should be auto-generated when configured in the Video server GUI.

Configuration file names with no path given as value to the Config parameter below will search 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 contains the following Connection String types:

- Standard Video Server Connection String
- General Configuration File Properties
- General Search Configuration File Properties
- 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
- SQL database connection string

15.1.1 Standard Video Server Connection String

Viz Mosart Video Server Connection String

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

- **Name:** Specifies a logical name to be associated with the device. This is used to recognize the devices in various places.
- **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.

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

15.1.2 General Configuration File Properties

- **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 will result in a clip refresh operation (Default=False).
- **ReleasePortOnDispose (boolean):** If true, the assigned video port will be released when server connection is closed (Default=True).
- **SupportSearchUsingObjSlug (boolean):** Activates searching using objSlug (Default=True).
- **SupportSearchUsingRefID (boolean):** Activates searching using RefID (Default=False).
- **RefreshModifiedClips (boolean):** Deprecated (Default=True).
- **VerifyModifiedClips (boolean):** If true, clip modified/update events for all clips will trigger a complete verification (for example treated as the clip was created). If false, only clips currently in the cache will be verified (Default=True).
- **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 msecs
- **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.
- **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 will be MyGallery_MyTemplateSet_MyClipName_160921.

15.1.3 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 will result in a exact case sensitive search.
    The following example sets up a regular expression filter searching for

    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>
```

15.1.4 AirSpeed Classic Connection String

Applies to Models

- AirSpeed
- AirSpace

See Also

- AirSpeed Multi Stream / AirSpeed 5000 Connection String

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).
  * 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 will be 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 will be retrieved for clips in rundown only (Default=True).
- **DelayEventsOnDatabaseInitialization (boolean):** If true, all clip handling during database load will be 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 will be verified, as if they were created (Default=True).
- **ReceiveBufferSize (integer):** Set buffer size used for receiving date from AMS. Default 16384 = 2 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 will store 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.

15.1.5 AirSpeed Multi Stream / AirSpeed 5000 Connection String

The AirSpeed Multi Stream driver supports up to 6 multiple player ports per connection.

Applies to Models

- AirSpeed Multi Stream 1.8
- AirSpeed 5000

See Also

- AirSpeed Classic Connection String

Connection String Properties

- **Type:** AirSpeedMultiStream
- **Name:** Defines the ID of this connection (Default=AirSpeedMultistream).
- **Server:** Defines the hostname or IP-address of the server.
- **Config:** Defines the name of the configuration file.
  Optional:

- **BasePort**:* Specifies the port number of the first port on the server (Default=59451).
- **Player**:* Specifies the player (1-6) that should be used for searches and clip verification (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).

  *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:

```
Type=AMS; Name=AirSpeedMultiStream; Server=10.211.114.104; BasePort:5
9451; Player=2; ChangeCheckInteral=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 will be 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 how 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. Info will be retrieved for clips in rundown only (Default=True).
- **DelayEventsOnDatabaseInitialization (boolean):** If true, all clip handling during database load will be 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 will be verified, as if they were created (Default=True).
- **ReceiveBufferSize (integer):** Set buffer size used for receiving date from AMS. Default 16384 = 2 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 will store the content in this file (Default=False).
- **DatabaseCacheUpdate (boolean):** Determines when the debug database cache is updated (either [Never,Always,Initial]).

15.1.6 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 will be used when getting clip info (Default=””). For more details, see the Amadeus documentation.
- **LowerCaseRundowns:** If true, then the rundown name will always be generated with lower case (Default=False).
- **IgnoreSTEG**: If true, all messages from Amadeus starting with STEG will be 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).

15.1.7 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 5 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).

  * 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 will be 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

15.1.8 JupiterWebService Connection String

JupiterWebService is BBC’s internally developed MAM system.

```
Name=myServerName; Url=myConfigFile
```

Example:

```
Name=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}

15.1.9 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

  *About 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=MVCP; Server=192.165.4.1; Port=5251
```

15.1.10 Omnibus OPUS Connection String

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

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

15.1.11 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.xm
```

Quantel Configuration File Properties

- **DefaultTakenPortAction (enum):** Action when not able to take a port when owned by someone else (Default=Steal). Options:
  - **Steal:** Takes control over the port even if in use by someone else.
  - **Share:** Shares control with the other user. Not recommended.
  - **Cancel:** Cancels the operation. Port will still be in control by other user.
- **IssueRequestForTakenPorts (boolean):** If true, the user will be prompted to confirm taking a port in use by someone else (Default=True). A dialog box will appear 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) will be released on player initiation (Default=True).
- **ServerPollFrequency (integer) :** Polling frequency for server, in seconds.This is used as heartbeat between Viz Mosart and Quantel. Note that heartbeats are not issued if data has been received from Quantel since last heartbeat. (Default=30).
- **ClipNameIsValidNumber (boolean):** If true then it is possible to search from Viz Mosart GUI using valid Quantel ClipIDs directly (Default=False).
- **QuantelSlugColumn (string):** Specifies the Quantel database column that is used for the clip slug (Default=Title).
- **QuantelClipIDColumn (string):** Specifies the Quantel database column that is used for the clip ID (Default=ClipID).
- **QuantelSearchColumn (string):** Specifies the Quantel database column that is used for the clip searches (Default=Title).
- **MaxPingAttempts (integer):** Maximum ping attempts before a lost connection is detected (Default=1). When Viz Mosart detects a lost connection, the current connection will be 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)**: If true, will force AV Automation to only search for clips using Quantel ClipId’s obtained from Media Administrator.
- **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 will only be 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 therefore 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="fa
lse" />
    <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” is ignored
- Only clips in the Quantel database with a PoolID that matches the pool identity of the corresponding video servers is returned
- All searches are case insensitive

15.1.12 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:** Optional. If set, performs searches associated with directory for the given player. If not set, the Directory option should be used to identify the search directory.
- **Directory:** Optional. Specifies the search directory
- **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 will return full clip paths. The clip ID will contain 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 will be 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 will be verified (as if they were created) (Default=False).
- **VerifyModifiedClips (boolean):** If true, all modified clip events for clip not in cache will be 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 will use 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 will be stopped, not paused (play with speed 0) when cuing a new clip. This will cause cuing to take longer than usual.
- **ClipNamePattern**: see General Configuration File Properties

  Search Properties: See General Search Configuration File Properties

15.1.13 Grass Valley K2 Connection String

Applies to Models

- K2
- Summit

Device Prerequisites

- The file *ClipServerK2.xml* must be placed in the *channeltemplates* folder on the Viz Mosart server.

Connection String Properties

- **Name:** Not important, used for logging purposes only.
- **Type:** Selects the Grass Valley K2 driver for clip handling.
- **Server:** Defines the IP address/hostname of the Grass Valley K2 server.
- **Domain:** Sets the domain as part of the user credentials.
- **User:** Sets the username as part of the user credentials.
- **Pass:** Sets the password as part of the user credentials.
- **Suite:** Defines the dedicated Grass Valley suite (Default=Mosart). This must be unique.
- **Volume:** Sets the volume of the file repository on the K2 (Default=V:). Should be “c:” when using an emulator.
- **Bin:** Sets the clip bin in the file repository on the K2 (Default=default).

Examples

AV Automation:

```
name=K2Server; type=GrassValleyK2; server=10.64.150.57;
config=ClipServerK2.xml; user=<user>;pass=<password>; suite=Mosart;
volume=V:; bin=default
```

Media Administrator:

```
Name=Summit; Type=GrassValleyK2; Server=K2SERVER1;
Config=ClipServerK2.xml; Bin=default; User=GVAdmin; Pass=GVPass;
Volume=V:; Suite=GVAdmin; Domain=K2SERVER1;LoadDatabase=true
```

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 will be updated on each update clip message from the server (Default=True).
- **VerifyClipDelay (integer):** Delay in frames to wait before verifying new clips from server.
- **VerifyModifiedClips (boolean):** If true, all modified clip events for clip not in cache will be verified (as if they were created) (Default=False).
- **DefaultTakenPortAction (Steal,Share,Cancel):** If something other than Viz Mosart has taken the required channel, this determines what Viz Mosart should do.
- **RefreshModifiedClips (boolean):** If true, all modified clip events for clip in cache will be refreshed (for example deleted and reinserted).
- **VerifyModifiedClips (boolean):** If true, all modified clip events for clip not in cache will be verified (as if they were created).

15.1.14 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=tr
ue; 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 will be 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 2) (Default=8).
- **LocalMtpcIpAddress:** Necessary only if the Viz Mosart server has several network adapters, e.g. a virtual adapter (or several virtual adapters) in addition to the physical adapter, e.g. when VMWare (or similar) is installed on a demo PC, or when connecting through VPN. In this case, use the IP address (in the usual format of four dot(.)-separated groups of 1-3 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* will be produced.) The options may be combined. For example, 0xFFFFFFFF will produce *all* log files.

Bitmask	**Description**	**Log file, given LogOutput=... \linx.log**
0x00000003	Function entry and exit points, with parameters and return code	linx.log
0x00000010	Connections	linxConnect.log
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).

15.1.15 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 that the parameters mentioned so far are insufficient to connect to a playout server. For this, the IP address and port number are required. These are specified (:-separated) 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 lo-res clip (Default=LORES-).
- **SkipGetDuration:** Whether to skip getting clip duration. Note that this operation may be slow and MAY be rendered unnecessary by future Media Administrator dev (Default=False).
- **User:** User for the CleanEdit DB.

Debug Parameters, Relevant for Media Administrator Only

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

15.1.16 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**

15.1.17 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:** Defines the server name (Default=Orad OCIP). Used for logging purposes only.
- **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

- **LoopPostRoll:** Number of frames before clip-end to restart loop. Should probably be set to a positive value to obtain smooth looping.
- **SkipAbortInLoadClip:** If true, skip the ChannelAbort command at cue (Default=False).
- **TimeoutDefault:** Timeout in milliseconds for commands with no specific timeout set (Default=1000 (1 second)).
- **TimeoutChannelAbort:** Timeout in milliseconds for the ChannelAbort command performed at cue.
- **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)).
- **UsePingOcipOperation (boolean):** If true, use an OCIP operation for Ping instead of the default TcpIpClient ping (Default=False).

Debug Parameters

- **WSACancelBlockingCallExceptionTraceEventType:** The TraceEventType for the WSACancelBlockingCall exception. Options: Critical, Error, Warning, Information, Verbose (Default=Error).

Applies to Models

- Orad VJ

Device Prerequisites

- OCIP 1.0.4

15.1.18 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 8 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 will be 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 will be 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 8 character files names padded with spaces if less than 8 characters.
- **UseAddedClipsCommand (boolean):** Enable to retrieve a list of new files added to the video server.
- **NextPingDelay (miliseconds)**: 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

15.1.19 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 will 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 will impact the startup time of the SimpleTestPlayer driver in Media Administrator for folders with a large number of video files.

15.1.20 SQL database connection string

This connection string (in MediaAdmin) is for the SqlMediaSearch driver, 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 pertaining to 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 found on a video server will be blue in the 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 been tested for Microsoft SQL Server only. The following versions have been tested:

- 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 malicious or careless users from changing the DB.)

General format

```
Name=DB;Type=SqlMediaSearch;ProviderName=System.Data.SqlClient;Config=C
```

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 tested.

- **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 provider tested) the format is documented in https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlconnection.connectionstring(v=vs.110).aspx. Quotes in the value have not been tested and may not work as expected.
- **FileClipProperties:** Same as the ClipServerSql.xml parameter; see below. Should be set in ClipServerSql.xml.

- **FileSelectStatements:** Same as the ClipServerSql.xml parameter; see below. Should be set in ClipServerSql.xml.
- **LogResult:** Same as the ClipServerSql.xml parameter; see below. Should be set in ClipServerSql. xml.
- **DefaultTableName:** Same as the ClipServerSql.xml parameter; see below. Should be set in ClipServerSql.xml.
- **MonitorFoundClips:** Same as the ClipServerSql.xml parameter; see below. Should be set in ClipServerSql.xml.
- **MonitorClipInterval:** Same as the ClipServerSql.xml parameter; see below. Should be set in ClipServerSql.xml.

Example

```
Name=DB;Type=SqlMediaSearch;ProviderName=System.Data.SqlClient;Config=C
```

Configuration files

There are *three* configuration files:

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

When Mosart Server is installed, these files are placed in the ConfigurationFiles sub-folder of the installation folder. If any of them needs to be changed, follow standard procedure: Copy the file to c:\channeltemplates, and *change the copy* . If you choose to change the name, you must also change (accordingly):

- If the name of ClipServerSql.xml is changed: The Config connection string parameter
- If the name of SelectStatements.xml is changed: The FileSelectStatements parameter in ClipServerSql.xml
- If the name of ClipProperties.xml is changed: The FileC lipProperties parameter in ClipServerSql. xml .

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. See below.
- **FileSelectStatements:** The name of the file containing the SELECT statements to be used. Default SelectStatements.xml. See below.
- **LogResult:** Whether query results should be logged (in MediaAdmin console and in Viz Mosart log). Default true. Set to false if everything works fine, and the messages are annoying.
- **DefaultTableName:** The name of the outer XML element shown when using LogResult. The default ('dummy') is probably OK.
- **MonitorFoundClips:** Whether found clips should be monitored. Default true. If set to false, missing or changed clips will *not* be detected.

- **MonitorClipInterval:** If the value is X (and if MonitorFoundClips is true), each found clip is requeried every X seconds. Increase if DB traffic is too high.
- **DbConnectionString:** Same as the connection string parameter; see above. Should be set in the connection string.
- **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:** Ignores search for objSlug if set to false (Default=True).
- **SupportSearchUsingRefID:** Ignores search for refID if set to false (Default=False)

SelectStatements.xml

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

If you find the description here somewhat abstract, please study the example(s) given 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 *constant* values

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

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

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

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

Attribute	Value(s)	Description	Comment
name		The DB column name	
type	String/Int	The DB column data type. For Microsoft SQL Server, • String is used for char and varchar columns.	Only String has been tested.

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

ClipProperties.xml

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

If you find the description here somewhat abstract, please study the example(s) given 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	Only these have been implemented: Description, InPoint, ObjSlug, OutPoint

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.	
DbExpression	The value of a DbExpression depends only on the value of a column in the DB row pertaining to the clip (the *clip row*).	
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 , i.e., applying an *operator* to the values of its *operands* (sub-expressions). Each operand / sub-expression may be of any of the four kinds of expression listed in this table. In particular, OperationExpressions may be nested to any depth.	

A ConstantExpression element has one attribute:

Attribute	Description
value	The (constant) value of the ConstantExpression

Both a ClipExpression and a DbExpression has two attributes:

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

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

Expression kind	Description of the name attribute	Expression value	Comment
ClipExpression	The name of a clip property	The value of the property for the given clip	Only BasePoint 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 must be described by a SelectColumn element in the SelectStatements file.

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, i.e., any number of ClipExpression, DbExpression, ConstantExpression, and OperationExpression sub-elements. These sub-elements are the operands / sub-expressions of the OperationExpression.

As promised, we will give some further details on each operator:

Operator	Expression value
Minus	If there is 1 operand, the negation of the value of that operand. If there are 2 (or more) operands, the difference between the values of the first operand (minuend) and the second operand (subtrahend).
Plus	For any number of operands, the sum of the values of the operands. (As a special case, the ‘sum’ of no operands is 0. However, the Operands sub-element itself must be present. Anyway, a `<ConstantExpression value="0" />` would be more convenient.)

15.2 Video Wall Connection Strings

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

- Watchout Connection String

Note: For general information on connection strings, see Video Server and MAM 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

- **AuthenticateTimeout**: Integer>0. The time (in ms) allowed for WATCHOUT to process an authenticate command and respond with Ready. Default 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. Default 1024.
- **BusyNonCommandTimeout**: Integer>0. The time (in ms) allowed for WATCHOUT to return to normal operation after a non-command Busy. Default 10000.
- **BusyTimeout**: Integer>0. The time (in ms) allowed for WATCHOUT to increase the % of otherwise equivalent Busy responses. Default 60000.
- **GetStatusThreshold**: Integer. The highest acceptable value of the 3rd parameter of the Reply feedback to the getStatus command (General health status of the cluster; 0: OK, 1: Suboptimal, 2: Problems, 3: Dead). Default 1 (Suboptimal).
- **LoadTimeout**: Integer>0. The time (in ms) allowed for WATCHOUT to process a load command and respond with Ready. Default 10000.
- **PingInterval**: Integer>0. The time (in ms) between ping commands sent to WATCHOUT. Default 10000.
- **PingTimeout**: Integer>0. The time (in ms) allowed for WATCHOUT to process a ping command and respond with Ready. Default 10000.
- **Port**: Integer>0. The port to connect to. Default 3039.
- **ResetAnyway**: true/false. Whether reset commands are always sent before run commands even if not present in the RecallAndMix argument. Default false.
- **ResetTimeout**: Integer>0. The time (in ms) allowed for WATCHOUT to process a reset command and respond with Ready. Default 10000.
- **RunTimeout**: Integer>0. The time (in ms) allowed for WATCHOUT to process a run command and respond with Ready. Default 10000.
- **ReconnectInterval**: Integer>0. The amount of time (in ms) to wait to reconnect after a failed connect or authenticate. Default 10000.
- **SequenceIndividualCommands**: true/false. Whether the driver should operate in that mode. Has no effect when SequenceRecallAndMixes=true. Default false.
- **SequenceRecallAndMixes**: true/false. Whether the driver should operate in that mode. Default false.
- **Server**: String. The DNS name of WATCHOUT. Default localhost.
- **UseGetStatus**: true/false. Whether the Ping operation should use the getStatus command instead of the ping command. Default false.

16 Device Configuration Files

The device configuration files are located in the following directory:

```
%ProgramFiles%\Mosart Medialab\Mosart Server\ConfigurationFiles
```

Note: Do not modify any files at this location, as they will be 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 up- 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's 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 the following topic is found in the previous section (Device Connection Strings):

- Video Server Configuration Files

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: In order 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’s 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*.

```
<?xml version="1.0" encoding="UTF-8"?>
<DeviceConfig name="TechnodollyConfiguration" connectionString="local
host">
   <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 1 prepare or 1 run. A value of 2 means 1 prepare and/or 1 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's Guide*.

Example configuration:

```
{{
<?xml version="1.0" encoding="UTF\-8"?>}}{{
<DeviceConfig name="ElectricFriendsConfiguration" connectionString="l
ocalhost">}} {{
    <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="localh
ost: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 roId of the commands. (The same as set in the mosconfig.xml file in the Avid MOS Gateway.
- **ConvertStoryIdToDecimal**: Specifies whether the story id should be converted from Hex to Decimal value.
- **ConnectedWhenIdle**: This value determines if the connection to the graphic devices are maintained while Viz Mosart Server is in Idle mode.
  True = Maintain connection to graphic devices while in Idle mode. This setting will also 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="localh
ost: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

This section contains the Subtitling types:

- ScreenLL Configuration File

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 fo
r to decide clip availability -->     <item name="AvailabilityLanguag
eCode" 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 Viz Mosart support use and will not be further described in this document.

16.4 Video Router Configuration Files

This section contains the Video Router types:

- Miranda NVision Configuration File

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 ms) 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:** In order 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

This section contains the Audio Mixer types:

- Calrec TCP/IP Configuration File
- SSL Configuration File

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 genral 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

.