Viz Channel Branding User Guide

Version 5.2

Copyright ©2024 Vizrt. All rights reserved.

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

Disclaimer

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

Antivirus

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

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

Technical Support

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

Created on

2024/06/05

Contents

1 Introduction 12
1.1 Introducing the Viz Channel Branding Workflow 12
1.2 Key Benefits 12
1.3 All-in-One Workflow 12
1.4 Graphics Workflow 13
1.4.1 Upstream 14
1.4.2 Downstream 14
1.4.3 Transition Logic 15
1.5 Related Documents 15
1.5.1 See also 16
1.6 Feedback 16
2 Requirements 17
2.1 General 17
2.2 Hardware 17
2.3 Software 18
2.4 Viz Channel Branding 18
2.5 Integration with Viz One 19
2.6 Network Ports and Connections Ports and Connections 19
3 Setup and Configuration 21
3.1 Installation 22
3.1.1 Bundle Installer 22
3.1.2 Unbundling the Bundle Installer 22
3.1.3 Installing Viz Channel Branding Client 22
3.1.4 Time Format and Time Zones 23
3.2 Playout Configuration 24
3.2.1 Playlist Importer 24
3.2.2 Client Placement 24
3.2.3 Installation of Viz Engine Machines for Playout 24
3.2.4 Media Sequencer 25
3.2.5 Optional Startup Parameter for the Viz Channel Branding Client 25
3.2.6 Configuration of Channels with channels.ini 25
3.2.7 Forked Execution 26
3.2.8 Configuring Device Channels 26

3.2.9 Configuring Playout Devices 29
3.2.10 Logging 33
3.3 Scheduling Configuration 35
3.4 Video Configuration 36
3.4.1 Configure Viz Engine to Pass through Video as DVE 36
3.5 Preview Server 37
3.5.1 Preview Server Web Interface 37
3.5.2 Installing and Configuring Preview Server 39
3.6 Quick Scheduler Server 41
3.6.1 Configuration 41
4 User Interface 43
4.1 Main Menu and Toolbar 45
4.1.1 Main Menu 45
4.1.2 Toolbar 46
4.2 Statistics Panel 48
4.3 Channels 50
4.3.1 Channel Selection 50
4.4 Playlists 51
4.4.1 Import a Channel Branding Exported Playlist 52
4.4.2 Channel Pool 52
4.4.3 Playlist Toolbar 52
4.4.4 Playlist Keyboard and Mouse Shortcuts 54
4.4.5 Playlist Columns 55
4.4.6 Playlist Context Menu 57
4.4.7 Playlist Filters 59
4.5 Templates, Pages and Pilot Data Elements 61
4.5.1 Templates 61
4.5.2 Pages 61
4.5.3 Pilot Data Elements 61
4.5.4 Template Context Menu 62
4.5.5 Pages Context Menu 63
4.5.6 Template and Page Filters 64
4.6 Clips 66
4.6.1 Clips Context Menu 66
4.7 Clip Bank 68
4.8 Actions 69
4.8.1 GPO 69

4.8.2 Viz Ticker and Viz Engine ... 69
4.9 Template Import ... 71
4.10 Rules ... 72
4.11 Page Editor ... 74
4.11.1 Scene Preview Player ... 75
4.11.2 Page Fields List ... 76
4.11.3 Page Fields Editors ... 77
4.11.4 General Actions ... 80
4.12 Timeline Editor ... 82
4.12.1 Add Graphics to the Video Timeline ... 82
4.12.2 Remove Graphics from the Video Timeline ... 82
4.13 Combo Page Editor ... 84
4.13.1 Working with the Combo Page Editor ... 84
5 Settings ... 86
5.1 Global Settings ... 87
5.1.1 General Settings ... 87
5.1.2 Import / Export Settings ... 88
5.1.3 External Devices ... 89
5.1.4 Logging ... 91
5.2 Channel Settings ... 95
5.2.1 Managing Channel Settings ... 96
5.2.2 General Settings ... 98
5.2.3 Schedule Import (Playlist Importer Only) ... 114
5.2.4 External Triggering ... 129
5.2.5 External Data Sources ... 133
5.3 Client Configuration File ... 142
6 Import Guide ... 143
6.1 Basic Mapping Table Field Names ... 144
6.1.1 Required Field Names ... 144
6.1.2 Optional Field Names ... 144
6.1.3 Additional Field Names ... 145
6.2 Importing from Text ... 146
6.2.1 Import Delimited Files ... 146
6.2.2 Import Delimited Files with Header Lines ... 148
6.2.3 Import Fixed Width Files ... 148
6.2.4 Sample Content - Fixed Width (Indexed) ... 149
6.3 Importing from XML ... 150

6.3.1 Map Primary and Secondary Events 150
6.3.2 Secondary Events at the Same Level as Primary Events 150
6.3.3 Secondary Events Nested Inside Primary Events 151
6.3.4 Channel Name and Playlist Date in a Separate Node 151
6.4 Importing from Excel 153
6.5 Harris ADC-100 155
6.5.1 To import LST Format Files 155
6.5.2 To build unique instructions in case of duplicate Reconcile keys in LST 156
6.6 Reading Channel and Date Values from Filenames 157
6.6.1 To read channel and date from the filename 157
6.6.2 Example 157
6.7 Merging Fields 158
6.7.1 Example 1 for Fixed Width Any Text Import Type 158
6.7.2 Example 2 for Delimited Any Text Import Type 158
6.7.3 Example 3 for Any XML Import Type 159
6.8 Splitting Fields 160
6.8.1 Example 160
7 Working with Scheduling 161
7.1 Scheduling System Integrations 162
7.2 Quick Scheduler 163
7.2.1 Configuration 163
7.2.2 Using Quick Scheduler 164
8 Playlist Importer 168
8.1 Configuration 168
8.2 Playlist Importer Log File 168
8.3 Playlist Importer Startup Parameters 168
8.4 Automatic Import of Files 169
8.5 Troubleshooting 169
9 Integrations Hub 170
9.1 Configuration Check 170
9.2 Selecting the Integrations Hub Media Sequencer 171
9.3 REST Interface for Integrations Hub Version 1.0 173
9.3.1 GET All Channels (Version 1.0) 174
9.3.2 DELETE All Channels (Version 1.0) 175
9.3.3 POST Channel (Version 1.0) 176
9.3.4 GET Channel and its Playlists (Version 1.0) 177

9.3.5 DELETE Channel (Version 1.0) ... 179
9.3.6 GET Channel Settings for a Channel (Version 1.0) ... 180
9.3.7 GET Page Pool (Version 1.0) ... 182
9.3.8 POST Pages to the Page Pool (Version 1.0) ... 183
9.3.9 GET Playlist from a Channel (Version 1.0) ... 185
9.3.10 POST Playlist to a Channel (Version 1.0) ... 187
9.3.11 DELETE Playlist (Version 1.0) ... 188
9.3.12 GET Channel Pool from a Channel (Version 1.0) ... 189
9.3.13 POST Group and Page to Channel Pool (Version 1.0) ... 191
9.3.14 GET Group from a Playlist (Version 1.0) ... 194
9.3.15 POST Group to Playlist (Version 1.0) ... 196
9.3.16 DELETE Group (Version 1.0) ... 201
9.3.17 GET Page from a Group (Version 1.0) ... 202
9.3.18 POST Pages to a Group (Version 1.0) ... 204
9.3.19 DELETE Page of a Group (Version 1.0) ... 207
9.3.20 GET All Events with Matching Event Id (Version 1.0) ... 208
9.4 REST Interface for Integrations Hub Version 2.0 ... 210
9.4.1 GET API Root (Version 2.0) ... 211
9.4.2 GET ALL Channels (Version 2.0) ... 212
9.4.3 GET a Channel (Version 2.0) ... 214
9.4.4 Channel Content (Version 2.0) ... 216
10 Page Content Filling ... 340
10.1 Content Source Alternatives ... 340
10.1.1 Manual Input ... 340
10.1.2 Referenced Pages ... 340
10.1.3 Dynamic Content from Playlist Metadata ... 341
10.1.4 Dynamic Content from Database ... 341
10.1.5 Clearing Dynamic Page Content ... 343
10.1.6 Page Content from External Update Services ... 344
10.2 Page Content Filling for Playlist Metadata ... 345
10.3 Translation Tables ... 349
11 Promo Server ... 351
11.1 About Cross-Promotion ... 351
11.2 Configuration ... 351
11.3 Promo Server Logs and Messages ... 351
12 Failover Helper ... 353
12.1 Configuring Failover Helper ... 354

12.1.1 Working with the Failover Helper ... 354
12.2 Monitoring the Failover Helper ... 356
12.2.1 Monitoring Status ... 356
13 XML Translator ... 360
13.1 Using XML Translator ... 360
13.2 External References ... 361
14 Channel Deck ... 362
14.1 Setting up Channel Deck ... 363
14.1.1 Setting up Channel Deck ... 363
14.2 Channel Deck Configuration and Layout ... 365
14.2.1 Settings ... 365
14.2.2 Working with Settings ... 366
14.3 Managing Channel Connections ... 368
14.3.1 Prerequisites ... 368
14.3.2 Working with Channel Connections ... 368
14.4 Monitoring Channel Deck ... 371
14.4.1 Monitoring Takes and Channel Branding ... 371
14.4.2 Monitoring Errors and Warnings ... 373
15 Housekeeper ... 376
15.1 Configuring the Housekeeper Service ... 377
15.1.1 Working with Housekeeper ... 377
16 Traffic Department Workflow ... 387
16.1 Working with Templates and Pages ... 388
16.1.1 About Pages ... 388
16.1.2 Importing Templates ... 388
16.1.3 Creating Pages ... 389
16.1.4 Previewing Pages ... 389
16.1.5 Editing Page Duration ... 389
16.1.6 Sending Templates and Pages ... 390
16.2 Working with Actions ... 391
16.2.1 About Actions ... 391
16.3 Ticker Actions ... 394
16.3.1 Working with Ticker Actions ... 394
16.4 Working with Playlists ... 397
16.4.1 Import a Schedule ... 397
16.4.2 Creating a Playlist ... 398

16.4.3 Deleting a Playlist ... 398
16.4.4 Adding Primary Events to a Playlist ... 399
16.4.5 Adding Secondary Events to a Playlist ... 399
16.4.6 Deleting Events from a Playlist ... 399
16.4.7 Updating a Playlist ... 399
16.4.8 Exporting a Playlist ... 400
16.4.9 Playlist State Definitions ... 400
16.4.10 Playlist Errors ... 402
16.5 Working with Video Clips ... 404
16.5.1 Preview of Video Clips ... 404
16.5.2 Video Clips in Playlists ... 404
16.5.3 Video Clips and Graphics ... 405
16.6 Working with Rules ... 407
16.6.1 The Rules UI ... 407
16.6.2 Defining Rules for Conditions and Actions ... 410
16.6.3 Rules Criteria and Conditions ... 412
16.6.4 Back up and Restoring Rules ... 414
16.7 Resolving Conflicts ... 415
16.8 Generating the Channels File ... 416
16.8.1 To regenerate the channels.ini file ... 416
17 Master Control Workflow ... 417
17.1 Activate the Playlist ... 417
17.1.1 To manually activate a playlist ... 417
17.2 Initialize the Playlist ... 417
17.2.1 To manually initialize a playlist ... 418
17.3 Run the Playlist ... 418
17.3.1 To manually run a playlist ... 418
17.4 Triggers ... 419
17.4.1 Primary Events ... 419
17.4.2 Secondary Events ... 419
17.5 Monitoring ... 419
17.6 Live Schedule Updates ... 420
17.7 Fail-over Procedure ... 420
18 Integration with Automation Systems ... 421
18.1 Automation Systems ... 421
18.2 Automation System Integration types ... 421

18.3 Overview of Direct Integration Architecture ... 423
18.3.1 Integration Module ... 423
18.4 Integration Module for Marina from Pebble Beach ... 424
18.4.1 Configuring the Integration Module for PB Marina ... 424
18.5 Integration Module for Neptune from Pebble Beach ... 428
18.5.1 Configuring the Integration Module for PB Neptune ... 428
18.6 Integration module for Harris ADC ... 432
18.6.1 Configuration and Startup ... 432
18.6.2 Pushing Optional Metadata from Harris to the resulting playlist ... 435
18.7 ADC-100 by Harris Broadcast ... 436
18.7.1 General Information and Setup ... 436
18.7.2 Triggering with VDCP ... 437
18.7.3 Triggering with USD ... 441
18.7.4 Automatic Playlist Updates Using RCV Files ... 448
18.8 D-Series by Harris Broadcast ... 451
18.8.1 General Information and Setup ... 451
18.8.2 Updating Live ... 451
18.8.3 Limitations ... 451
19 Viz One Integration ... 452
19.1 Viz One and Viz Engine Configuration ... 452
19.2 Viz Channel Branding Configuration ... 452
19.3 Video Preview Configuration ... 453
19.4 Installation Options ... 453
19.4.1 To install codecs for local preview ... 454
19.4.2 To set preferred decoder ... 454
20 Appendix ... 455
20.1 Upgrading Viz Channel Branding ... 456
20.1.1 Working with Upgrade Tool ... 456
20.1.2 Upgrading from Legacy Viz Multichannel from 2.6.2+ to 4.x and Migrating to Viz Channel Branding 5.0+ ... 456
20.2 Vizrt Data Format (VDF) ... 457
20.2.1 Example of a VDF document ... 457
20.3 Page Content Filling Aliases ... 458
20.3.1 Page Content Filling Aliases ... 458
20.3.2 Page Content Filling Example ... 459
20.4 Installing and Upgrading the Microsoft .NET library ... 467
20.5 Enabling Windows Crash Dumps ... 468

20.5.1 MS Windows Logging 468
20.6 Configuring Import and Update Folders for Network Drives 469
20.6.1 Working with Import and Update Folders 469
20.7 Promo Server Remote Configuration REST API 472
20.7.1 GET API Root 473
20.7.2 GET All Channels 474
20.7.3 GET Channel 475
20.7.4 GET All Machine Addresses under Channel 476
20.7.5 GET Machine Address 478
20.7.6 GET Status of Machine Address 480
20.7.7 GET Global Settings 482
20.7.8 POST Channel 483
20.7.9 DELETE Channel 485
20.7.10 PUT Channel 486
20.7.11 POST Machine Address 488
20.7.12 DELETE Machine Address 490
20.7.13 PUT Global Settings 491
20.8 Logging 492
20.8.1 Daily Log (As Run) 492
20.8.2 Log File Locations 492
20.9 Glossary 494
20.9.1 Viz Channel Branding Terminology 494
20.9.2 Abbreviations 496

1 Introduction

Viz Channel Branding provides high-quality Vizrt graphics for channel branding. It simplifies the playout of playlists, potentially as a fully automated workflow that integrates with existing channel traffic and scheduling solutions. Viz Channel Branding combines pre-scheduled playlist operations with the ability to add real-time 2D and 3D graphic effects. In complex environments, Viz Channel Branding helps centralize entire workflows to one or a few desktops.

- Introducing the Viz Channel Branding Workflow
- Key Benefits
- All-in-One Workflow
- Graphics Workflow
- Related Documents
- Feedback

1.1 Introducing The Viz Channel Branding Workflow

For seamless integration with existing hardware, the solution employs standard rack mountable PCs. Updates and upgrades are easily applied without requiring modifications to the solution as a whole.

1.2 Key Benefits

- Translates network promotions and branding strategies to rules that automatically create graphics playlists.
- Operators can use pre-defined graphic templates.
- Simplified sponsoring, including automatic logging and reporting, time frame-accurate reports about aired content, and “As-Run” logs.
- Integrates with popular automation systems, using standard protocols.
- Full support for SD and HD formats, real-time 2D/3D graphics and DVE effects.
- For Viz Engines with compatible graphic / video cards, 4K and IP streaming is supported.
- Access to Viz One video content.
- Automate an almost unlimited number of channels and monitor them all from a single Viz Channel Branding client.
- Automatically adapt changes in the playlist, up to the very last minute.
- Allows the promotion of programs and shows across multiple channels.

1.3 All-In-One Workflow

Viz Channel Branding integrates with existing systems used in the traffic department and master control room. The application utilizes schedules from the scheduling and automation systems and is able to import any fixed format text file like XML, CSV, or Microsoft Excel with Playlist Importer.

Copyright © 2024 Vizrt

With Integrations Hub network update and schedule fetching from automation systems with the standard HTTP REST protocol can be implemented.

After importing the schedule and its data, a new Viz Channel Branding playlist, including a graphic layer, is automatically created. If the playlist is *continuous*, the playlist will be updated to reflect the new data received.

To streamline workflow as much as possible, Viz Channel Branding contains two layers of automation:

1. The first layer is Viz Channel Branding's capability to read schedules. By analyzing the schedule, the application can automatically launch graphics at a defined offset. This can be the beginning, the credits or the end time of a specific event, according to pre-defined rules.
2. The second layer of automation is the linkage between the different properties of objects from graphics (texts, images, geometry, etc.) and the information as it exists in the imported schedule. This linkage can be customized to create a fully automatic system that "decides" which graphics to use and add to which primary event, and what data to take from the schedule and insert to the matching object of that graphics. In other words: Viz Channel Branding is capable of automatically choosing the right graphics and filling it with adequate content.
   After receiving the updated schedule from the scheduling system, the automation system takes control during playout. Using standard protocols like *VDCP* and intelligent interface, the automation system then triggers when certain events should be run. With the Broadcast eXchange Format (BXF) Viz Channel Branding also provides an interface mechanism that allows broadcast systems to exchange mission-critical data.

Viz Channel Branding has been designed to maintain full control over its playlist up to the very last minute before playout. Graphics that are connected to a specific event will collect information like program name and start time automatically. In case any changes to the playlist occur, the graphics will update all the relevant information. The playlist can be populated with graphics effects from all sources that follow the ODBC standard.

In addition, an automatic error check makes sure, that changes in the original schedule do not corrupt the imported playlist. Viz Channel Branding offers advanced error protection features that alert about animations overlapping, mismatches or any other deviations from the schedule.

1.4 Graphics Workflow

Viz Engine supports several workflows with regards to how and when graphics and video are mixed. To simplify it we can say that Viz Engine supports graphics and video mixing in an *Upstream* or *Downstream* workflow.

An **upstream** workflow means that graphics are rendered separately and independently of the video signal before it is keyed on top of the video signal, traditionally by a vision mixer.

Opposite to upstream mixing you have **downstream** mixing. Downstream mixing means that graphics and video are composited and mixed by Viz Engine. This can be live video or clips treated as DVE for graphics insertion and/or squeeze backs.

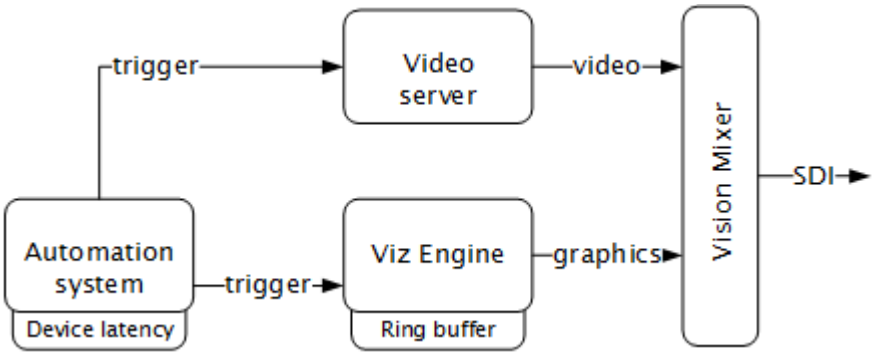
> ⚠ **Note:** The diagrams below are simplified. Details such as audio, keying (key out) etc. are not shown. For details about the full feature set of the Viz Engine see the Viz Engine manual at the Documentation site.

Copyright © 2024 Vizrt

1.4.1 Upstream

An **upstream** workflow basically means that graphics are rendered separately and independently of the video signal before it is keyed on top of the video signal, traditionally by a vision mixer.

Note that mixing graphics and video in an upstream workflow introduces **latency** because graphics must be rendered before it can be mixed with the video signal for a final SDI output.

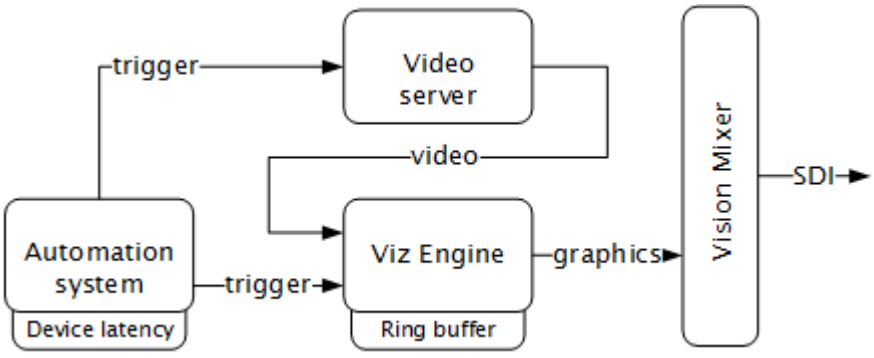


Therefore most automation systems, including Media Sequencer, are able to adjust and cater for latency introduced by devices such as Viz Engine due to the buffer it needs to render graphics in real-time.

In the case of Viz Engine and graphics output the latency has to match Viz Engine's **ring buffer**. Ring buffer is simply put the time (in frames) Viz has before its graphics should be ready for on air, and sent to your vision mixer.

Setting the ring buffer size is often simply a matter of setting a parameter called **Ring buffer size** using Viz Engine's configuration tool; however, the ring buffer size is only a maximum value which means that graphics rendered faster than the set size will be on air before it was intended. The latter is often the case when an automation system simultaneously takes video and graphics on air.

1.4.2 Downstream



Opposite to Upstream mixing you have **downstream** mixing. Downstream mixing basically means that graphics and video are composited and mixed by Viz Engine. This can be live video or clips treated as DVE for graphics insertion and/or squeeze backs.

Copyright © 2024 Vizrt

In the case of the automation system, the downstream workflow allows such systems to simultaneously take video and graphics on air without worrying about latency as Viz Engine manages compositing and mixing of graphics and video.

In case graphics rendering should fail the video will simply pass through before the graphics engine is on air again and graphics compositing can commence.

1.4.3 Transition Logic

Most broadcast video systems have three independent physical layers: *Front, Middle* and *Back*. In certain situations this can be limiting. With TL you can have any number of **logical layers**. TL layers are always created and shown in the "physical" Middle layer and will not interfere with the Front or Back layer. Transition Logic allows for more than one scene to be on air at a time. For example, a graphics covering the lower third of the screen, and another covering the left for over the shoulder graphics, can be on air at the same time.

Playout with transition logic means that the graphics taken to air are loaded into a background scene that can always be on-air. By using this method several independent graphical elements can be on air at the same time and can be taken in and out independently. Intelligent transitions can be built into the background scene ensuring smooth and seamless transitions between changing elements.

Summary

- A Transition Logic scene is not a single scene, but a set of Viz graphic scenes that consist of a background scene that may have multiple layers of graphics that can be on air at the same time and independently controlled.
- Each layer in the background scene may have multiple referring foreground scenes; however, each layer can only show one foreground scene at a time.
- A TL layer can have zero or 1 (one) states. If the TL layer doesn't have a state, then this layer will not be part of the combo template. Only *one* state can be active or chosen for each layer, however the scene designer can create as many layers as needed.

1.5 Related Documents

- Viz Engine Administrator Guide : Installing Viz Engine software; supported hardware and how to configure Viz Artist and Viz Engine.
- Viz Artist User Guide: How to use Viz Artist to create 3D graphics.

For more information about all Vizrt products, visit:

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

Copyright © 2024 Vizrt

1.5.1 See also

- Glossary

1.6 Feedback

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

Copyright © 2024 Vizrt

2 Requirements

There are compatibility requirements, access policies and rights for the software components in a Viz Channel Branding work flow.

IMPORTANT! Always check component software Release Notes, bundled with the software installers at download.vizrt.com and in the Documentation sub-directory of the program directory once the program is installed.

- General
- Hardware
- Software
- Viz Channel Branding
- Integration with Viz One
- Network Ports and Connections Ports and Connections

2.1 General

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

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

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

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

2.2 Hardware

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

For older hardware, such as Viz Engine, that is used with newer versions of Vizrt software, it is always recommended to check the current hardware specifications for the new software version to make sure that the set-up will work.

Copyright © 2024 Vizrt

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

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

2.3 Software

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

IMPORTANT! Always read the Release Notes for late minute changes or requirements, if any. If there are differences in the hardware or software requirements listed here, the Release Notes supersede this manual.

2.4 Viz Channel Branding

The Viz Channel Branding client is used by the traffic department to schedule programs that are controlled by automation systems in the master control room. It has a TCP connection to Media Sequencer. The client gives the user access to creation, editing and playout of pages with graphics and video clips.

Component	Requirement
Software	Media Sequencer 4.1 or higher. Microsoft .NET Framework 4.8. Microsoft Visual C++ 2010 Redistributable Package (free download from http://www.microsoft.com/en\-us/download/details.aspx?id=5555). Viz Engine 3.6 or higher. Viz Engine 5.0 recommended. If you are using a local Viz Engine to preview video clips from Viz One, Viz Engine must be installed with video codecs: (Windows 7 = K-lite Codec Pack full) Viz One 5.11 or higher if using Viz One integration.
Operating system	Windows 10 64-bit Windows 7 64-bit Windows Server 2008 R2

Copyright © 2024 Vizrt

2.5 Integration With Viz One

Viz Channel Branding integrates with the Viz One MAM system.

2.6 Network Ports And Connections Ports And Connections

Component	Port	Who may use it
Media Sequencer	8580, 8594	Integrations Hub, Viz Channel Branding Client
Integrations Hub	9991	Playlist Importer
Integrations Hub	9990	Integration Modules, Quick Scheduler Server
Promo Server	9980	Failover Helper
Quick Scheduler Server	9970	Quick Scheduler
Promo Server	9005	Integrations Hub, Viz Channel Branding Client
Viz Engine	6100, 6800	Media Sequencer, Viz Channel Branding Client
Preview Server	5400, 21098	Viz Channel Branding Client
Viz One	80	Integrations Hub, Viz Channel Branding Client
Failover Helper	9960	Any browser

External Component	Port	Who may use it
Harris Automation	10050	Harris Integration Module
Marina Automation	18083	Marina Integration Module

Ports in table above are TCP. Read the *From* and *To* in the table as initiator and receiver. Example: Viz Channel Branding client uses TCP-port 8580 and 8594 when exchanging data with the Media

Copyright © 2024 Vizrt

Sequencer. Not all ports are used at all times, but they can be used and must be available. Traffic can be assumed bi-directional, i.e. from and to the services using the ports indicated in the table.

Note: Viz Channel Branding Marina Integration sends traffic on port 18083 to the Marina Automation and uses port 12212 bi-directional.

IMPORTANT! Some ports are configurable. For instance the Viz Engine uses port 6100 (and additional port 6800 for dual-channel engines) but can be configured to use other ports.

Copyright © 2024 Vizrt

3 Setup And Configuration

When setting up Viz Channel Branding, you may also install software components in addition to the Viz Channel Branding Client application. Referenced below, are relevant manuals to help with the installation of non-bundled software components.

When planning setup and configuration, two aspects in particular must be carefully considered before the installation and configuration:

- How will the schedule be provided and how will it be ingested?
  A Viz Channel Branding system may be connected to an automation system that automatically provides the schedule derived from a scheduling department. Other installations may read the schedule from a text file, a database, XML, Excel or by other means. Channel Branding refers to the received and ingested schedule (what to do and when to do it) as a resulting 'playlist'. At any given time, Channel Branding must have a valid, active and running playlist for the planned automated actions to be performed.
- How will graphics be triggered?
  There are many protocols (VDCP, REST, GPI and Intelligent Interface) that can trigger graphics. In some cases, external systems (automation or traffic systems) will trigger graphics by sending a command utilizing one of the protocols mentioned. These are called externally triggered events. In some cases, the automation system will only trigger main events (for example: a film) and graphics will then be applied by a time offset from the main event.

Viz Channel Branding is typically installed in the following locations:

- Viz Channel Branding's Playlist Importer and Integrations Hub on the Media Sequencer machine in the **server room**. Media Sequencer is in most cases installed on the Viz Engine (program) machine.
- Viz Channel Branding Client on a client machine in the **master control room**.

This section covers the following topics:

- Installation
- Playout Configuration
- Scheduling Configuration
- Video Configuration
- Preview Server
- Quick Scheduler Server

Copyright © 2024 Vizrt

3.1 Installation

3.1.1 Bundle Installer

The bundle installer can be used to install one or all of the separate components (programs) that are part of Viz Channel Branding. It is typically distributed as:

```
ChannelBrandingBundleInstaller-x64-<VERSION-NUMBERS>.exe
```

Start the installer and select the components you require for installation, repair or deletion. The installer will also perform the required basic configuration and starting service processes. The bundle installer is the most convenient installation method and is therefore recommended. See below if you require the separate MSI components.

> ⚠ **Note:** If you need to install from separate + installers, see Unbundling the Wrapped Installer below.

3.1.2 Unbundling the Bundle Installer

You can unpack the separate .MSI installers in the bundle installer by executing from the command-line with the argument "*/msi*".
For example:

```
ChannelBrandingBundleInstaller-x64-<VERSION-NUMBER>.exe /msi
```

This will create a new directory with the installer name that contains the separate MSI installers.

The bundled installer will also accept the standard */passive* and /_silent _options:

- **/silent:** no GUI shown
- **/passive:** GUI shown, but no need to press any buttons

3.1.3 Installing Viz Channel Branding Client

Viz Channel Branding Client in the master control room must be connected to Viz Engines for program and preview output; however, a Viz Engine is often required for local preview in addition. Ensure that Viz Engines are ready before starting the installation steps below.

Since a Viz Engine is also used for local preview by the Page Editor, a **VGA / Local Preview Core** version must be installed on the Client machine.

To install the Viz Channel Branding client

1. Start the Viz Channel Branding Client installer.
2. Optionally adjust installation path.

Copyright © 2024 Vizrt

3. Click **Install**.
4. Click **Finish**.

3.1.4 Time Format and Time Zones

All internal time handling calculations and operations (the time reference in Media Sequencer), is always UTC (Universal Time Code) regardless of the timezone and local settings on your Windows servers and PCs. This is a reference time only, Window PC clients that are running the User Interface will often have different time-settings. Please keep the following rules in mind:

- The Media Sequencer reference time is UTC.
- The Viz Channel Branding Client will show the playlist information in *client* machine time. If you click on the time-display in the Viz Channel branding, UI it will show Media Sequencer's time and local offset, if any.

When importing and handling time-dependent items in Viz Channel Branding please make sure that:

- Logic in rules and Page Content Filling (PCF) should use the channel's time zone setting, see **Time Zone Settings**.

A timecode in the **Rules** section must always follow full time-code format, the value you manually enter must be in the form **HH:mm:ss:ff**.

See Also

- Promo Server
- Working with Rules
- Page Content Filling Aliases

Copyright © 2024 Vizrt

3.2 Playout Configuration

- Installation of Viz Engine Machines for Playout
- Media Sequencer
- Optional Startup Parameter for the Viz Channel Branding Client
- Configuration of Channels with channels.ini
- Forked Execution
- Configuring Device Channels
- Configuring Playout Devices
- Logging

3.2.1 Playlist Importer

Viz Channel Branding 5.0 introduced a re-written Playlist Importer (PLI) that exclusively communicates with Integrations Hub via v2 of the REST API, that in turns pushes it to Media Sequence Engine (MSE).

3.2.2 Client Placement

The main typical install-location of the Viz Channel Branding Client application is the master control room (MCR), for monitoring purposes and manual update of data. It can also be used for manual playout.

Integrated with an automation system, Viz Channel Branding can import its schedule(s) and allow the automation system to control the playout of video and graphics without any manual intervention.

The following software components must be installed on the client machine in the MCR:

- Viz Channel Branding Client
- Viz Engine (local preview core)

See Also

- Master Control Workflow
- Schedule Collector
- Playlist Importer
- Configuring Scheduling

3.2.3 Installation of Viz Engine Machines for Playout

It is recommended to install Playlist Importer, Integrations Hub and Media Sequencer on the same machine as the Viz Engine defined as your program output channel.

IMPORTANT! Ensure that clocks on all machines in the installation are synchronized. It is also recommended to set all date formats on all machines to the same format.

Copyright © 2024 Vizrt

3.2.4 Media Sequencer

There must **always** be a connection to Media Sequencer(s) when running Viz Channel Branding playout system. By default, the Media Sequencer runs as a Windows service, and is automatically launched on machine startup. You can check Media Sequencer by starting the Media Sequencer Launcher application. From the Launcher you can start, stop and set run-mode. Alternatively, start the Windows Services component and make sure Sequencer is running and set to autostart. You can see the complete Media Sequencer Documentation (assuming it's running on standard port) at:

```
http://YourMediaSequencerhost:8580/doc
```

3.2.5 Optional Startup Parameter for the Viz Channel Branding Client

Parameter	Description	Example

`-msehost`	Hostname or IP address for Media Sequencer. If `none` is used Channel Branding Client will work in MCR mode, reading each channel's Media Sequencer from will read `channel.ini` file from the default path, `[ProgramData]\Vizrt\Viz Channel Branding\channels.ini --` See Configuring Playout and Configuration of Channels with channels.ini. If no parameter is provided, MCP will attempt to connect to `localhost`.	`-msehost none`
		`-msehost 10.210.0.10`
		`-msehost hostname`

See Also

- Vizrt's Documentation Center for more information on setup and configuration of Viz Engine
- Import Guide

3.2.6 Configuration of Channels with channels.ini

In order for the Master Control Room machine(s) that use the MCR-mode of Channel Branding to have access to all channels, you **must** populate its `channels.ini` file with all channels defined on the Media Sequencer machine(s). If the *channel.ini* configuration file is not present or incorrectly formatted, the Channel Branding Client in MCR-mode will abort startup with an error message.

To configure channels with channels.ini

1. In Windows Explorer, locate the Viz Channel Branding's directory where the *channels.ini* is:
   - For Windows 7 or Windows Server 2008 R2 and higher (64-bit Windows) this will normally be:

Copyright © 2024 Vizrt

- `[ProgramData]\Vizrt\Viz Channel Branding\channels.ini`

2. Open the `channels.ini` file in the text editor of your choice.
   - The file could be empty if not previously configured.
3. Enter each channel as a separate line in the text file.
   - The format is: `ChannelName=<IP address>` **or** `<hostname>.`
   - Example: If you have two channels named *channel1* and *channel2* then the content of *channels.ini* would be (using a random example IP-address):

```
channel1=10.10.10.10
channel2=20.20.20.20
```

4. Save and close the file.

   ⚠ Note: The *channels.ini* is a normal text file so make sure you save the file as type text.

5. Start Channel Branding Client using the *Viz Channel Branding Client - MCR Mode* shortcut.

See Also

- Channel Settings

3.2.7 Forked Execution

Forked execution allows you to set up a single channel with multiple concepts, triggering one Viz Engine per concept. Forked execution supports standalone and Transition Logic scenes. Concepts are defined per channel when Configuring Device Channels.

⚠ **Note:** As an option you can also use this setup to have the same graphics concept being rendered on multiple Viz Engines.

See Also

- *Viz Artist User Guide* section on *Transition Logic*

3.2.8 Configuring Device Channels

Viz Channel Branding is able to handle multiple TV channels. In order to accomplish this you need to define separate configuration settings per channel. You therefore need to define playout devices as part of the configuration. Playout is defined as graphics and/or video output.

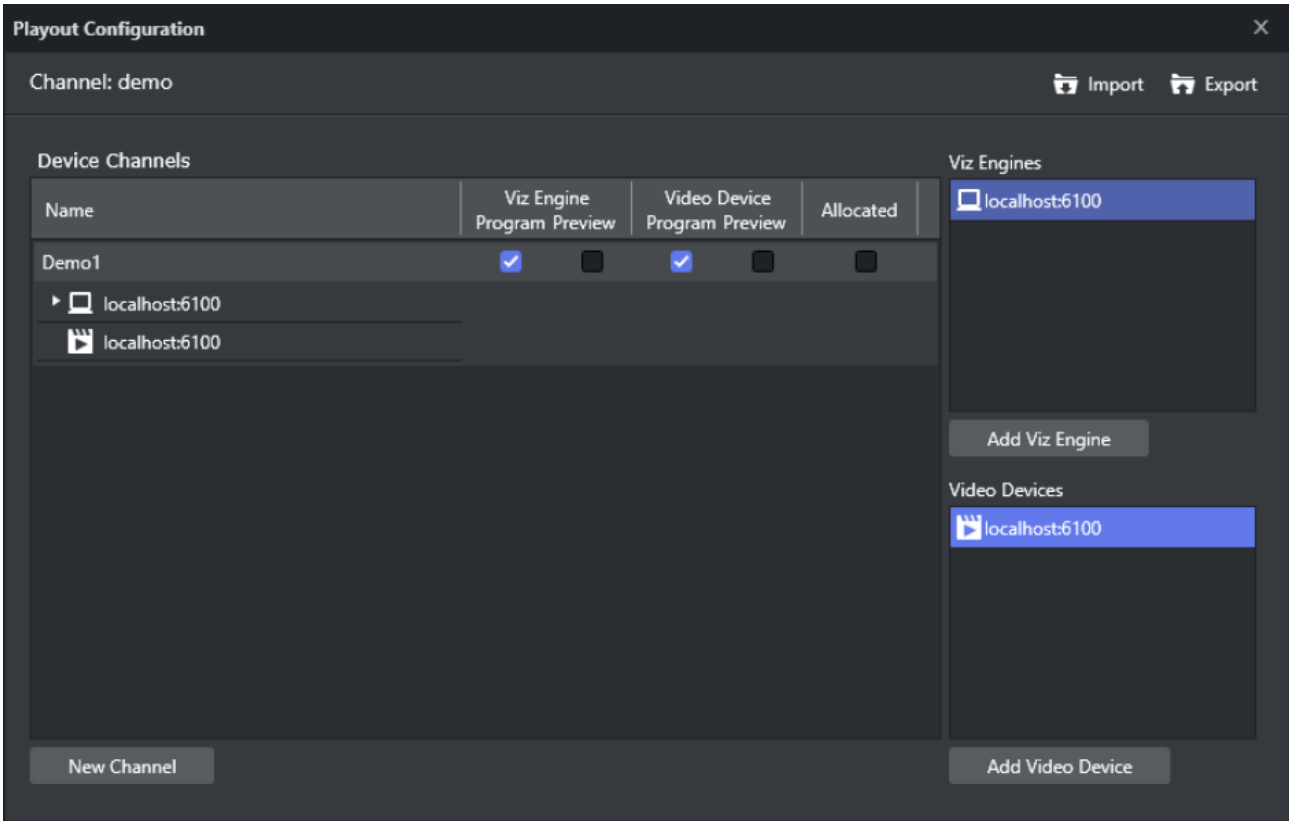
Viz Engines must be assign for *graphics output*. For *video output*, you can either assign Viz Engines or other video playout servers that are able to handle your video output. For more information, see Upstream and Downstream workflows in the Graphics Workflow section.

Device channels are essentially placeholders for playout devices. You use device channels to group the graphics and video devices into logical areas of responsibility (program and preview channels for graphics or video devices).

Copyright © 2024 Vizrt

Graphics device channels may also be configured to apply a given graphics design concept (for example *weather*, *sports*, *election*, *news*).

Click the **Playout configuration** icon on the main menu-bar to display the **Profile Configuration** panel:



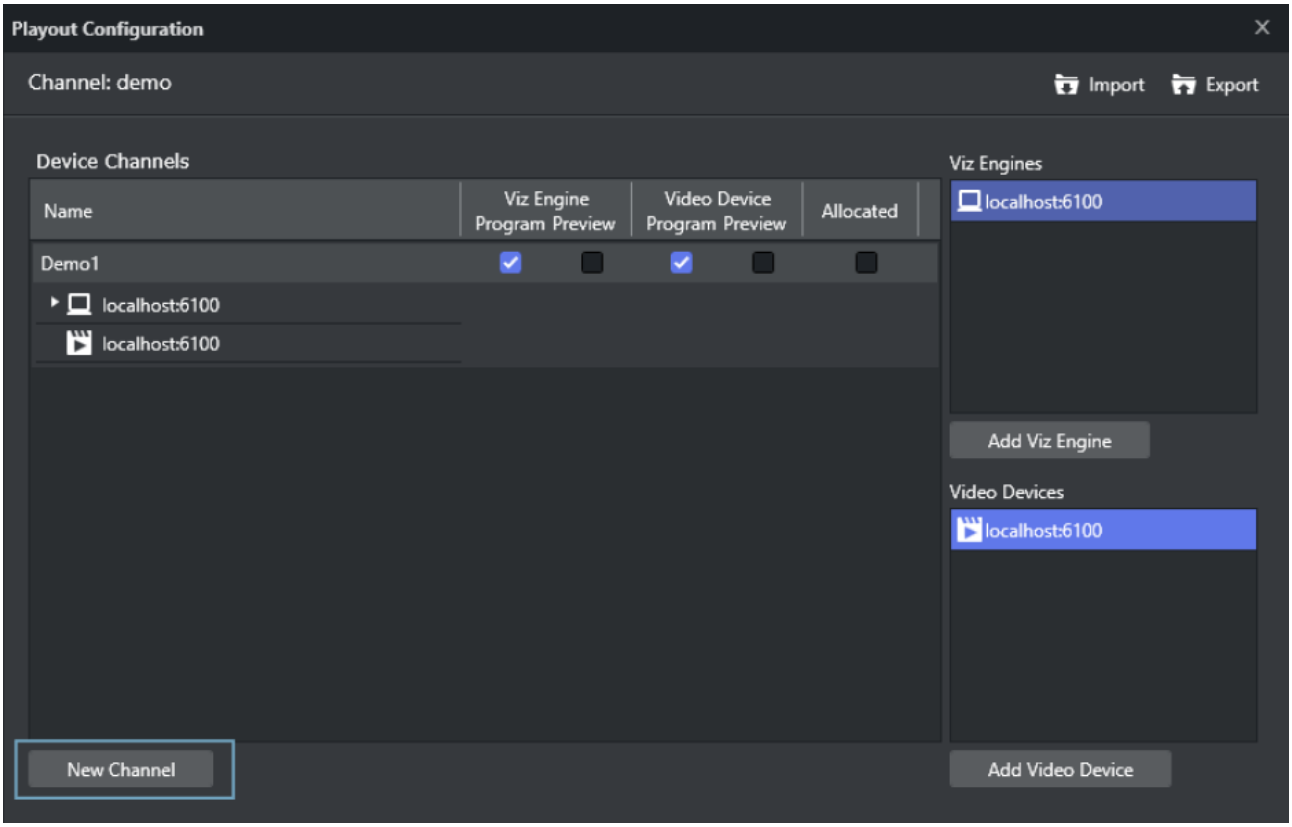
You can use the **Import** and **Export** buttons to import (or save) a configuration to an XML-file for safekeeping. Importing a configuration from an XML-file makes it easy to quickly change configuration parameters.

This section contains the procedures:

- Add a Playout Device to the Device Channels List
- Add a Concept Override for a Channel's Playout Device
- Rename a Playout Device in the Device Channels List
- Remove a Playout Device in the Device Channels List

Copyright © 2024 Vizrt

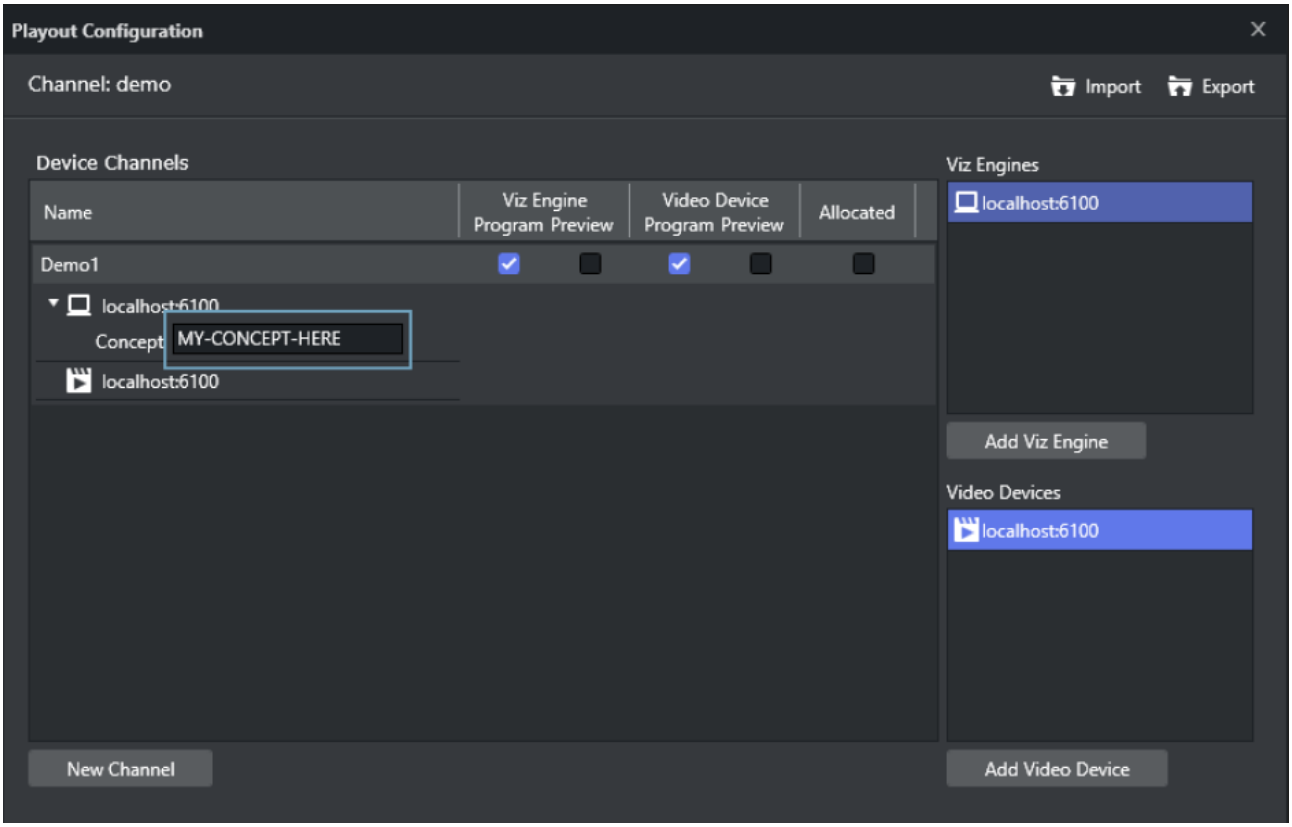
To add a playout device to the Device Channels list



- Click the **New Channel** button, or drag and drop a playout device to the **Channels** list.

Copyright © 2024 Vizrt

To add a Concept Override for a channel's playout device



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

> **Caution:** Concept names are case sensitive.

To rename a playout device in the Device Channels list

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

To remove a playout device in the Device Channels list

- Select the channel and press the **Delete** button or right-click the device and select **Remove Output**.

3.2.9 Configuring Playout Devices

Playout devices are physical output devices that play out program or preview content of graphics or video. These devices are added to *device channels* for the purpose of logically grouping playout

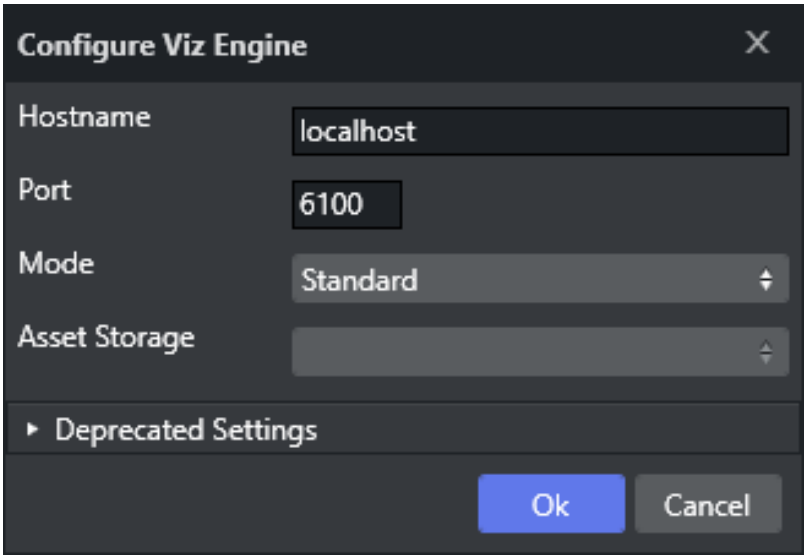
Copyright © 2024 Vizrt

devices into functional areas of responsibility; they act as program or preview channels for either graphics or video.

This section contains the procedures:

- Configure a Graphics Device
- Configure a Video Device
- Configure a Dummy Video Device
- Edit a Video Device
- Delete a Video Device
- Add a Video Device to the Channels List
- Enable Scene Transitions

To configure a graphics device



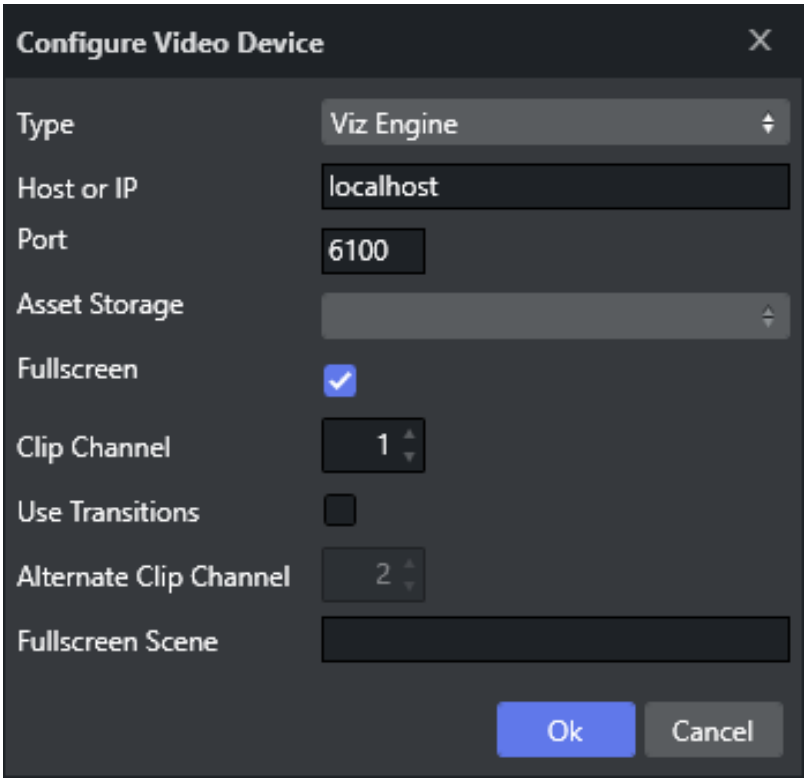
1. Click the **Add Viz** button to open the Configure Viz Engine dialog box.
    a. Enter the hostname and port.
    b. Default port for Viz Engine is 6100.
2. *Optional*: Select **Mode**.
    a. **Scene Transitions**: See To Enable Scene Transitions.
    b. **Still Preview**: See To Enable Scene Transitions.
3. *Optional*: Select an **Asset Storage** location.
    a. Asset Storage lists available Viz Engine storage for clip transfer and playout.
4. Click **OK**.

A status indicator will show if the renderer is on-air.

> ⚠ **Note:** *Deprecated Settings* are not used with Viz Channel Branding.

Copyright © 2024 Vizrt

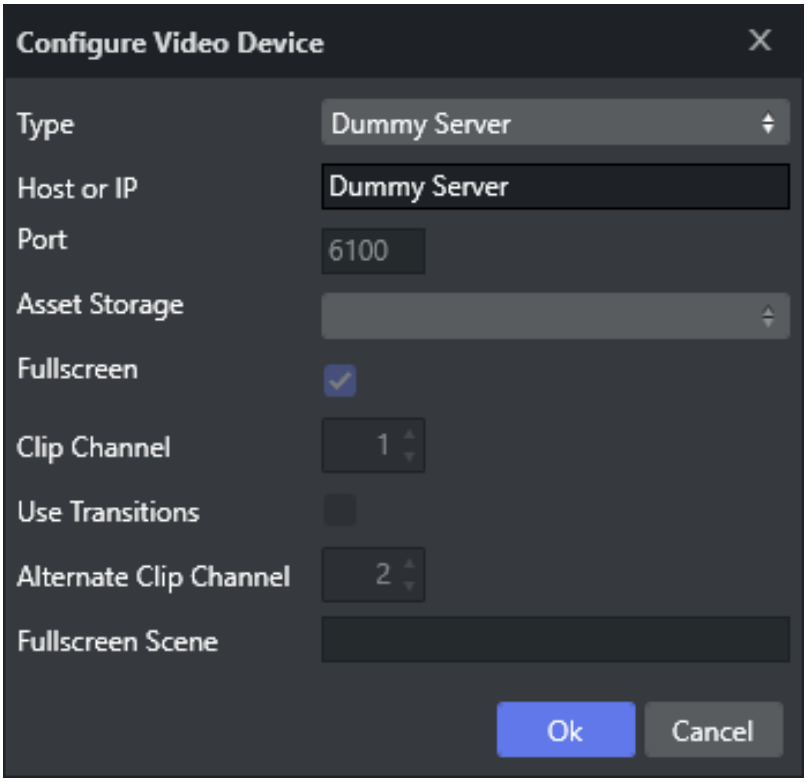
To configure a video device



1. Click the **Add Video** button to open the **Configure Video Device** dialog.
2. Select the video server **Type**.
3. Enter the **Host** or **IP address** and the **port**.
   - Default port for MVCP is 5250.
   - Default port for Viz Engine is 6100.
4. Select an **Asset Storage** location.
   - Asset Storage lists available Viz Engine storage for clip transfer and playout.
5. Enable/Disable **Fullscreen** mode.
   - When Fullscreen is *enabled*, which is the default behavior, stand-alone video clip elements are played out full screen in the back layer. Graphics in the middle or front layer will still play.
   - When Fullscreen is *disabled.* stand-alone video clip elements will only replace the current video clip without triggering animations or transition effects.
6. Select **Clip Channel** (only relevant for Viz Engine).
7. Click **OK**.
   A status indicator will show if the device is on-air/online.

Copyright © 2024 Vizrt

To configure a dummy video device



1. Click the **Add Video** button to open the **Configure Video Device** dialog.
2. Select the video server **Dummy Server**.
3. Enter a name for your dummy server (e.g. Dummy Server).
4. Click **OK**.

> ⚠ **Note:** This will enable use of the Timeline Editor when Viz One is simply a gateway and video is served from an alternative asset management system.

To edit a video device

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

To delete a video device

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

To add a video device to the Channels list

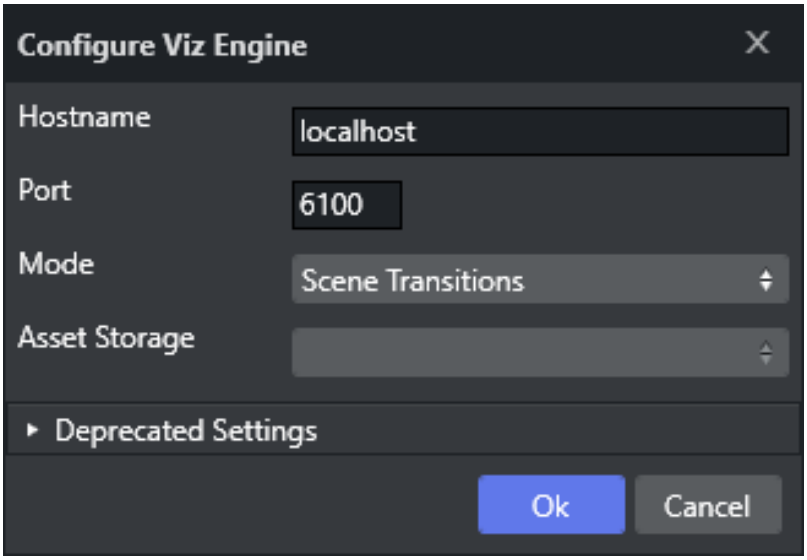
- Simply **drag and drop** a Viz Engine or video device onto the channel in the Channels list, or select it and from the context menu and choose:

Copyright © 2024 Vizrt

- **Add to profile** (creating a new channel) or
- **Add to selected channel**.

To enable scene transitions

Transition Effect scenes are made in Viz Artist and can be applied to data elements to create custom transition effects from one scene to the other. If an effect is specified, the effect will be shown when the scene is taken on-air.
Effects include wipes, dissolves and alpha fades.



1. Configure the Viz Engine settings as described in how Configure a Graphics Device.
2. Set the **Mode** to **Scene Transitions** to allow the renderer to copy (or snapshot) the scenes for creating a transition effect between them.
3. Click **OK**.
4. Add the program renderer to the program channel.

> **Note:** The program channel must be configured and on-air in order for the effects to be visible.

3.2.10 Logging

Enabling Logging

To enable logging

- On the Media Sequencer machine, create the following folder path: `%ProgramData%\Vizrt\Media Sequencer` to generate a Daily Log.

Copyright © 2024 Vizrt

Note: The log produced by Viz Channel Branding only shows what the Media Sequencer attempted to play out (and not what was actually played out).

For more information on how to format the output, see the Media Sequencer manual. See also the Appendix topic Logging.

Copyright © 2024 Vizrt

3.3 Scheduling Configuration

A typical installation includes one or several Viz Channel Branding Client and a local Viz Engine, running in Scheduling Mode against a 'Large MSE'. Playlist Importer (PLI) are typically installed in order to collect and ingest schedule files into Viz Channel Branding.

This setup allows the Scheduling Center to prepare and preview playlists before sending them to playout by the Viz Engine(s).

The following software components are in use:

- Viz Channel Branding Client
- Viz Channel Brandings's Playlist Importer (PLI) and Integrations Hub
- Viz Engine (local preview core, per Client)
- Media Sequencer, on a stand-alone machine if using more than one client.

See Also

- Playout
- Vizrt Documentation Center

Copyright © 2024 Vizrt

3.4 Video Configuration

To enable video to pass through Viz Engine when no scenes are loaded, you must activate the Matrox Watchdog functionality and enable video to loop through the Matrox X.mio or X.mio 2 board.

> ⚠ **Note:** The procedure below is not relevant for Viz Engine 3.6 and X.mio 2 plus users.

3.4.1 Configure Viz Engine to Pass through Video as DVE

1. Start Viz Engine's configuration tool (Viz Config).
2. In the **Matrox** section under the General tab enable **Use Watchdog**.
   - **Use Watchdog:** When activated, this setting enables the Matrox X.mio watchdog feature. It defines what kind of key is produced when the watchdog takes over control. Default mode is Inactive. When the watchdog is enabled, turning off the video out will activate the software bypass after a given timeout (see next setting).
3. Set the **Timeout** setting (e.g. 100ms).
   - **Timeout:** Defines the time in milliseconds until the watchdog takes over control. This value should not be smaller than the time of two fields/frames. Default value is 999 milliseconds.
4. In the **Video Output** section, tweak the **h-phase** and **v-phase** settings to have a glitch free switching to and from the Watchdog.
   - *H-Phase and V-Phase: *Shifts the output signal with respect to the sync signal.
5. Open the Viz configuration file.
6. Navigate to the `TARGA3200` section and set the following parameters.
   - Set `video_loopthrough_mode = 2`
   - Set `video_loopthrough_reactivation = 1`

With these configurations you should have a DVE pass through as long as no scene or DVE is activated on the Viz Engine.

See Also

- Viz One Integration
- Global Settings
- Client Configuration File
- Channel Settings

Copyright © 2024 Vizrt

3.5 Preview Server

Preview Server is used in multiple Vizrt client applications to fetch previews of overlay graphics for the Timeline Editor and also, in situations where Viz Engine provides frames for snapshot or thumbnail generation.

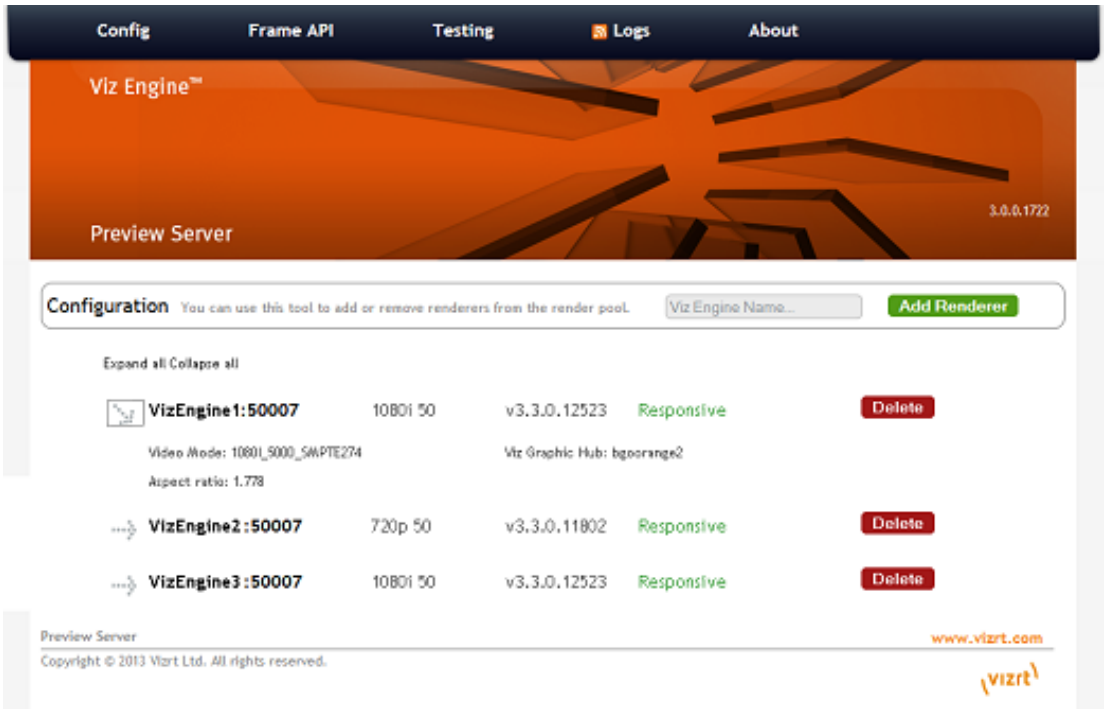
Preview Server can manage a pool of Viz Engines. Clients can query it for a Viz Engine that matches a specific video mode. Preview Server provides load balancing, video mode matching and can be used to create a redundant renderer pool. This section covers the following topics:

- Preview Server Web Interface
- Installing and Configuring Preview Server

See Also

- The Viz Engine Administrator Guide for information on how to install Viz Engine.

3.5.1 Preview Server Web Interface



Accessing the Preview Server Web Interface

- Open the Preview Server Web Interface by navigating the File Explorer to **All Programs > Vizrt > Preview Server > Preview Server Config**.
- Alternatively, access the Preview Server Web Interface in a web browser, using the host name of the machine running Preview Server, and port 54000. For example, *http://<hostname>:54000/*.

The Preview Server menu bar has links to the following pages:

- Config
- Frame API
- Testing
- Logs
- About

Config

Preview Server can interact with one or more Viz Engines. These can be configured on the home page of the Preview Server web interface.

http://<hostname>:54000/

Adding a Viz Engine

- Enter the Viz Engine hostname or IP address in the Configuration field, and click **Add Renderer**.

  Tip: Click the arrow next to the Viz Engine entity or the **Expand All** link to see information on video mode, aspect ratio and Viz Graphic Hub connection.

  Note: If not specified, Viz Engine is added with the default port 50007. When using Viz Engines in a *dual channel* setup, it is possible to run multiple instances of Viz Engine on a single host. Each instance uses a unique port, which means that two Viz Engines are only considered duplicates when both the hostnames and port numbers are identical. When referring to Viz Engines in a dual channel setup, type the port number after the host name (for example *VizEngine1:51007*).

Removing a Viz Engine

- Click **Delete** next to the Viz Engine entity that is to be removed.

Frame API

The Preview Server REST API page displays the REST API provided by the Preview Server and includes information on the Resource Types and Data Types that are used in the interface.

http://<hostname>:54000/doc/rest_manual

Testing

This page provides:

- Snapshot Testing - Test snapshot service.
- Snapshot Set Testing - Test snapshot service with set of positions.

Copyright © 2024 Vizrt

To use these features, enter the snapshot XML data, payload XML data and the path to a scene. When the **Test this data** button is pressed, a feed is returned containing links to the requested snapshots.

http://<hostname>:54000/testing

Logs

This page provides a feed with links to Preview Server logs.

http://<hostname>:54000/logs

About

Click to see information on Preview Server version and Third Party Component Credits.

3.5.2 Installing and Configuring Preview Server

This section covers:

- Requirements
- Installation
- Configuration

Requirements

Preview Server requires Microsoft .NET version 4 or newer. The free .NET 4.0 bootstrap installer may be downloaded directly from Microsoft.

Installation

Preview Server is a separate installer and can be downloaded from ftp.vizrt.com in the products/VizPreviewServerConnectionBroker/Latest Version folder. Please read the Release Notes before installing the product.

Installing or Upgrading Preview Server

As a user with Administrative rights:

1. Run the installer *Preview Server-<version>.msi*.
2. Click **Next**.
   - *Optional*: Select the location of the installed files.
3. Click **Next**.
4. Click **Install** to start the installation process.
5. Click **Finish**.
   The installer will install the Preview Server as an automatically started Windows service and start the service. This could require the user to open the Preview Server port if using a Firewall.

Copyright © 2024 Vizrt

To check that the service is correctly installed and running, go to **Windows: Control Panel > System and Security > Administrative Tools > Services** and check that the **Preview Service** is running and set to automatically start.

Configuration

Preview Server Service

Preview Server is a Windows Service. It sets up an HTTP REST service on **port 54000** on the host it is running on. This service accepts requests for graphical snapshots of playout elements from a Viz Engine.

Viz Engines

Although Preview Server can run on the same machine as a Viz Engine (also used for the Newsroom Component data element preview), it is also possible to specify additional Viz Engines in order to spread the load. Frame requests will be balanced across available Viz engines in a way that will minimize the memory load of each engine.

The pool of Viz Engines is configured on the Preview Server Web Interface **Config** page.

Scaling of the system should be monitored to avoid excessive client waiting time and potential overload of the Viz Engines.

Frame Cache

Frames are cached in memory, allowing quicker responses when requesting frames that have been requested previously. By default, frames will be cached for a maximum of one day. This behavior can be changed by modifying the `CacheMaxAge` setting in the `Preview Server.exe.config` file , which specifies the maximum number of seconds to retain cached results.

Copyright © 2024 Vizrt

3.6 Quick Scheduler Server

Quick Scheduler Server is a back-end component that hosts all Quick Scheduler client.

This section covers the following topics:

- Configuration
  - Quick Scheduler Server Configuration
  - Channel Settings

3.6.1 Configuration

Quick Scheduler Server Configuration

To configure Quick Scheduler Server follow these steps:

1. Use Windows Explorer to locate the directory *[ProgramData]\Vizrt\Viz Channel Branding\Quick Scheduler Server* and locate the config file.
   Normally: *C:\ProgramData\Vizrt\Viz Channel Branding\Quick Scheduler Server\QuickSchedulerServer.config*
2. Open *QuickSchedulerServer.config* in a text editor.
3. For key ScheduleChannels change the attribute name value to be the target channel names. If you have several channels, each channel is separated by a "," (comma) character.
   Example format: `<add key="ScheduleChannels" value="MUSIC,MOVIE" />`
4. Save the *QuickSchedulerServer.config* file as a normal textfile.
   After you have configured (or changed) the *QuickSchedulerServer.config* file you need to restart the Quick Schedule Server service process.
5. To restart the Quick Schedule Server and to configure the service-level follow these steps:
   a. Start Windows Services administrator utility.
   b. On Windows 7 like systems: Press Windows Start menu button, type *services.msc* in the search bar and press **Enter** to start the program.

Channel Settings

Once you have completed Quick Scheduler Server Configuration and restarted the service process, there are folders for each channel in the directory data directory listed above.

To configure the channel settings, follow the steps below:

1. Use Windows Explorer to locate the directory *[ProgramData]\Vizrt\Viz Channel Branding\Quick Scheduler Server\[channel_name]* and locate the settings file.
2. Open *settings.xml* in a text editor.
3. Modify the settings:
   IntegrationsHubHost: host name or ip of Integrations Hub.
   PublishTimeType: Can be *absolute* or *relative.*
   a. *absolute*: The server will publish schedule to Integrations Hub at the time specific in PublishTime.
      Example: if PublishTime is "05:00:00", the schedule will be publish on "05:00:00".

Copyright © 2024 Vizrt

b. *relative*: The server will publish schedule to Integrations Hub [xx:xx:xx time] before the Broadcast Day Start Time.
Example: if PublishTime is "01:00:00" and Broadcast Day Start Time is "06:00:00", the schedule will be publish 1 hour before Broadcast Day Start Time which is "05:00:00".

PublishTime: Time that server will publish schedule to Integrations Hub in hh:mm:ss format. Can be only positive value. Max value is "99:59:59".

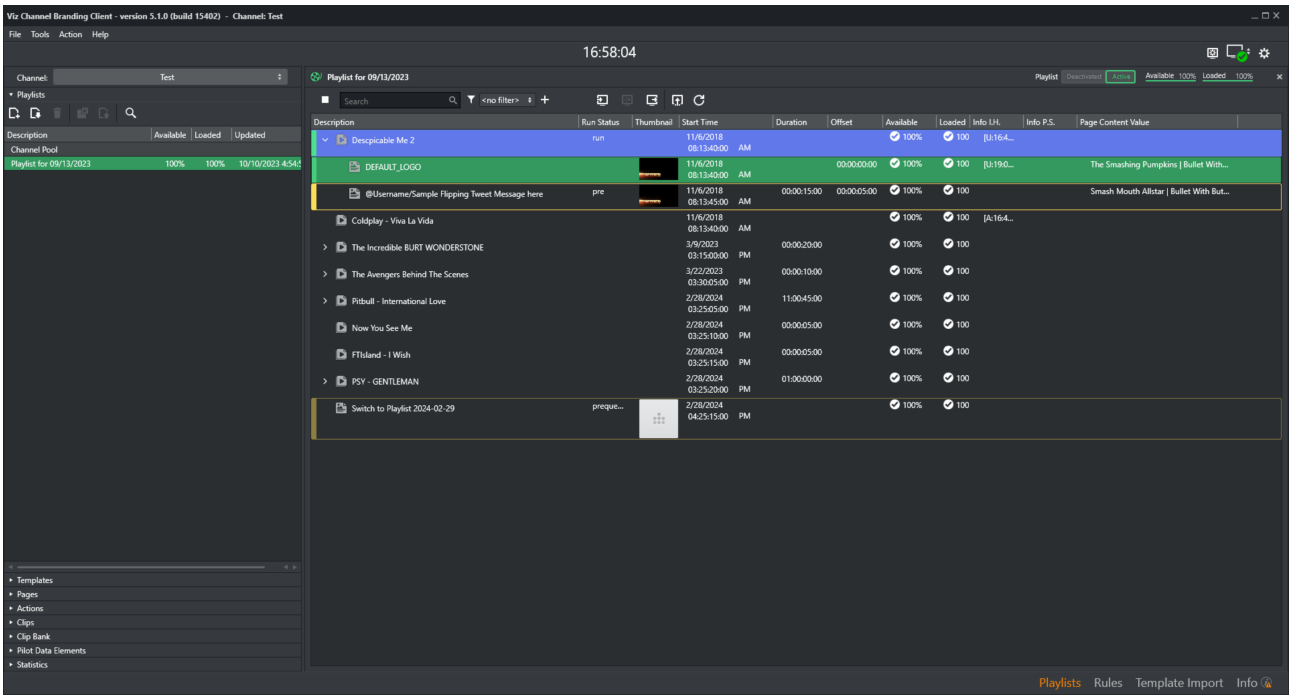
```
<ChannelSettings xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http:/
/www.w3.org/2001/XMLSchema-instance">
  <SettingsVersion>1</SettingsVersion>
  <IntegrationsHubHost>localhost</IntegrationsHubHost>
  <PublishTimeType>absolute</PublishTimeType>
  <PublishTime>05:00:00</PublishTime>
</ChannelSettings>
```

4. Save the `settings.xml` file as a standard text file.
5. After you've configured (or changed) the *settings.xml* file, you must restart the Quick Schedule Server service process.
To restart the Quick Schedule Server and to configure the service-level follow these steps:
   a. Start Windows Services administrator utility.
   b. On Windows 7 like systems: Press Windows Start menu button, type `services.msc` in the search bar.
   c. Press **Enter** to start the program.

Copyright © 2024 Vizrt

4 User Interface

This chapter is a reference to the user interface and contains relevant information for the Viz Channel Branding Client when used in scheduling and master control room (MCR) mode.



The main window is divided into the following sections:

1. Main menu with options **File, Tools, Actions** and **Help**. The toolbar icons can, if required, all be hidden or visible (except the **Time, Local Engine Status** and **Configuration** icons that are always visible) using the **Action** menu page options.
2. Select panel to work on: **Playlists**, **Templates**, **Pages**, **Actions**, **Clips**, **Pilot Data Elements** or **Statistics**.
3. Work area for the selected **Playlist** or **Channel Pool**.
4. Menu-selector for easy access to **Playlists**, **Rules**, **Template Import** or the **Information** panel.

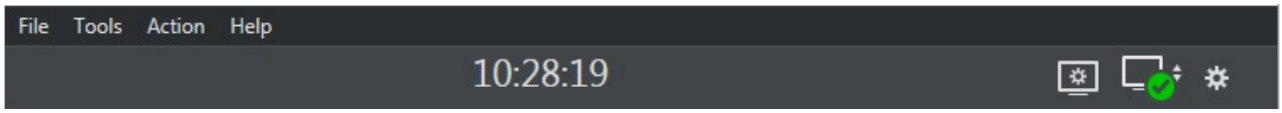
This section also contains:

Copyright © 2024 Vizrt

- Main Menu and Toolbar
- Statistics Panel
- Channels
- Playlists
- Templates, Pages and Pilot Data Elements
- Clips
- Clip Bank
- Actions
- Template Import
- Rules
- Page Editor
- Timeline Editor
- Combo Page Editor

Copyright © 2024 Vizrt

4.1 Main Menu And Toolbar



This section describes the main menu at the top of the interface and the dynamic toolbar below it. The toolbar is dynamic since you can configure which icons (representing actions) are displayed by clicking **Action** in the main menu.

This section covers the following topics:

- Main Menu
- Toolbar

4.1.1 Main Menu

- **File**
  - **Exit**: Closes the application.
- **Tools**
  - **Generate Channels File**: Generates a file containing information about the channels file. This is required when running Viz Channel Branding Client in MCR mode (see Master Control Workflow).
- **Action**
  - **Edit Mode**: Enables edit mode.
  - **Unload Engines**: Clears all loaded graphics from the program and preview renderer memory for the output profile currently in use. This is done before initializing a new playlist or in order to re-initialize the same playlist into the renderer memory. Note that unload commands will affect all clients that are connected to the same Media Sequencer and that use the same output profile (see **Clear Layers** below).
  - **Clear Layers**: Cleans up all layers and background scenes, or clears the output of specific layers (front, middle or back) for your channel's configured Viz Engines.

    > ⚠ **Note:** The graphics will still remain in the memory (see **Toolbar** and **Unload Engines**).

  - **Initialize**: Manually initializes either the currently selected Channel or the Channel Pool. Initializing will fetch and load resources to the memory of the Viz Engine, ensuring that resources such as graphics are ready for playout. If it is not initialized, the Viz Engine will still load resources when needed, but pre-fetching can provide flicker and artifact-free playout in some situations.
  - **Logo Page**: Lets you take or take out the defined logo page.
  - **Error Page**: Lets you take or take out the defined error page.
  - **Video-only Page**: Lets you take or take out the defined video-only page.
  - **Template**: Lets you import or export templates from template pool to XML-files.

Copyright © 2024 Vizrt

- **Pages**: Lets you import or export pages from page pool to XML-files.
- **Actions**: Lets you import or export actions from an XML-file.
- **Ticker**: Lets you import or export Viz Ticker3D playlists (see Ticker Actions).
- **Video bypass**: When enabled, only video is shown. When disabled, video and graphics are shown.

- **Help**
  - **Help**: Launches the documentation in your preferred browser.
  - **About**: Shows the version of the application.

4.1.2 Toolbar

Icon	Description
	Configuration menu.
	Local Viz Engine status. Sub-Commands that can be selected: Show Commands, Show Status and Restart Viz. Note that these commands are sent to the local Viz Engine, normally used for preview. For more information about any given status, see the log messages in your Information Panel.
	Profile settings, click this icon for the Playout Configuration dialog where you can add playout devices and device channels.
	Shows the time of the currently selected channel's Media Sequencer server, with the local client (Viz Channel Branding UI) machine offset applied. Example: If the Media Sequencer machine is UTC and it is 12:00 and UI is CET (UTC+1), then the clock will show 13:00. Click the clock icon to show the Media Sequencer machine time with only the local offset and timezone-offset applied. Given the data in the example above, the time shown is now 12:00, and the time-zone info shown is UTC+0.
	Take **in** the defined Logo Page.
	Take **out** the defined Logo Page.
	Take **in** the defined Error Page.
	Take **out** the defined Error Page.
	Take **in** the Video Only Page.

Copyright © 2024 Vizrt

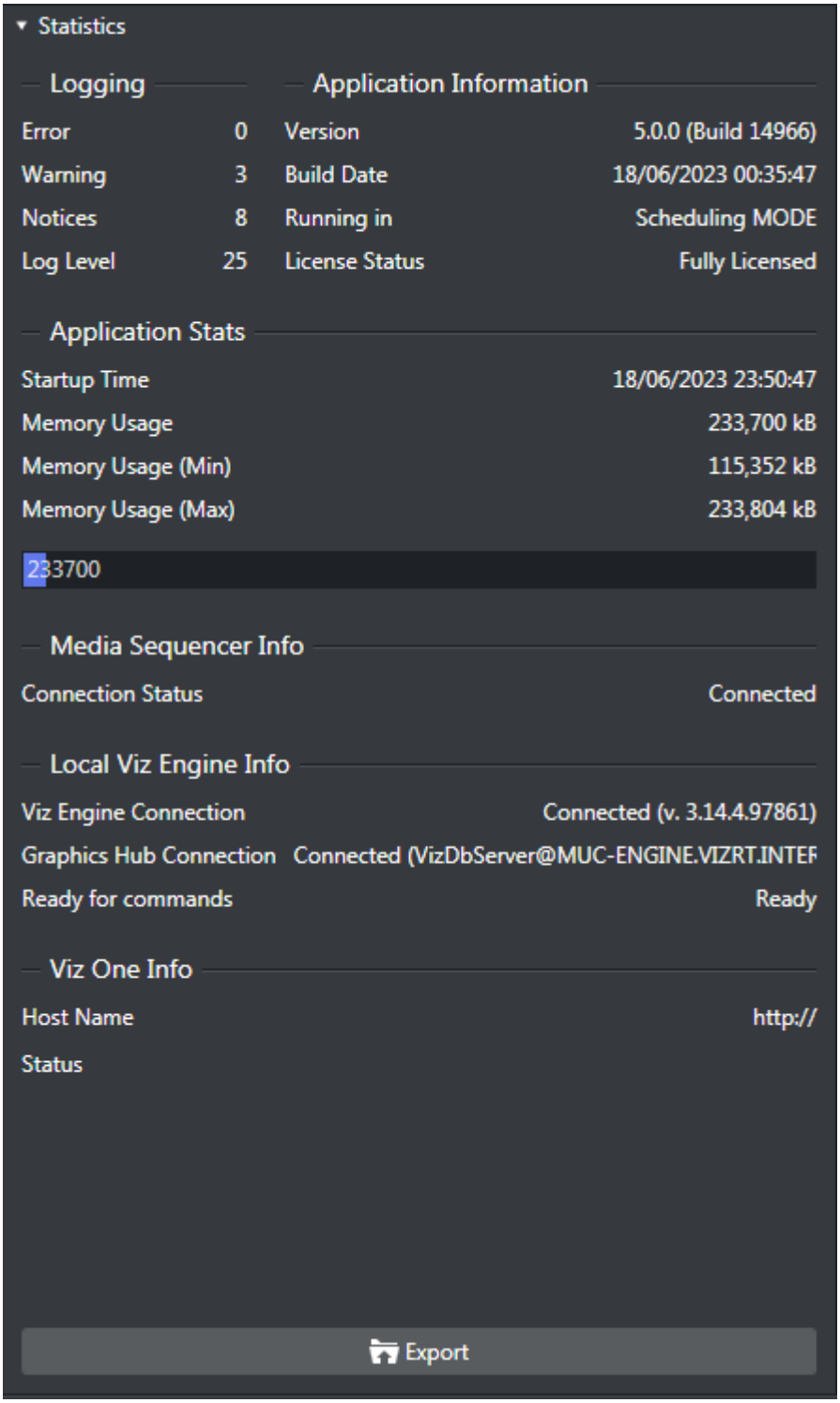
	Take **out** the Video Only Page.
	Import a Viz Ticker definition file (XML).
	Export a Viz Ticker definition file (XML).
	Import Actions definitions (from XML file).
	Export Actions definitions (to XML file).
	Import Templates (and Pages) to the connected Graphics Hub.
	Export Templates (and Pages) from the connected Graphics Hub.
	Enables the mechanical bypass on the Matrox video card. This effectively bypasses any rendering and stops Viz Engine from impacting the video signal. **Important:** For this to work, Viz Engine must be properly configured. See the Matrox and Watchdog sections in the Viz Engine Administrator Guide.
	Disables the mechanical bypass on the Matrox video card. This effectively bypasses any rendering and stops Viz Engine from impacting the video signal. **Important:** For this to work, Viz Engine must be properly configured. See the Matrox and Watchdog sections in the Viz Engine Administrator Guide.

Copyright © 2024 Vizrt

4.2 Statistics Panel

The Statistics panel is an overview dashboard, seeing run-time information and general connectivity status.

Copyright © 2024 Vizrt

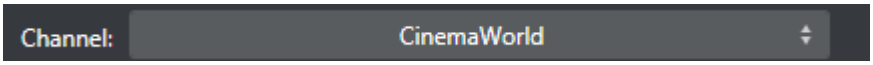


- Use the **Export** button to export statistic information to a text file.

Copyright © 2024 Vizrt

4.3 Channels

4.3.1 Channel Selection



At the top of the left pane area is the **Channel** selection drop-down menu.

- It lists your different channels and allows you to switch between them.
- You can also click on the channel icon above the drop-down list in order to switch to the corresponding channel.

See Also

- Channel Settings

4.4 Playlists

This section covers:

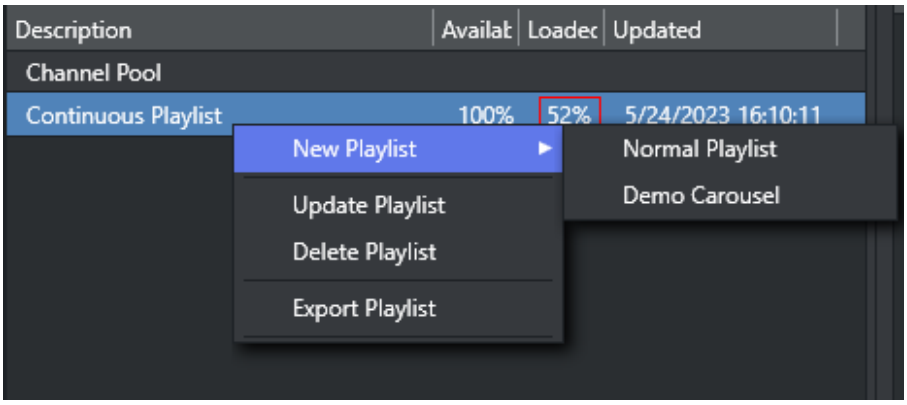
- Import a Channel Branding Exported Playlist
- Channel Pool
- Playlist Toolbar
- Playlist Keyboard and Mouse Shortcuts
- Playlist Columns
- Playlist Context Menu
- Playlist Filters

The Playlists panel displays the playlist(s) created for the selected channel. There is supports for two types of playlists types:

- Continuous playlists
- Broadcast day playlists

In addition, the panel contains a location named **Channel Pool** to store prepared non-scheduled events.

When a playlist is opened, the primary and secondary events are displayed in the right pane. In the Playlist pane you can view all the playlists you have for a specific channel.



Right-clicking on a playlist in the left pane displays a context menu of playlist commands:

- **New Playlist:** Opens a sub-menu where you can select which blank playlist to create.
  - **Normal Playlist:** Creates a blank *Broadcast Day* or *Continuous* playlist.
  - **Demo Carousel:** Creates a blank non-timed playlist intended for quick demos or simple play-out.
- **Update Playlist:** Opens a dialog to select a schedule file to Update Playlist using the Playlist Importer.
- **Delete Playlist:** Deletes the playlist.
- **Export Playlist:** Exports the playlist to a native Viz Channel Branding Client XML format. You can also export the Graphics and Data as part of the export.

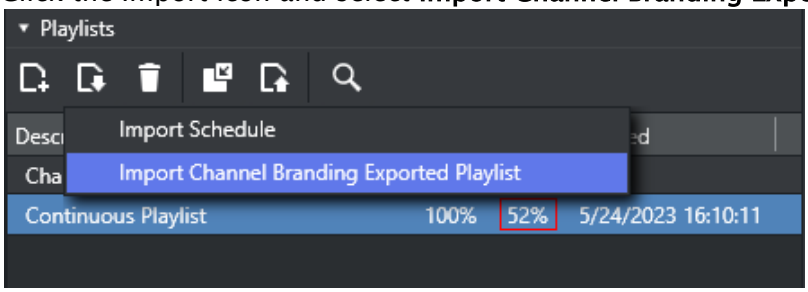
Copyright © 2024 Vizrt

4.4.1 Import a Channel Branding Exported Playlist

A playlist that has previously been exported from Viz Channel Branding Client, as an XML file can be imported back using the **Import Channel Branding Exported Playlist** menu item.
This will import the XML playlist to Viz Channel Branding without performing any mapping settings. Playlist data is set as it appears in the imported file.

> **Note:** If the broadcast date specified already exists, the imported data will not replace the existing playlist. You must first delete the existing playlist.

- Click the Import icon and select **Import Channel Branding Exported Playlist**.



4.4.2 Channel Pool

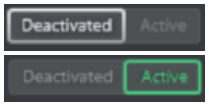
The **Channel Pool** playlist is used for those elements that may require special handling, such as a series of animated logos, an advanced trouble slide, or a specially pre-prepared legally mandated breaking news event. These elements can be accessible to and used by external control applications in order to trigger elements that are not scheduled in a normal playlist, for example.

Pages added to the Channel Pool will have the page name set as the Media ID by default. The Media ID can be changed manually by editing the **Media ID** column (individually for each element). Enable editing by checking the **Edit Events** option for Media IDs in **Mapping Settings**.

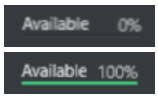
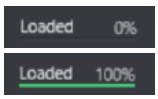
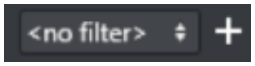
A separate channel pool is added for each channel.

4.4.3 Playlist Toolbar

The following table describes the different buttons, modes and status icons available to the playlist.

Icon	Description
	When green, the playlist is running. When gray, the playlist has stopped.
	When a playlist is **active** it is added to the Media Sequencer, which starts checking that all resources (such as videos and imagery) are available (see **Available status**) to Viz Engine for playout. This means that videos are transferred from Viz One to Viz Engine, for example.

Copyright © 2024 Vizrt

Icon	Description
	**100% available** means that full screen video clips are available to Viz Engine for playout.
	**100% loaded** means that all pages (the actual graphics including embedded video clips and imagery) are loaded and ready for playout.
	Enables you to **run** and **stop** the playlist. Note that in order to run the playlist successfully your elements must have a begin and end time.
	Searches the description field of your playlist elements. To see the next result, press Enter.
	Predefined filters allow you to filter the playlist. Click the funnel icon to create, edit or delete filters. See **To add a playlist filter**.
	Takes the selected playlist element (or group of elements) on air.
	Continues the animation of the playlist element.
	Takes the current playlist element off air.
	Initializes the playlist's pages on the preview and program renderer. When complete, the loaded status is set to 100%. See the **Playlist Context Menu** on how to initialize the playlist and individual elements from the context menu.
	Refreshes the data in use by the pages.
	Performs a Playlist Continue if the playlist is in a Hold state due to playing an element with no hard set duration, such as a LIVE show. Will also perform a Refresh of data upon continuing playout.

Copyright © 2024 Vizrt

4.4.4 Playlist Keyboard and Mouse Shortcuts

Function	Key	Mouse
Move one or more secondary events to another primary event (up or down)	Drag and drop. Combine mouse-drag with Ctrl to prevent the playlist from auto-scrolling while dragging.	
Move selected entries to previous group in the playlist	Alt + Arrow Up	
Move selected entries to next group in the playlist	Alt + Arrow Down	
Collapse the group	Arrow + Left	Double-click left mouse button
Expand the group	Arrow + Right	Double-click left mouse button
Expand all groups	Ctrl + Numpad +	
Collapse all groups	Ctrl + Numpad -	
Select all events	Ctrl + A	
Open search	Ctrl + F	
Select individual events	Ctrl	Left mouse button
Remove the selected event(s)	Delete	
Make an event’s Description field editable	F2	Double-click left mouse button on the Description field
Move the focus to the top of the playlist	Home	
Move the focus to the end of the playlist	End	

Copyright © 2024 Vizrt

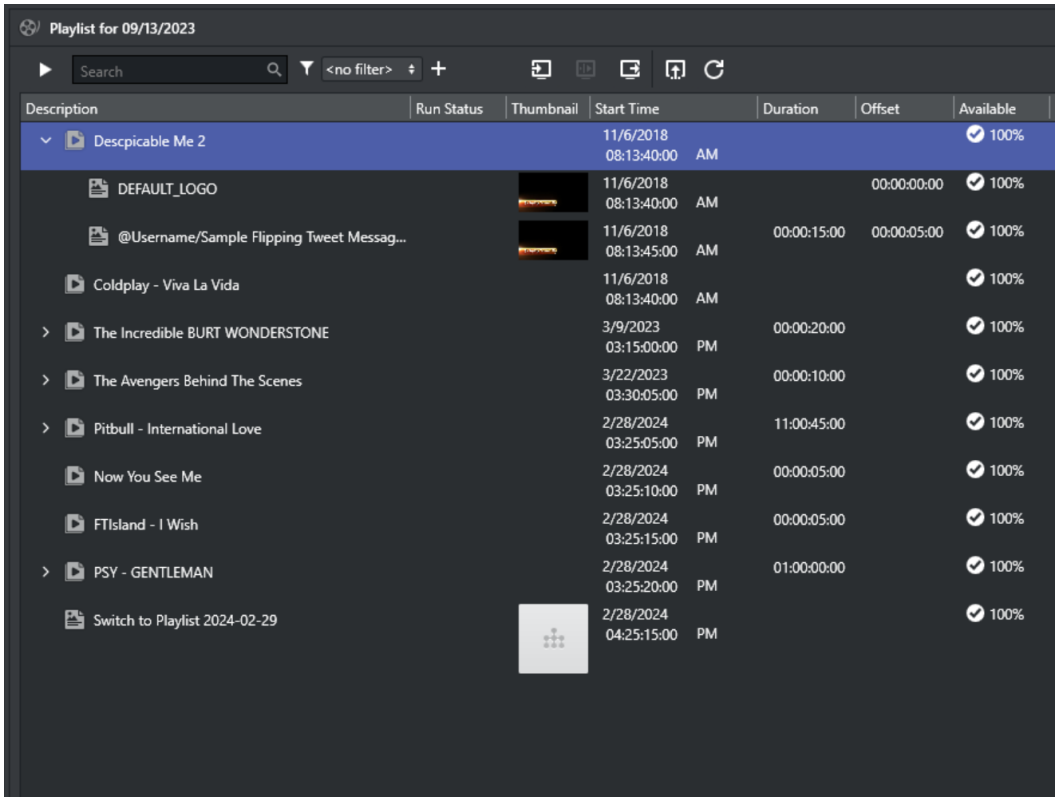
Function	Key	Mouse
Move the focus one section down	Page Down	
Move the focus one section up	Page Up	
Select events (up or down)	Shift	Left mouse button
Display the context menu		Right mouse button

4.4.5 Playlist Columns

When the right pane displays the playlist, all events along with pages and actions scheduled together are displayed.

To edit the playlist columns

1. Check the **Edit Events** check-box in your mapping table (see **Table Columns**) for each column.



2. The layout and visibility of the playlist columns can be customized by drag-and-drop and by right-clicking one of the columns to select or de-select them.
   Secondary events that have been edited will have a color indicator in the left side of the edited row.

Copyright © 2024 Vizrt

Note: If there are any errors found in the playlist, the primary element(s) causing the error will be shown in red. Hover your cursor over the red no-entry icon of the error element(s) for more information.

The following default columns are available, by default order:

- **Description**: Shows a description of the element. An **arrow-head** next to a description indicates other elements are group together with the event. This may be a primary event with pages and/or actions scheduled for it as secondary events. Click the **arrow-head** to expand and show the secondary events.
- **Thumbnail**: Shows a snapshot/thumbnail of the template used in a Secondary Event.
- **Start Time**: Shows the element's begin time.
- **Offset**: Shows the offset of the secondary event relative to its primary event.
- **Duration**: Shows the duration of the element.
- **Available**: Shows if a video clip is available to your playout device.
- **Loaded**: Shows the loaded status of pages on the Viz Engine program device channel.
- **Info I.H.**: Displays timestamp and action-annotation for any events added or updated via Integrations Hub
- **Info P.S.**: Displays timestamp indicating most recent moment the secondary event's Page Content Value was evaluated
- **Page Content Value**: Shows a generated summary of the current values for Page Field's, no-matter if manually set or grabbed from dynamic sources via Promo Server.

Other columns, not shown by default, in default order:

- **End Time**: Calculated moment in time (Start Time + Duration) a Primary Event will end.
- **Run Status**: Holds the current run-information for this event as annotated by Media Sequencer.
- **Take Count**: Number indicating how many times this event has been taken since playlist was last Activated.
- **Event Id:** The internal id for this event, as addressable via Integrations Hub REST API or seen in VDom in Media Sequencer.
- **Trigger Id**: Shows the string-value to use as look-up property or trigger id for remote execution via Media Sequencer (depending on protocol used).
- **P. page**: 'Page pool page' reference, so what page this originated from in the page pool.
- **Cf. page**: 'Copied from page' reference, indicates the page that this page was cloned from, will differ from *P. page* if this page was copy-pasted from one location to another in the playlist by an operator.
- **Layer**: Shows the value this element will run in, either a hard layer (annotated with [ ]) or a virtual TL layer.
- **Concept**: Current or override setting as what Concept to use as default playout if no Concept is set in Playout Configuration dialog.
- **Mark In**: Special start-time marker for video clips.
- **Mark Out**: Special End-time marker for video clips.
- **Ignore**: Indicates if a flag is set to skip over this event for evaluation of content for Page Content Filling. Can hold values True or False.

Copyright © 2024 Vizrt

- **Created By**: Annotates what component or method that created this element. An entry saying 'manual' indicates it was done by hand via the Channel Branding Client.
- **Template Name**: Shows the original master template name that this page relies upon.
- **Template Description**: Shows the description as set on the master template in the template pool, either default from Viz Artist or hand-written in template pool.

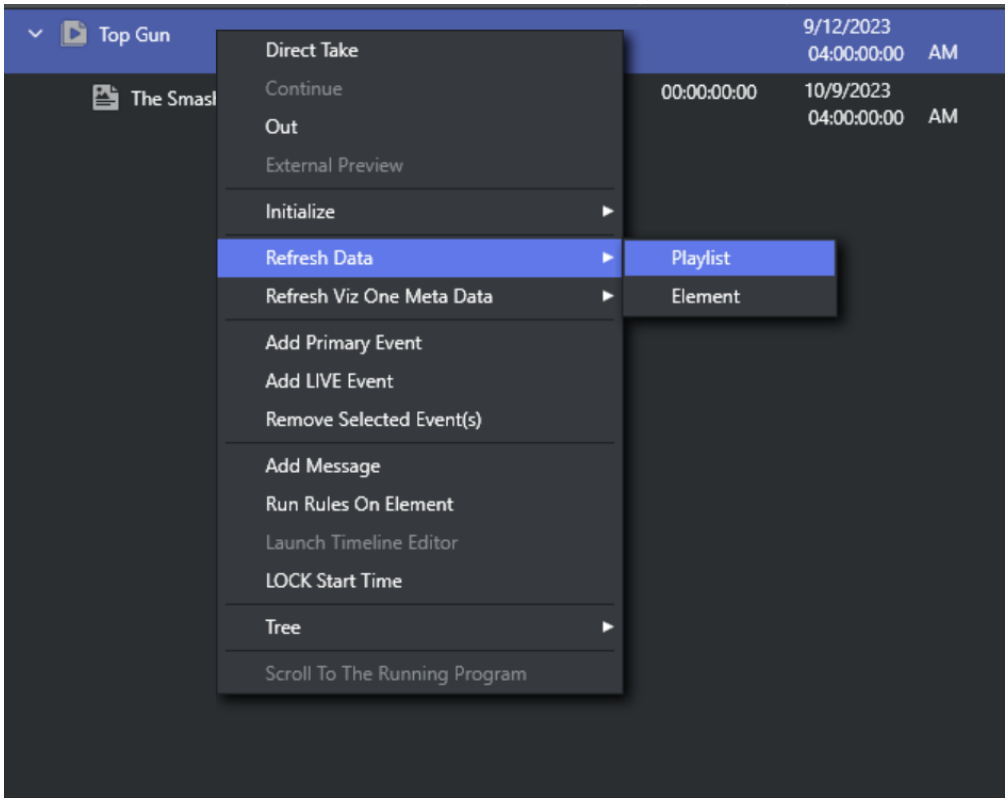
Elements in the playlist that are **disabled** are shown with a darkened style, as shown in the screenshot below:

4.4.6 Playlist Context Menu

Right-clicking the playlist opens the following context menu:

- **Direct Take**: Takes the playlist element directly to air. If the element is available and/or loaded it will play immediately. If not, the element will be taken on air as soon as it is.
- **Continue**: Continues the animation of the graphics if the graphics has a continue point.
- **Out**: Takes the element off air (i.e. hard cut).
- **External Preview**: Previews the element on your configured preview channel (see section *Configuring Device Channels* and *Configuring Playout Devices*).
- **Regenerate payload**: *Only* for secondary events. Option to Forcefully regenerate the payload for the element.
- **Enable** or **Disable Element**: *Only* for secondary events. Allows you to either enable or disable a secondary element.
- **Initialize** (see section *Playlist Toolbar*).
  - **Playlist**: Initializes the entire playlist.
  - **Element**: Initializes the selected playlist element.
- **Refresh Data:** Request a refresh of automatic page content or primary event Viz One metadata, as shown in the screenshot below:

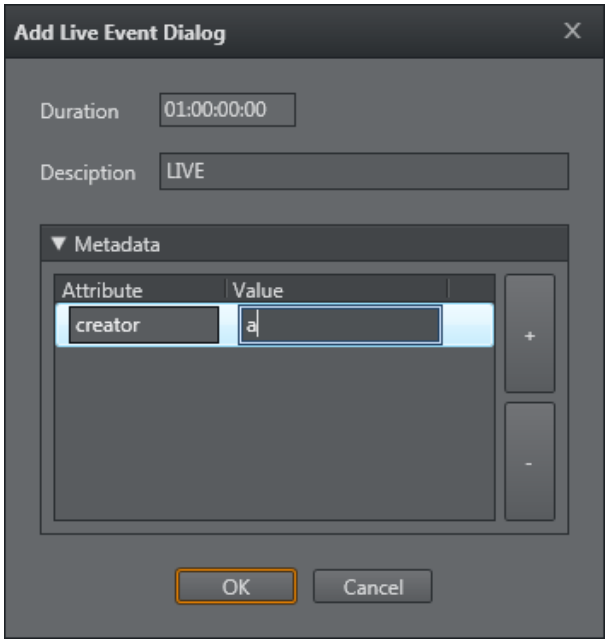
Copyright © 2024 Vizrt



  - **Playlist**: Clicking here iterates through the entire playlist and update all elements.
  - **Element**: Clicking here updates page data or Viz One metadata for the selected element of the playlist only.
- **Refresh Viz One Meta Data**: forcefully re-read meta data from this event in Viz One, if applicable, and generate Event Meta Data from it.
  - **Playlist**: Do this for full every event in the playlist.
  - **Element**: Do this for this single event.
- **Add Primary Event**: Adds a primary event to the playlist.

Copyright © 2024 Vizrt

- **Add Live Event**: Adds a live event.



- **Remove Selected Event(s)**: Removes one or several selected events from the playlist. This action cannot be undone.
- **Add Message**: Add a Message Event, that is in-playlist annotations that have no impact on the playout - only intended to share information with other operators.
- **Run Rules On Element**: enabled for group and video primary events only. When selected, defined Rules will be run for the selected primary element only.
- **Launch Timeline Editor**: Launches the Timeline Editor, see Working with Video Clips.
- **LOCK Start Time**: Lock the event. Note that you can modify live or locked events in the playlist column.
- **Tree**:
  - **Select All**: Selects all events.
  - **Select Inverse**: Inverts your current selection, selecting all other elements.
  - **Select None**: Cancels all selections.
  - **Expand All**: Expands all primary events to show secondary events.
  - **Collapse All**: Collapses all primary events to hide secondary events.
  - **Hide Empty Groups**: Hides all empty groups.
  - **Wrap Text**: Wraps the text according to column width.
  - **Font...**: Allows you to set your own system font for the playlist.
- **Scroll To The Running Program**: Scrolls the playlist view to show the current running primary element at the top.

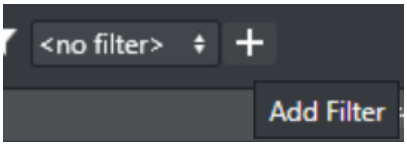
4.4.7 Playlist Filters

The Playlist lets you add filters to quickly sort the list of events.

Copyright © 2024 Vizrt

Adding a Playlist Filter

To add a playlist



1. Click the **funnel** icon, and click **New...**.
    - This opens the Edit filter window.
2. Enter a **Filter name**.
3. Select whether the filter should apply if:
    - All conditions match.
    - One of the conditions matches.
    - None of the conditions match.
4. Select whether or not to Filter on Group.
5. Continue to define your filter by adding one or multiple conditions. The list of events is filtered as you define your conditions, allowing you to verify that the filter is configured correctly.

Managing a Playlist Filter

To manage a playlist filter

1. Select your filter.
2. Click the **funnel** icon, and from the menu click **Edit...** or **Delete**.

Copyright © 2024 Vizrt

4.5 Templates, Pages And Pilot Data Elements

In the **Templates** and **Pages** panels, you can view all imported templates and created pages respectively. You can also filter your templates and pages view and add content to pages using the Page Content Filling Window.

4.5.1 Templates

Templates (and the pages created from templates) can have fields or elements that fill in based on the current playlist data. This will typically be information such as “What is the name of the next film?” or “When does the next film start?”.

4.5.2 Pages

Pages are graphic elements ready for playout that can be manually or automatically added to the playlist. Pages are created from Templates that a graphic artist creates in Viz Artist.

4.5.3 Pilot Data Elements

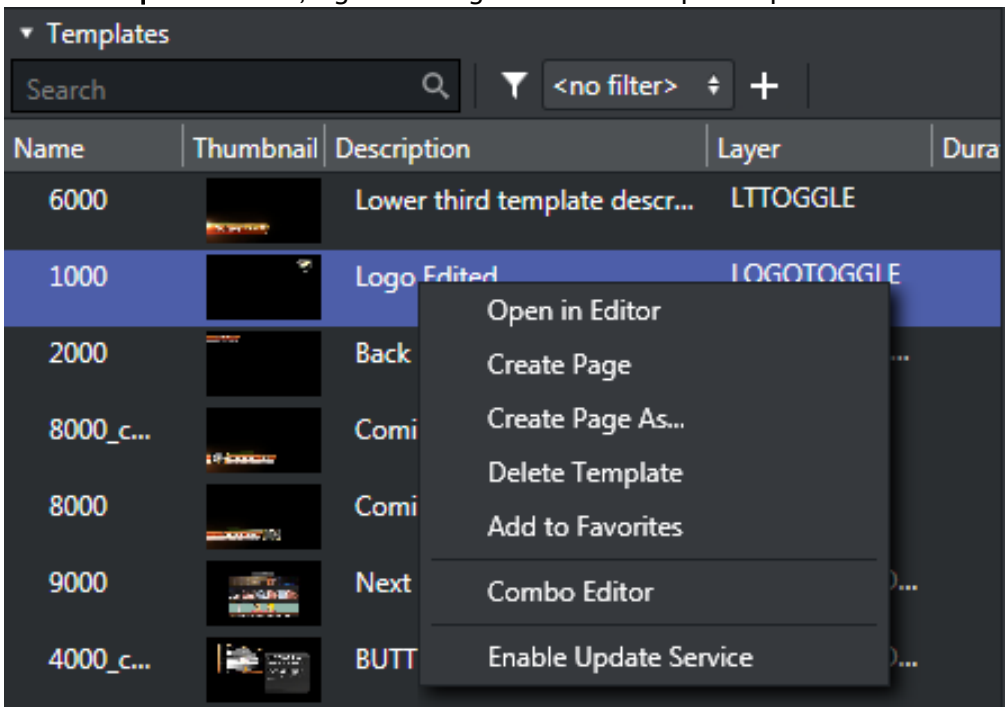
In the Pilot Data Elements panel all locally known Viz Pilot elements are shown, to be used for manual drag into a playlist or to be used by Rules. Even if an element isn't yet known by the local Media Sequencer but exists in the Pilot Database it can be referenced and used by a channel and will be added to the local list once the playlist has been imported.

- Templates
- Pages
- Pilot Data Elements
- Template Context Menu
- Pages Context Menu
- Template and Page Filters

Copyright © 2024 Vizrt

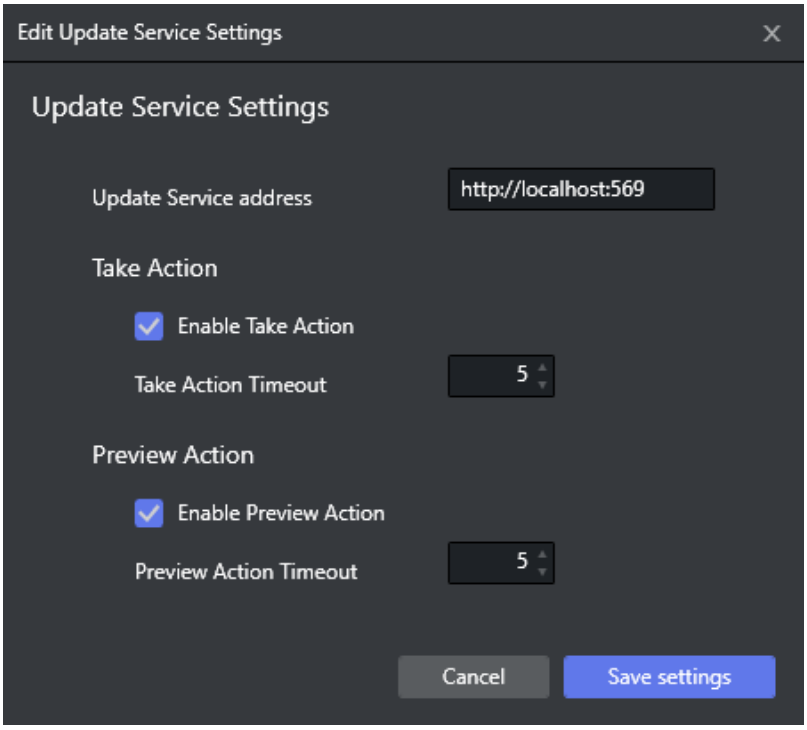
4.5.4 Template Context Menu

- In the **Templates** menu, right-clicking a selected template opens a context menu:



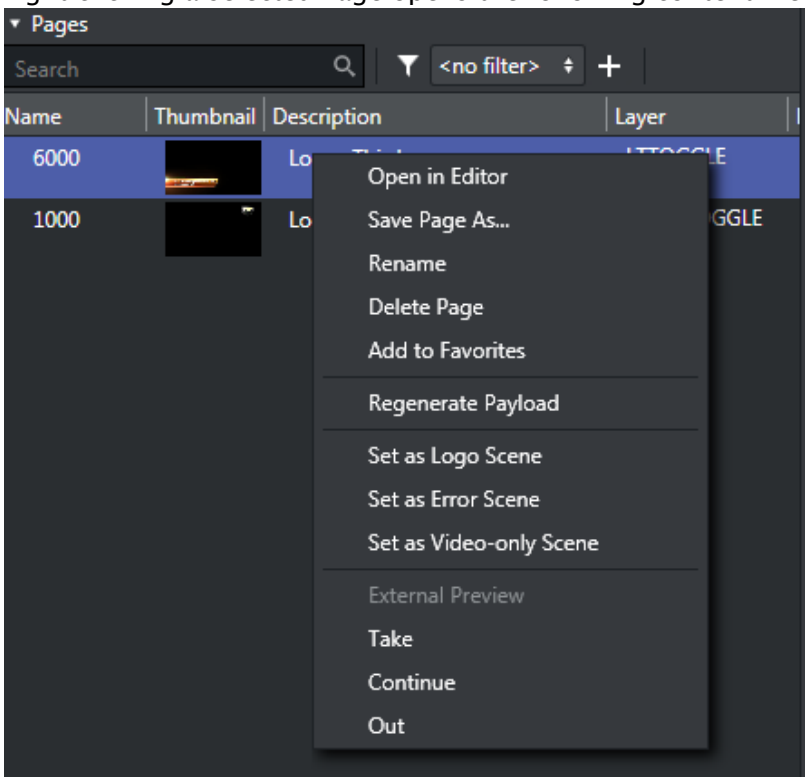
- **Open in Editor**: Opens the template editor which allows you to create a new page based on the template.
- **Create Page**: Creates a page based on the selected template using the same name as the template.
- **Create page As**: Create a page with a given new name.
- **Delete Template**: Deletes the selected template/data.
- **Add To Favorites**: Adds the template to your pre-defined Favorites filter.
- **Combo Editor**: Opens the Combo (short for Combination) template editor where you can change templates based on multiple Transition Logic (TL) scenes.
  For more detailed information, see section Combo Page Editor.
- **Enable Update Service**: Enables you to assign an Update Service for the selected template.
- **Disable Update Service**: Disables the Update Service for the selected template.
- **Edit Update Service**: A pop-up that enables editing the Update Service for the selected template (by setting the service URL).
  A Timeout on the available commands (Take and Preview) can be set. This timeout delays the given command sent by Media Sequencer to the Viz Engine.

Copyright © 2024 Vizrt



4.5.5 Pages Context Menu

- Right-clicking a selected Page opens the following context menu:



Copyright © 2024 Vizrt

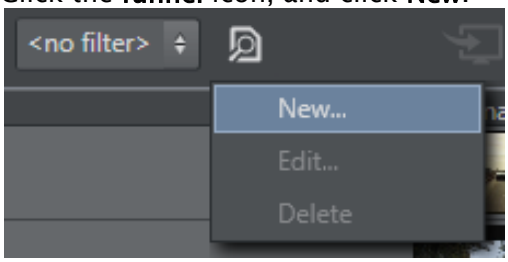
- **Open in Editor**: Opens the Template Editor for updating the page or saving it as a new page, based on the same template.
- **Save Page as**: Creates a page based on an existing page. The page can be opened in the Template Editor for editing.
- **Rename**: Changes the name of the selected page. Duplicate page names are not allowed.
- **Delete Page**: Deletes the selected page(s).
- **Add to/Remove from Favorites**: Adds or removes the page from the pre-defined Favorites filter, see Templates, Pages and Pilot Data Elements.
- **Regenerate Payload**: Regenerate the internal payload as set via Page Editor or Page Content Filling
- **Set as Logo Scene**: Sets a selected page as the default Logo Scene that can be controlled from the toolbar.
- **Set as Error Scene**: Sets a selected page as the default Error Scene that can be controlled from the toolbar.
- **Set as Video-only Scene**: Sets the selected video scene as the default video-only scene that can be controlled from the toolbar.
- **External Preview**: Shows the scene on your defined preview Viz Engine.
- **Take**: Immediately takes the page on-air on your program Viz Engine.
- **Continue**: Continues the page on-air on your program Viz Engine.
- **Out**: Immediately takes the page off-air (hard cut) on your program Viz Engine.

4.5.6 Template and Page Filters

The **Templates** and **Pages** panels allow you to add filters to quickly sort the list of templates or pages.

To add a Template or Page filter

1. Click the **funnel** icon, and click **New**.



   This opens the **Edit filter** panel.
2. Enter a *Filter name*.
3. Select if the filter should apply when:
   - All conditions match.
   - One of the conditions matches.
   - None of the conditions match.
4. Continue to define your filter by adding one or multiple conditions.
   - The list of templates or pages is filtered as you define your conditions, allowing you to verify that the filter is configured correctly.

Copyright © 2024 Vizrt

> **Note:** There is also a built-in *Favorites* filter.

To manage a Template or Page filter

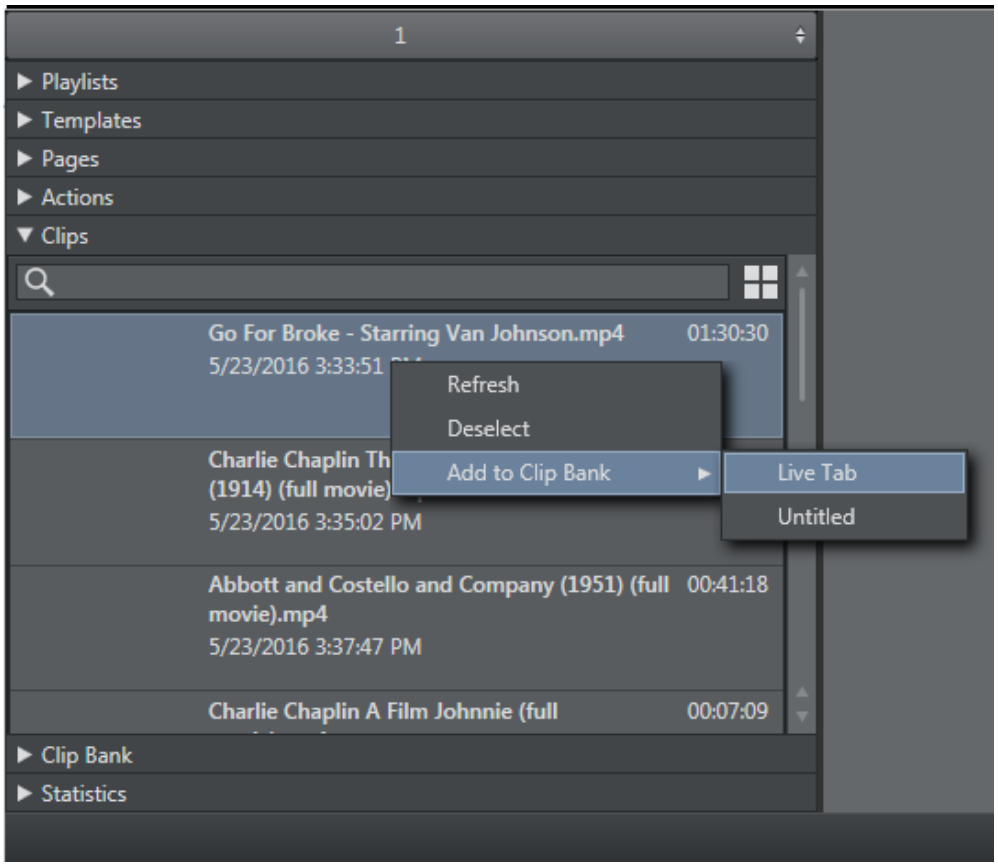
1. Select your filter.
2. Click the **funnel** icon, and from the menu click **Edit** or **Delete**.

See Also

- Template Import
- Working with Templates and Pages

4.6 Clips

Viz Channel Branding Client integrates with Viz One that enables the operator to search and browse for clips stored in the Viz One from within the client user interface. Video clips that are used in the playlist are automatically transferred to the relevant Viz Engines by Viz One. Video clips can either be added as full screens, or embedded into the graphics.



The **Clips** button opens up a search frame allowing you to search for video clips on Viz One.

The search field allows you to use common search operations such as:

- Search for a word: soccer
- Search for an exact word or phrase: “world football championship”
- Search for either word: soccer OR football
- Search for both words: soccer AND football
- Search using an asterisk (*) to “fill in the blanks”: foot*, *ball*, *ball

4.6.1 Clips Context Menu

Right-Click on Clips to show the context menu providing these options:

- **Refresh**: Reload all clips from Viz One.
- **Deselect**: Deselect a clip.

Copyright © 2024 Vizrt

- **Add to Clip Bank**: Add the selected Viz One clip into the Clip Bank. You can select which clip bank tab the clip should be added to.

See Also

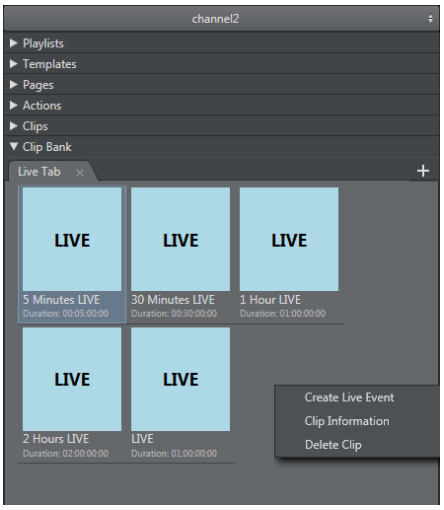
- Viz One Integration

Copyright © 2024 Vizrt

4.7 Clip Bank

Clip Bank contains LIVE events and Viz One clips. You can have LIVE events pre-created in this clip bank to allow for rapid insertion of LIVE events into the running playlist.

When you first create a channel, the default clip tab name **LIVE** is created with four default LIVE events.



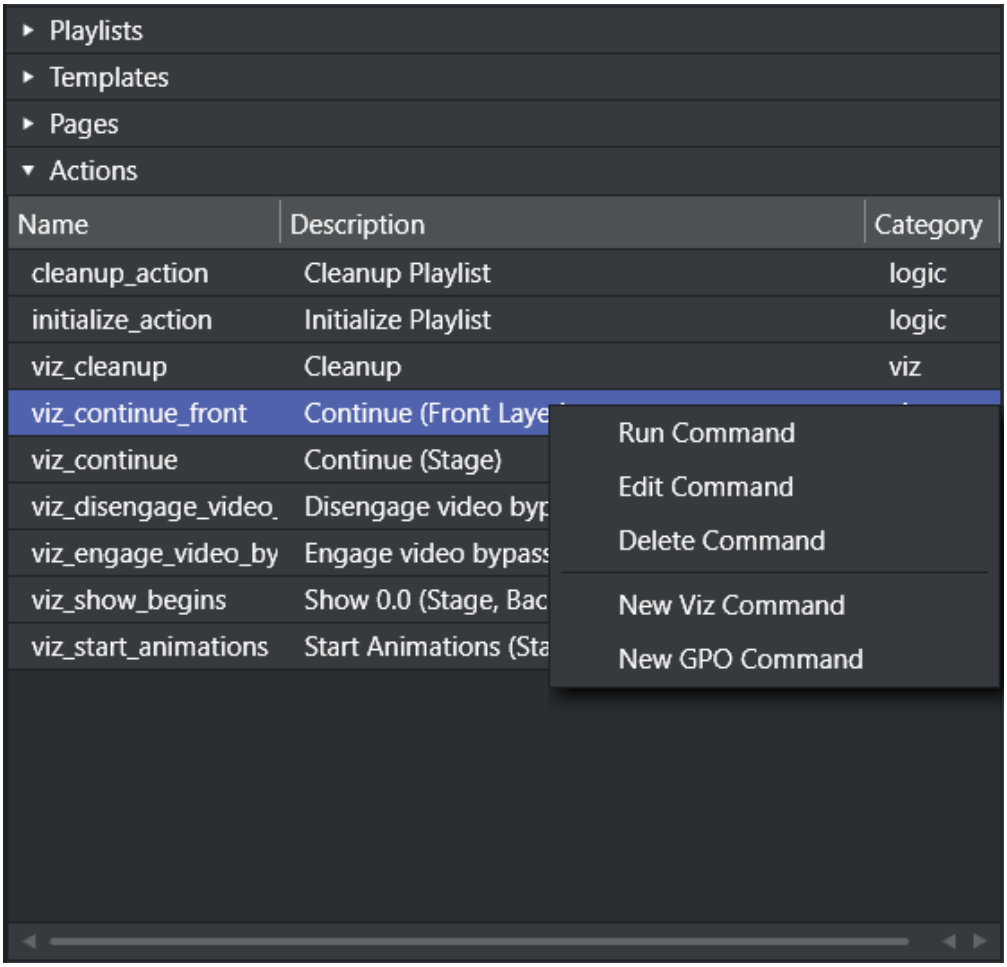
You can add a new tab by clicking the + sign at the top right of Clip Bank tab.

Right-click for the options menu where you can:

- Create Live Event
- Clip Information
- Delete Clip.

Copyright © 2024 Vizrt

4.8 Actions



The actions pane allows you to define General Purpose Output (GPO) commands, Viz Ticker commands and Viz Engine commands.

4.8.1 GPO

GPO commands are handled by Media Sequencer.

4.8.2 Viz Ticker and Viz Engine

Both Viz Ticker and Viz Engine commands can be used in Viz Channel Branding to automate certain tasks.

The commands behave as secondary events, in that you can also apply rules and schedule them in the playlist. Viz Ticker actions are accessible from the **Actions** panel seen in the left pane. These actions integrate tickers from Viz Ticker. Once the ticker has been created, it can be imported via the Viz Channel Branding Client.

Actions can for example include:

Copyright © 2024 Vizrt

- Playing specific data elements on air.
- Clearing specific render layer.
- Sending Stop, Continue, Take and other such commands to a specific render layer.
- Associated with playlists as secondary events.

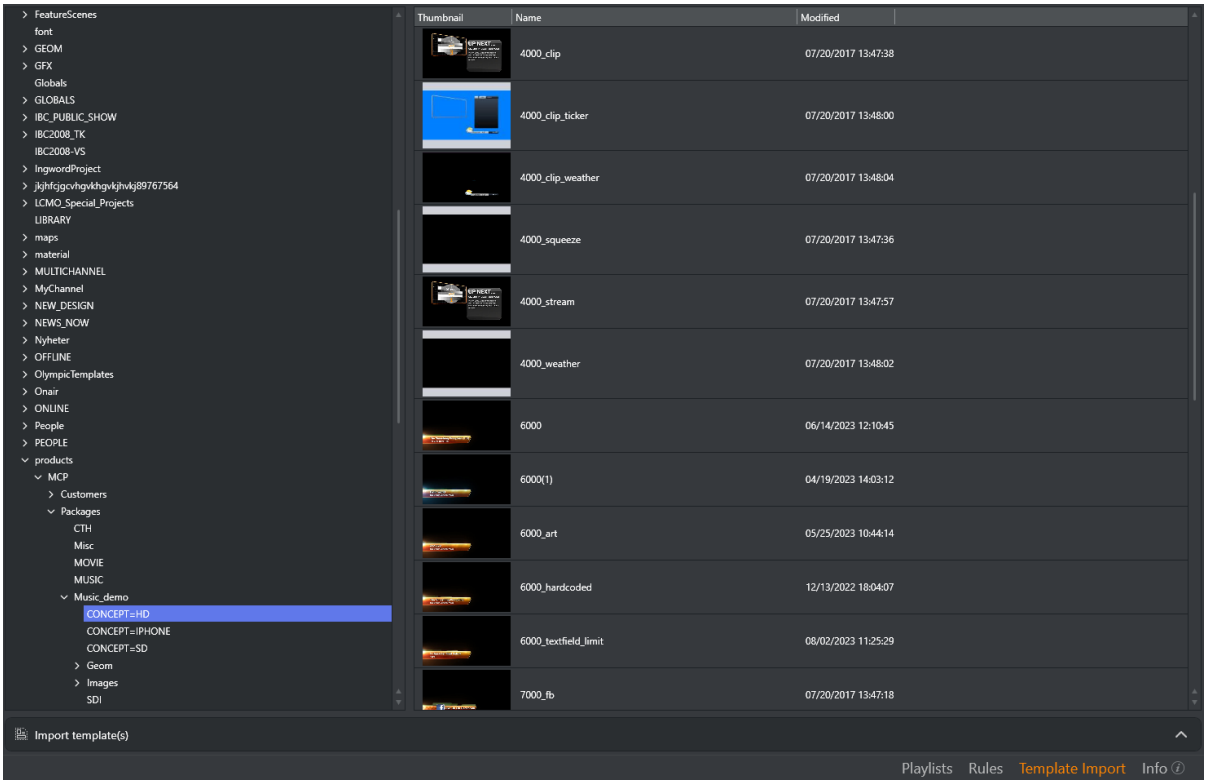
> **Note:** Media Sequencer (MSE) can only handle one ticker at a time. When working with more than one channel in the Traffic Department Workflow the Ticker actions are *virtual*, meaning that you can schedule them in playlists, but not actually preview them.

See Also

- Working with Actions
- Ticker Actions
- Rules
- Working with Rules

Copyright © 2024 Vizrt

4.9 Template Import



- Click the **Template Import** button on the lower right to view the contents of your Viz Graphics Hub and select scenes to import.
- Double-click scenes to import or select several and right-click and select **Import Scene to Channel**.

> ⚠ **Note:** You must have a working Viz Graphics Hub connection in order to view the content. For information on how to configure your local Viz Engine Database connection settings, see the Viz Engine Administrator Guide.

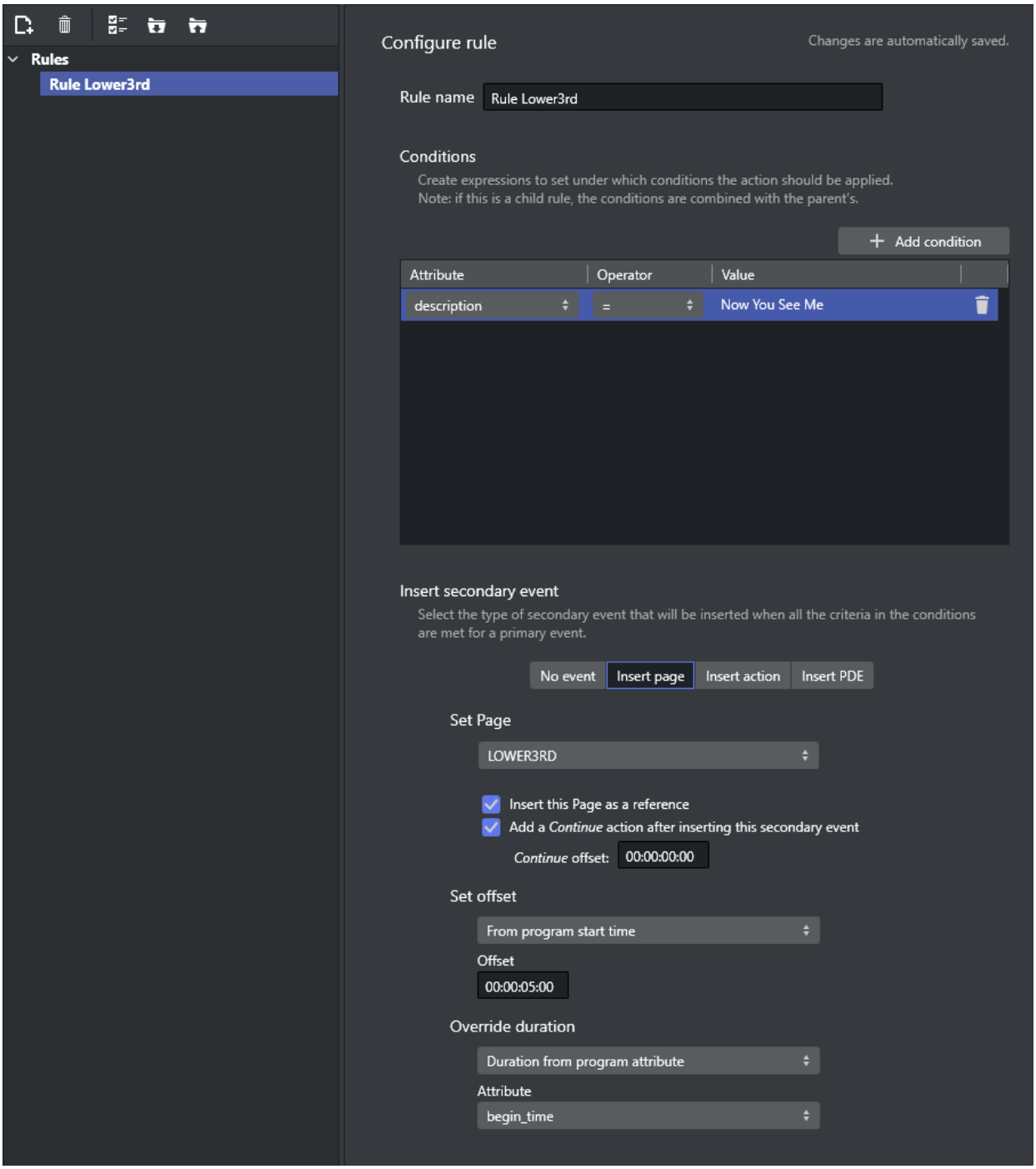
See Also

- Templates, Pages and Pilot Data Elements
- Working with Templates and Pages

Copyright © 2024 Vizrt

4.10 Rules

The possibility to predefine rules can describe a variety of playlist rules, such as adding graphics with a relative offset to a primary event with a predefined length.



This is the area where you set up playlist rules. Clicking the **Rules** button shows all current rules in the right panel.

Copyright © 2024 Vizrt

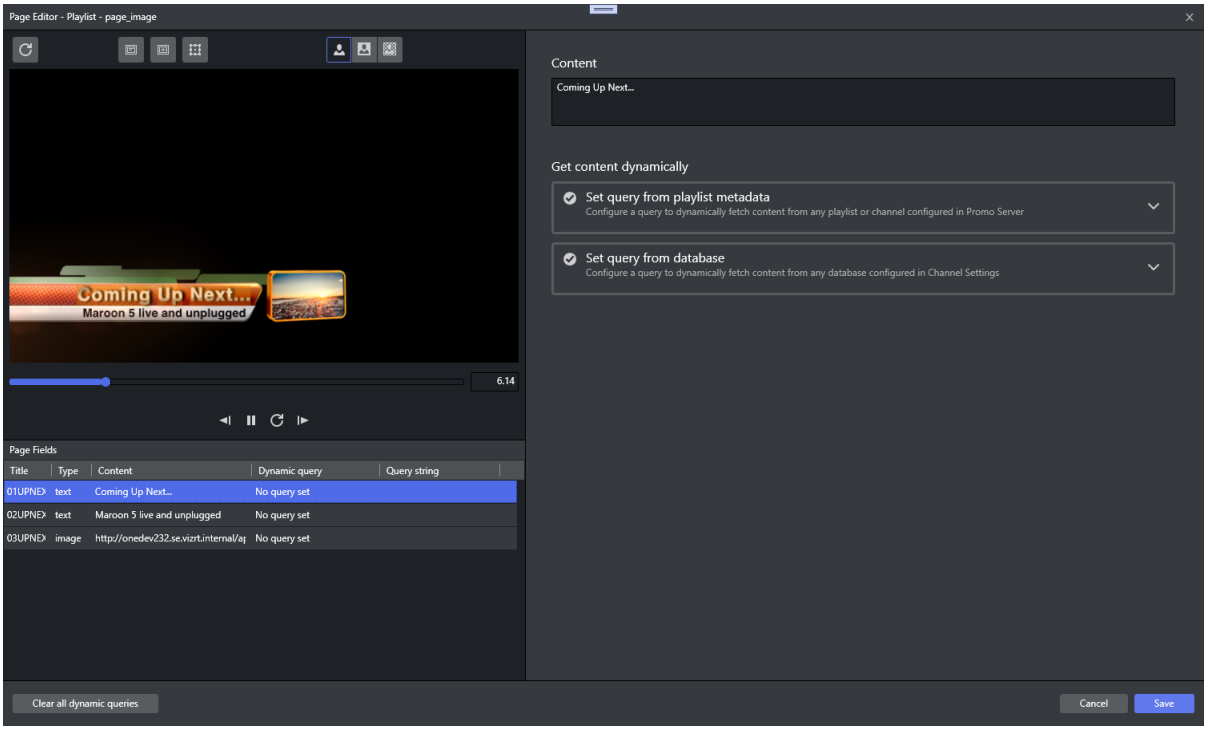
- **Rule name:** Allows you to define a name for the rule. This can help differentiate among several rules applied to a single page or action.
- **Conditions:** Lists the conditions that must be met for an event to happen.
- **Insert page - Set Page:** Sets the page from page pool on which the rule will be applied.
    - **Insert this Page as a reference:** If checked, inserts the page as a reference (see Referenced Pages).
- **Insert action - Set Action:** Sets the action from action pool on which the rule will be applied.
- **Insert PDE - Set Pilot Data Element:** Sets the Pilot Data Element on which the rule will be applied.
- **Add a Continue action after inserting this secondary event:** Allows you to select the check box if the page (the graphics) has been designed with a stop point, and a Continue action is needed for full execution. By enabling this, you can set the offset time for the Continue action, if required by the previous step.
- **Set Offset:** Allows you to set the offset time. There are 4 ways to set offset:
    - **From program start time:** Sets offset relative to the program start time. The format is *hh:nn:ss:ff*.
    - **From program end time:** Sets offset relative to the program end time. The format is *hh:nn:ss:ff*.
    - **From % of program duration:** Sets the offset relative to the program duration, as a percentage. Selecting this option enables a slider that allows you to configure the position in the program when you want the secondary event to be added.
    - **From event attribute:** Sets the offset from event attribute.
        - **Additional offset:** You can set additional offset based on the value of the selected event attribute. When **Positive** is checked, **Additional offset** is added to the value, otherwise, it is subtracted from the value. Minimum offset is 00:00:00:00.
- **Override duration:** This section only exists in **Insert page**. Sets the duration for the page in the following ways:
    - **Do not override:** Does not set the duration.
    - **Specific duration value:** Sets the duration to a specific value. The format is *hh:nn:ss:ff*.
    - **Duration from program attribute:** Sets the duration to the value of the selected event attribute.

See Also

- Working with Rules
- Actions
- Working with Actions
- Ticker Actions

Copyright © 2024 Vizrt

4.11 Page Editor



The Page Editor allows editing of secondary events such as graphics and graphics with embedded video clips. The content for each page field can be manually set directly with a value using editor or dynamically set by Page Content Filling.

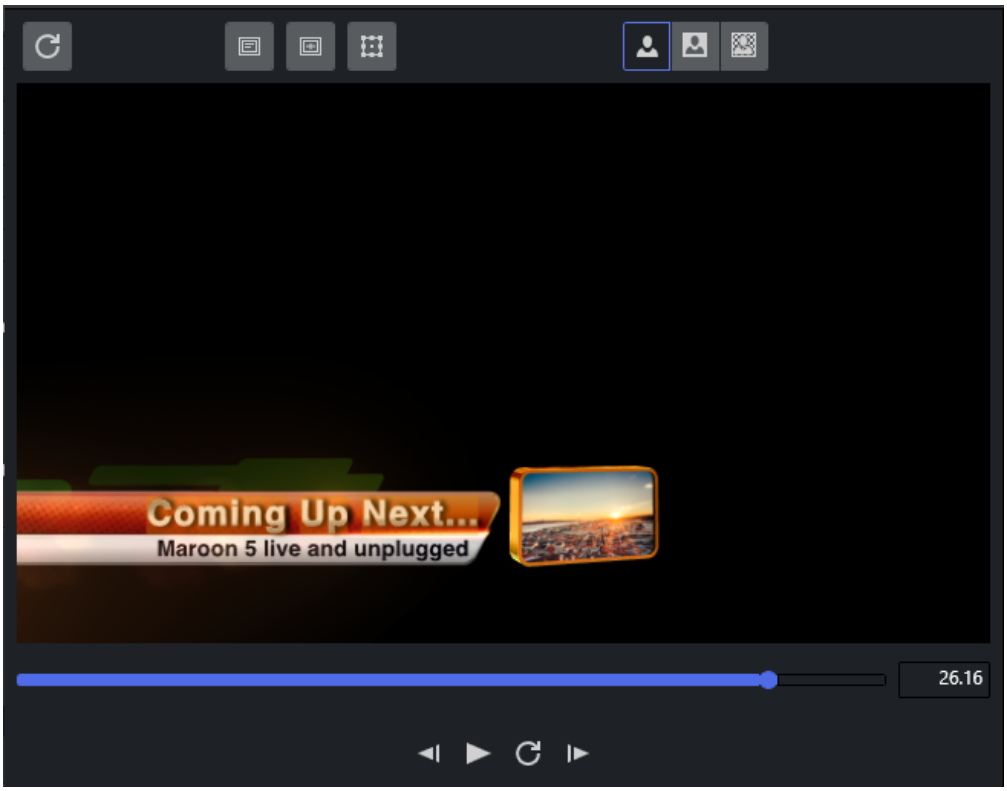
> ⚠ **Note**: Page Editor can only be opened when the local Viz Engine is successfully connected.

This section covers the following topics:

- Scene Preview Player
- Page Fields List
- Page Fields Editors
- General Actions

Copyright © 2024 Vizrt

4.11.1 Scene Preview Player



The Preview Player presents a representation of the graphics and video clips. In addition, it shows the defined title and safe area and the bounding boxes. Bounding boxes are related to the graphics scene's editable elements (see Page Fields List below).

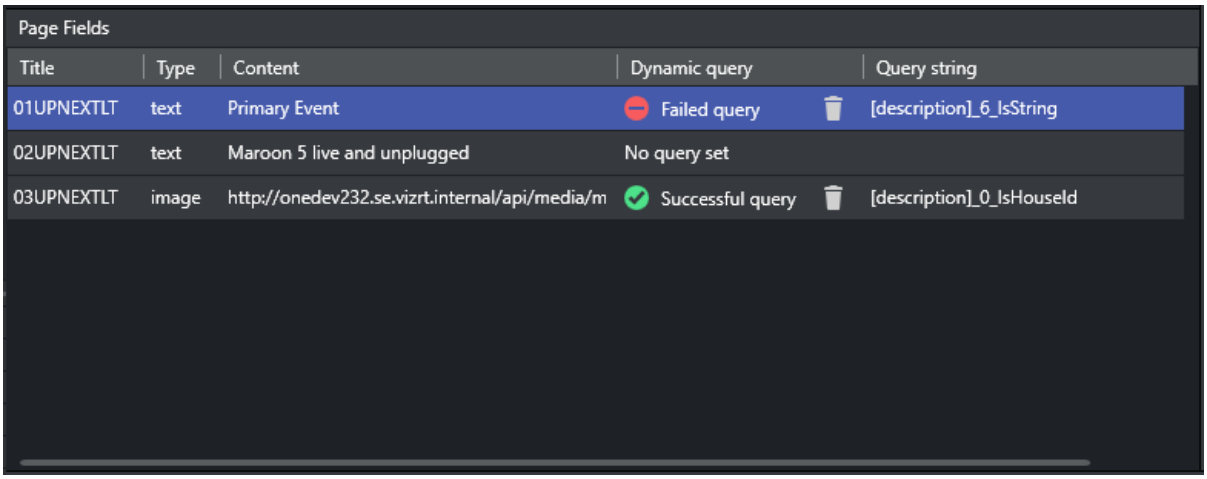
Preview window buttons

Icon	Description
	Continues or pauses the scene animation currently loaded if it contains any stop points.
	Steps back one millisecond.
	Steps next one millisecond.
	Shows or hides the Title Area in the Preview Player.

Copyright © 2024 Vizrt

Icon	Description
	Shows or hides the Safe Area in the Preview Player.
	Shows or hides the Bounding Box for the page field currently selected in the Preview Player.
	Preview RGB version of the graphics with a default background color (black). Preview the Key version of the graphics. Preview the graphics keyed on top of a still image (if the Viz Engine is so configured).
	Refreshes the scene in the Preview Player.

4.11.2 Page Fields List



The lower-left panel of the Page Editor presents available Page Fields (editable elements of the graphics) for the currently loaded Page or Template.

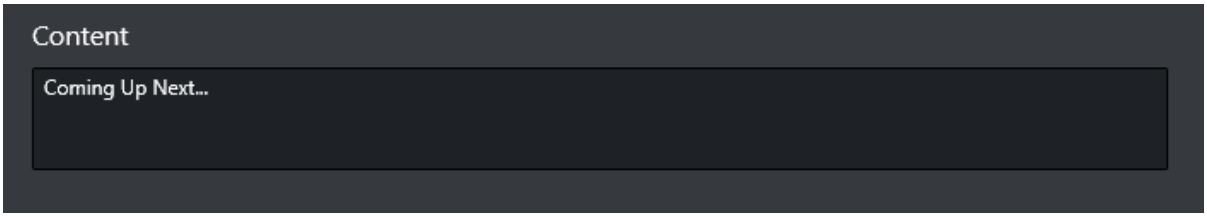
- Click on a page field to display the available options.
  - **Title:** Name of the field.
  - **Type:** Field characteristic, like text, rich-text, image, video, clock, etc.
  - **Content:** The final content of the field.
  - **Dynamic query:** The Page Content Filling status.
    Clicking the **Trashcan** icon will remove the Page Content Filling mapping done for this field. It is necessary to press **Save** afterwards to save the page.
  - **Query String:** Shows the query string grabbing data from playlist metadata, or database query string.

Copyright © 2024 Vizrt

4.11.3 Page Fields Editors

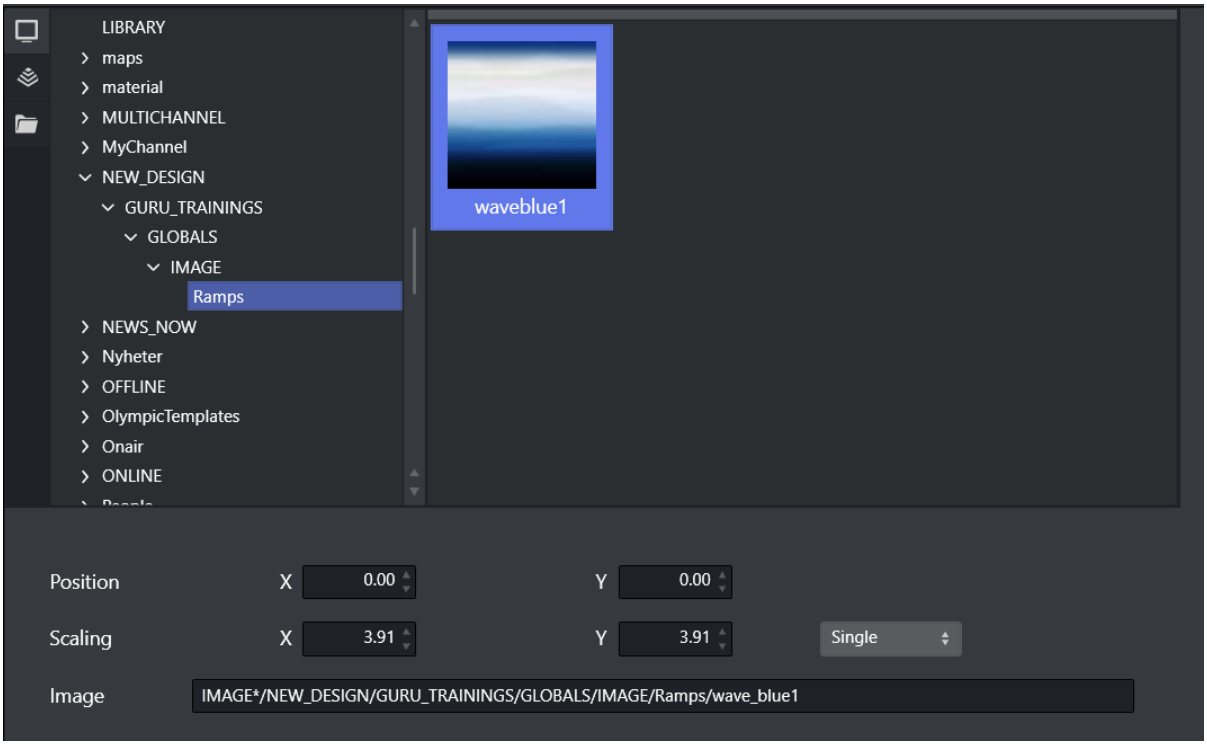
The panel on the right of the Page Editor presents various editors, corresponding to the type of the selected page field.

Text Editor



The Text Editor allows typing in the text input box and the content for the corresponding element of the graphics is updated.

Image Editor



The Image Editor presents three different browsers for selecting an image. The image path or URL is presented in the Image textbox when an image is set.

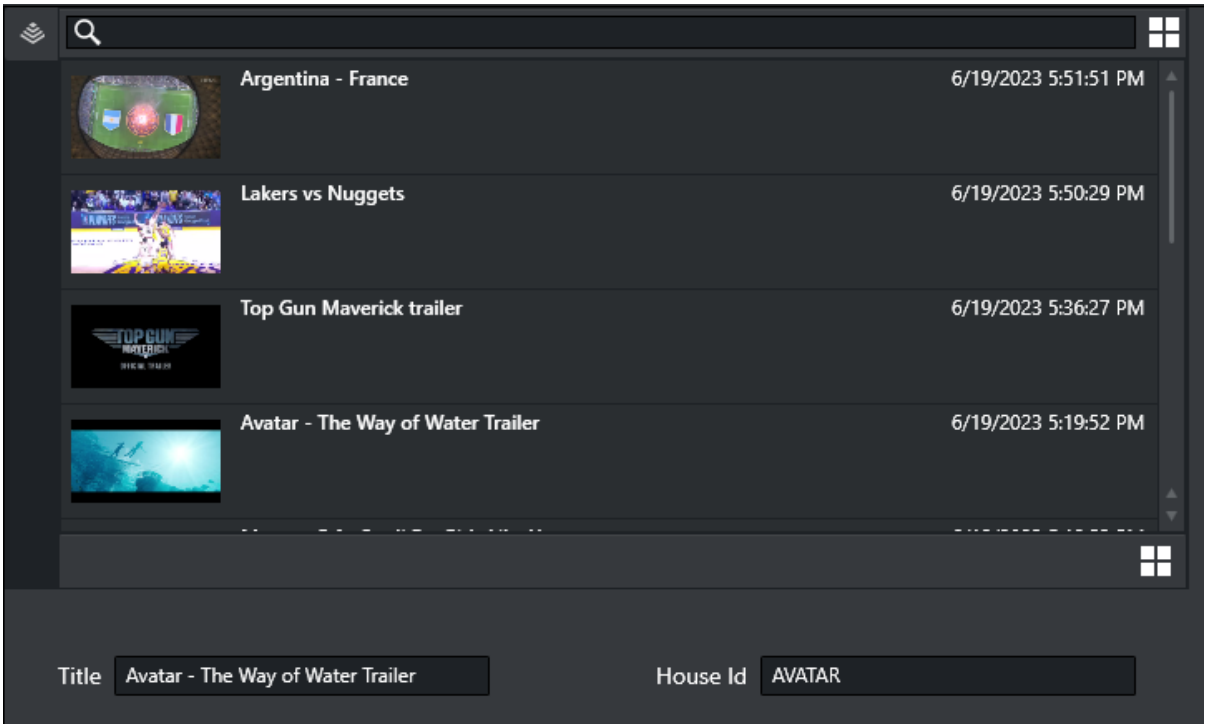
Copyright © 2024 Vizrt

- **Graphic Hub Browser:** Select an image from Graphic Hub. Graphic Hub needs to be configured in Global Settings first.
- **Viz One Browser:** Select an image from Viz One. Viz One needs to be enabled in Global Settings first.
- **File Browser:** Select an image from a local directory.

It is possible to adjust the position or scaling of an image, and the adjustment can be verified in the Scene Preview Player. There are three options when adjusting scaling of an image:

- **Linked:** x and y values update at the same time, with the same value.
- **Proportional:** x and y values update at the same time, with the initial proportion.
- **Single:** only x or y value are updated.

Video Editor



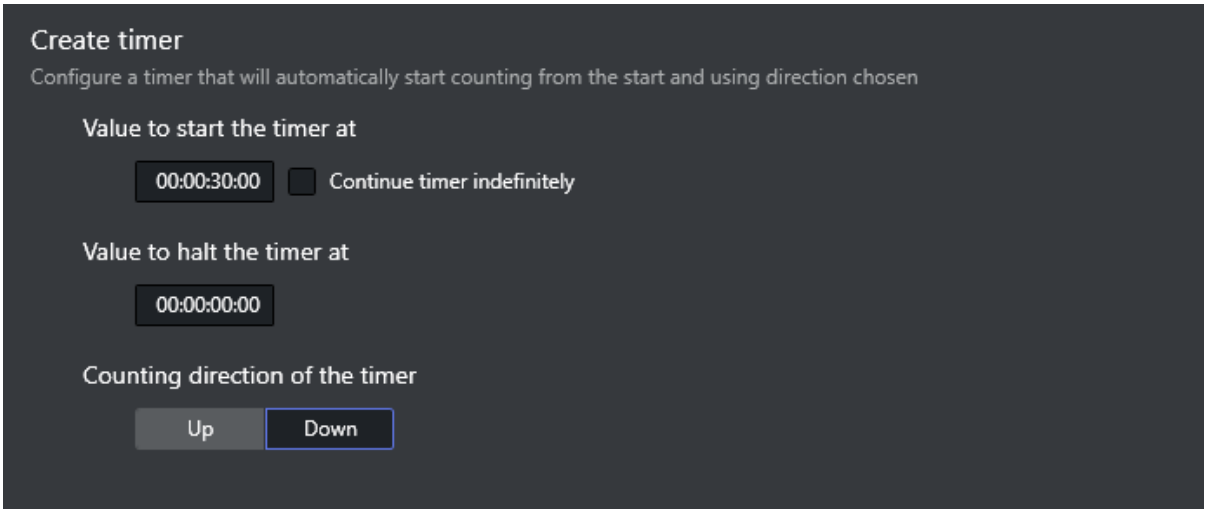
The **Video Editor** allows selecting a video from the Viz One browser. Viz One needs to be enabled in Global Settings first.

This is the only browser available for this editor.

Once a video is selected, *Title* and *House Id* can be verified in the textboxes below.

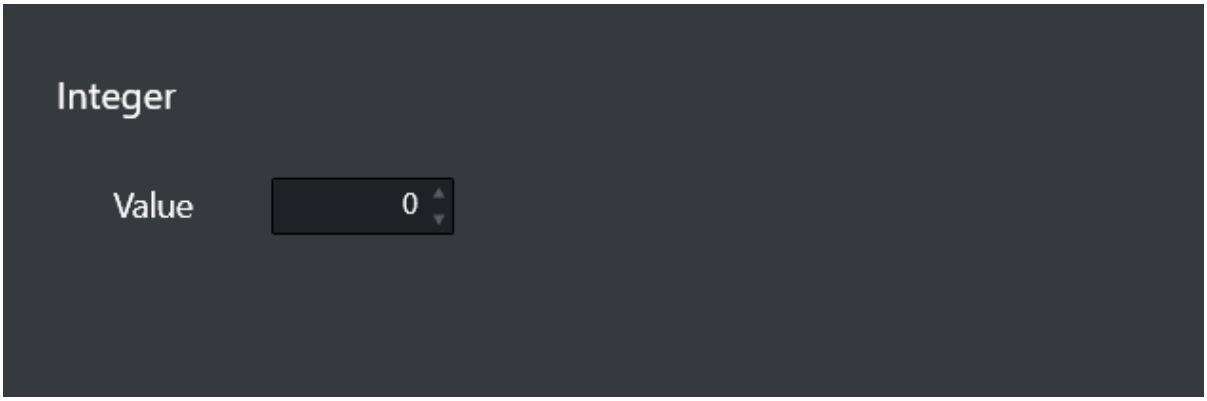
Copyright © 2024 Vizrt

Clock Editor



The Clock Editor is used to create a timer. It is possible to set the start and halt time for the timer, or let the timer count indefinitely. The timer can be set to count up or count down.

Integer Editor



The Integer Editor allows typing an integer in the number input box and the content for the corresponding element of the graphics is updated.

Copyright © 2024 Vizrt

Float Editor

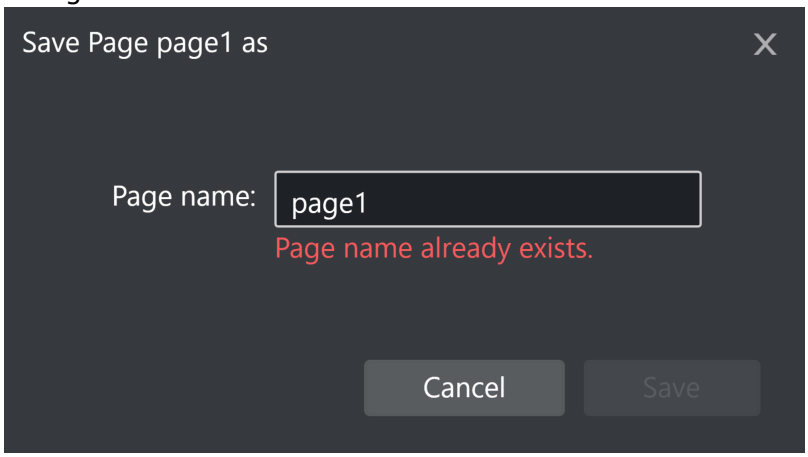
The Float Editor allows typing a float in the number input box and the content for the corresponding element of the graphics is updated.

4.11.4 General Actions

- **Clear all dynamic queries:** Removes all Page Content Filling mapping done for each page field.

  > **Note**: Click **Save** after your actions, to save the page.

- **Cancel:** Closes Page Editor without saving changes.
- **Save as:** Saves all the changes to a new page. A pop up a dialog window is presented to submit a new page name. This option is only available for Pages opened from the Template or Page Pool.



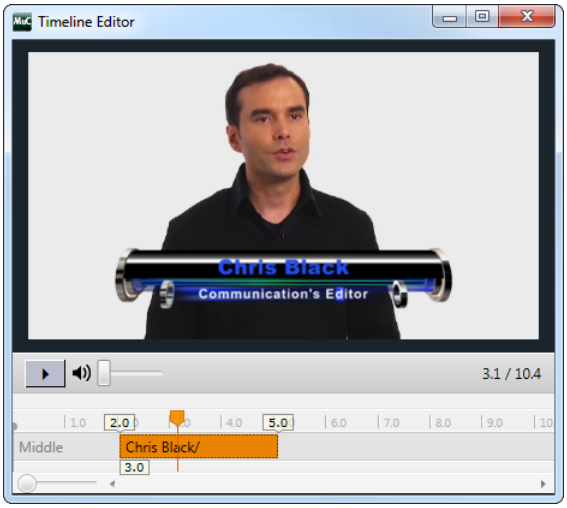
- **Save:** Saves all the changes in the current page. This button is only available for Pages opened from the Page Pool or from a Playlist.

Copyright © 2024 Vizrt

See Also

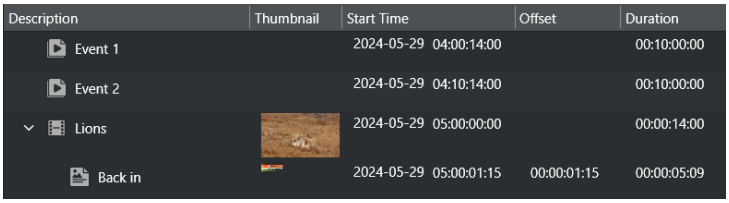
- Timeline Editor
- Page Content Filling

4.12 Timeline Editor



The Timeline Editor allows you to edit secondary events such as video clips and video clips with embedded graphics.

4.12.1 Add Graphics to the Video Timeline



1. Open the playlist.
2. Add a video clip to a primary event.
3. Drag and drop single or multiple pages onto the video clip.
   When adding graphics on top of video clips you should always make sure you have set **Duration** beforehand.
   If not, graphics will not appear in the timeline as the default duration is 00:00:00:00.
4. Double-click the video clip.
5. Edit the duration of the graphic(s).
   The Duration in the playlist changes simultaneously.
6. Close the window.

4.12.2 Remove Graphics from the Video Timeline

1. Open the playlist.
2. Expand the video element in the playlist, exposing the time graphics elements.
3. Select the graphics element and press **DELETE**, or
4. Right-click the element and select **Remove**.

See Also

Copyright © 2024 Vizrt

- Page Editor
- Clips
- Working with Video Clips
- Video Preview Configuration

Copyright © 2024 Vizrt

4.13 Combo Page Editor

A combo (combination) template is a template based on multiple **Transition Logic** (TL) foreground scenes from different **logical layers**. Having a combo template allows you to create combo pages that can perform several Transition Logic actions in one page-take. A TL *action* changes the **state** for a TL page. A single combo page could for instance take out a lower-third and an animation while simultaneously inserting a logo. A combo page forms a template or page that performs several Transition Logic steps or actions, similar to creating a script to perform several actions in a procedure. TL states are typically be IN, OUT, LEFT, RIGHT, UP, DOWN and so forth.

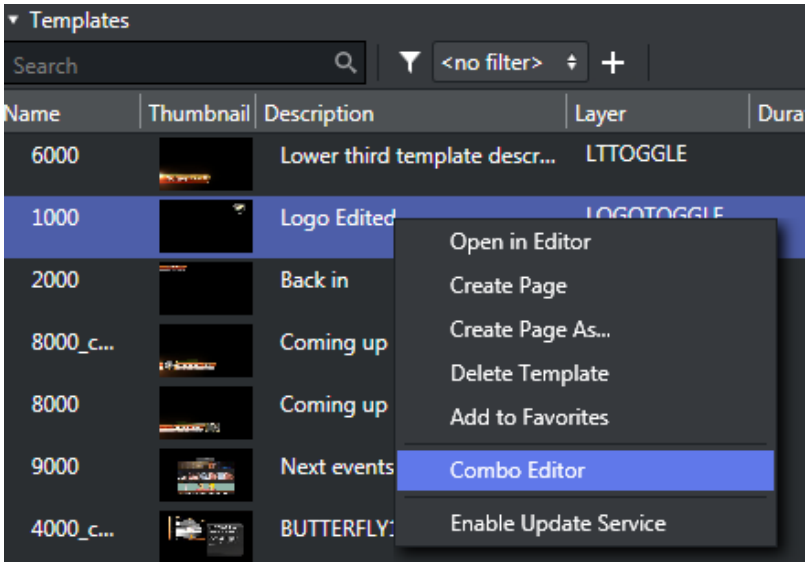
Combo pages are based on and require Transition Logic templates, as described below.

4.13.1 Working with the Combo Page Editor

As described above, a combo template can only be created from transition logic templates. In order to create a combo page, the templates must be in different layers. This must be ensured by the graphical designer when the scenes are created in Viz Artist.

Create a combo template:

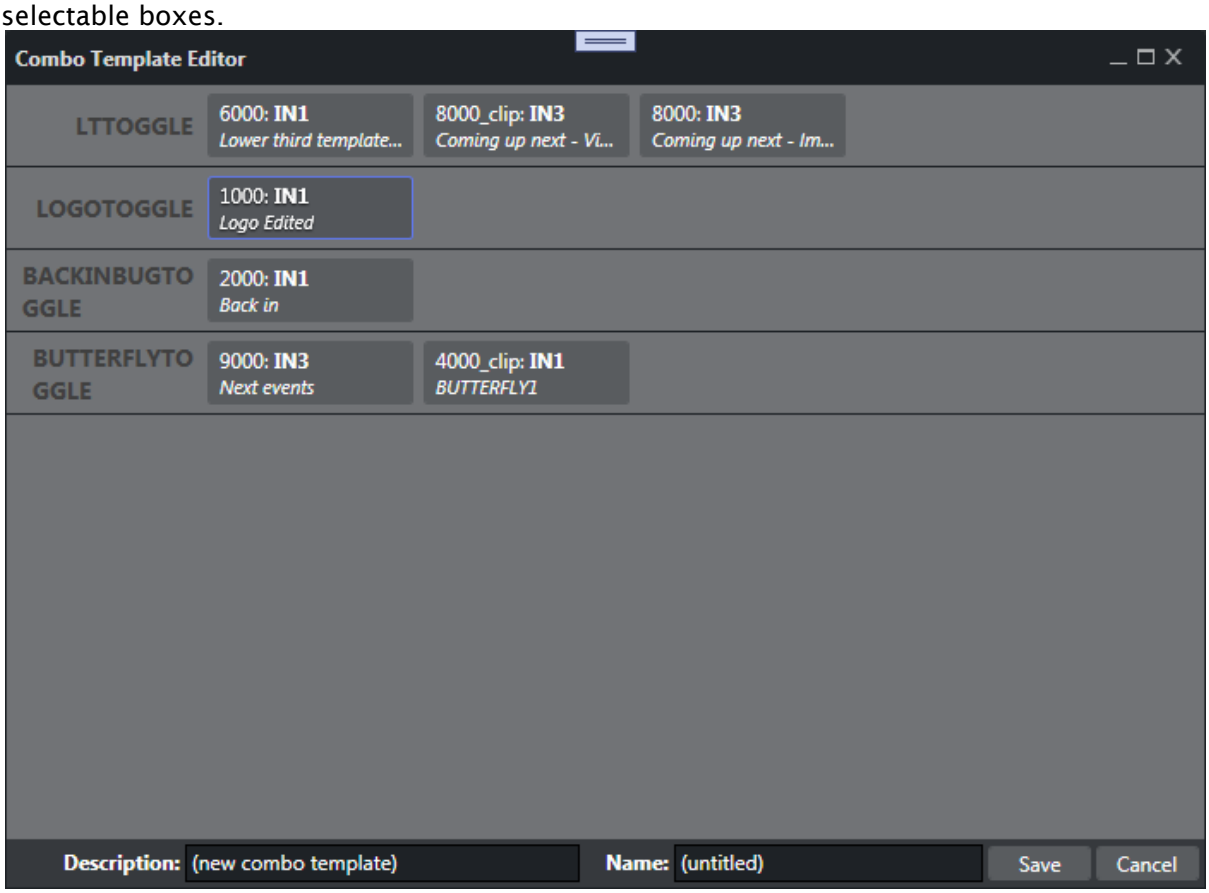
1. **Right-click** a template in the **Templates and Pages** drop-down menu and select **Combo Editor**:



2. The Combo Template Editor is shown. For each TL layer (in the example screenshot below: logo_zehd, music_name, etc) the various states that can be selected will be shown as

Copyright © 2024 Vizrt

selectable boxes.



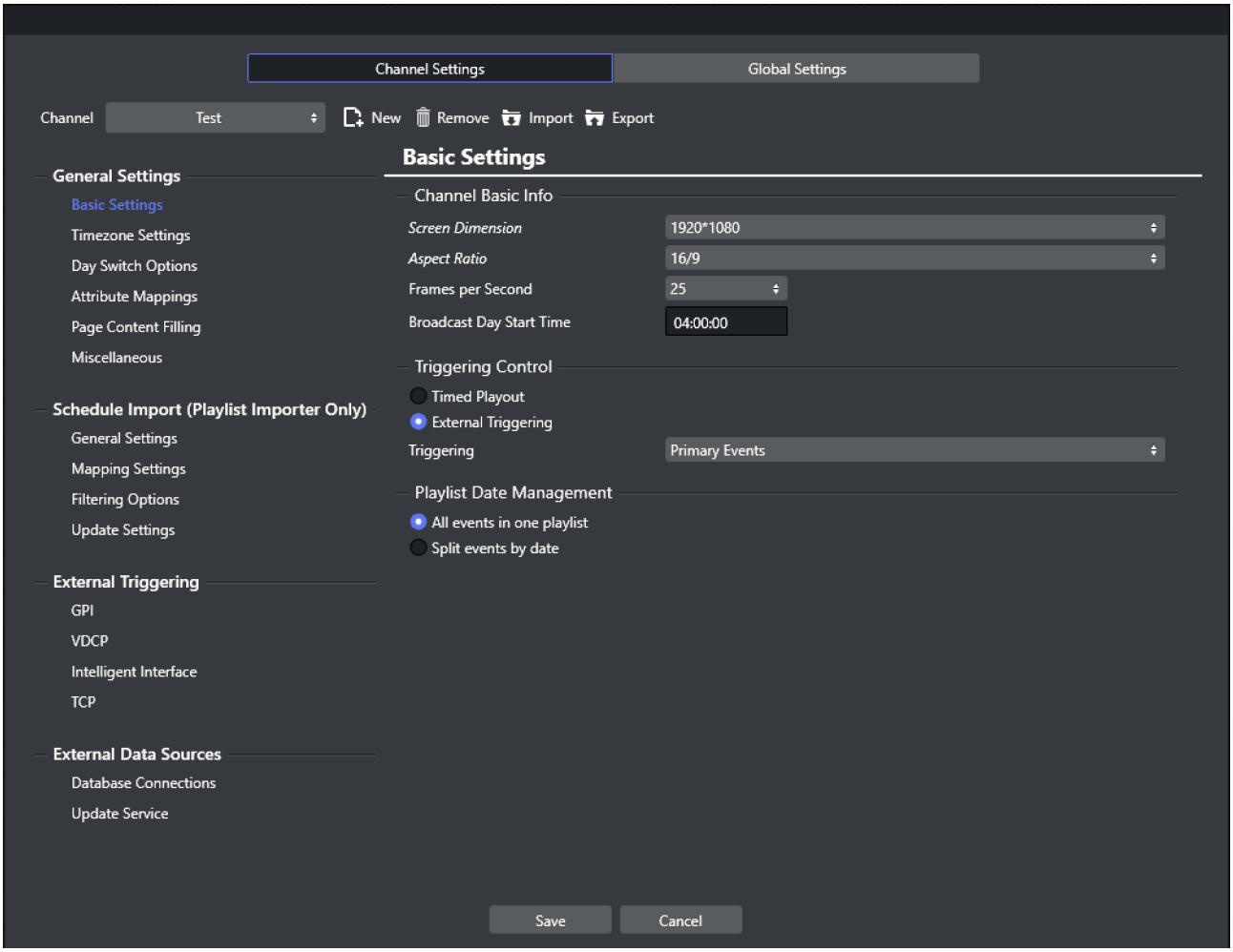
3. In the **Combo Template Editor**, select the TL state that should be activated when pages based on the TL template are shown on air.
   Selected states are highlighted in orange.
4. Give the new combo template a text description and a unique name.
5. Press **Save**.

Copyright © 2024 Vizrt

5 Settings

Settings are split into two distinct categories: **Channel** and **Global**. Channels are the various channels that have been defined by the user using Viz Channel Branding Client.

- **Channel settings** define the output channels connected to your scheduling system. Most of the configuration work is done on a *per channel* basis. When configuring channel settings, ensure that you have selected the required channel beforehand. Most settings are set using the **Settings** window.
- **Global settings** apply to all channels.



This section covers the following topics:

- Global Settings
- Channel Settings
- Client Configuration File

Copyright © 2024 Vizrt

5.1 Global Settings

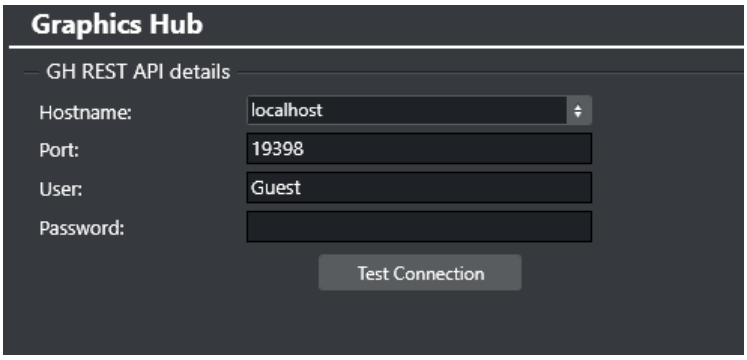
In addition to Channel Settings, there are **Global Settings** that affect all channels stored in the same Media Sequencer (MSE).

Each of the settings groups are presented below:

- General Settings
- Import / Export Settings
- External Devices
- Logging

5.1.1 General Settings

Graphics Hub



Template Import panel, as well as some editors for Page Fields, have a Graphics Hub Browser where the operator can navigate around the contents of a Graphics Hub.

To define the remote Graphic Hub

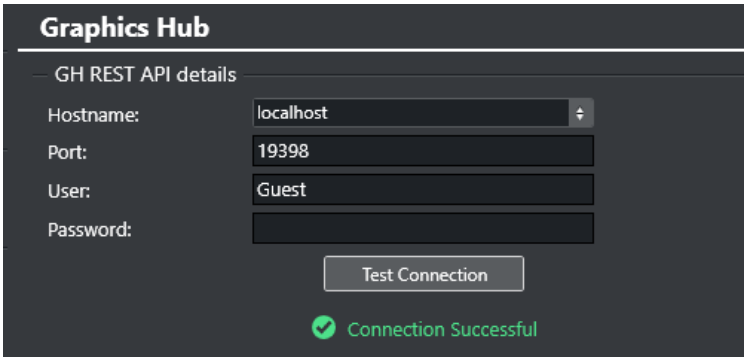
- **Hostname:** <Graphic Hub server name> Can be localhost or a different PC where it is installed. Most often a server machine.
- **Port:** Graphics Hub (GH) port name.
- **User** + **Password:** GH database user and password. Recommended to use 'Guest' if so configured on the GH.

To test the Graphic Hub connection

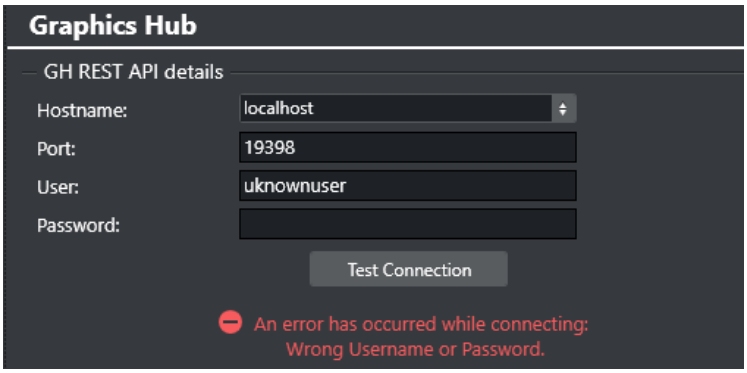
- Use the **Test Connection button to verify** the connection to a Graphic Hub database, with the given credentials.

Below are examples of successful and failed connection:

Successful connection

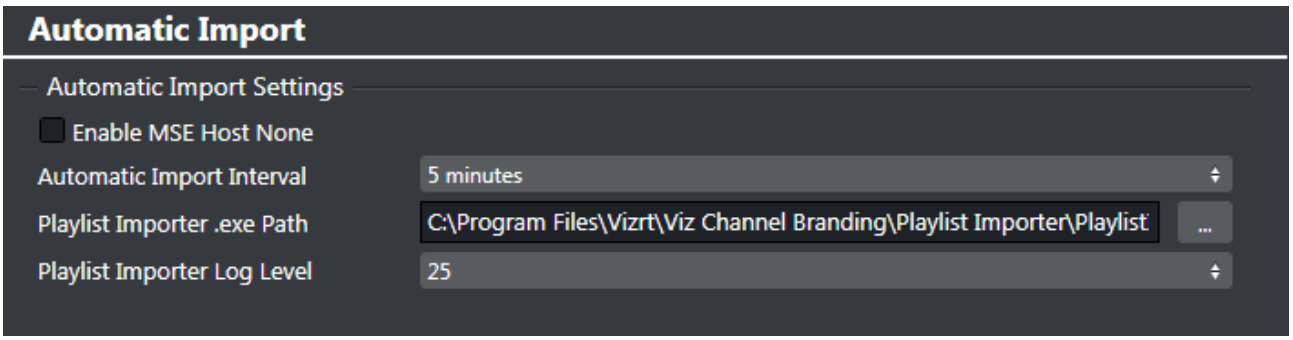


Failed connection



5.1.2 Import / Export Settings

Automatic Import



Automatic Import settings are global for all channels.

You must enable automatic import for *each* affected channel to automatically import a schedule for a specific channel (see General Settings above).

- **Enable MSE Host None**: *Selected*, enable automatic imports for all channel in *channel.ini* file, as oppose to limiting the request to set channels in a single Media Sequencer.
- **Automatic Import Interval:** Select the interval at which automatic imports of schedules will occur.

Copyright © 2024 Vizrt

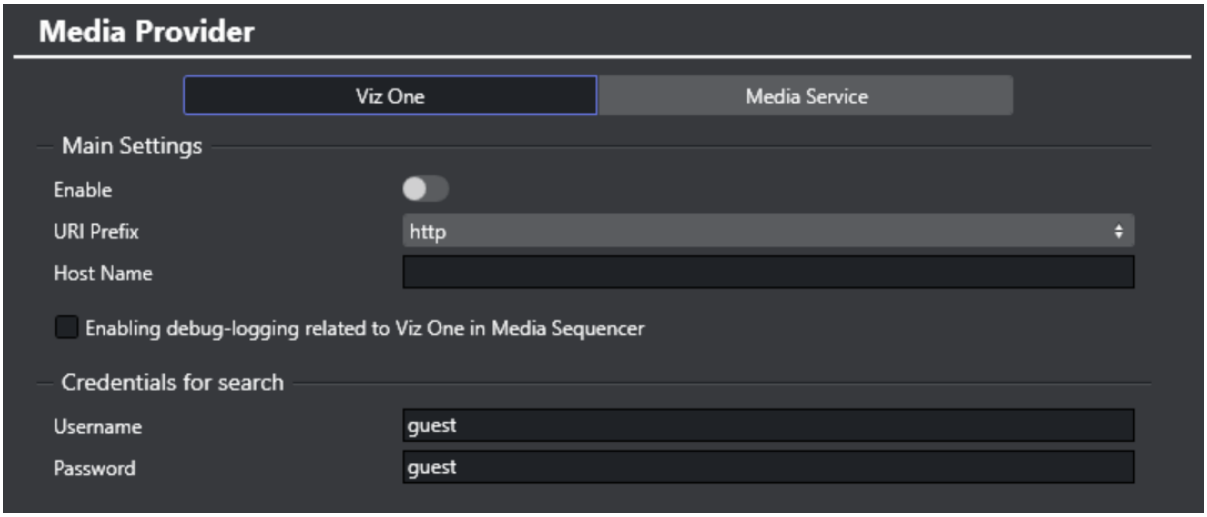
- **Playlist Importer .exe Path:** The file path on the remote machine where the Playlist Importer is installed.
  Select *PlaylistImporter.exe* or similar batch file.
  Playlist Importer is installed on the Media Sequencer machine by default.
- **Playlist Importer Log level:** Sets the log level of the Playlist Importer. See the Playlist Importer's **Startup Parameters**.

5.1.3 External Devices

Media Provider

You can select from either **Viz One** or **Media Service** as a media asset provider:

Viz One



These global Viz One settings are used by *all* channels sharing the same Media Sequencer, and define the Viz One connection to the Media Sequencer.
Once connected to Viz One, Media Sequencer can request that media stored in Viz One be transferred to the Viz Engine for playout.

- **URI Prefix:** Select protocol prefix to either **http** or **https**.
- **Host name:** The host name of Viz One. The Viz One server instance based on either an IP address or host name. It is recommended to use a fully qualified host name.

  > ⚠ **Note:** Whichever addressing method you choose, (for example IP or a fully qualified host name) always use the *same method* throughout the entire setup process.

- **Username/Password:** The depth of Viz One searches is based on the permissions associated with the Viz One user. For example, a user with elevated permissions can see a wider set of assets, and can save modifications; a user with restricted Viz One permissions might be prevented from viewing certain types of video assets or files.

Copyright © 2024 Vizrt

To configure a Viz One connection

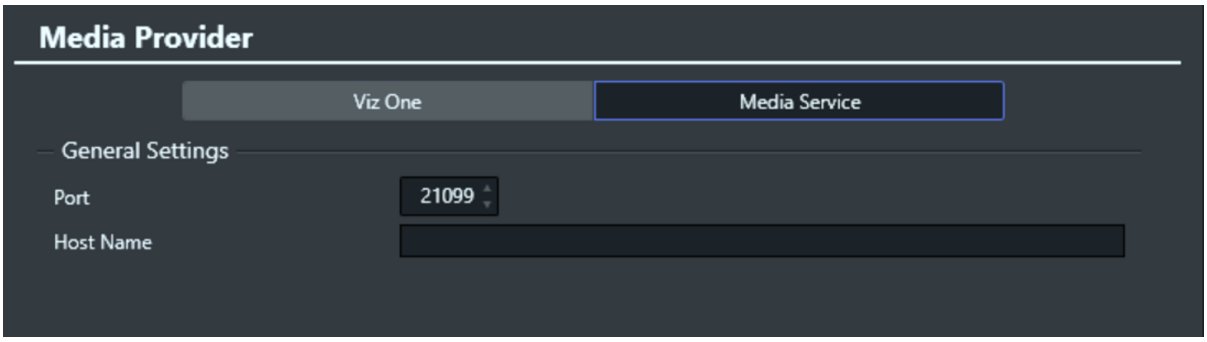
1. Make sure that **Viz One** is enabled.
2. Enter the Viz One **host name**.
3. Enter **Username** and **Password**.
4. Click **Save**.
   You can now select the **Video Clips** options and search for a clip in Viz One (to verify the connection).

See Also

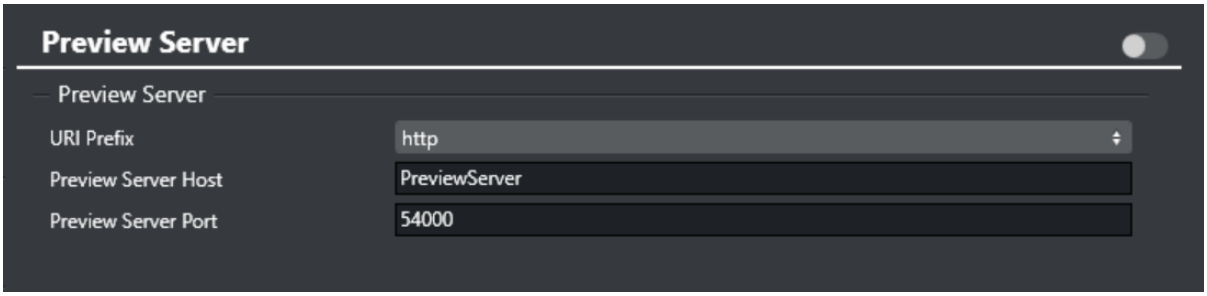
- Viz One Integration

Media Service

Media Provider is a simple REST-based service enabling operators to search and use video resources from a Viz Engine clip folder. The Media Provider standard TCP/IP port is **21099**. The port-number is configurable and must match the IP port used by the Media Service server.



Preview Server



With Preview Server, Viz Channel Branding Client can request frames of playlist elements to create a visual references in the playlist or to add graphics to secondary event video clips.

> ⚠ **Note:** This configuration requires a running Preview Server. Preview Server must be installed on a *separate* machine.

Preview Server is configured to connect to a pool of Viz Engines, and uses port number 54000 as its default listener port. The port must be changed if the Preview Server uses a non-standard port number. The settings window contains an **On/Off switch**. You can configure settings with the

Copyright © 2024 Vizrt

device in on or off mode. When **off**, the settings will take effect the next time the device is turned on.

- **URI Prefix:** Select protocol prefix to either **http** or **https**.
- **Host name:** The host name of Preview Server. Either an IP address or host name. It is recommended to use fully qualified host name.

  > ⚠ **Note:** Whichever addressing method you choose, (for example IP or fully qualified host name) always use the *same method* throughout the entire setup process.

- **Preview Server Port**: Port number for your Preview Server.

> ⚠ **Note:** It is recommended to run the Preview Server on a dedicated Viz Engine machine.

To configure a Preview Server

1. Select protocol type (*http/https*).
2. Enter your Preview Server **host name**.
   and **port number**.
3. Click **Save**.

5.1.4 Logging

The **As Run Log** records all *Take* commands sent to the Media Sequencer.

When enabled, the log can be stored as a standard text output (one line per take) or in a more structured JSON format for further processing.

> ⚠ **Note:** A log entry is *not* a guarantee that media went on air.

For more information (such as how to customize log entries), please see the *Media Sequencer Manual*, section *As Run Log*.

> ⚠ **Note**: The *Media Sequencer Manual* is available at http://localhost:8580/mse_manual.html when the MSE is running. You can also access the manual by opening the local file www/mse_manual/index.html in the MSE install directory.

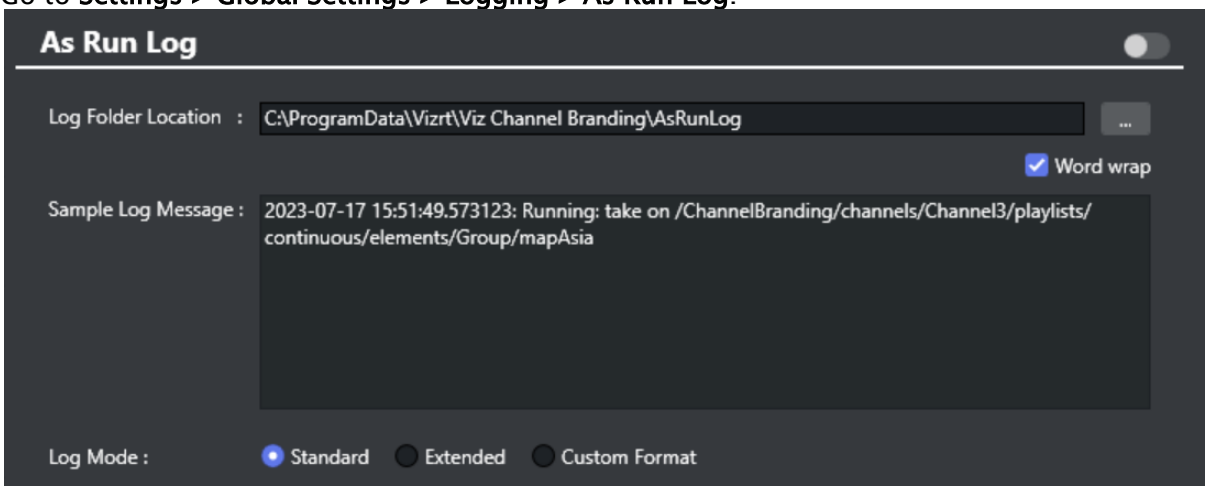
Enabling As Run Logging

To enable an As Run log

In most cases, the Media Sequencer runs on a separate machine (usually the same machine as your Viz Engine) when Viz Channel Branding Client is being run in MCR mode.

Copyright © 2024 Vizrt

1. Go to **Settings > Global Settings > Logging > As Run Log**.



2. Select **ON** to enable logging.
3. Enter the **Log Folder Location** if required, or use the default location.
4. Select **Standard** or **Extended** or **Custom Format**.
   - (Optional) **Log Mode** is *Extended* or *Custom Format*
     Perform customization. See Global Settings below.
5. Click **Save**.

Use *Extended* or *Custom Format* when there are **As Run** logging requirements not covered in Standard mode.

Log Mode and Format

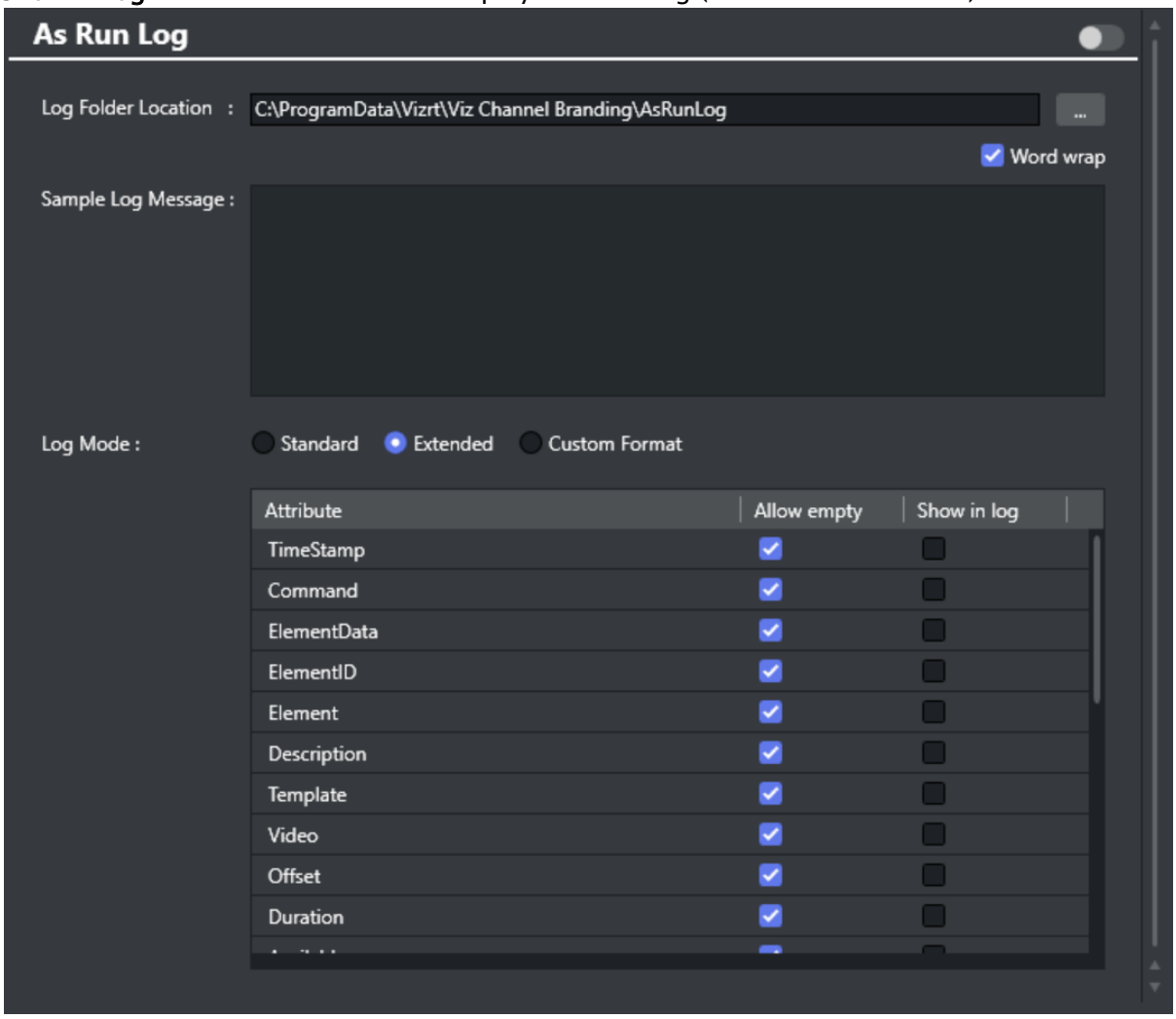
Extended mode

Use *Extended* mode to create an As Run log format

- **Allow empty**: Display content when its value is empty (default is selected).

Copyright © 2024 Vizrt

- **Show in log**: Attributes that will be displayed in the log (default is unselected).

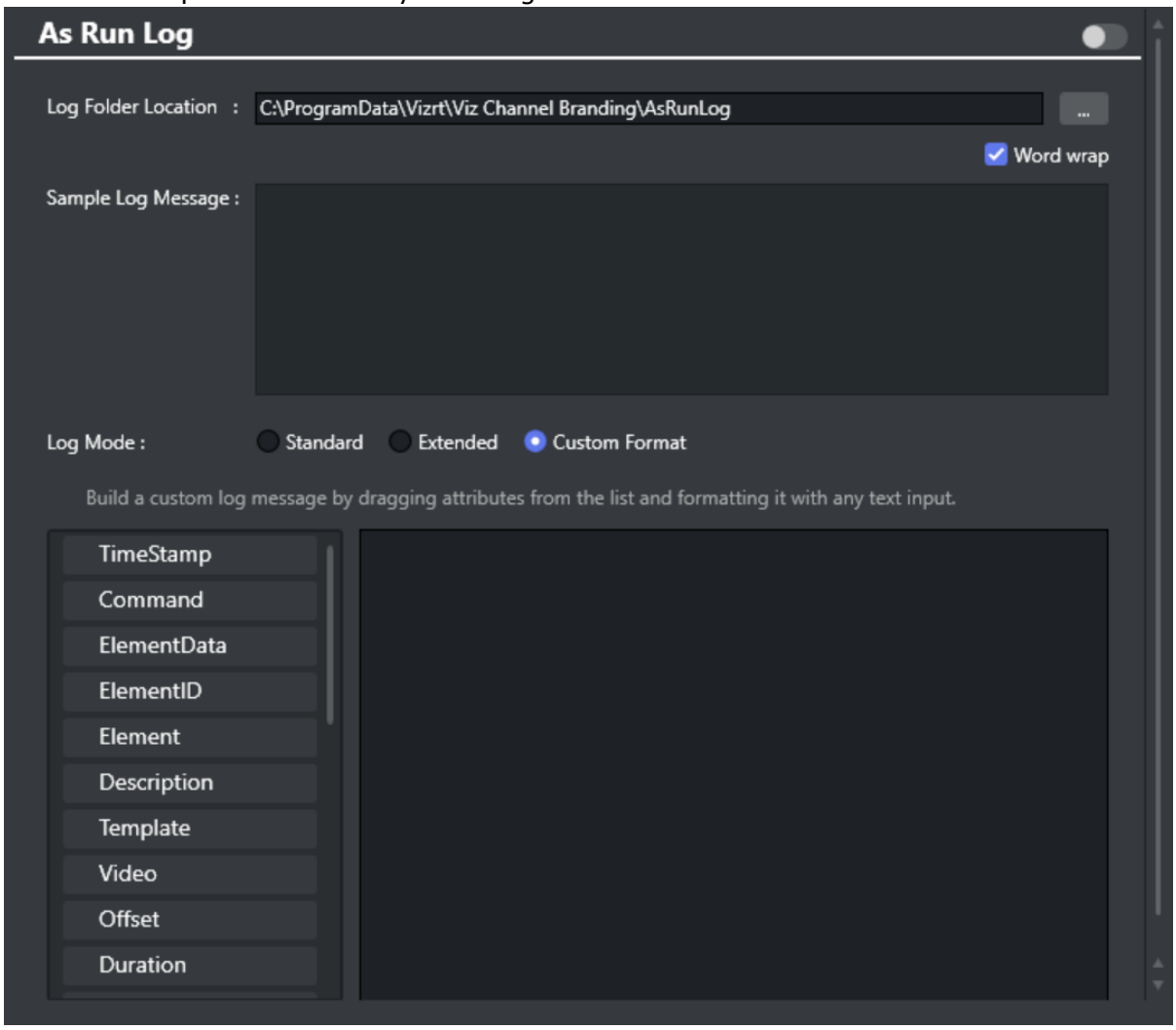


Custom Format

Use **Custom Format** for creating a specific As Run log format.

To create a log format

- When **Custom Format** is chosen, you can drag the available formatting keys (attributes) from the left to the panel immediately to the right.



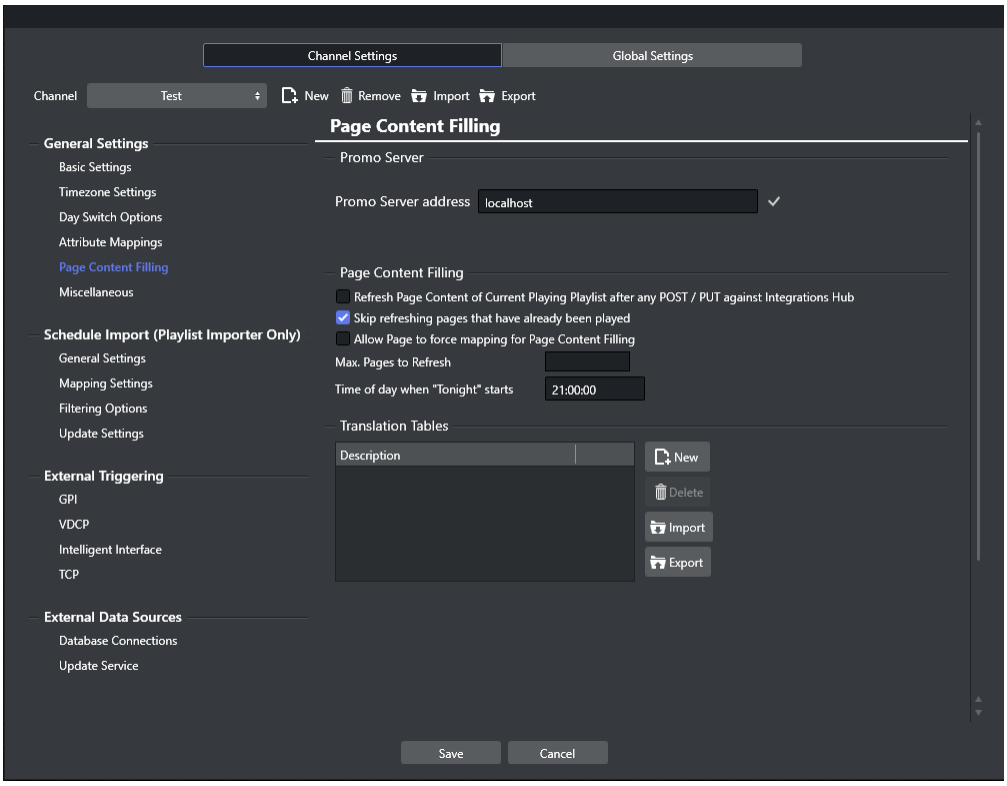
See Also

- Client Configuration File
- Channel Settings
- Video Configuration

Copyright © 2024 Vizrt

5.2 Channel Settings

In addition to Global Settings, each individual channel has specific settings affecting import/update and playout.

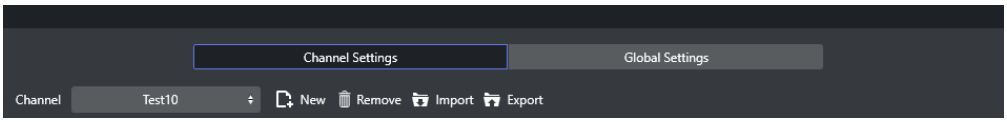


This section covers individual channel configuration. Follow these topics to configure the system:

- Managing Channel Settings
- General Settings
- Schedule Import (Playlist Importer Only)
- External Triggering
- External Data Sources

Copyright © 2024 Vizrt

5.2.1 Managing Channel Settings



The following sections cover how to **add**, **remove**, **export** or **import** channel settings.

- Creating a Channel
- Removing a Channel
- Exporting a Channel
- Importing a Channel

Creating a Channel

Create a new channel by pressing the **New** icon and give the channel a name in the dialog.

To create a new channel

1. Open the **Settings** > **Channel Settings** window.
2. Select **New**.
   You are prompted to enter the new channel's name.
3. Enter the channel name *without spaces* (a-z, A-Z, 0-9), and click **OK**.

Removing a Channel

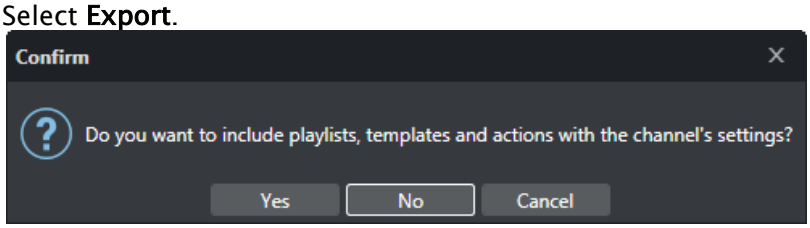
To remove a new channel

1. Select channel in Channel drop-down you wish to remove.
2. Select **Remove**.
   You are asked if you are sure you want to remove the selected channel.
3. Click **Yes**.
   The last removed channel is automatically saved to a *%ProgramData%Vizrt\Viz Channel Branding\backup channel\[ChannelName].bkp* file, which can later be re-imported back.

Exporting a Channel

To export a channel

1. Select channel in Channel drop-down you wish to export.
2. Select **Export**.



Copyright © 2024 Vizrt

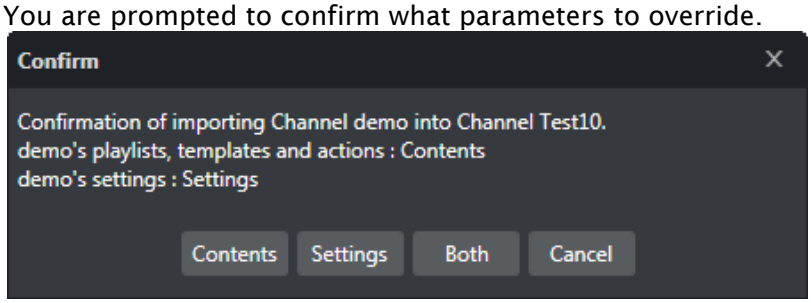
You are asked if you want to include the playlists and templates along with the selected channel's settings.

3. Click **Yes** or **No**, as required.
4. Enter a file name and click **Save**.

Importing a Channel

To import a channel

1. Select **Import**.
2. Select the XML file and click **Open**.
   You are prompted to confirm what parameters to override.



3. Select one of the following:
   - **Contents**
   - **Settings**
   - **Both**

> **Note:** If the imported channel uses the same path to the Viz Graphic Hub as the original, you will be prompted to confirm the overwrite of templates.

Copyright © 2024 Vizrt

5.2.2 General Settings

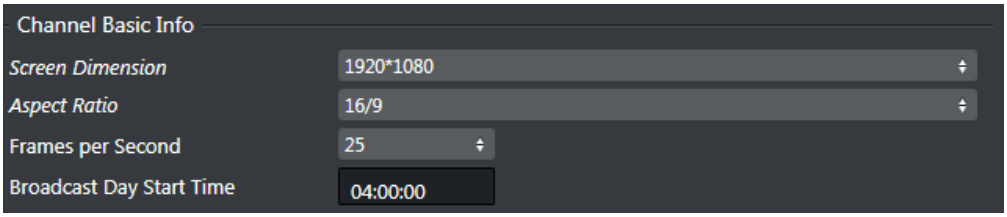
- Basic Settings
- Timezone Settings
- Day Switch Options
- Attribute Mappings
- Page Content Filling
- Miscellaneous

Copyright © 2024 Vizrt

Basic Settings

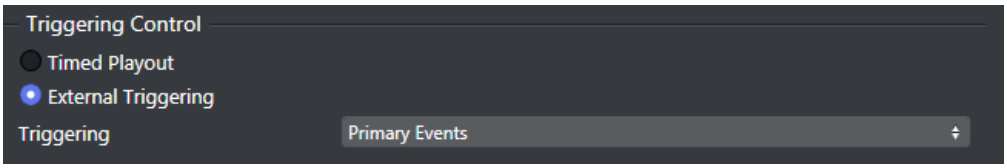
- Channel Basic Info
- Triggering Control
- Playlist Date Management

Channel Basic Info



- **Screen Dimension:** Reference-value for Page Editor *Preview* panel. Options are:
  - SD NTSC: 720 x 486
  - SD PAL/SECAM: 720 x 576
  - HD 720: 1280 x 720
  - HD 1080: 1920 x 1080 (current default value)
- **Aspect Ratio:** Defines the aspect ratio of the Page Editor's *Preview* panel. Options are 4:3 and 16:9.
- **Frames per Second:** Defines the frame rate at which the output will be displayed. The Media Sequencer may be responsible for the entire playlist or only secondary events (when an external system controls the primary events).
  Unless this setting is correct, the Media Sequencer will not calculate the events correctly, creating a *frame inaccurate* system. See Triggering Control.
- **Broadcast Day Start Time:** The broadcast day start time in use for the channel.

Triggering Control



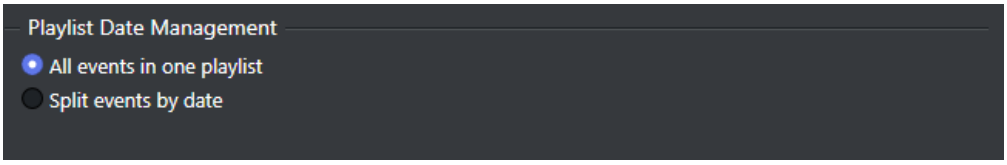
- **Timed Playout:** Allows Media Sequencer to control the triggering of the playlist's timed events.
- **External Triggering:** Allows the automation system to trigger the playlist's timed events.
- **Triggering:** Allows you to define the level at which the automation system will have control over the timed events.
  - *Primary Events*:
    Media Sequencer will trigger the secondary events based on offsets to the timed primary events.

Copyright © 2024 Vizrt

- *Secondary and Video Events*:
  The automation system is in full control of the playlist and Media Sequencer will act as secondary.

  > ⚠ **Note**: Using *Secondary and Video Events* and assets from Viz One system as Primary Events, the video of the primary Primary and all other Secondary types can be played out in combination and controlled from Automation.

Playlist Date Management

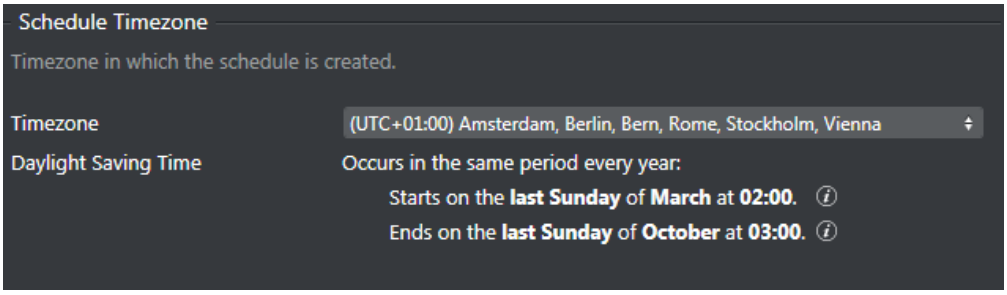


- **All events in one playlist:** Allow only one continuous playlist exist per channel. Regardless of time and on-air date, all events are included in the imported playlist. This setting is recommended.
- **Split events by date:** Schedules received from automation systems *are split by date*. When selected, events are separated by date and begin time into broadcast-date playlists according to the Broadcast Day Start Time settings.

Copyright © 2024 Vizrt

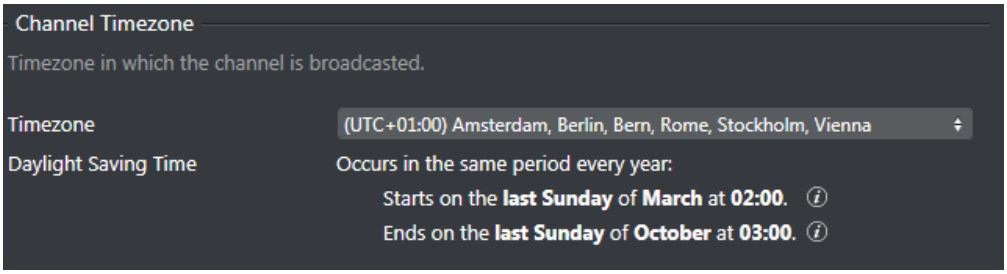
Timezone Settings

Schedule Timezone



- Defines the timezone of the location where the schedule was created and is being imported.
- The system stores time in UTC.
- These settings are necessary for the system to be able to transform the times included in the schedule in local time to UTC.

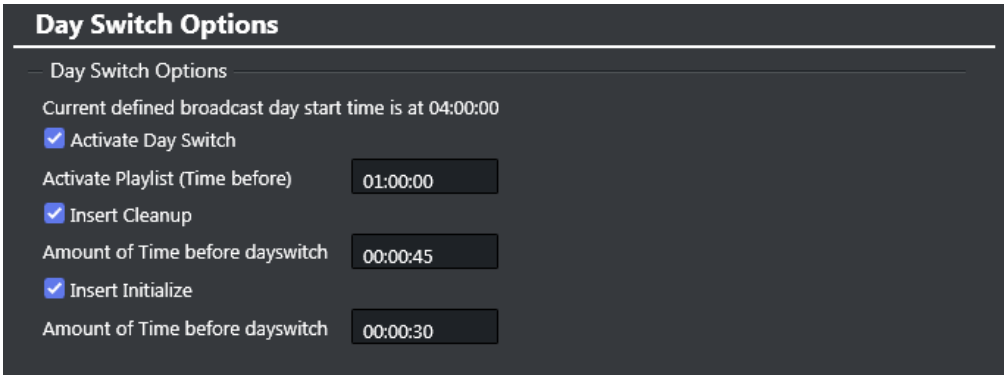
Channel Timezone



- Defines the timezone of the location where the channel is being broadcast.
- The system stores time in UTC.
- These settings are necessary for the system to be able to transform from the stored time in UTC to the local time of where the channel schedule is being broadcast.

Copyright © 2024 Vizrt

Day Switch Options



- **Activate Day Switch:** Activates day switch for split playlists. The default Day Switch is 04:00. This can be changed in Channel Settings as required.
- **Insert Cleanup:** Cleans up the program in Viz Engine a set *amount of time before the day switch* takes place.
- **Insert Initialize:** Initializes the program in Viz Engine a set *amount of time before the day switch* takes place.

> ⚠ **Note:** *Time* is relative to the day switch action that is inserted as the *last* element of the day. For example, if the primary event ends 30 minutes after the preferred day switch time then the switch and all other actions are offset by 30 minutes to adjust for this.

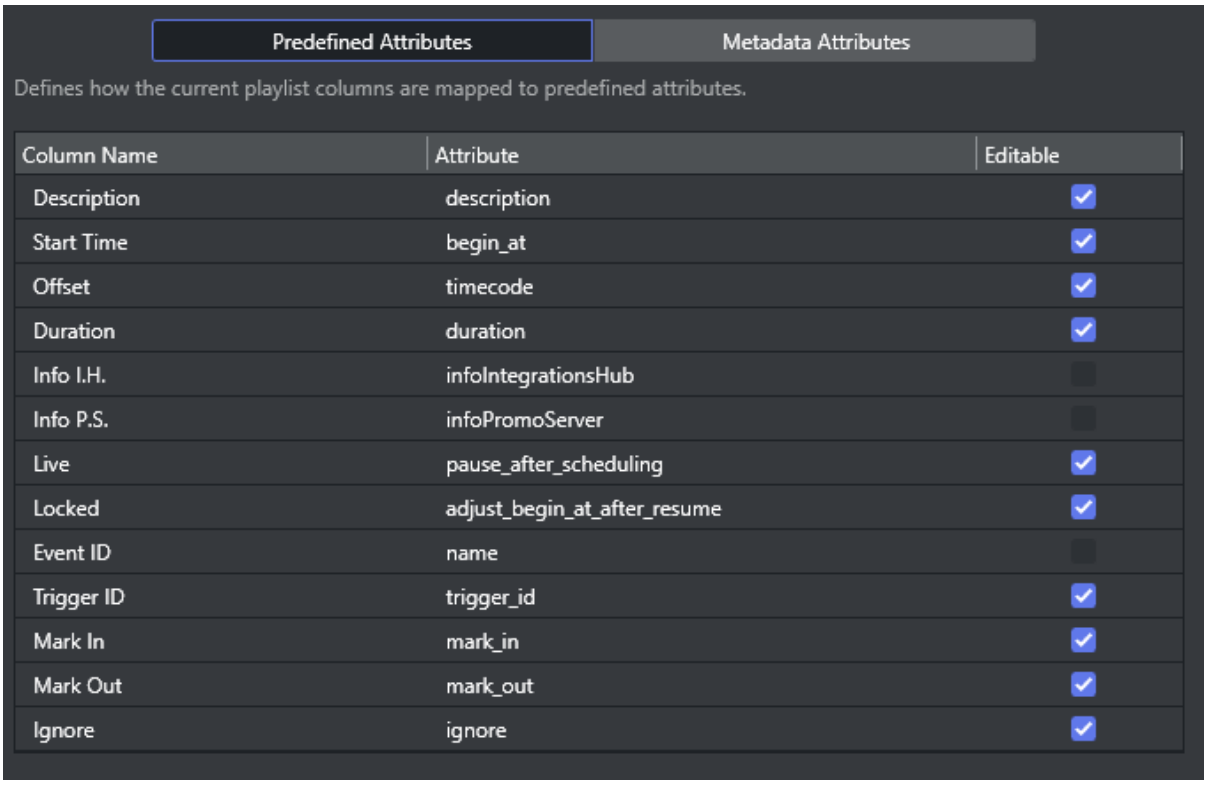
Copyright © 2024 Vizrt

Attribute Mappings

Select from the tabbed choices at the top of the Attributes manu:

- Attribute Mappings
- Attribute Mappings.

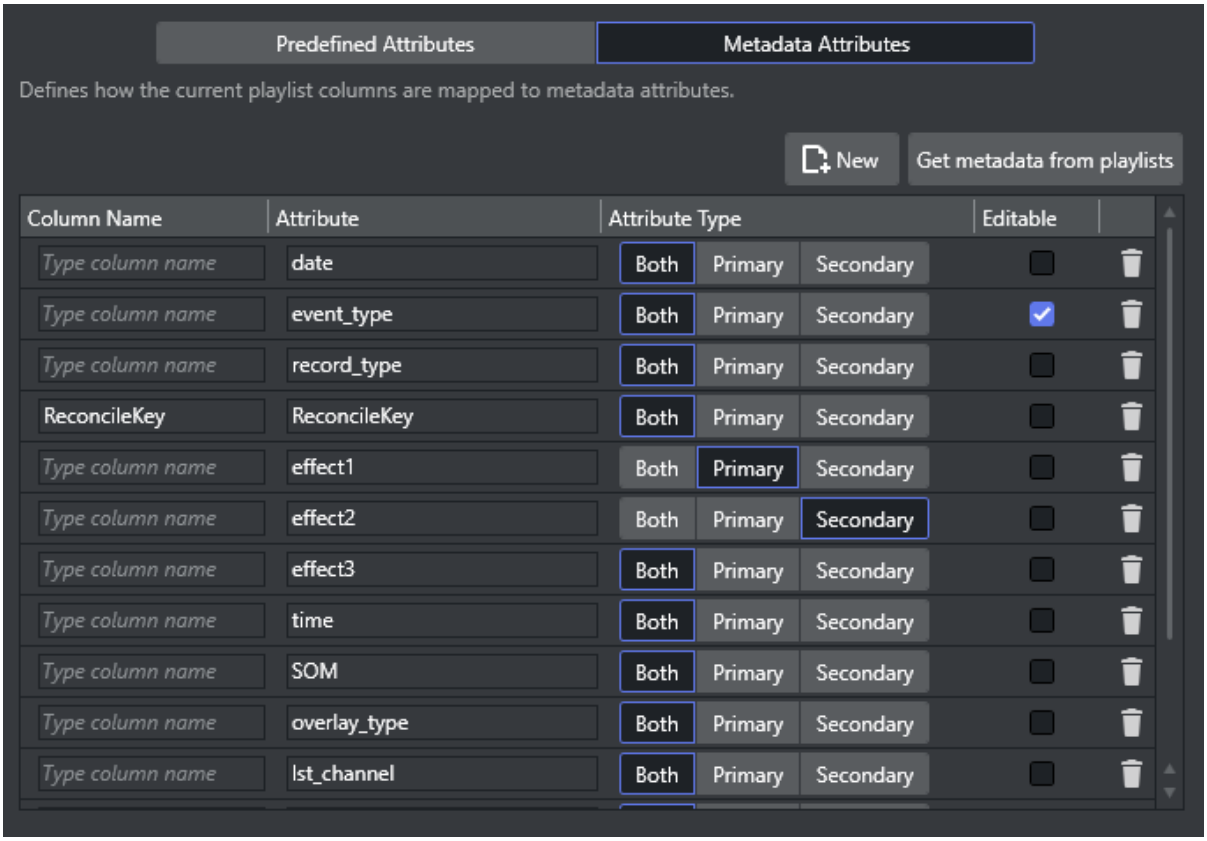
Predefined Attributes



- **Column Name:** Specify what attribute name to show on the playlist.
- **Attribute:** All attributes in this column are used in *Page Content Filling Dialog*, *Rules* condition and *Filtering Options*.
- **Editable:** If value can be edited when shown in the *Playlist* menu.

Copyright © 2024 Vizrt

Metadata Attributes



Selecting the tab **Metadata Attributes** reveals more selections:

- **Attribute Type:** In what kind of events this attribute is found.
- **Get metadata from playlists** triggers the client to examine all playlists in the channel and
  - Find all attributes in primary and secondary events.
  - Automatically adds all attributes in the **Playlist Data Mapping** table with **Column Name** blank.

To Show Playlist Metadata In The Playlist Column

- Right-click on playlist column > *Use External Data* and specify a **Column Name**.

Copyright © 2024 Vizrt

Page Content Filling

This page covers the following topics:

- Initial Registration
- Post-Registration in the Promo Server
- Page Content Filling
- Translation Tables

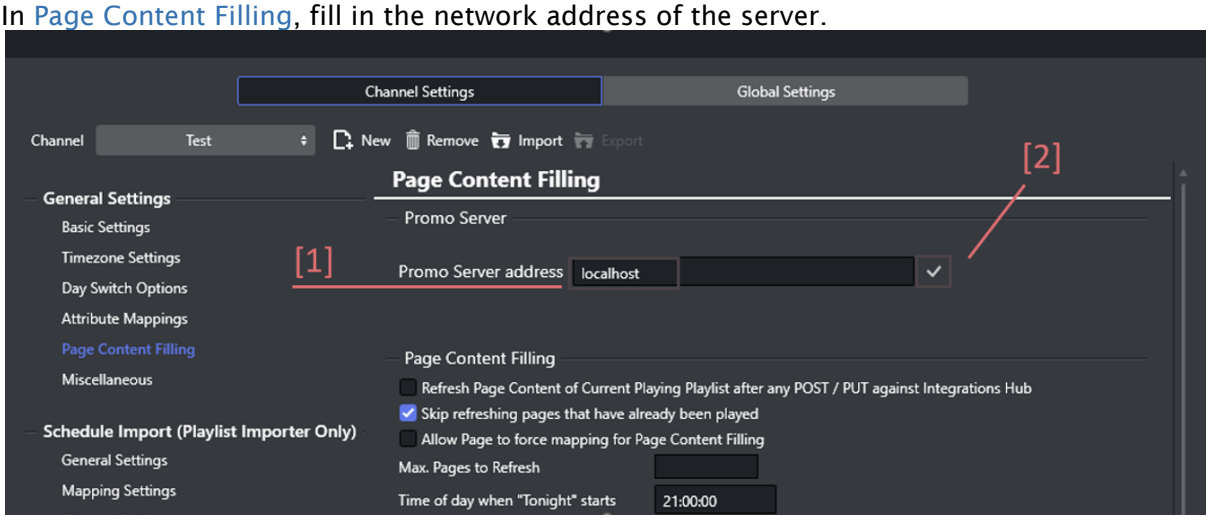
Working with Page Content Filling

Page content characteristics are saved in the Promo Server.

Initial Registration

To register a new address in Promo Server

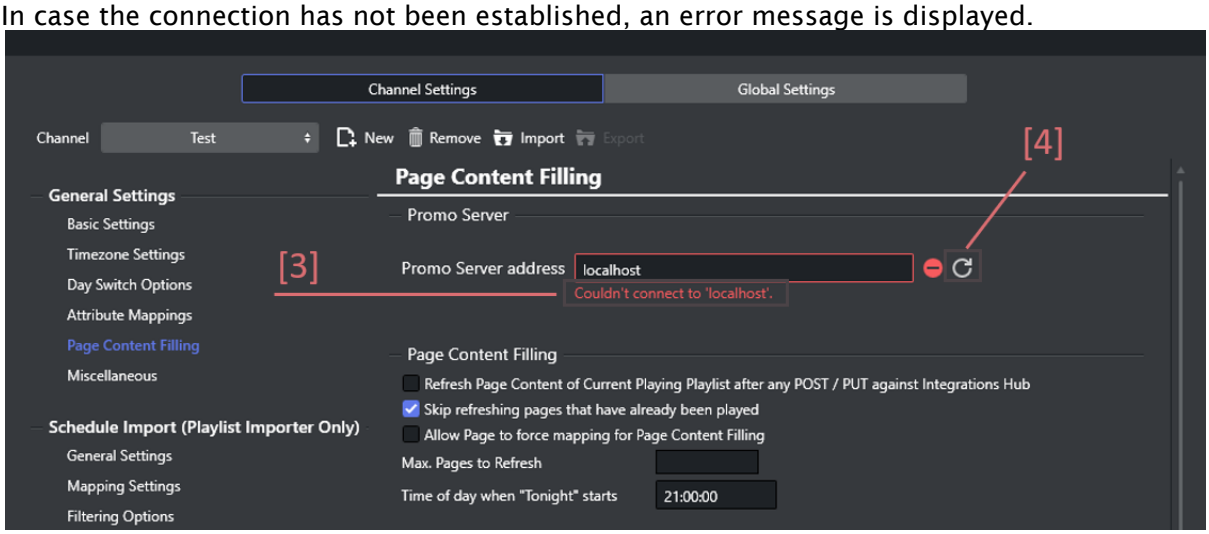
1. In Page Content Filling, fill in the network address of the server.



2. Clickthe **Connect** ( ) button.
   An attempt is made to connect to the Promo Server.
   If successful, the Promo Server address is filled in.

Troubleshooting Registration

In case the connection has not been established, an error message is displayed.



- If the problem can be resolved (for example, you used an invalid address), click the **Refresh** ( ) button.

Post-Registration In The Promo Server

When the Page Content Filling/connection has been established successfully you have three possible scenarios for the selected channel:

1. Current channel and machine address successfully registered in Promo Server
2. Current channel is present in the Promo Server configuration, this machine address is not registered
3. Current channel is not present in the Promo Server configuration, this machine address is not registered.

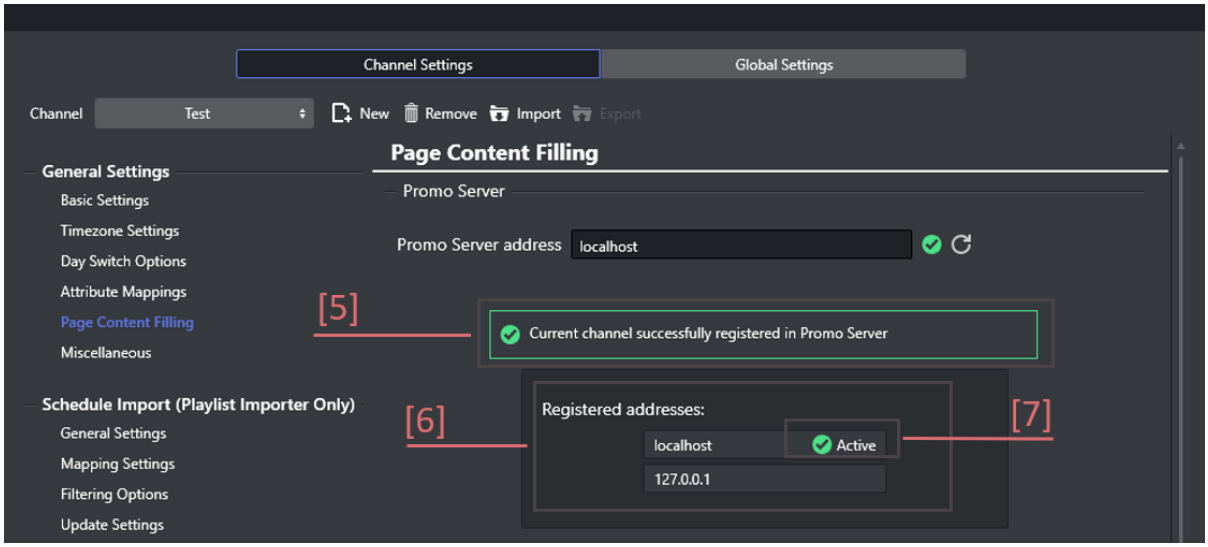
(1) Current channel and machine address successfully registered in Promo Server

The current channel is present in the Promo Server configuration as well as the current machine address, so no configuration is necessary.

- A notification [5] acknowledges successful registration.
- The list of registered addresses [6] for the channel is displayed.

Copyright © 2024 Vizrt

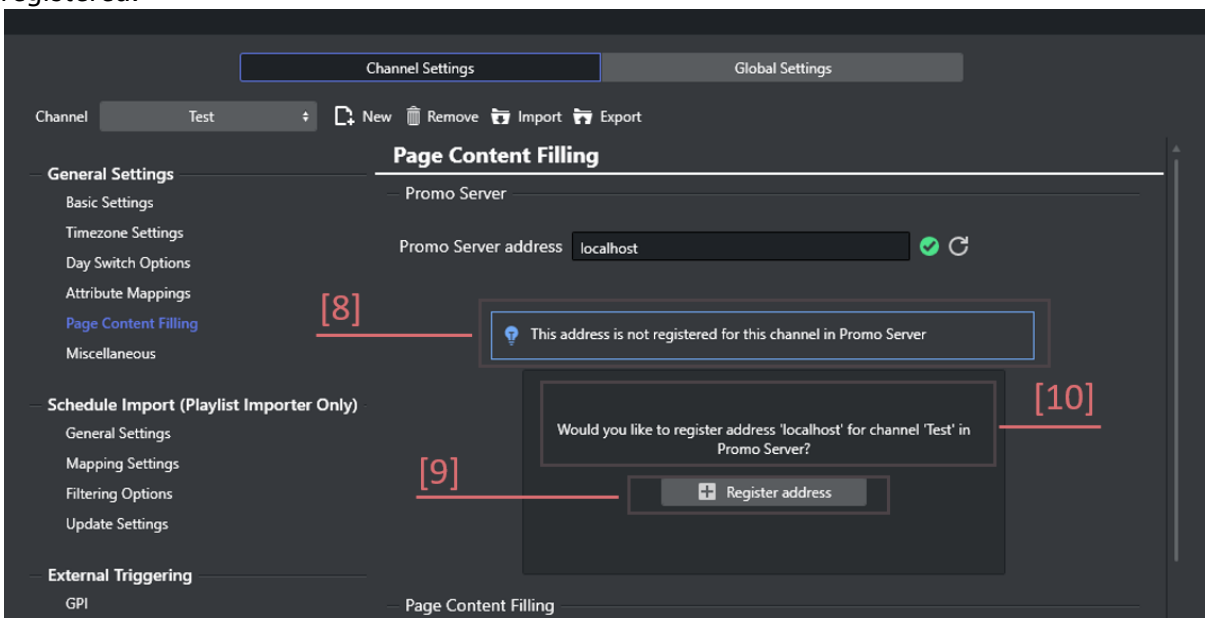
- The respective active channel [7] label indicated.



(2) Current channel is present in the Promo Server configuration, but this machine address is not registered

The current channel is present in the Promo Server configuration, but this machine address is not registered.

- A notification [8] informs that the address is not registered for this selected channel.
- The button **Register address** [9] is displayed.
- A registration prompt notification [10] indicates which address and channel will be registered.



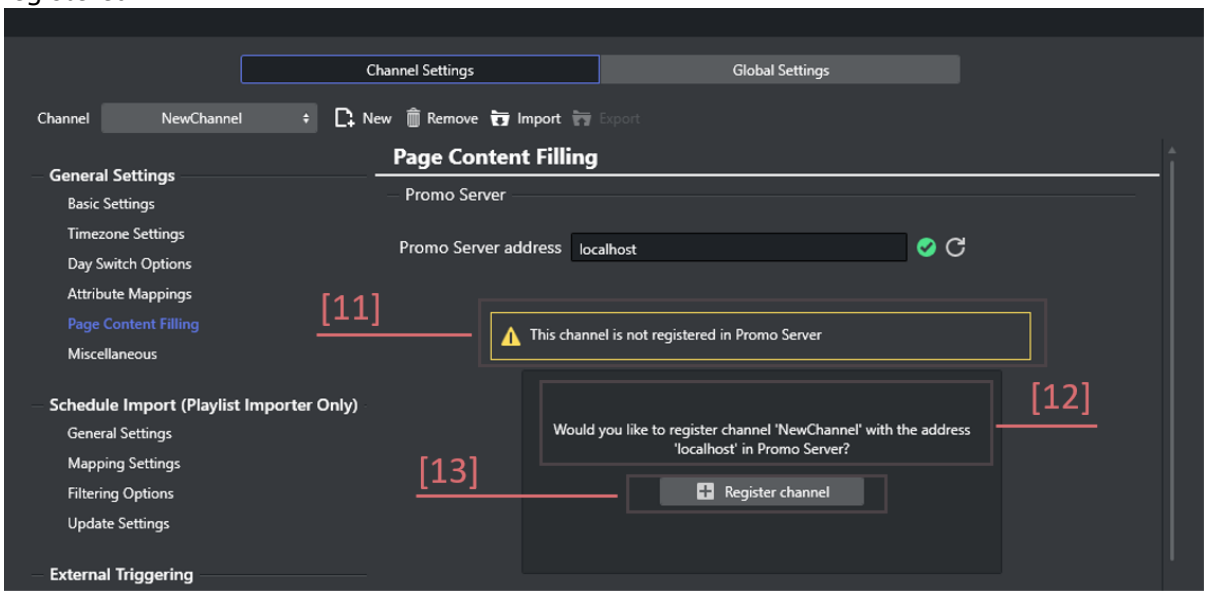
1. Click **Register address** [9] to register in Promo Server.
2. An acknowledgement notification confirms
    - Address is added
    - Address is verified.

Copyright © 2024 Vizrt

(3) Current channel is not present in the Promo Server configuration and this machine address is not registered

In his situation, neither the selected channel is present in the Promo Server configuration, nor is the machine address is registered.

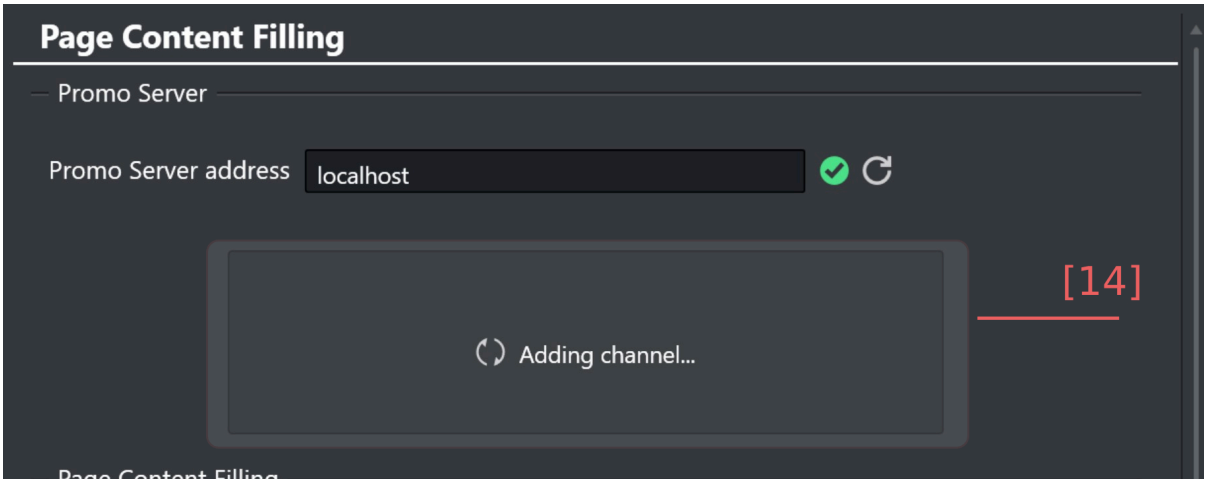
- A notification [11] informs that that the channel is not registered.
- The button **Register channel** [13] is displayed.
- A registration prompt notification [12] indicates which address and channel will be registered.



1. Click **Register channel** [13] to register in Promo Server.
2. An acknowledgement notification confirms
    - The channel is added
    - Address is added
    - Address is verified.

Troubleshooting Promo Server configuration

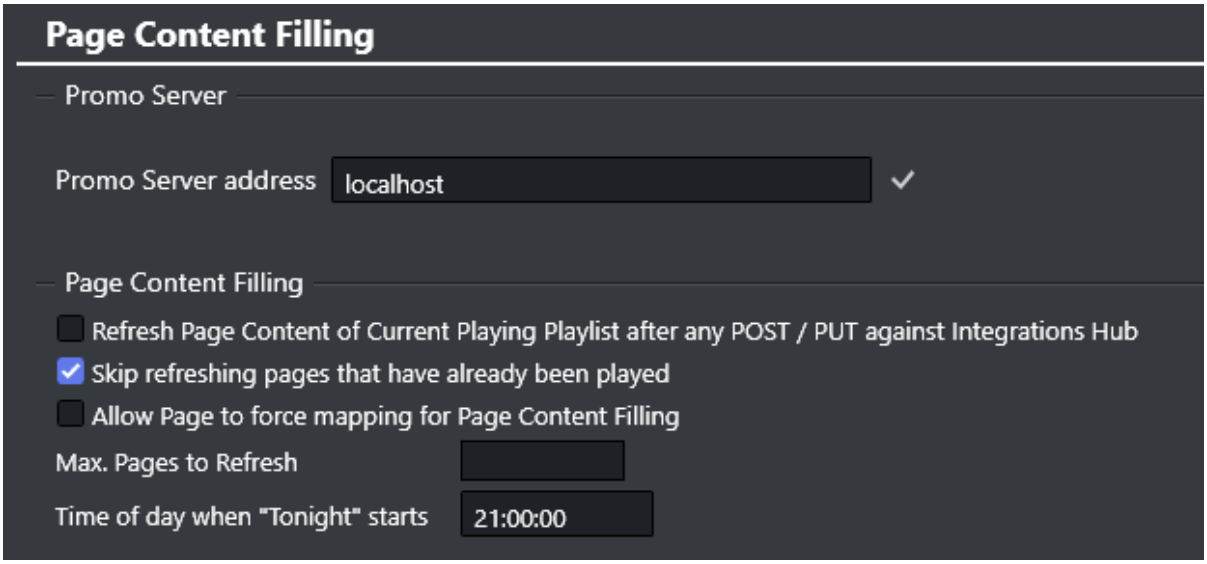
If a certain registration step is taking time, it will be indicated with the status notification [14].



- Restart the step, by clicking **Register address** [9] or **Register channel** [13] respectively.

Page Content Filling

These menu selections control the appearance of what is displayed.

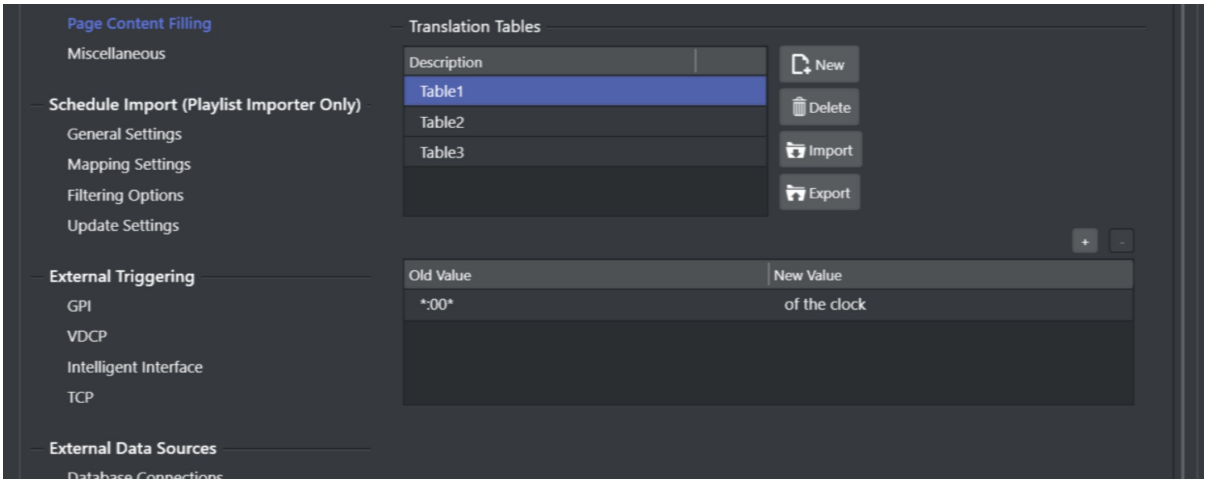


- **Refresh Page Content of Current Playing Playlist after any POST / PUT against Integrations Hub**: The current playing playlist will be refreshed when making a POST / PUT request to another playlist.
  Specifically, when adding or editing a primary element or a secondary element.
- **Skip refreshing pages that have already been played**: The system will skip page content filling for pages that have already been played (default).
  If this option is *unchecked*, the system will perform page content filling for all pages for which lookup rules or database connections have been set.
- **Allow Page to force mapping for Page Content filling**: Allow the name of a page field to act as an instruction-set against Promo Server without needing to set the mapping manually.
- **Max. Pages to Refresh**: (Optional) Limits a full playlist refresh to the defined number of pages.
- **Time of day when "Tonight" starts**: The time of day setting for when "Today" changes to "Tonight" the fault time is 21:00:00.

Translation Tables

Translation Tables are global for translating values when performing Content Filling. Although you can have several tables, only one can be active at any given time.

Copyright © 2024 Vizrt



To manage translation tables

- Use **New** to create a table and add *Old Value* and *New Value* for translations as required.
- Use **Import** and **Export** buttons to load or save translation tables from/to XML-files.
- Use **Delete** to remove a translation table.

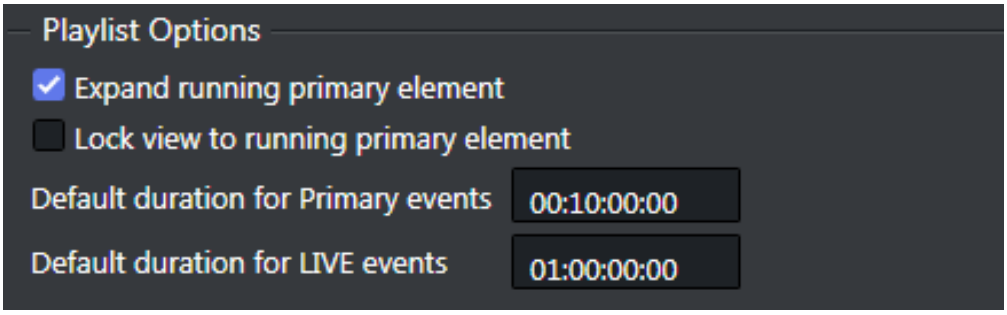
See: Translation Tables

Copyright © 2024 Vizrt

Miscellaneous

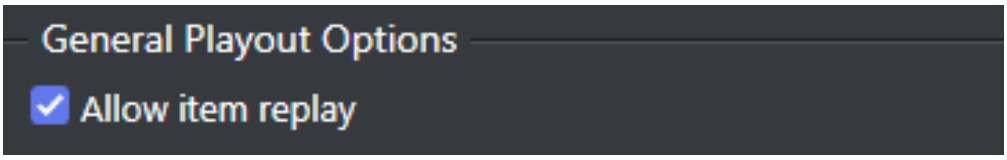
- Playlist Options
- General Playout Options
- Viz One Integration
- Default Scenes
- Define Quick Take Toolbar

Playlist Options



- **Expand running primary element:** If set to true the playing primary element will be expanded when it is started (triggered).
- **Lock view to running primary element:** If enabled, this option will scroll the view of the playlist to the current running group (primary element) to the top whenever the primary element is changed.

General Playout Options

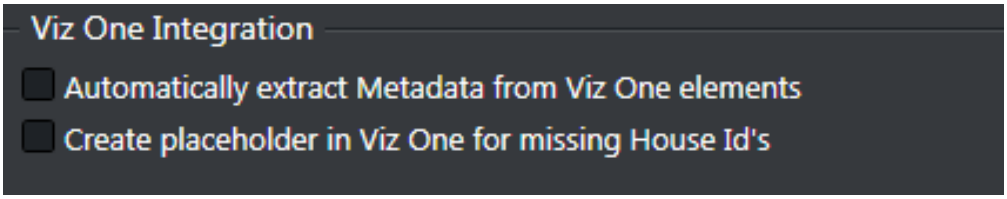


- **Allow item replay:** When checked, the automation system requests the replay of an item in a playlist (same Media ID). Viz Channel Branding will go back and play the requested item (even if marked as played). When unchecked, this setting prevents an event from being played twice, and instead searches for the next event with the requested ID that is marked as not played.

  > ⚠ **Note:** If you are using duplicate Media IDs, uncheck this option.

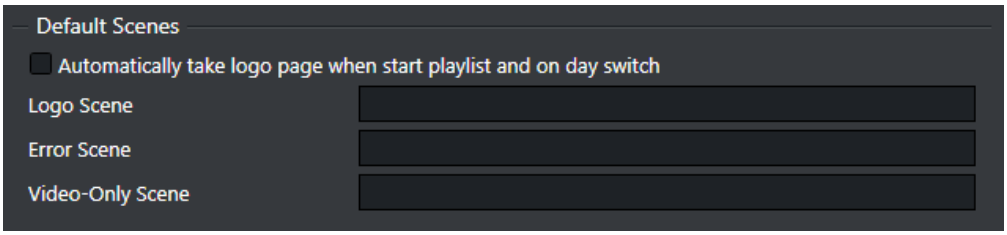
Copyright © 2024 Vizrt

Viz One Integration



- **Automatically extract Meta Data from Viz One elements:** If this option is checked, Viz Channel Branding will add metadata from the Viz One item as primary event attributes while building video elements when importing a schedule.
- **Create placeholder in Viz One for missing House Id's:** Optionally create a placeholder in Viz One for any requests towards a Viz One item where the House Id is not found in the Viz One system yet.

Default Scenes

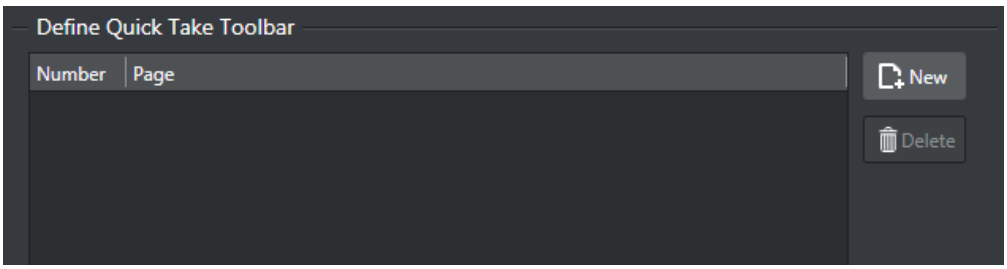


- **Automatically take logo page when starting a playlist and on day switch:** Once the day switch occurs, the logo scene goes on air.
- **Logo Scene:** Enter the scene name of the required logo scene to be played.
- **Error Scene:** Enter the scene name of the error scene that should be easily accessible in the event of an error.
- **Video-only Scene:** If *Play Video Through the Viz Engine* has been selected in the Basic Settings, enter the scene name of the video-only scene that will be played when Viz Channel Branding has no templates or playlists in memory.
  If you have a Matrox X.mio 2 plus card this is not required.

> **Tip:** The above scenes can also be set using the context menu available to pages listed in the Templates, Pages and Pilot Data Elements pane.

Copyright © 2024 Vizrt

Define Quick Take Toolbar



For adding and defining **Quick take** buttons to show in the Client menu bar.

- Press **New** to add an entry and define what page should be *taken* when pressed.

Copyright © 2024 Vizrt

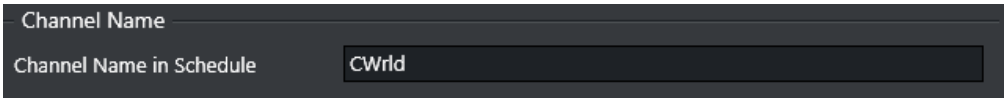
5.2.3 Schedule Import (Playlist Importer Only)

- General Settings
- Mapping Settings
- Filtering Options
- Update Settings

Copyright © 2024 Vizrt

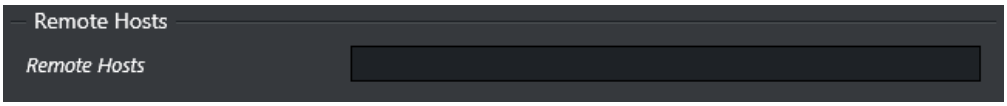
General Settings

Channel Name



- **Channel Name in Schedule:** Sets the channel name that corresponds to the channel name given by the imported schedule.
  Setting the name is useful if there are schedules for multiple channels in import or update folders.

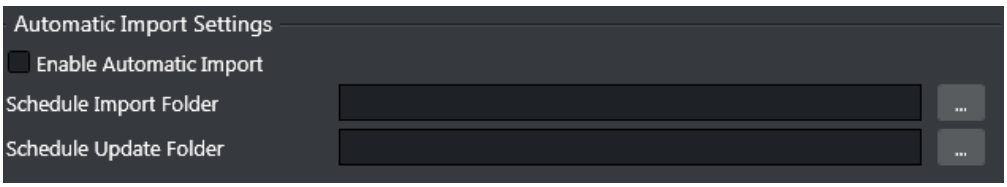
Remote Hosts



- **Remote Hosts:** Defines remote machines that are to receive a copy of the playlist. Multiple hosts are separated by a comma.

> ⚠ **Note:** This option is included for reference and will not be used until a future version.

Automatic Import Settings

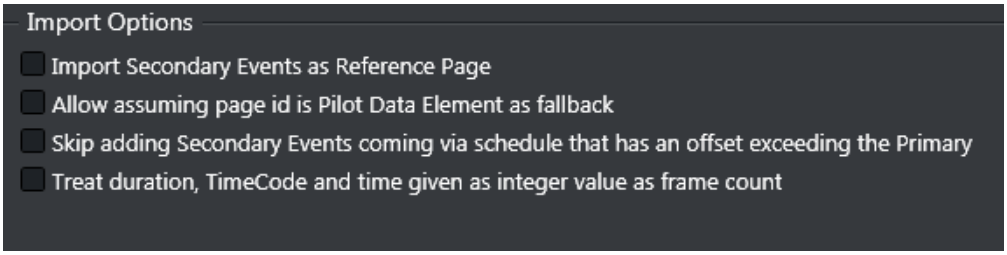


- **Enable Automatic Import:** When checked, automatic imports and updates are enabled. When unchecked, only manual import operations will occur.
- **Schedule Import Folder:** Enter the folder path where schedules imported from external systems will be found.
- **Schedule Update Folder:** The path where a schedule file is located for the purpose of Update of the playlists in the system.

> ⚠ **Note:** It is highly recommended that the **import** and **update** folders be defined as **separate folders**.

Copyright © 2024 Vizrt

Import Options



- **Import Secondary Events as Reference:** When checked, all secondary events associated to Pages will be imported as Referenced Pages.
- **Allow assuming page id is Pilot Data Element as fallback:** When checked, any secondary element with an page id that is numerical and that does not have a corresponding entry in the Page Pool or Action Pool, will be assumed to be a Pilot Data Element, stored in a Pilot Database, and the import process will create a Pilot Data Element reference page in the playlist.
- **Skip adding Secondary Events coming via schedule that has an offset exceeding the Primary**: If the resulting offset is beyond the duration of the Primary Event, skip adding the event to the playlist.
- **Treat duration, TimeCode and time given as integer value as frame count**: If a value comes as a integer override default behavior that is assuming the value is a frame-count.

Copyright © 2024 Vizrt

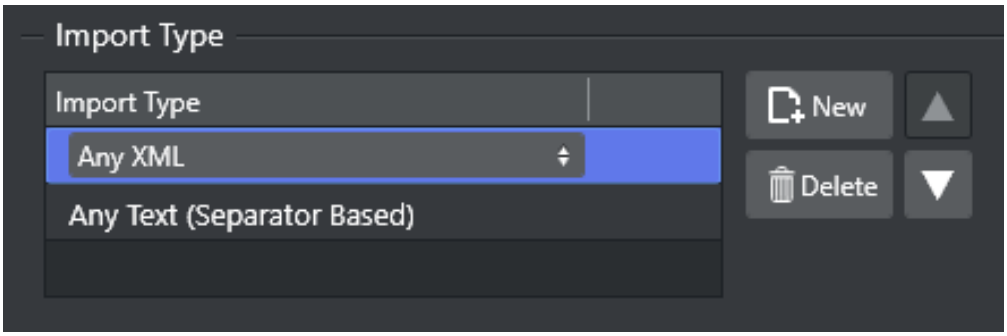
Mapping Settings

- General Settings
- Any Text (Separator Based) Import Settings
- Any Text (Index Based) Import Settings
- Any XML Import Settings
- Excel Import Settings
- LST Import Settings

Copyright © 2024 Vizrt

General Settings

Import Type



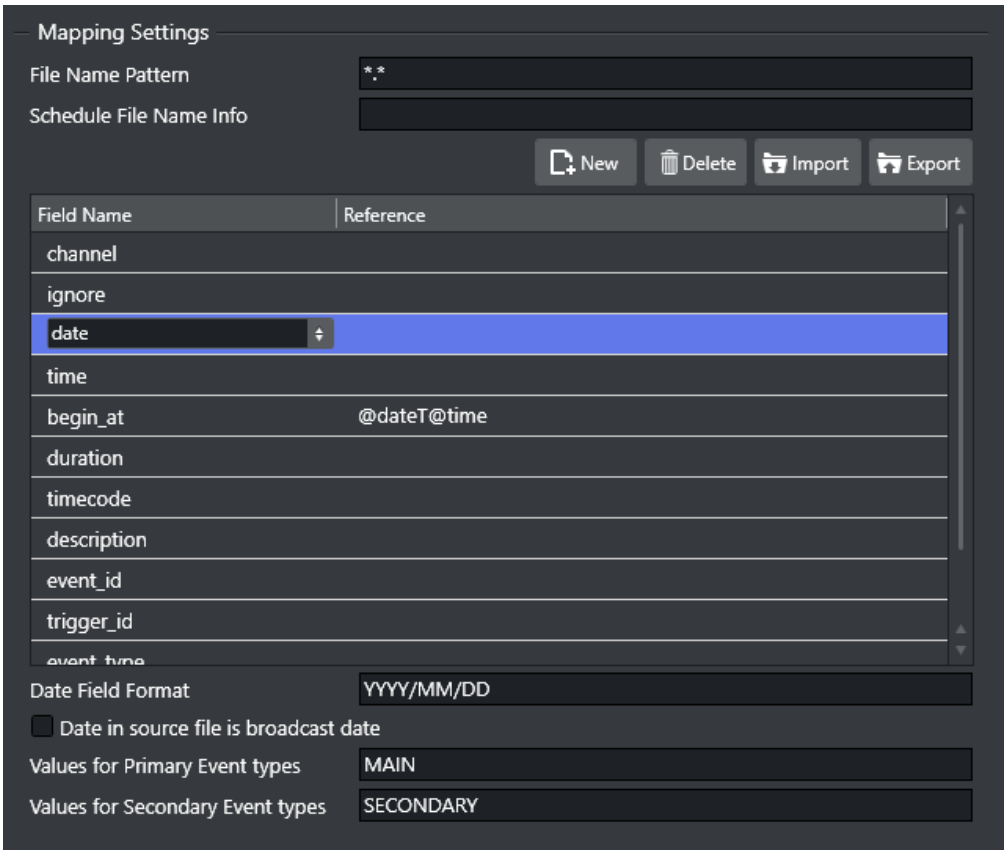
Define what Import Mapping Module to use and what priority order to run them.

Mapping Settings

- **Import Type:** Select the type of external playlist.
- **File name pattern:** Enter the specific file format to search for in the import/update folders, for example `*.xml`, `*.lst`, `*10.txt`, etc.
- **Schedule File Name Info:** When Reading Channel and Date Values from Filenames, indicate the format that the Import/Update process should expect. The filename must contain the D/M/Y to represent the day, month and year character location in the filename. C must represent the channel name. Note that the number of occurrences of the letter C is the same as the number of characters in the channel name. X can be used in order to skip characters.
  - `Channel10_20100824.txt` = `CCCCCCCC_YYYYMMDD`.
  - `20130314_TV2Test.xml` = `YYYYMMDD_XXXCCCC`.

Copyright © 2024 Vizrt

Table Columns



The mapping table is used to add all the rows required to map the source playlist to the playlist to be created.

- **Field Name:** Internal field name to store this bit of data under. You can either edit the value directly or select values from a drop-down list: click the Field Name entry to show a selectable drop-down list of available Field Name types. See the Basic Mapping Table Field Names section for more information on required and optional field names, and how to use additional field names.
- **Reference:** Reference to the data in the schedule file. The reference is then mapped to the *Field Name*. The actual reference code depends on the input data format (textfile, XML etc).

  > **IMPORTANT!** Field names and reference values vary depending on the automation system and data formats in use. For more information, see the Import Guide.

- **Date Field Format:** This field is optional. If defined, the import/update process will expect to find dates in the source file based on the format provided here.
- **Date in source file is broadcast date:** Check this box if the date in the source file is the same as the broadcast date. Otherwise, leave it unchecked.
- **Values for Primary Event Types:** When primary and secondary events are at the same hierarchical level in the source file, enter the text within the fields that denotes primary events. If there is more than one type of denoting text, separate with commas. For

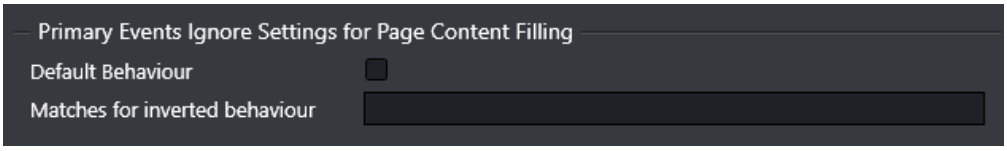
Copyright © 2024 Vizrt

example `Program,Promo,Commercial`. This value will be called from within the `event_type` field.

- **Values for Secondary Event Types:** Enter the text within the fields that denotes secondary events. If there is more than one type of denoting text, separate with commas. For example `Logo,Bug`. When primary and secondary events are at the same hierarchical level in the source file, the value will be called from within the `event_type` field. Otherwise, the value will be called from within the `secondary_event_type` field.

> ⚠ **Note:** If a field name is empty, duplicate or invalid, the field name(s) in error will be shown with a red border. Fields containing errors must be corrected.

Primary Events Ignore Settings For Page Content Filling

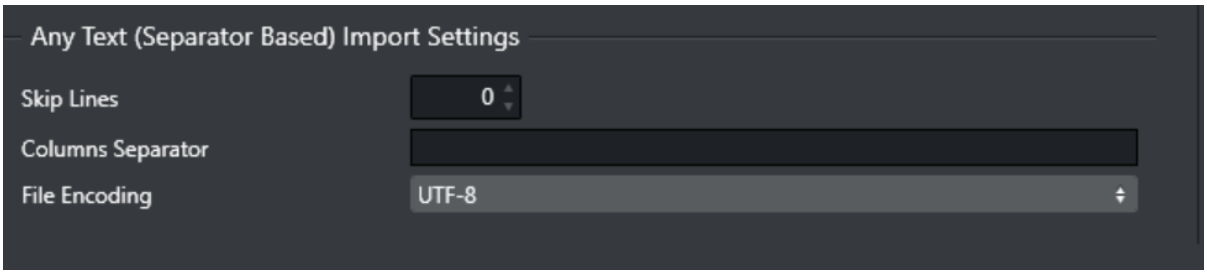


- **Default behavior:** Defines the default behavior during import of primary events for Page Content Filling purposes.
  - When set to FALSE the default behavior is to *use* all primary events for page content filling except those values listed in the Values field.
  - When set to TRUE the default behavior is to *not use* all primary events for page content filling except those values listed in the Values field.
- **Matches for inverted behavior:** Primary events that match the comma-separated values will invert the selected default behavior. Note that you need to fill in a reference for the Ignore field name in the mapping table.
  Supported wildcards:
- `LIVE` - any string that is an **exact** match “LIVE” is a hit, for example: “LIVE”.
- `*LIVE` - any string that **ends** with “LIVE” is a hit, for example: “MUSICLIVE”, “NOTLIVE” and “LIVE”.
- `LIVE*` - any string that **starts** with “LIVE” is a hit, for example: “LIVESHOW”, “LIVEEVENT” and “LIVE”.
- `*LIVE*` - any string that **contains** “LIVE” is a hit, for example: “VERYLIVEMUSIC”, “MUSICLIVE”, “NOTLIVE”, “LIVESHOW”, “LIVEEVENT” and “LIVE”.

Copyright © 2024 Vizrt

Any Text (Separator Based) Import Settings

Any Text (Separator Based) Import Settings

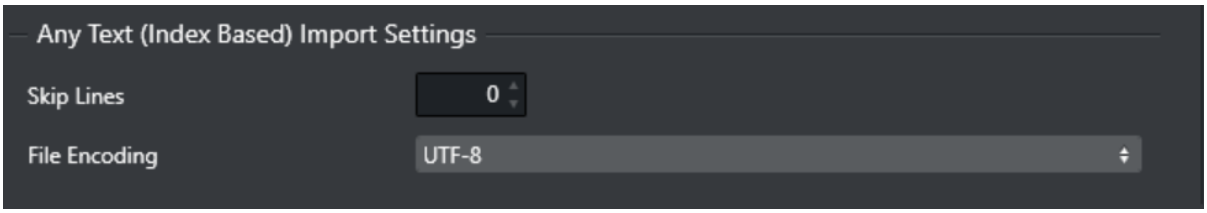


- **Skip Lines:** The number of lines to skip before beginning to read the import source file for data.
- **Columns Separator:** The text symbol (for example `*`) that separates the columns in the text file.
- **File Encoding:** The expected file encoding of the imported text file.
  Alternatives: *UTF-8* and *UTF-16*.

Copyright © 2024 Vizrt

Any Text (Index Based) Import Settings

Any Text (Index Based) Import Settings

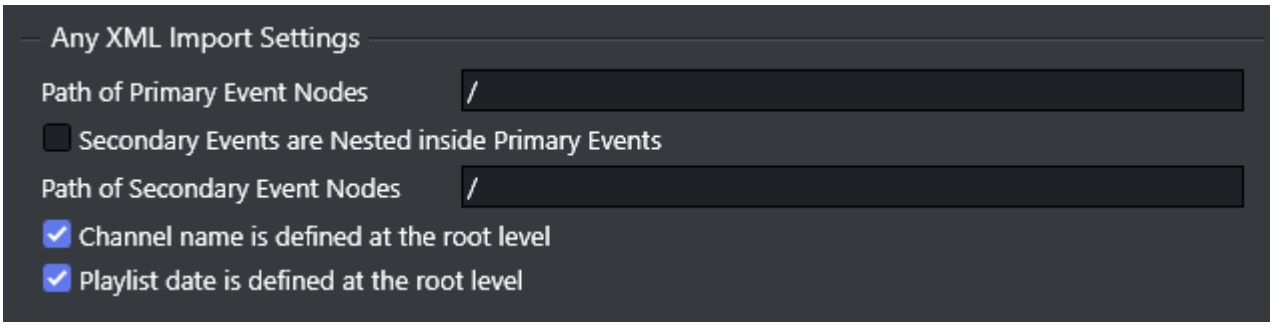


- **Skip Lines:** The number of lines to skip before beginning to read the import source file for data.
- **File Encoding:** Expected file encoding of the imported text file.
  Alternatives: *UTF-8* and *UTF-16*.

Copyright © 2024 Vizrt

Any XML Import Settings

Any XML Import Settings

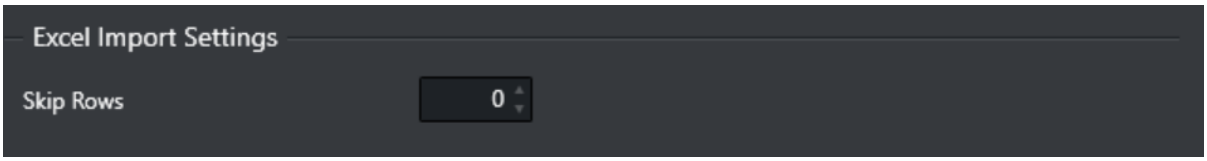


- **Path of Primary Event Nodes:** Path to the element that contains the primary events.
- **Secondary Events Nested in Primary Events:** Check this box if secondary events are nested as child elements within primary event nodes.
- **Path of Secondary Event Nodes:** Path to the element that contains the secondary events.
- **Channel name is defined at the root level:** Assumes that the channel name is defined as a meta data entry (at the root level) outside the primary element entries.
- **Playlist date is defined at the root level:** Assumes that the playlist date is defined as a meta data entry (at the root level) outside the primary element entries.

Copyright © 2024 Vizrt

Excel Import Settings

Excel Import Settings

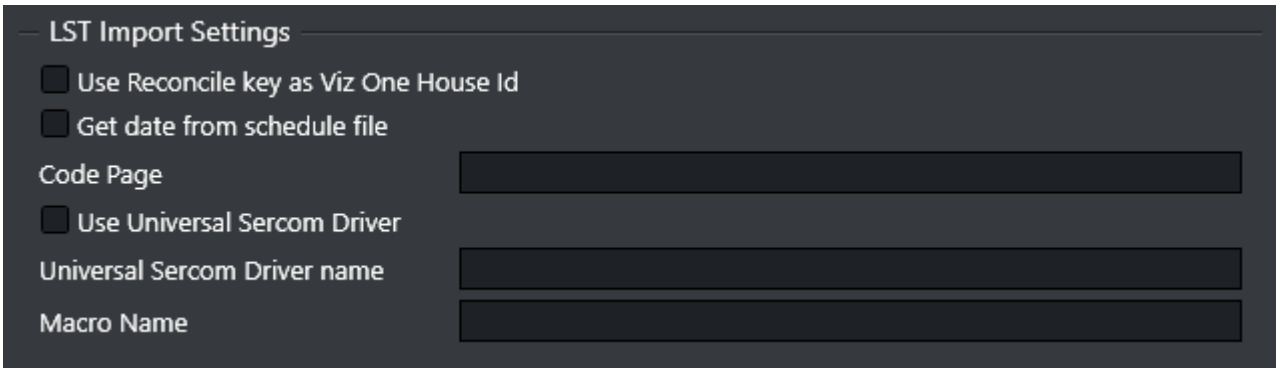


- **Skip Rows:** Number of potential blank lines to be skipped in the Excel file. If there is a header-row, it should also be counted as a row to skip.

Copyright © 2024 Vizrt

LST Import Settings

LST Import Settings

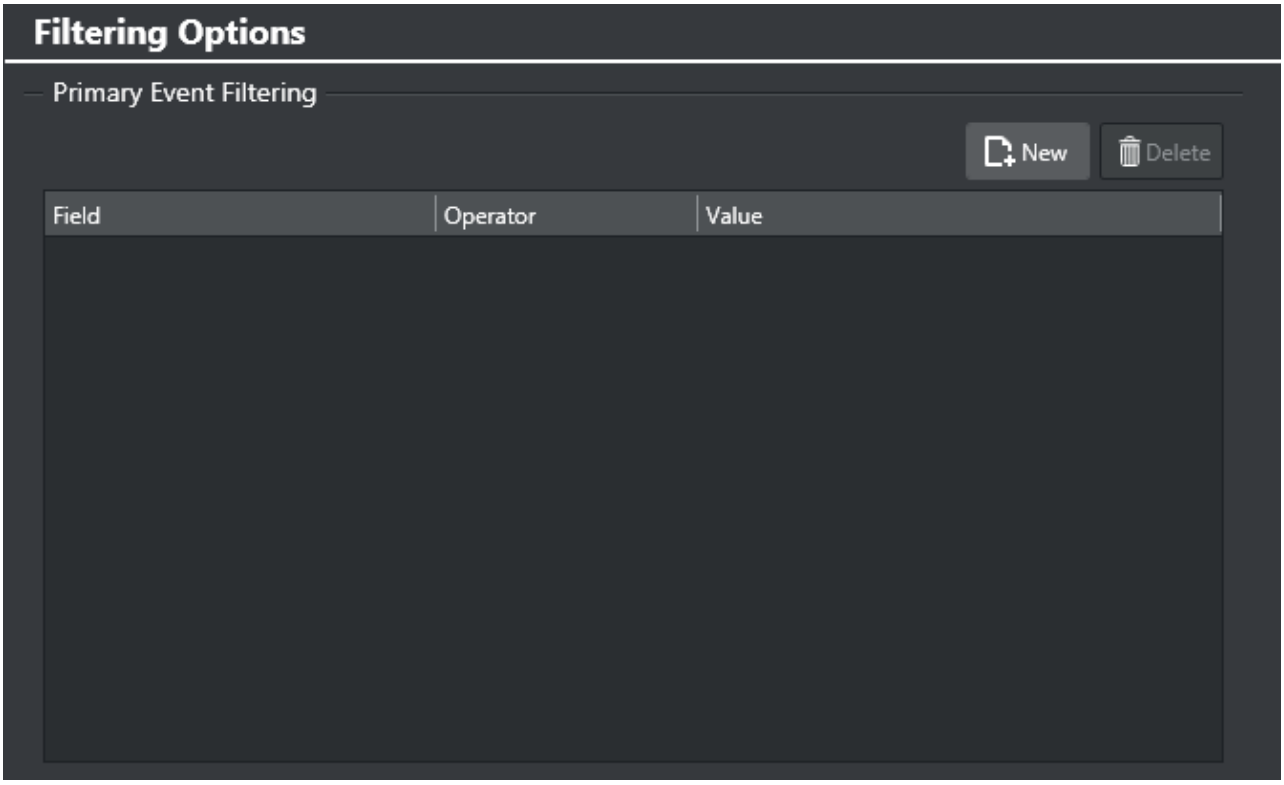


- **Use Reconcile key as Viz One House Id:** After a playlist has been imported, the reconcile keys are used to set the IDs for the Viz One assets.
- **Get date from schedule file:** If the Schedule File Name (see Mapping Settings) is empty, the schedule date will be retrieved from the ADC-100 system's RCV file.
- **Code Page:** During import, bytes in an LST file, stored in the TITLE field, will be converted according to the defined Windows code page. For example, code page with identifier 1256 will convert bytes to ANSI Arabic.
- **Use Universal Sercom Driver:** Enable this when USD is used.
- **Universal Sercom Driver name:** If left empty, exports Secondary Events as standard secondary events for Harris. If defined, exports secondary events as macro commands for that Universal Sercom Driver (USD) device for Harris (ID=USD_NAME, Title=VIZRT:Media ID, Reconcile Key=Template_name).
- **Macro Name:** Sets the macro name used in the Harris system.

Copyright © 2024 Vizrt

Filtering Options

Primary Event Filtering



These settings allow you to filter the import process, so that source primary events must meet certain criteria in order to be included in the playlist.

Creating A Primary Event Filter

1. Select your **channel**.
2. Click the **Settings** button to open the Settings window.
3. Select Filtering Options.
4. Under the **Primary Event Filtering** section click the **Add Condition** icon.
5. Click the **Attribute** field to select an attribute from the Field Name column.
   These relate to both the Mapping Settings and the Filtering Options.
6. Click the **Operator** field and make a selection (see table below).
7. Click the **Value** field and enter a time, number or text. Time format is `hh:mm:ss:ff`.

If the Attribute is begin_time (alias of time), the Value (time) should be in Channel timezone.
If the Attribute is begin_date (alias of date), the Value (date) should be in Channel timezone.

For a list of operator options, see:

Operator	Description
=	equal to

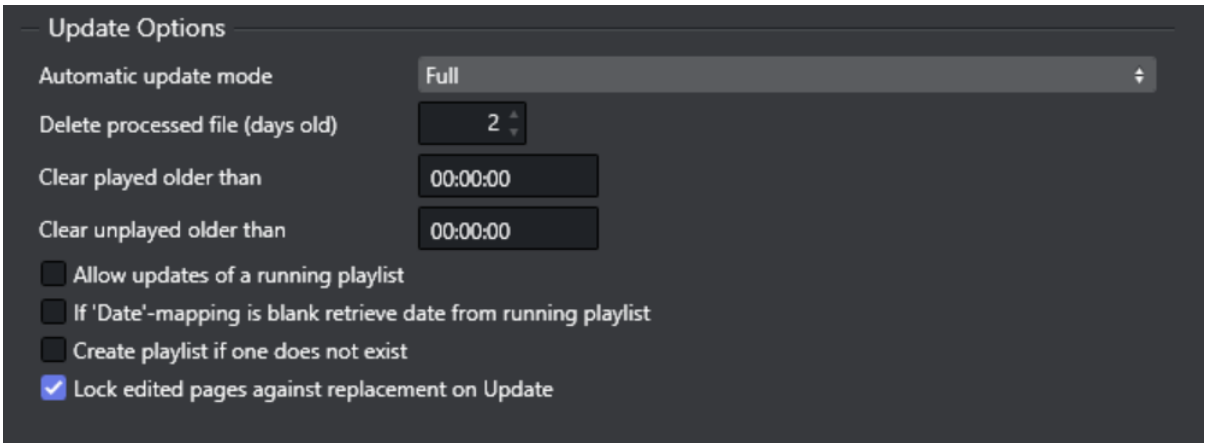
Copyright © 2024 Vizrt

Operator	Description
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
<>	different from
~=	Contain string (fuzzy search)
!~=	Does NOT contain string (fuzzy search)
Reg Exp	A Regular Expression

Copyright © 2024 Vizrt

Update Settings

Update Options



- **Automatic update mode**
  - **Full:** Merges the playlist in its current state in Viz Channel Branding with changes made in the original source file.
  - **Times:** Updates primary event begin time and duration, and any changes made to the original source file. Any primary event additions or deletions are ignored. Only the times are updated.
  - **Full update without event deletion:** Merges the playlist in its current state in Viz Channel Branding with changes that have been made to the original source file, without deleting any (primary) elements.
- **Delete processed files (days old):** The number of days after which processed playlist files (**.done*) will be deleted.
- **Clear played older than (hh:mm):** When updating the playlist, primary and secondary events that have not been played and are older than the specified time, are removed.
- **Clear unplayed older than (hh:mm):** Same as **Clear played older than,** but validation is also performed against unplayed elements.
- **Allow updates for a running playlist**
  - Selected: Playlists that are currently active are updated.
  - Unselected: Only playlists that are not playing are updated.
- **If 'Date'-mapping is blank retrieve date from running playlist:** When selected, an attempt is made to find the date from the running playlist when "Date"-mapping is blank.
- **Create playlist if one does not exist:** If you update a playlist that does not have an existing date, the playlist is created/imported for you.
- **Lock edited pages against replacement on Update:** When selected, any secondary event originating from a schedule that has been manually edited via the client is skipped during next update.

Copyright © 2024 Vizrt

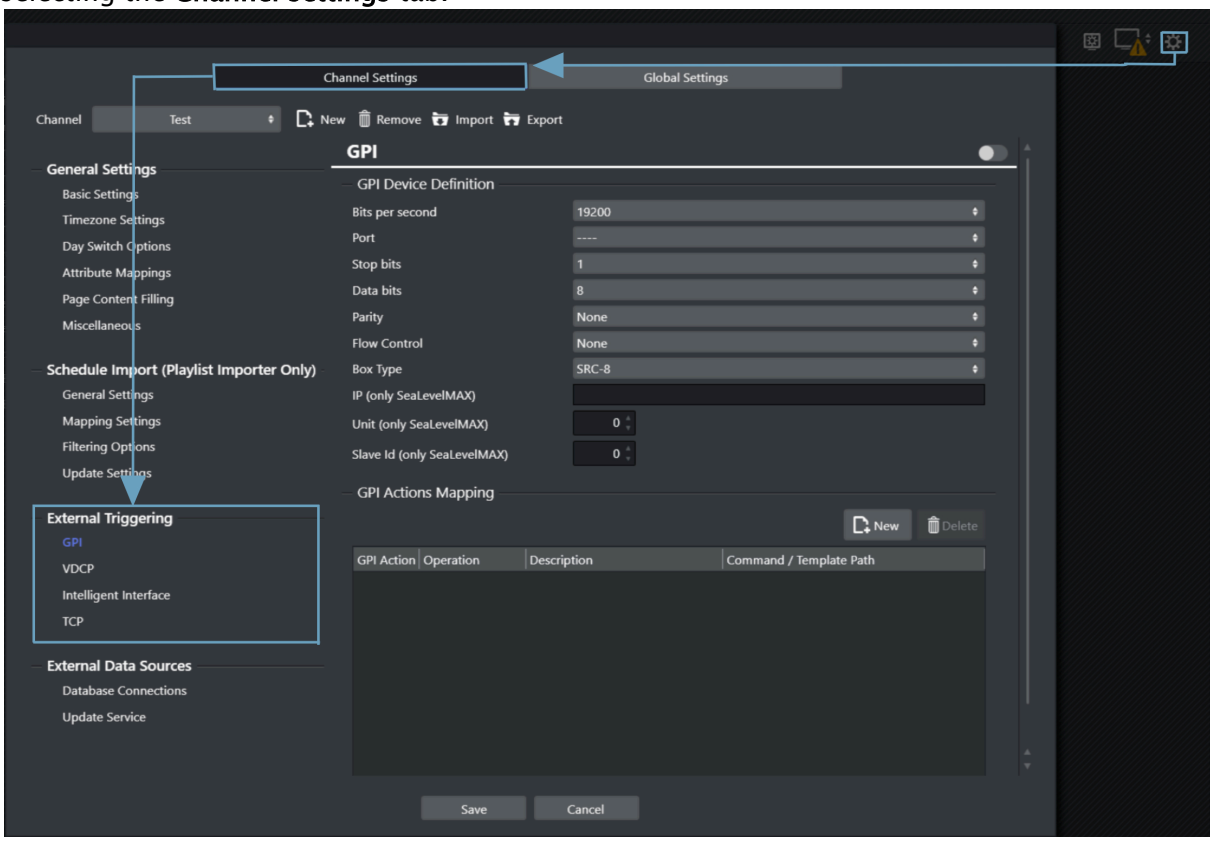
5.2.4 External Triggering

External triggering is relevant to the control room environment as the protocols involved may be used in the ongoing broadcast process.
External triggering is usually *not* relevant to a Traffic Department.

- GPI
- VDCP
- Intelligent Interface
- TCP

Working with External Triggers

- Channel-specific external triggering settings are accessed from the **Cog wheel** icon, then selecting the **Channel settings** tab.



At the top-right corner of the **External Triggering** panel, there is an On/Off (**0** : **1**) switch. You can configure settings with the devices *on* or *off*. When *off*, the settings will take effect the next time the device is turned on.

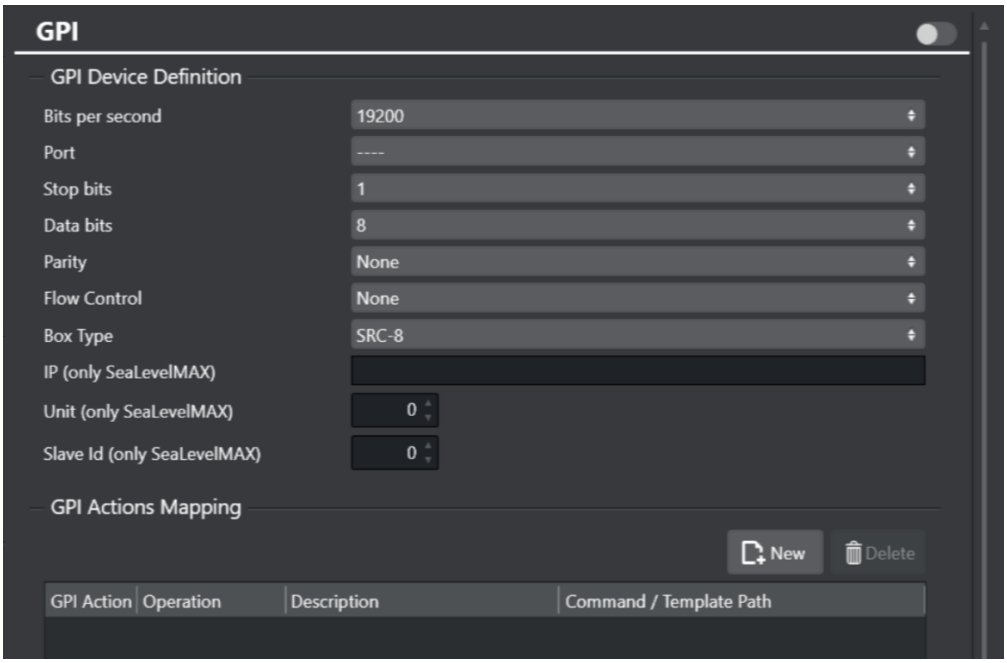
The **External Triggering** settings group comprises:

- GPI
- VDCP
- Intelligent Interface

Copyright © 2024 Vizrt

- TCP

GPI

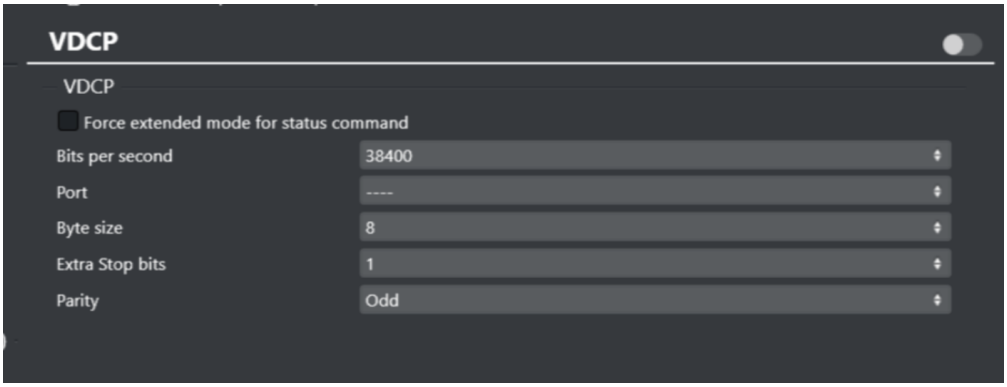


GPI is a type of trigger that does not relate directly to a playlist event. A list of commands allows you to define what happens when a playlist is received from an external device.

- **Bits per second:** Default is **19200**.
- **Port:** Select the relevant COM port on the Media Sequencer machine. Default is "--".
- **Stop bits:** Default is **1**.
- **Data bits:** Default is **8**.
- **Parity:** Default is **None**.
- **Box Type:** Default is **SRC-8**.
- **Flow Control:** Default is **None**.
- Each **GPI Action** is number DL_n_ or DH_n_, where *n* is a sequential number, beginning from 0.
- Under the **Commands** column, all Viz Commands, editable templates and general actions such as show logo, remove logo, stop active playlist are available as options.
- **Description** can be defined freely.
- **IP/Unit/Slave Id:** only for SeaLevelMAX.

Copyright © 2024 Vizrt

VDCP

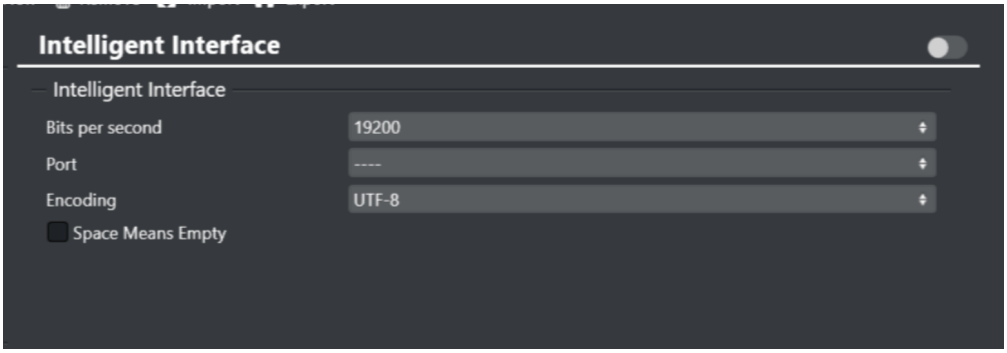


VDCP is a protocol for communicating with external automation systems. Set UI button switch to *ON* if using the VDCP protocol, otherwise set it to *OFF*.

- **Force extended mode for status command:** If checked, all IDs longer than 8 characters in "PORT STATUS REQUEST" and "SYS STATUS REQUEST" VDCP commands will be included.
- **Bits per second:** Default is **38400**.
- **Port:** Select the relevant COM port on the Media Sequencer machine.
- **Byte size:** Default is *8*.
- **Stop bits:** Default is *1*.
- **Parity:** Default is *Odd*.

> **Tip:** When setting up the interface to an automation system, start Media Sequencer in *console* mode to see if VDCP commands are properly sent and received as required. In normal production mode is it recommended to run Media Sequencer as an automatically started service process.

Intelligent Interface



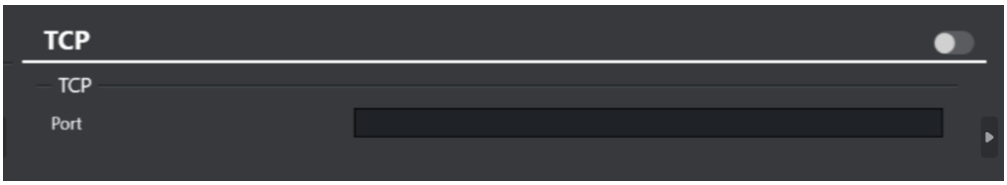
Intelligent Interface is one of the protocols used to communicate with external automation systems.

- **Bits per second:** Default is **19200**.
- **Port:** Select the relevant COM port on the Media Sequencer machine.
- **Encoding:** Default is **UTF-8**.

Copyright © 2024 Vizrt

- **Space Means Empty:** Default is **No**.

TCP



If you want to receive TCP triggers, you must set a Media Sequencer listening port.

- **Port:** The listening port number. This option is disabled when the field is left blank.

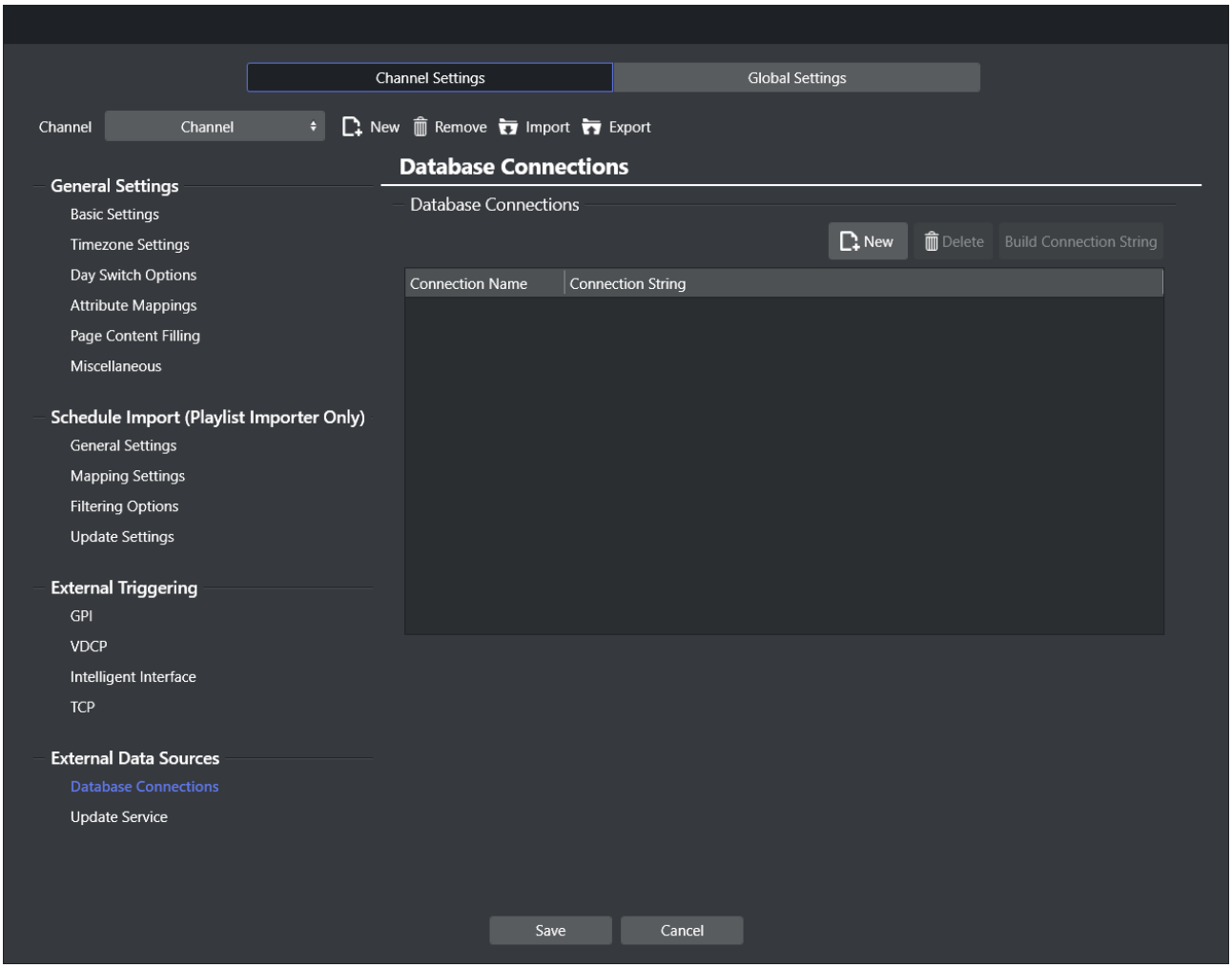
See Also

- Media Sequencer documentation on **pb_listener** actor.
- Media Sequencer documentation regarding **vdcp_client_actor**.
- Media Sequencer documentation regarding **Intelligentif actor**.

Copyright © 2024 Vizrt

5.2.5 External Data Sources

Channel-specific general settings are accessed from the cog wheel icon. Select the **Channel settings** tab.



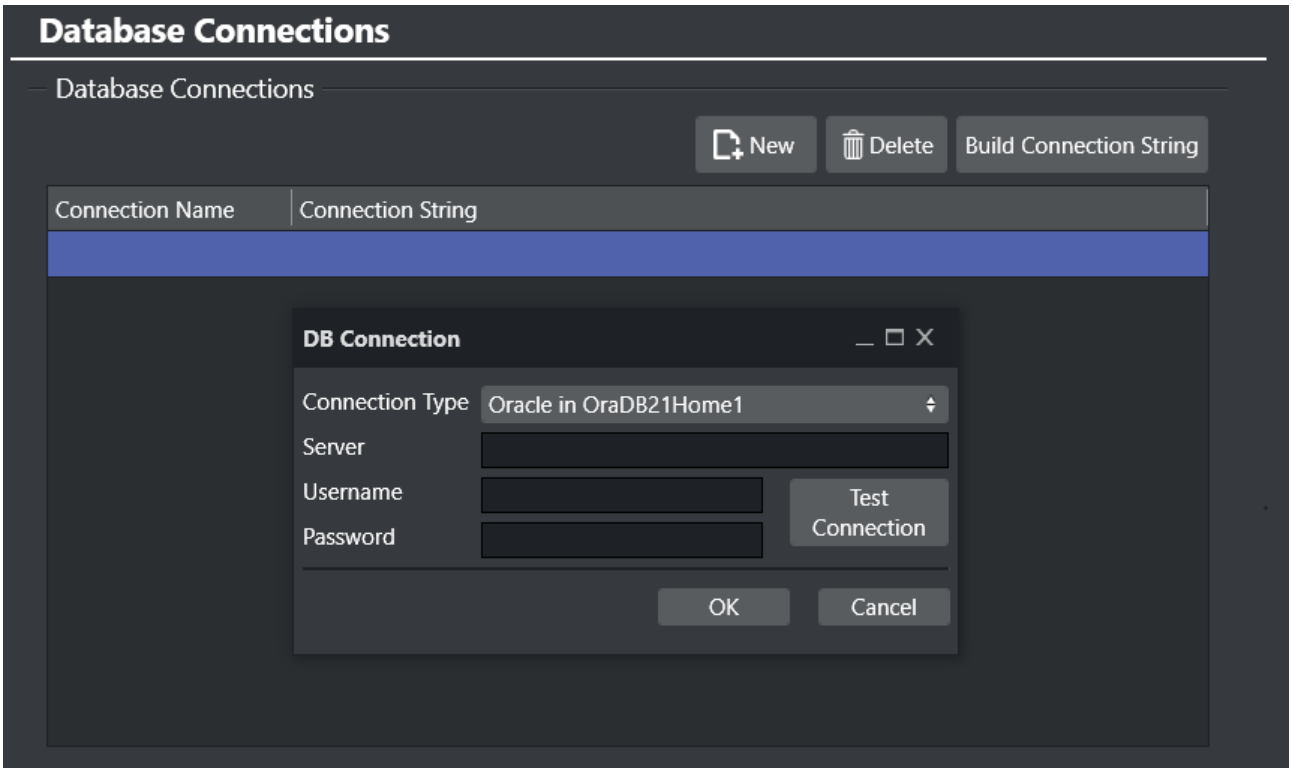
The **External Data Sources** settings group comprises:

- Database Connections
  - Creating Database Connections
    - To configure the connection to an external data source
    - To map from a Microsoft Excel spreadsheet
    - To map from a Microsoft Access database
    - To map from a Microsoft Access text
    - To map from a Microsoft SQL server database
    - To map from a MySql database version 5.1
    - To map from a MySql database version 8.0 using Unicode
    - To map from an Oracle database
- Update Service

Copyright © 2024 Vizrt

- Timeout

Database Connections



This section describes the database connections interface, listing all available database connections.

- **New connection (first button):** Allows you to manually enter a connection string. Clicking the New connection button adds a new entry to the list of available connections.
- **Delete connection (second button):** Deletes the selected connection.
- **Build connection string (third button):** Allows you to build a connection string. Clicking the Build connection string button opens a dialog box that allows you to set all parameters needed in order to successfully connect to your source.

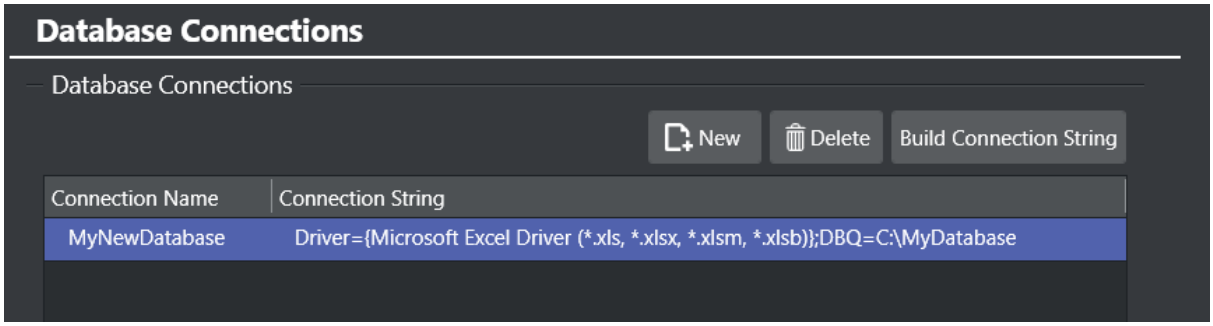
Creating Database Connections

Before populating page content from database look-ups you must first configure the connection to the external data source. You can then proceed with mapping according to your data source.

To Configure The Connection To An External Data Source

1. In the Settings dialog, click **Database Connections**.
2. Click **New**.
3. Enter a name for the connection.
4. Enter the connection string provided by your administrator.

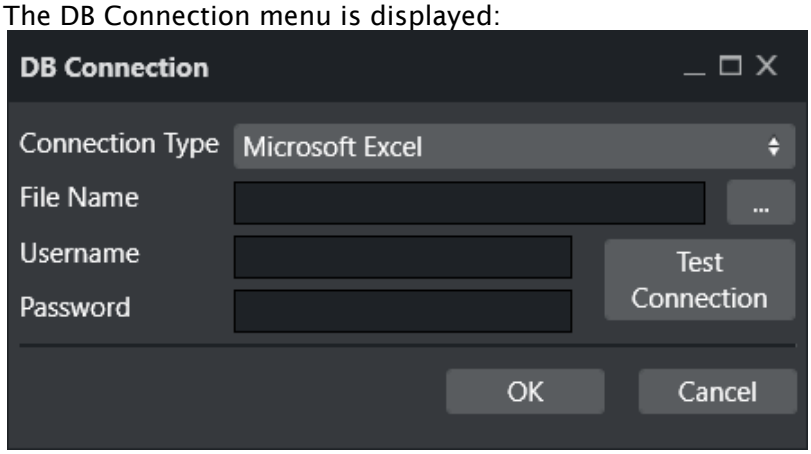
Copyright © 2024 Vizrt



5. Click **OK**.

To Map From A Microsoft Excel Spreadsheet

1. Perform the procedure To configure the connection to an external data source above.
2. Click **Build Connection String**.
   The DB Connection menu is displayed:

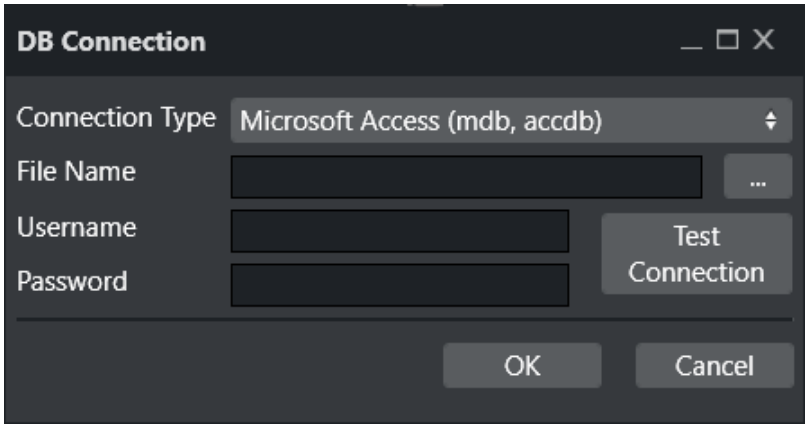


3. For Connection Type, select **Microsoft Excel**.
4. Click the ellipsis to select the desired Excel file.
5. Enter the database's username and password.
6. Click the **Test Connection** button.
7. If the test is successful, click **OK**.

```
Driver={Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm,
*.xlsb)};DBQ=<filepath>;UID=<username>;PWD=<password>
```

To Map From A Microsoft Access Database

1. Perform the procedure Configure the Connection to an External Data Source.
2. Click **Build Connection String**.
   The DB Connection menu is displayed:



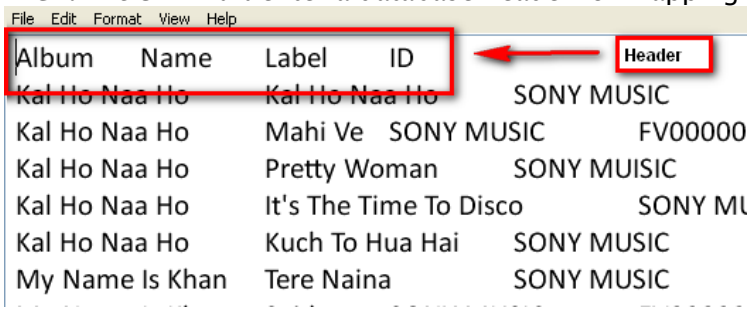
3. For Connection Type, select **Microsoft Access (mdb, accdb)**.
4. Click the ellipsis to select the desired Access Database file.
5. Enter the database's username and password.
6. Click the **Test Connection** button.
7. If the test is successful, click **OK**.

```
Driver={Microsoft Access Driver (*.mdb,
*.accdb)};DBQ=<filepath>;DefaultDir;DriverId=25;FIL=MS
Access;MaxBufferSize=2048;PageTimeout=5;UID=<username>;PWD=<password>
```

To Map From A Microsoft Access Text

Preliminary setup:

1. Create a folder for the text file database, for example *C:\Channel32DB*.
2. In the folder, create two files, such as *Channel32DB.txt* and *Schema.ini*.
   The structural details for the *Schema.ini* file can be found at:
   - http://msdn.microsoft.com/en-us/library/ms709353(v=VS.85).aspx
3. In *Channel32DB.txt* enter a database header for mapping.

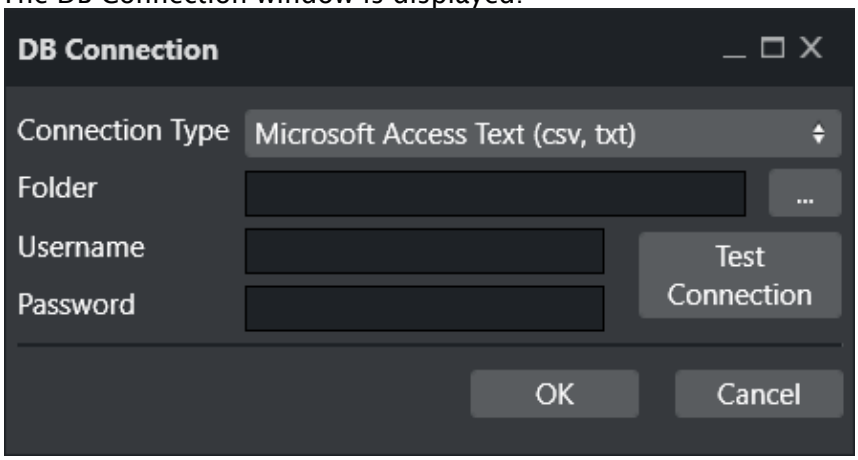


4. In *Schema.ini*:
   - `[v-ind.txt]` : Name of txt file database.
   - `Format=TabDelimited` : This example uses TabDelimited.
   - `ColNameHeader=True` : If using a header (recommended), set to `True`.

⚠ **Note:** Another option is not to use a header, in which case you set this parameter to `False`; the column mapping will then be F1, F2, F3 and F4.

Build connection string:

1. Perform the procedure Configure the Connection to an External Data Source.
2. Click the **Build Connection String** button.
   The DB Connection window is displayed:



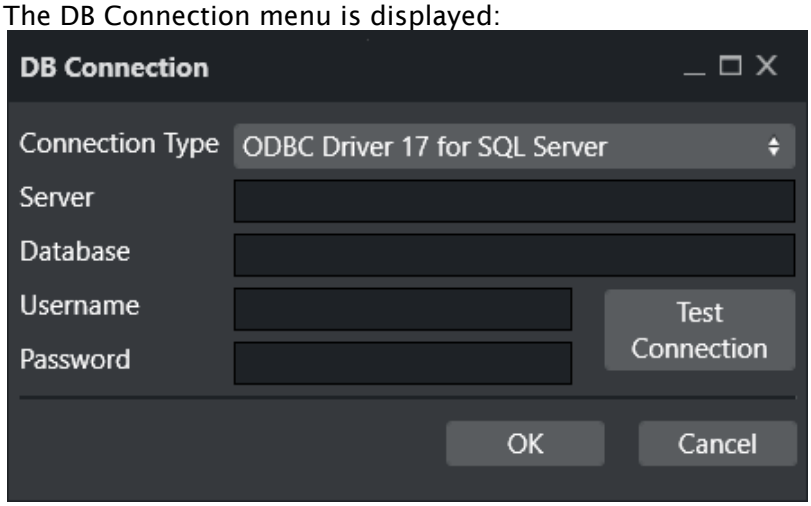
3. For Connection Type, select *Microsoft Access Text (csv, txt)*.
4. Click the ellipsis to select the folder of the desired Access Text file.
5. Enter the database's username and password.
6. Click **Test Connection**.
7. If the test is successful, click **OK**.

```
Driver={Microsoft Access Text Driver (*.txt,
*.csv)};DBQ=<folderpath>;Extensions=asc,csv,tab,txt;UID=<username>;PWD=<passwor
d>
```

To Map From A Microsoft SQL Server Database

1. Perform the procedure Configure the Connection to an External Data Source.

2. Click **Build Connection String**.
   The DB Connection menu is displayed:

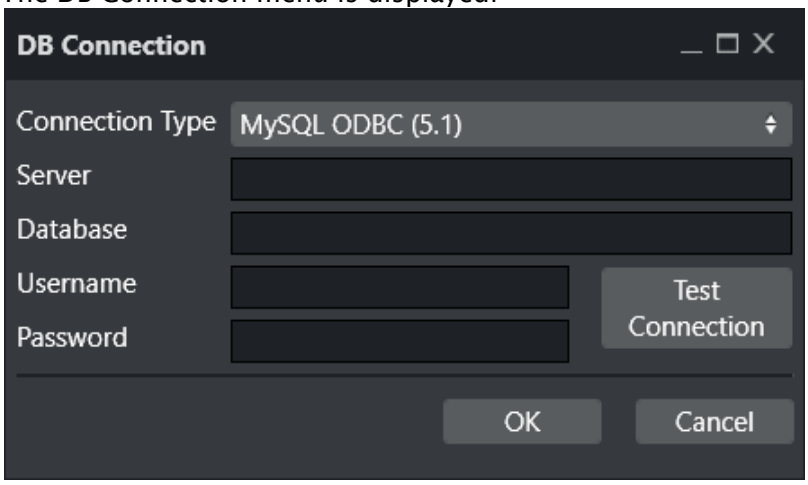


3. In **Connection Type**, select *ODBC Driver 17 for SQL Server*.
4. Enter the database's server name, using the following format: *<hostname>:<port>* or *<IP address>:<port>*.
5. Enter the database's name.
6. Enter the database's username and password.
7. Click **Test Connection**.
8. If the test is successful, click **OK**.

```
Driver={ODBC Driver 17 for SQL Server};Persist Security
Info=True;Mode=ReadWrite;SERVER=<hostname>:<port>;DATABASE=<name>;UID=<username
>;PWD=<password>
```

To Map From A MySql Database Version 5.1

1. Perform the procedure Configure the Connection to an External Data Source.
2. Click **Build Connection String**.
   The DB Connection menu is displayed:



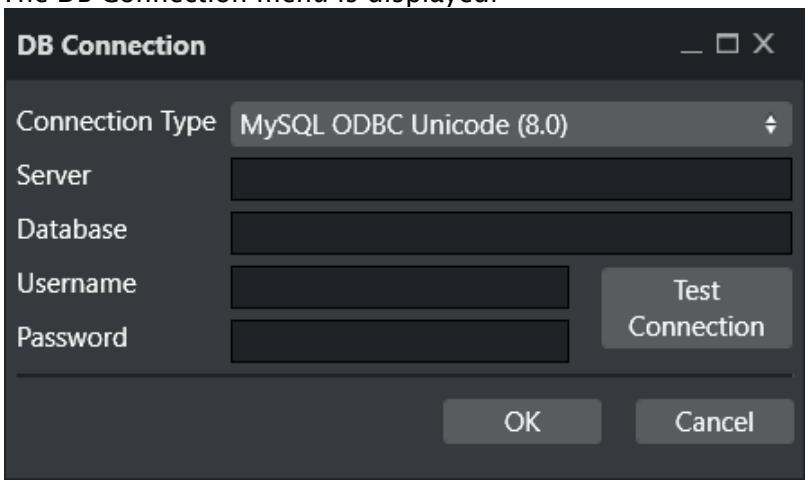
3. For **Connection Type**, select *MySQL ODBC (5.1).*

4. Enter the database's server name, using the following format: *<hostname>:<port>* or *<IP address>:<port>*.
5. Enter the database's name.
6. Enter the database's username and password.
7. Click **Test Connection**.
8. If the test is successful, click **OK**.

```
Driver={MySQL ODBC 5.1
 Driver};SERVER=<hostname>:<port>;DATABASE=<name>;USER=<username>;
PASSWORD=<password>
```

To Map From A MySql Database Version 8.0 Using Unicode

1. Perform the procedure Configure the Connection to an External Data Source.
2. Click **Build Connection String**.
   The DB Connection menu is displayed:

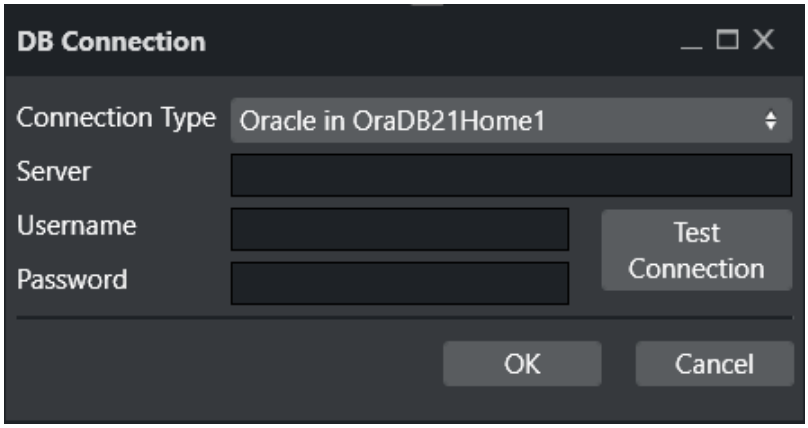


3. For **Connection Type**, select *MySQL ODBC Unicode (8.0)*.
4. Enter the database's server name, using the following format: *<hostname>:<port>* or *<IP address>:<port>*.
5. Enter the database's name.
6. Enter the database's username and password.
7. Click the **Test Connection** button.
8. If the test is successful, click **OK**.

```
Driver={MySQL ODBC 8.0 Unicode
Driver};SERVER=<hostname>:<port>;DATABASE=<name>;USER=<username>;
PASSWORD=<password>
```

To Map From An Oracle Database

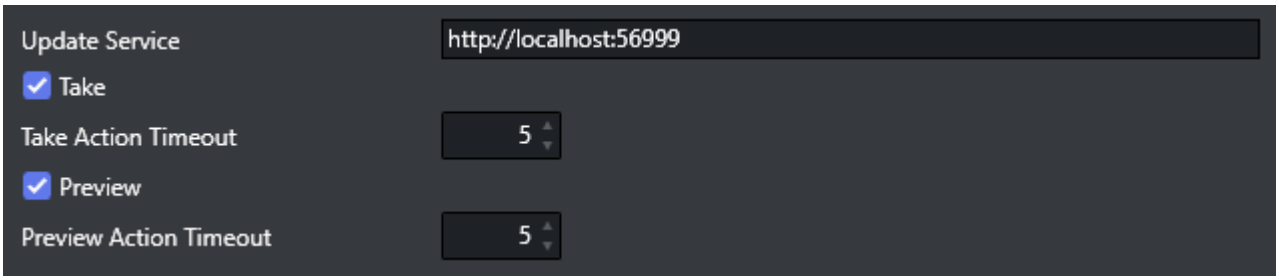
1. Perform the procedure Configure the Connection to an External Data Source.
2. Click **Build Connection String**.
   The DB Connection menu is displayed:



3. For **Connection Type**, select *Oracle in OraDB21Home1*.
4. Enter the database's server name, using the following format: *<hostname>:<port>* or <IP *address>:<port>*.
5. Enter the database's username and password.
6. Click **Test Connection**.
7. If the test is successful, click **OK**.

```
DRIVER={Oracle in OraDB21Home1};SERVER=<hostname>:<port>;UID=<username>;
PWD=<password>
```

Update Service



Update Service is an option that uses your own custom web service for page content filling. Update Service settings are *per channel* settings. In addition you may define separate update service settings *per template*.

When configured and enabled on one or more templates, the Update Service prompts Media Sequencer (MSE) to send an *XML Payload* to a custom web service when a Page is pre-queued for triggering.

The web service can modify the content of the payload and send it back. When Media Sequencer Engine receives the payload parses it and puts the data into the page's tab field.

This could typically be used for sport results or latest headlines from a feed. Since it relies on a custom web service, the solution can be tailored. This update service is *not* an existing server solution, it is a path for external modules to be able to integrate and function with Viz Channel Branding and Media Sequencer.

Copyright © 2024 Vizrt

Timeout

A **timeout** of N seconds may be set on the available actions (**Take** and **Preview**).

This is to ensure that a reply is sent back within the configured timeout (default is 5 seconds) for a given action.

If no reply is received within the timeout, the action has not been sent/issued, even though a reply will be received later.

See Also

- Playlists

5.3 Client Configuration File

Some settings are configured directly in Viz Channel Branding Client's configuration files, rather than through the Configuration menus. The config file *VizChannelBrandingClient.config*, located in the %ProgramData% folder under *\Vizrt\Viz Channel Branding\Client*

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <add key="FileLogLevel" value="100" />
    <add key="WindowState" value="Maximized" />
    <add key="WindowWidth" value="1440" />
    <add key="WindowHeight" value="760" />
    <add key="WindowLeft" value="1926" />
    <add key="WindowTop" value="11" />
  </appSettings>
</configuration>
```

Copyright © 2024 Vizrt

6 Import Guide

This section describes the supported import modules and how to set up Viz Channel Branding to import schedules and thereby create playlists.

To aid in filling out the Mapping Settings and Secondary Mapping Settings, the following topics explain how to perform various types of imports:

- Basic Mapping Table Field Names
- Importing from Text
- Importing from XML
- Importing from Excel
- Harris ADC-100
- Reading Channel and Date Values from Filenames
- Merging Fields
- Splitting Fields

See Also

- Page Content Filling

Copyright © 2024 Vizrt

6.1 Basic Mapping Table Field Names

The following field names are the basic field names used by Viz Channel Branding. Specific import types may have additional requirements.

> **Note:** Field Name variables must follow a specific format: they should start with ASCII character (a-z/A-Z) or underscore (_), followed by alphanumeric characters (a-z/A-Z/0-9) and/or special characters (_-).

6.1.1 Required Field Names

If the following mandatory field names are not present during import, you will not be able to import the schedule.

- **channel**: Is compared against the *Channel Name in Schedule*. If it does not match, primary and/or secondary events will not be imported or updated. If the schedule file has no channel name available you must leave *Channel Name in Schedule* blank.
- **date:** The date the primary event is supposed to be played out.
- **time:** The time the primary event is supposed to be played out.
- **begin_at:** The combination of date and time the primary event is supposed to be played out.
- **event_id:** A unique ID of the primary event, and is used to track updates. When you cannot guarantee unique event IDs, you must create a combined mapping using other Field Names (for example *ReconcileKey* and *segment*).

  > **Note:** The field format (date, time, etc.) depends on the data input type and is defined in Configuration settings.

6.1.2 Optional Field Names

The following field names are optional.

- **ignore:** Used to define which Primary Events Ignore Settings for Page Content Filling. An update of your playlist is always required if these settings are changed.
- **duration:** Is the duration of the primary event. An invalid *duration* value will be regarded as an empty value.
- **description:** Is the description of the primary/secondary event.
- **event_type:** Is the type of event, primary or secondary. Values for primary and secondary event types are set in the Mapping Settings.
- **trigger_id:** Is the ID for triggering events by the automation system.
- **secondary_page:** Is the page name, and must match the name of the page created in Viz Channel Branding Client. See Working with Templates and Pages.
- **secondary_timecode:** Offset relative to the primary event's start time.
- **secondary_duration:** Duration of the secondary event. An invalid *duration* value will be regarded as an empty value.

Copyright © 2024 Vizrt

- **house_id:** Map the house ID to the *house_id* keyword. This enables the system to resolve assets on Viz One (see the Viz One Integration chapter). Will create a composite group (a primary element containing a video asset) in your playlist . The composite group may also contain pages. This requires Viz One 5.4 or later.
- **isLive:** The value can be '1' or 'true' for live events, not case sensitive.
- **isLocked:** The value can be '1' or 'true' if the event is locked, not case sensitive.

> **Note:** When a live event is scheduled at a preset time, the timetrigger will pause. The elements following the live event will be scheduled only when the user clicks Continue from the LIVE show button. When a playlist is resumed from a live event earlier or later than planned, the begin time of events following the live event are shifted to be taken on air immediately. The begin time of events will be adjusted up until the first locked event.

6.1.3 Additional Field Names

Additional field names can be used to add in extra columns in the playlist to show metadata or to add data to pages when editing an attribute (see Page Content Filling Window's Attribute options). You can define as many additional Field Names as you like. Additional Field Names are supported by any format apart from the LST Import Type.

Copyright © 2024 Vizrt

6.2 Importing From Text

The **Any Text** import module supports any format of text file. There are two such types of text file: with a *delimited field* structure and with a *fixed width information* structure.

In a *delimited field structure* file, each line contains a list of values, delimited by a configured character such as a comma or tab. In a *fixed width information* file, each line contains a list of values, and each value is stored in a given place, such as the first to tenth characters.

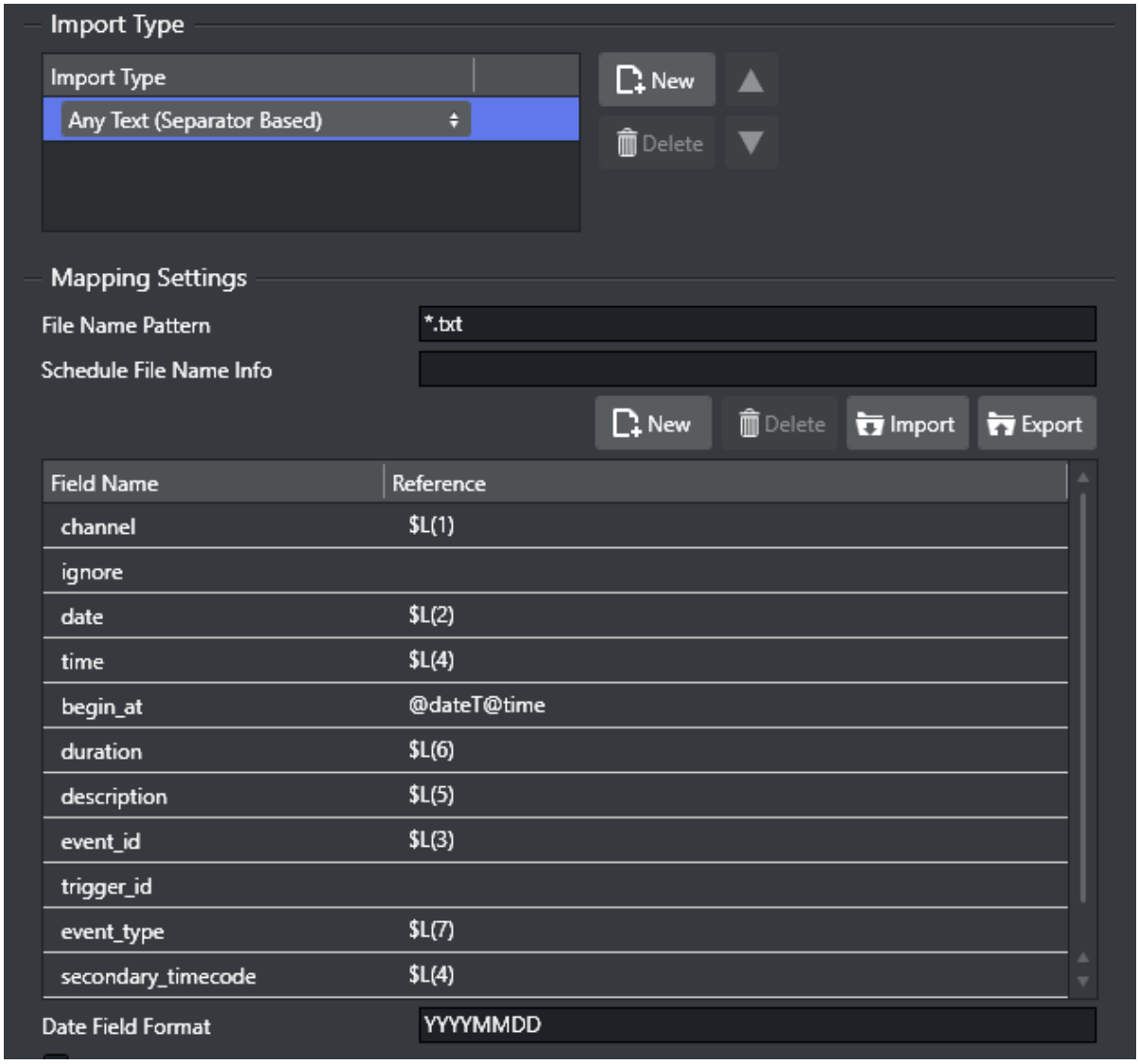
Viz Channel Branding's import facility also supports reading information from the file header lines of the imported file (delimited or fixed width (indexed)).

Before performing either of the procedures below you need to ensure that a **schedule** is available in the required format (Any Text) and a channel has been created.

6.2.1 Import Delimited Files

1. Open the **Settings** window and scroll down to **Schedule Import** section.
2. Select the General Settings section and set the **Channel Name in Schedule**. This procedure uses channel *demo* as seen in the example shown below.
3. Select the Mapping Settings section and set **Import Type** to **Any Text (Separator Based).**
4. Set the **File name pattern** to match the file extension (for example *.txt).
5. For the **Any Text (Separator Based) Import Settings** section enter the **delimiter** being used in the **Column Separator (Any Text)** field (for example *).

Copyright © 2024 Vizrt



6. In Mapping Settings, map the **Field Name** and **Reference** fields by adding a location counter for each relevant **Reference** field.
   - Use the format *$L(counter)*, where *L* refers to the *location* of the segment.
   - $L(1) maps the first value (for example, demo) in the list, $L(2) the second (for example, 20100923), $L(3) the third (for example, 1051270_23_VM_16_01_1), and so on.

```
demo*20100923*1051270_23_VM_16_01_1*05:48:17:07*Le radeau LE CRASH DU VOL
29,,gg,hh*32293*MAIN
demo*20100923*nextshow*00:00:10:00**0*SECONDARY
```

   Note: The segment counter starts at 1, hence, it is not zero-indexed.

7. Click the **Import Schedule** button, select your schedule file and click **Open.**

Copyright © 2024 Vizrt

- This will import the schedule and create a new playlist.
  Note that you may not have all the **secondary elements** (pages) in your playlist. Check the **schedule file** for secondary elements, **create** the pages required and perform an **update** of your playlist.

8. Right-click your playlist (for example, Continuous Playlist) and select **Update Playlist.**
9. Select your schedule file and click **Open.**

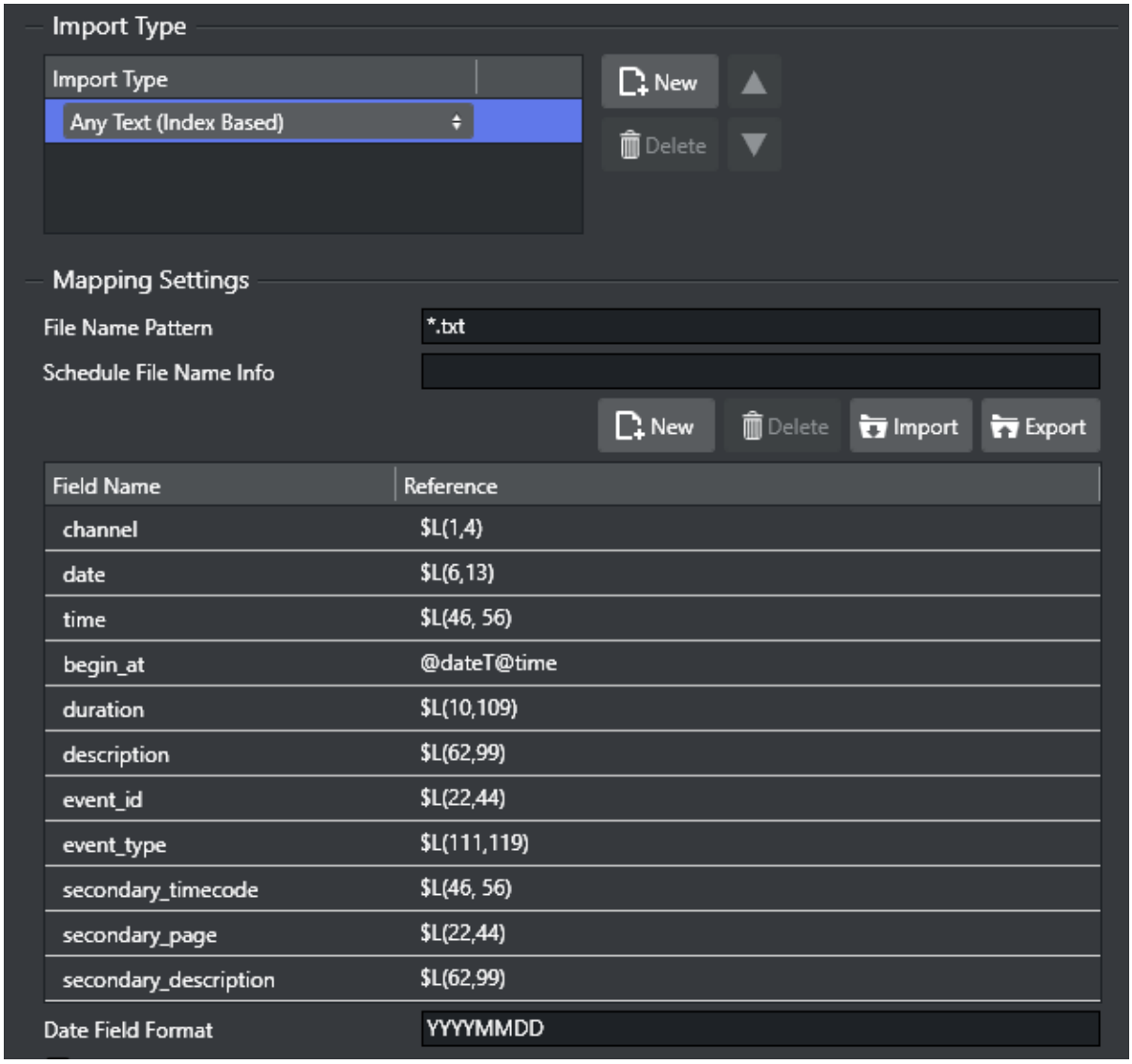
6.2.2 Import Delimited Files with Header Lines

If a delimited file contains header lines, configure the number of header lines for the system to recognize them as such, with the setting **Skip Lines** counter.

6.2.3 Import Fixed Width Files

1. Open the **Settings** window and scroll down to **Schedule Import** section.
2. Select the **General** Settings section and set the **Channel Name in Schedule**. This procedure uses channel *demo* as seen in the example shown below.
3. Select the **Mapping Settings** section and set **Import Type** to **Any Text (Index Based)**
4. Set the *File name pattern *to match the file extension (for example *.txt).

Copyright © 2024 Vizrt



5. In the **Mapping Settings**, map the **Field Name** and **Reference** fields by adding starting and ending character indexes for each relevant Reference field. Use the format *$L(start- index, end-index)*, where *L* means line.

6.2.4 Sample Content – Fixed Width (Indexed)

```
demo 20100923       1051867_1_VM_16_01_1   07:50:01:23    Man of Steel - Official Trailer 3      32293   MAIN
demo 20100923       1053297_21_VM_16_01_1  08:10:05:00    The Avengers Behind the Scenes         4271    SECONDARY
```

Example: *$L(1,4)* maps the first value (for example, demo) in the list, *$L(6,13)* the second (for example, 20100923), and so on.

If the imported file contains header lines, configure the number of the header lines for the system to recognize them as such, with the setting **Skip Lines** counter.

Copyright © 2024 Vizrt

6.3 Importing From XML

The Any XML file import module supports any XML structures that are available for playlist information with the following features:

- Reading values from any node's text or from any node's attributes.
- XML containing primary events only or both primary events and secondary events.
- Secondary events structured at the same level as the primary events or secondary events nested inside each primary event.
- Reading the channel name and playlist date from every primary event or from one external location taken by path relative to the XML root.

Note: All the settings mentioned in the following procedure are located in the **Mapping Settings**.

6.3.1 Map Primary and Secondary Events

Map Primary and Secondary Events

1. Open the **Settings** window and scroll down to **Schedule Import** section.
2. Select **Mapping Settings**.
   See examples below on how to set the paths in advance.
3. Under **Any XML Import Settings** set the parameters **Path of Primary Event Nodes** and **Path of Secondary Event Nodes** to the main event's path relative to the XML root.
   - Do not include the XML root node in the path and/or the secondary event nodes' path relative to the main event node.
   - Do not include the root/main event node in the path.
   - Start the path with the "/" character but don't end it with "/".
4. Check the **Secondary events Nested inside Primary Events** box if the **Secondary events** are nested inside primary events.
5. Check the parameters **Channel name is defined at the root level** and **Playlist date is defined at the root level** if these values are at the root level according to the XML format.
6. Map each field relative to the main/secondary event node.
   - Start the path with the "/" character, and do not include the main/secondary node name in the path.
   - If the field value is a node's text, end the path with the "/" character.
   - If the field value is an attribute, write the path, then the "/" character and then "@" with the attribute name.

6.3.2 Secondary Events at the Same Level as Primary Events

The following example shows a one-level XML where the secondary events are placed at the same level as the primary events. Mapping for both events under the **Any XML Import Settings** section will be as follows:

Copyright © 2024 Vizrt

- **Path of Primary Event Nodes**: `/event`
- **Path of Secondary Event Nodes**: `/event`

```
<playlist>
      <event>
         …
         <type>MAIN</type>
      </event>
      <event>
            …
         <type>SECONDARY</type>
      </event>
      <event>
          …
          <type>MAIN</type>
      </event>
           …
</playlist>
```

6.3.3 Secondary Events Nested Inside Primary Events

The following example shows a two-level XML where the secondary events are nested inside the primary events. Mapping for both events under the **Any XML Import Settings** section will be as follows:

- **Path of Primary Event Nodes**: `/primary_event`
- **Path of Secondary Event Nodes:** `/secondary_events/secondary`

```
<playlist>
   <primary_event>
       …
     <secondary_events>
     <secondary>…</secondary>
     <secondary>…</secondary>
     </secondary_events>
   </primary_event>
   <primary_event>
     …
   </primary_event>
</playlist>
```

6.3.4 Channel Name and Playlist Date in a Separate Node

The following example shows how to map the channel name and the playlist date when those tags are placed in other nodes than the default primary element nodes.

Copyright © 2024 Vizrt

⚠ **Note:** Keeping the playlist date separate *limits the playlist to one broadcast day*.

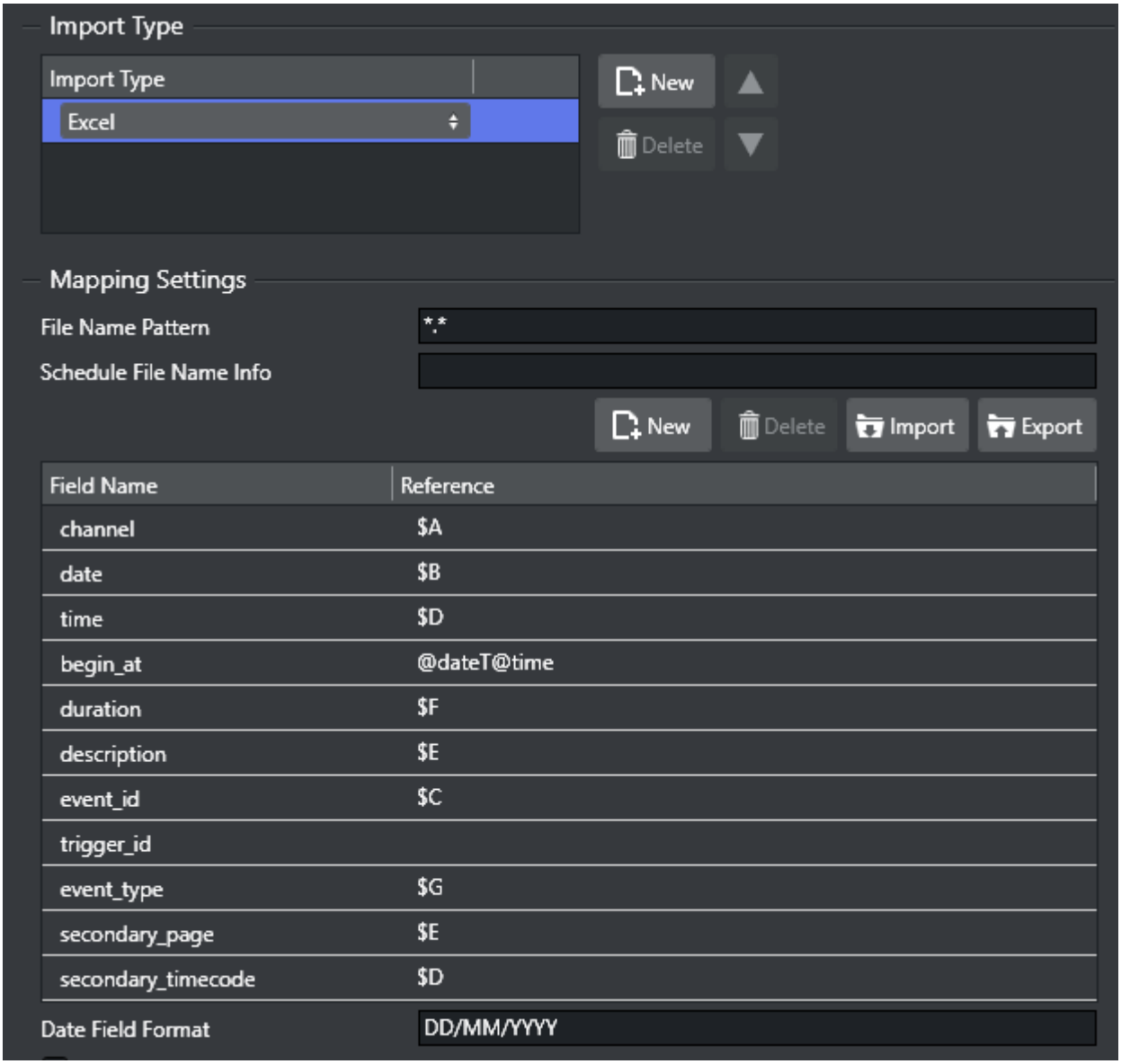
Note that the following example mapping requires you to set an **absolute path** for the channel name and playlist date in your **mapping table**. Your mapping table under the **Mapping Settings** section will be as follows:

- **channel:** `/general_info/channel_name/` (ensure **Channel name** is defined at the *root level* setting in the **Any XML Import Settings** section)
- **date:** `/general_info/date/` (ensure **Playlist date** is defined at the *root level* setting in the **Any XML Import Settings** section)
- **begin:** `/begin_time/@time`
- **duration:** `/duration/@time`
- **description:** `/title/`
- **type:** `/type/`

```
<playlist>
   <general_info>
     <channel_name>…</channel_name>
     <date>…</date>
   </general_info>
     <event>
    <begin_time time="…" />
    <duration time="…" />
   <title>…</title>
   <type>MAIN</type>
   </event>
</playlist>
```

Copyright © 2024 Vizrt

6.4 Importing From Excel



The Excel module supports both *.xls and *.xlsx formats.

Copyright © 2024 Vizrt

	A	B	C	D	E	F	G
1	**Channel**	**Date**	**ID**	**Time**	**Description**	**Duration**	**Type**
2	demo	31/01/2013	1	06:30:00	Despicable Me 2	00:01:35:00	MAIN
3	demo	31/01/2013		00:00:00:00	ticker_system_Main		SECONDARY
4	demo	31/01/2013		00:00:01:00	program_Main		SECONDARY
5	demo	31/01/2013	2	06:31:00	The Incredible BURT WONDERSTONE	00:02:20:00	MAIN
6	demo	31/01/2013		00:00:05:00	BUG		SECONDARY
7	demo	31/01/2013		00:02:10:00	viz_cleanup		SECONDARY
8	demo	31/01/2013	3	06:33:30	Main of Steel - Official Trailer 3	00:02:56:00	MAIN
9	demo	31/01/2013	4	06:47:00	The Avengers Behind the Scenes	00:13:25:00	MAIN
10	demo	31/01/2013		00:00:10:00	BUG		SECONDARY
11	demo	31/01/2013		00:01:00:00	LOWER3RD		SECONDARY
12	demo	31/01/2013		00:13:22:00	viz_cleanup		SECONDARY
13	demo	31/01/2013	5	12:10:00	Coldplay - Viva La Vida	00:03:59:00	MAIN
14	demo	31/01/2013	6	16:00:00	GIRLS GENERATION - Oh	00:03:42:00	MAIN
15	demo	31/01/2013	7	16:30:00	Pitbull - International Love	00:04:02:00	MAIN
16	demo	31/01/2013	8	18:30:00	Now You See Me	00:02:15:00	MAIN

The Excel sheet column is your **Reference** in the mapping table. For example, for column A set $A, and so on.

When working with Excel files, the first row should always be the header row, which must span all columns in use. The **Skip Lines** option should then be set to at least 1 as a value.

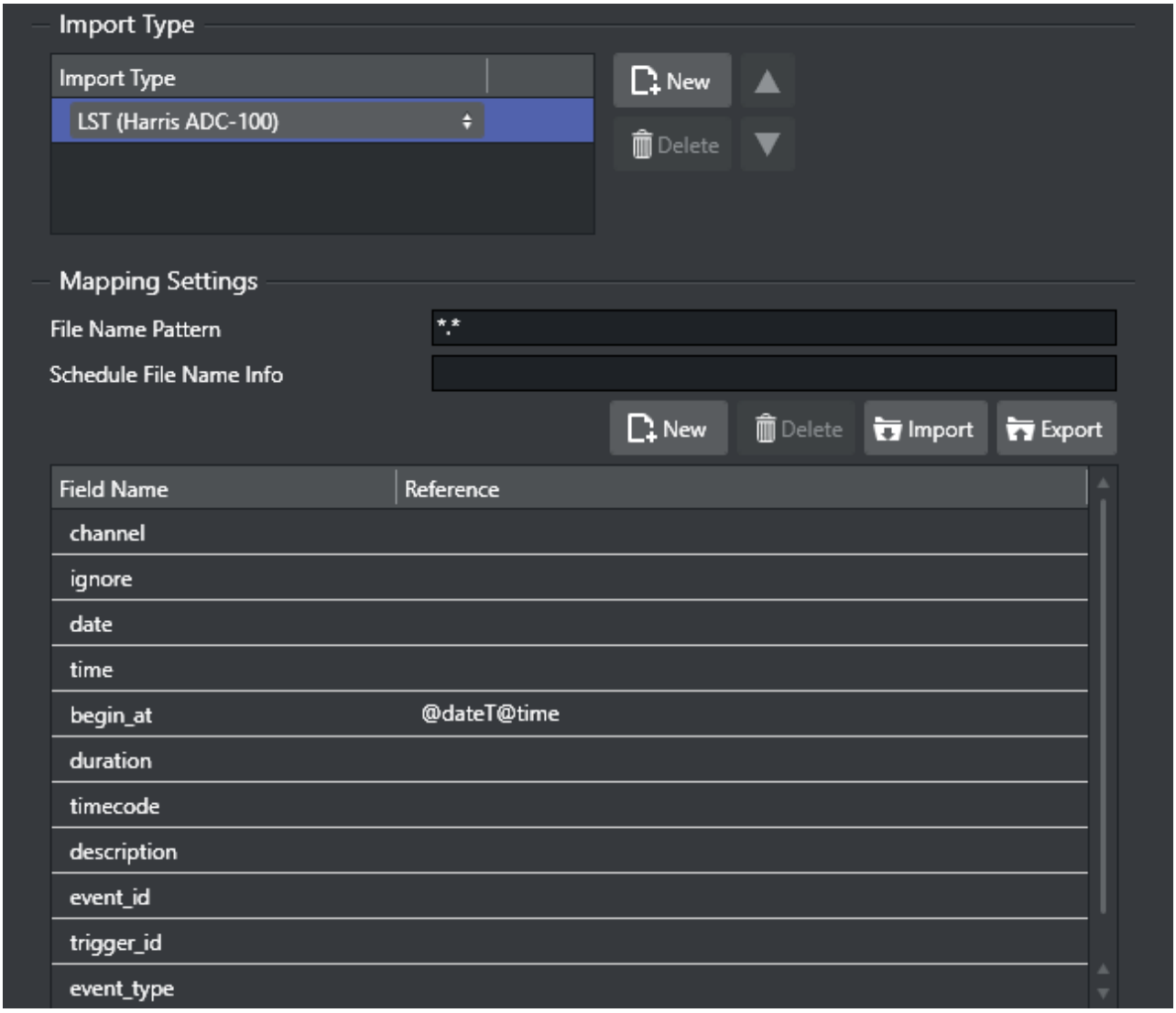
Enter $ and the letter representing the sequential column. You can call multiple columns/elements in a chain. You can also add characters to separate them. For example: `$A / $G / $X`

IMPORTANT! To use the Excel Import Type you must install Microsoft® Excel on the machine that imports the schedule file.

Note: There is a limitation of 100 columns when importing from Excel files.

Copyright © 2024 Vizrt

6.5 Harris ADC-100



The LST import module supports importing schedules based on binary *.lst and *.rcv files from the Harris ADC-100 automation systems.

*.rcv files are generally only used when it is necessary to receive last-minute updates from the Harris automation system.

Before performing the procedure below you must ensure a **schedule** is available in the required format - LST (Harris ADC-100) - and that a **channel** has been created.

6.5.1 To import LST Format Files

1. Open the **Settings** window and scroll down to **Schedule Import** section.
2. Select the **General Settings** section and set the **Channel Name in Schedule**.
3. Select the **Mapping Settings** section and set **Import Type** to **LST (Harris ADC-100)**.

Copyright © 2024 Vizrt

4. **Mapping Table** As LST-files are hard-coded there is no requirement to have any fields pre-mapped
5. Set the **Code Page**, seen under the **LST Import Settings** section.
6. Click **OK**.
7. Click the **Import Schedule** button, select your schedule file and click **Open**.
   This will import the schedule and create a playlist.
   Note that you may not have all the **secondary elements** (i.e. pages) in your playlist. Check the schedule in your Harris system for secondary elements and **Create** the pages required and perform an **Update** of your playlist.
   a. Right-click your playlist (for example Continuous Playlist) and select **Update Playlist**.
   b. Select your schedule file and click **Open**.

6.5.2 To build unique instructions in case of duplicate Reconcile keys in LST

In case of duplicate reconcile keys in the LST file, you can generate a unique instruction by using combination mapping.

In the screenshot below is an example of creating a unique instruction by concatenating the **segment** and **ReconcileKey**, and then use it as **event_id**.

ReconcileKey	@ReconcileKey
segment	@segment
SegmentAndRKey	@segment@ReconcileKey
event_id	@SegmentAndRKey

Note: The fields used for concatenation (segment and ReconcileKey in the example) should be defined before the concatenating field (SegmentAndRKey)

Copyright © 2024 Vizrt

6.6 Reading Channel And Date Values From Filenames

If your collected schedule does not contain *channel* and *date* values, you can use the schedule's file name to extract information.

The import/update process can read these two fields from the imported filename when using

- Importing from Text
- Importing from XML
- Importing from Excel
- Harris ADC-100.

6.6.1 To read *channel* and *date* from the filename

- Set their mapping to *$Filename*
- Define the filename format in the **Schedule File Name Info** parameter in **Mapping Settings**.
- The file format should contain Y/M/D to represent the year/month/day character location in the file name, and C to represent the character locations of the channel name in the file name.
  Characters that do not belong to the channel/date can be written as any character except for Y/M/D/C.

6.6.2 Example

For a filename *Channel10_20090627.txt*, set **Schedule File Name Info** to `CCCCCCCCC_YYYYMMDD` or `CCCCCCCCC_YYYYMMDD.txt`.

Copyright © 2024 Vizrt

6.7 Merging Fields

The ability to join fields and store info in a Mapping Table for an import module and instantly re-use it to another mapped variable. By default, the variable begin_at is mapped up to be a merge of the Date- and Time-field, with a value-separator.

date = /date/
time = /time/
begin_at = @dateT@time

The text between joined fields can be of any size.

title = /title/
date = /date/
description = Video @title playing on @date

Fields with same the same starting name will have the longer field prioritized. In the following example, @titleid will be prioritized over @title"id".

title = /title/
titleid = /titleid/
description = DescriptionFor@titleidInPlaylist

IMPORTANT! The merging field must be located in the table after its parts-fields.

6.7.1 Example 1 for Fixed Width Any Text Import Type

- **StartHour** `= $L(1,2)`
- **StartMin** `= $L(3,4)`
- **StartSecond** `= $L(5,6)`
- **Begin** `= @StartHour:@StartMin:@StartSecond`

6.7.2 Example 2 for Delimited Any Text Import Type

- **StartHour** `= $L(4)`
- **StartMin** `= $L(5)`
- **StartSecond** `= $L(6)`
- **Begin** `= @StartHour:@StartMin:@StartSecond`

Copyright © 2024 Vizrt

6.7.3 Example 3 for Any XML Import Type

This example shows time read in a format `HH-MM-SS-FF` from an entry `/begin_time/time/`.

- **TheStartTime** `= /begin_time/time/`
- **StartHour** `= &(TheStartTime,"-",1,1)`
- **StartMin** `= &(TheStartTime,"-",2,1)`
- **StartSecond** `= &(TheStartTime,"-",3,1)`
- **StartFrame** `= &(TheStartTime,"-",4,1)`
- **Begin** `= @StartHour:@StartMin:@StartSecond:@StartFrame`

Copyright © 2024 Vizrt

6.8 Splitting Fields

The ability to split fields is to be able to annotate in the Mapping Table an instruction for the Playlist Importer on how to extract partial information and mapped it a named variable.

The notation for splitting a field is like this: `&(FIELD_NAME,”DELIMITER”,FROM,COUNT)`. If the delimiter is empty, it will split into characters.

6.8.1 Example

Assuming the schedule file contains the value '31.12.2006' and that value is mapped to variable begin_date_temp, we can then assign this to another variable using the split annotation to extract partial value.

```
&(begin_date_temp,"",1,2) => "31"
&(begin_date_temp,"",11,2) => ""
&(begin_date_temp,"",10,2) => "6"
&(begin_date_temp,".",1,2) => "31.12"
&(begin_date_temp,".",2,2) => "12.2006"
&(begin_date_temp,".",2,3) => "12.2006"
&(begin_date_temp,".",4,1) => ""
&(begin_date_temp,".",3,2) => "2006"
&(begin_date_temp,"/",2,1) => ""
&(begin_date_temp,"/",1,1) => "31.12.2006"
```

Copyright © 2024 Vizrt

7 Working With Scheduling

The following sections describe tools while working with scheduling.

This section covers the following topics:

- Scheduling System Integrations
- Quick Scheduler

Copyright © 2024 Vizrt

7.1 Scheduling System Integrations

The Viz Channel Branding system can be integrated with external systems. Third party systems are often categorized as being a scheduling or an automation system.

In general, the Viz Channel Branding Client has two major working modes: **Scheduling mode** and **Master Control Room (MCR) mode**.

A **Scheduling System** (typically a module within a larger traffic system) is located in the traffic department (or Promotion/Content Department, or Preparation area) and is responsible for building schedules for playout. A schedule will contain all primary events (programs, promotions, commercials, and so on) and may also contain secondary events (Vizrt graphics or other device actions).

The integration with scheduling systems means reading (importing) the schedule that is delivered from them in some way. Reading a schedule can be done in two ways:

1. The scheduling system generates a schedule file in any given format (XML, text, Microsoft Excel, LST and more). The imports of the file parses it to an internal playlist which eventually contains all primary and secondary events (typically Vizrt graphics).
2. When an API is provided by the scheduling system, an Integration Module may exist that can read the schedule directly from it, skipping the file part.
   Optional integration is reporting based on As Run Logs.
3. The automation system can integrate towards the solution using the REST API (v1 or v2) as supplied by Integrations Hub component.

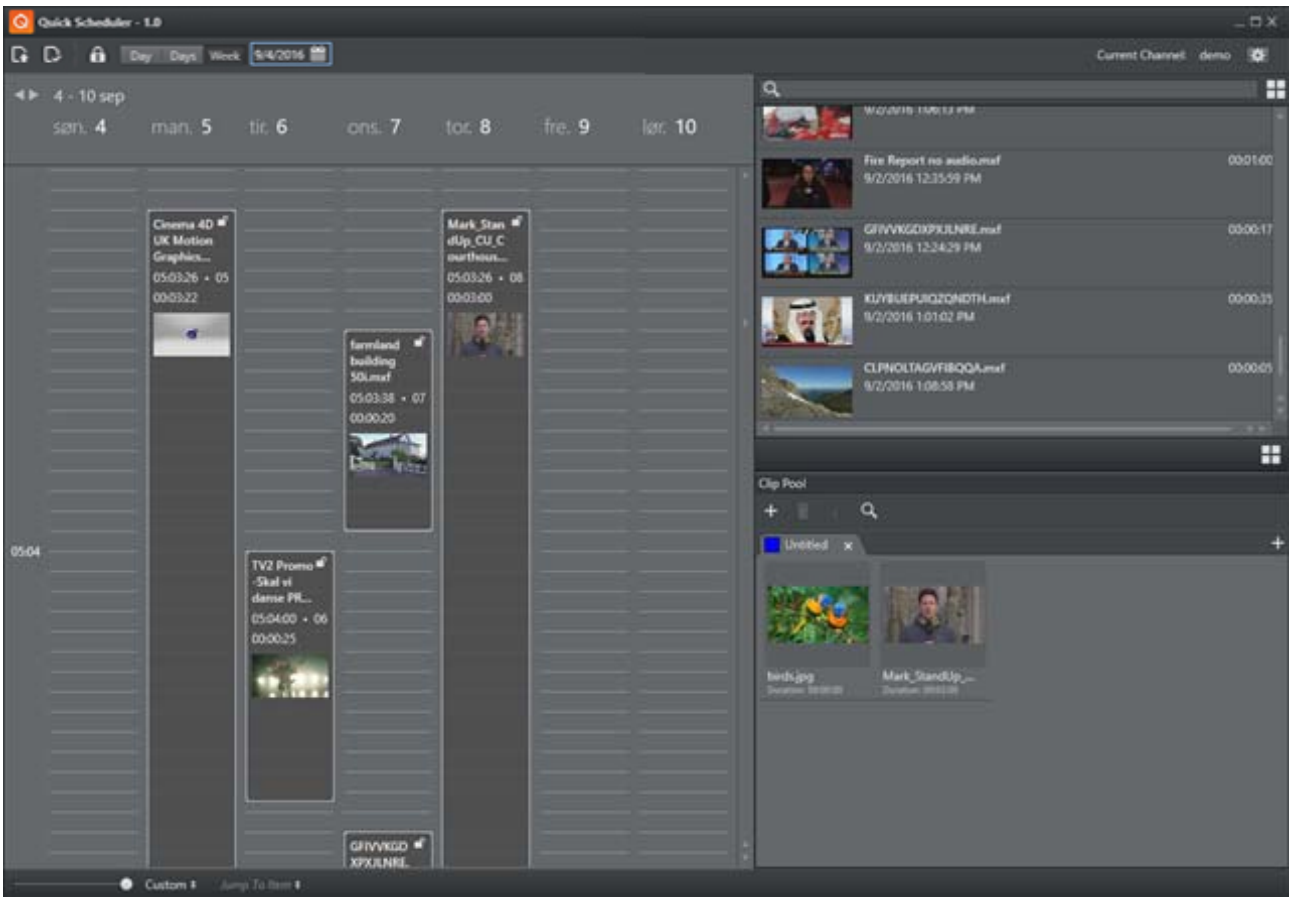
See Also

- Integration with Automation Systems

Copyright © 2024 Vizrt

7.2 Quick Scheduler

Quick Scheduler (QS) is an optional tool for creating schedules. A *schedule* in this context is a playlist with items or complete rundowns. Schedules created with Quick Scheduler can be published to and used by a Viz Channel Branding system using the Quick Scheduler Server.



- Configuration
- Using Quick Scheduler

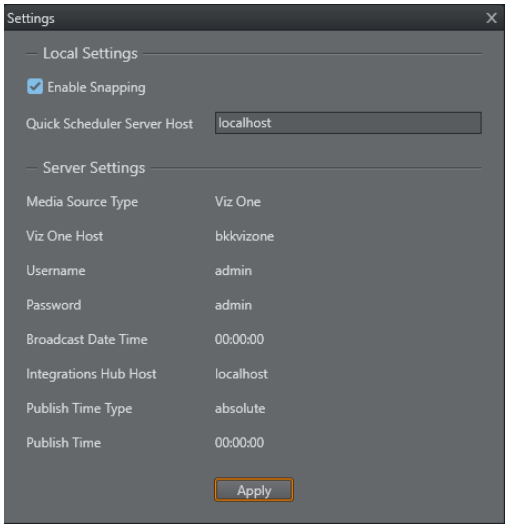
7.2.1 Configuration

You need to configure Quick Scheduler. Depending on your usage you will need to know:

- Enable Snapping: Items on the timeline automatically snap to the closet grid line.
- Quick Scheduler Server Host: Host name Quick Scheduler Server.

  Other Quick Scheduler Server settings are read only.

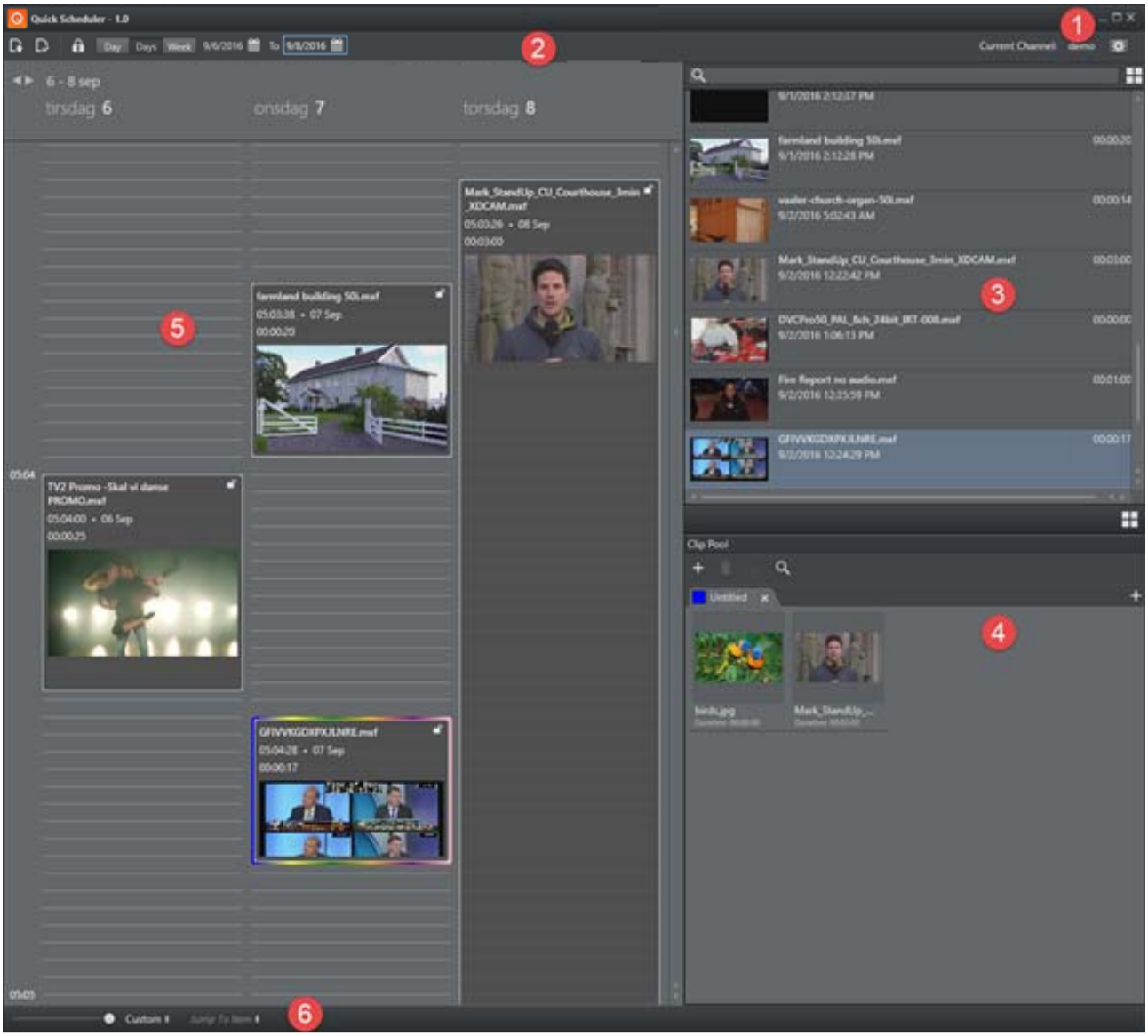
Copyright © 2024 Vizrt



7.2.2 Using Quick Scheduler

Using Quick Scheduler is mostly drag-and-drop. The numbers in the image below correspond to the items in the bullet list.

Copyright © 2024 Vizrt



[1] Click the **Configure** button to configure Quick Scheduler.

[2] Use the **Calendar** buttons to select the date / time view you require. You can view by *date*, *multiple dates* or *week*. The time-view type you select will be reflected in the Schedule-list [5].
Use the **Request Edit Access** buttons to edit items in timeline.

[3] Scroll and view media items (from Viz One or Media Service) in the media pane.
Click the **Grid**/**List** view button for different item views. You can also search for items in the search bar.
Select items by clicking them. Right-click on any item the displayed item-list to:

- Refresh (**F5**)
- Deselect
- Sort by Title
- Sort by Name

Copyright © 2024 Vizrt

[4] The **Clip Pool** list contains items (clips) you often work with, typically recurring clips. Drag items from the media list to the Clip Pool.

[5] The Schedule or playlist items. New items can be created by dragging from the media lists or by right-click in the **Media** panel and select **New Item**.

[6] Use the slider to *Zoom* the Schedule view or use the **Custom** button to select a Schedule view from full day to 15 minutes.

You can also use the mouse wheel to zoom:

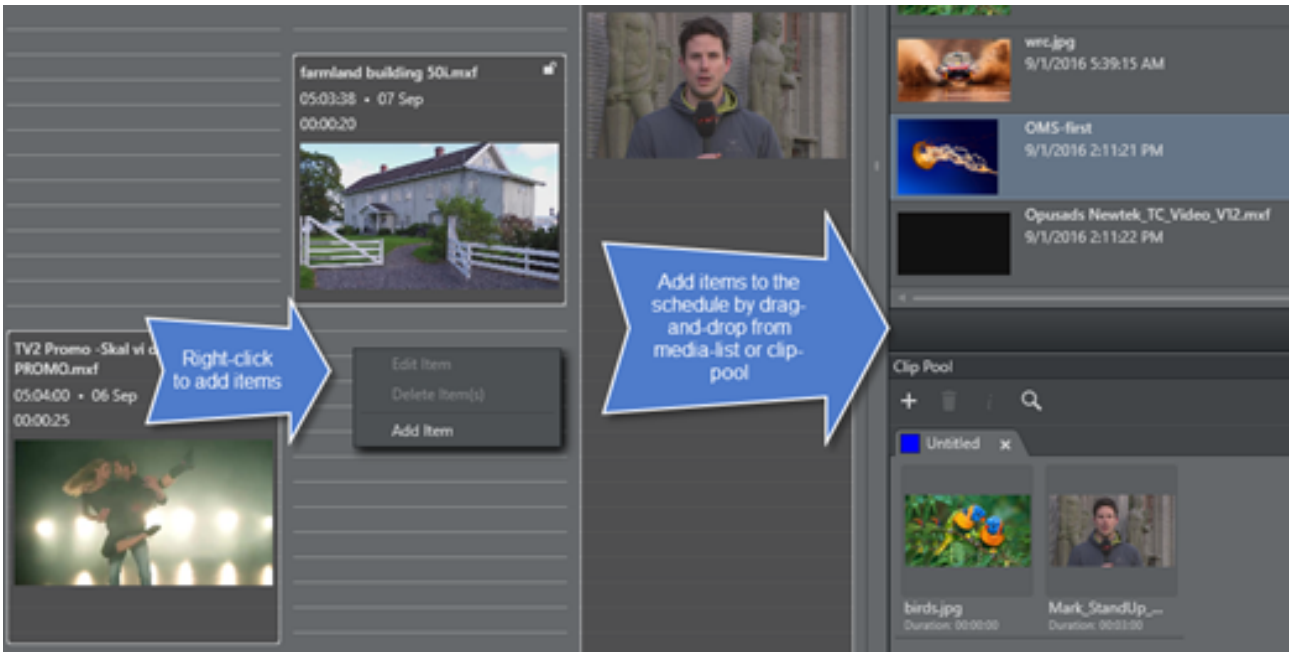
CTRL + Wheel UP: Zoom in

CTRL + Wheel DOWN: Zoom out

CTRL + SHIFT + Wheel UP: Zoom in more granularly

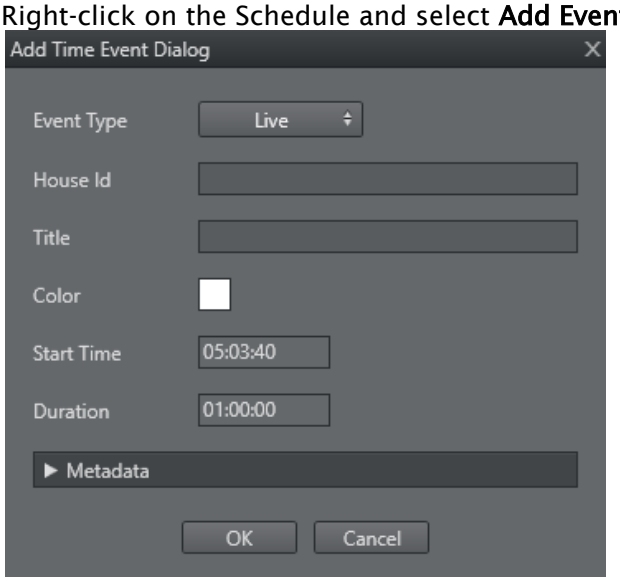
CTRL + SHIFT + Wheel DOWN: Zoom out more granularly

[6] You can use **Jump to Item,** which is *dynamic* (the list is updated depending on what is in the Schedule). This allows you to quickly jump to items in the Schedule.



To add a new item to a Schedule

1. Right-click on the Schedule and select **Add Event** to access the **Add Time Event** dialog:



2. Fill in the various data options as required.
   Event Type can be *Live* or *Placeholder* (*Placeholder* means *reserve a spot in Schedule*).

To modify item metadata in a Schedule

- Right-click an existing item in the schedule and select **Edit item** to view or change metadata for the item.
  If the item originates from Viz One, you can view or change item data by selecting **Edit metadata in Viz One**.

> **Note:** You must have credentials for saving to Viz One to be able to use the options and save options presented.

Copyright © 2024 Vizrt

8 Playlist Importer

Playlist Importer (PLI) is the component responsible for reading a schedule file from disk and translating it to orders against Integration Hub's REST API (v2). The Playlist Importer can be run automatically by the Media Sequencer, or by external scripting.

- Configuration
- Playlist Importer Log File
- Playlist Importer Startup Parameters
- Automatic Import of Files
- Troubleshooting

8.1 Configuration

For the Playlist Importer to be launched by Media Sequencer, the **Enable Automatic Import** option must be selected in the **Automatic Import Settings** section.

IMPORTANT! From Media Sequencer v5.5+ additional steps may be required due to operational security to enable automatic import being launched from a timed interval from Media Sequencer itself.

8.2 Playlist Importer Log File

The Playlist Importer (PLI) application writes a log file when executing. The log file is normally written to the directory: %ProgramData%\Vizrt\Viz Channel Branding\Playlist Importer\

Note: *%ProgramData%* is an environment variable, the fully expanded path will often be *C:\ProgramData\Vizrt\Viz Channel Branding\Playlist Importer*

- The log file is a normal textfile that can be inspected with any text editor. Each time PLI is run, a new log file is created with a timestamp. The verbosity level of the log file is set by the "*-loglevel <LEVEL HERE AS NUMBER>*" startup parameter.
- The loglevel should be set in the channel configuration settings found in **Settings > Global Settings**.

Caution: Care should be taken to delete old log files when they are no longer needed or they will eventually fill up the server hard-drive.

8.3 Playlist Importer Startup Parameters

The following parameters may be used during startup of the Play List Importer.

Copyright © 2024 Vizrt

Parameter	Description	Example
`-channel`	Defines which channels to process within the Integrations Hub/Media Sequencer for the Playlist Importer. Each channel name is bracketed by an '`@`' sign.	`-channel @`*<channelName1>*`@`*<channelName2>*`@`
`-loglevel`	Specifies the log level of the Playlist Importer. Values range from 0 to 100. Default is 25, 50 is Increased, 75 is Extended, 100 is Debug.	`-loglevel 100`
`-inthub`	Host name or IP address for Integrations Hub. When host name is "none" Playlist Importer will read each channel's information from `channels.ini`. When no parameter is provided, Playlist Importer will attempt to connect to *localhost*.	`-inthub none` `-inthub 10.11.10.11` `-inthub localhost`
`--refresh`	PLI will only refresh current running playlist. It will not import/update the playlist	`--refresh`

8.4 Automatic Import Of Files

Files imported automatically by the Playlist Importer are given a timestamp and one of the following suffixes:

- **.done** is appended if the file was successfully imported.
- .**fail** is appended if the file was not successfully imported.
- **.partial** is appended if the file was not fully successfully imported but part of it was.

8.5 Troubleshooting

- Error - Automatic import of Microsoft Excel schedule files fail:
  This error is a known issue with certain Windows installations, particularly when the Media Sequencer is running in *Service* mode.
  To resolve this issue, *create the following directory* on your Windows 64-bit system:
  - C:\Windows\SysWOW64\config\systemprofile\Desktop.
  - Set full control permissions for the new directory.

Copyright © 2024 Vizrt

9 Integrations Hub

The Integrations Hub provides a standard, easy to use interface for Direct Integration modules that can read playlists and schedule data from automation systems, integrating this information with Viz Channel Branding.

In addition, the Integrations Hub provides a REST-based interface for services and components to integrate with the solution. Representational state transfer (REST) is a way to create, read, update or delete information on a server using simple HTTP calls. It is beyond the scope of this manual to fully explain REST-based architecture. For general information the Wikipedia article REST (external link) can be a good starting point.

The Integrations Hub offers Direct Integration modules provided by Vizrt; and customers and 3rd party system vendors and integrators can create additional services with the REST interface.

The Direct Integration (DI) and Direct Integration modules are software tools that directly integrate with automation systems. The integration module handles all the details of interfacing the automation systems and will thereafter deliver the data in a standard http-format to Viz Channel Branding using the REST interface.

This section presents:

- Configuration Check
- Selecting the Integrations Hub Media Sequencer

Note: For REST documentation, see sections:
- REST Interface for Integrations Hub Version 1.0
- REST Interface for Integrations Hub Version 2.0

9.1 Configuration Check

The component will by default connect against the local Media Sequencer and open a couple of ports to receive incoming orders to.

Important! The Integrations Hub runs as a Windows Service process. It is recommended that the Integrations Hub service process is enabled to start automatically and that the Integrations Hub is run as an Administrator service level process.

The default ports it opens at are port 9990 and 9991.

1. Check the installation using Integrations Hub v1:
   Perform a GET towards http://localhost:9990/restapi
   Expected response: HTTP-answer listing all channels
   OR
2. Check the installation using Integrations Hub v2:
   perform a GET towards EITHER http://localhost:9990/v2 OR http://localhost:9991/api/channels

The /v2 is a request for information the address to version 2 of the API, the second link is the answer given. The reply to the second link is a overview of all available channels in the local Media Sequencer.

Copyright © 2024 Vizrt

9.2 Selecting The Integrations Hub Media Sequencer

By default, the Integrations Hub service connects to the local Media Sequencer - the Media Sequencer running at localhost = 127.0.0.1. If required, use the following steps to direct the Integrations Hub to connect to a Media Sequencer at another host:

1. Open Windows File Explorer and browse to the programdata directory for Integrations Hub. The environment variable %PROGRAMDATA% will point to the actual location of the Program Data directory, normally the full path expanded will be C:\ProgramData\Vizrt\Viz Channel Branding\Integrations Hub

   > ⚠ **Note:** `\ProgramData` is normally a hidden Windows directory so you may need to explicitly enter the path `\ProgramData` in Windows File Explorer

2. In the programdata directory, open the *IntegrationsHub.config* file with a text editor, for example Notepad.
3. Find node **appsettings**.
   Change the attributes as needed. Example file:

   ```
   <?xml version="1.0" encoding="utf-8"?>
   <configuration>
     <appSettings>
       <add key="MseHost" value="localhost" />
       <add key="ContentFillingWaitTime" value="2" />
       <!--LogMode can be Off, Basic, Full-->
       <add key="LogMode" value="Basic" />
       <add key="ChannelCacheTime" value="30" />
       <!-- V1 and V2 Port is a port number or 0 if api is disabled.-->
       <add key="V1Port" value="9990" />
       <add key="V2Port" value="9991" />
     </appSettings>
   </configuration>
   ```

   MseHost: IP address or hostname of server the Media Sequencer (MSE) is running on.
   ContentFillingWaitTime: Time in seconds the application will wait before performing fill content.
   This is a method to improve performance of the application due to the burst notifications from integration systems.
   ChannelCacheTime: Time in seconds cache data will be used.
   Value is dependent on usage, but values between 15-30 secs should be reasonable. When this cache-time is elapsed, the application will fetch new data.
   It is important to be aware that ChannelCacheTime will affect settings, rules and pages. If ChannelCacheTime is set to, say, 60 seconds, Channel Settings set via Viz Channel Branding Client could be up to 60 seconds delayed before acted upon.
   Although using cache can improve throughput in an application, it could have side effects due to delay that are hard to catch - always consider your options carefully.
4. Save the file.
5. Restart the windows service.

See Also

Copyright © 2024 Vizrt

- REST Interface for Integrations Hub Version 1.0
- REST Interface for Integrations Hub Version 2.0
- Promo Server

Copyright © 2024 Vizrt

9.3 REST Interface For Integrations Hub Version 1.0

This section describes the simplified REST interface implementation and specification and is of most interest to advanced users or creators of software to integrate with Integrations Hub.

This simplified version 1.0 of the Integrations Hub REST interface uses the standard verbs GET, POST and DELETE and is optimized for speed.

> **Note:** Version 2.0 of the REST Interface for Integrations Hub is a more REST-compliant API set, offering full data representation. For further details, please refer to Appendix section REST Interface for Integrations Hub Version 2.0.

- GET All Channels (Version 1.0)
- DELETE All Channels (Version 1.0)
- POST Channel (Version 1.0)
- GET Channel and its Playlists (Version 1.0)
- DELETE Channel (Version 1.0)
- GET Channel Settings for a Channel (Version 1.0)
- GET Page Pool (Version 1.0)
- POST Pages to the Page Pool (Version 1.0)
- GET Playlist from a Channel (Version 1.0)
- POST Playlist to a Channel (Version 1.0)
- DELETE Playlist (Version 1.0)
- GET Channel Pool from a Channel (Version 1.0)
- POST Group and Page to Channel Pool (Version 1.0)
- GET Group from a Playlist (Version 1.0)
- POST Group to Playlist (Version 1.0)
- DELETE Group (Version 1.0)
- GET Page from a Group (Version 1.0)
- POST Pages to a Group (Version 1.0)
- DELETE Page of a Group (Version 1.0)
- GET All Events with Matching Event Id (Version 1.0)

Copyright © 2024 Vizrt

9.3.1 GET All Channels (Version 1.0)

GET All Channels

To obtain a collection of all channels known by a given Integrations Hub (i.e. channels that exists on the underlying Media Sequencer) use the following command:

Syntax

```
GET /restapi
```

Request

```
GET /restapi HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshub+xml; type=integrationshub
<integrationshub xmlns="http://www.vizrt.com/integrationshub">
  <title>Integrations Hub info for http://localhost:9990/restapi</title>
  <link href="http://localhost:9990/restapi" rel="self" type="application/
vizrt.integrationshub+xml; type=integrationshub" />
  <channel>
    <name>Channel1</name>
    <link href="http://localhost:9990/restapi/Channel1" rel="self" type="application/
vizrt.integrationshub+xml; type=channel" />
  </channel>
  <channel>
    <name>Channel2</name>
    <link href="http://localhost:9990/restapi/Channel2" rel="self" type="application/
vizrt.integrationshub+xml; type=channel" />
  </channel>
</integrationshub>
```

Note

If trying to POST to this root end-point, a *HTTP 403 Forbidden* error will be returned.

Copyright © 2024 Vizrt

9.3.2 DELETE All Channels (Version 1.0)

DELETE All Channels

Channels can only be deleted via the Viz Channel Branding Client.

Syntax

```
DELETE /restapi/
```

Request

```
DELETE /restapi/ HTTP/1.1
```

Response

Returns a **403 Forbidden** status message.

Copyright © 2024 Vizrt

9.3.3 POST Channel (Version 1.0)

POST Channel

A Channel can only be added via the Viz Channel Branding Client.

Syntax

```
POST /restapi/
```

Request

```
POST /restapi/ HTTP/1.1
```

Reponse

Returns a **403 Forbidden** status message.

Copyright © 2024 Vizrt

9.3.4 GET Channel and its Playlists (Version 1.0)

GET Channel

To obtain a collection of the playlists (headers) that exist under a specific channel use the following command.

> **Note:** Each channel has a link to its full collection of groups.

Syntax

```
GET /restapi/[channel_name]
```

Example Request

```
GET /restapi/Channel1 HTTP/1.1
```

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshub+xml; type=channel
<channel xmlns="http://www.vizrt.com/integrationshub">
  <title>Channel info for http://localhost:9990/restapi/Channel1</title>
  <name>Channel1</name>
  <link href="http://localhost:9990/restapi/Channel1" rel="self" type="application/
vizrt.integrationshub+xml; type=channel" />
  <playlist>
    <description>Channel Pool</description>
    <name>channel_pool</name>
    <link href="http://localhost:9990/restapi/Channel1/channel_pool" rel="self" type="
application/vizrt.integrationshub+xml; type=playlist" />
    <ischannelpool>true</ischannelpool>
    <ispagepool>false</ispagepool>
  </playlist>
  <playlist>
    <description>Continuous Playlist</description>
    <name>continuous</name>
    <link href="http://localhost:9990/restapi/Channel1/continuous" rel="self" type="a
pplication/vizrt.integrationshub+xml; type=playlist" />
    <ischannelpool>false</ischannelpool>
    <ispagepool>false</ispagepool>
  </playlist>
```

Copyright © 2024 Vizrt

```
  <playlist>
    <description>Page Pool</description>
    <name>page_pool</name>
    <link href="http://localhost:9990/restapi/Channel1/page_pool" rel="self" type="ap
plication/vizrt.integrationshub+xml; type=pagepool" />
    <ischannelpool>false</ischannelpool>
    <ispagepool>true</ispagepool>
  </playlist>
</channel>
```

9.3.5 DELETE Channel (Version 1.0)

DELETE Channel

A Channel can only be deleted via the Viz Channel Branding Client.

Syntax

```
DELETE /restapi/[channel_name]
```

Request

```
DELETE /restapi/Channel1 HTTP/1.1
```

Response

Returns a **403 Forbidden** status message.

9.3.6 GET Channel Settings for a Channel (Version 1.0)

GET Channel settings for a channel

Query Integrations Hub for a channels settings and their current values, using REST.

> **Note:** Query /channelsettings to get a full list of channels that have settings available, for proper discovery of the URI used in the example below.

Syntax

```
GET /channelsettings/[channelname]
```

Request

```
GET /channelsettings/Channel1 HTTP/1.1
```

Response

Example expected reply:

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshub+xml; type=channelsettings
<channelsettings xmlns="http://www.vizrt.com/integrationshub">
  <title>Channel Settings info for http://localhost:9990/channelsettings/Channel1</
title>
  <name>Channel1</name>
  <link href="http://localhost:9990/channelsettings/Channel1" rel="self" type="applic
ation/vizrt.integrationshub+xml; type=channelsettings" />
  <settingsversion>4.0.0</settingsversion>
  <setting name="GeneralSettings.BasicSettings.ChannelBasicInfo.ScreenDimension">
    <value>1920*1080</value>
  </setting>
  <setting name="GeneralSettings.BasicSettings.ChannelBasicInfo.AspectRatio">
    <value>16/9</value>
  </setting>
[..]
  <setting
name="GeneralSettings.BasicSettings.PlaylistDataMapping.ColumnDataMappings">
    <value>
      <entry attributeName="begin_at" columnName="Begin Time" editable="true" />
      <entry attributeName="duration" columnName="Duration" editable="true" name="ent
ry#2" />
```

```
      <entry attributeName="timecode" columnName="Offset" editable="true" name="entry
#3" />
      <entry attributeName="description" columnName="Description" editable="true"
 name="entry#4" />
      <entry attributeName="event_id" columnName="Event ID" name="entry#5" />
      <entry attributeName="trigger_id" columnName="Media ID" name="entry#6" />
    </value>
  </setting>
[..]
</channelsettings>
```

Copyright © 2024 Vizrt

9.3.7 GET Page Pool (Version 1.0)

GET Page Pool

Syntax

```
GET /restapi/[channelname]/page_pool
```

Example Request

```
GET /restapi/Channel1/page_pool HTTP/1.1
```

Example Result GET page_pool playlist

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshub+xml; type=channel
<pagepool xmlns="http://www.vizrt.com/integrationshub">
  <title>Page pool info for http://localhost:9990/restapi/Channel1/page_pool</title>
  <description>Page Pool</description>
  <name>page_pool</name>
  <link href="http://localhost:9990/restapi/Channel1/page_pool" rel="self" type="appl
ication/vizrt.integrationshub+xml; type=pagepool" />
</pagepool>
```

Copyright © 2024 Vizrt

9.3.8 POST Pages to the Page Pool (Version 1.0)

POST Pages to the Page Pool

Use the command below to add or update a collection of pages for a channel's Page Pool.

The Request (example)/Response example provided shows how to post one page to the Page Pool.

> ⚠ **Note:** Multiple resources can be posted. Each specific element is linked in the verbose response.

Syntax

```
POST /restapi/[channel_name]/page_pool
```

Request (example)

```
POST /restapi/VizMusic/page_pool HTTP/1.1
Content-Type: application/vizrt.integrationshub+xml; type=payload
<payload>
  <page name="example3" template="8000_clip" />
</payload>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshub+xml; type=pagepoolpostreply
<pagepoolpostreply xmlns="http://www.vizrt.com/integrationshub">
  <title>Post result info to http://localhost:9990/restapi/VizMusic/page_pool</title>
  <ispagepool>true</ispagepool>
  <link href="http://localhost:9990/restapi/VizMusic/page_pool" rel="self" type="appl
ication/vizrt.integrationshub+xml; type=pagepoolpostreply" />
  <page>
    <title>Page info for http://localhost:9990/restapi/VizMusic/page_pool/example3</
title>
    <description>Coming Up Next.../Famous Band live and unplugged/...</description>
    <name>example3</name>
    <link href="http://localhost:9990/restapi/VizMusic/page_pool/example3" rel="self"
 type="application/vizrt.integrationshub+xml; type=pagepoolpage" />
    <take_duration />
    <timetolive />
    <attributes>
      <attribute name="description">
```

```
        <value>Coming Up Next.../Famous Band live and unplugged/...</value>
      </attribute>
      <attribute name="guid">
        <value>MuC_Client_52fe29ac-0b20-46c3-a12a-e1684ea75a42</value>
      </attribute>
      <attribute name="showautodescription">
        <value>false</value>
      </attribute>
      <attribute name="layer">
        <value>LTTOGGLE</value>
      </attribute>
      <attribute name="templatedescription">
        <value>Coming up next - Video clip</value>
      </attribute>
      <attribute name="name">
        <value>example3</value>
      </attribute>
    </attributes>
    <fields>
      <field name="clip">
        <mapping />
        <value />
      </field>
      <field name="01UPNEXTLT">
        <mapping />
        <value>Coming Up Next...</value>
      </field>
      <field name="02UPNEXTLT">
        <mapping />
        <value>Famous Band live and unplugged</value>
      </field>
    </fields>
  </page>
</pagepoolpostreply>
```

Copyright © 2024 Vizrt

9.3.9 GET Playlist from a Channel (Version 1.0)

GET Playlist from a Channel

To obtain a collection of playlists associated with a given channel with links to each, use the following GET command. The command will obtain a full representation of the playlist's header and a full list of primary elements with a limited set of data and links to each primary elements.

Syntax for GET

```
GET /restapi/[channel_name]/[playlist_name]
```

Example Request

```
GET /restapi/Channel1/continuous HTTP/1.1
```

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshub+xml; type=playlist
<playlist xmlns="http://www.vizrt.com/integrationshub">
  <title>Playlist info for http://localhost:9990/restapi/Channel1/continuous</title>
  <description>Continuous Playlist</description>
  <name>continuous</name>
  <link href="http://localhost:9990/restapi/Channel1/continuous" rel="self" type="app
lication/vizrt.integrationshub+xml; type=playlist" />
  <broadcast_date>continuous</broadcast_date>
  <group>
    <description>Schedule1</description>
    <name>1958a5d9-87bb-42f0-a208-5ea7968c7649</name>
    <link href="http://localhost:9990/restapi/Channel1/continuous/1958a5d9-87bb-42f0-
a208-5ea7968c7649" rel="self" type="application/vizrt.integrationshub+xml;
type=group" />
    <begin_at>2017-08-28T10:00:00.000Z</begin_at>
    <duration>5289.12</duration>
    <event_id>1001</event_id>
    <trigger_id>1</trigger_id>
    <take_count />
  </group>
  <group>
    <description>Despicable Me 2</description>
    <name>b536b347-ad9a-4ff7-9fb7-b097cd4b8230</name>
```

Copyright © 2024 Vizrt

```
    <link href="http://localhost:9990/restapi/Channel1/continuous/b536b347-
ad9a-4ff7-9fb7-b097cd4b8230" rel="self" type="application/vizrt.integrationshub+xml;
type=group" />
    <begin_at>2017-08-28T11:28:09.120Z</begin_at>
    <duration>300</duration>
    <event_id>1002</event_id>
    <trigger_id>2</trigger_id>
    <take_count />
  </group>
  <group>
    <description>The Incredible BURT WONDERSTONE</description>
    <name>d84ac0fc-bcbe-4a8a-a134-6ff0384ac8e0</name>
    <link href="http://localhost:9990/restapi/Channel1/continuous/d84ac0fc-bcbe-4a8a-
a134-6ff0384ac8e0" rel="self" type="application/vizrt.integrationshub+xml;
type=group" />
    <begin_at>2017-08-28T11:33:09.120Z</begin_at>
    <duration>3525.04</duration>
    <event_id>1003</event_id>
    <trigger_id>3</trigger_id>
    <take_count />
  </group>
  <group>
    <description>Man of Steel - Official Trailer 3</description>
    <name>63e76944-f62e-4c24-b425-15fd603942b0</name>
    <link href="http://localhost:9990/restapi/Channel1/continuous/63e76944-f62e-4c24-
b425-15fd603942b0" rel="self" type="application/vizrt.integrationshub+xml;
type=group" />
    <begin_at>2017-08-28T12:31:54.160Z</begin_at>
    <duration>181.12</duration>
    <event_id>1004</event_id>
    <trigger_id>4</trigger_id>
    <take_count />
  </group>
  <ischannelpool>false</ischannelpool>
</playlist>
```

9.3.10 POST Playlist to a Channel (Version 1.0)

POST Playlist to a Channel

Playlists can only be added via the Viz Channel Branding Client.

Syntax

```
POST /restapi/[channel_name]
```

Request

```
POST /restapi/Channel1 HTTP/1.1
```

Response

Returns a **403 Forbidden** status message.

Copyright © 2024 Vizrt

9.3.11 DELETE Playlist (Version 1.0)

DELETE Playlist

Playlists can only be deleted via the Viz Channel Branding Client.

Syntax

```
DELETE /restapi/[channel_name]/[playlist_name]
```

Request

```
DELETE /restapi/Channel1/continuous HTTP/1.1
```

Response

Returns a **403 Forbidden** status message.

Copyright © 2024 Vizrt

9.3.12 GET Channel Pool from a Channel (Version 1.0)

GET channel pool playlist from a channel

For retrieving channel pool data.

Syntax for GET

```
GET /restapi/[channel_name]/channel_pool
```

Example Request

```
GET /restapi/Channel1/channel_pool HTTP/1.1
```

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshub+xml; type=playlist
<playlist xmlns="http://www.vizrt.com/integrationshub">
  <title>Playlist info for http://localhost:9990/restapi/Channel1/channel_pool</
title>
  <description>Channel Pool</description>
  <name>channel_pool</name>
  <ischannelpool>true</ischannelpool>
  <link href="http://localhost:9990/restapi/Channel1/channel_pool" rel="self" type="a
pplication/vizrt.integrationshub+xml; type=playlist" />
  <page>
    <description>15:00/16:00/ /1/Jerry the dog/KZA/ /C:/vizrt/mcp-clips/
Francois0000.avi/</description>
    <name>69aad93a-d2dc-41fc-928b-6ce3cfe6d2f3</name>
    <link href="http://localhost:9990/restapi/Channel1/channel_pool/69aad93a-
d2dc-41fc-928b-6ce3cfe6d2f3" rel="self" type="application/vizrt.integrationshub+xml;
type=page" />
    <timecode />
    <take_duration />
    <page>556</page>
  </page>
  <group>
    <description>Test</description>
    <name>82b793fe-3f1a-4807-a110-3cb2610e8c12</name>
    <link href="http://localhost:9990/restapi/Channel1/channel_pool/
82b793fe-3f1a-4807-a110-3cb2610e8c12" rel="self" type="application/
vizrt.integrationshub+xml; type=group" />
```

Copyright © 2024 Vizrt

```
    <begin_at />
    <duration>600</duration>
    <event_id />
    <trigger_id />
  </group>
</playlist>
```

9.3.13 POST Group and Page to Channel Pool (Version 1.0)

POST Group and Page to Channel Pool

Syntax for POST

```
POST /restapi/[channel_name]/channel_pool
```

Example Request

```
POST /restapi/MovieChannel/channel_pool/ HTTP/1.1
Content-Type: application/vizrt.integrationshub+xml; type=payload
<payload>
  <page name="page1" take_duration="50" page="basic" />
  <group name="group1" event_id="1" description="group 1" duration="3600" begin_at="2
014-06-13T01:00:00.000Z" />
</payload>
```

Example Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshub+xml; type=playlistpostreply
Content-Location: http://localhost:9990/restapi/MovieChannel/channel_pool
<playlistpostreply xmlns="http://www.vizrt.com/integrationshub">
    <title>Post result info to http://localhost:9990/restapi/Test/channel_pool</
title>
    <description>Channel Pool</description>
    <name>channel_pool</name>
    <ischannelpool>true</ischannelpool>
    <link href="http://localhost:9990/restapi/Test/channel_pool" rel="self" type="app
lication/vizrt.integrationshub+xml; type=playlistpostreply" />
    <group>
       <title>Group info for http://localhost:9990/restapi/Test/channel_pool/group1</
title>
       <description>group 1</description>
       <name>group1</name>
       <link href="http://localhost:9990/restapi/Test/channel_pool/group1" rel="self"
 type="application/vizrt.integrationshub+xml; type=group" />
       <begin_at>2014-06-13T01:00:00.000Z</begin_at>
       <duration>3600</duration>
       <event_id>12</event_id>
       <trigger_id />
       <attributes>
```

```
          <attribute name="adjust_begin_at_after_resume">
            <value>yes</value>
          </attribute>
          <attribute name="begin_at">
            <value>2014-06-13T01:00:00.000Z</value>
          </attribute>
          <attribute name="description">
            <value>group 1</value>
          </attribute>
          <attribute name="duration">
            <value>3600</value>
          </attribute>
          <attribute name="event_created_by">
            <value>schedule</value>
          </attribute>
          <attribute name="infoIntegrationsHub">
            <value>[A:16:22:14]</value>
          </attribute>
          <attribute name="instance">
            <value>composite_Test</value>
          </attribute>
          <attribute name="name">
            <value>group1</value>
          </attribute>
          <attribute name="event_id">
            <value>12</value>
          </attribute>
        </attributes>
    </group>
    <page>
      <title>Page info for http://localhost:9990/restapi/MovieChannel/channel_pool/
page1</title>
      <description>My Page</description>
      <name>page1</name>
      <link href="http://localhost:9990/restapi/MovieChannel/channel_pool/page1" rel="s
elf" type="application/vizrt.integrationshub+xml; type=page" />
      <timecode />
      <take_duration>50</take_duration>
      <page>basic</page>
      <attributes>
        <attribute name="description">
          <value>My Page</value>
        </attribute>
        <attribute name="event_created_by">
          <value>schedule</value>
        </attribute>
        <attribute name="page">
          <value>basic</value>
        </attribute>
        <attribute name="showautodescription">
          <value>false</value>
        </attribute>
```

Copyright © 2024 Vizrt

```
      <attribute name="take_duration">
        <value>50</value>
      </attribute>
      <attribute name="name">
        <value>page1</value>
      </attribute>
    </attributes>
    <fields />
  </page>
</playlistpostreply>
```

9.3.14 GET Group from a Playlist (Version 1.0)

GET Group from a Playlist

To obtain a group and associated pages of a group in a playlist for a given channel use the following command.

Syntax

```
GET /restapi/[channel_name]/[playlist_name]/[group_name]
```

Request

```
GET /restapi/Channel1/continuous/1958a5d9-87bb-42f0-a208-5ea7968c7649 HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshub+xml; type=group
<group xmlns="http://www.vizrt.com/integrationshub">
  <title>Group info for http://localhost:9990/restapi/Channel1/continuous/
1958a5d9-87bb-42f0-a208-5ea7968c7649</title>
  <description>Schedule1</description>
  <name>1958a5d9-87bb-42f0-a208-5ea7968c7649</name>
  <link href="http://localhost:9990/restapi/Channel1/continuous/1958a5d9-87bb-42f0-
a208-5ea7968c7649" rel="self" type="application/vizrt.integrationshub+xml;
type=group" />
  <begin_at>2017-08-28T10:00:00.000Z</begin_at>
  <duration>5289.12</duration>
  <event_id />
  <trigger_id />
  <take_count />
  <attributes>
    <attribute name="adjust_begin_at_after_resume">
      <value>yes</value>
    </attribute>
    <attribute name="begin_at">
      <value>2017-08-28T10:00:00.000Z</value>
    </attribute>
    <attribute name="description">
      <value>Schedule1</value>
    </attribute>
    <attribute name="duration">
```

Copyright © 2024 Vizrt

```
      <value>5289.12</value>
    </attribute>
    <attribute name="house_id">
      <value>QZJBJEQXMVERLAET</value>
    </attribute>
    <attribute name="instance">
      <value>composite_Channel1</value>
    </attribute>
    <attribute name="presentation">
      <value>composite</value>
    </attribute>
    <attribute name="name">
      <value>1958a5d9-87bb-42f0-a208-5ea7968c7649</value>
    </attribute>
  </attributes>
  <page>
    <description>Good words need great graphics/A bad graphic is worse than no
graphic/Size DOES matters! Superb HD graphics from Vizrt/Your analogue station  just
entered  a new era - history/No Viz. No Fizz/It...s TV. You must VIZualize it/Mood
music for the eyes. Vizrt/No story is complete without a Viz graphic</description>
    <name>page1</name>
    <link href="http://localhost:9990/restapi/Channel1/continuous/1958a5d9-87bb-42f0-
a208-5ea7968c7649/page1" rel="self" type="application/vizrt.integrationshub+xml;
type=page" />
    <timecode>00:00:10:00</timecode>
    <take_duration>30</take_duration>
    <page>BUG</page>
  </page>
  <page>
    <description>Good words need great graphics/A bad graphic is worse than no
graphic/Size DOES matters! Superb HD graphics from Vizrt/Your analogue station  just
entered  a new era - history/No Viz. No Fizz/It...s TV. You must VIZualize it/Mood
music for the eyes. Vizrt/No story is complete without a Viz graphic</description>
    <name>page2</name>
    <link href="http://localhost:9990/restapi/Channel1/continuous/1958a5d9-87bb-42f0-
a208-5ea7968c7649/page2" rel="self" type="application/vizrt.integrationshub+xml;
type=page" />
    <timecode>00:00:40:00</timecode>
    <take_duration>30</take_duration>
    <page>LOWER3RD</page>
  </page>
</group>
```

Copyright © 2024 Vizrt

9.3.15 POST Group to Playlist (Version 1.0)

POST Group to Playlist

To add or update a single or collection of groups in a playlist for a given channel use the POST command documented below.

For this command two examples are given. The first example (Request 1 - Example with 2 groups/ Response 1 - Example with 2 groups) shows how to post two groups. Note that for this example page(s) is added by rules for both groups.

The second example (Request 2 - Example with 1 group and 2 pages/Response 2 - Example with 1 group and 2 pages) shows how to add one group and two pages. No additional pages are added by rules.

Groups can be added with or without pages depending on the requirements.

> **Note:** Posting of multiple resources are allowed. The Location in the reponse points to the nearest parent (i.e. the entry the POST was made to). Each specific element is linked in the verbose response.

> **Note:** After this request is done the system is primed to perform a Page Content Filling process against Promo Server as soon as there is no immediate additional requests sent.

> **Note:** Including a timezone annotation in begin_at property is recommended, preferably in UTC (Z) time. If timezone annotation is missing the time is assumed to be in an unspecified component-centric local time.

Syntax

```
POST /restapi/[channel_name]/[playlist_name]
```

Request 1 – Example with 2 groups

```
POST /restapi/MovieChannel/continuous/ HTTP/1.1
Content-Type: application/vizrt.integrationshub+xml; type=payload
<payload>
  <group name="group1" event_id="1" description="group 1" duration="3600" begin_at="2
014-06-13T01:00:00.000Z" />
  <group name="group2" event_id="2" description="group 2" duration="3600" begin_at="2
014-06-13T00:00:00.000Z" />
</payload>
```

Copyright © 2024 Vizrt

Response 1 - Example with 2 groups

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshub+xml; type=playlistpostreply
Content-Location: http://localhost:9990/restapi/MovieChannel/continuous/
<playlistpostreply xmlns="http://www.vizrt.com/integrationshub">
  <title>Post result info to http://localhost:9990/restapi/MovieChannel/continuous/</
title>
  <description>Continuous Playlist</description>
  <name>continuous</name>
  <link href="http://localhost:9990/restapi/MovieChannel/continuous/" rel="self"
 type="application/vizrt.integrationshub+xml; type=playlistpostreply" />
  <broadcast_date>continuous</broadcast_date>
  <group>
    <title>Group info for http://localhost:9990/restapi/MovieChannel/continuous/
group1</title>
    <description>group 1</description>
    <name>group1</name>
    <link href="http://localhost:9990/restapi/MovieChannel/continuous/group1" rel="se
lf" type="application/vizrt.integrationshub+xml; type=group" />
    <begin_at>2014-06-13T01:00:00.000Z</begin_at>
    <duration>3600</duration>
    <event_id>1</event_id>
    <trigger_id />
    <attributes>
      <attribute name="begin_at">
        <value>2014-06-13T01:00:00.000</value>
      </attribute>
      <attribute name="description">
        <value>group 1</value>
      </attribute>
      <attribute name="duration">
        <value>3600</value>
      </attribute>
      <attribute name="event_id">
        <value>1</value>
      </attribute>
      <attribute name="instance">
        <value>relative_MovieChannel</value>
      </attribute>
      <attribute name="name">
        <value>group1</value>
      </attribute>
    </attributes>
    <page>
      <description>00:27/00:30/00:33/00:36/00:39/Africa 2/sso4/SSO3/SSO2/confomr a
conform SD/FOX 2007 EMMY WINNERS/FOX 2007 EMMY WINNERS</description>
      <name>nextshow</name>
      <link href="http://localhost:9990/restapi/MovieChannel/continuous/group1/
nextshow" rel="self" type="application/vizrt.integrationshub+xml; type=page" />
```

```
      <timecode>00:00:05:00</timecode>
      <take_duration />
      <page>nextshow</page>
    </page>
  </group>
  <group>
    <title>Group info for http://localhost:9990/restapi/MovieChannel/continuous/
group2</title>
    <description>group 2</description>
    <name>group2</name>
    <link href="http://localhost:9990/restapi/MovieChannel/continuous/group2" rel="se
lf" type="application/vizrt.integrationshub+xml; type=group" />
    <begin_at>2014-06-13T00:00:00.000Z</begin_at>
    <duration>3600</duration>
    <event_id>2</event_id>
    <trigger_id />
    <attributes>
      <attribute name="begin_at">
        <value>2014-06-13T00:00:00.000</value>
      </attribute>
      <attribute name="description">
        <value>group 2</value>
      </attribute>
      <attribute name="duration">
        <value>3600</value>
      </attribute>
      <attribute name="event_id">
        <value>2</value>
      </attribute>
      <attribute name="instance">
        <value>relative_MovieChannel</value>
      </attribute>
      <attribute name="name">
        <value>group2</value>
      </attribute>
    </attributes>
    <page>
      <description>00:27/00:30/00:33/00:36/00:39/Africa 2/sso4/SSO3/SSO2/confomr a
conform SD/FOX 2007 EMMY WINNERS/FOX 2007 EMMY WINNERS</description>
      <name>nextshow</name>
      <link href="http://localhost:9990/restapi/MovieChannel/continuous/group2/
nextshow" rel="self" type="application/vizrt.integrationshub+xml; type=page" />
      <timecode>00:00:05:00</timecode>
      <take_duration />
      <page>nextshow</page>
    </page>
  </group>
</playlistpostreply>
```

Copyright © 2024 Vizrt

Request 2 – Example with 1 group and 2 pages

```
POST /restapi/MovieChannel/continuous/ HTTP/1.1
Content-Type: application/vizrt.integrationshub+xml; type=payload
<payload>
  <group name="group1" event_id="1" description="group 1" duration="3600" begin_at="2
014-06-13T01:00:00.000Z">
    <page name="page1" page="now" take_duration="30" timecode="00:00:10:00" />
    <page name="page2" page="nextshow" take_duration="14" timecode="00:59:30:00" />
  </group>
</payload>
```

Response 2 – Example with 1 group and 2 pages

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshub+xml; type=playlistpostreply
Content-Location: http://localhost:9990/restapi/MovieChannel/continuous/
<playlistpostreply xmlns="http://www.vizrt.com/integrationshub">
  <title>Post result info to http://localhost:9990/restapi/MovieChannel/continuous/</
title>
  <description>Continuous Playlist</description>
  <name>continuous</name>
  <link href="http://localhost:9990/restapi/MovieChannel/continuous/" rel="self"
 type="application/vizrt.integrationshub+xml; type=playlistpostreply" />
  <broadcast_date>continuous</broadcast_date>
  <group>
    <title>Group info for http://localhost:9990/restapi/MovieChannel/continuous/
group1</title>
    <description>group 1</description>
    <name>group1</name>
    <link href="http://localhost:9990/restapi/MovieChannel/continuous/group1" rel="se
lf" type="application/vizrt.integrationshub+xml; type=group" />
    <begin_at>2014-06-13T01:00:00.000Z</begin_at>
    <duration>3600</duration>
    <event_id>1</event_id>
    <trigger_id />
    <attributes>
      <attribute name="begin_at">
        <value>2014-06-13T01:00:00.000</value>
      </attribute>
      <attribute name="description">
        <value>group 1</value>
      </attribute>
      <attribute name="duration">
        <value>3600</value>
      </attribute>
      <attribute name="event_id">
```

Copyright © 2024 Vizrt

```
        <value>1</value>
      </attribute>
      <attribute name="instance">
        <value>relative_MovieChannel</value>
      </attribute>
      <attribute name="name">
        <value>group1</value>
      </attribute>
    </attributes>
    <page>
      <description>Good words need great graphics/A bad graphic is worse than no
graphic/Size DOES matters! Superb HD graphics from Vizrt/Your analogue station  just
entered  a new era - history/No Viz. No Fizz/It...s TV. You must VIZualize it/Mood
music for the eyes. Vizrt/No story is complete without a Viz graphic</description>
      <name>page1</name>
      <link href="http://localhost:9990/restapi/MovieChannel/continuous/group1/page1"
 rel="self" type="application/vizrt.integrationshub+xml; type=page" />
      <timecode>00:00:10:00</timecode>
      <take_duration>30</take_duration>
      <page>now</page>
    </page>
    <page>
      <description>00:27/00:30/00:33/00:36/00:39/Africa 2/sso4/SSO3/SSO2/confomr a
conform SD/FOX 2007 EMMY WINNERS/FOX 2007 EMMY WINNERS</description>
      <name>page2</name>
      <link href="http://localhost:9990/restapi/MovieChannel/continuous/group1/page2"
 rel="self" type="application/vizrt.integrationshub+xml; type=page" />
      <timecode>00:59:30:00</timecode>
      <take_duration>14</take_duration>
      <page>nextshow</page>
    </page>
  </group>
</playlistpostreply>
```

Example of content for POST Group to Playlist with isLive and isLocked set to true

```
POST /restapi/MovieChannel/continuous/ HTTP/1.1
Content-Type: application/vizrt.integrationshub+xml; type=payload
<payload>
  <group name="group3" event_id="3" description="group 1" duration="3600" begin_at="2
014-06-13T01:00:00.000Z" isLive="true" isLocked="true" />
</payload>
```

Copyright © 2024 Vizrt

9.3.16 DELETE Group (Version 1.0)

DELETE Group

To remove a specific group and all pages a part of it use the following command.

> **Note:** After this request is done the system is primed to perform a Page Content Filling process against Promo Server as soon as there is no immediate additional requests sent.

Syntax

```
DELETE /restapi/[channel_name]/[playlist_name]/[group_name]
```

Request

```
DELETE /restapi/Channel1/continuous/group1 HTTP/1.1
```

Response

Returns a **204 No Content** status message as no content is found because the group was found and its content **successfully deleted**.

Returns a **404 Not Found** status message because the group was not found.

Copyright © 2024 Vizrt

9.3.17 GET Page from a Group (Version 1.0)

GET Pages from a Group

To obtain a page for a specific group in a playlist for a given channel use the following command.

Syntax

```
GET /restapi/[channel_name]/[playlist_name]/[group_name]/[page_name]
```

Request

```
GET /restapi/Channel1/continuous/group1/page2 HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshub+xml; type=page
<page xmlns="http://www.vizrt.com/integrationshub">
  <title>Page info for http://[true_ip]:9990/restapi/[channelname]/[playlist_name]/
[group_name]/[page_name]</title>
  <description>[page_description]</description>
  <name>[page_name]</name>
  <timecode>[page_timecode]</timecode>
  <take_duration>[page_take_duration]</take_duration>
  <page>[page_page]</page>
  <scene>[scene info from sceneinfo]</scene>
  <attributes>
    <attribute name="[attribute_name]">
      <value>[value]</value>
    </attribute>
    <attribute name="[attribute_name]">
     <value>[value]</value>
    </attribute>
    [..]
  </attributes>
  <fields>
    <field name="[tabfield_name]">
      <mapping>[tabfield_mapping]</mapping>
      <value>[tabfield_value]</value>
      <type>[tabfield_type]</type>
    </field>
    [..]
```

```
  </fields>
  <link href="http://[true_ip]:9990/restapi/[channelname]/[playlist_name]/
[group_name]/[page_name]" rel="self" type="application/vizrt.integrationshub+xml;
type=page" />
</page>
```

9.3.18 POST Pages to a Group (Version 1.0)

POST Pages to a Group

To add or update a collection of pages for a given group, in a playlist for a given channel use the following command.

The Request (example)/Response example given shows how to post two pages to a group. No additional pages are added by rules.

> **Note:** Posting off multiple resources are allowed. The Location in the response points to the nearest parent (i.e. the entry the POST was made to). Each specific element is linked in the verbose response.

> **Note:** After this request is done the system is primed to perform a Page Content Filling process against Promo Server as soon as there is no immediate additional requests sent.

Syntax

```
POST /restapi/[channel_name]/[playlist_name]/[group_name]
```

Request (example)

```
POST /restapi/MovieChannel/continuous/group1 HTTP/1.1
Content-Type: application/vizrt.integrationshub+xml; type=payload
<payload>
  <page name="page1" page="now" take_duration="30" timecode="00:00:10:00" />
  <page name="page2" page="nextshow" take_duration="14"  timecode="00:59:30:00"/>
</payload>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshub+xml; type=grouppostreply
Content-Location: http://localhost:9990/restapi/MovieChannel/continuous/group1
<grouppostreply xmlns="http://www.vizrt.com/integrationshub">
  <page>
    <title>Page info for http://localhost:9990/restapi/MovieChannel/continuous/
group1/page1</title>
    <description>Good words need great graphics/A bad graphic is worse than no
graphic/Size DOES matters! Superb HD graphics from Vizrt/Your analogue station  just
```

```
entered  a new era - history/No Viz. No Fizz/It...s TV. You must VIZualize it/Mood
music for the eyes. Vizrt/No story is complete without a Viz graphic</description>
    <name>page1</name>
    <link href="http://localhost:9990/restapi/MovieChannel/continuous/group1/page1"
 rel="self" type="application/vizrt.integrationshub+xml; type=page" />
    <timecode>00:00:10:00</timecode>
    <take_duration>30</take_duration>
    <page>now</page>
    <attributes>
      <attribute name="description">
        <value>Good words need great graphics/A bad graphic is worse than no graphic/
Size DOES matters! Superb HD graphics from Vizrt/Your analogue station  just entered
a new era - history/No Viz. No Fizz/It...s TV. You must VIZualize it/Mood music for
 the eyes. Vizrt/No story is complete without a Viz graphic</value>
      </attribute>
      <attribute name="event_created_by">
        <value>schedule</value>
      </attribute>
      <attribute name="page">
        <value>now</value>
      </attribute>
      <attribute name="pageContentValue">
        <value>The Smashing Pumpkins | Bullet With Butterfly Wings</value>
       </attribute>
      <attribute name="pagepool_page">
        <value>6000</value>
      </attribute>
      <attribute name="take_duration">
        <value>30</value>
      </attribute>
      <attribute name="showautodescription">
        <value>false</value>
      </attribute>
      <attribute name="timecode">
       <value>00:00:10:00</value>
      </attribute>
      <attribute name="name">
       <value>page1</value>
      </attribute>
    </attributes>
    <fields>
      <field name="1">
        <value>Good words need great graphics</value>
        <mapping />
      </field>
      <field name="2">
        <value>A bad graphic is worse than no graphic</value>
        <mapping />
      </field>
      <field name="3">
        <value>Size DOES matters! Superb HD graphics from Vizrt</value>
        <mapping />
```

Copyright © 2024 Vizrt

```
      </field>
      <field name="4">
        <value>Your analogue station  just entered  a new era - history</value>
        <mapping />
      </field>
      <field name="5">
        <value>No Viz. No Fizz</value>
        <mapping />
      </field>
      <field name="6">
        <value>It...s TV. You must VIZualize it</value>
        <mapping />
      </field>
      <field name="7">
        <value>Mood music for the eyes. Vizrt</value>
        <mapping />
      </field>
      <field name="8">
        <value>No story is complete without a Viz graphic</value>
        <mapping />
      </field>
    </fields>
  </page>
[... MORE ... deleted to save space ...]
</fields>
  </page>
</grouppostreply>
```

Copyright © 2024 Vizrt

9.3.19 DELETE Page of a Group (Version 1.0)

DELETE Page of a Group

To remove a specific page of a given group use the following command.

Syntax

```
DELETE /restapi/[channel_name]/[playlist_name]/[group_name]/[page_name]
```

Request

```
DELETE /restapi/Channel1/continuous/group1/page1 HTTP/1.1
```

Response

Returns a **204 No Content *status message as no content is found because the page was found and *successfully deleted**.

Returns a **404 Not Found** status message because the page was not found.

Copyright © 2024 Vizrt

9.3.20 GET All Events with Matching Event Id (Version 1.0)

GET All Events with Matching Event Id

To retrieve a selected list of one or more items with a specific event-id associated with it, call this end-point and specify event-id to filter by:

Syntax

```
GET /GetEventsByEventId/[channel_name]/[playlist_name]/[event_id]
```

Request

```
GET /GetEventsByEventId/test/continuous/070bf70c-0dbd-4fa6-a609-71bd34d6361c HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshub+xml; type=geteventsbyeventid
<ResponseGetEventsByEventId xmlns="http://www.vizrt.com/integrationshub">
   <title>GetEventsByEventId result from channel: continuous and Event Id filter:
070bf70c-0dbd-4fa6-a609-71bd34d6361c</title>
   <description>GetEventsByEventId result from channel: continuous and Event Id
filter: 070bf70c-0dbd-4fa6-a609-71bd34d6361c with Uri called at: http://
127.0.0.1:9990/GetEventsByEventId/test/continuous/070bf70c-0dbd-4fa6-
a609-71bd34d6361c</description>
   <name>GetEventsByEventId result from channel: continuous and Event Id filter:
070bf70c-0dbd-4fa6-a609-71bd34d6361c</name>
   <link href="http://127.0.0.1:9990/GetEventsByEventId/test/continuous/
070bf70c-0dbd-4fa6-a609-71bd34d6361c" rel="self" type="application/
vizrt.integrationshub+xml; type=ResponseGetEventsByEventId"/>
   <primaryeventcount>0</primaryeventcount>
   <secondaryeventcount>1</secondaryeventcount>
   <page>
      <description>TEST 4 - The Smashing Pumpkins/Bullet With Butterfly Wings</
description>
      <name>070bf70c-0dbd-4fa6-a609-71bd34d6361c</name>
      <link href="http://127.0.0.1:9990/restapi/test/continuous/
21479131448478/070bf70c-0dbd-4fa6-a609-71bd34d6361c" rel="self" type="application/
vizrt.integrationshub+xml; type=page"/>
      <timecode>60</timecode>
      <take_duration>0</take_duration>
      <page>lowerthird</page>
```

Copyright © 2024 Vizrt

```
      <take_count>1</take_count>
   </page>
</ResponseGetEventsByEventId>
```

9.4 REST Interface For Integrations Hub Version 2.0

Version 2 of the Integrations Hub REST APIs offer full data representation.

For the legacy version (1.0) of this API, please refer to section REST Interface for Integrations Hub Version 1.0.

- GET API Root (Version 2.0)
- GET ALL Channels (Version 2.0)
- GET a Channel (Version 2.0)
- Channel Content (Version 2.0)

Note: If an integration using the version 2.0 of the Integrations Hub REST API is intended, please contact Vizrt for further support.

Copyright © 2024 Vizrt

9.4.1 GET API Root (Version 2.0)

The starting point for the structure of the REST API is found by calling the address and IP for the service. The port for Integrations Hub v2 REST API is by default 9991.

Example request using curl against the service on a local machine would be: curl -X GET http://127.0.0.1:9991/

The rest of the guide assumes each request is prefixed with the localhost/IP and port as needed and only lists the end-points addresses.

In general, if the Media Sequencer service is down, this will result in a HTTP/1.1 503 ServiceUnavailable response.

Syntax

```
GET /
```

Request

```
GET / HTTP/1.1
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=root
3  <inthub:root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:inthub="http://www.vizrt.com/integrationshub">
4     <inthub:refLink rel="channels" type="application/
   vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
   channels" />
5  </inthub:root>
```

Disallowed Actions

This end-point does not allow POST, PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

9.4.2 GET ALL Channels (Version 2.0)

Get the full collection of channels represented via this instance of the Integrations Hub, as referred by the API Root reply

Syntax

```
GET /api/channels
```

Request

```
GET /api/channels HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=list
<inthub:list xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="inthub:channelsList" xmlns:inthub="http://www.vizrt.com/
integrationshub">
    <inthub:channel>
      <inthub:name>test</inthub:name>
      <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=channel" href="http://127.0.0.1:9991/
api/channels/test" />
      <inthub:refLink rel="playlists" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/playlists" />
      <inthub:refLink rel="templates" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/templates" />
      <inthub:refLink rel="channel_pool" type="application/
vizrt.integrationshubv2+xml; type=channelpool" href="http://
127.0.0.1:9991/api/channels/test/channel_pool" />
      <inthub:refLink rel="page_pool" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/page_pool" />
      <inthub:refLink rel="action_pool" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/action_pool" />
      <inthub:refLink rel="settings" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/settings" />
    </inthub:channel>
    <inthub:channel>
```

Copyright © 2024 Vizrt

```
15        <inthub:name>otherchannel</inthub:name>
16        <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=channel" href="http://127.0.0.1:9991/
api/channels/otherchannel" />
17        <inthub:refLink rel="playlists" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/otherchannel/playlists" />
18        <inthub:refLink rel="templates" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/otherchannel/templates" />
19        <inthub:refLink rel="channel_pool" type="application/
vizrt.integrationshubv2+xml; type=channelpool" href="http://
127.0.0.1:9991/api/channels/otherchannel/channel_pool" />
20        <inthub:refLink rel="page_pool" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/otherchannel/page_pool" />
21        <inthub:refLink rel="action_pool" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/otherchannel/action_pool" />
22        <inthub:refLink rel="settings" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/otherchannel/settings" />
23    </inthub:channel>
24    </inthub:list>
```

Disallowed Actions

This end-point does not allow POST, PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

9.4.3 GET a Channel (Version 2.0)

Get the top endpoint for a single channel.

Syntax

```
GET /api/channels/[channel]
```

Request

```
GET /api/channels/test HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=channel
<inthub:channel xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>test</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=channel" href="http://127.0.0.1:9991/
api/channels/test" />
    <inthub:refLink rel="playlists" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/playlists" />
    <inthub:refLink rel="templates" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/templates" />
    <inthub:refLink rel="channel_pool" type="application/
vizrt.integrationshubv2+xml; type=channelpool" href="http://
127.0.0.1:9991/api/channels/test/channel_pool" />
    <inthub:refLink rel="page_pool" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/page_pool" />
    <inthub:refLink rel="action_pool" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/action_pool" />
    <inthub:refLink rel="settings" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/settings" />
</inthub:channel>
```

Copyright © 2024 Vizrt

Disallowed Actions

This end-point does not allow POST, PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

9.4.4 Channel Content (Version 2.0)

- GET Settings for Channel (Version 2.0)
- Playlists (Version 2.0)
- Templates (Version 2.0)
- Pages (Version 2.0)
- Actions (Version 2.0)
- Channel Pool (Version 2.0)

Copyright © 2024 Vizrt

GET Settings for Channel (Version 2.0)

Get the complete settings for a specific channel. Modifying settings can only be done through Channel Settings in the Viz Channel Branding Client.

Syntax

```
GET /api/channels/[channel]/settings
```

Request

```
GET /api/channels/test/settings HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml;
type=channelsettings
 <inthub:channelSettings xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=channelsettings" href="http://
127.0.0.1:9991/api/channels/test/settings" />
    <inthub:general>
        <inthub:setting name="GeneralSettings.BasicSettings.ChannelBasicIn
fo.ScreenDimension">1920*1080</inthub:setting>
        <inthub:setting name="GeneralSettings.BasicSettings.ChannelBasicIn
fo.AspectRatio">16/9</inthub:setting>
        <inthub:setting name="GeneralSettings.BasicSettings.ChannelBasicIn
fo.FramesPerSec">25</inthub:setting>
        <inthub:setting name="GeneralSettings.BasicSettings.ChannelBasicIn
fo.BroadcastDayStartTime">04:00:00</inthub:setting>
        <inthub:setting name="GeneralSettings.BasicSettings.TriggeringCont
rol.TimedPlayout">false</inthub:setting>
        <inthub:setting name="GeneralSettings.BasicSettings.TriggeringCont
rol.TriggerType">Primary Events</inthub:setting>
        <inthub:setting name="GeneralSettings.BasicSettings.PlaylistDateMa
nagement.AllEventsInOnePlaylist">true</inthub:setting>
        <inthub:setting name="GeneralSettings.TimezoneSettings.ScheduleTim
eZoneId">W. Europe Standard Time</inthub:setting>
        <inthub:setting name="GeneralSettings.TimezoneSettings.ChannelTime
ZoneId">W. Europe Standard Time</inthub:setting>
        <inthub:setting name="GeneralSettings.DaySwitchOptions.DaySwitchOp
tions.ActivateDaySwitch">false</inthub:setting>
```

Copyright © 2024 Vizrt

```
16          <inthub:setting name="GeneralSettings.DaySwitchOptions.DaySwitchOptions.ActivatePlaylist">00:00:00</inthub:setting>
17          <inthub:setting name="GeneralSettings.DaySwitchOptions.DaySwitchOptions.InsertCleanUp">false</inthub:setting>
18          <inthub:setting name="GeneralSettings.DaySwitchOptions.DaySwitchOptions.TimeBeforeCleanUp">00:00:00</inthub:setting>
19          <inthub:setting name="GeneralSettings.DaySwitchOptions.DaySwitchOptions.InsertInitialize">false</inthub:setting>
20          <inthub:setting name="GeneralSettings.DaySwitchOptions.DaySwitchOptions.TimeBeforeInitialize">00:00:00</inthub:setting>
21          <inthub:setting name="GeneralSettings.AttributeMappings.AttributeMappings.PredefinedDataMappings"></inthub:setting>
22          <inthub:setting name="GeneralSettings.AttributeMappings.AttributeMappings.MetaDataMappings"></inthub:setting>
23          <inthub:setting name="GeneralSettings.PageContentFilling.PageContentFilling.RefreshPageContentOfPlaylist">true</inthub:setting>
24          <inthub:setting name="GeneralSettings.PageContentFilling.PageContentFilling.RefreshPageContentOfRunningPlaylist">false</inthub:setting>
25          <inthub:setting name="GeneralSettings.PageContentFilling.PageContentFilling.SkipRefreshAlreadyPlayedPage">true</inthub:setting>
26          <inthub:setting name="GeneralSettings.PageContentFilling.PageContentFilling.AllowPageToForceMappingForPageContentFilling">false</inthub:setting>
27          <inthub:setting name="GeneralSettings.PageContentFilling.PageContentFilling.PromoServerAddress"></inthub:setting>
28          <inthub:setting name="GeneralSettings.PageContentFilling.PageContentFilling.MaximumNumberOfPagesToRefreshInPlaylist"></inthub:setting>
29          <inthub:setting name="GeneralSettings.PageContentFilling.PageContentFilling.TranslationTables"></inthub:setting>
30          <inthub:setting name="GeneralSettings.PageContentFilling.PageContentFilling.TonightStartTime">21:00:00</inthub:setting>
31          <inthub:setting name="GeneralSettings.Miscellaneous.GeneralPlayoutOptions.AllowItemReplay">true</inthub:setting>
32          <inthub:setting name="GeneralSettings.Miscellaneous.PlaylistOptions.IsAutoExpandRunningElement">true</inthub:setting>
33          <inthub:setting name="GeneralSettings.Miscellaneous.PlaylistOptions.LockViewToRunningPrimaryEvent">false</inthub:setting>
34          <inthub:setting name="GeneralSettings.Miscellaneous.PlaylistOptions.DefaultPrimaryEventDuration">00:10:00:00</inthub:setting>
35          <inthub:setting name="GeneralSettings.Miscellaneous.PlaylistOptions.DefaultLiveEventDuration">01:00:00:00</inthub:setting>
36          <inthub:setting name="GeneralSettings.Miscellaneous.VizOneIntegration.AutoExtractVizOneMetaData">false</inthub:setting>
37          <inthub:setting name="GeneralSettings.Miscellaneous.VizOneIntegration.CreatePlaceholderInVizOne">false</inthub:setting>
38          <inthub:setting name="GeneralSettings.Miscellaneous.DefaultScenes.AutomaticallyTakeLogoPage">false</inthub:setting>
39          <inthub:setting name="GeneralSettings.Miscellaneous.DefaultScenes.LogoScene"></inthub:setting>
40          <inthub:setting name="GeneralSettings.Miscellaneous.DefaultScenes.ErrorScene"></inthub:setting>
41          <inthub:setting name="GeneralSettings.Miscellaneous.DefaultScenes.VideoOnlyScene"></inthub:setting>
```

Copyright © 2024 Vizrt

```
42          <inthub:setting name="GeneralSettings.Miscellaneous.QuickTakePages
    "></inthub:setting>
43          <inthub:setting name="ScheduleImportSettings.GeneralSettings.Chann
    elName.ChannelName"></inthub:setting>
44          <inthub:setting name="ScheduleImportSettings.GeneralSettings.Remot
    eHosts.RemoteHosts"></inthub:setting>
45          <inthub:setting name="ScheduleImportSettings.GeneralSettings.Autom
    aticImportSettings.EnableAutomaticImport">false</inthub:setting>
46          <inthub:setting name="ScheduleImportSettings.GeneralSettings.Autom
    aticImportSettings.ImportFolder"></inthub:setting>
47          <inthub:setting name="ScheduleImportSettings.GeneralSettings.Autom
    aticImportSettings.UpdateFolder"></inthub:setting>
48          <inthub:setting name="ScheduleImportSettings.GeneralSettings.Impor
    tOptions.ImportSecondaryAsRef">false</inthub:setting>
49          <inthub:setting name="ScheduleImportSettings.GeneralSettings.Impor
    tOptions.AllowUsingPilotDataServerElement">false</inthub:setting>
50          <inthub:setting name="ScheduleImportSettings.GeneralSettings.Impor
    tOptions.SkipSecondaryEventsExceedingPrimaryDuration">false</inthub:settin
    g>
51          <inthub:setting name="ScheduleImportSettings.GeneralSettings.Impor
    tOptions.TreatDurationTimeCodeAndTimeGivenAsIntegerAsFrameCount">false</
    inthub:setting>
52          <inthub:setting name="ScheduleImportSettings.MappingSettings"></
    inthub:setting>
53          <inthub:setting name="ScheduleImportSettings.FilteringOptions.Prim
    aryEventFiltering"></inthub:setting>
54          <inthub:setting name="ScheduleImportSettings.UpdateSettings.Update
    Options.AutomaticUpdateMode">Full</inthub:setting>
55          <inthub:setting name="ScheduleImportSettings.UpdateSettings.Update
    Options.DeleteProcessedFiles">2</inthub:setting>
56          <inthub:setting name="ScheduleImportSettings.UpdateSettings.Update
    Options.ClearPlayedEvents">00:00:00</inthub:setting>
57          <inthub:setting name="ScheduleImportSettings.UpdateSettings.Update
    Options.ClearUnplayedEvents">00:00:00</inthub:setting>
58          <inthub:setting name="ScheduleImportSettings.UpdateSettings.Update
    Options.AllowUpdatesOfRunningPlaylist">false</inthub:setting>
59          <inthub:setting name="ScheduleImportSettings.UpdateSettings.Update
    Options.LockEditedPageWhenUpdate">true</inthub:setting>
60          <inthub:setting name="ScheduleImportSettings.UpdateSettings.Update
    Options.RetrieveDateFromRunningPlaylist">false</inthub:setting>
61          <inthub:setting name="ScheduleImportSettings.UpdateSettings.Update
    Options.CreatePlaylistIfNotExists">false</inthub:setting>
62          <inthub:setting name="ExternalTriggering.GPI.IsEnabled">false</
    inthub:setting>
63          <inthub:setting name="ExternalTriggering.GPI.GPIDeviceDefinition.B
    itsPerSecond">19200</inthub:setting>
64          <inthub:setting name="ExternalTriggering.GPI.GPIDeviceDefinition.P
    ort">----</inthub:setting>
65          <inthub:setting name="ExternalTriggering.GPI.GPIDeviceDefinition.S
    topBits">1</inthub:setting>
66          <inthub:setting name="ExternalTriggering.GPI.GPIDeviceDefinition.D
    ataBits">8</inthub:setting>
```

Copyright © 2024 Vizrt

```
67          <inthub:setting name="ExternalTriggering.GPI.GPIDeviceDefinition.P
    arity">None</inthub:setting>
68          <inthub:setting name="ExternalTriggering.GPI.GPIDeviceDefinition.F
    lowControl">None</inthub:setting>
69          <inthub:setting name="ExternalTriggering.GPI.GPIDeviceDefinition.B
    oxType">SRC-8</inthub:setting>
70          <inthub:setting name="ExternalTriggering.GPI.GPIDeviceDefinition.I
    P"></inthub:setting>
71          <inthub:setting name="ExternalTriggering.GPI.GPIDeviceDefinition.U
    nit">0</inthub:setting>
72          <inthub:setting name="ExternalTriggering.GPI.GPIDeviceDefinition.S
    laveId">0</inthub:setting>
73          <inthub:setting name="ExternalTriggering.GPI.GPIActionsMapping"></
    inthub:setting>
74          <inthub:setting name="ExternalTriggering.VDCP.IsEnabled">false</
    inthub:setting>
75          <inthub:setting name="ExternalTriggering.VDCP.VDCP.IsForceExtended
    Mode">false</inthub:setting>
76          <inthub:setting name="ExternalTriggering.VDCP.VDCP.BitsPerSecond">
    38400</inthub:setting>
77          <inthub:setting name="ExternalTriggering.VDCP.VDCP.Port">----</
    inthub:setting>
78          <inthub:setting name="ExternalTriggering.VDCP.VDCP.ByteSize">8</
    inthub:setting>
79          <inthub:setting name="ExternalTriggering.VDCP.VDCP.ExtraStopBits">
    1</inthub:setting>
80          <inthub:setting name="ExternalTriggering.VDCP.VDCP.Parity">1</
    inthub:setting>
81          <inthub:setting name="ExternalTriggering.IntelligentInterface.IsEn
    abled">false</inthub:setting>
82          <inthub:setting name="ExternalTriggering.IntelligentInterface.Inte
    lligentInterface.BitsPerSecond">19200</inthub:setting>
83          <inthub:setting name="ExternalTriggering.IntelligentInterface.Inte
    lligentInterface.Port">----</inthub:setting>
84          <inthub:setting name="ExternalTriggering.IntelligentInterface.Inte
    lligentInterface.Encoding">UTF-8</inthub:setting>
85          <inthub:setting name="ExternalTriggering.IntelligentInterface.Inte
    lligentInterface.Space">false</inthub:setting>
86          <inthub:setting name="ExternalTriggering.TCP.IsEnabled">false</
    inthub:setting>
87          <inthub:setting name="ExternalTriggering.TCP.TCP.Port"></inthub:se
    tting>
88          <inthub:setting name="ExternalDataSources.DatabaseConnections.Data
    baseConnections.Links"></inthub:setting>
89          <inthub:setting name="ExternalDataSources.UpdateService.UpdateServ
    ice.HostName">http://localhost:56999</inthub:setting>
90          <inthub:setting name="ExternalDataSources.UpdateService.UpdateServ
    ice.Take">true</inthub:setting>
91          <inthub:setting name="ExternalDataSources.UpdateService.UpdateServ
    ice.TakeTimeout">5</inthub:setting>
92          <inthub:setting name="ExternalDataSources.UpdateService.UpdateServ
    ice.Preview">true</inthub:setting>
```

Copyright © 2024 Vizrt

```
93          <inthub:setting name="ExternalDataSources.UpdateService.UpdateServ
    ice.PreviewTimeout">5</inthub:setting>
94          <inthub:setting name="ExternalDataSources.UpdateService.UpdateServ
    ice.Update">true</inthub:setting>
95          <inthub:setting name="ExternalDataSources.UpdateService.UpdateServ
    ice.UpdateTimeout">5</inthub:setting>
96          <inthub:setting name="ExternalDataSources.DatabaseConnections.Data
    baseConnections.ConnectionDelay"></inthub:setting>
97          <inthub:setting name="QuickActions"></inthub:setting>
98      </inthub:general>
99      <inthub:mappings>
100         <inthub:mapping xsi:type="inthub:anytextseparator">
101             <inthub:fileNamePattern>*.txt</inthub:fileNamePattern>
102             <inthub:scheduleFileNameInfo />
103             <inthub:columnMappings>
104                 <inthub:columnMapping mappingReference="" fieldName="chann
    el" />
105                 <inthub:columnMapping mappingReference="$L(3)" fieldName="
    date" />
106                 <inthub:columnMapping mappingReference="$L(4)" fieldName="
    time" />
107                 <inthub:columnMapping mappingReference="@dateT@time"
    fieldName="begin_at" />
108                 <inthub:columnMapping mappingReference="$L(6)" fieldName="
    duration" />
109                 <inthub:columnMapping mappingReference="$L(5)" fieldName="
    description" />
110                 <inthub:columnMapping mappingReference="$L(1)" fieldName="
    event_id" />
111                 <inthub:columnMapping mappingReference="$L(2)" fieldName="
    event_type" />
112                 <inthub:columnMapping mappingReference="$L(3)" fieldName="
    secondary_page" />
113                 <inthub:columnMapping mappingReference="$L(4)" fieldName="
    secondary_timecode" />
114                 <inthub:columnMapping mappingReference="$L(6)" fieldName="
    secondary_duration" />
115                 <inthub:columnMapping mappingReference="$L(2)" fieldName="
    secondary_event_id" />
116                 <inthub:columnMapping mappingReference="$L(7)" fieldName="
    some_metadata" />
117                 <inthub:columnMapping mappingReference="$L(1)" fieldName="
    secondary_event_type" />
118             </inthub:columnMappings>
119             <inthub:primaryDateFormat>yyyy/mm/dd</inthub:primaryDateFormat
    >
120             <inthub:isBroadcastDate>true</inthub:isBroadcastDate>
121             <inthub:primaryEventType>PRIMARY</inthub:primaryEventType>
122             <inthub:secondaryEventType>SECONDARY</inthub:secondaryEventTyp
    e>
123             <inthub:isDefaultBehavior>False</inthub:isDefaultBehavior>
124             <inthub:ignoreValues>C,B,P</inthub:ignoreValues>
125             <inthub:skipLines />
```

Copyright © 2024 Vizrt

```
126             <inthub:columnSeparator>|</inthub:columnSeparator>
127             <inthub:encodingType>UTF-8</inthub:encodingType>
128         </inthub:mapping>
129         <inthub:mapping xsi:type="inthub:anytextindex">
130             <inthub:fileNamePattern>*.txt</inthub:fileNamePattern>
131             <inthub:scheduleFileNameInfo />
132             <inthub:columnMappings>
133                 <inthub:columnMapping mappingReference="" fieldName="chann
    el" />
134                 <inthub:columnMapping mappingReference="$L(19,28)"
    fieldName="date" />
135                 <inthub:columnMapping mappingReference="$L(29,39)"
    fieldName="time" />
136                 <inthub:columnMapping mappingReference="@dateT@time"
    fieldName="begin_at" />
137                 <inthub:columnMapping mappingReference="$L(79,89)"
    fieldName="duration" />
138                 <inthub:columnMapping mappingReference="$L(40,78)"
    fieldName="description" />
139                 <inthub:columnMapping mappingReference="$L(10,18)"
    fieldName="event_id" />
140                 <inthub:columnMapping mappingReference="$L(1,7)" fieldName="
    event_type" />
141                 <inthub:columnMapping mappingReference="$L(23,27)"
    fieldName="secondary_page" />
142                 <inthub:columnMapping mappingReference="$L(28,38)"
    fieldName="secondary_timecode" />
143                 <inthub:columnMapping mappingReference="$L(47,57)"
    fieldName="secondary_duration" />
144                 <inthub:columnMapping mappingReference="$L(1,13)"
    fieldName="secondary_event_id" />
145                 <inthub:columnMapping mappingReference="$L(90,119)"
    fieldName="some_metadata" />
146                 <inthub:columnMapping mappingReference="$L(14,22)"
    fieldName="secondary_event_type" />
147             </inthub:columnMappings>
148             <inthub:primaryDateFormat>yyyy/mm/dd</inthub:primaryDateFormat
    >
149             <inthub:isBroadcastDate>true</inthub:isBroadcastDate>
150             <inthub:primaryEventType>PRIMARY</inthub:primaryEventType>
151             <inthub:secondaryEventType>SECONDARY</inthub:secondaryEventTyp
    e>
152             <inthub:isDefaultBehavior>False</inthub:isDefaultBehavior>
153             <inthub:ignoreValues>C,B,P</inthub:ignoreValues>
154             <inthub:skipLines />
155             <inthub:encodingType>UTF-8</inthub:encodingType>
156         </inthub:mapping>
157         <inthub:mapping xsi:type="inthub:xml">
158             <inthub:fileNamePattern>*.xml</inthub:fileNamePattern>
159             <inthub:scheduleFileNameInfo />
160             <inthub:columnMappings>
161                 <inthub:columnMapping mappingReference="" fieldName="chann
    el" />
```

Copyright © 2024 Vizrt

```
                <inthub:columnMapping mappingReference="" fieldName="ignore" />
                <inthub:columnMapping mappingReference="/date/" fieldName="date" />
                <inthub:columnMapping mappingReference="/time/" fieldName="time" />
                <inthub:columnMapping mappingReference="@dateT@time" fieldName="begin_at" />
                <inthub:columnMapping mappingReference="/duration/" fieldName="duration" />
                <inthub:columnMapping mappingReference="/time/" fieldName="secondary_timecode" />
                <inthub:columnMapping mappingReference="/title/" fieldName="description" />
                <inthub:columnMapping mappingReference="/id/" fieldName="event_id" />
                <inthub:columnMapping mappingReference="" fieldName="trigger_id" />
                <inthub:columnMapping mappingReference="/type/" fieldName="event_type" />
                <inthub:columnMapping mappingReference="/id/" fieldName="secondary_page" />
                <inthub:columnMapping mappingReference="/duration/" fieldName="secondary_duration" />
                <inthub:columnMapping mappingReference="/secondary_type/" fieldName="secondary_event_type" />
            </inthub:columnMappings>
            <inthub:primaryDateFormat>YYYYMMDD</inthub:primaryDateFormat>
            <inthub:isBroadcastDate>false</inthub:isBroadcastDate>
            <inthub:primaryEventType>MAIN</inthub:primaryEventType>
            <inthub:secondaryEventType>SECONDARY</inthub:secondaryEventType>
            <inthub:isDefaultBehavior>false</inthub:isDefaultBehavior>
            <inthub:ignoreValues />
            <inthub:primaryEventPath>/event/</inthub:primaryEventPath>
            <inthub:secondaryInsidePrimaryEvents>false</inthub:secondaryInsidePrimaryEvents>
            <inthub:secondaryEventPath>/event/</inthub:secondaryEventPath>
            <inthub:channelNodeLocationTakenFromRoot>false</inthub:channelNodeLocationTakenFromRoot>
            <inthub:dateNodeLocationTakenFromRoot>false</inthub:dateNodeLocationTakenFromRoot>
        </inthub:mapping>
        <inthub:mapping xsi:type="inthub:excel">
            <inthub:fileNamePattern />
            <inthub:scheduleFileNameInfo />
            <inthub:columnMappings>
                <inthub:columnMapping mappingReference="" fieldName="channel" />
                <inthub:columnMapping mappingReference="" fieldName="ignore" />
                <inthub:columnMapping mappingReference="$B" fieldName="date" />
```

```
                <inthub:columnMapping mappingReference="$D" fieldName="tim
e" />
                <inthub:columnMapping mappingReference="@dateT@time"
fieldName="begin_at" />
                <inthub:columnMapping mappingReference="$F" fieldName="dur
ation" />
                <inthub:columnMapping mappingReference="$D" fieldName="sec
ondary_timecode" />
                <inthub:columnMapping mappingReference="$E" fieldName="des
cription" />
                <inthub:columnMapping mappingReference="$C" fieldName="eve
nt_id" />
                <inthub:columnMapping mappingReference="" fieldName="trigg
er_id" />
                <inthub:columnMapping mappingReference="$G" fieldName="eve
nt_type" />
                <inthub:columnMapping mappingReference="$D" fieldName="sec
ondary_page" />
            </inthub:columnMappings>
            <inthub:primaryDateFormat>YYYYMMDD</inthub:primaryDateFormat>
            <inthub:isBroadcastDate>false</inthub:isBroadcastDate>
            <inthub:primaryEventType>MAIN</inthub:primaryEventType>
            <inthub:secondaryEventType>SECONDARY</inthub:secondaryEventTyp
e>
            <inthub:isDefaultBehavior>false</inthub:isDefaultBehavior>
            <inthub:ignoreValues />
            <inthub:skipRows />
        </inthub:mapping>
        <inthub:mapping xsi:type="inthub:lst">
            <inthub:fileNamePattern>*.*</inthub:fileNamePattern>
            <inthub:scheduleFileNameInfo>DD_MM_YYYY_CCCC</inthub:scheduleF
ileNameInfo>
            <inthub:columnMappings>
                <inthub:columnMapping mappingReference="channel" fieldName="
channel" />
                <inthub:columnMapping mappingReference="ignore" fieldName="
ignore" />
                <inthub:columnMapping mappingReference="date" fieldName="d
ate" />
                <inthub:columnMapping mappingReference="time" fieldName="t
ime" />
                <inthub:columnMapping mappingReference="@dateT@time"
fieldName="begin_at" />
                <inthub:columnMapping mappingReference="duration"
fieldName="duration" />
                <inthub:columnMapping mappingReference="" fieldName="timec
ode" />
                <inthub:columnMapping mappingReference="description"
fieldName="description" />
                <inthub:columnMapping mappingReference="event_id"
fieldName="event_id" />
                <inthub:columnMapping mappingReference="trigger_id"
fieldName="trigger_id" />
```

```
227                 <inthub:columnMapping mappingReference="segment" fieldName="
    segment" />
228                 <inthub:columnMapping mappingReference="event_id"
    fieldName="ReconcileKey" />
229                 <inthub:columnMapping mappingReference="SOM" fieldName="SO
    M" />
230                 <inthub:columnMapping mappingReference="effect1" fieldName="
    effect1" />
231                 <inthub:columnMapping mappingReference="effect2" fieldName="
    effect2" />
232                 <inthub:columnMapping mappingReference="effect3" fieldName="
    effect3" />
233                 <inthub:columnMapping mappingReference="secondary_page"
    fieldName="secondary_page" />
234                 <inthub:columnMapping mappingReference="secondary_timecode
    " fieldName="secondary_timecode" />
235                 <inthub:columnMapping mappingReference="secondary_event_id
    " fieldName="secondary_event_id" />
236                 <inthub:columnMapping mappingReference="secondary_trigger_
    id" fieldName="secondary_trigger_id" />
237                 <inthub:columnMapping mappingReference="event_type"
    fieldName="event_type" />
238                 <inthub:columnMapping mappingReference="overlay_type"
    fieldName="overlay_type" />
239                 <inthub:columnMapping mappingReference="@time+@ReconcileKe
    y" fieldName="TimeAndRKey" />
240             </inthub:columnMappings>
241             <inthub:primaryDateFormat>DD/MM/YYYY</inthub:primaryDateFormat
    >
242             <inthub:isBroadcastDate>false</inthub:isBroadcastDate>
243             <inthub:primaryEventType>MAIN</inthub:primaryEventType>
244             <inthub:secondaryEventType>SECONDARY</inthub:secondaryEventTyp
    e>
245             <inthub:isDefaultBehavior>false</inthub:isDefaultBehavior>
246             <inthub:ignoreValues />
247             <inthub:useReconcileKeyAsHouseId>false</inthub:useReconcileKey
    AsHouseId>
248             <inthub:getDateFromSchedule>false</inthub:getDateFromSchedule>
249             <inthub:codePage />
250             <inthub:useUsd>false</inthub:useUsd>
251             <inthub:usdName />
252             <inthub:macroName />
253         </inthub:mapping>
254     </inthub:mappings>
255     <inthub:primaryEventFilters />
256 </inthub:channelSettings>
```

Disallowed Actions

This end-point does not allow POST, PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

Playlists (Version 2.0)

- GET Playlists in Channel (Version 2.0)
- POST a Playlist to Channel (Version 2.0)
- GET a Playlist in Channel (Version 2.0)
- PUT on Playlist in Channel (Version 2.0)
- DELETE a Playlist in Channel (Version 2.0)
- GET State of Playlist (Version 2.0)
- POST Refresh request on Playlist (Version 2.0)
- Playlist (Version 2.0)

Copyright © 2024 Vizrt

GET Playlists in Channel (Version 2.0)

Get the full collection of playlists associated with this channel.

Syntax

```
GET /api/channels/[channel]/playlists
```

Request

```
GET /api/channels/test/playlists HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=list
<inthub:list xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:playlistElements" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:playlist>
        <inthub:name>continuous</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=playlist" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous" />
        <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=playlist" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous" />
        <inthub:refLink rel="primaryevents" type="application/vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents" />
        <inthub:refLink rel="state" type="application/vizrt.integrationshubv2+xml; type=playliststate" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/state" />
        <inthub:refLink rel="refresh" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/refresh" />
        <inthub:description>Continuous Playlist</inthub:description>
        <inthub:broadcastDate>continuous</inthub:broadcastDate>
        <inthub:loaded />
        <inthub:available />
        <inthub:updated>2023-01-25T16:24:39Z</inthub:updated>
    </inthub:playlist>
</inthub:list>
```

Copyright © 2024 Vizrt

Disallowed Actions

This end-point does not allow PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

POST a Playlist to Channel (Version 2.0)

Add a playlist to a channel. For normal installations a single continuous playlist is the recommended approach.

Name can only be "continuous" or in the date format "yyyy_MM_dd".

Broadcast date must match the name, for continuous playlist it can only be "continuous", for a playlist in date format, the broadcast date must match the playlist date with format "yyyy-MM-dd".

Description is automatically set if left empty. For continuous playlist is set to "Continuous Playlist" and for playlist in date format is set to "Playlist for yyyy-MM-dd" by default.

If any values are set for **loaded**, **available** and **update**, the payload will be accepted but the values will be ignored as they can't be set.

Attempting to add a playlist that already exists will return a '*HTTP 409 Can't add playlist 'NNN', it already exists*' .

Syntax

```
POST /api/channels/[channel]/playlists
```

Continuous Playlist

Request - Continuous Playlist

```
POST /api/channels/test/playlists HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=playlist
<inthub:playlist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="
http://www.vizrt.com/integrationshub">
    <inthub:name>continuous</inthub:name>
    <inthub:description>Continuous Playlist</inthub:description>
    <inthub:broadcastDate>continuous</inthub:broadcastDate>
</inthub:playlist>
```

Response

```
1  HTTP/1.1 201 Created
2  Content-Type: application/vizrt.integrationshubv2+xml; type=playlist
3  <inthub:playlist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:inthub="http://www.vizrt.com/integrationshub">
4      <inthub:name>continuous</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=playlist" href="http://127.0.0.1:9991/
   api/channels/test/playlists/continuous" />
```

Copyright © 2024 Vizrt

```
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=playlist" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous" />
    <inthub:refLink rel="primaryevents" type="application/vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents" />
    <inthub:refLink rel="state" type="application/vizrt.integrationshubv2+xml; type=playliststate" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/state" />
    <inthub:refLink rel="refresh" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/refresh" />
    <inthub:description>Continuous Playlist</inthub:description>
    <inthub:broadcastDate>continuous</inthub:broadcastDate>
    <inthub:loaded />
    <inthub:available />
    <inthub:updated />
</inthub:playlist>
```

Playlist In Date Format

If events should be split by date or playlist for other usage is required, see the following example:

Request - Date Playlist

```
POST /api/channels/test/playlists HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=playlist
<inthub:playlist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>2019_12_01</inthub:name>
    <inthub:description>Playlist for 2019-12-01</inthub:description>
    <inthub:broadcastDate>2019-12-01</inthub:broadcastDate>
</inthub:playlist>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=playlist
<inthub:playlist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>2019_12_01</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=playlist" href="http://127.0.0.1:9991/api/channels/test/playlists/2019_12_01" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=playlist" href="http://127.0.0.1:9991/api/channels/test/playlists/2019_12_01" />
```

Copyright © 2024 Vizrt

```
    <inthub:refLink rel="primaryevents" type="application/vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/channels/test/playlists/2019_12_01/primaryevents" />
    <inthub:refLink rel="state" type="application/vizrt.integrationshubv2+xml; type=playliststate" href="http://127.0.0.1:9991/api/channels/test/playlists/2019_12_01/state" />
    <inthub:refLink rel="refresh" href="http://127.0.0.1:9991/api/channels/test/playlists/2019_12_01/refresh" />
    <inthub:description>Playlist for 2019-12-01</inthub:description>
    <inthub:broadcastDate>2019-12-01</inthub:broadcastDate>
    <inthub:loaded />
    <inthub:available />
    <inthub:updated />
</inthub:playlist>
```

Note: The full contract can be used for POST, fields that are read only will be ignored.

Copyright © 2024 Vizrt

GET a Playlist in Channel (Version 2.0)

By following the downlink from the collection of playlists under a channel we can retrieve a single named playlist.

Syntax

```
GET /api/channels/[channel]/playlists/[playlist]
```

Request

```
GET /api/channels/test/playlists/continuous HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=playlist
<inthub:playlist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>continuous</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=playlist" href="http://127.0.0.1:9991/
api/channels/test/playlists/continuous" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=playlist" href="http://127.0.0.1:9991/
api/channels/test/playlists/continuous" />
    <inthub:refLink rel="primaryevents" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/playlists/continuous/primaryevents" />
    <inthub:refLink rel="state" type="application/
vizrt.integrationshubv2+xml; type=playliststate" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/state" />
    <inthub:refLink rel="refresh" href="http://127.0.0.1:9991/api/
channels/test/playlists/continuous/refresh" />
    <inthub:description>Continuous Playlist</inthub:description>
    <inthub:broadcastDate>continuous</inthub:broadcastDate>
    <inthub:loaded>1.00</inthub:loaded>
    <inthub:available>0.81</inthub:available>
    <inthub:updated>2023-01-25T10:47:44Z</inthub:updated>
</inthub:playlist>
```

Disallowed Actions

This end-point does not allow POST, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

PUT on Playlist in Channel (Version 2.0)

Update a playlist definition.

Name and **broadcast date** can't be adjusted from it's original value.

Description is automatically set, if left empty.

If any values are set for **loaded, available** and **update**, the payload will be accepted but the values will be ignored as they can't be set.

Syntax

```
PUT /api/channels/[channel]/playlists/[playlist]
```

Request

```
PUT /api/channels/test/playlists/continuous HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=playlist
<inthub:playlist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="
http://www.vizrt.com/integrationshub">
    <inthub:name>continuous</inthub:name>
    <inthub:description>Continuous Playlist1</inthub:description>
    <inthub:broadcastDate>continuous</inthub:broadcastDate>
</inthub:playlist>
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=playlist
3  <inthub:playlist xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:inthub="http://www.vizrt.com/integrationshub">
4      <inthub:name>continuous</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=playlist" href="http://127.0.0.1:9991/
   api/channels/test/playlists/continuous" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=playlist" href="http://127.0.0.1:9991/
   api/channels/test/playlists/continuous" />
7      <inthub:refLink rel="primaryevents" type="application/
   vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
   channels/test/playlists/continuous/primaryevents" />
8      <inthub:refLink rel="state" type="application/
   vizrt.integrationshubv2+xml; type=playliststate" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/state" />
```

Copyright © 2024 Vizrt

```
    <inthub:refLink rel="refresh" href="http://127.0.0.1:9991/api/
channels/test/playlists/continuous/refresh" />
    <inthub:description>Continuous Playlist1</inthub:description>
    <inthub:broadcastDate>continuous</inthub:broadcastDate>
    <inthub:loaded />
    <inthub:available />
    <inthub:updated>2023-01-25T15:13:10Z</inthub:updated>
</inthub:playlist>
```

Copyright © 2024 Vizrt

DELETE a Playlist in Channel (Version 2.0)

Delete an existing Playlist from a Channel.

Syntax

```
DELETE /api/channels/[channel]/playlists/[playlist]
```

Request

```
DELETE /api/channels/test/playlists/2022_07_03 HTTP/1.1
```

Response

```
1  HTTP/1.1 204 No Content
```

Note: If trying to delete a non-existing Playlist, the returned error-code will be *HTTP 404* with a reference to relevant info used in the request-uri.

Copyright © 2024 Vizrt

GET State of Playlist (Version 2.0)

Retrieve the state-information of a playlist.

Syntax

```
GET /api/channels/[channel]/playlists/[playlist]/state
```

Request

```
GET /api/channels/test/playlists/continuous/state HTTP/1.1
```

Response if Playlist state is NOT SET

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=playliststate
<inthub:playlistState xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=playliststate" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/state" />
    <inthub:state>not set</inthub:state>
</inthub:playlistState>
```

Response if Playlist state is RUNNING

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=playliststate
<inthub:playlistState xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=playliststate" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/state" />
    <inthub:state>running</inthub:state>
</inthub:playlistState>
```

Response if Playlist state is STOPPED

```
HTTP/1.1 200 OK
```

Copyright © 2024 Vizrt

```
Content-Type: application/vizrt.integrationshubv2+xml; type=playliststate
<inthub:playlistState xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=playliststate" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/state" />
    <inthub:state>stopped</inthub:state>
</inthub:playlistState>
```

Disallowed Actions

This end-point does not allow POST, PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

POST Refresh request on Playlist (Version 2.0)

Request Integrations Hub to perform a Refresh of the data gathered during a Page Content Filling run, on a specific playlist. This allows the data to be refreshed on the full playlist even if no data in the playlist itself was adjusted. Best used when other related playlists, in same or other channels, or dependent Database's data, is updated and needed to be reflected in the playlist in question.

Syntax

```
POST /api/channels/[channel]/playlists/[playlist]/refresh
```

Request

```
POST /api/channels/test/playlists/2022_07_03/refresh HTTP/1.1
```

Response

```
1  HTTP/1.1 204 No Content
```

Copyright © 2024 Vizrt

Playlist (Version 2.0)

- GET Primary Events in Playlist (Version 2.0)
- POST a Primary Event to Playlist (Version 2.0)
- GET a Primary Event in a Playlist (Version 2.0)
- PUT on Primary Event in Playlist (Version 2.0)
- DELETE a Primary Event in a Playlist (Version 2.0)
- POST a Secondary Event under Primary Event (Version 2.0)
- GET Secondary Events under Primary Event (Version 2.0)
- GET a Secondary Event under Primary Event in Playlist (Version 2.0)
- PUT on Secondary Event under a Primary Event in Playlist (Version 2.0)
- DELETE a Secondary Event under a Primary Event in Playlist (Version 2.0)
- GET Page Content from a Secondary Event under a Primary Event in Playlist (Version 2.0)
- PUT Page Content from a Secondary Event under a Primary Event in Playlist (Version 2.0)

Copyright © 2024 Vizrt

GET Primary Events In Playlist (Version 2.0)

By following the downlink from the description of a single playlist we can retrieve a collection of primary events.

Take note that the collection of primary events contains a compact version of a Primary Event. Use the downlink for each Primary Event to get details.

Syntax

```
GET /api/channels/[channel]/playlists/[playlist]/primaryevents
```

Request

```
GET /api/channels/test/playlists/continuous/primaryevents HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=list
<inthub:list xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:primaryEventsCompact" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:primaryEventCompact>
        <inthub:name>primary_element_1_0_0</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=primaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_element_1_0_0" />
        <inthub:refLink rel="secondaryevents" type="application/vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_element_1_0_0/secondaryevents" />
        <inthub:description>Le radeau LE CRASH DU VOL 29</inthub:description>
        <inthub:startTime>2008-04-23T03:48:17.280Z</inthub:startTime>
        <inthub:duration>08:58:13.000</inthub:duration>
        <inthub:ignore>false</inthub:ignore>
        <inthub:triggerId />
        <inthub:houseId />
        <inthub:isLocked>false</inthub:isLocked>
        <inthub:isLive>false</inthub:isLive>
        <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
        <inthub:available>1.00</inthub:available>
        <inthub:loaded>1.00</inthub:loaded>
```

Copyright © 2024 Vizrt

```
        <inthub:infoIntegrationsHub>[A:11:47:44]</inthub:infoIntegrationsHub>
        <inthub:createdBy>schedule</inthub:createdBy>
        <inthub:error />
    </inthub:primaryEventCompact>
    <inthub:primaryEventCompact>
        <inthub:name>primary_element_2_0_0</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=primaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_element_2_0_0" />
        <inthub:refLink rel="secondaryevents" type="application/vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_element_2_0_0/secondaryevents" />
        <inthub:description>Flicka</inthub:description>
        <inthub:startTime>2008-04-23T04:15:54.160Z</inthub:startTime>
        <inthub:duration>37:49:25.000</inthub:duration>
        <inthub:ignore>false</inthub:ignore>
        <inthub:triggerId />
        <inthub:houseId />
        <inthub:isLocked>false</inthub:isLocked>
        <inthub:isLive>false</inthub:isLive>
        <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
        <inthub:available>1.00</inthub:available>
        <inthub:loaded>1.00</inthub:loaded>
        <inthub:infoIntegrationsHub>[A:11:47:44]</inthub:infoIntegrationsHub>
        <inthub:createdBy>schedule</inthub:createdBy>
        <inthub:error />
    </inthub:primaryEventCompact>
</inthub:list>
```

Disallowed Actions

This end-point does not allow PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

POST A Primary Event To Playlist (Version 2.0)

Add one Primary Event to an existing playlist.

Description, duration, ignore, trigger id, house id, mark in, mark out, created by and **metadata** can be empty.

Name, is live, is locked and **start time** cannot be empty. **Start time** should be in format "yyyy-MM-ddTHH:mm:ss.fffZ".

Ignore defaults to "false" if empty.

Created by defaults to "inthub" if empty.

House id should be specified to create a primary video event, where the house id is a Viz One video asset. If the house id exists in VizOne, the video element will be correctly created. If the house id does not exist in VizOne, then a placeholder can be created if it was enabled in the Channel Settings, otherwise the primary event will be created, with error.

If **mark in** and **mark out** have values when house id is empty, they will be ignored as they can't be set for non-video events.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **error**, the payload will be accepted but the values will be ignored as they can't be set.

Attempting to add a primary event that already exists will return a '*HTTP 409 Can't add primary event 'NNN', it already exists'* .

> **Note:** After this request is done the system may perform a Page Content Filling process, against Promo Server, when no additional requests are sent within a configurable timeout.

> **Note:** After this request is done the system runs the rules specified under the Rules section.

Syntax

```
POST /api/channels/[channel]/playlists/[playlist]/primaryevents
```

Primary event with no video (house id empty)

Request

```
POST /api/channels/test/playlists/continuous/primaryevents HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=primaryevent
<inthub:primaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>primary_1</inthub:name>
    <inthub:description>Top Gun</inthub:description>
    <inthub:startTime>2018-11-06T07:13:40.000Z</inthub:startTime>
    <inthub:duration>02:12:00.000</inthub:duration>
```

Copyright © 2024 Vizrt

```
    <inthub:ignore>False</inthub:ignore>
    <inthub:triggerId>primary1001</inthub:triggerId>
    <inthub:houseId />
    <inthub:markIn />
    <inthub:markOut />
    <inthub:isLocked>false</inthub:isLocked>
    <inthub:isLive>false</inthub:isLive>
    <inthub:createdBy>component</inthub:createdBy>
    <inthub:metadata>
        <inthub:attribute name="date">2022-03-10</inthub:attribute>
        <inthub:attribute name="time">21:00:00:00</inthub:attribute>
        <inthub:attribute name="event_type">PRIMARY</inthub:attribute>
        <inthub:attribute name="some_metadata">this shows random metadata</inthub:att
ribute>
    </inthub:metadata>
</inthub:primaryEvent>
```

Response

```
1  HTTP/1.1 201 Created
2  Content-Type: application/vizrt.integrationshubv2+xml; type=primaryevent
3  <inthub:primaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:inthub="http://www.vizrt.com/integrationshub">
4      <inthub:name>primary_1</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=primaryevent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_1" />
6      <inthub:refLink rel="secondaryevents" type="application/
   vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
   channels/test/playlists/continuous/primaryevents/primary_1/
   secondaryevents" />
7      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=primaryevent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_1" />
8      <inthub:description>Top Gun</inthub:description>
9      <inthub:startTime>2018-11-06T07:13:40.000Z</inthub:startTime>
10     <inthub:duration>02:12:00.000</inthub:duration>
11     <inthub:ignore>false</inthub:ignore>
12     <inthub:triggerId>primary1001</inthub:triggerId>
13     <inthub:houseId />
14     <inthub:markIn />
15     <inthub:markOut />
16     <inthub:isLocked>false</inthub:isLocked>
17     <inthub:isLive>false</inthub:isLive>
18     <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
19     <inthub:available />
20     <inthub:loaded />
21     <inthub:infoIntegrationsHub>[A:15:26:43]</inthub:infoIntegrationsHub>
22     <inthub:createdBy>component</inthub:createdBy>
```

Copyright © 2024 Vizrt

```
    <inthub:error />
    <inthub:metadata>
        <inthub:attribute name="date">2022-03-10</inthub:attribute>
        <inthub:attribute name="time">21:00:00:00</inthub:attribute>
        <inthub:attribute name="event_type">PRIMARY</inthub:attribute>
        <inthub:attribute name="some_metadata">this shows random
metadata</inthub:attribute>
    </inthub:metadata>
</inthub:primaryEvent>
```

Primary event with video (house id set)

Request

```
POST /api/channels/test/playlists/continuous/primaryevents HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=primaryevent
<inthub:primaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>primary_1_video</inthub:name>
    <inthub:description>Top Gun Trailer</inthub:description>
    <inthub:startTime>2018-11-06T07:13:40.000Z</inthub:startTime>
    <inthub:duration>00:00:50.000</inthub:duration>
    <inthub:ignore>False</inthub:ignore>
    <inthub:triggerId>primary1001</inthub:triggerId>
    <inthub:houseId>ABCDEFG</inthub:houseId>
    <inthub:markIn>00:00:01.000</inthub:markIn>
    <inthub:markOut>00:00:09.000</inthub:markOut>
    <inthub:isLocked>false</inthub:isLocked>
    <inthub:isLive>false</inthub:isLive>
    <inthub:createdBy />
    <inthub:metadata />
</inthub:primaryEvent>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=primaryevent
<inthub:primaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>primary_1_video</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=primaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_1_video" />
    <inthub:refLink rel="secondaryevents" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/playlists/continuous/primaryevents/primary_1_video/
secondaryevents" />
```

Copyright © 2024 Vizrt

```
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=primaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_1_video" />
    <inthub:description>Top Gun Trailer</inthub:description>
    <inthub:startTime>2018-11-06T07:13:40.000Z</inthub:startTime>
    <inthub:duration>00:00:50.000</inthub:duration>
    <inthub:ignore>false</inthub:ignore>
    <inthub:triggerId>primary1001</inthub:triggerId>
    <inthub:houseId>ABCDEFG</inthub:houseId>
    <inthub:markIn>00:00:01.000</inthub:markIn>
    <inthub:markOut>00:00:09.000</inthub:markOut>
    <inthub:isLocked>false</inthub:isLocked>
    <inthub:isLive>false</inthub:isLive>
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:available />
    <inthub:loaded />
    <inthub:infoIntegrationsHub>[A:15:26:43]</inthub:infoIntegrationsHub>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:error />
    <inthub:metadata />
</inthub:primaryEvent>
```

Note: The full contract can be used for POST, fields that are read only will be ignored.

Copyright © 2024 Vizrt

GET A Primary Event In A Playlist (Version 2.0)

Following the downlink while retrieving the collection of Primary Events under a Playlist one can also do a GET directly on a single Primary Event.

Take note that the collection of entries under `meta_data` sub-section is not set in size or content, and depends on the actual meta data attributes on the primary event in the playlist.

Syntax

```
GET /api/channels/[channel]/playlists/[playlist]/primaryevents/[primary_event]
```

Request

```
GET /api/channels/test/playlists/continuous/primaryevents/primary_element_1_0_0 HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=primaryevent
<inthub:primaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>primary_element_1_0_0</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=primaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_element_1_0_0" />
    <inthub:refLink rel="secondaryevents" type="application/vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_element_1_0_0/secondaryevents" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=primaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_element_1_0_0" />
    <inthub:description>Le radeau LE CRASH DU VOL 29</inthub:description>
    <inthub:startTime>2008-04-23T03:48:17.280Z</inthub:startTime>
    <inthub:duration>08:58:13.000</inthub:duration>
    <inthub:ignore>false</inthub:ignore>
    <inthub:triggerId />
    <inthub:houseId />
    <inthub:markIn />
    <inthub:markOut />
    <inthub:isLocked>false</inthub:isLocked>
    <inthub:isLive>false</inthub:isLive>
```

Copyright © 2024 Vizrt

```
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:available>1.00</inthub:available>
    <inthub:loaded>1.00</inthub:loaded>
    <inthub:infoIntegrationsHub>[A:11:47:44]</inthub:infoIntegrationsHub>
    <inthub:createdBy>schedule</inthub:createdBy>
    <inthub:error />
    <inthub:metadata>
        <inthub:attribute name="date">2008-04-23</inthub:attribute>
        <inthub:attribute name="time">05:48:17:07</inthub:attribute>
        <inthub:attribute name="event_type">MAIN</inthub:attribute>
    </inthub:metadata>
</inthub:primaryEvent>
```

Disallowed Actions

This end-point does not allow POST, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

PUT On Primary Event In Playlist (Version 2.0)

Update a Primary Event definition.

Name, house id and **created by** can't be adjusted from it's original value.

Description, duration, ignore, trigger id, mark in, mark out and **metadata** can be empty.

Ignore defaults to "false" if empty.

If **mark in** and **mark out** have values when house id is empty, they will be ignored as they can't be set for non-video events.

Metadata section will update completely according to how it is specified. If the metadata section is empty, all metadata will be removed. If a metadata attribute is removed, then that metadata will be removed. If metadata values are updated, they will be updated accordingly.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **error**, the payload will be accepted but the values will be ignored as they can't be set.

> ⚠ **Note:** After this request is done the system may perform a Page Content Filling process, against Promo Server, when no additional requests are sent within a configurable timeout.

> ⚠ **Note:** After this request is done the system runs the rules specified under the Rules section.

Syntax

```
PUT /api/channels/[channel]/playlists/[playlist]/primaryevents/[primary_event]
```

Request

```
PUT /api/channels/test/playlists/continuous/primaryevents/primary_element_1_0_0 HTTP/
1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=primaryevent
<inthub:primaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>primary_element_1_0_0</inthub:name>
    <inthub:description>New description</inthub:description>
    <inthub:startTime>2018-11-06T07:13:40.000Z</inthub:startTime>
    <inthub:duration>02:12:00.000</inthub:duration>
    <inthub:ignore>False</inthub:ignore>
    <inthub:triggerId>primary1001</inthub:triggerId>
    <inthub:createdBy>schedule</inthub:createdBy>
    <inthub:houseId />
    <inthub:markIn />
    <inthub:markOut />
    <inthub:isLocked>false</inthub:isLocked>
    <inthub:isLive>false</inthub:isLive>
```

Copyright © 2024 Vizrt

```
    <inthub:metadata>
        <inthub:attribute name="date">2022-03-10</inthub:attribute>
        <inthub:attribute name="time">21:00:00:00</inthub:attribute>
        <inthub:attribute name="event_type">PRIMARY</inthub:attribute>
        <inthub:attribute name="some_metadata">this shows updated metadata</inthub:at
tribute>
    </inthub:metadata>
</inthub:primaryEvent>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=primaryevent
<inthub:primaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>primary_element_1_0_0</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=primaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_element_1_0_0" />
    <inthub:refLink rel="secondaryevents" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/playlists/continuous/primaryevents/primary_element_1_0_0/
secondaryevents" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=primaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_element_1_0_0" />
    <inthub:description>New description</inthub:description>
    <inthub:startTime>2018-11-06T07:13:40.000Z</inthub:startTime>
    <inthub:duration>02:12:00.000</inthub:duration>
    <inthub:ignore>false</inthub:ignore>
    <inthub:triggerId>primary1001</inthub:triggerId>
    <inthub:houseId />
    <inthub:markIn />
    <inthub:markOut />
    <inthub:isLocked>false</inthub:isLocked>
    <inthub:isLive>false</inthub:isLive>
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:available />
    <inthub:loaded />
    <inthub:infoIntegrationsHub>[U:16:23:40]</inthub:infoIntegrationsHub>
    <inthub:createdBy>schedule</inthub:createdBy>
    <inthub:error />
    <inthub:metadata>
        <inthub:attribute name="date">2022-03-10</inthub:attribute>
        <inthub:attribute name="time">21:00:00:00</inthub:attribute>
        <inthub:attribute name="event_type">PRIMARY</inthub:attribute>
        <inthub:attribute name="some_metadata">this shows updated
metadata</inthub:attribute>
    </inthub:metadata>
```

Copyright © 2024 Vizrt

```
30  </inthub:primaryEvent>
```

Note: If doing a PUT against a URI that doesn't represent an existing Primary Event, the response is '*HTTP/1.1 404 Can't get primary event 'primary_event' for playlist 'playlist_name' in channel 'channel_name'. Primary event not found* ' and no update is performed.

DELETE A Primary Event In A Playlist (Version 2.0)

Delete a Primary Event from a playlist.

Syntax

```
DELETE /api/channels/[channel]/playlists/[playlist]/primaryevents/[primary_event]
```

Request

```
DELETE /api/channels/test/playlists/continuous/primaryevents/primary_1 HTTP/1.1
```

Response

```
1  HTTP/1.1 204 No Content
```

Note: If trying to delete a non-existing Primary Event the returned error-code will be *HTTP 404*, with a reference to relevant info used in the request-uri.

Copyright © 2024 Vizrt

POST A Secondary Event Under Primary Event (Version 2.0)

Add one secondary event to an existing primary event. There are several types of secondary events, examples below.

Attempting to add a secondary event that already exists, will return a '*HTTP 409 Can't add secondary event 'NNN', it already exists*'.

> **Note:** After this request is done the system performs a Page Content Filling process against Promo Server as soon as there is no immediate additional requests sent.

Syntax

```
POST /api/channels/[channel]/playlists/[playlist]/primaryevents/[primary_event]/
secondaryevents
```

Secondary event page

Description, duration, trigger id, offset, created by and **metadata** can be empty.

Name and **page id** cannot be empty. Attempting to add a secondary event with page id does not exist in page pool, will return a '*HTTP 400 Can't create secondary page event, page 'NNN' does not exist*'.

Description is set from the page of given page id, if left empty.

Duration is set from the page of the given page id, if left empty.

Created by defaults to "inthub" if empty.

If any values are set for **is manually edited, loaded, available, info integrations hub, info promo server** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents
HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_1</inthub:name>
    <inthub:description>not used</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:metadata>
        <inthub:attribute name="date">2022-03-10</inthub:attribute>
```

Copyright © 2024 Vizrt

```
        <inthub:attribute name="time">21:00:00:00</inthub:attribute>
        <inthub:attribute name="event_type">SECONDARY</inthub:attribute>
        <inthub:attribute name="some_metadata">this shows random metadata</inthub:att
ribute>
    </inthub:metadata>
</inthub:secondaryEvent>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/
integrationshub">
    <inthub:name>secondary_1_1</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_1/secondaryevents/secondary_1_1" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_1/secondaryevents/secondary_1_1" />
    <inthub:refLink rel="pagecontent" type="application/
vizrt.integrationshubv2+xml; type=pagecontent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_1/secondaryevents/secondary_1_1/pagecontent" />
    <inthub:description>not used</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:takeCount>0</inthub:takeCount>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[A:15:56:49]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:infoPromoServer />
    <inthub:metadata>
        <inthub:attribute name="date">2022-03-10</inthub:attribute>
        <inthub:attribute name="time">21:00:00:00</inthub:attribute>
        <inthub:attribute name="event_type">SECONDARY</inthub:attribute>
        <inthub:attribute name="some_metadata">this shows random
metadata</inthub:attribute>
    </inthub:metadata>
</inthub:secondaryEvent>
```

Secondary event reference page

Copyright © 2024 Vizrt

Description, duration, trigger id, offset and **created by** can be empty.

Name and **page id** cannot be empty. Attempting to add a reference secondary event with page id does not exist in page pool, will return a '*HTTP 400 Can't create reference secondary page event, page 'NNN' does not exist*'.

Description is set from the page of given page id, if left empty.

Duration is set from the page of the given page id, if left empty.

Created by defaults to "inthub" if empty.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents
HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:refPageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_ref</inthub:name>
    <inthub:description>Default Logo</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
1  HTTP/1.1 201 Created
2  Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
3  <inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance" xsi:type="inthub:refPageEvent" xmlns:inthub="http://
   www.vizrt.com/integrationshub">
4      <inthub:name>secondary_1_ref</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_1/secondaryevents/secondary_1_ref" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_1/secondaryevents/secondary_1_ref" />
7      <inthub:description>Default Logo</inthub:description>
8      <inthub:triggerId>735</inthub:triggerId>
9      <inthub:offset>00:00:10.000</inthub:offset>
10     <inthub:takeCount />
```

Copyright © 2024 Vizrt

```
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[A:16:10:05]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
</inthub:secondaryEvent>
```

Secondary event action

Description, trigger id, offset and created by can be empty.

Name and **action id** cannot be empty. Attempting to add a secondary action event with action id does not exist in action pool, will return a '*Can't create secondary action event, action 'NNN' does not exist*'.

Description is set from the action of given action id, if left empty.

Created by defaults to "inthub" if empty.

If any values are set for **is manually edited, loaded, available, info integrations hub, take count, action command** and **action category**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents
HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:actionEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_action</inthub:name>
    <inthub:description>A cleanup action</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:actionId>viz_cleanup</inthub:actionId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent

<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="inthub:actionEvent" xmlns:inthub="http://
www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_action</inthub:name>
```

Copyright © 2024 Vizrt

```
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents/secondary_1_action" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents/secondary_1_action" />
    <inthub:description>A cleanup action</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:takeCount />
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[A:16:24:12]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:actionId>viz_cleanup</inthub:actionId>
    <inthub:actionCommand>SCENE CLEANUP</inthub:actionCommand>
    <inthub:actionCategory>viz</inthub:actionCategory>
</inthub:secondaryEvent>
```

Secondary event pilot data element

Description, duration, trigger id, offset and **created by** can be empty.

Name and **pilot data element id** cannot be empty. **Pilot data element id** must be an integer.

Created by defaults to "inthub" if empty.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:pilotDataElementEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_pde</inthub:name>
    <inthub:description>Pilot data element</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:secondaryEvent>
```

Copyright © 2024 Vizrt

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:pilotDataElementEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_pde</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents/secondary_1_pde" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents/secondary_1_pde" />
    <inthub:description>Pilot data element</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:takeCount />
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[A:16:16:30]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
</inthub:secondaryEvent>
```

Secondary event message

Description, offset and **created by** can be empty.

Name cannot be empty.

Created by defaults to "inthub" if empty.

If any values are set for **info integrations hub**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:msgEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_message</inthub:name>
    <inthub:description>This is a message</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
```

Copyright © 2024 Vizrt

```
    <inthub:createdBy>schedule</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent

<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:msgEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_message</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents/secondary_1_message" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents/secondary_1_message" />
    <inthub:description>This is a message</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:createdBy>schedule</inthub:createdBy>
    <inthub:infoIntegrationsHub>[A:14:55:14]</inthub:infoIntegrationsHub>
</inthub:secondaryEvent>
```

Note: The full contract can be used for POST, fields that are read only will be ignored.

Copyright © 2024 Vizrt

GET Secondary Events Under Primary Event (Version 2.0)

Follow the downlink under a Primary Event to get a collection of secondary events under this Primary Event.

Take note that the collection of secondary events contains a compact version of a Secondary Event, and also that the content is depending on the type of Secondary Event. Use the downlink for each Secondary Event to get details.

Syntax

```
GET /api/channels/[channel]/playlists/[playlist]/primaryevents/[primary_event]/
secondaryevents
```

Request

```
GET /api/channels/test/playlists/continuous/primaryevents/primary_element_1_0_0/
secondaryevents HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=list
<inthub:list xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="inthub:secondaryEventsCompact" xmlns:inthub="http://
www.vizrt.com/integrationshub">
    <inthub:secondaryEventCompact xsi:type="inthub:pageEventCompact">
        <inthub:name>f4e0a087-c6f4-4f40-bca9-a51cdaf43634</inthub:name>
        <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_element_1_0_0/secondaryevents/f4e0a087-c6f4-4f40-bca9-
a51cdaf43634" />
        <inthub:description>The Smashing Pumpkins/Bullet With Butterfly
Wings</inthub:description>
        <inthub:triggerId />
        <inthub:offset>00:00:02.000</inthub:offset>
        <inthub:takeCount>0</inthub:takeCount>
        <inthub:createdBy>manual</inthub:createdBy>
        <inthub:infoIntegrationsHub />
        <inthub:loaded>1.00</inthub:loaded>
        <inthub:available>1.00</inthub:available>
        <inthub:isManuallyEdited>true</inthub:isManuallyEdited>
        <inthub:duration>00:00:10.000</inthub:duration>
        <inthub:pageId>6000</inthub:pageId>
        <inthub:infoPromoServer>[14:08:30]</inthub:infoPromoServer>
```

Copyright © 2024 Vizrt

```
    </inthub:secondaryEventCompact>
    <inthub:secondaryEventCompact xsi:type="inthub:refPageEventCompact">
        <inthub:name>1000</inthub:name>
        <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_element_1_0_0/secondaryevents/1000" />
        <inthub:description>DEFAULT_LOGO</inthub:description>
        <inthub:triggerId />
        <inthub:offset>00:00:00.000</inthub:offset>
        <inthub:takeCount>0</inthub:takeCount>
        <inthub:createdBy>schedule</inthub:createdBy>
        <inthub:infoIntegrationsHub>[A:14:07:54]</inthub:infoIntegrationsH
ub>
        <inthub:loaded>1.00</inthub:loaded>
        <inthub:available>1.00</inthub:available>
        <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
        <inthub:duration>00:09:07.890</inthub:duration>
        <inthub:pageId>1000</inthub:pageId>
    </inthub:secondaryEventCompact>
    <inthub:secondaryEventCompact xsi:type="inthub:pilotDataElementEventCo
mpact">
        <inthub:name>9000</inthub:name>
        <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_element_1_0_0/secondaryevents/9000" />
        <inthub:description />
        <inthub:triggerId />
        <inthub:offset>00:00:00.000</inthub:offset>
        <inthub:takeCount />
        <inthub:createdBy>schedule</inthub:createdBy>
        <inthub:infoIntegrationsHub>[A:14:08:28]</inthub:infoIntegrationsH
ub>
        <inthub:loaded />
        <inthub:available />
        <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
        <inthub:duration>00:09:07.890</inthub:duration>
        <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
    </inthub:secondaryEventCompact>
    <inthub:secondaryEventCompact xsi:type="inthub:actionEventCompact">
        <inthub:name>1c720ee7-8481-4d00-aa53-fe577b85bcbb</inthub:name>
        <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_element_1_0_0/secondaryevents/1c720ee7-8481-4d00-aa53-
fe577b85bcbb" />
        <inthub:description>Cleanup</inthub:description>
        <inthub:triggerId />
        <inthub:offset />
        <inthub:takeCount>0</inthub:takeCount>
        <inthub:createdBy>manual</inthub:createdBy>
        <inthub:infoIntegrationsHub />
```

Copyright © 2024 Vizrt

```
59          <inthub:loaded>1.00</inthub:loaded>
60          <inthub:available>1.00</inthub:available>
61          <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
62          <inthub:actionId>viz_cleanup</inthub:actionId>
63          <inthub:actionCommand>SCENE CLEANUP</inthub:actionCommand>
64          <inthub:actionCategory>viz</inthub:actionCategory>
65      </inthub:secondaryEventCompact>
66      <inthub:secondaryEventCompact xsi:type="inthub:msgEventCompact">
67          <inthub:name>secondary_msg_1</inthub:name>
68          <inthub:refLink rel="self" type="application/
    vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
    127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
    primary_element_1_0_0/secondaryevents/secondary_msg_1" />
69          <inthub:description>Element with Id '' could not be identified</
    inthub:description>
70          <inthub:offset>00:00:10.000</inthub:offset>
71          <inthub:createdBy>schedule</inthub:createdBy>
72          <inthub:infoIntegrationsHub>[A:14:55:14]</inthub:infoIntegrationsH
    ub>
73      </inthub:secondaryEventCompact>
74  </inthub:list>
```

Disallowed Actions

This end-point does not allow PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

GET A Secondary Event Under Primary Event In Playlist (Version 2.0)

Following the downlink while retrieving the collection of Secondary Events under a Primary Event one can also do a GET directly on a single Secondary Event.

Take note that the content depends on the type of Secondary Event, and the collection of entries under `meta_data` sub-section is not set in size or content, and depends on the actual meta data attributes on the secondary events in the playlist.

Syntax

```
GET /api/channels/[channel]/playlists/[playlist]/primaryevents/[primary_event]/
secondaryevents/[secondary_event]
```

Request

```
GET /api/channels/test/playlists/continuous/primaryevents/primary_element_1_0_0/
secondaryevents/f4e0a087-c6f4-4f40-bca9-a51cdaf43634 HTTP/1.1
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
3  <inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance" xsi:type="inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/
   integrationshub">
4      <inthub:name>f4e0a087-c6f4-4f40-bca9-a51cdaf43634</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_element_1_0_0/secondaryevents/f4e0a087-c6f4-4f40-bca9-
   a51cdaf43634" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_element_1_0_0/secondaryevents/f4e0a087-c6f4-4f40-bca9-
   a51cdaf43634" />
7      <inthub:refLink rel="pagecontent" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_element_1_0_0/secondaryevents/f4e0a087-c6f4-4f40-bca9-
   a51cdaf43634/pagecontent" />
8      <inthub:description>The Smashing Pumpkins/Bullet With Butterfly
   Wings</inthub:description>
9      <inthub:triggerId />
10     <inthub:offset>00:00:02.000</inthub:offset>
```

Copyright © 2024 Vizrt

```
    <inthub:takeCount>0</inthub:takeCount>
    <inthub:createdBy>manual</inthub:createdBy>
    <inthub:infoIntegrationsHub />
    <inthub:loaded>1.00</inthub:loaded>
    <inthub:available>1.00</inthub:available>
    <inthub:isManuallyEdited>true</inthub:isManuallyEdited>
    <inthub:duration>00:00:10.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:infoPromoServer>[13:40:14]</inthub:infoPromoServer>
    <inthub:metadata>
        <inthub:attribute name="some_metadata">some data</inthub:attribute
>
    </inthub:metadata>
</inthub:secondaryEvent>
```

Disallowed Actions

This end-point does not allow POST, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

PUT On Secondary Event Under A Primary Event In Playlist (Version 2.0)

Update a Secondary Event definition. There are several types of secondary events, examples below.

> ⚠ **Note:** After this request is done the system performs a Page Content Filling process against Promo Server as soon as there is no immediate additional requests sent.

Syntax

```
PUT /api/channels/[channel]/playlists/[playlist]/primaryevents/[primary_event]/
secondaryevents/[secondary_event]
```

Secondary event page

Name, page id and **created by** can't be adjusted from it's original value.

Description, duration, trigger id, offset and **metadata** can be empty.

Metadata section will update completely according to how it is specified. If the metadata section is empty, all metadata will be removed. If a metadata attribute is removed, then that metadata will be removed. If metadata values are updated, they will be updated accordingly.

If any values are set for **is manually edited, loaded, available, info integrations hub, info promo server** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/playlists/continuous/primaryevents/primary_element_2_0_0/
secondaryevents/secondary_1_1 HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_1</inthub:name>
    <inthub:description>new description</inthub:description>
    <inthub:offset>00:00:05.000</inthub:offset>
    <inthub:duration>00:00:01.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:metadata>
        <inthub:attribute name="date">2022-03-10</inthub:attribute>
        <inthub:attribute name="time">21:00:00:00</inthub:attribute>
        <inthub:attribute name="event_type">SECONDARY</inthub:attribute>
        <inthub:attribute name="some_metadata">this shows random metadata</inthub:att
ribute>
    </inthub:metadata>
</inthub:secondaryEvent>
```

Copyright © 2024 Vizrt

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_1</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_element_2_0_0/secondaryevents/secondary_1_1" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_element_2_0_0/secondaryevents/secondary_1_1" />
    <inthub:refLink rel="pagecontent" type="application/vizrt.integrationshubv2+xml; type=pagecontent" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/primary_element_2_0_0/secondaryevents/secondary_1_1/pagecontent" />
    <inthub:description>new description</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:05.000</inthub:offset>
    <inthub:takeCount>0</inthub:takeCount>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[U:17:15:15]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:01.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:infoPromoServer />
    <inthub:metadata>
        <inthub:attribute name="date">2022-03-10</inthub:attribute>
        <inthub:attribute name="time">21:00:00:00</inthub:attribute>
        <inthub:attribute name="event_type">SECONDARY</inthub:attribute>
        <inthub:attribute name="some_metadata">this shows random metadata</inthub:attribute>
    </inthub:metadata>
</inthub:secondaryEvent>
```

Secondary event reference page

Name, page id and **created by** can't be adjusted from it's original value.

Description, duration, trigger id and **offset** can be empty.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Copyright © 2024 Vizrt

Request

```
PUT /api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents/
secondary_1_ref HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:refPageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_ref</inthub:name>
    <inthub:description>New Logo</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
3  <inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance" xsi:type="inthub:refPageEvent" xmlns:inthub="http://
   www.vizrt.com/integrationshub">
4      <inthub:name>secondary_1_ref</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_1/secondaryevents/secondary_1_ref" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_1/secondaryevents/secondary_1_ref" />
7      <inthub:description>New Logo</inthub:description>
8      <inthub:triggerId>735</inthub:triggerId>
9      <inthub:offset>00:00:10.000</inthub:offset>
10     <inthub:takeCount />
11     <inthub:createdBy>inthub</inthub:createdBy>
12     <inthub:infoIntegrationsHub>[U:17:10:05]</inthub:infoIntegrationsHub>
13     <inthub:loaded />
14     <inthub:available />
15     <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
16     <inthub:duration>00:00:20.000</inthub:duration>
17     <inthub:pageId>6000</inthub:pageId>
18 </inthub:secondaryEvent>
```

Secondary event action

Name, action id and **created by** can't be adjusted from it's original value.

Description, trigger id, offset and **action command** can be empty.

If any values are set for **is manually edited, loaded, available, info integrations hub, take count** and **action category**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents/
secondary_1_action HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:actionEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_action</inthub:name>
    <inthub:description>A cleanup action</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:actionId>viz_cleanup</inthub:actionId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:actionCommand>SCENE CLEANUP 1</inthub:actionCommand>
</inthub:secondaryEvent>
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent

3  <inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance" xsi:type="inthub:actionEvent" xmlns:inthub="http://
   www.vizrt.com/integrationshub">
4      <inthub:name>secondary_1_action</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_1/secondaryevents/secondary_1_action" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_1/secondaryevents/secondary_1_action" />
7      <inthub:description>A cleanup action</inthub:description>
8      <inthub:triggerId>735</inthub:triggerId>
9      <inthub:offset>00:00:10.000</inthub:offset>
10     <inthub:takeCount />
11     <inthub:createdBy>inthub</inthub:createdBy>
12     <inthub:infoIntegrationsHub>[U:16:34:12]</inthub:infoIntegrationsHub>
13     <inthub:loaded />
14     <inthub:available />
15     <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
16     <inthub:actionId>viz_cleanup</inthub:actionId>
17     <inthub:actionCommand>SCENE CLEANUP 1</inthub:actionCommand>
```

Copyright © 2024 Vizrt

```
    <inthub:actionCategory>viz</inthub:actionCategory>
</inthub:secondaryEvent>
```

Secondary event pilot data element

Name, pilot data element id and **created by** can't be adjusted from it's original value.

Description, duration, trigger id and **offset** can be empty.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents/
secondary_1_pde HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:pilotDataElementEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_pde</inthub:name>
    <inthub:description>Pilot data element</inthub:description>
    <inthub:offset>00:00:12.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="inthub:pilotDataElementEvent" xmlns:inthub="http://
www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_pde</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_1/secondaryevents/secondary_1_pde" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_1/secondaryevents/secondary_1_pde" />
    <inthub:description>Pilot data element</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:12.000</inthub:offset>
    <inthub:takeCount />
    <inthub:createdBy>inthub</inthub:createdBy>
```

Copyright © 2024 Vizrt

```
    <inthub:infoIntegrationsHub>[U:21:10:30]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
</inthub:secondaryEvent>
```

Secondary event message

Name and **created by** can't be adjusted from it's original value.

Description and **offset** can be empty.

If any values are set for **info integrations hub**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/playlists/continuous/primaryevents/primary_1/secondaryevents/
secondary_1_message HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:msgEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_message</inthub:name>
    <inthub:description>This is an updated message</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:createdBy>schedule</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent

<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="inthub:msgEvent" xmlns:inthub="http://www.vizrt.com/
integrationshub">
    <inthub:name>secondary_1_message</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_1/secondaryevents/secondary_1_message" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
primary_1/secondaryevents/secondary_1_message" />
    <inthub:description>This is an updated message</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
```

Copyright © 2024 Vizrt

```
    <inthub:createdBy>schedule</inthub:createdBy>
    <inthub:infoIntegrationsHub>[U:14:57:04]</inthub:infoIntegrationsHub>
</inthub:secondaryEvent>
```

Copyright © 2024 Vizrt

DELETE A Secondary Event Under A Primary Event In Playlist (Version 2.0)

Delete a Secondary Event from a playlist.

Syntax

```
DELETE /api/channels/[channel]/playlists/[playlist]/primaryevents/[primary_event]/
secondaryevents/[secondary_event]
```

Request

```
DELETE /api/channels/test/playlists/continuous/primaryevents/primary_element_1_0_0/
secondaryevents/6000 HTTP/1.1
```

Response

```
1 HTTP/1.1 204 No Content
```

Note: If trying to delete a non-existing Secondary Event the returned error-code will be *HTTP 404*, with a reference to relevant info used in the request-uri.

Copyright © 2024 Vizrt

GET Page Content From A Secondary Event Under A Primary Event In Playlist (Version 2.0)

Following the downlink for a secondary event under a primary event to get page content.

Syntax

```
GET /api/channels/[channel]/playlists/[playlist]/primaryevents/[primary_event]/
secondaryevents/[secondary_event]/pagecontent
```

Request

```
GET /api/channels/test/playlists/continuous/primaryevents/primary_event_1/
secondaryevents/secondary_event_1/pagecontent HTTP/1.1
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=pagecontent
3  <inthub:pageContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:inthub="http://www.vizrt.com/integrationshub">
4      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_event_1/secondaryevents/secondary_event_1/pagecontent" />
5      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_event_1/secondaryevents/secondary_event_1/pagecontent" />
6      <inthub:pageField name="01butterfly">UP NEXT...</inthub:pageField>
7      <inthub:pageField name="02butterfly">CELEBRITY DEATH OCTAGON</inthub:p
   ageField>
8      <inthub:pageField name="03butterfly">Watch Lady Gaga go up against
   Marilyn Manson in the battle for subliminal supremacy. With guest referee
   Ozzy Osbourne.</inthub:pageField>
9      <inthub:pageField name="Videoclip"></inthub:pageField>
10 </inthub:pageContent>
```

Disallowed Actions

This end-point does not allow POST or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

PUT Page Content From A Secondary Event Under A Primary Event In Playlist (Version 2.0)

Update page content of a secondary event of type page.

Page field values are adjustable (empty value allowed), but the name of the field must remain unchanged. The full list of page fields needs to be present in the request, as page fields can't be added or deleted.

Syntax

```
PUT /api/channels/[channel]/playlists/[playlist]/primaryevents/[primary_event]/
secondaryevents/[secondary_event]/pagecontent
```

Request

```
PUT /api/channels/test/playlists/continuous/primaryevents/primary_event_1/
secondaryevents/secondary_event_1/pagecontent HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=pagecontent
<inthub:pageContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:pageField name="01LT">The Smashing Pumpkins (HD)</inthub:pageField>
    <inthub:pageField name="02LT">Updated song title</inthub:pageField>
</inthub:pageContent>
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=pagecontent
3  <inthub:pageContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:inthub="http://www.vizrt.com/integrationshub">
4      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_event_1/secondaryevents/secondary_event_1/pagecontent" />
5      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
   primary_event_1/secondaryevents/secondary_event_1/pagecontent" />
6      <inthub:pageField name="01LT">The Smashing Pumpkins (HD)</inthub:pageF
   ield>
7      <inthub:pageField name="02LT">Updated song title</inthub:pageField>
8  </inthub:pageContent>
```

Copyright © 2024 Vizrt

Templates (Version 2.0)

- GET Template Pool (Version 2.0)
- GET a Template from Template Pool (Version 2.0)

Copyright © 2024 Vizrt

GET Template Pool (Version 2.0)

Get all templates for a channel's template pool.

Syntax

```
GET /api/channels/[channel]/templates
```

Request

```
GET /api/channels/test/templates HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=list <?xml version="1.0" encoding="utf-8"?>
<inthub:list xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:masterTemplateList" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:masterTemplate>
        <inthub:name>1000</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=mastertemplate" href="http://localhost:9991/api/channels/test/templates/1000" />
        <inthub:concept>HD</inthub:concept>
        <inthub:updated>2021-12-10T23:04:53Z</inthub:updated>
        <inthub:importDatetime>2021-12-10T23:04:53.918Z</inthub:importDatetime>
        <inthub:layer>/ChannelBranding/channels/test/templates/sceneselectors/1000</inthub:layer>
    </inthub:masterTemplate>
    <inthub:masterTemplate>
        <inthub:name>4000_c1_c3</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=mastertemplate" href="http://localhost:9991/api/channels/test/templates/4000_c1_c3" />
        <inthub:concept>SD</inthub:concept>
        <inthub:updated>2023-12-15T12:10:15Z</inthub:updated>
        <inthub:importDatetime>2022-06-10T13:01:33.124Z</inthub:importDatetime>
        <inthub:layer>/ChannelBranding/channels/test/templates/sceneselectors/4000_c1_c3</inthub:layer>
    </inthub:masterTemplate>
    <inthub:masterTemplate>
        <inthub:name>4000_C1</inthub:name>
```

Copyright © 2024 Vizrt

```
22          <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=mastertemplate" href="http://
localhost:9991/api/channels/test/templates/4000_C1" />
23          <inthub:concept>IPHONE</inthub:concept>
24          <inthub:updated>2023-12-15T12:10:20Z</inthub:updated>
25          <inthub:importDatetime>2022-06-10T13:05:12.326Z</inthub:importDate
time>
26          <inthub:layer>/ChannelBranding/channels/test/templates/
sceneselectors/4000_C1</inthub:layer>
27      </inthub:masterTemplate>
28  </inthub:list>
```

Disallowed Actions

This end-point does not allow POST, PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

GET a Template from Template Pool (Version 2.0)

Get a template from the templates pool in a channel.

Syntax

```
GET /api/channels/[channel]/templates/[template]
```

Request

```
GET /api/channels/test/templates/4000 HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=mastertemplate
<inthub:masterTemplate xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>4000</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=mastertemplate" href="http://localhost:9991/api/channels/test/templates/4000" />
    <inthub:concept>SD</inthub:concept>
    <inthub:updated>2023-12-12T15:21:06Z</inthub:updated>
    <inthub:importDatetime>2021-12-10T23:04:53.918Z</inthub:importDatetime>
    <inthub:layer>/ChannelBranding/channels/test/templates/sceneselectors/4000</inthub:layer>
</inthub:masterTemplate>
```

Disallowed Actions

This end-point does not allow POST, PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

Pages (Version 2.0)

- GET Page Pool (Version 2.0)
- POST a Page to Page Pool (Version 2.0)
- GET a Page from Page Pool (Version 2.0)
- PUT on Page in Page Pool (Version 2.0)
- DELETE a Page from Page Pool (Version 2.0)
- GET Page Content from a Page in Page Pool (Version 2.0)
- PUT Page Content for a Page in Page Pool (Version 2.0)

Copyright © 2024 Vizrt

GET Page Pool (Version 2.0)

Get all pages in the channel page pool.

Syntax

```
GET /api/channels/[channel]/page_pool
```

Request

```
GET /api/channels/test/page_pool HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=list
<inthub:list xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:pagesList" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:page>
        <inthub:name>6000</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/channels/test/page_pool/6000" />
        <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/channels/test/page_pool/6000" />
        <inthub:refLink rel="pagecontent" type="application/vizrt.integrationshubv2+xml; type=pagecontent" href="http://127.0.0.1:9991/api/channels/test/page_pool/6000/pagecontent" />
        <inthub:template>6000</inthub:template>
        <inthub:description>The Smashing Pumpkins/Bullet With Butterfly Wings</inthub:description>
        <inthub:defaultDuration />
        <inthub:timeToLive />
    </inthub:page>
    <inthub:page>
        <inthub:name>8000_clip</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/channels/test/page_pool/8000_clip" />
        <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/channels/test/page_pool/8000_clip" />
```

Copyright © 2024 Vizrt

```
18          <inthub:refLink rel="pagecontent" type="application/
vizrt.integrationshubv2+xml; type=pagecontent" href="http://
127.0.0.1:9991/api/channels/test/page_pool/8000_clip/pagecontent" />
19          <inthub:template>8000_clip</inthub:template>
20          <inthub:description>Coming Up Next.../Maroon 5 live and
unplugged</inthub:description>
21          <inthub:defaultDuration />
22          <inthub:timeToLive />
23      </inthub:page>
24      <inthub:page>
25          <inthub:name>1000</inthub:name>
26          <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/
channels/test/page_pool/1000" />
27          <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/
channels/test/page_pool/1000" />
28          <inthub:refLink rel="pagecontent" type="application/
vizrt.integrationshubv2+xml; type=pagecontent" href="http://
127.0.0.1:9991/api/channels/test/page_pool/1000/pagecontent" />
29          <inthub:template>1000</inthub:template>
30          <inthub:description>DEFAULT_LOGO</inthub:description>
31          <inthub:defaultDuration />
32          <inthub:timeToLive>2023-12-31</inthub:timeToLive>
33      </inthub:page>
34  </inthub:list>
```

Disallowed Actions

This end-point does not allow PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

POST a Page to Page Pool (Version 2.0)

Add one page to a channel's Page Pool.

Description is set from template, if left empty.

Name and **template** cannot be empty. Attempting to add a page with template does not exist in template pool, will return a '*HTTP 400 Failed to create page, missing template: NNN*'.

Default duration and **time to live** can be empty.

Attempting to add a Page that already exists will return a '*HTTP 409 Can't add page 'NNN', it already exists*'.

Syntax

```
POST /api/channels/[channel]/page_pool
```

Request

```
POST /api/channels/test/page_pool HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=page
<inthub:page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http
://www.vizrt.com/integrationshub">
    <inthub:name>test1</inthub:name>
    <inthub:template>6000</inthub:template>
    <inthub:description>New description</inthub:description>
    <inthub:defaultDuration>00:00:10.000</inthub:defaultDuration>
    <inthub:timeToLive>2023-12-31</inthub:timeToLive>
</inthub:page>
```

Response

```
1  HTTP/1.1 201 Created
2  Content-Type: application/vizrt.integrationshubv2+xml; type=page
3  <inthub:page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:inthub="http://www.vizrt.com/integrationshub">
4      <inthub:name>test1</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/
   channels/test/page_pool/test1" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/
   channels/test/page_pool/test1" />
```

Copyright © 2024 Vizrt

```
    <inthub:refLink rel="pagecontent" type="application/vizrt.integrationshubv2+xml; type=pagecontent" href="http://127.0.0.1:9991/api/channels/test/page_pool/test1/pagecontent" />
    <inthub:template>6000</inthub:template>
    <inthub:description>New description</inthub:description>
    <inthub:defaultDuration>00:00:10.000</inthub:defaultDuration>
    <inthub:timeToLive>2023-12-31</inthub:timeToLive>
</inthub:page>
```

Page With Empty Optional Values

Syntax

```
POST /api/channels/[channel]/page_pool
```

Request

```
POST /api/channels/test/page_pool HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=page
<inthub:page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>test2</inthub:name>
    <inthub:template>6000</inthub:template>
    <inthub:description></inthub:description>
    <inthub:defaultDuration></inthub:defaultDuration>
    <inthub:timeToLive></inthub:timeToLive>
</inthub:page>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=page
<inthub:page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>test2</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/channels/test/page_pool/test2" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/channels/test/page_pool/test2" />
    <inthub:refLink rel="pagecontent" type="application/vizrt.integrationshubv2+xml; type=pagecontent" href="http://127.0.0.1:9991/api/channels/test/page_pool/test2/pagecontent" />
    <inthub:template>6000</inthub:template>
```

Copyright © 2024 Vizrt

```
    <inthub:description>Lower third</inthub:description>
    <inthub:defaultDuration />
    <inthub:timeToLive />
</inthub:page>
```

Copyright © 2024 Vizrt

GET a Page from Page Pool (Version 2.0)

Get a page from the Page Pool of a channel.

Syntax

```
GET /api/channels/[channel]/page_pool/[page]
```

Request

```
GET /api/channels/test/page_pool/6000 HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=page
<inthub:page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>6000</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/
channels/test/page_pool/6000" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/
channels/test/page_pool/6000" />
    <inthub:refLink rel="pagecontent" type="application/
vizrt.integrationshubv2+xml; type=pagecontent" href="http://
127.0.0.1:9991/api/channels/test/page_pool/6000/pagecontent" />
    <inthub:template>6000</inthub:template>
    <inthub:description>The Smashing Pumpkins/Bullet With Butterfly
Wings</inthub:description>
    <inthub:defaultDuration>00:00:10.000</inthub:defaultDuration>
    <inthub:timeToLive>2023-12-31</inthub:timeToLive>
</inthub:page>
```

Disallowed Actions

This end-point does not allow POST, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

PUT on Page in Page Pool (Version 2.0)

Update a page in the Page Pool of a channel.

Name and **template** can't be adjusted from it's original value.

Description, **default duration** and **time to live** can be empty.

Syntax

```
PUT /api/channels/[channel]/page_pool/[page]
```

Request

```
PUT /api/channels/test/page_pool/8000 HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=page
<inthub:page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http
://www.vizrt.com/integrationshub">
    <inthub:name>8000</inthub:name>
    <inthub:template>8000</inthub:template>
    <inthub:description>Adjusted description.</inthub:description>
    <inthub:defaultDuration>00:00:11.000</inthub:defaultDuration>
    <inthub:timeToLive>2023-11-30</inthub:timeToLive>
</inthub:page>
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=page
3  <inthub:page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:inthub="http://www.vizrt.com/integrationshub">
4      <inthub:name>8000</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/
   channels/test/page_pool/8000" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=page" href="http://127.0.0.1:9991/api/
   channels/test/page_pool/8000" />
7      <inthub:refLink rel="pagecontent" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/page_pool/8000/pagecontent" />
8      <inthub:template>8000</inthub:template>
9      <inthub:description>Adjusted description.</inthub:description>
10     <inthub:defaultDuration>00:00:11.000</inthub:defaultDuration>
11     <inthub:timeToLive>2023-11-30</inthub:timeToLive>
12 </inthub:page>
```

Copyright © 2024 Vizrt

DELETE a Page from Page Pool (Version 2.0)

Delete a Page from the Channel Page Pool

Syntax

```
DELETE /api/channels/[channel]/page_pool/[page]
```

Request

```
DELETE /api/channels/test/page_pool/6000 HTTP/1.1
```

Response

```
1 HTTP/1.1 204 No Content
```

Note: If trying to delete a non-existing Page the returned error-code will be *HTTP 404*, with a reference to relevant info used in the request-uri.

Copyright © 2024 Vizrt

GET Page Content from a Page in Page Pool (Version 2.0)

Following the downlink for a page in page pool to get page content.

Syntax

```
GET /api/channels/[channel]/page_pool/[page]/pagecontent
```

Request

```
GET /api/channels/test/page_pool/6000/pagecontent HTTP/1.1
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=pagecontent
3  <inthub:pageContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:inthub="http://www.vizrt.com/integrationshub">
4      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/page_pool/6000/pagecontent" />
5      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/page_pool/6000/pagecontent" />
6      <inthub:pageField name="01LT">The Smashing Pumpkins (HD)</inthub:pageF
   ield>
7      <inthub:pageField name="02LT">The Song Name (HD)</inthub:pageField>
8  </inthub:pageContent>
```

Disallowed Actions

This end-point does not allow POST or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

PUT Page Content for a Page in Page Pool (Version 2.0)

Update page content of a page.

Page field values are adjustable (empty value allowed), but the name of the field must remain unchanged. The full list of page fields needs to be present in the request, as page fields can't be added or deleted.

Syntax

```
PUT /api/channels/[channel]/page_pool/[page]/pagecontent
```

Request

```
PUT /api/channels/test/page_pool/6000/pagecontent HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=pagecontent
<inthub:pageContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:pageField name="01LT">The Smashing Pumpkins (HD)</inthub:pageField>
    <inthub:pageField name="02LT">Updated song title</inthub:pageField>
</inthub:pageContent>
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=pagecontent
3  <inthub:pageContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:inthub="http://www.vizrt.com/integrationshub">
4      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   localhost:9991/api/channels/test/page_pool/6000/pagecontent" />
5      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   localhost:9991/api/channels/test/page_pool/6000/pagecontent" />
6      <inthub:pageField name="01LT">The Smashing Pumpkins (HD)</inthub:pageF
   ield>
7      <inthub:pageField name="02LT">Updated song title</inthub:pageField>
8  </inthub:pageContent>
```

Copyright © 2024 Vizrt

Actions (Version 2.0)

- GET Action Pool (Version 2.0)
- GET an Action from Action Pool (Version 2.0)

Copyright © 2024 Vizrt

GET Action Pool (Version 2.0)

Get all actions in the channel action pool.

Syntax

```
GET /api/channels/[channel]/action_pool
```

Request

```
GET /api/channels/test/action_pool HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=list
<inthub:list xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:actionsList" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:action>
        <inthub:name>viz_continue</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=action" href="http://127.0.0.1:9991/api/channels/test/action_pool/viz_continue" />
        <inthub:description>Continue (Stage)</inthub:description>
    </inthub:action>
    <inthub:action>
        <inthub:name>viz_continue_front</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=action" href="http://127.0.0.1:9991/api/channels/test/action_pool/viz_continue_front" />
        <inthub:description>Continue (Front Layer)</inthub:description>
    </inthub:action>
    <inthub:action>
        <inthub:name>viz_start_animations</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=action" href="http://127.0.0.1:9991/api/channels/test/action_pool/viz_start_animations" />
        <inthub:description>Start Animations (Stage)</inthub:description>
    </inthub:action>
    <inthub:action>
        <inthub:name>viz_show_begins</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=action" href="http://127.0.0.1:9991/api/channels/test/action_pool/viz_show_begins" />
```

Copyright © 2024 Vizrt

```
22          <inthub:description>Show 0.0 (Stage, Back Layer)</inthub:descripti
    on>
23      </inthub:action>
24      <inthub:action>
25          <inthub:name>viz_cleanup</inthub:name>
26          <inthub:refLink rel="self" type="application/
    vizrt.integrationshubv2+xml; type=action" href="http://127.0.0.1:9991/api/
    channels/test/action_pool/viz_cleanup" />
27          <inthub:description>Cleanup</inthub:description>
28      </inthub:action>
29      <inthub:action>
30          <inthub:name>viz_engage_video_bypass</inthub:name>
31          <inthub:refLink rel="self" type="application/
    vizrt.integrationshubv2+xml; type=action" href="http://127.0.0.1:9991/api/
    channels/test/action_pool/viz_engage_video_bypass" />
32          <inthub:description>Engage video bypass</inthub:description>
33      </inthub:action>
34      <inthub:action>
35          <inthub:name>viz_disengage_video_bypass</inthub:name>
36          <inthub:refLink rel="self" type="application/
    vizrt.integrationshubv2+xml; type=action" href="http://127.0.0.1:9991/api/
    channels/test/action_pool/viz_disengage_video_bypass" />
37          <inthub:description>Disengage video bypass</inthub:description>
38      </inthub:action>
39      <inthub:action>
40          <inthub:name>initialize_action</inthub:name>
41          <inthub:refLink rel="self" type="application/
    vizrt.integrationshubv2+xml; type=action" href="http://127.0.0.1:9991/api/
    channels/test/action_pool/initialize_action" />
42          <inthub:description>Initialize Playlist</inthub:description>
43      </inthub:action>
44      <inthub:action>
45          <inthub:name>cleanup_action</inthub:name>
46          <inthub:refLink rel="self" type="application/
    vizrt.integrationshubv2+xml; type=action" href="http://127.0.0.1:9991/api/
    channels/test/action_pool/cleanup_action" />
47          <inthub:description>Cleanup Playlist</inthub:description>
48      </inthub:action>
49  </inthub:list>
```

Disallowed Actions

This end-point does not allow POST, PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

GET an Action from Action Pool (Version 2.0)

Get an action from the Action Pool of a channel.

Syntax

```
GET /api/channels/[channel]/action_pool/[action]
```

Request

```
GET /api/channels/test/action_pool/viz_cleanup HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=action
<inthub:action xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>viz_cleanup</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=action" href="http://127.0.0.1:9991/api/
channels/test/action_pool/viz_cleanup" />
    <inthub:description>Cleanup</inthub:description>
</inthub:action>
```

Disallowed Actions

This end-point does not allow POST, PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

Channel Pool (Version 2.0)

- GET Channel Pool (Version 2.0)
- POST an Event to Channel Pool (Version 2.0)
- GET Events in Channel Pool (Version 2.0)
- GET an Event in Channel Pool (Version 2.0)
- PUT on Event under Channel Pool (Version 2.0)
- DELETE an Event in Channel Pool (Version 2.0)
- GET Page Content from an Event in Channel Pool (Version 2.0)
- PUT Page Content for an Event in Channel Pool (Version 2.0)
- GET Secondary Events under Group Event in Channel Pool (Version 2.0)
- POST a Secondary Event under Group Event in Channel Pool (Version 2.0)
- GET a Secondary Event under Group Event in Channel Pool (Version 2.0)
- PUT on Secondary Event under a Group Event in Channel Pool (Version 2.0)
- DELETE a Secondary Event under a Group Event in Channel Pool (Version 2.0)
- GET Page Content from a Secondary Event under a Group in Channel Pool (Version 2.0)
- PUT Page Content from a Secondary Event under a Group in Channel Pool (Version 2.0)

Copyright © 2024 Vizrt

GET Channel Pool (Version 2.0)

By following the downlink from channel pool under a channel we can retrieve a channel's channel pool.

Syntax

```
GET /api/channels/[channel]/channel_pool
```

Request

```
GET /api/channels/test/channel_pool HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=channelpool
<inthub:channelPool xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>channel_pool</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=channelpool" href="http://127.0.0.1:9991/api/channels/test/channel_pool" />
    <inthub:refLink rel="events" type="application/vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events" />
    <inthub:description>Channel Pool</inthub:description>
    <inthub:loaded>0.30</inthub:loaded>
    <inthub:available>0.50</inthub:available>
</inthub:channelPool>
```

Disallowed Actions

This end-point does not allow POST, PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

POST an Event to Channel Pool (Version 2.0)

Add one Event to the Channel Pool. There channel pool allows both group events and secondary events to be added, see example below.

Attempting to add an event that already exists will return a '*HTTP 409 Can't add event 'NNN', it already exists*' .

Group Event

Description, start time, duration, ignore, isLive, isLocked, trigger id, house id, mark in, mark out, created by and **metadata** can be empty.

Name cannot be empty.

Ignore, isLive, isLocked defaults to "false" if empty.

Created by defaults to "inthub" if empty.

House id should be specified to create a group video event, where the house id is a Viz One video asset. If the house id exists in VizOne, the video element will be correctly created. If the house id does not exist in VizOne, then a placeholder can be created if it was enabled in the Channel Settings, otherwise the group event will be created, with error.

If **mark in** and **mark out** have values when house id is empty, they will be ignored as they can't be set for non-video events.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **error**, the payload will be accepted but the values will be ignored as they can't be set.

Syntax

```
POST /api/channels/[channel]/channel_pool/events
```

Group Event With No Video (House Id Empty)

Request

```
POST /api/channels/test/channel_pool/events HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:
groupEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>group_1</inthub:name>
    <inthub:description>Top Gun</inthub:description>
    <inthub:duration>02:12:00.000</inthub:duration>
    <inthub:triggerId>primary1001</inthub:triggerId>
    <inthub:houseId />
    <inthub:markIn />
    <inthub:markOut />
```

Copyright © 2024 Vizrt

```
    <inthub:createdBy>component</inthub:createdBy>
</inthub:event>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="inthub:groupEvent" xmlns:inthub="http://www.vizrt.com/
integrationshub">
    <inthub:name>group_1</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/group_1" />
    <inthub:refLink rel="secondaryevents" type="application/
vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/group_1/secondaryevents" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/group_1" />
    <inthub:description>Top Gun</inthub:description>
    <inthub:startTime />
    <inthub:duration>02:12:00.000</inthub:duration>
    <inthub:ignore>false</inthub:ignore>
    <inthub:triggerId>primary1001</inthub:triggerId>
    <inthub:houseId />
    <inthub:markIn />
    <inthub:markOut />
    <inthub:isLocked>false</inthub:isLocked>
    <inthub:isLive>false</inthub:isLive>
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:available />
    <inthub:loaded />
    <inthub:infoIntegrationsHub>[A:15:26:43]</inthub:infoIntegrationsHub>
    <inthub:createdBy>component</inthub:createdBy>
    <inthub:error />
    <inthub:metadata />
</inthub:event>
```

Group Event With Video (House Id Set)

Request

```
POST /api/channels/test/channel_pool/events HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:
groupEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
```

Copyright © 2024 Vizrt

```
    <inthub:name>group_1_video</inthub:name>
    <inthub:description>Top Gun Trailer</inthub:description>
    <inthub:duration>00:00:50.000</inthub:duration>
    <inthub:triggerId>primary1001</inthub:triggerId>
    <inthub:houseId>ABCDEFG</inthub:houseId>
    <inthub:markIn>00:00:01.000</inthub:markIn>
    <inthub:markOut>00:00:09.000</inthub:markOut>
    <inthub:createdBy />
</inthub:event>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:groupEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>group_1_video</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1_video" />
    <inthub:refLink rel="secondaryevents" type="application/vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1_video/secondaryevents" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1_video" />
    <inthub:description>Top Gun Trailer</inthub:description>
    <inthub:startTime />
    <inthub:duration>00:00:50.000</inthub:duration>
    <inthub:ignore>false</inthub:ignore>
    <inthub:triggerId>primary1001</inthub:triggerId>
    <inthub:houseId>ABCDEFG</inthub:houseId>
    <inthub:markIn>00:00:01.000</inthub:markIn>
    <inthub:markOut>00:00:09.000</inthub:markOut>
    <inthub:isLocked>false</inthub:isLocked>
    <inthub:isLive>false</inthub:isLive>
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:available />
    <inthub:loaded />
    <inthub:infoIntegrationsHub>[A:15:26:43]</inthub:infoIntegrationsHub>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:error />
    <inthub:metadata />
</inthub:event>
```

Secondary Event Page

Description, duration, trigger id, offset, created by and **metadata** can be empty.

Copyright © 2024 Vizrt

Name and **page id** cannot be empty. Attempting to add a secondary event with page id does not exist in page pool, will return a '*HTTP 400 Can't create page event, page 'NNN' does not exist*'.

Description is set from the page of given page id, if left empty.

Duration is set from the page of the given page id, if left empty.

Created by defaults to "inthub" if empty.

If any values are set for **is manually edited, loaded, available, info integrations hub, info promo server** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/channel_pool/events HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_1</inthub:name>
    <inthub:description>not used</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:event>
```

Response

```
1  HTTP/1.1 201 Created
2  Content-Type: application/vizrt.integrationshubv2+xml; type=event
3  <inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:type="inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/
   integrationshub">
4      <inthub:name>secondary_1_1</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
   channels/test/channel_pool/events/secondary_1_1" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
   channels/test/channel_pool/events/secondary_1_1" />
7      <inthub:refLink rel="pagecontent" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/secondary_1_1/
   pagecontent" />
8      <inthub:description>not used</inthub:description>
9      <inthub:triggerId>735</inthub:triggerId>
10     <inthub:offset>00:00:10.000</inthub:offset>
11     <inthub:takeCount>0</inthub:takeCount>
12     <inthub:createdBy>inthub</inthub:createdBy>
```

Copyright © 2024 Vizrt

```
    <inthub:infoIntegrationsHub>[A:15:56:49]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:infoPromoServer />
    <inthub:metadata />
</inthub:event>
```

Secondary Event Reference Page

Description, duration, trigger id, offset and **created by** can be empty.

Name and **page id** cannot be empty. Attempting to add a reference secondary event with page id does not exist in page pool, will return a '*HTTP 400 Can't create reference page event, page 'NNN' does not exist*'.

Description is set from the page of given page id, if left empty.

Duration is set from the page of the given page id, if left empty.

Created by defaults to "inthub" if empty.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/channel_pool/events HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:
refPageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_ref</inthub:name>
    <inthub:description>Default Logo</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:event>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="inthub:refPageEvent" xmlns:inthub="http://www.vizrt.com/
integrationshub">
    <inthub:name>secondary_1_ref</inthub:name>
```

Copyright © 2024 Vizrt

```
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/secondary_1_ref" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/secondary_1_ref" />
    <inthub:description>Default Logo</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:takeCount />
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[A:16:10:05]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
</inthub:event>
```

Secondary Event Action

Description, trigger id, offset and created by can be empty.

Name and **action id** cannot be empty. Attempting to add a secondary action event with action id does not exist in action pool, will return a '*Can't create action event, action 'NNN' does not exist*'.

Description is set from the action of given action id, if left empty.

Created by defaults to "inthub" if empty.

If any values are set for **is manually edited, loaded, available, info integrations hub, take count, action command** and **action category**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/channel_pool/events HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:actionEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_action</inthub:name>
    <inthub:description>A cleanup action</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:actionId>viz_cleanup</inthub:actionId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:event>
```

Copyright © 2024 Vizrt

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="inthub:actionEvent" xmlns:inthub="http://www.vizrt.com/
integrationshub">
    <inthub:name>secondary_1_action</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/secondary_1_action" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/secondary_1_action" />
    <inthub:description>A cleanup action</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:takeCount />
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[A:16:24:12]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:actionId>viz_cleanup</inthub:actionId>
    <inthub:actionCommand>SCENE CLEANUP</inthub:actionCommand>
    <inthub:actionCategory>viz</inthub:actionCategory>
</inthub:event>
```

Secondary Event Pilot Data Element

Description, duration, trigger id, offset and **created by** can be empty.

Name and **pilot data element id** cannot be empty. **Pilot data element id** must be an integer.

Created by defaults to "inthub" if empty.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/channel_pool/events HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:
pilotDataElementEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_pde</inthub:name>
    <inthub:description>Pilot data element</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
```

Copyright © 2024 Vizrt

```
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:event>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="inthub:pilotDataElementEvent" xmlns:inthub="http://
www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_pde</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/secondary_1_pde" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/secondary_1_pde" />
    <inthub:description>Pilot data element</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:takeCount />
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[A:16:16:30]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
</inthub:event>
```

Secondary Event Message

Description, offset and **created by** can be empty.

Name cannot be empty.

Created by defaults to "inthub" if empty.

If any values are set for **info integrations hub**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/channel_pool/events HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:
msgEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_message</inthub:name>
```

Copyright © 2024 Vizrt

```
    <inthub:description>This is a message</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:createdBy>schedule</inthub:createdBy>
</inthub:event>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="inthub:msgEvent" xmlns:inthub="http://www.vizrt.com/
integrationshub">
    <inthub:name>secondary_1_message</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/secondary_1_message" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/secondary_1_message" />
    <inthub:description>This is a message</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:createdBy>schedule</inthub:createdBy>
    <inthub:infoIntegrationsHub>[A:14:55:14]</inthub:infoIntegrationsHub>
</inthub:event>
```

Note: The full contract can be used for POST, fields that are read only will be ignored.

Copyright © 2024 Vizrt

GET Events in Channel Pool (Version 2.0)

By following the downlink from the channel pool we can retrieve a collection of events. The channel pool can have both secondary events and group events at the root level.

Take note that the collection of events contains a compact version of them. Use the downlink for each event to get details.

Syntax

```
GET /api/channels/[channel]/channel_pool/events
```

Request

```
GET /api/channels/test/channel_pool/events HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=list
<inthub:list xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:eventsCompact" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:eventCompact xsi:type="inthub:groupEventCompact">
        <inthub:name>primary_element_1_0_0</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/primary_element_1_0_0" />
        <inthub:refLink rel="secondaryevents" type="application/vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/primary_element_1_0_0/secondaryevents" />
        <inthub:description>Le radeau LE CRASH DU VOL 29</inthub:description>
        <inthub:startTime />
        <inthub:duration>08:58:13.000</inthub:duration>
        <inthub:ignore>false</inthub:ignore>
        <inthub:triggerId />
        <inthub:houseId />
        <inthub:isLocked>false</inthub:isLocked>
        <inthub:isLive>false</inthub:isLive>
        <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
        <inthub:available>1.00</inthub:available>
        <inthub:loaded>1.00</inthub:loaded>
        <inthub:infoIntegrationsHub>[A:11:47:44]</inthub:infoIntegrationsHub>
```

Copyright © 2024 Vizrt

```
20          <inthub:createdBy>schedule</inthub:createdBy>
21          <inthub:error />
22      </inthub:eventCompact>
23      <inthub:eventCompact xsi:type="inthub:pageEventCompact">
24          <inthub:name>f4e0a087-c6f4-4f40-bca9-a51cdaf43634</inthub:name>
25          <inthub:refLink rel="self" type="application/
    vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
    channels/test/channel_pool/events/f4e0a087-c6f4-4f40-bca9-a51cdaf43634" />
26          <inthub:description>The Smashing Pumpkins/Bullet With Butterfly
    Wings</inthub:description>
27          <inthub:triggerId />
28          <inthub:offset>00:00:02.000</inthub:offset>
29          <inthub:takeCount>0</inthub:takeCount>
30          <inthub:createdBy>manual</inthub:createdBy>
31          <inthub:infoIntegrationsHub />
32          <inthub:loaded>1.00</inthub:loaded>
33          <inthub:available>1.00</inthub:available>
34          <inthub:isManuallyEdited>true</inthub:isManuallyEdited>
35          <inthub:duration>00:00:10.000</inthub:duration>
36          <inthub:pageId>6000</inthub:pageId>
37          <inthub:infoPromoServer>[14:08:30]</inthub:infoPromoServer>
38      </inthub:eventCompact>
39      <inthub:eventCompact xsi:type="inthub:refPageEventCompact">
40          <inthub:name>1000</inthub:name>
41          <inthub:refLink rel="self" type="application/
    vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
    channels/test/channel_pool/events/1000" />
42          <inthub:description>DEFAULT_LOGO</inthub:description>
43          <inthub:triggerId />
44          <inthub:offset>00:00:00.000</inthub:offset>
45          <inthub:takeCount>0</inthub:takeCount>
46          <inthub:createdBy>schedule</inthub:createdBy>
47          <inthub:infoIntegrationsHub>[A:14:07:54]</inthub:infoIntegrationsH
    ub>
48          <inthub:loaded>1.00</inthub:loaded>
49          <inthub:available>1.00</inthub:available>
50          <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
51          <inthub:duration>00:09:07.890</inthub:duration>
52          <inthub:pageId>1000</inthub:pageId>
53      </inthub:eventCompact>
54      <inthub:eventCompact xsi:type="inthub:pilotDataElementEventCompact">
55          <inthub:name>9000</inthub:name>
56          <inthub:refLink rel="self" type="application/
    vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
    channels/test/channel_pool/events/9000" />
57          <inthub:description />
58          <inthub:triggerId />
59          <inthub:offset>00:00:00.000</inthub:offset>
60          <inthub:takeCount />
61          <inthub:createdBy>schedule</inthub:createdBy>
62          <inthub:infoIntegrationsHub>[A:14:08:28]</inthub:infoIntegrationsH
    ub>
63          <inthub:loaded />
```

Copyright © 2024 Vizrt

```
64          <inthub:available />
65          <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
66          <inthub:duration>00:09:07.890</inthub:duration>
67          <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
68      </inthub:eventCompact>
69      <inthub:eventCompact xsi:type="inthub:actionEventCompact">
70          <inthub:name>1c720ee7-8481-4d00-aa53-fe577b85bcbb</inthub:name>
71          <inthub:refLink rel="self" type="application/
    vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
    channels/test/channel_pool/events/1c720ee7-8481-4d00-aa53-fe577b85bcbb" />
72          <inthub:description>Cleanup</inthub:description>
73          <inthub:triggerId />
74          <inthub:offset />
75          <inthub:takeCount>0</inthub:takeCount>
76          <inthub:createdBy>manual</inthub:createdBy>
77          <inthub:infoIntegrationsHub />
78          <inthub:loaded>1.00</inthub:loaded>
79          <inthub:available>1.00</inthub:available>
80          <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
81          <inthub:actionId>viz_cleanup</inthub:actionId>
82          <inthub:actionCommand>SCENE CLEANUP</inthub:actionCommand>
83          <inthub:actionCategory>viz</inthub:actionCategory>
84      </inthub:eventCompact>
85      <inthub:eventCompact xsi:type="inthub:msgEventCompact">
86          <inthub:name>secondary_msg_1</inthub:name>
87          <inthub:refLink rel="self" type="application/
    vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
    channels/test/channel_pool/events/secondary_msg_1" />
88          <inthub:description>Element with Id '' could not be identified</
    inthub:description>
89          <inthub:offset>00:00:10.000</inthub:offset>
90          <inthub:createdBy>schedule</inthub:createdBy>
91          <inthub:infoIntegrationsHub>[A:14:55:14]</inthub:infoIntegrationsH
    ub>
92      </inthub:eventCompact>
93  </inthub:list>
```

Disallowed Actions

This end-point does not allow PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

GET an Event in Channel Pool (Version 2.0)

Following the downlink while retrieving the collection of Events under channel pool one can also do a GET directly on a single Event.

Take note that the content depends on the type of Event, which can be of type Group Event or any type of Secondary Event.

Syntax

```
GET /api/channels/[channel]/channel_pool/events/[event]
```

Request

```
GET /api/channels/test/channel_pool/events/element_1_0_0 HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:groupEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>element_1_0_0</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/element_1_0_0" />
    <inthub:refLink rel="secondaryevents" type="application/vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/element_1_0_0/secondaryevents" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/element_1_0_0" />
    <inthub:description>Le radeau LE CRASH DU VOL 29</inthub:description>
    <inthub:startTime>2024-01-17T04:00:00.000Z</inthub:startTime>
    <inthub:duration>00:10:00.000</inthub:duration>
    <inthub:ignore>false</inthub:ignore>
    <inthub:triggerId />
    <inthub:houseId />
    <inthub:markIn />
    <inthub:markOut />
    <inthub:isLocked>false</inthub:isLocked>
    <inthub:isLive>false</inthub:isLive>
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:available>1.00</inthub:available>
```

Copyright © 2024 Vizrt

```
    <inthub:loaded>1.00</inthub:loaded>
    <inthub:infoIntegrationsHub />
    <inthub:createdBy>manual</inthub:createdBy>
    <inthub:error />
    <inthub:metadata />
</inthub:event>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>element_2_0_0</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/element_2_0_0" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/element_2_0_0" />
    <inthub:refLink rel="pagecontent" type="application/vizrt.integrationshubv2+xml; type=pagecontent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/element_2_0_0/pagecontent" />
    <inthub:description>The Smashing Pumpkins/Bullet With Butterfly Wings</inthub:description>
    <inthub:triggerId>element_2_0_0</inthub:triggerId>
    <inthub:offset />
    <inthub:takeCount />
    <inthub:createdBy>manual</inthub:createdBy>
    <inthub:infoIntegrationsHub />
    <inthub:loaded>1.00</inthub:loaded>
    <inthub:available>1.00</inthub:available>
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration />
    <inthub:pageId>page0</inthub:pageId>
    <inthub:infoPromoServer />
    <inthub:metadata />
</inthub:event>
```

Disallowed Actions

This end-point does not allow POST, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

PUT on Event under Channel Pool (Version 2.0)

Update an Event definition under Channel Pool. Can be group event or several types of secondary events, examples below.

Syntax

```
PUT /api/channels/[channel]/channel_pool/events/[event]
```

Group Event

Name, house id and **created by** can't be adjusted from it's original value.

Description, startTime, duration, ignore, trigger id, mark in, mark out, isLive, isLocked and **metadata** can be empty.

Ignore, isLive and **isLocked** defaults to "false" if empty.

If **mark in** and **mark out** have values when house id is empty, they will be ignored as they can't be set for non-video events.

Metadata section will update completely according to how it is specified. If the metadata section is empty, all metadata will be removed. If a metadata attribute is removed, then that metadata will be removed. If metadata values are updated, they will be updated accordingly.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **error**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/channel_pool/events/group_1 HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:
groupEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>group_1</inthub:name>
    <inthub:description>New description</inthub:description>
    <inthub:startTime>2018-11-06T07:13:40.000Z</inthub:startTime>
    <inthub:duration>02:12:00.000</inthub:duration>
    <inthub:ignore>False</inthub:ignore>
    <inthub:triggerId>primary1001</inthub:triggerId>
    <inthub:createdBy>schedule</inthub:createdBy>
    <inthub:houseId />
    <inthub:markIn />
    <inthub:markOut />
    <inthub:isLocked>false</inthub:isLocked>
    <inthub:isLive>false</inthub:isLive>
    <inthub:metadata />
</inthub:event>
```

Copyright © 2024 Vizrt

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:groupEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>group_1</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1" />
    <inthub:refLink rel="secondaryevents" type="application/vizrt.integrationshubv2+xml; type=list" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/secondaryevents" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1" />
    <inthub:description>New description</inthub:description>
    <inthub:startTime>2018-11-06T07:13:40.000Z</inthub:startTime>
    <inthub:duration>02:12:00.000</inthub:duration>
    <inthub:ignore>false</inthub:ignore>
    <inthub:triggerId>primary1001</inthub:triggerId>
    <inthub:houseId />
    <inthub:markIn />
    <inthub:markOut />
    <inthub:isLocked>false</inthub:isLocked>
    <inthub:isLive>false</inthub:isLive>
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:available />
    <inthub:loaded />
    <inthub:infoIntegrationsHub>[U:16:23:40]</inthub:infoIntegrationsHub>
    <inthub:createdBy>schedule</inthub:createdBy>
    <inthub:error />
    <inthub:metadata />
</inthub:event>
```

Secondary Event Page

Name, page id and **created by** can't be adjusted from it's original value.

Description, duration, trigger id, offset and **metadata** can be empty.

Metadata section will update completely according to how it is specified. If the metadata section is empty, all metadata will be removed. If a metadata attribute is removed, then that metadata will be removed. If metadata values are updated, they will be updated accordingly.

If any values are set for **is manually edited, loaded, available, info integrations hub, info promo server** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Copyright © 2024 Vizrt

Request

```
PUT /api/channels/test/channel_pool/events/secondary_1_page HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:
pageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_page</inthub:name>
    <inthub:description>new description</inthub:description>
    <inthub:offset>00:00:05.000</inthub:offset>
    <inthub:duration>00:00:01.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:metadata />
</inthub:event>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/
integrationshub">
    <inthub:name>secondary_1_page</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/secondary_1_page" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/secondary_1_page" />
    <inthub:refLink rel="pagecontent" type="application/
vizrt.integrationshubv2+xml; type=pagecontent" href="http://
127.0.0.1:9991/api/channels/test/channel_pool/events/secondary_1_page/
pagecontent" />
    <inthub:description>new description</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:05.000</inthub:offset>
    <inthub:takeCount>0</inthub:takeCount>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[U:17:15:15]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:01.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:infoPromoServer />
    <inthub:metadata />
</inthub:event>
```

Copyright © 2024 Vizrt

Secondary Event Reference Page

Name, page id and **created by** can't be adjusted from it's original value.

Description, duration, trigger id and **offset** can be empty.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/channel_pool/events/secondary_1_ref HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:
refPageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_ref</inthub:name>
    <inthub:description>New Logo</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:event>
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=event
3  <inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:type="inthub:refPageEvent" xmlns:inthub="http://www.vizrt.com/
   integrationshub">
4      <inthub:name>secondary_1_ref</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
   channels/test/channel_pool/events/secondary_1_ref" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
   channels/test/channel_pool/events/secondary_1_ref" />
7      <inthub:description>New Logo</inthub:description>
8      <inthub:triggerId>735</inthub:triggerId>
9      <inthub:offset>00:00:10.000</inthub:offset>
10     <inthub:takeCount />
11     <inthub:createdBy>inthub</inthub:createdBy>
12     <inthub:infoIntegrationsHub>[U:17:10:05]</inthub:infoIntegrationsHub>
13     <inthub:loaded />
14     <inthub:available />
15     <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
16     <inthub:duration>00:00:20.000</inthub:duration>
17     <inthub:pageId>6000</inthub:pageId>
```

Copyright © 2024 Vizrt

```
</inthub:event>
```

Secondary Event Action

Name, action id and **created by** can't be adjusted from it's original value.

Description, trigger id, offset and **action command** can be empty.

If any values are set for **is manually edited, loaded, available, info integrations hub, take count** and **action category**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/channel_pool/events/secondary_1_action HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:actionEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_action</inthub:name>
    <inthub:description>A cleanup action</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:actionId>viz_cleanup</inthub:actionId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:actionCommand>SCENE CLEANUP 1</inthub:actionCommand>
</inthub:event>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:actionEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_action</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/secondary_1_action" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/secondary_1_action" />
    <inthub:description>A cleanup action</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:takeCount />
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[U:16:34:12]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
```

Copyright © 2024 Vizrt

```
    <inthub:actionId>viz_cleanup</inthub:actionId>
    <inthub:actionCommand>SCENE CLEANUP 1</inthub:actionCommand>
    <inthub:actionCategory>viz</inthub:actionCategory>
</inthub:event>
```

Secondary Event Pilot Data Element

Name, pilot data element id and **created by** can't be adjusted from it's original value.

Description, duration, trigger id and **offset** can be empty.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/channel_pool/events/secondary_1_pde HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:
pilotDataElementEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_pde</inthub:name>
    <inthub:description>Pilot data element</inthub:description>
    <inthub:offset>00:00:12.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:event>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="inthub:pilotDataElementEvent" xmlns:inthub="http://
www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_pde</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/secondary_1_pde" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/
channels/test/channel_pool/events/secondary_1_pde" />
    <inthub:description>Pilot data element</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:12.000</inthub:offset>
    <inthub:takeCount />
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[U:21:10:30]</inthub:infoIntegrationsHub>
```

Copyright © 2024 Vizrt

```
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
</inthub:event>
```

Secondary Event Message

Name and **created by** can't be adjusted from it's original value.

Description and **offset** can be empty.

If any values are set for **info integrations hub**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/channel_pool/events/secondary_1_message HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:msgEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_message</inthub:name>
    <inthub:description>This is an updated message</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:createdBy>schedule</inthub:createdBy>
</inthub:event>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=event
<inthub:event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:msgEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_message</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/secondary_1_message" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=event" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/secondary_1_message" />
    <inthub:description>This is an updated message</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:createdBy>schedule</inthub:createdBy>
    <inthub:infoIntegrationsHub>[U:14:57:04]</inthub:infoIntegrationsHub>
</inthub:event>
```

Copyright © 2024 Vizrt

DELETE an Event in Channel Pool (Version 2.0)

Delete an Event from channel pool.

Syntax

```
DELETE /api/channels/[channel]/channel_pool/events/[event]
```

Request

```
DELETE /api/channels/test/channel_pool/events/event_1 HTTP/1.1
```

Response

```
1 HTTP/1.1 204 No Content
```

Note: If trying to delete a non-existing Event the returned error-code will be *HTTP 404*, with a reference to relevant info used in the request-uri.

Copyright © 2024 Vizrt

GET Page Content from an Event in Channel Pool (Version 2.0)

Following the downlink for an event of type secondary event in channel pool to get page content.

Syntax

```
GET /api/channels/[channel]/channel_pool/events/[event]/pagecontent
```

Request

```
GET /api/channels/test/channel_pool/events/secondary_event_1/pagecontent HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=pagecontent
<inthub:pageContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=pagecontent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/secondary_event_1/pagecontent" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=pagecontent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/secondary_event_1/pagecontent" />
    <inthub:pageField name="01butterfly">UP NEXT...</inthub:pageField>
    <inthub:pageField name="02butterfly">CELEBRITY DEATH OCTAGON</inthub:pageField>
    <inthub:pageField name="03butterfly">Watch Lady Gaga go up against Marilyn Manson in the battle for subliminal supremacy. With guest referee Ozzy Osbourne.</inthub:pageField>
    <inthub:pageField name="Videoclip"></inthub:pageField>
</inthub:pageContent>
```

Disallowed Actions

This end-point does not allow POST or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

PUT Page Content for an Event in Channel Pool (Version 2.0)

Update page content of event of type secondary event page.

Page field values are adjustable (empty value allowed), but the name of the field must remain unchanged. The full list of page fields needs to be present in the request, as page fields can't be added or deleted.

Syntax

```
PUT /api/channels/[channel]/channel_pool/events/[event]/pagecontent
```

Request

```
PUT /api/channels/test/channel_pool/events/secondary_1_page/pagecontent HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=pagecontent
<inthub:pageContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:pageField name="01LT">The Smashing Pumpkins (HD)</inthub:pageField>
    <inthub:pageField name="02LT">Updated song title</inthub:pageField>
</inthub:pageContent>
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=pagecontent
3  <inthub:pageContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:inthub="http://www.vizrt.com/integrationshub">
4      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/secondary_1_page/
   pagecontent" />
5      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/secondary_1_page/
   pagecontent" />
6      <inthub:pageField name="01LT">The Smashing Pumpkins (HD)</inthub:pageF
   ield>
7      <inthub:pageField name="02LT">Updated song title</inthub:pageField>
8  </inthub:pageContent>
```

Copyright © 2024 Vizrt

GET Secondary Events under Group Event in Channel Pool (Version 2.0)

Follow the downlink under a Group Event to get a collection of secondary events under this Group Event.

Take note that the collection of secondary events contains a compact version of a Secondary Event, and also that the content is depending on the type of Secondary Event. Use the downlink for each Secondary Event to get details.

Syntax

```
GET /api/channels/[channel]/channel_pool/events/[group_event]/secondaryevents
```

Request

```
GET /api/channels/test/channel_pool/events/group_element_1_0_0/secondaryevents HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=list
<inthub:list xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:secondaryEventsCompact" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:secondaryEventCompact xsi:type="inthub:pageEventCompact">
        <inthub:name>f4e0a087-c6f4-4f40-bca9-a51cdaf43634</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_element_1_0_0/secondaryevents/f4e0a087-c6f4-4f40-bca9-a51cdaf43634" />
        <inthub:description>The Smashing Pumpkins/Bullet With Butterfly Wings</inthub:description>
        <inthub:triggerId />
        <inthub:offset>00:00:02.000</inthub:offset>
        <inthub:takeCount>0</inthub:takeCount>
        <inthub:createdBy>manual</inthub:createdBy>
        <inthub:infoIntegrationsHub />
        <inthub:loaded>1.00</inthub:loaded>
        <inthub:available>1.00</inthub:available>
        <inthub:isManuallyEdited>true</inthub:isManuallyEdited>
        <inthub:duration>00:00:10.000</inthub:duration>
        <inthub:pageId>6000</inthub:pageId>
        <inthub:infoPromoServer>[14:08:30]</inthub:infoPromoServer>
    </inthub:secondaryEventCompact>
    <inthub:secondaryEventCompact xsi:type="inthub:refPageEventCompact">
```

Copyright © 2024 Vizrt

```
        <inthub:name>1000</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_element_1_0_0/secondaryevents/1000" />
        <inthub:description>DEFAULT_LOGO</inthub:description>
        <inthub:triggerId />
        <inthub:offset>00:00:00.000</inthub:offset>
        <inthub:takeCount>0</inthub:takeCount>
        <inthub:createdBy>schedule</inthub:createdBy>
        <inthub:infoIntegrationsHub>[A:14:07:54]</inthub:infoIntegrationsHub>
        <inthub:loaded>1.00</inthub:loaded>
        <inthub:available>1.00</inthub:available>
        <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
        <inthub:duration>00:09:07.890</inthub:duration>
        <inthub:pageId>1000</inthub:pageId>
    </inthub:secondaryEventCompact>
    <inthub:secondaryEventCompact xsi:type="inthub:pilotDataElementEventCompact">
        <inthub:name>9000</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_element_1_0_0/secondaryevents/9000" />
        <inthub:description />
        <inthub:triggerId />
        <inthub:offset>00:00:00.000</inthub:offset>
        <inthub:takeCount />
        <inthub:createdBy>schedule</inthub:createdBy>
        <inthub:infoIntegrationsHub>[A:14:08:28]</inthub:infoIntegrationsHub>
        <inthub:loaded />
        <inthub:available />
        <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
        <inthub:duration>00:09:07.890</inthub:duration>
        <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
    </inthub:secondaryEventCompact>
    <inthub:secondaryEventCompact xsi:type="inthub:actionEventCompact">
        <inthub:name>1c720ee7-8481-4d00-aa53-fe577b85bcbb</inthub:name>
        <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_element_1_0_0/secondaryevents/1c720ee7-8481-4d00-aa53-fe577b85bcbb" />
        <inthub:description>Cleanup</inthub:description>
        <inthub:triggerId />
        <inthub:offset />
        <inthub:takeCount>0</inthub:takeCount>
        <inthub:createdBy>manual</inthub:createdBy>
        <inthub:infoIntegrationsHub />
        <inthub:loaded>1.00</inthub:loaded>
        <inthub:available>1.00</inthub:available>
        <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
```

Copyright © 2024 Vizrt

```
62          <inthub:actionId>viz_cleanup</inthub:actionId>
63          <inthub:actionCommand>SCENE CLEANUP</inthub:actionCommand>
64          <inthub:actionCategory>viz</inthub:actionCategory>
65      </inthub:secondaryEventCompact>
66      <inthub:secondaryEventCompact xsi:type="inthub:msgEventCompact">
67          <inthub:name>secondary_msg_1</inthub:name>
68          <inthub:refLink rel="self" type="application/
    vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
    127.0.0.1:9991/api/channels/test/channel_pool/events/group_element_1_0_0/
    secondaryevents/secondary_msg_1" />
69          <inthub:description>Element with Id '' could not be identified</
    inthub:description>
70          <inthub:offset>00:00:10.000</inthub:offset>
71          <inthub:createdBy>schedule</inthub:createdBy>
72          <inthub:infoIntegrationsHub>[A:14:55:14]</inthub:infoIntegrationsH
    ub>
73      </inthub:secondaryEventCompact>
74  </inthub:list>
```

Disallowed Actions

This end-point does not allow PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

POST a Secondary Event under Group Event in Channel Pool (Version 2.0)

Add one secondary event to an existing group event. There are several types of secondary events, examples below.

Attempting to add a secondary event that already exists, will return a '*HTTP 409 Can't add secondary event 'NNN', it already exists*'.

Syntax

```
POST /api/channels/[channel]/channel_pool/events/[group_event]/secondaryevents
```

Secondary Event Page

Description, duration, trigger id, offset, created by and **metadata** can be empty.

Name and **page id** cannot be empty. Attempting to add a secondary event with page id does not exist in page pool, will return a '*HTTP 400 Can't create secondary page event, page 'NNN' does not exist*'.

Description is set from the page of given page id, if left empty.

Duration is set from the page of the given page id, if left empty.

Created by defaults to "inthub" if empty.

If any values are set for **is manually edited, loaded, available, info integrations hub, info promo server** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/channel_pool/events/group_1/secondaryevents HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_1</inthub:name>
    <inthub:description>not used</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
1  HTTP/1.1 201 Created
2  Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
```

Copyright © 2024 Vizrt

```
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_1</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_1" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_1" />
    <inthub:refLink rel="pagecontent" type="application/vizrt.integrationshubv2+xml; type=pagecontent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_1/pagecontent" />
    <inthub:description>not used</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:takeCount>0</inthub:takeCount>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[A:15:56:49]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:infoPromoServer />
    <inthub:metadata />
</inthub:secondaryEvent>
```

Secondary Event Reference Page

Description, duration, trigger id, offset and **created by** can be empty.

Name and **page id** cannot be empty. Attempting to add a reference secondary event with page id does not exist in page pool, will return a '*HTTP 400 Can't create reference secondary page event, page 'NNN' does not exist*'.

Description is set from the page of given page id, if left empty.

Duration is set from the page of the given page id, if left empty.

Created by defaults to "inthub" if empty.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/channel_pool/events/group_1/secondaryevents HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
```

Copyright © 2024 Vizrt

```
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:refPageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_ref</inthub:name>
    <inthub:description>Default Logo</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="inthub:refPageEvent" xmlns:inthub="http://
www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_ref</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
secondaryevents/secondary_1_ref" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
secondaryevents/secondary_1_ref" />
    <inthub:description>Default Logo</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:takeCount />
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[A:16:10:05]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
</inthub:secondaryEvent>
```

Secondary Event Action

Description, trigger id, offset and created by can be empty.

Name and **action id** cannot be empty. Attempting to add a secondary action event with action id does not exist in action pool, will return a '*Can't create secondary action event, action 'NNN' does not exist*'.

Description is set from the action of given action id, if left empty.

Created by defaults to "inthub" if empty.

Copyright © 2024 Vizrt

If any values are set for **is manually edited, loaded, available, info integrations hub, take count, action command** and **action category**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/channel_pool/events/group_1/secondaryevents HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:actionEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_action</inthub:name>
    <inthub:description>A cleanup action</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:actionId>viz_cleanup</inthub:actionId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
1  HTTP/1.1 201 Created
2  Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
3  <inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance" xsi:type="inthub:actionEvent" xmlns:inthub="http://
   www.vizrt.com/integrationshub">
4      <inthub:name>secondary_1_action</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_action" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_action" />
7      <inthub:description>A cleanup action</inthub:description>
8      <inthub:triggerId>735</inthub:triggerId>
9      <inthub:offset>00:00:10.000</inthub:offset>
10     <inthub:takeCount />
11     <inthub:createdBy>inthub</inthub:createdBy>
12     <inthub:infoIntegrationsHub>[A:16:24:12]</inthub:infoIntegrationsHub>
13     <inthub:loaded />
14     <inthub:available />
15     <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
16     <inthub:actionId>viz_cleanup</inthub:actionId>
17     <inthub:actionCommand>SCENE CLEANUP</inthub:actionCommand>
18     <inthub:actionCategory>viz</inthub:actionCategory>
19 </inthub:secondaryEvent>
```

Copyright © 2024 Vizrt

Secondary Event Pilot Data Element

Description, duration, trigger id, offset and **created by** can be empty.

Name and **pilot data element id** cannot be empty. **Pilot data element id** must be an integer.

Created by defaults to "inthub" if empty.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/channel_pool/events/group_1/secondaryevents HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:pilotDataElementEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_pde</inthub:name>
    <inthub:description>Pilot data element</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
1  HTTP/1.1 201 Created
2  Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
3  <inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance" xsi:type="inthub:pilotDataElementEvent" xmlns:inthub="http://
   www.vizrt.com/integrationshub">
4      <inthub:name>secondary_1_pde</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_pde" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_pde" />
7      <inthub:description>Pilot data element</inthub:description>
8      <inthub:triggerId>735</inthub:triggerId>
9      <inthub:offset>00:00:10.000</inthub:offset>
10     <inthub:takeCount />
11     <inthub:createdBy>inthub</inthub:createdBy>
12     <inthub:infoIntegrationsHub>[A:16:16:30]</inthub:infoIntegrationsHub>
13     <inthub:loaded />
14     <inthub:available />
```

Copyright © 2024 Vizrt

```
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
</inthub:secondaryEvent>
```

Secondary Event Message

Description, offset and **created by** can be empty.

Name cannot be empty.

Created by defaults to "inthub" if empty.

If any values are set for **info integrations hub**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
POST /api/channels/test/channel_pool/events/group_1/secondaryevents HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:msgEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_message</inthub:name>
    <inthub:description>This is a message</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:createdBy>schedule</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
HTTP/1.1 201 Created
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent

<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="inthub:msgEvent" xmlns:inthub="http://www.vizrt.com/
integrationshub">
    <inthub:name>secondary_1_message</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
secondaryevents/secondary_1_message" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
secondaryevents/secondary_1_message" />
    <inthub:description>This is a message</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:createdBy>schedule</inthub:createdBy>
    <inthub:infoIntegrationsHub>[A:14:55:14]</inthub:infoIntegrationsHub>
```

Copyright © 2024 Vizrt

```
11  </inthub:secondaryEvent>
```

Note: The full contract can be used for POST, fields that are read only will be ignored.

Copyright © 2024 Vizrt

GET a Secondary Event under Group Event in Channel Pool (Version 2.0)

Following the downlink while retrieving the collection of Secondary Events under a Group Event one can also do a GET directly on a single Secondary Event.

Take note that the content depends on the type of Secondary Event.

Syntax

```
GET /api/channels/[channel]/channel_pool/events/[group_event]/secondaryevents/
[secondary_event]
```

Request

```
GET /api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_1
HTTP/1.1
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
3  <inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance" xsi:type="inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/
   integrationshub">
4      <inthub:name>secondary_1_1</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_1" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_1" />
7      <inthub:refLink rel="pagecontent" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_1/pagecontent" />
8      <inthub:description>The Smashing Pumpkins/Bullet With Butterfly
   Wings</inthub:description>
9      <inthub:triggerId />
10     <inthub:offset>00:00:02.000</inthub:offset>
11     <inthub:takeCount>0</inthub:takeCount>
12     <inthub:createdBy>manual</inthub:createdBy>
13     <inthub:infoIntegrationsHub />
14     <inthub:loaded>1.00</inthub:loaded>
15     <inthub:available>1.00</inthub:available>
```

Copyright © 2024 Vizrt

```
    <inthub:isManuallyEdited>true</inthub:isManuallyEdited>
    <inthub:duration>00:00:10.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:infoPromoServer>[13:40:14]</inthub:infoPromoServer>
    <inthub:metadata />
</inthub:secondaryEvent>
```

Disallowed Actions

This end-point does not allow POST, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

PUT on Secondary Event under a Group Event in Channel Pool (Version 2.0)

Update a Secondary Event definition under a Group Event in Channel Pool. There are several types of secondary events, examples below.

Syntax

```
PUT /api/channels/[channel]/channel_pool/events/[group_event]/secondaryevents/
[secondary_event]
```

Secondary Event Page

Name, page id and **created by** can't be adjusted from it's original value.

Description, duration, trigger id, offset and **metadata** can be empty.

Metadata section will update completely according to how it is specified. If the metadata section is empty, all metadata will be removed. If a metadata attribute is removed, then that metadata will be removed. If metadata values are updated, they will be updated accordingly.

If any values are set for **is manually edited, loaded, available, info integrations hub, info promo server** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_page
HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_page</inthub:name>
    <inthub:description>new description</inthub:description>
    <inthub:offset>00:00:05.000</inthub:offset>
    <inthub:duration>00:00:01.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:metadata />
</inthub:secondaryEvent>
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
```

Copyright © 2024 Vizrt

```
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:pageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_page</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_page" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_page" />
    <inthub:refLink rel="pagecontent" type="application/vizrt.integrationshubv2+xml; type=pagecontent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_page/pagecontent" />
    <inthub:description>new description</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:05.000</inthub:offset>
    <inthub:takeCount>0</inthub:takeCount>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[U:17:15:15]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:01.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
    <inthub:infoPromoServer />
    <inthub:metadata />
</inthub:secondaryEvent>
```

Secondary Event Reference Page

Name, page id and **created by** can't be adjusted from it's original value.

Description, duration, trigger id and **offset** can be empty.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_ref HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:refPageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_ref</inthub:name>
    <inthub:description>New Logo</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
```

Copyright © 2024 Vizrt

```
    <inthub:pageId>6000</inthub:pageId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="inthub:refPageEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_ref</inthub:name>
    <inthub:refLink rel="self" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_ref" />
    <inthub:refLink rel="edit" type="application/vizrt.integrationshubv2+xml; type=secondaryevent" href="http://127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_ref" />
    <inthub:description>New Logo</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:takeCount />
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[U:17:10:05]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pageId>6000</inthub:pageId>
</inthub:secondaryEvent>
```

Secondary Event Action

Name, action id and **created by** can't be adjusted from it's original value.

Description, trigger id, offset and **action command** can be empty.

If any values are set for **is manually edited, loaded, available, info integrations hub, take count** and **action category**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_action HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
```

```
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:actionEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_action</inthub:name>
    <inthub:description>A cleanup action</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:actionId>viz_cleanup</inthub:actionId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:actionCommand>SCENE CLEANUP 1</inthub:actionCommand>
</inthub:secondaryEvent>
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent

<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:type="inthub:actionEvent" xmlns:inthub="http://
www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_action</inthub:name>
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
secondaryevents/secondary_1_action" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
secondaryevents/secondary_1_action" />
    <inthub:description>A cleanup action</inthub:description>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:takeCount />
    <inthub:createdBy>inthub</inthub:createdBy>
    <inthub:infoIntegrationsHub>[U:16:34:12]</inthub:infoIntegrationsHub>
    <inthub:loaded />
    <inthub:available />
    <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
    <inthub:actionId>viz_cleanup</inthub:actionId>
    <inthub:actionCommand>SCENE CLEANUP 1</inthub:actionCommand>
    <inthub:actionCategory>viz</inthub:actionCategory>
</inthub:secondaryEvent>
```

Secondary Event Pilot Data Element

Name, pilot data element id and **created by** can't be adjusted from it's original value.

Description, duration, trigger id and **offset** can be empty.

If any values are set for **is manually edited, loaded, available, info integrations hub** and **take count**, the payload will be accepted but the values will be ignored as they can't be set.

Copyright © 2024 Vizrt

Request

```
PUT /api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_pde
HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:pilotDataElementEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_pde</inthub:name>
    <inthub:description>Pilot data element</inthub:description>
    <inthub:offset>00:00:12.000</inthub:offset>
    <inthub:duration>00:00:20.000</inthub:duration>
    <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
    <inthub:triggerId>735</inthub:triggerId>
    <inthub:createdBy>inthub</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
3  <inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance" xsi:type="inthub:pilotDataElementEvent" xmlns:inthub="http://
   www.vizrt.com/integrationshub">
4      <inthub:name>secondary_1_pde</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_pde" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_pde" />
7      <inthub:description>Pilot data element</inthub:description>
8      <inthub:triggerId>735</inthub:triggerId>
9      <inthub:offset>00:00:12.000</inthub:offset>
10     <inthub:takeCount />
11     <inthub:createdBy>inthub</inthub:createdBy>
12     <inthub:infoIntegrationsHub>[U:21:10:30]</inthub:infoIntegrationsHub>
13     <inthub:loaded />
14     <inthub:available />
15     <inthub:isManuallyEdited>false</inthub:isManuallyEdited>
16     <inthub:duration>00:00:20.000</inthub:duration>
17     <inthub:pilotDataElementId>9000</inthub:pilotDataElementId>
18 </inthub:secondaryEvent>
```

Secondary Event Message

Name and **created by** can't be adjusted from it's original value.

Copyright © 2024 Vizrt

Description and **offset** can be empty.

If any values are set for **info integrations hub**, the payload will be accepted but the values will be ignored as they can't be set.

Request

```
PUT /api/channels/test/channel_pool/events/group_1/secondaryevents/
secondary_1_message HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent
<inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
inthub:msgEvent" xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:name>secondary_1_message</inthub:name>
    <inthub:description>This is an updated message</inthub:description>
    <inthub:offset>00:00:10.000</inthub:offset>
    <inthub:createdBy>schedule</inthub:createdBy>
</inthub:secondaryEvent>
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=secondaryevent

3  <inthub:secondaryEvent xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance" xsi:type="inthub:msgEvent" xmlns:inthub="http://www.vizrt.com/
   integrationshub">
4      <inthub:name>secondary_1_message</inthub:name>
5      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_message" />
6      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=secondaryevent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_message" />
7      <inthub:description>This is an updated message</inthub:description>
8      <inthub:offset>00:00:10.000</inthub:offset>
9      <inthub:createdBy>schedule</inthub:createdBy>
10     <inthub:infoIntegrationsHub>[U:14:57:04]</inthub:infoIntegrationsHub>
11 </inthub:secondaryEvent>
```

Copyright © 2024 Vizrt

DELETE a Secondary Event under a Group Event in Channel Pool (Version 2.0)

Delete a Secondary Event under a Group Event from channel pool.

Syntax

```
DELETE /api/channels/[channel]/channel_pool/events/[group]/secondaryevents/
[secondary_event]
```

Request

```
DELETE /api/channels/test/channel_pool/events/group_1/secondaryevents/6000 HTTP/1.1
```

Response

```
1 HTTP/1.1 204 No Content
```

Note: If trying to delete a non-existing Secondary Event the returned error-code will be *HTTP 404*, with a reference to relevant info used in the request-uri.

Copyright © 2024 Vizrt

GET Page Content from a Secondary Event under a Group in Channel Pool (Version 2.0)

Following the downlink for a secondary event under a group to get page content.

Syntax

```
GET /api/channels/[channel]/channel_pool/events/[group]/secondaryevents/
[secondaryevent]/pagecontent
```

Request

```
GET /api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_event_1/
pagecontent HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.integrationshubv2+xml; type=pagecontent
<inthub:pageContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:refLink rel="self" type="application/
vizrt.integrationshubv2+xml; type=pagecontent" href="http://
127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
secondaryevents/secondary_event_1/pagecontent" />
    <inthub:refLink rel="edit" type="application/
vizrt.integrationshubv2+xml; type=pagecontent" href="http://
127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
secondaryevents/secondary_event_1/pagecontent" />
    <inthub:pageField name="01butterfly">UP NEXT...</inthub:pageField>
    <inthub:pageField name="02butterfly">CELEBRITY DEATH OCTAGON</inthub:p
ageField>
    <inthub:pageField name="03butterfly">Watch Lady Gaga go up against
Marilyn Manson in the battle for subliminal supremacy. With guest referee
Ozzy Osbourne.</inthub:pageField>
    <inthub:pageField name="Videoclip"></inthub:pageField>
</inthub:pageContent>
```

Disallowed Actions

This end-point does not allow POST or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

PUT Page Content from a Secondary Event under a Group in Channel Pool (Version 2.0)

Update page content of a secondary event of type page.

Page field values are adjustable (empty value allowed), but the name of the field must remain unchanged. The full list of page fields needs to be present in the request, as page fields can't be added or deleted.

Syntax

```
PUT /api/channels/[channel]/channel_pool/events/[group_event]/secondaryevents/
[secondary_event]/pagecontent
```

Request

```
PUT /api/channels/test/channel_pool/events/group_1/secondaryevents/secondary_1_page/
pagecontent HTTP/1.1
Content-Type: application/vizrt.integrationshubv2+xml; type=pagecontent
<inthub:pageContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:inthub="http://www.vizrt.com/integrationshub">
    <inthub:pageField name="01LT">The Smashing Pumpkins (HD)</inthub:pageField>
    <inthub:pageField name="02LT">Updated song title</inthub:pageField>
</inthub:pageContent>
```

Response

```
1  HTTP/1.1 200 OK
2  Content-Type: application/vizrt.integrationshubv2+xml; type=pagecontent
3  <inthub:pageContent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:inthub="http://www.vizrt.com/integrationshub">
4      <inthub:refLink rel="self" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_page/pagecontent" />
5      <inthub:refLink rel="edit" type="application/
   vizrt.integrationshubv2+xml; type=pagecontent" href="http://
   127.0.0.1:9991/api/channels/test/channel_pool/events/group_1/
   secondaryevents/secondary_1_page/pagecontent" />
6      <inthub:pageField name="01LT">The Smashing Pumpkins (HD)</inthub:pageF
   ield>
7      <inthub:pageField name="02LT">Updated song title</inthub:pageField>
8  </inthub:pageContent>
```

Copyright © 2024 Vizrt

10 Page Content Filling

Page content can come from multiple sources. You are advised to use Promo Server for content filling. Viz Channel Branding supports the following ways of adding content to pages:

10.1 Content Source Alternatives

- Manual Input
- Referenced Pages
- Dynamic Content from Playlist Metadata
- Dynamic Content from Database
- Clearing Dynamic Page Content
- Page Content from External Update Services

10.1.1 Manual Input

Pages can be filled with content without any connection to a schedule, playlist, database or web service (using update service).

You can open a page in the **Page Editor** and fill it in with data by directly typing or selecting content.

You can also reference pages that are already filled with data. Adding pages as **referenced pages** allows you to edit a single page that will also change all other referenced pages.

To edit page fields

1. Preview the page in **Page Editor**.
2. Click a page field to access that field type editor.
3. Edit the field as necessary and move on to the next field.
4. When you are finished editing, click **Save**.

10.1.2 Referenced Pages

Once a playlist is sent to the control room, there is often little scope for across-the-board changes. Each occurrence of a page in a playlist is *independent*, and editing it from within the playlist will only change that specific instance.

However, you can set up a playlist in such a way as to allow for last-minute global changes by associating **referenced pages** to primary events. In this use case, a page is 'added by reference' to the playlist, meaning the originating page as it remains in the **Page Pool** holds the actual data. All used ref-pages merely point back to this page as to what data to show on air. When the content of the page has changed, either in the **Page Pool** list (left panel) or by double-clicking on one of the referenced pages in the actual playlist, all occurrences in all playlists in the current channel are changed both retroactively and henceforth.

Copyright © 2024 Vizrt

To create a referenced page

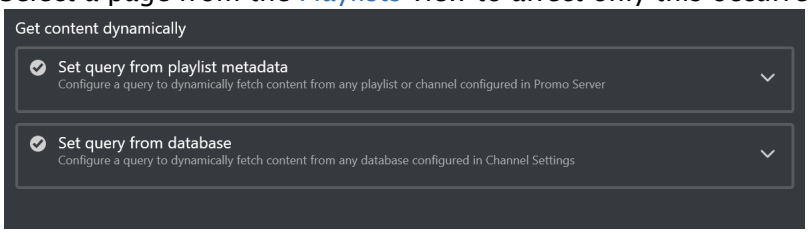
1. Preview the page in **Page Editor** from the **Page Pool**.
2. Edit the page.
3. Press and hold **ALT + drag and drop** the page to the playlist. The page receives the following icon:

10.1.3 Dynamic Content from Playlist Metadata

Pages can automatically retrieve information from primary event metadata in any playlist or channel, using Page Content Filling mapping that is carried out and fulfilled by the Promo Server. Changes to a page in the Page Pool view will affect all referenced occurrences of that page in a playlist. Changes to a page in the Playlists view will only have an effect on page.

To populate a page based on Playlist Metadata content

1. Select your channel.
2. Open the Page Pool or the Playlists view.
3. Double-click a page. This opens the **Page Editor**.
   - Select a page from the Templates, Pages and Pilot Data Elements view to affect all subsequent occurrences of a page in the playlist.
   - Select a page from the Playlists view to affect only this occurrence.



4. Select a field from the page fields list, and you will see **Get content dynamically** on the right side of the Page Editor, if the field type supports Page Content Filling.
5. Click the first expander.
6. Make your configurations. See the section Page Content Filling for Playlist Metadata.
7. Click **Save** or **Save as**.

10.1.4 Dynamic Content from Database

Pages can automatically retrieve information from a range of Database Connections, using the Promo Server to carry out the queries.

When working with databases, it is important to always make sure the database columns **match the data type** of a page's field. If a database column is of type *Integer*, and your field is of type *String,* you will not be able to make any updates.

Note that changes to a page in the Page Pool view will affect all referenced occurrences of that page in a playlist. Changes to a page in the Playlists view will only have an effect on page.

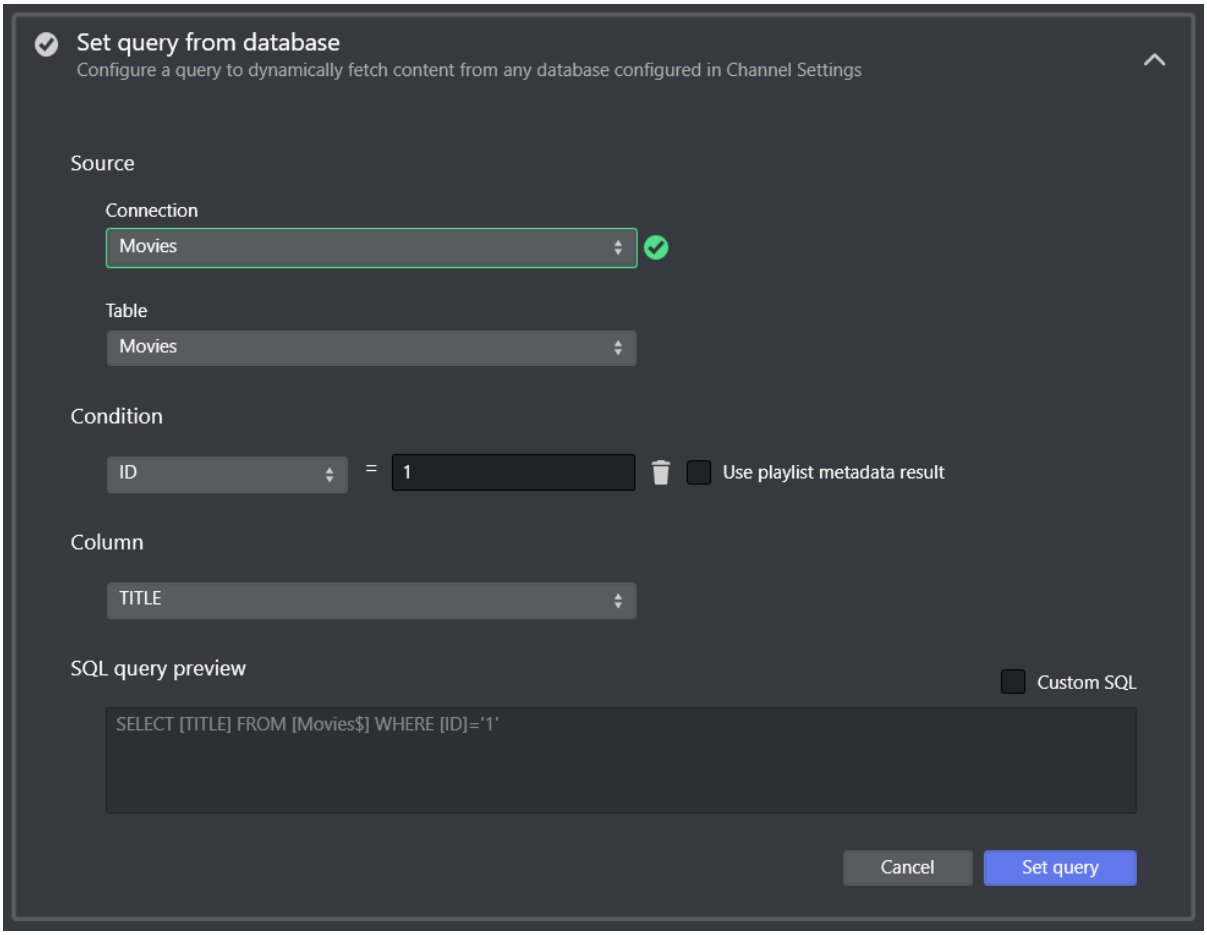
Copyright © 2024 Vizrt

To populate a Template from a Database

> **Note:** Before you start make sure you have configured your Database Connections.

1. Select your channel.
2. Open the Page Pool or the Playlists view.
3. Double-click a page. This opens the **Page Editor**.
4. Select a field from the page fields list, and you will see **Get content dynamically** on the right side of the Page Editor, if this field type supports Page Content Filling.
5. Click the second expander.
6. Make your configurations.
7. Click **Save** or **Save as**.

To configure a database query



1. After selecting a connection, there is an attempt to connect that will fetch all the available data in the database.
   If it failed to connect to the database, you can either try to reconnect, or check Database Connection in Channel Settings.
2. When it is successfully connected, you can choose to generate query string automatically by selecting table, column and conditions, or use custom query by check **Custom SQL**.

Copyright © 2024 Vizrt

3. Click **Set query** when query is configured, Promo Server will send back the result if the page is opened from playlist.
4. Press **Save** or **Save as** to save the query.

Note: For **Oracle** database, we currently support columns with the data types listed below.

Supported data types
CHAR
NUMBER
DATE
VARCHAR
TIMESTAMP
BINARY_DOUBLE
BINARY_FLOAT

Info: For inquiries about additional data types, please contact your local Vizrt customer support team.

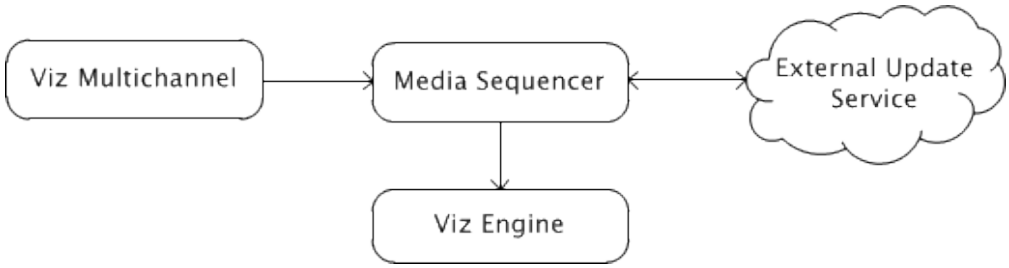
10.1.5 Clearing Dynamic Page Content

There are three ways to clear dynamic query in Page Editor:

- When query is set, you can remove the playlist query or database query in the **Get content dynamically** section on the right side of the Page Editor. There is a trash-bin icon for each expander.
- To clear dynamic query for a field, click the trash-bin icon in the Dynamic query column in Page Fields List on the left bottom of the Page Editor.
- To clear all the dynamic queries for all page fields inside a page, click the **Clear all dynamic queries** button below Page Fields List.
- Press **Save** or **Save as** to save the changes.

Copyright © 2024 Vizrt

10.1.6 Page Content from External Update Services



Page content filling can be done by pointing your template (without a script) to an external update service (for example third party), that can return updated data using the Vizrt Data Format (VDF).

In addition to the service mechanism, you can also select which actions are triggered on the Media Sequencer during an update (read, take etc.) and set the timeout for each action.

Tip: Vizrt has developed a small stub-implementation for an external update service called ExternalUpdateServiceSample for interested developers and third party integrators. This project is available as source code on request from your Support organization at no cost. It can be used as a starting point for developing external update services.

See Also

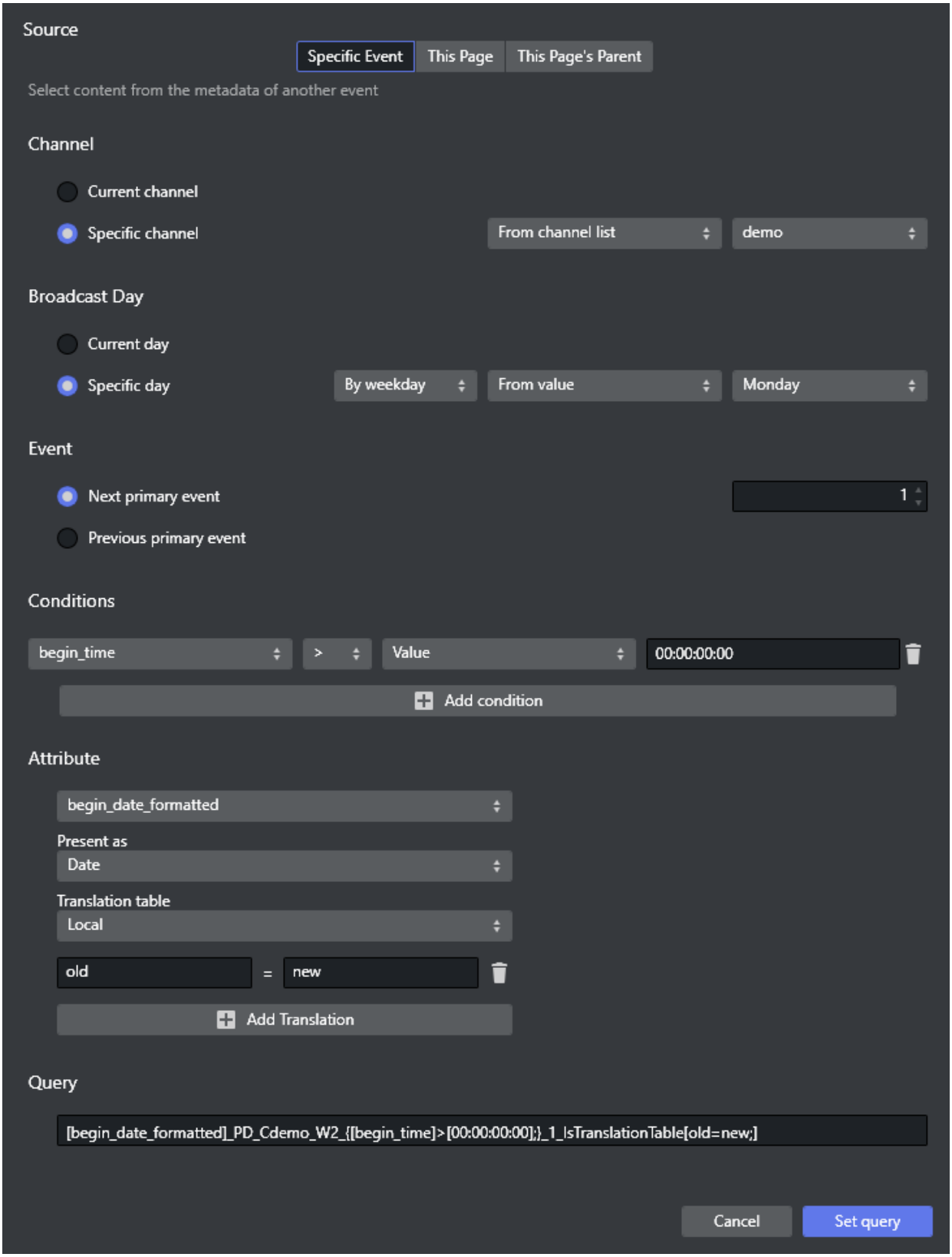
- Vizrt Data Format (VDF)
- Working with Rules
- Page Content Filling for Playlist Metadata

Copyright © 2024 Vizrt

10.2 Page Content Filling For Playlist Metadata

This section provides guidelines and sample values for working with **Page Content Filling** in the Playlist Metadata expander.

Copyright © 2024 Vizrt



Copyright © 2024 Vizrt

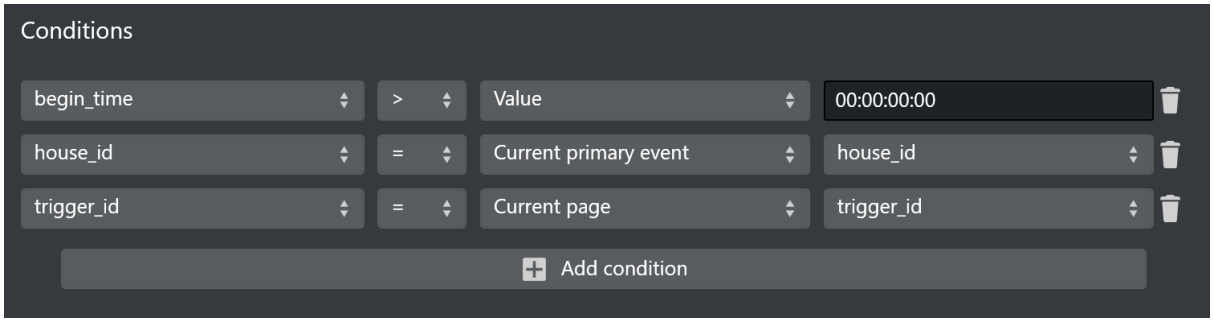
- **Source:** Gets information from a specific event, or the page itself, or the primary event the page resides under. Only the specific event source allows setting the **Channel**, **Broadcast Day**, **Event** and **Conditions** options.
- **Channel:** Lets you select current channel or the relevant channel if you want to populate your pages with primary event content from another channel. Can have one of three values.
  - **From channel list:** Select the relevant channel from the list. Channels are configured in Promo Server configuration (see Promo Server Installation and Configuration).
  - **From current primary event:** Channel name derived from the specific attribute value of primary event.
  - **From current page:** Channel name derived from the specific attribute value of page itself.
- **Broadcast Day:** Lets you select a primary event based on broadcast date. There are three choices for specific day:
  - **By date:** Specify day by selecting a date. If date is derived from an attribute, the value has to be in the ISO date format (yyyy-MM-dd) or the local culture date format from the machine where Promo Server is installed.
  - **By weekday:** Specify day by selecting the next weekday. If weekday is derived from an attribute, the value has to be specified as a number, where 1=Sunday, 2=Monday, etc.
  - **By relative day:** Specify day by a set number of days in the future. For example, **1** for the following day, **2** for two days after the current date, etc.

  It is possible to select where the values are derived from. There are three options:
  - **From value:** Specify a value that should be used.
  - **From current primary event:** Value derived from specific attribute value of primary event.
  - **From current page:** Value derived from specific attribute value of page itself.
- **Event:** Lets you select an item from a different event than the current one.

  > ⚠ **Note:** The old behavior of selecting an item by time or relative to a current event has been replaced by the **Event** option. Unlike the **Item** option, **Event** does not permit negative (relative) values for events in another broadcast day or another channel. In addition, setting the time is now mandatory.

- **Conditions:** Looks for the first primary event that meets the conditions in the table. Any entry prefixed with *secondary_* commands the system to check data from the page (itself) and not the parent (primary element). This lets you use secondary data (for example, *secondary_begin_time*) to find relevant primary event data, giving more dynamic conditions. Any entry prefixed with *metadata_* commands the system to check custom data added in metadata section.

Copyright © 2024 Vizrt

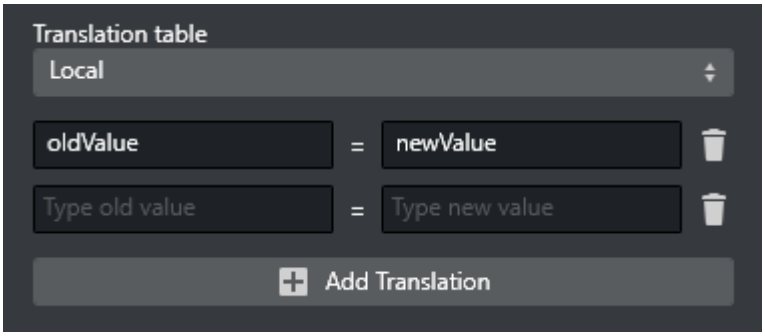


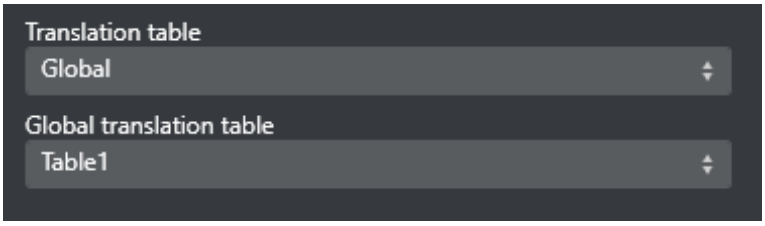
- **Value Type**, which has three options:
  - **Value:** Compares attribute value with the condition value.
  - **Current primary event:** Compares attribute value with the specific attribute value of primary event.
  - **Current page:** Compares attribute value with the specific attribute value of the page itself.
- **Attribute:** Lets you select an attribute from a located primary event or the current secondary event.
  - **Attribute type:** Lets you select type for selected attribute. It does not work for attribute *begin_date, begin_time, begin_date_formatted,* and *duration*. There are six options:
    - **House ID:** Treats the attribute as House ID in Viz One. This option is only valid for Image and Video page field type.
    - **String:** Treats the attribute as String. This option is only valid for Image, Clock and Text type page field.
    - **Integer:** Treats the attribute as Integer. This option is only valid for Integer type page field.
    - **Float:** Treats the attribute as Float. This option is only valid for Float type page field.
    - **UTC Date:** Treats the attribute as Date in UTC (Coordinated Universal Time). This option is only valid for Text page field type.
    - **UTC Time:** Treats the attribute as Time in UTC (Coordinated Universal Time). This option is only valid for Text page field type.
    - **Local Date:** Treats the attribute as Date in Local Time zone. This option is only valid for Text page field type.
    - **Local Time:** Treats the attribute as Time in Local Time zone. This option is only valid for Text page field type.
  - **Translation Tables:** See Translation Tables
- **Query:** Displays the resulting string based on your configuration of the respective fields. Can be edited.

See Also: Page Content Filling Aliases

Copyright © 2024 Vizrt

10.3 Translation Tables





Translation tables can be used to define arbitrary value-translations, via an oldValue=newValue table. There are two kinds of translation tables:

- **Local:** settings configured locally per page field.
- **Global:** the Global Translation Table can be configured in Channel Settings under Page Content Filling > Translation Tables.
- **Match and replace rules in translation table:** There are some special characters can be used in the old value (left part) of the translation table that have special rules when trying to match and replace the value. It also supports partial match, * means one or more characters, ? means one character.

Symbol(s)	Description	Examples
*	Match and replace - wildcard pattern	· He*=cat - "He is an old dog" is translated to "cat". · H*g=cat - "He is an old dog" is translated to "cat".' · d*=cat - "He is an old dog" has no match and the result is "He is an old dog".

Copyright © 2024 Vizrt

Symbol(s)	Description	Examples
?	Match and replace - wildcard a specific number of characters	· He???=Goodbye - "Hello" is translated to "Goodbye". · He??o=Hi - "Hello" is translated to "Hi". · H?=cat - "He is an old dog" has no match and the result is "He is an old dog".
()	Match and replace a **substring**	· (d*)=cat - "He is an old dog" is translated to "He is an old cat". · (??)=cat - "He is an old dog" is translated to "cat cat cat old dog". · (o*)=cat - "He is an old dog" is translated to "He is an cat dog". · (*o*)=cat or (?o*)=cat - "He is an old dog" is translated to "He is an old cat".
[]	Match and replace the **character**.	· [o]=x - "He is an old dog" is translated to "He is an xld dxg". · [er]=PPP - "He is a very old dog" is translated to "He is a vPPPy old dog". · [d?d]=123 - "He is a very old dog" is translated to "He is a very ol123og".

Copyright © 2024 Vizrt

11 Promo Server

Promo Server is a back-end Windows service process that lets you handle page-filling and playlist data transfers efficiently, particularly for cross-promotional purposes.

> **Note:** Use the Viz Channel Branding Client Configuration settings to configure the server address of the Promo Server and to direct Channel Branding system to use Promo Server for a specific channel.

This section covers the following topics:

- About Cross-Promotion
- Configuration
- Promo Server Logs and Messages

11.1 About Cross-Promotion

Cross-promotion is typically used in scenarios where one or more channels need to access information about events in other channels in order to fetch data for page-filling. For example, the channel MUSIC might need to know what events are scheduled on channel MOVIES in order to answer such queries as “The Next event on channel MUSIC is program Foo while channel MOVIES now shows the film Bar”. Promo Server is a utility for handling such cross-channel queries.

11.2 Configuration

Promo Server is configured to handle a specific channel by registering it via Settings → Page Content Filling → Promo Server Address

While there is a configuration file in the directory *%ProgramData%\Vizrt\Viz Channel Branding\Promo Server* it is not advisable to manually edit this unless expert knowledge is had.

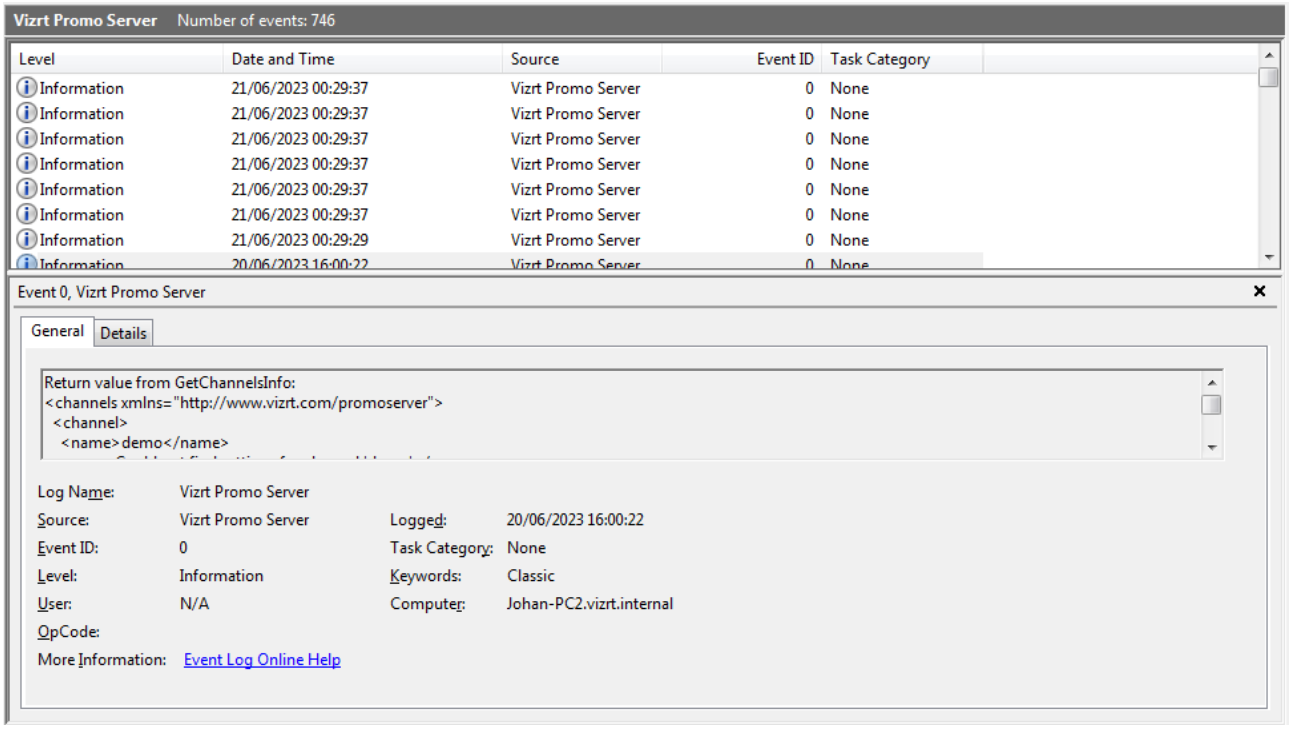
11.3 Promo Server Logs And Messages

Perform the steps below to check Promo Server logs.

> **Note:** This procedure is for Windows 7. Although minor differences may occur depending on your Windows version, the basic procedure will be the same.

1. Press the Windows **Start** button and type **Event Viewer** in the search bar. Press **Enter** to start the Event Viewer.
2. In the resulting Event Viewer application, expand **Application and Services** in the left pane.
3. Select and click the service process log you wish to view. In this example, Vizrt Promo Server, as seen in the screenshot below:

Copyright © 2024 Vizrt

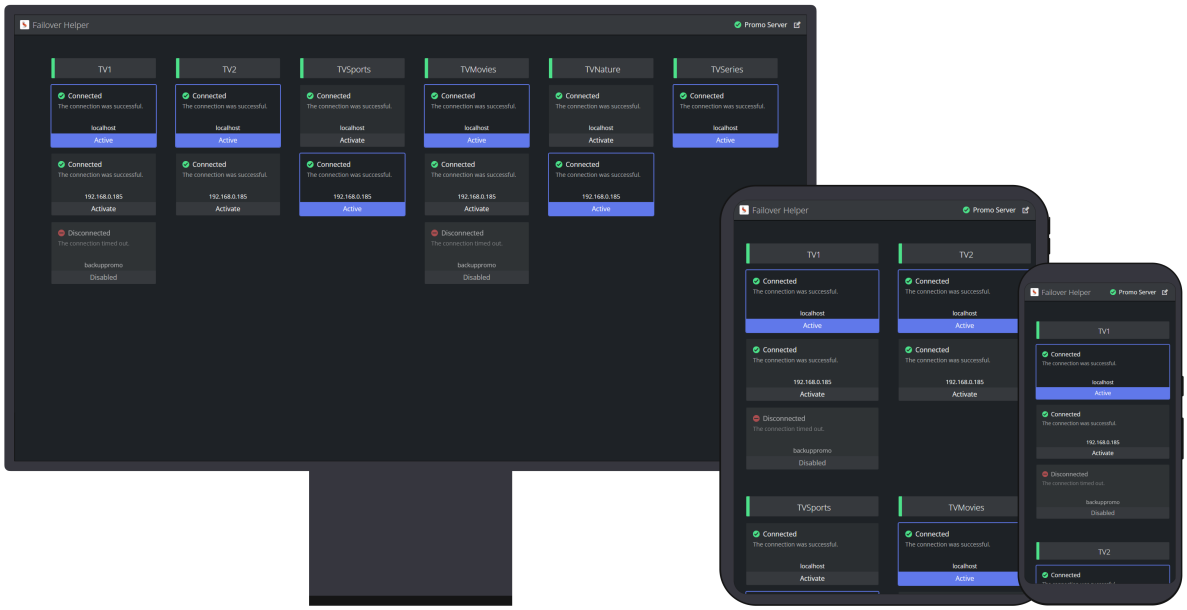

Vizrt Promo Server Number of events: 746
Level Date and Time Source Event ID Task Category
Information 21/06/2023 00:29:37 Vizrt Promo Server 0 None
Information 21/06/2023 00:29:37 Vizrt Promo Server 0 None
Information 21/06/2023 00:29:37 Vizrt Promo Server 0 None
Information 21/06/2023 00:29:37 Vizrt Promo Server 0 None
Information 21/06/2023 00:29:37 Vizrt Promo Server 0 None
Information 21/06/2023 00:29:37 Vizrt Promo Server 0 None
Information 21/06/2023 00:29:29 Vizrt Promo Server 0 None
Event 0, Vizrt Promo Server
General Details
Return value from GetChannelsInfo:
<channels xmlns="http://www.vizrt.com/promoserver">
<channel>
<name>demo</name>
Log Name: Vizrt Promo Server
Source: Vizrt Promo Server
Logged: 20/06/2023 16:00:22
Event ID: 0
Task Category: None
Level: Information
Keywords: Classic
User: N/A
Computer: Johan-PC2.vizrt.internal
OpCode:
More Information: Event Log Online Help

12 Failover Helper

Failover Helper is a web application that offers a preview of the status of each channel that is configured in **Promo Server**. In a failure scenario, **Failover Helper** allows for quickly activating another instance of a channel that is available.

It communicates with the Promo Server REST API for Remote Configuration to retrieve and update the current definitions of channels and their allowed machines (IPs).

The web application is responsive and can be used in desktop, tablet and mobile.



- Configuring Failover Helper
- Monitoring the Failover Helper

12.1 Configuring Failover Helper

12.1.1 Working with the Failover Helper

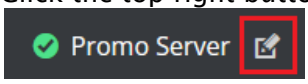
- After installation, from the location *C:\Program Files\Vizrt\Viz Channel Branding\Failover Helper* launch the **Failover Helper** application.

- The port for **Failover Helper** is by default 9960. It can be accessed through any browser in http://YOURHOSTNAME:9960, for example http://127.0.0.1:9960.
- When opening Failover Helper on a browser it will by default attempt to connect to **Promo Server** in *localhost*.

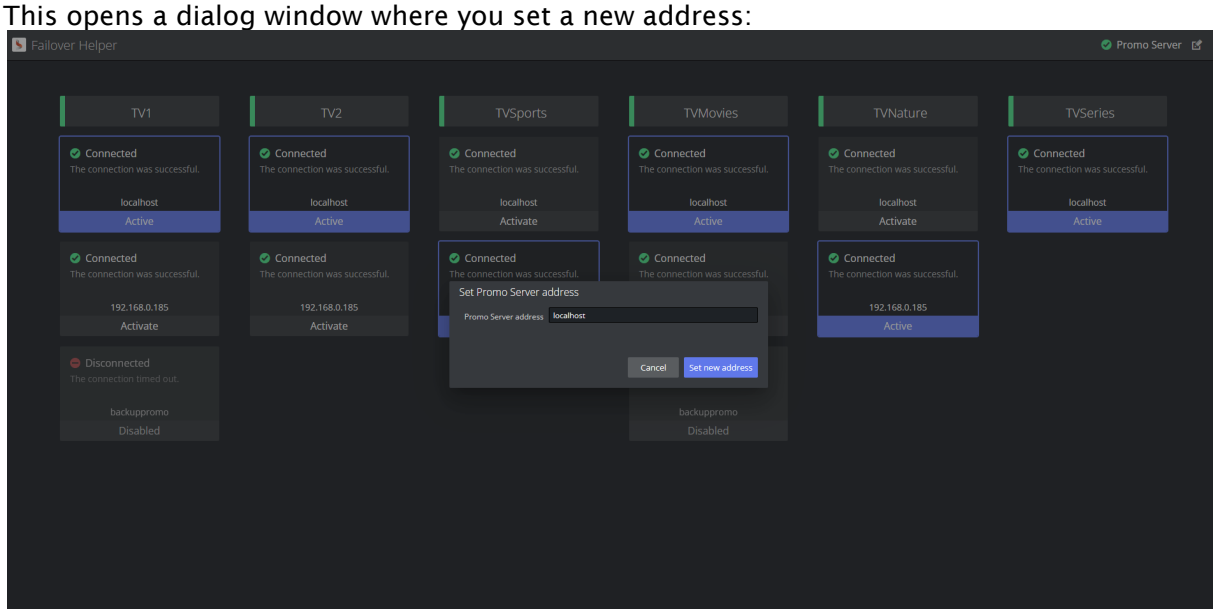
Updating the Promo Server Address

To update the Promo Server address

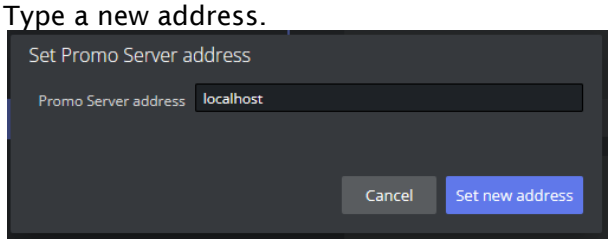
1. Click the top-right button **Promo Server**.



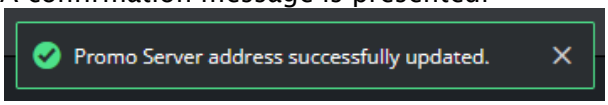
2. This opens a dialog window where you set a new address:



3. Type a new address.



4. Confirm your changes by clicking **Set new address**.
5. A confirmation message is presented:



The dashboard is now updated and fetches the channels configured in that Promo Server.

Channels and Channel Instances

To add a Channel Instance

Channels and machine addresses associated with each channel can be configured from their respective client machines in **Channel Settings**.

- Refer to section Page Content Filling.

See Also

- Promo Server Remote Configuration REST API

12.2 Monitoring The Failover Helper

Every five seconds, the Failover Helper sends a request to the Promo Server to fetch channel information, in order to display the current status of each channel and its instances.

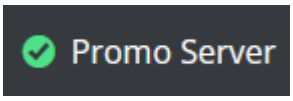
12.2.1 Monitoring Status

This section covers the following topics:

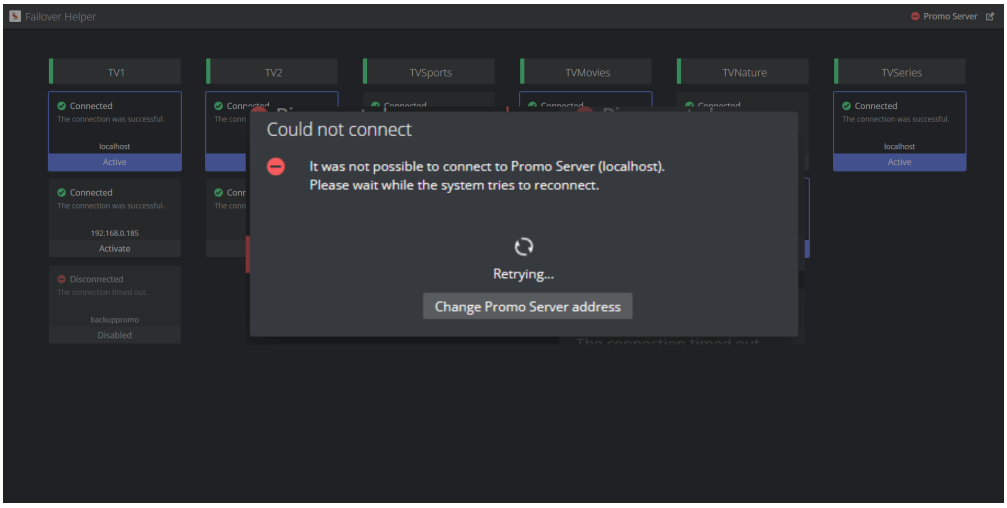
- Promo Server Status
- Channel List
- Channel Instance
- Activating a Channel Instance
- Channel Instance Errors

Promo Server Status

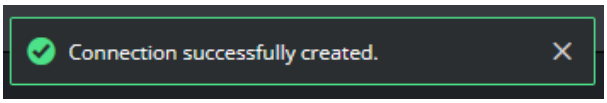
The status of the connection to **Promo Server** is always displayed in the right corner of the top bar of the application:



If there is a problem with the connection, a dialog box will lock the application and it will automatically try to reconnect to **Promo Server**.

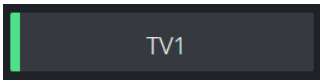
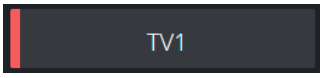


When the connection is again successful the dialog box automatically closes and a success message displayed.
You can now continue to interact with the application.

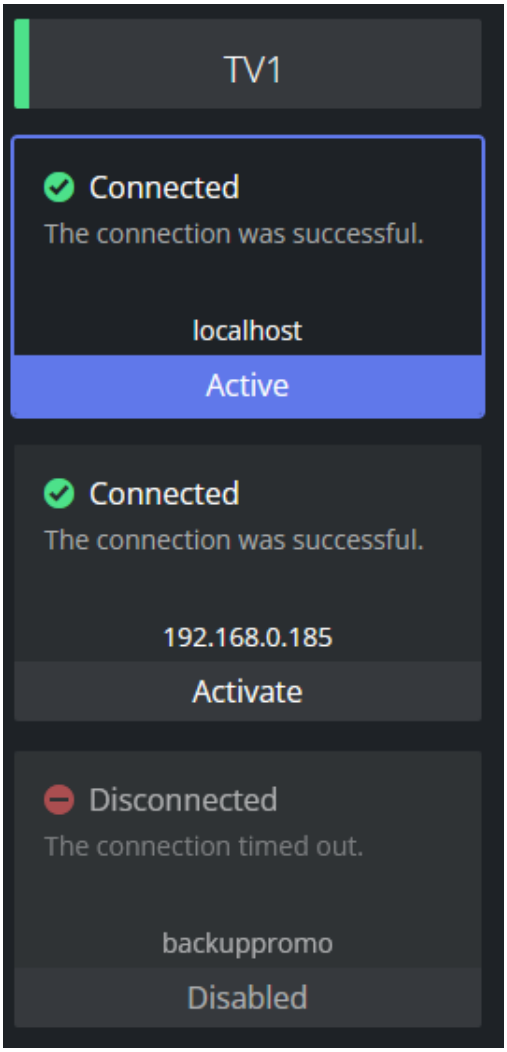


Channel List

For each channel, there is a title card with the name of the channel and color-coded status.

Status type	Example	Status description
Success (green)		The active instance that is configured for this channel is available and it is possible to successfully connect to it.
Failure (red)		The active instance that is configured for this channel is in faulty (see possible errors below).

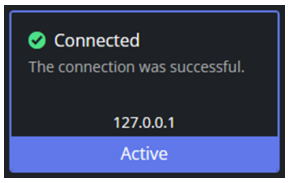
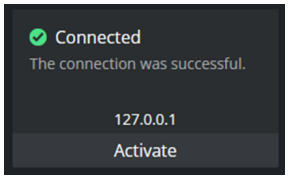
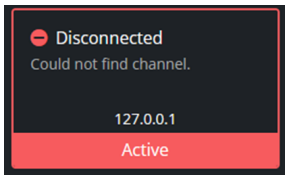
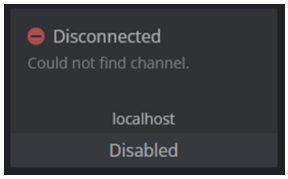
For each channel, there is a vertical list of the configured machine addresses. Each instance of a channel machine is represented by a card that includes the current status of the connection and the machine address.



Copyright © 2024 Vizrt

Channel Instance

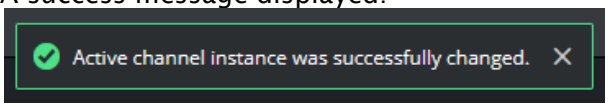
Channel instance cards can be presented in different styles according to the state of each machine:

State type	Example	State description
Connected and Active		There are no issues with the connection and the instance is the active one for the channel.
Connected and Available		There are no issues with the connection so the instance is available to be selected as active for the channel.
Disconnected and Active		There is an issue with the connection and the instance is the active one for the channel. This is the state that activates the red signal in the channel title.
Disconnected and Disabled		There is an issue with the connection and the instance is disabled as it should not be activated for the channel.

Activating a Channel Instance

To activate a Channel Instance

It is possible to update the active instance of a specific channel. Any instance in the state *Connected and Available* can be set as active.

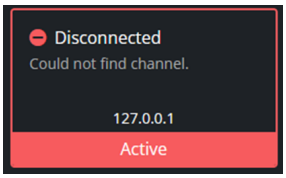
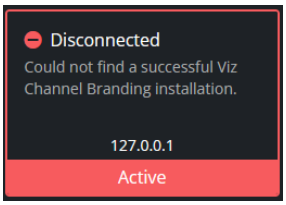
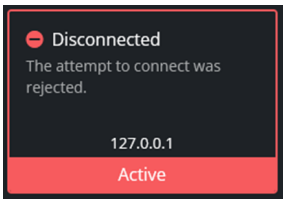
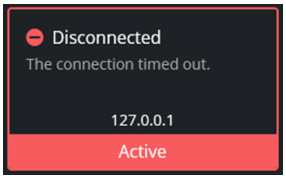
1. Click anywhere on the card.
2. A success message displayed:
   

3. After activation
   a. The clicked card will be updated to the *Connected and Active* state.
   b. If the previously active instance was in state *Disconnected and Active*, the channel title will be updated to status *Success (green)*.

Copyright © 2024 Vizrt

Channel Instance Errors

Various error messages are presented when there is a problem connecting to a given machine.

Error message	Example	Description
Could not find channel		It is possible to connect to the Media Sequencer (MSE), but this specific channel could not be found.
Could not find a successful Viz Channel Branding installation		It is possible to connect to Media Sequencer (MSE), but there is no valid *Viz Channel Branding* definition.
The attempt to connect was rejected		The attempt to connect was rejected, could be that the port was not open, among other issues.
The connection timed out		There was an attempt to reach the address but there was no reply, it could be that the address is not valid.

Copyright © 2024 Vizrt

13 XML Translator

XML Translator is a command-line tool for transforming a schedule-file in XML (BXF) format to a format Playlist Importer can import. Content is not changed - the data format is only transformed so the Playlist Importer can process it.

Any third-party can define an XSLT (.xst) transformation to an importer-friendly structure, currently:

- Ordered collections
- Two (2) layers deep only

This section covers the following topics:

- Using XML Translator
- External References

13.1 Using XML Translator

Open a command window (**Windows-button+R CMD <enter>**) to use the XML translator. If you need admin rights when using the tool, start CMD as Administrator. Next, **right-click CMD** and select **Run as Administrator** to start an elevated session. In the following, it is assumed that you have XML translator program in the Windows PATH. If not, you must provide the full path in order to use the tool.

```
XMLTranslator.exe -SourcePath %path% -DestinationPath %dpath% -SuccessPath %spath%
-FailPath %fpath% -TranslationFile %tpath%\XSL.xslt
```

The example invocation above is on one line. All arguments are required. To get usage instructions, run the program with the help flag (-? or -h or):

```
XmlTranslator.exe -–help
```

Program argument definitions:

- **SourcePath** the directory that contains the XML file to translate
- **DestinationPath** a directory in which the translated XML file is created
- **SuccessPath** a directory to which the source XML file is moved when translation succeeds
- **FailPath** a directory to which the source XML file is moved if the translation fails
- **TranslationFile** path to the XSL file defining the transformations

A typical run (the command must be entered on a single line) could be:

```
XMLTranslator.exe -SourcePath D:\Source -DestinationPath D:\Destination -SuccessPath
D:\Success -FailPath D:\Fail -TranslationFile D:\BroadViewXML.xsl
```

Copyright © 2024 Vizrt

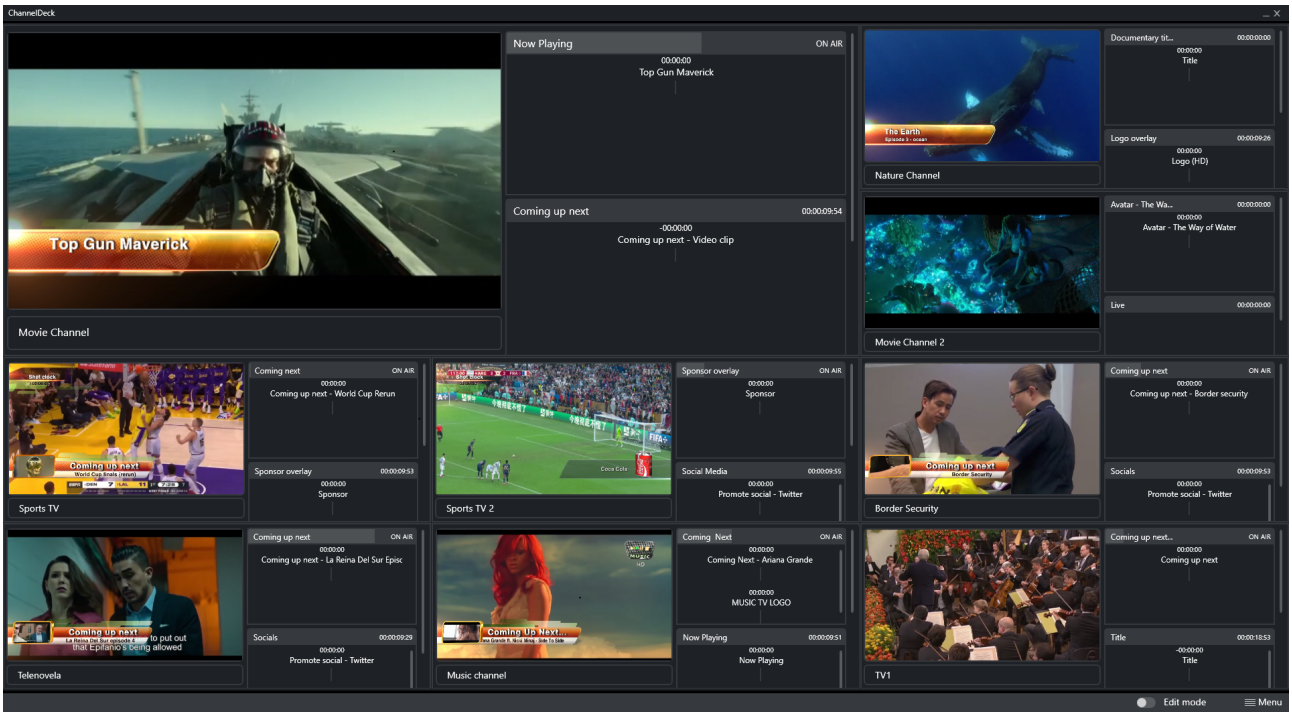
13.2 External References

- Wikipedia: Extensible Markup Language: XML
- Wikipedia: Extensible Stylesheet Language Transformations: XSLT
- Wikipedia: Broadcast Exchange Format: BXF

Copyright © 2024 Vizrt

14 Channel Deck

Channel Deck is an easy to use tool for monitoring of multiple channel brandings. From a single dashboard, users can monitor up to twelve playouts and their respective state.



- Setting up Channel Deck
- Channel Deck Configuration and Layout
- Managing Channel Connections
- Monitoring Channel Deck

Copyright © 2024 Vizrt

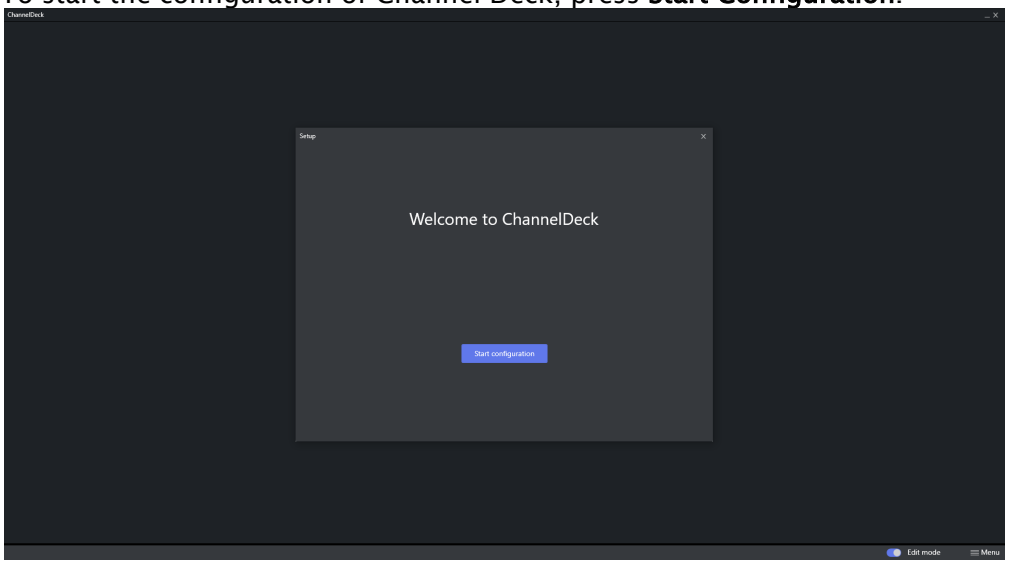
14.1 Setting Up Channel Deck

This page walks through the initial set up of Channel Deck.

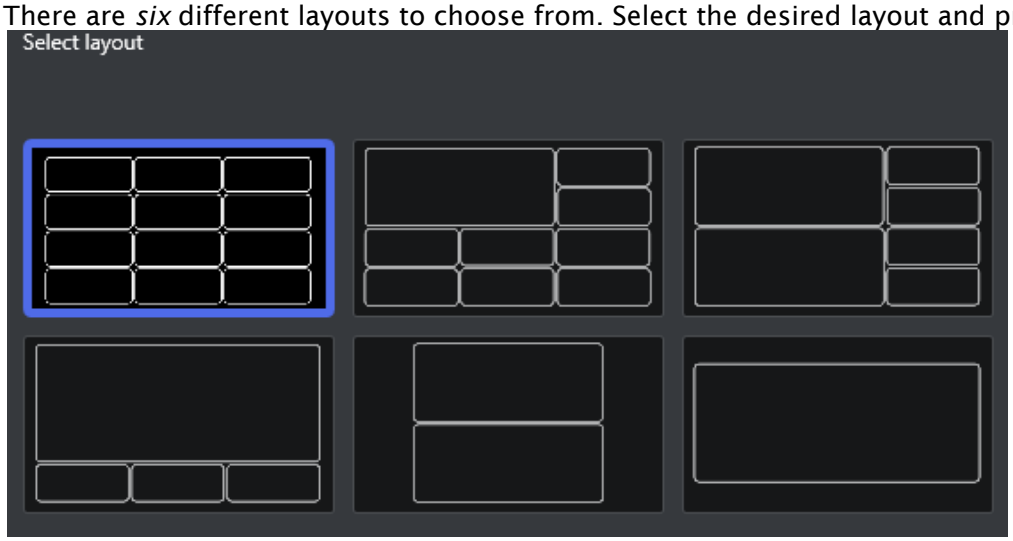
14.1.1 Setting up Channel Deck

First Run Start Up

1. To start the configuration of Channel Deck, press **Start Configuration**.

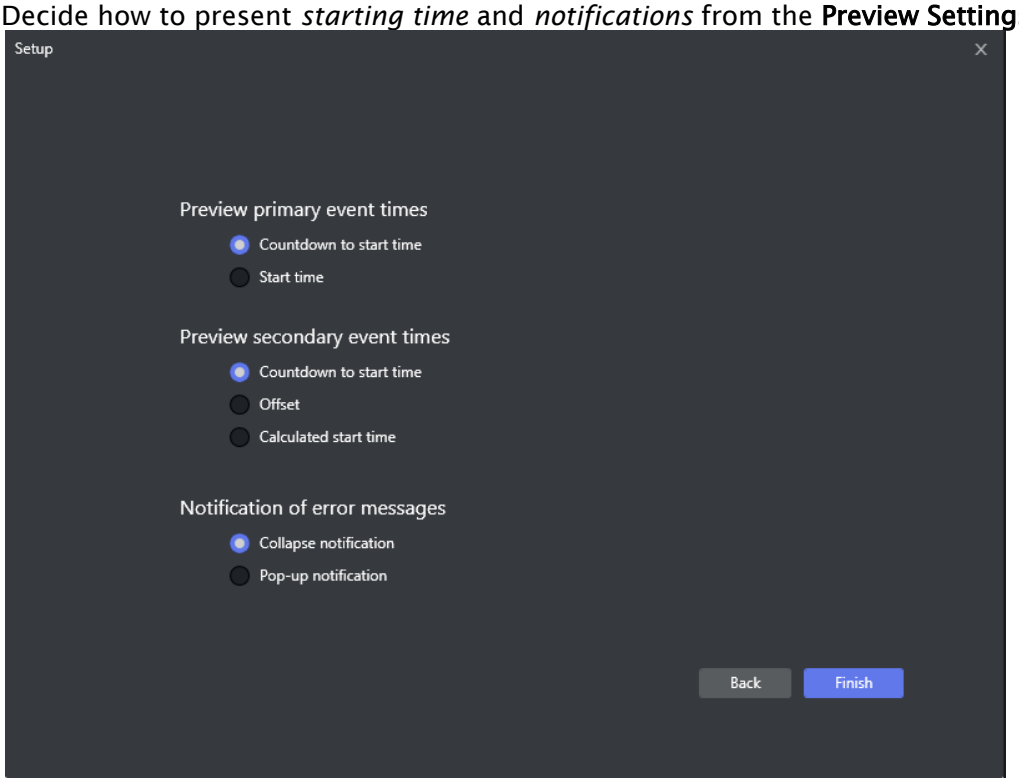


2. There are *six* different layouts to choose from. Select the desired layout and press **Next**.



Copyright © 2024 Vizrt

3. Decide how to present *starting time* and *notifications* from the **Preview Settings** menu.



> You can change both *layout* and *Preview settings* by either pressing **Back** or at a later stage, by using the **Menu** button in the bottom right corner of the application. For more information on configurations and changing configuration or layout, section Channel Deck Configuration and Layout.

4. Once the desired settings are selected press **Finish**.
5. Move on to the next step (see section connect to a channel)**.**

Copyright © 2024 Vizrt

14.2 Channel Deck Configuration And Layout

14.2.1 Settings

Primary event Settings	Description
Countdown to start time	Count down timer for scheduled playout "Start Time" **Format:** *DD:HH:MM:SS*
Start time	Static time display based on playout "Start Time" **Format:** *Datetime based on System settings eg.* · *Nordic: YYYY-MM-DD HH:MM:SS* · *US: DD/MM/YYYY HH:MM:SS AM/PM*
Secondary event Settings	**Description**
Countdown to start time	Count down timer for scheduled playout "Start Time" **Format:** *HH:MM:SS*
Offset	Static offset time based on playout secondary event "Offset" **Format:** *HH:MM:SS*
Calculated start time	Static time display based on playout "Start Time" + Secondary event "Offset" **Format:** *Datetime based on System settings eg.* · *Nordic: YYYY-MM-DD HH:MM:SS* · *US: DD/MM/YYYY HH:MM:SS AM/PM*
Notification of error messages	**Description**
Collapse notification	Error "pop-up" is by default collapsed and requires a button click to display details
Pop-up notification	Error "pop-up" is by default showing and displays details, can be closed by pressing the close button

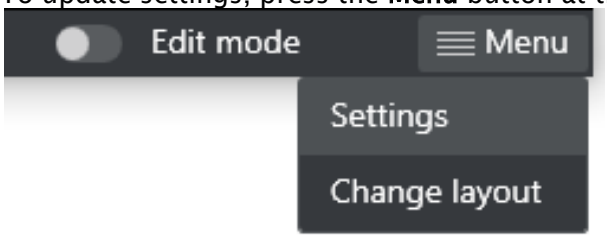
Copyright © 2024 Vizrt

14.2.2 Working with Settings

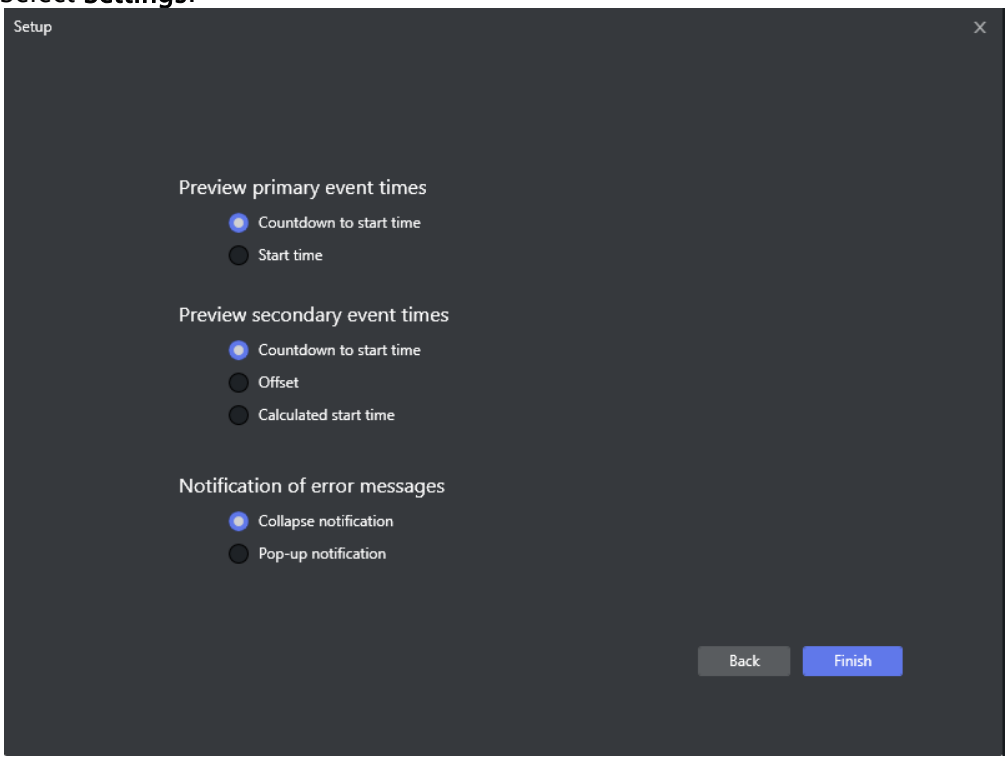
Changing Settings

To update settings

1. To update settings, press the **Menu** button at the bottom-right.



2. Select **Settings**.

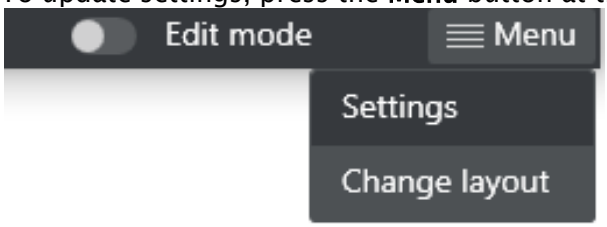


Copyright © 2024 Vizrt

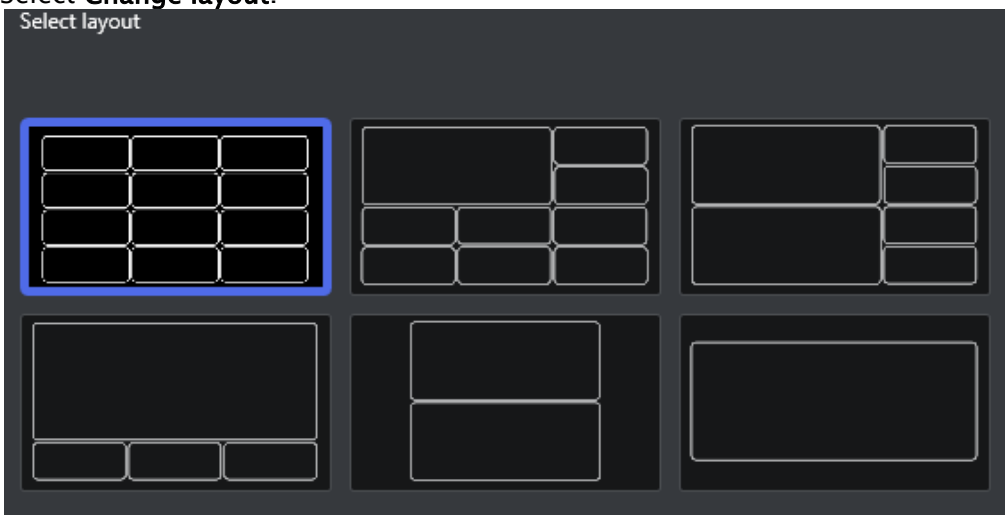
Changing Layout

To Change the view layout

1. To update settings, press the **Menu** button at the bottom-right.



2. Select **Change layout**.



14.3 Managing Channel Connections

14.3.1 Prerequisites

Before connecting to a Media Sequencer make sure that the playout configuration has been done. See playout configuration - Configuring Device Channels.

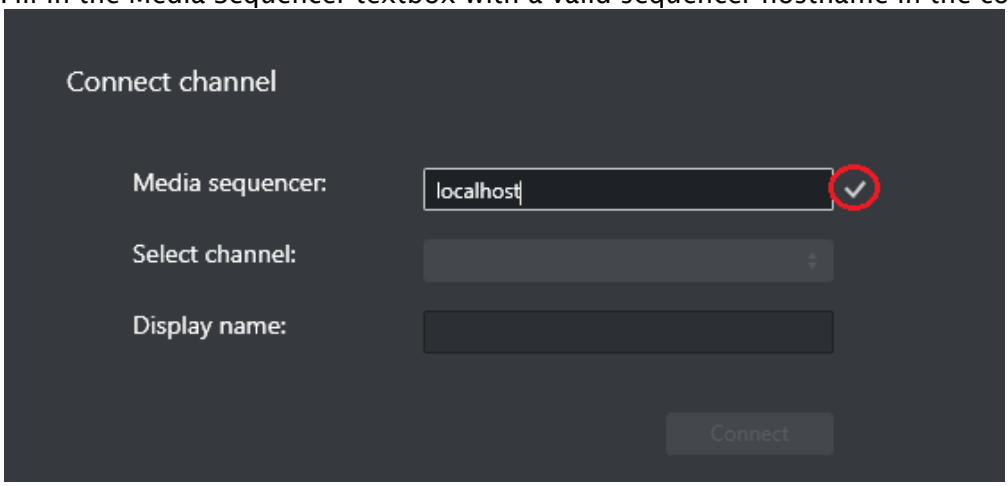
- Creating a new channel connection
- Edit Mode

14.3.2 Working with Channel Connections

Creating a new channel connection

To create a new channel

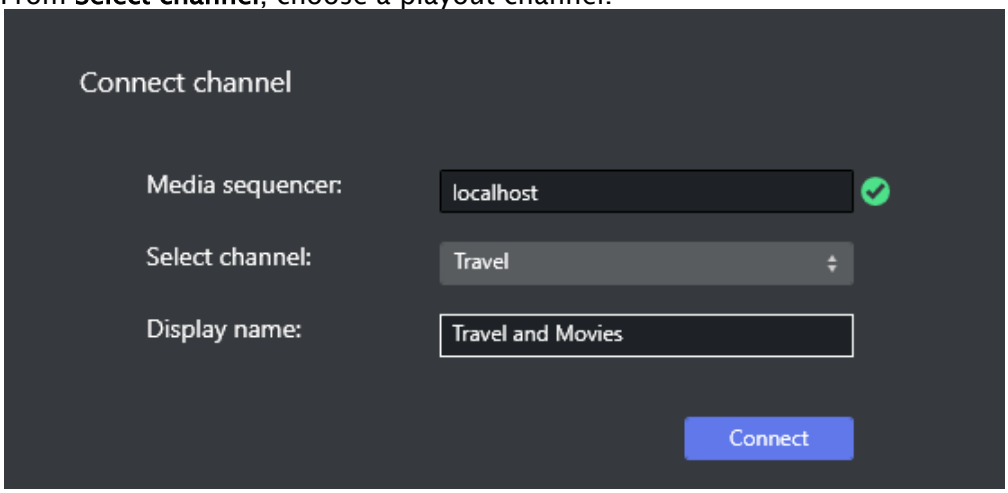
1. Fill in the Media Sequencer textbox with a valid sequencer hostname in the connection form.



2. Press the **checkmark** to establish the connection.

Copyright © 2024 Vizrt

3. From **Select channel**, choose a playout channel.



4. Adjust **Displayed name** for the channel if it is not sufficiently descriptive.
5. Press **Connect** to start monitoring.

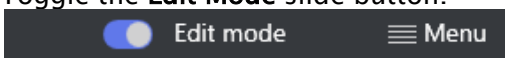
For more information on monitoring see section Monitoring Channel Deck.

Edit Mode

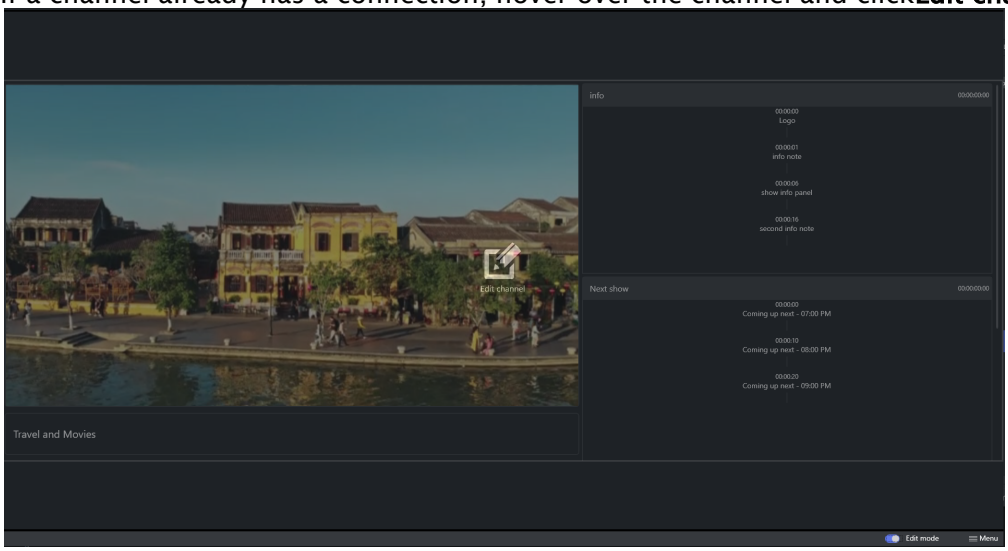
Edit mode allows setting up a new channel connection, and editing an existing one.

To edit an existing channel

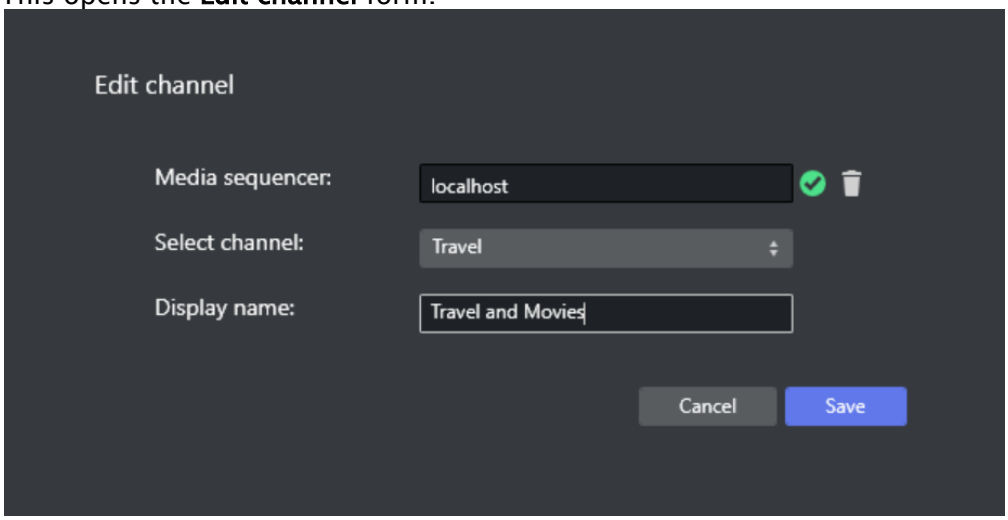
1. Toggle the **Edit Mode** slide button.



2. If a channel already has a connection, hover over the channel and click**Edit channel**.



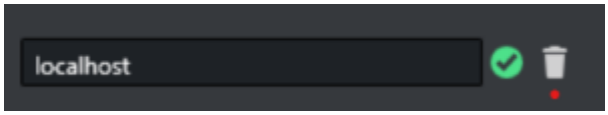
3. This opens the **Edit channel** form.



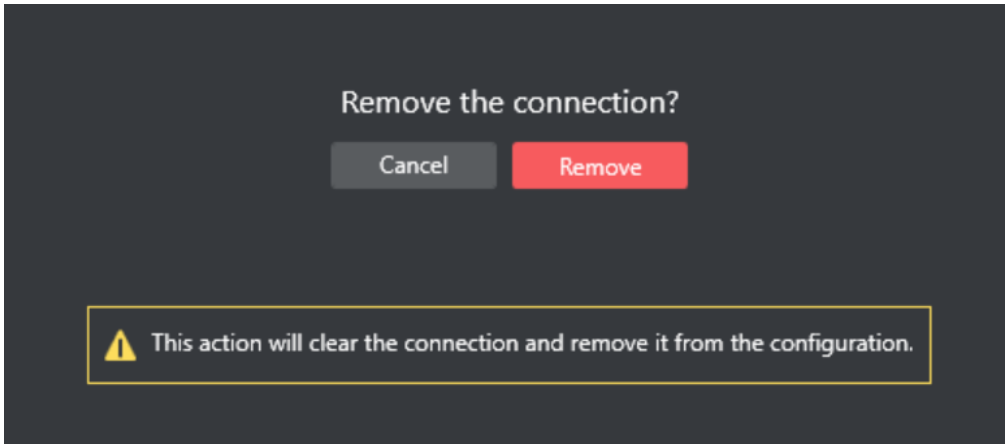
4. You can change values for:
   a. Media Sequencer
   b. Channel
   c. Display Name.

To remove an existing channel

1. Open the **Edit channel** form.
2. Click on the Trash can icon.



3. Click **Remove**.



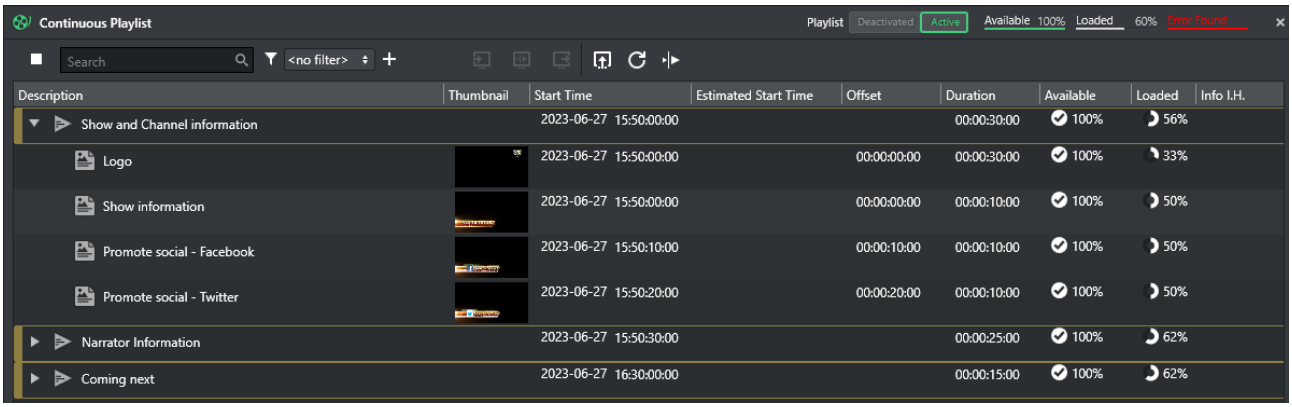
4. This removes the channel from the configuration and closes the connection. (Click **Cancel** to return back to the **Edit channel** form).

Copyright © 2024 Vizrt

14.4 Monitoring Channel Deck

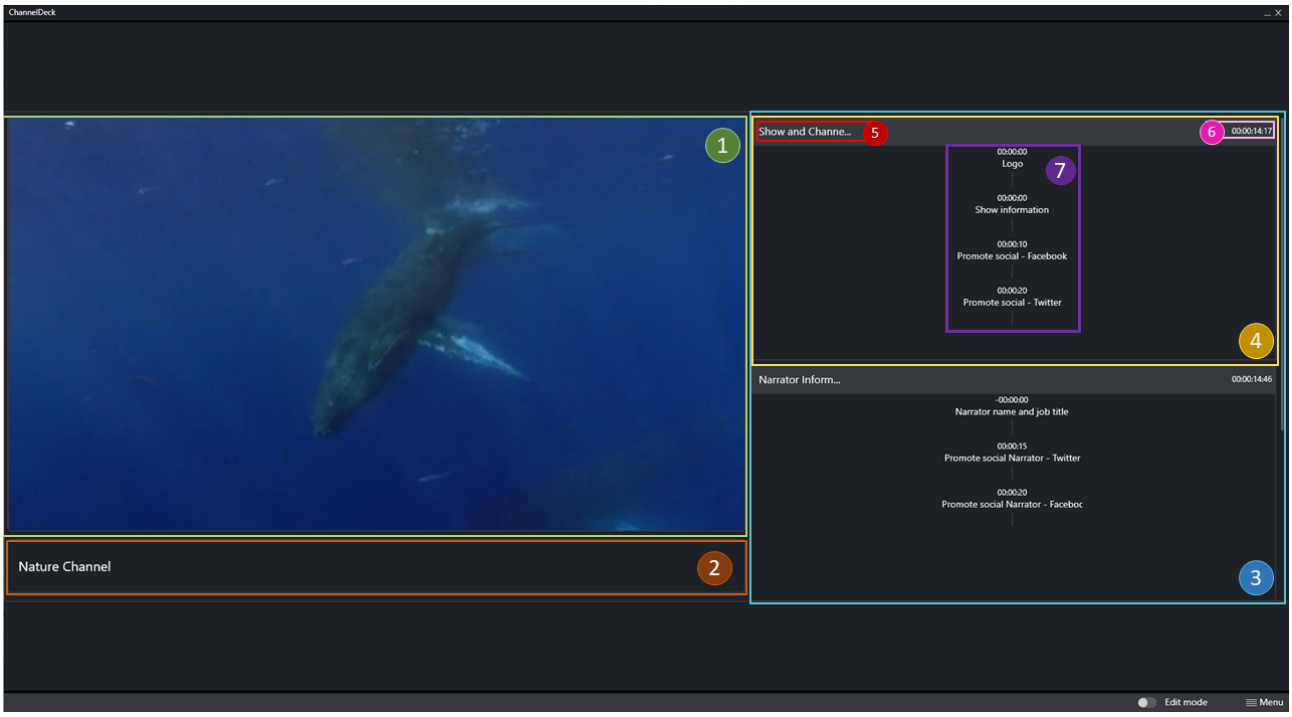
14.4.1 Monitoring Takes and Channel Branding

Channel Branding Client



Channel Deck

Channel Deck displays the contents of a playlist.



Number	Description
1	NDI player stream
2	Display name (set during connection) The Display name component is a button that can toggle error and warning messages if there are any
3	Playlist that presents a list of primary events
4	Primary event
5	Primary event description
6	Primary event planned take time
7	List of secondary events Each event shows: · Planned take time in relation to the primary planned take time (see item 6) · Secondary event description

ON AIR Status

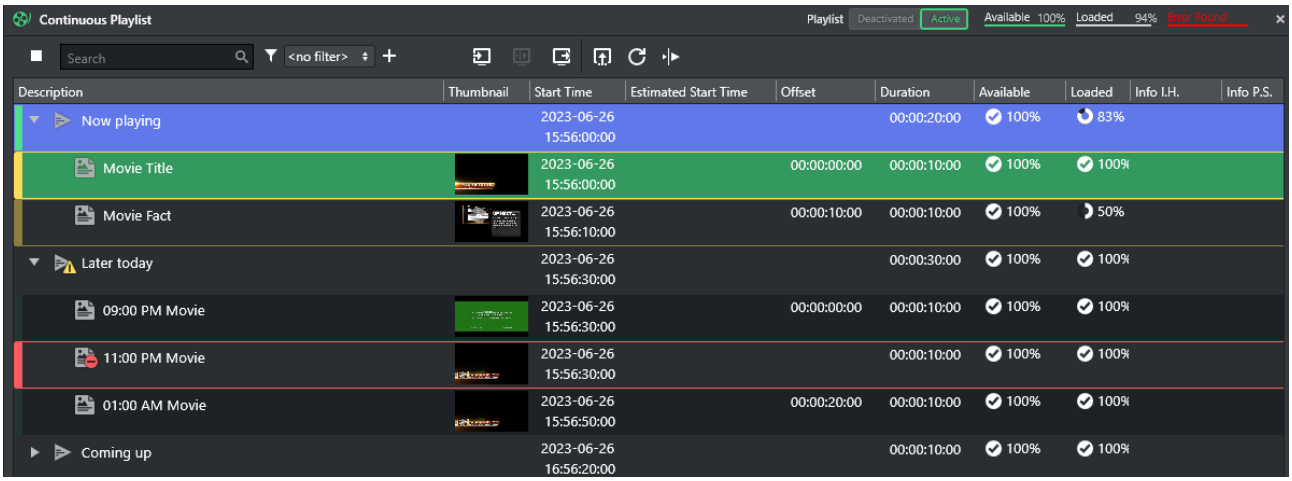
Image of a primary event header

When a take is started, the timer on the right side of the **primary event** header switches to ***"ON AIR"***, indicating that the primary is now in progress.

The progress bar tracks how much of a primary event has been completed, and how much is left.

Copyright © 2024 Vizrt

14.4.2 Monitoring Errors and Warnings



Errors present in the **Channel Branding Client** will be directly reflected in **Channel Deck.**

Two levels should be taken into account:

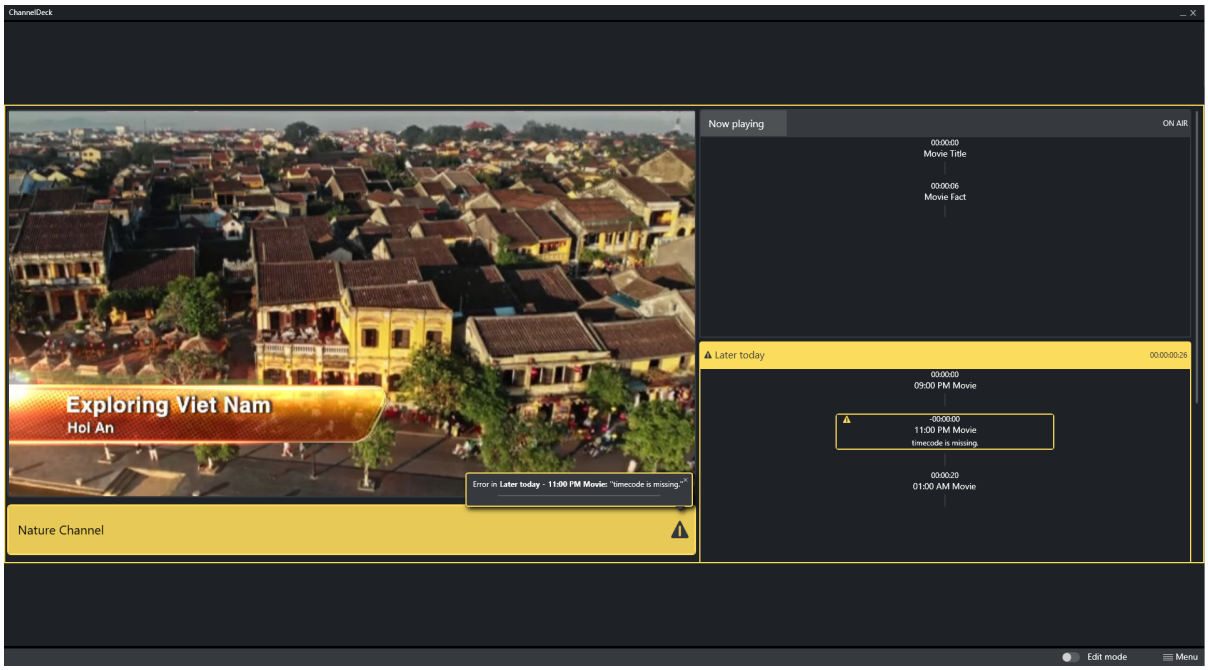
Type	Color	Description
Error		When a channel view and a primary event have **red** borders, an error is present in a primary that is currently being played out and needs to be addressed urgently.
Warning		When a channel view and a primary event have **yellow** borders, it means that primaries that are planned for take or have been played out, contain errors and should be addressed before a take.

Three example images of primaries with **errors** and **warnings**:

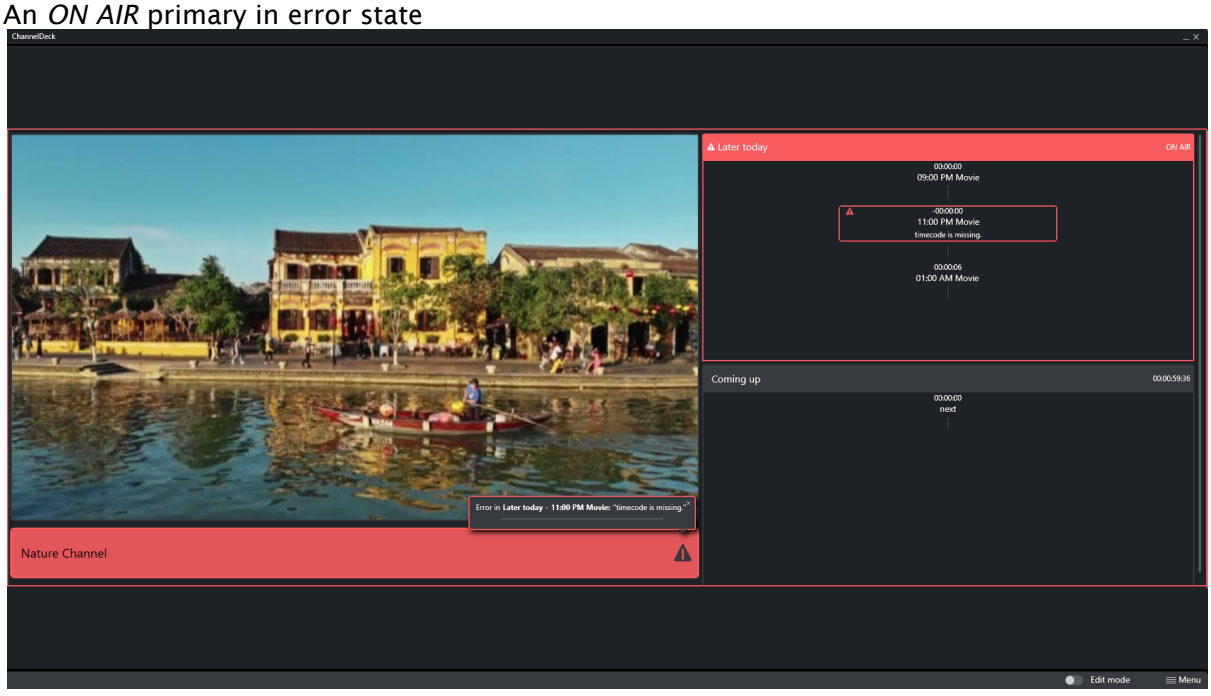
1. An *ON AIR* primary, and a planned take of a primary in a warning state with a warning message "Timecode is missing"

   > The "Display name" can be pressed to present or hide the error/warning log

Copyright © 2024 Vizrt

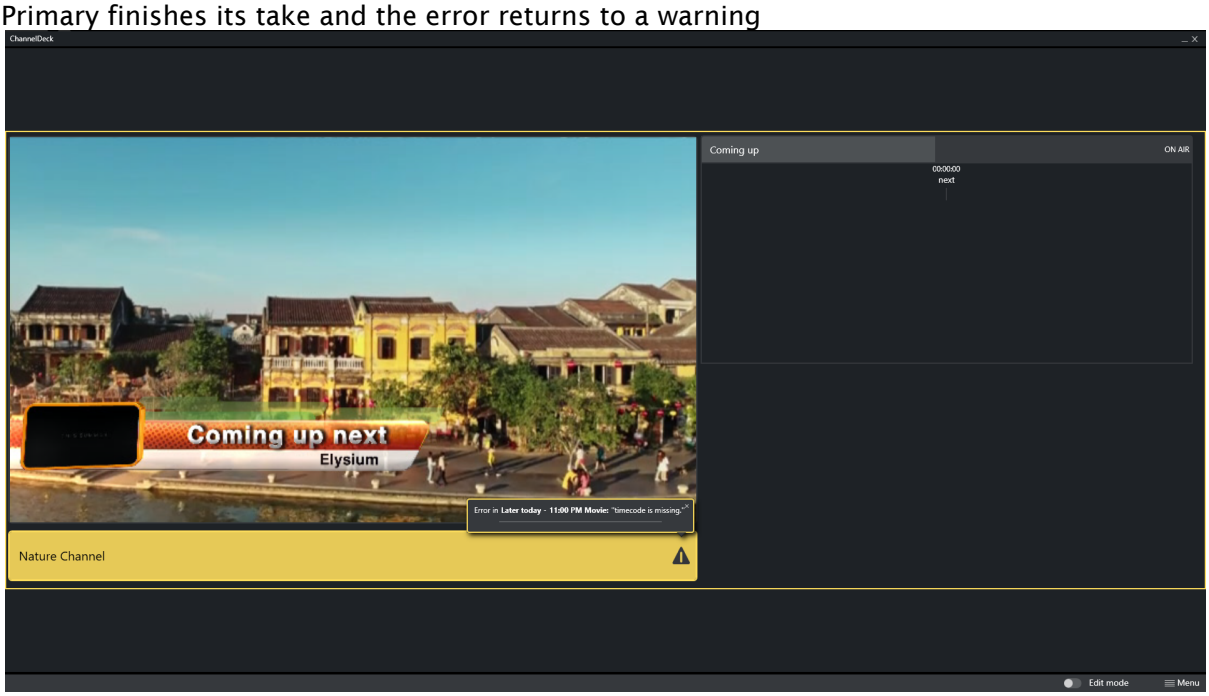


2. An *ON AIR* primary in error state



Copyright © 2024 Vizrt

3. Primary finishes its take and the error returns to a warning



15 Housekeeper

The Housekeeper is a context-aware data synchronization service, for pushing prepared pages either

- From scheduling/traffic department to Master Control Room
- From main to backup
- From one to many channels as a way to push campaign data.

You can configure the Housekeeper service to automatically synchronize data across any number of channels. Any channel can be registered as a source for the synchronization operation and set up with multiple synchronization groups as destinations.
For each group, it is possible to configure the data that is synchronized and the details about the destination channels.

- Configuring the Housekeeper Service

Copyright © 2024 Vizrt

15.1 Configuring The Housekeeper Service

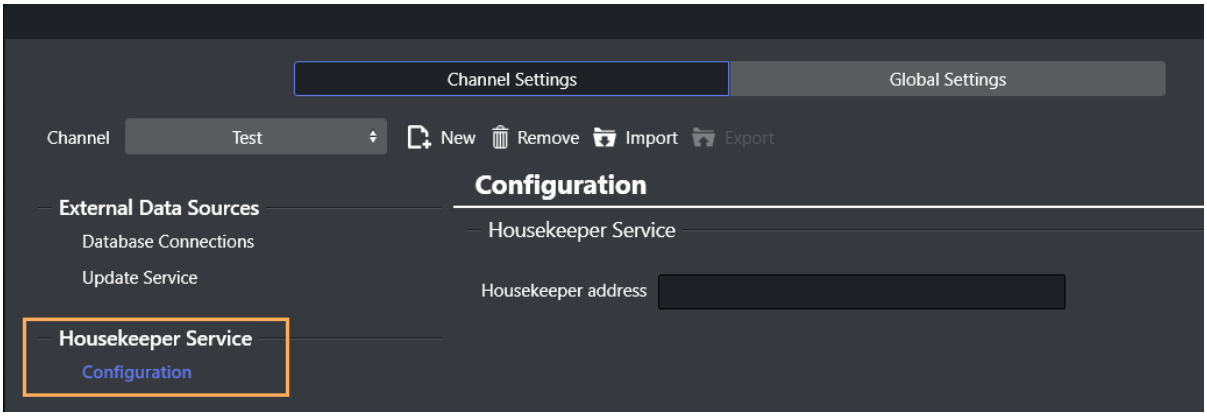
Housekeeper settings are made by navigating to **Viz Channel Branding client > Channel settings > Housekeeper Service > Configuration**.

- Service Configuration
- Channel Registration
  - To register a channel in Housekeeper Service
  - Troubleshooting
- Synchronization Groups
  - To create a synchronization group
  - To modify a synchronization group
  - To remove a synchronization group
- Troubleshooting Housekeeper Configuration

15.1.1 Working with Housekeeper

Service Configuration

You perform configuration with the Viz Channel Branding client application, per channel, in the **Channel Settings** menu.

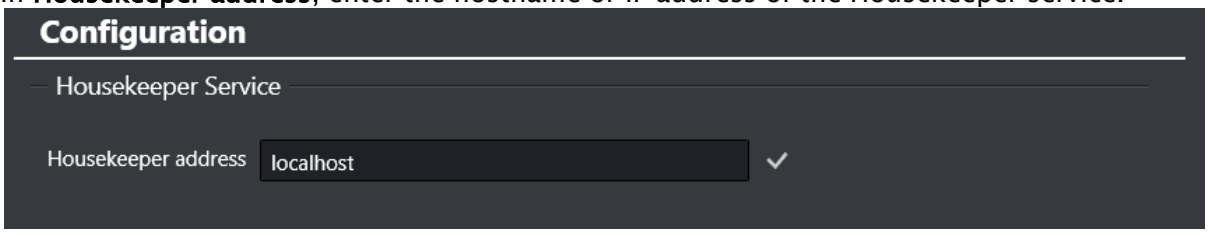


Channel Registration

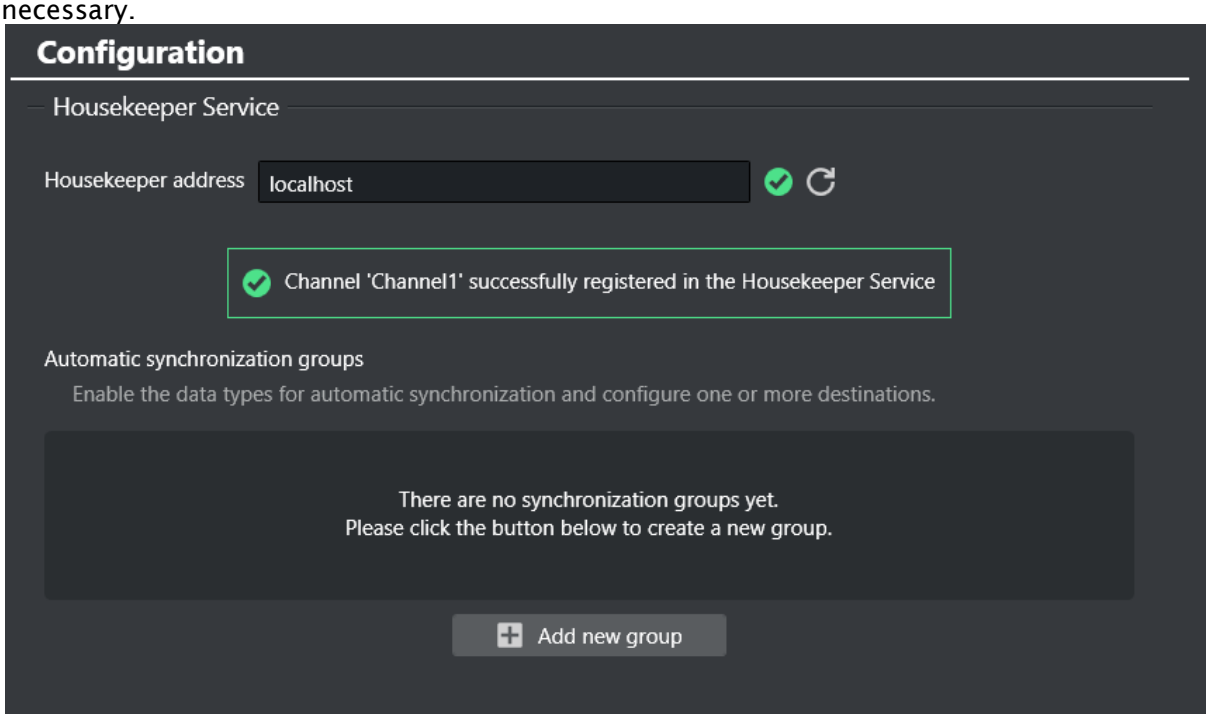
To set up the service, you must first register the current channel in the Housekeeper service.

To register a channel in Housekeeper Service

1. In **Housekeeper address**, enter the hostname or IP address of the Housekeeper service.



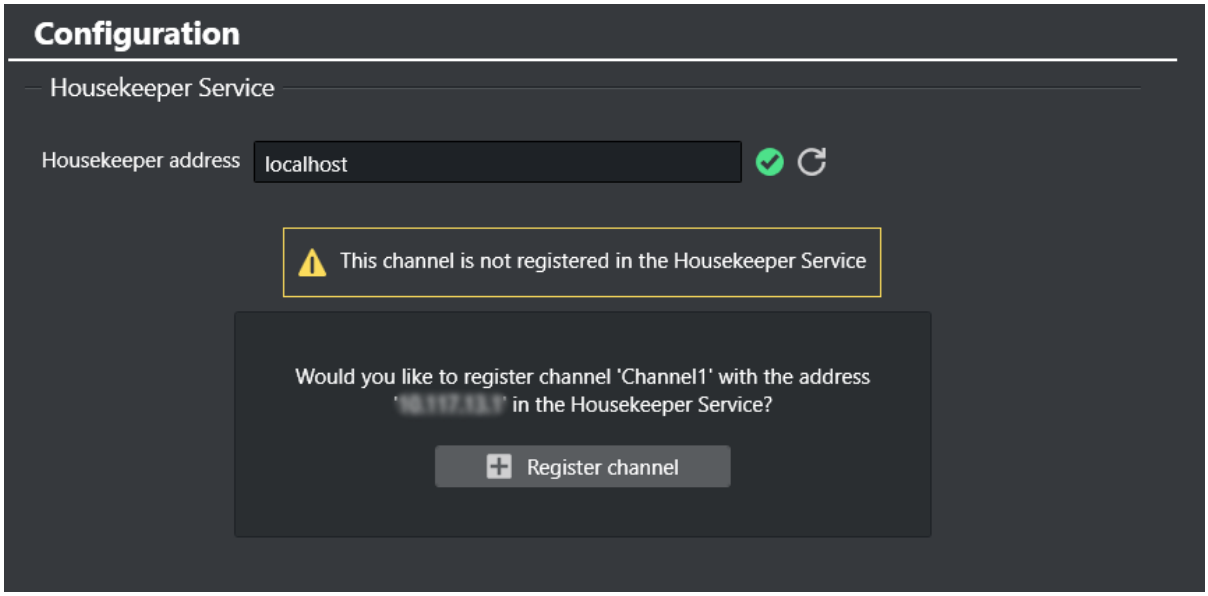
2. Click **Connect**.
   An attempt is made to connect to the Housekeeper service.
   If there are issues, see Troubleshooting.
3. If the the current channel is present in the Housekeeper service, *no further configuration* is necessary.



   A list of synchronization groups for the current channel is presented. Please now refer to section *Synchronization Groups* below.

4. If the current channel is not present in the Housekeeper configuration, a prompt indicates which address and channel will be registered.

Copyright © 2024 Vizrt

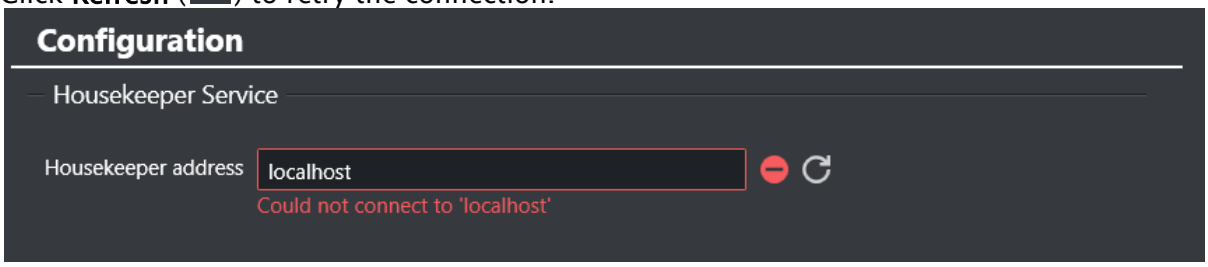


5. Click **Register channel**.
6. A confirmation message reports successful registration.

Troubleshooting

If the connection cannot been established, an error message is displayed.

- Click **Refresh** ( ) to retry the connection.



A common issue is an invalid address.

Synchronization Groups

After successfully registering the channel, it is possible to create any number of synchronization groups, each with a distinct configuration.

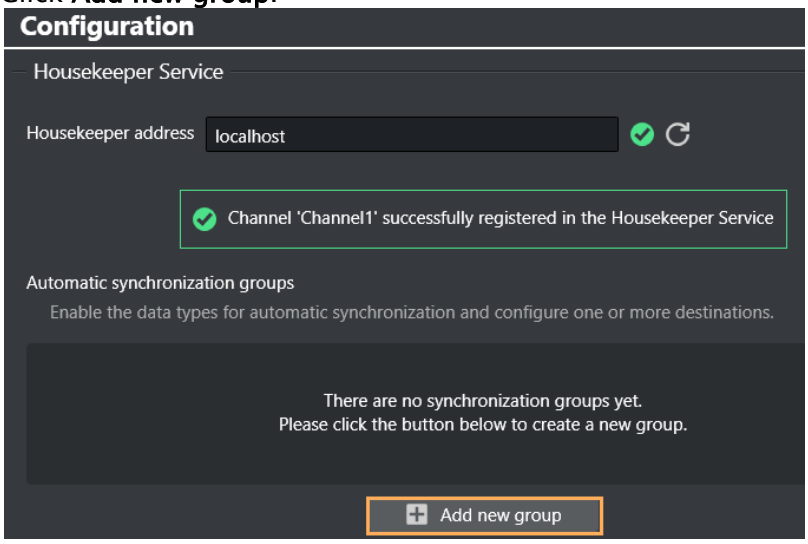
The Housekeeper service

- stores the created groups and initiates automatic synchronization of the chosen data types from the current channel, to the designated destinations
- monitors any alterations in the current channel and ensure automatic synchronization across all destinations.

Copyright © 2024 Vizrt

To create a synchronization group

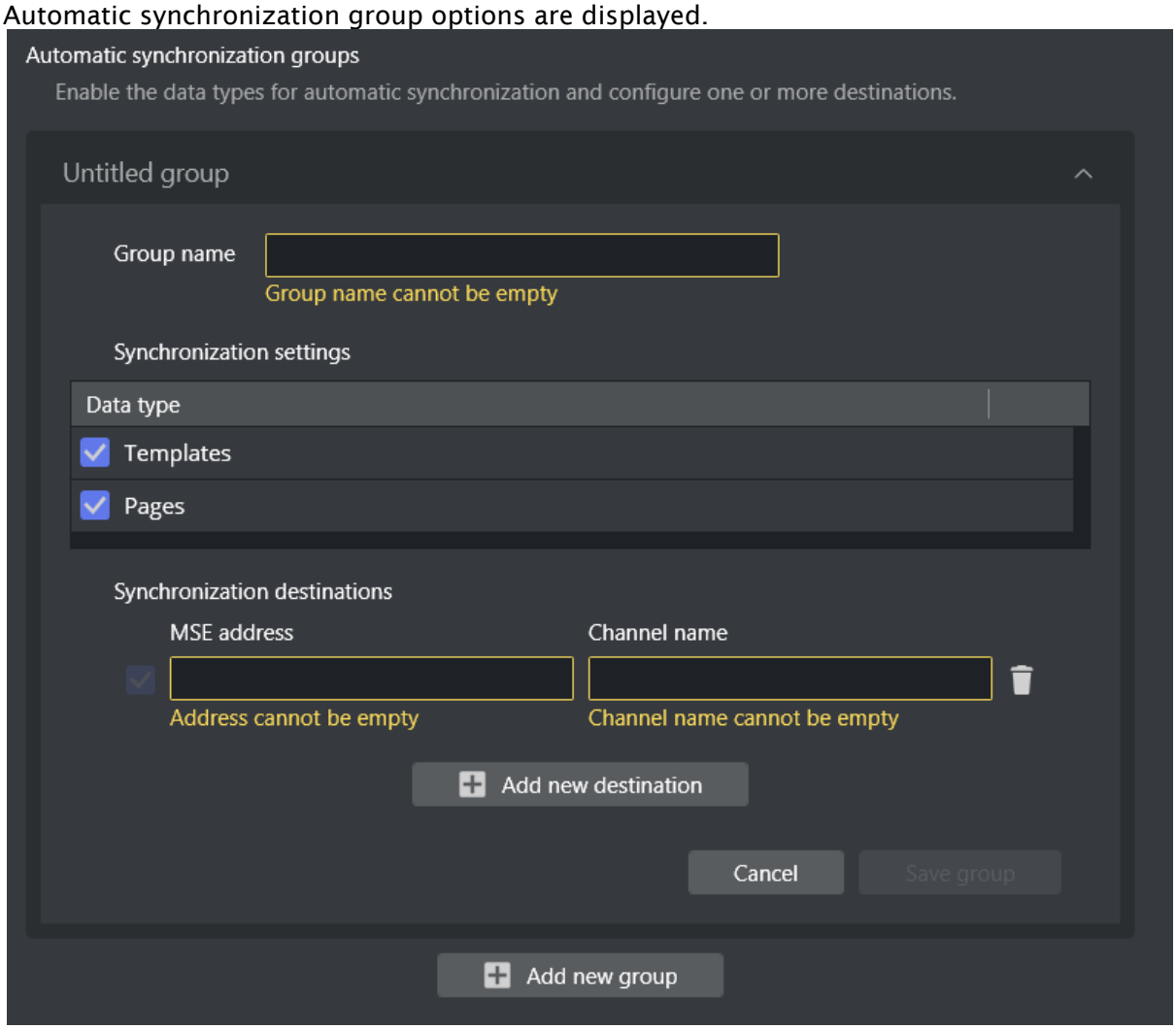
1. Click **Add new group**.



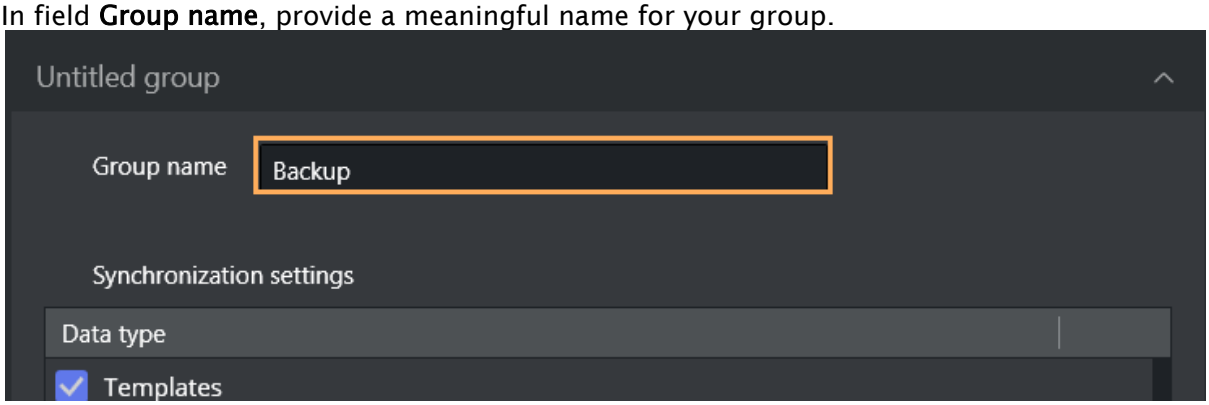
> **Note**: The list of synchronization groups is empty in the initial configuration.

Copyright © 2024 Vizrt

2. Automatic synchronization group options are displayed.

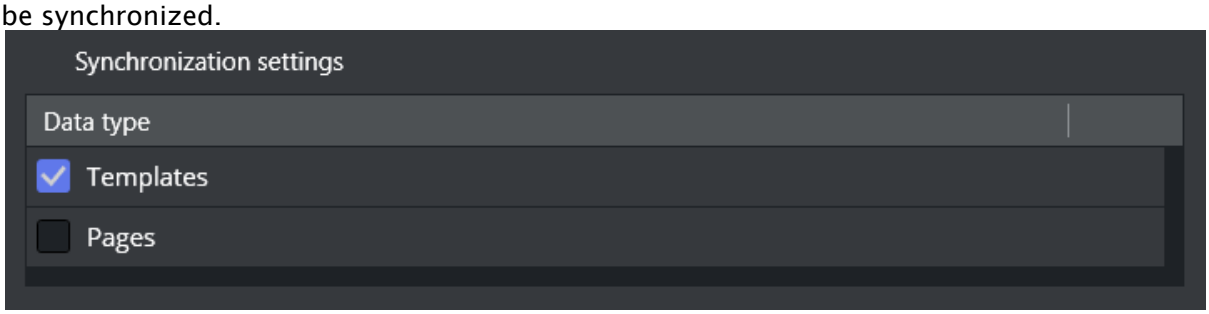


3. In field **Group name**, provide a meaningful name for your group.

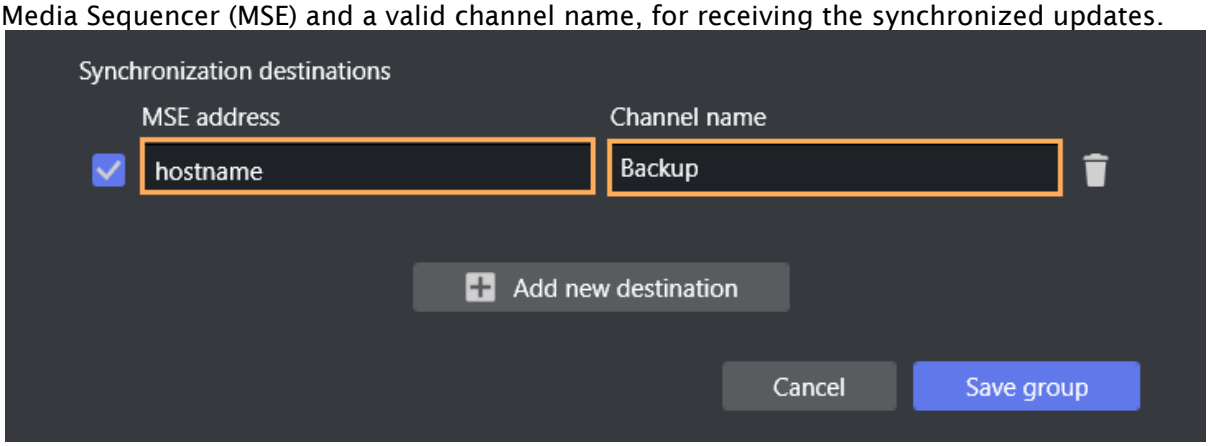


Copyright © 2024 Vizrt

4. (Optional) Under **Synchronization settings**, you can enable or disable the *data types* that will be synchronized.



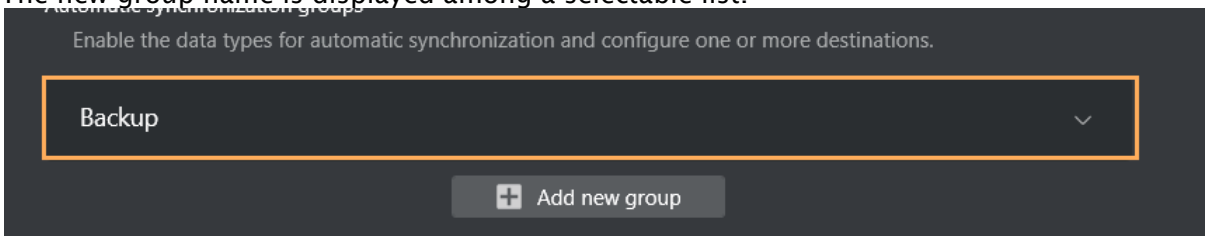
5. In field **Synchronization destinations** define the targets, by specifying the address to the Media Sequencer (MSE) and a valid channel name, for receiving the synchronized updates.



When **Synchronization destinations** and **Channel name** are valid, a checkbox is available to enable or disable the destination.

> ⚠ **Note**: Disabling (checkbox is unchecked) a destination halts synchronization for the destination.

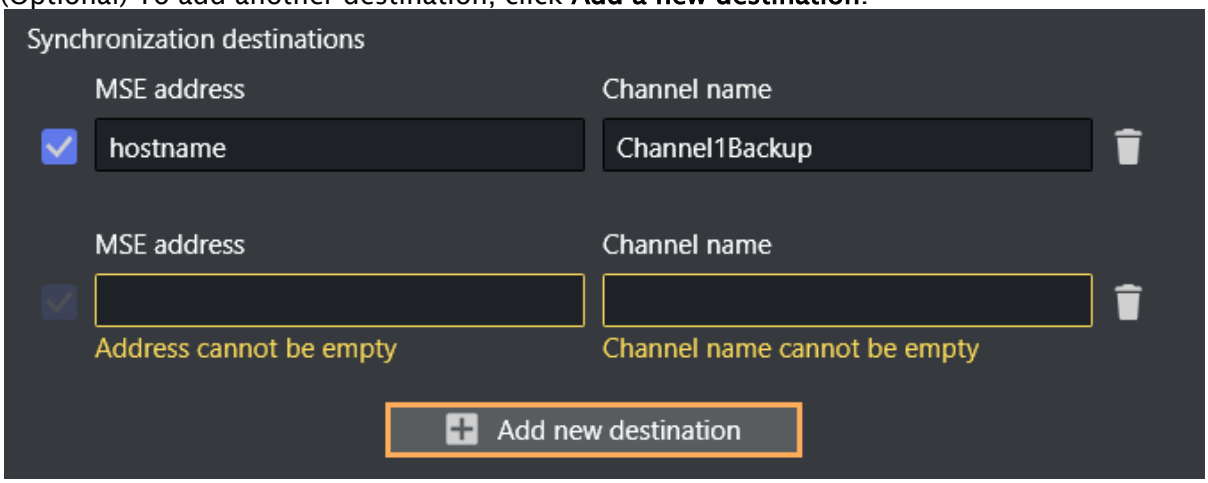
6. Click **Save group**.
The new group name is displayed among a selectable list.



The Housekeeper service now begins monitoring the current channel for any changes, to initiate any synchronization.

Copyright © 2024 Vizrt

7. (Optional) To add another destination, click **Add a new destination**.



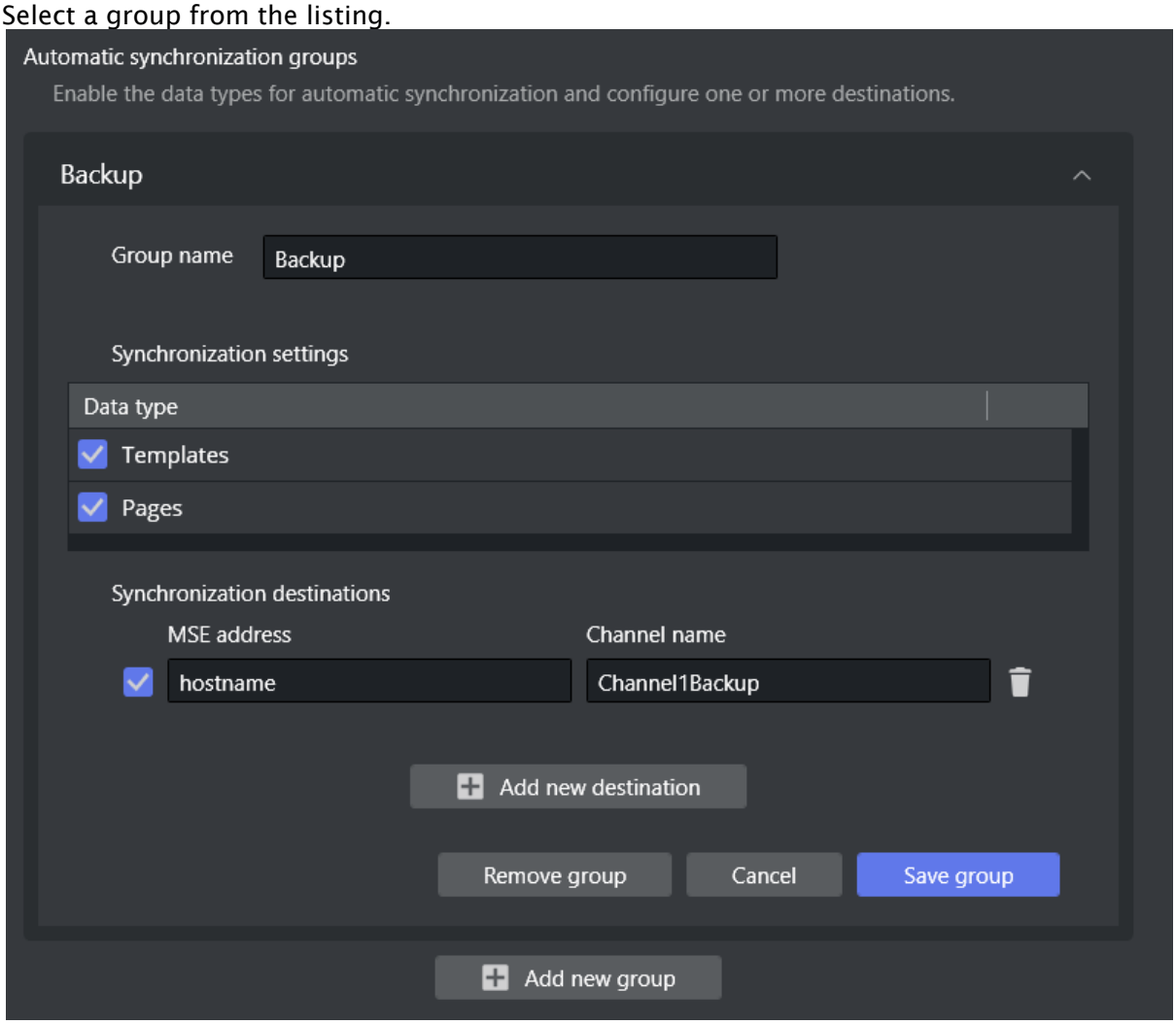
A new row for configuration is displayed. Repeat steps **2**-**6**.

To modify a synchronization group

You can browse or edit a synchronization group, the expander can be opened and all the fields updated and the changes saved.

1. Click the expander to open.

2. Select a group from the listing.



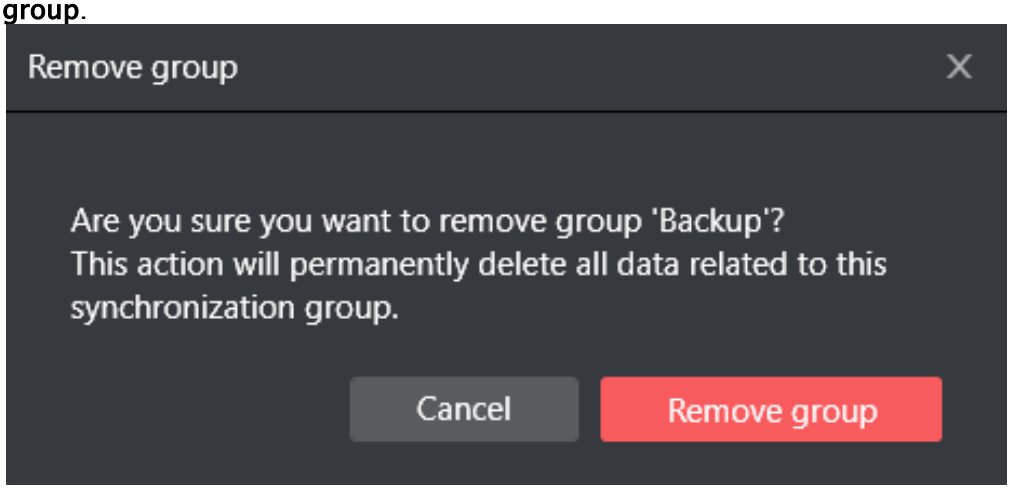
3. Make required modifications.
4. Click **Save group**.

To remove a synchronization group

1. Click the expander to open.
2. Select a group from the listing.
3. At the bottom of the form, click **Remove group**.

Copyright © 2024 Vizrt

4. Removing a group is an irreversible operation. From the confirmation dialog, click **Remove group**.



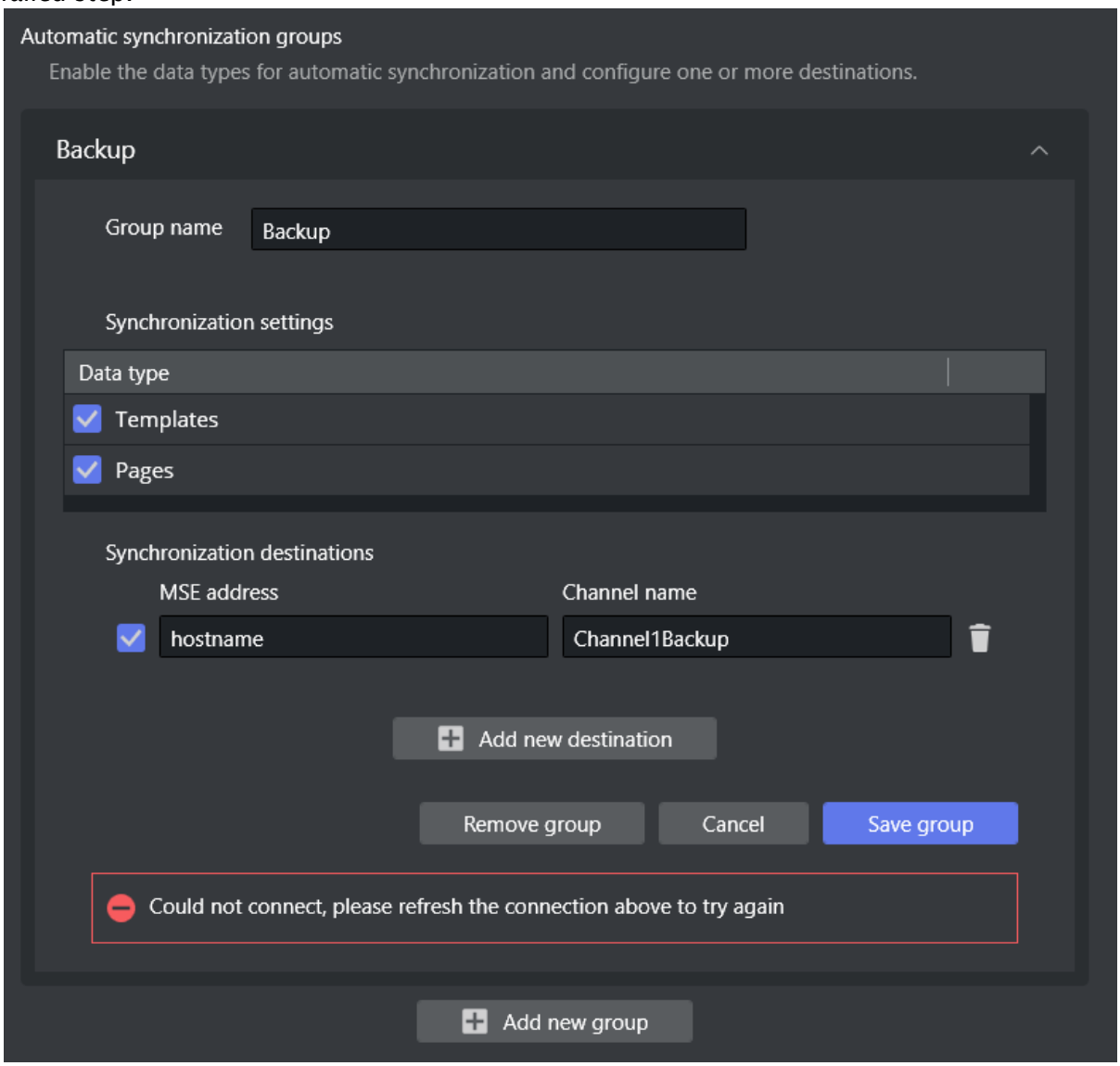
Once a group is removed, the Housekeeper service will cease all synchronization processes linked to that group's configuration.

> ⚠ **Note**: If you wish to *temporarily* halt the synchronization, it is recommended to *disable* the destinations by canceling the checkbox as described in step 5 above .

Copyright © 2024 Vizrt

Troubleshooting Housekeeper Configuration

- If any of the steps described above fail, you can restart or continue the operation from the failed step.



16 Traffic Department Workflow

This chapter covers common Traffic Department procedures performed before handing playlist control to the automation system in master control.

This chapter covers the following topics:

- Working with Templates and Pages
- Working with Actions
- Ticker Actions
- Working with Playlists
- Working with Video Clips
- Working with Rules
- Resolving Conflicts
- Generating the Channels File

See Also

- Master Control Workflow

Copyright © 2024 Vizrt

16.1 Working With Templates And Pages

This section covers the following topics:

- About Pages
- Importing Templates
- Creating Pages
- Previewing Pages
- Editing Page Duration
- Sending Templates and Pages

16.1.1 About Pages

The first step in populating a playlist with graphics is to import Viz Artist scenes into Viz Channel Branding system. These scenes become templates and are saved as an editable form known as a *Page*.

> ⚠ **Note:** All pages added to a playlist are added as secondary events.

A page can be added to the playlist, creating a standalone instance of that page. Changes to the added page do not affect the original page. Likewise, you can also create *rules* on pages, and each page resulting from the rule in the playlist is also standalone.

See Also

- Page Content Filling

16.1.2 Importing Templates

Templates are scenes imported from Viz Artist. Although the scene cannot be edited, information in the tab fields can be adjusted and pages (instances of the template) can be created that can be edited from the Template, see Creating Pages.

To import a Template

1. Click the Template Import button.
   - A tree representing the Viz Graphics Hub is displayed.
2. Browse for scene(s) to import, and do one of the following:
   - **Double-click** the scene to import it.
   - **Right-click** the scene and from the appearing context menu select **Import Template(s)**.
   - **Select the scene(s)** and click the **Import** button (lower-right).
   - **Select a scene folder**, right-click and from the appearing context menu select **Import Recursively** to import all scenes in that folder and sub-folders.
3. Click **OK**.
   - The imported templates now appear in the Templates view (left panel).

Copyright © 2024 Vizrt

16.1.3 Creating Pages

If the template contains editable elements, you can add text, images, and videos to it before adding your page to the playlist. Once a page is added to the playlist it is treated as a secondary event.

To create a Page

1. Select your channel.
2. Click the Templates, Pages and Pilot Data Elements button (left pane).
   All templates are displayed in the left pane.
3. Right-click the template and from the context menu select either **Create Page** or **Create Page As**.
   The **Create Page** option will create a page with the same name as the Template selected, while **Create Page As** allows you to create a page with a new name you select.
4. Enter a name and click **OK**.
   The page is now added to the Page list, seen below the Templates list.

See Also

- Page Content Filling

16.1.4 Previewing Pages

You can preview all pages, regardless of whether or not they are editable.

To previewing a Page

1. Select your channel.
2. Click the Templates, Pages and Pilot Data Elements button (left pane).
   All templates and pages are shown.
3. Double-click the page you wish to preview.
   The page is shown in the preview window.

16.1.5 Editing Page Duration

You may edit the duration field for a page by setting a default duration before adding it to your playlist.

To edit page duration

1. Select your channel.
2. Click the Templates, Pages and Pilot Data Elements button (left pane).
3. Select the page you want to edit.
4. Double-click the **Duration** field.
   The Duration field becomes editable.
5. Enter the desired duration and press **Enter.**

See Also

Copyright © 2024 Vizrt

- Resolving Conflicts

16.1.6 Sending Templates and Pages

Templates and pages can be manually or automatically sent to the MCR setup or another machine for backup purposes. Both options require that Media Sequencer host for the remote machine is defined in the **Remote Hosts** section under Working with Templates and Pages.

To manually send templates and pages to a remote host

1. Select your channel.
2. Click the Templates, Pages and Pilot Data Elements button (left pane).
3. Right-click the relevant template and select **Send Template(s)** and **Send Page(s)**.
   This will send templates and pages to the remote Media Sequencer host.

To automatically send templates and pages to a remote host

1. Open the Channel Settings, and select Update Settings.
2. Under the Update Options section check the **Send Templates and Pages to remote host(s)** option.
   When the Playlist Importer updates the playlist, templates and pages are sent to the remote Media Sequencer host.

See Also

- Sending Playlists to the Master Control

Copyright © 2024 Vizrt

16.2 Working With Actions

This section covers the following topics:

- Defining Actions
- Referenced Actions

16.2.1 About Actions

Actions, such as *Viz Commands* and *GPI Functions*, can automate certain tasks. For example, Viz commands can be used to:

- Take specific pages on air.
- Clear specific render layers.
- Send Stop, Continue, Start and other commands to a specific render layer.

Viz commands are created globally, and can be used on all defined channels.

> **Note:** All commands must first be defined using the Viz Channel Branding client, in order to be available as secondary events.

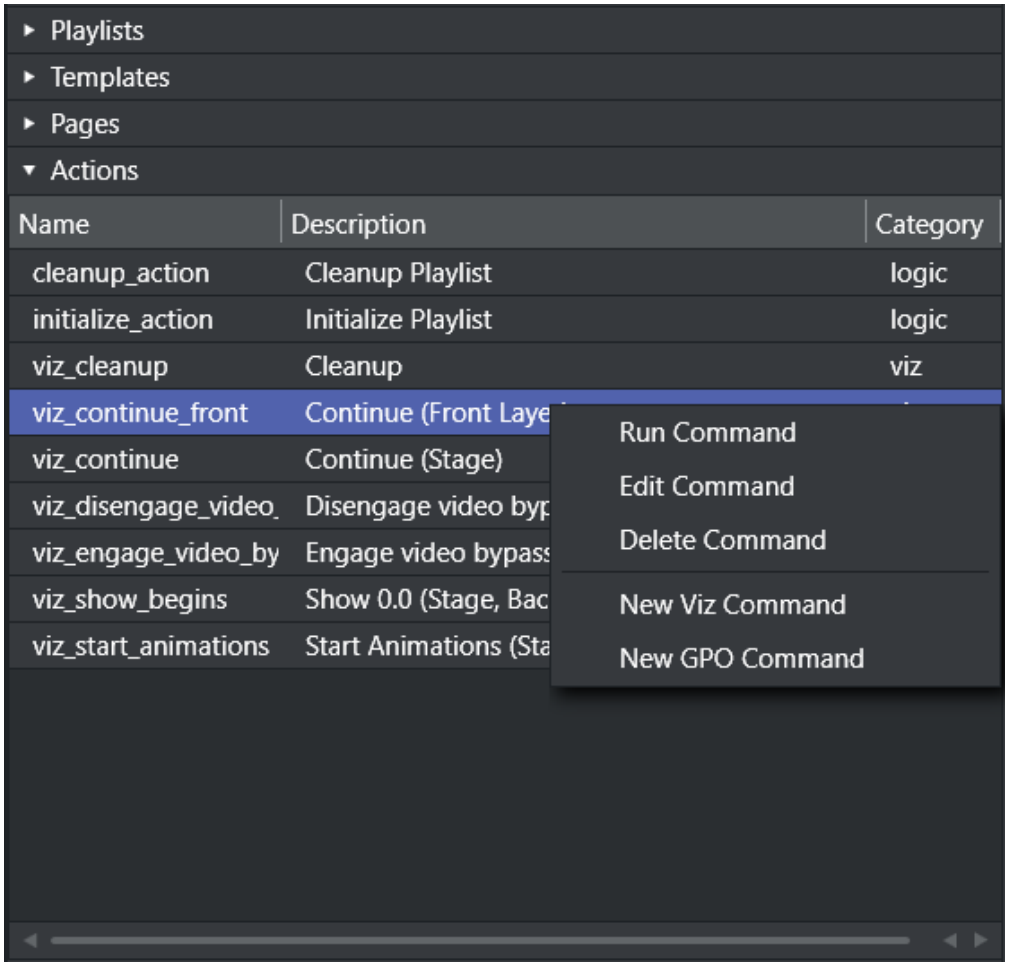
Defining Actions

To define secondary event actions

1. Click Actions

Copyright © 2024 Vizrt

- Predefined commands are displayed:



2. Right-click one of the actions and select **New Viz Command** or **New GPO Command**
3. In the dialog box enter:
   - **Description** is the descriptive term that is seen in the Actions list, as well as any drop-down list in the Rules and GPI settings panels.
   - **Command** is the actual command line sent.
   - **Name** is a pre-generated name based on type selected (for example `viz_newcommand`) that is verified as unique upon generation and may be changed into something more descriptive. Name is the value used for scheduling (as "page name"').
4. Click **OK**.

> ⚠ **Note:** For more information about Viz Commands, consult the *Viz Command* documentation included with your Viz Engine/Viz Artist installation.

See Also

- Ticker Actions

Copyright © 2024 Vizrt

Referenced Actions

Each occurrence of a Viz or GPI action in a playlist is *independent*, and editing it from within the playlist will only change that specific instance.

However, you can set up a playlist in such a way as to allow for last-minute global changes by associating referenced actions to primary events. In such a use case, an action is associated by reference to primary events in the playlist. When the action has been changed, either in the actions list (left panel) or by double-clicking on one of the referenced actions in the actual playlist, all occurrences in all playlists in the current channel are changed both retroactively and henceforth.

To create a referenced action

1. Use an existing action or create a new one.
2. Press and hold **Alt + drag and drop** the action to the playlist.
   The action will receive the following icon: .

Copyright © 2024 Vizrt

16.3 Ticker Actions

Use Ticker actions to automate the playout and control of tickers.

16.3.1 Working with Ticker Actions

In order to automate playout, you must first export available ticker actions from your Viz Channel Branding Client in the master control room (MCR). When ticker actions are exported they can be imported into a system in the scheduling department. This lets you schedule ticker actions as secondary events and test the playout of tickers from scheduling.

This section covers the following topics:

- Export Ticker Actions
- Import Ticker Actions
- Scheduling Ticker Actions Manually
- Scheduling Tickers Actions Automatically
- Playing Tickers Manually

See Also

- Actions
- Working with Actions

Exporting Ticker Actions

This procedure takes the ticker action's structure in the Media Sequencer and virtually maps its XML structure (without the logic) for use by Viz Channel Branding Client in scheduling mode. The export is performed on the Media Sequencer where the real ticker resides, in the MCR.

To export Ticker actions

1. Click the Actions menu on the Main Menu.
2. Select **Ticker > Export**.
3. Enter a file name and click **Save**.

Importing Ticker Actions

This procedure imports a virtual version of the ticker actions only (without its logic). Before you can import ticker actions you must Exporting Ticker Actions.

To import Ticker actions

1. Click the **Actions** menu on the Main Menu.
2. Select **Ticker > Import**.
3. Select the file to import and click **Open**.

Copyright © 2024 Vizrt

dayswitch_action	Switch To Next Playlist	logic
LowerThird Main	LowerThird Main	ticker
program_Main	Program Main	ticker
RSS-Ticker Main	RSS-Ticker Main	ticker
Side Main	Side Main	ticker
ticker_system_Main	Ticker System Main	ticker
viz_cleanup	Cleanup	viz
viz_continue_front	Continue (Front Layer)	viz

Ticker actions are displayed in the left pane.

IMPORTANT! The Ticker actions that are read from the imported Ticker XML-file need to include **ON** or **OFF** (capital or lowercase will both work) in the ticker button name for the action to be imported correctly. For example, ticker buttons named *All ON*, *All off* and *Command ON* will all import correctly, while buttons named *Engine IN* or *Engine OUT* will **not** import correctly.

Scheduling Ticker Actions Manually

Ticker actions can be manually added to the playlist as secondary events in the same way as pages and other actions.

To manually add a Ticker action to a playlist

- Follow the same procedure as to Manually Add a Page to the Playlist.

Scheduling Tickers Actions Automatically

Ticker actions can be automatically added to the playlist as secondary events in the the same way as pages and other actions. You can also add actions through import rules.

Note: It is possible to map to all actions via a schedule by referring to the action's name while mapped as a secondary event.
- See the **secondary_page** setting in the **Table Column**.

To automatically schedule a ticker

- Use the name of the ticker actions in your schedule.

To define a rule on a ticker command

1. Select your channel.
2. Click the Rules button.
3. Continue with the procedures in the Working with Rules section.

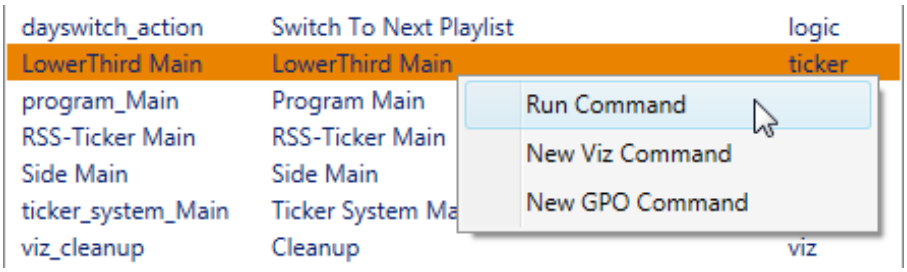
Copyright © 2024 Vizrt

Playing Tickers Manually

This procedure is performed on Viz Channel Branding client in the MCR where the real tickers reside.

To manually play tickers

1. Select your channel.
2. Click the Actions button.



3. Right-click the command you want to play and select **Run Command**.

16.4 Working With Playlists

Viz Channel Branding's main task is to import schedules, and consequently create, update and execute programming playlists. A playlist is composed of primary events (programs), which in turn can contain associated secondary events (pages and actions).

Playlists can be created as a continuous playlist or as a playlist per broadcast day, as selected via the settings for a specific channel.

Each new broadcast day playlist will have a different date, regardless of the day on which it was created. For example, if you create two playlists today, the first one will be dated today by default, and the next one will be dated tomorrow.

This section covers the following topics:

- Import a Schedule
- Creating a Playlist
- Deleting a Playlist
- Adding Primary Events to a Playlist
- Adding Secondary Events to a Playlist
- Deleting Events from a Playlist
- Updating a Playlist
    - Full Update
    - Full Update Without Event Deletion
    - Update Times
- Exporting a Playlist
- Playlist State Definitions
- Playlist Errors

16.4.1 Import a Schedule

The system can import a schedule from an external source. However, certain parameters must be configured the first time you import a schedule.

Schedules are generally composed by the traffic departments. Many scheduling systems can create schedules with both primary events and secondary events - a schedule plus associated pages. the solution also supports user-defined rules for automatic insertion of secondary events.

The following schedule file formats can be imported as playlists:

- Importing from Text (*.csv, *.txt, *.vicc)
- Importing from XML (*.xml)
- Importing from Excel (*.xls, *.xlsx)
- Harris ADC-100 (*.lst, *.rcv)

If this is the first time you are importing a schedule, you must first complete the procedures in the following sections:

- Working with Playlists
- Working with Rules

Copyright © 2024 Vizrt

To import a Schedule

1. Select your channel.
2. From the toolbar, click the **Import schedule** button.
3. Select the file to import and click **Open**.

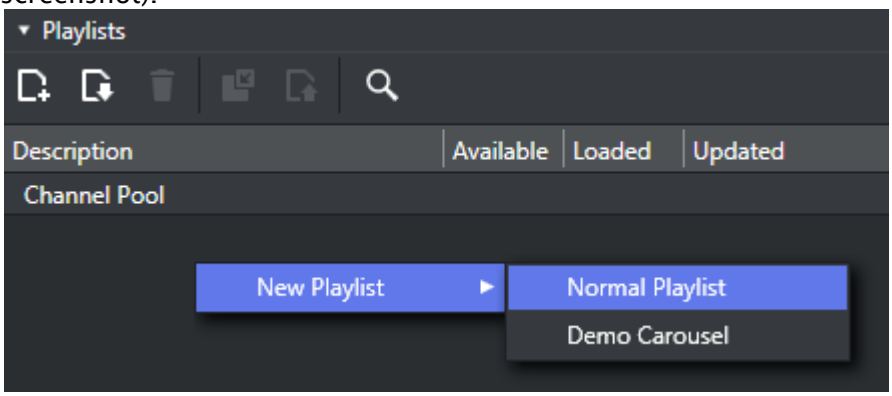
16.4.2 Creating a Playlist

The solution can create a playlist based on an imported schedule or create a playlist manually.

> **Note:** The most common and preferred method is to import schedules from scheduling systems.

To manually create a playlist

1. Select your channel.
2. Click the Playlists pool. All existing playlists for the selected channel are displayed.
3. EITHER Click the **New Playlist** button (top left). or the context-sensitive menu option (see screenshot).



4. A new row representing the new playlist is added.

16.4.3 Deleting a Playlist

To manually delete a playlist

1. Select your channel.
2. Click the Playlists pool. All existing playlists for the selected channel are displayed.
3. Select the playlist and do one of the following:
   - Right-click the playlist item to show context-sensitive menu
   - Click the **Delete Playlist** button on the toolbar.
4. Click **Yes** to confirm.

Copyright © 2024 Vizrt

16.4.4 Adding Primary Events to a Playlist

To manually add primary events to a playlist

1. Select your channel.
2. Click the Playlists pool. All existing playlists for the selected channel are displayed.
3. Open your Playlist.
4. Right-click in the open playlist and from the context menu, select **Add Primary Event**.

16.4.5 Adding Secondary Events to a Playlist

Secondary events can be pages and video clips, or a combination of both.

To manually add a page to the playlist

1. Select your channel.
2. Click the Playlists pool. All existing playlists for the selected channel are displayed.
3. Open your playlist.
4. Click the **Templates**, **Pages** or **Video Clips** pool.
5. Drag and drop a page or video clip on top of a primary event.

   > ⚠ **Note:** If you press **ALT** when dragging a page onto the playlist, the resulting secondary event will be associated to the original page by reference only.

6. The secondary event appears below the primary event. Any several secondary events will be sorted according to their offsets.
7. The most important fields to edit are generally the **Offset** and **Duration** fields.

   > ⚠ **Note:** If a field is not editable, you can check the relevant column's **Edit Events** check-box seen in the mapping table.

16.4.6 Deleting Events from a Playlist

To manually delete events from a playlist

1. Select your channel.
2. Click the Playlists button.
   All existing playlists for the selected channel are displayed.
3. Open your playlist.
4. In the playlist, right-click the relevant event and select **Remove Selected Event(s)** from the appearing context menu.

16.4.7 Updating a Playlist

If there are changes to be made to a playlist, you have the following options:

Copyright © 2024 Vizrt

- Full Update
- Full Update Without Event Deletion
- Update Times

> **Note:** You can schedule automatic updates in the **Update Settings**.

Full Update

The **Full Update** option merges the current playlist with changes made to the schedule.

Priorities and updates of existing secondary events are determined by configurations done in the **Full Update Options** for **Secondary Events** section.

See also the Working with Rules and Resolving Conflicts sections.

Full Update Without Event Deletion

The **Full Update Without Event Deletion** option merges the current playlist with changes made to the schedule, without deleting existing primary events.

Update Times

The **Update Times** option is for imported schedules only. It is an update the begin time, duration and offset from changes that have been made to the schedule for primary events. Any primary event additions or deletions are ignored. Only times are updated.

16.4.8 Exporting a Playlist

To export a playlist

1. Select your channel.
2. Click the Playlists button. All existing playlists for the selected channel are displayed.
3. Open your playlist.
4. From the toolbar, click the **Export Playlist** button. If the original playlist was LST, then the exported playlist can be merged with the original file.

For all other formats, you are prompted to save the file and assign a name in the relevant format.

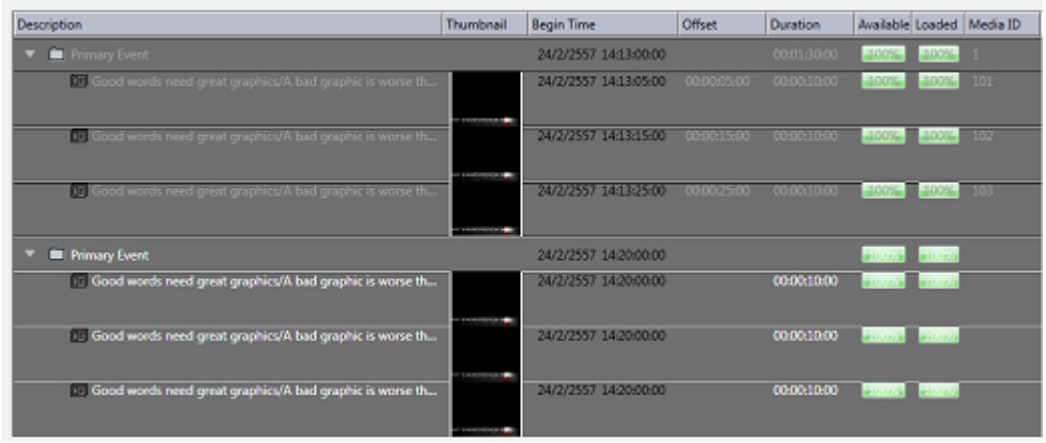
16.4.9 Playlist State Definitions

The playlist currently has six states for elements that reside inside the playlist. Each of these states uses a color on the playlist elements to provide a visual indication of their state:

- **Normal state**: The state of an un-played element. Elements are shown in *White* text.

Copyright © 2024 Vizrt

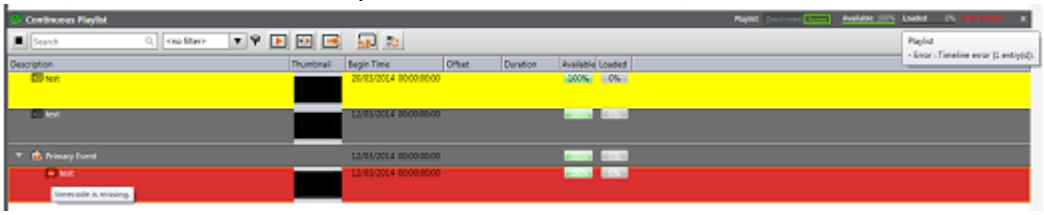
- **Played state**: The state of an element that has been played out at least once. Elements in are shown in *Gray* text.



- **Cue state**: The state of an element in a group has been called by MSE, to prepare for playing. These elements are shown in *Yellow*.
- **Pre-queue state**: The state of a next element that will start playing within 5 seconds. Elements are shown in *Orange*.

- **Playing state**: The state of an element that is currently playing. These elements are shown in *Green*. The cue state of a playlist that is activated by an *external trigger* has a *yellow* arrow in front of its element, otherwise the playing state has a *green* arrow in front. External triggering can be a VDCP-signal from the automation system, for example.

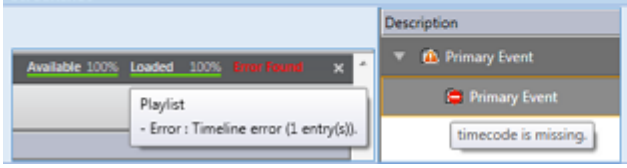
- **Error state**: The state of an element where it has been set incorrectly and cannot be played. Elements in error are shown in *Red* and will also have a tool-tip indicating what caused the error, if the error condition provides this information, which most errors do.



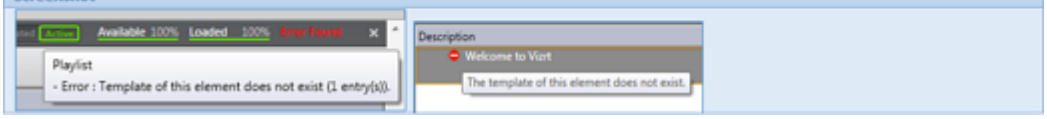
16.4.10 Playlist Errors

Playlist Errors in the playlist will most likely stop the playlist instantly or make the playlist work incorrectly. All indicated errors should therefore be treated as **severe**. The cause of the error must be investigated and corrective action must be taken to prevent the error from happening again. Errors can occur if the playlist was set incorrectly, if the elements residing in the playlist aren't created correctly, or as a result of other circumstances. Errors are separated into six groups as shown in the list below. There is one additional error type if the MAM system Viz One is enabled:

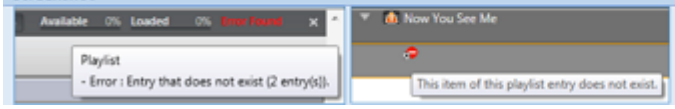
- Timeline error:

Error Type	Timeline error
Description	This error occurred because a timeline in the playlist isn't defined for element(s). To fix this, user need to define a timecode for each element correctly such as a begin time, offset, duration, etc. The errors that usually occurred for this type of error are timecode has already passed, timecode is missing.
Screenshot	

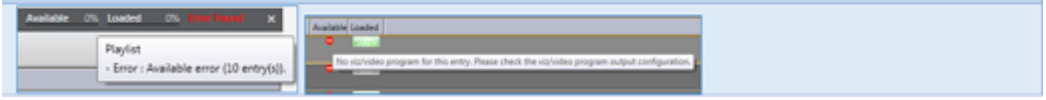
- Template does not exist:

Error Type	Template of this element does not exist
Description	This error occurred because the master template of elements inside the playlist is missing.
Screenshot	

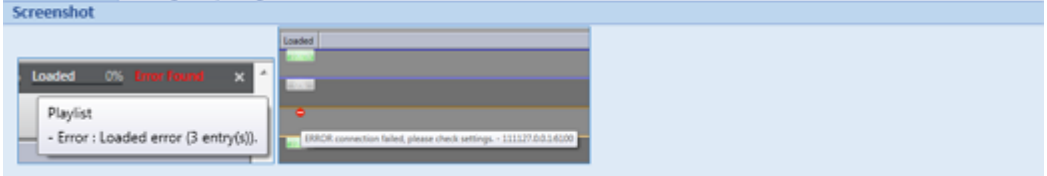
- Entry does not exist:

Error Type	Entry that does not exist
Description	This error is occurred because there are some entries missing in any elements inside the playlist. For example, an element which created by a page as a reference element then refers to that page. If the page has been removed then this error is occurred as it has no entry which it refers to.
Screenshot	

- Available error for playout devices:

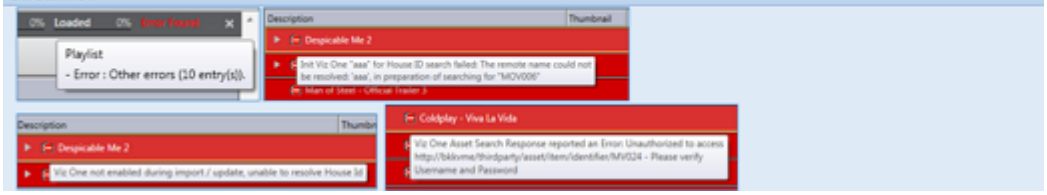
Error Type	Available error
Description	This error occurred when the **Playout Devices** isn't set correctly in a setting form.
Screenshot	

- Loading error:

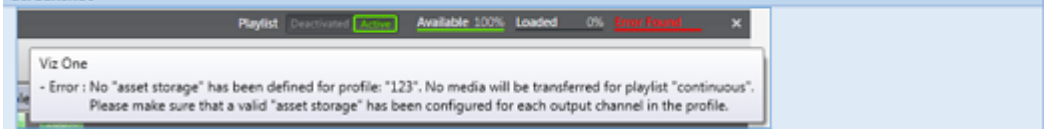
Error Type	Loaded error
Description	Loaded error occurred when the playout device was set with an incorrect host. After that, the client try to load the scene into Viz engine by using an initialize command.
Screenshot	

Copyright © 2024 Vizrt

- Other error types:

Error Type	Other Errors
Description	Other errors aren't specific errors it can be occurred by any reason which is not related to above cases.
Screenshot	

- Viz One specific errors: A Viz One Error is an error related to an asset storage. There are specific details on the solution in the tooltip.

Error Type	Error
Description	This error occurred because of an asset storage isn't defined. To fix this, go to **Settings** > **Playout Devices** and click edit at the defined **Viz Engines** and **Video Devices**. A configuration form with a drop down box for selecting an **Asset Storage** will appears.
Screenshot	

See Also

- Working with Templates and Pages
- Working with Video Clips
- Master Control Workflow on how to activate, initialize and manually take events On Air
- Working with Actions as secondary event actions in a playlist

Copyright © 2024 Vizrt

16.5 Working With Video Clips

Viz Channel Branding integrates with Viz One which allows you to add video clips to your graphics, and graphics to your video clips. Graphics with video clips can be added to the playlist as secondary events.

In order to preview video clips locally, you must install and configure the relevant codecs, see Video Preview Configuration. In order to work with graphics and video clips together, a Preview Server must also be configured and running.

This section covers the following topics:

- Preview of Video Clips
- Video Clips in Playlists
- Video Clips and Graphics

16.5.1 Preview of Video Clips

Video clip preview in Viz Channel Branding Client is based on a proxy version streamed from Viz One.

Note that in order to preview video clips locally you need to install and configure the relevant codecs, see Video Preview Configuration.

to preview a video clip

1. Click the Clips button.
2. Search for a video.
3. Double-click the video to preview it.

Alternatively, you can:

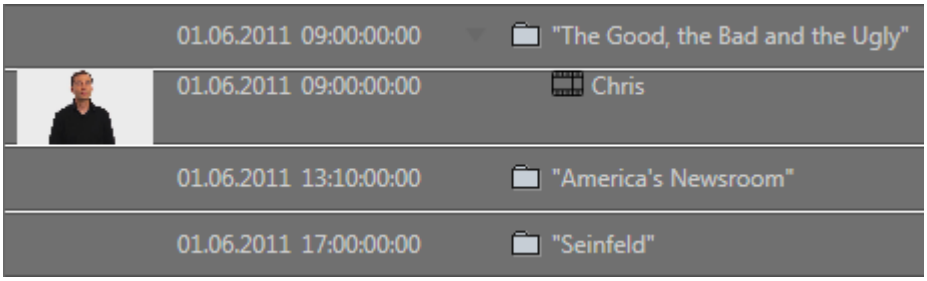
1. Click the Playlists button and open your playlist.
2. Search for a video.
3. Double-click the video to preview it.

16.5.2 Video Clips in Playlists

Note that video clips may be manually added to the playlist as **secondary events**; however, it is more common to import clips from the schedule.

Copyright © 2024 Vizrt

To add a video clip to the playlist



1. Select your channel.
2. Click the Playlists button and open your playlist.
3. Click the **Video Clips** button.
4. Search for a video.
5. Drag and drop your video placing it as a secondary event.
6. Double-click the video to preview it in the Timeline Editor.

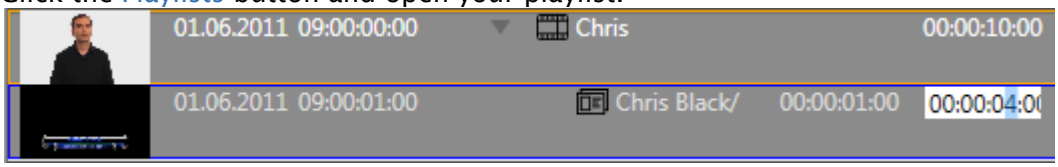
16.5.3 Video Clips and Graphics

Viz Channel Branding supports two ways of combining graphics and video. You can add graphics on top of a video clip, or you can add video to graphics.

Note that in order to work with graphics and video clips together you also need to have a Preview Server configured and running.

To add a graphic to a video clip

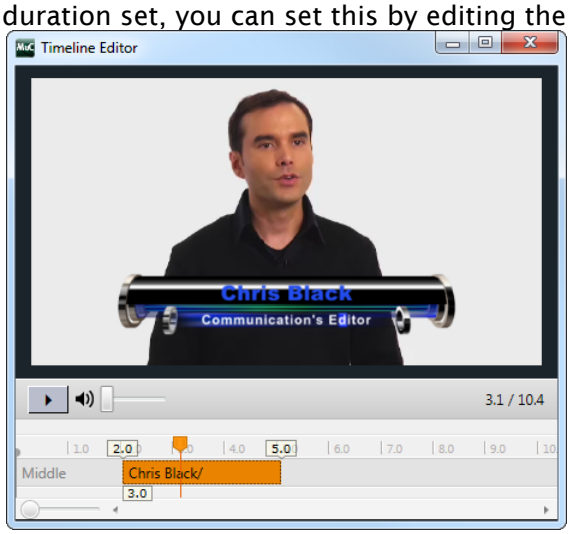
1. Select your **channel.**
2. Click the Playlists button and open your playlist.



3. Add a video clip as a secondary event (see Add Video Clips to the Playlist).
4. Add a page to the video clip (see Manually Add a Page to the Playlist). This adds the page as a sub element of the video clip, grouping them together. If the page does not have a default

Copyright © 2024 Vizrt

duration set, you can set this by editing the page's **Duration** column.



5. Double-click the video clip to open the Timeline Editor. This lets you adjust the offset and duration of the graphics.

To add a video clip to a graphic

1. Select your **channel**.
2. Click on the Page Pool section.
3. Open your template or page to open the Page Editor.
4. Select the tab field that will hold the video. This opens the video search frame.
5. Search for and select (for example double-click) a video. If the clip channel is set to automatically play the video you should be able to see a proxy version of the video.
6. Click **Save As** and enter a new page name.

See Also

- Page Editor
- Timeline Editor

Copyright © 2024 Vizrt

16.6 Working With Rules

For imported and subsequently updated playlists, you can create sets of rules that govern how pages can be automatically inserted into certain events within a playlist. Rules are applied when working with schedules or updating playlists from an external source. Any number of conditions can be applied to defined rules for exact filtering.

When rules are defined and marked as active, the following actions will automatically be applied upon receiving or updating playlists:

When schedules are received or changed:

- For each event in the schedule, the rules engine will try to match each rule against the current event read.
- If the rule matches it will then apply the action/page specified in the rule; if not, no action will be taken and the event will be inserted in the playlist unchanged.

Repeat this until all events are read and processed.

This section presents:

- The Rules UI
- Defining Rules for Conditions and Actions
- Rules Criteria and Conditions
- Back up and Restoring Rules

16.6.1 The Rules UI

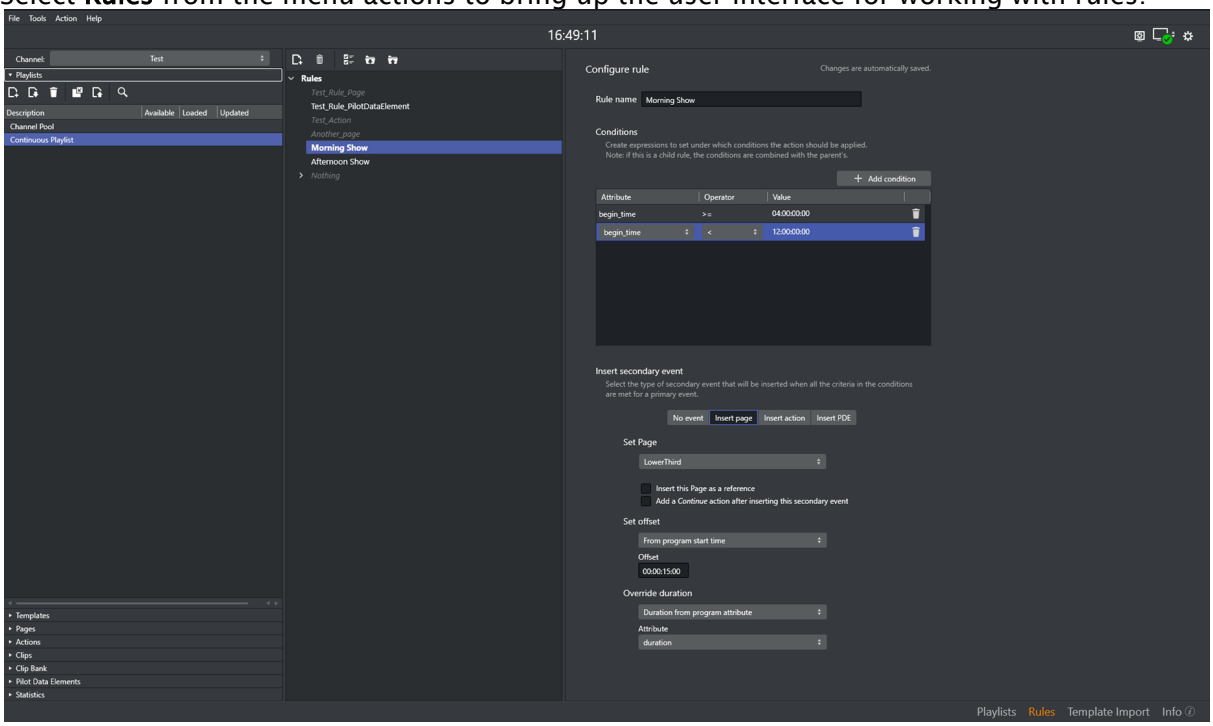
Rules Overview

To create, delete or change the definitions of a rule

1. From the main menu, ensure that the channel you want to work with is selected.

Copyright © 2024 Vizrt

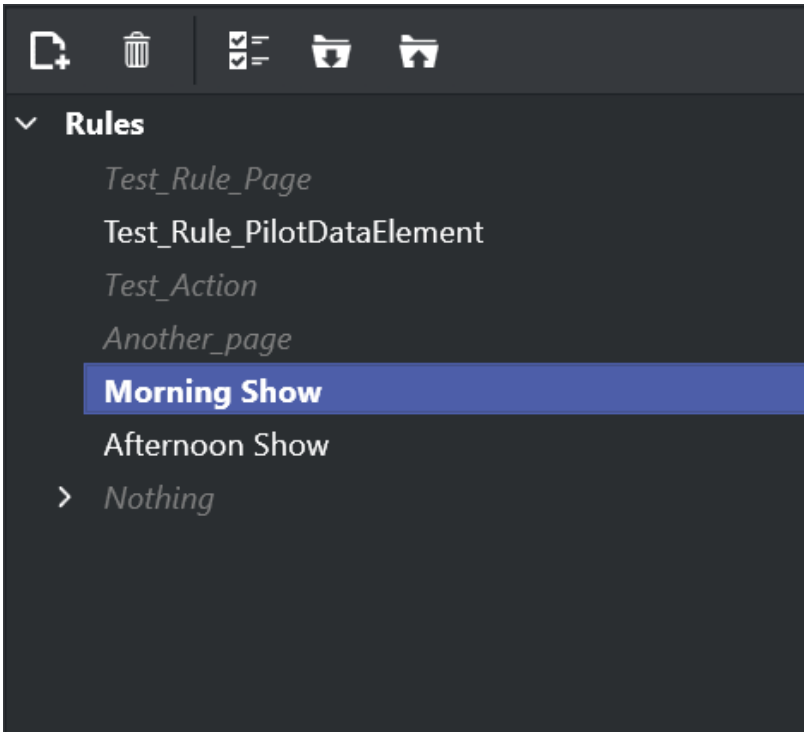
2. Select **Rules** from the menu actions to bring up the user interface for working with rules:



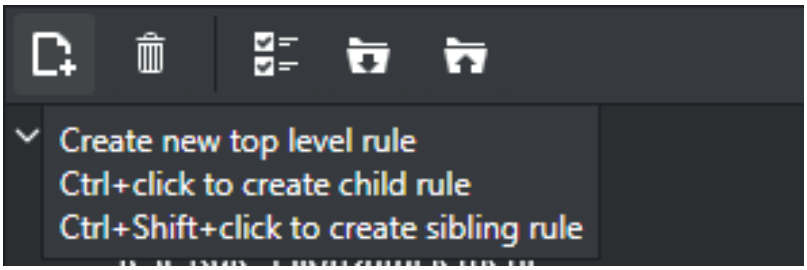
3. In the screenshot above, there are now two window panes on the right-hand side:
    a. In the left pane, you can create, delete, activate/deactivate, select and import/export rules.
    b. In the right pane, you can work with a selected rule: name the rule and give it the appropriate conditions and actions.

Copyright © 2024 Vizrt

Additional Rules Operations

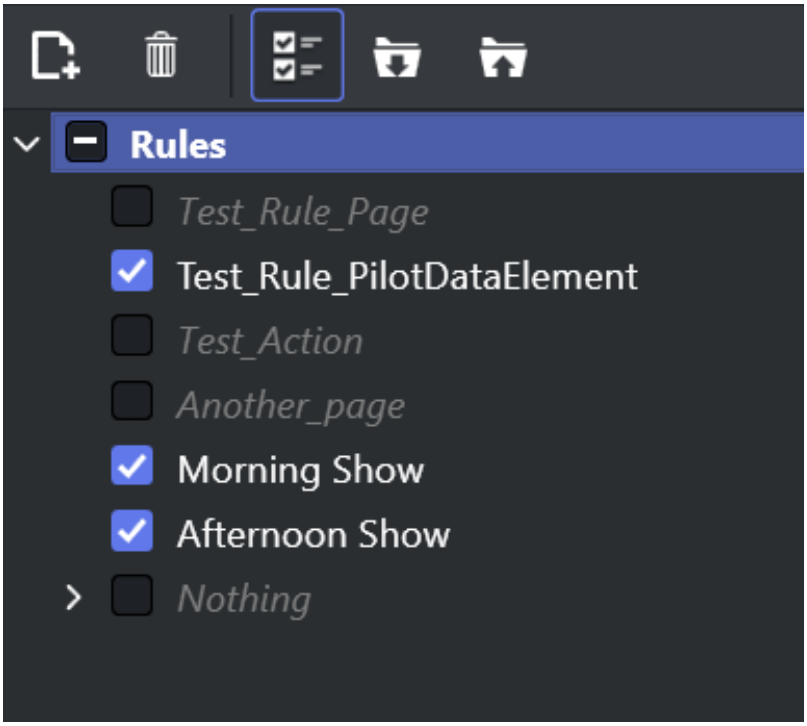


- Select by clicking the rule you want to work with, or select the appropriate icon to create, delete, activate/deactivate or import/export rules.
- Rules can be grouped into a hierarchy with children and siblings. Hover the cursor over the create rule icon to see the options: **CTRL + click** to create a child rule and **CTRL + SHIFT + click** to create a sibling rule:



- Existing rules can be activated/deactivated as required. In the example below for example the rule *Another_page* is deactivated while *Morning_Show* are activated.

Copyright © 2024 Vizrt



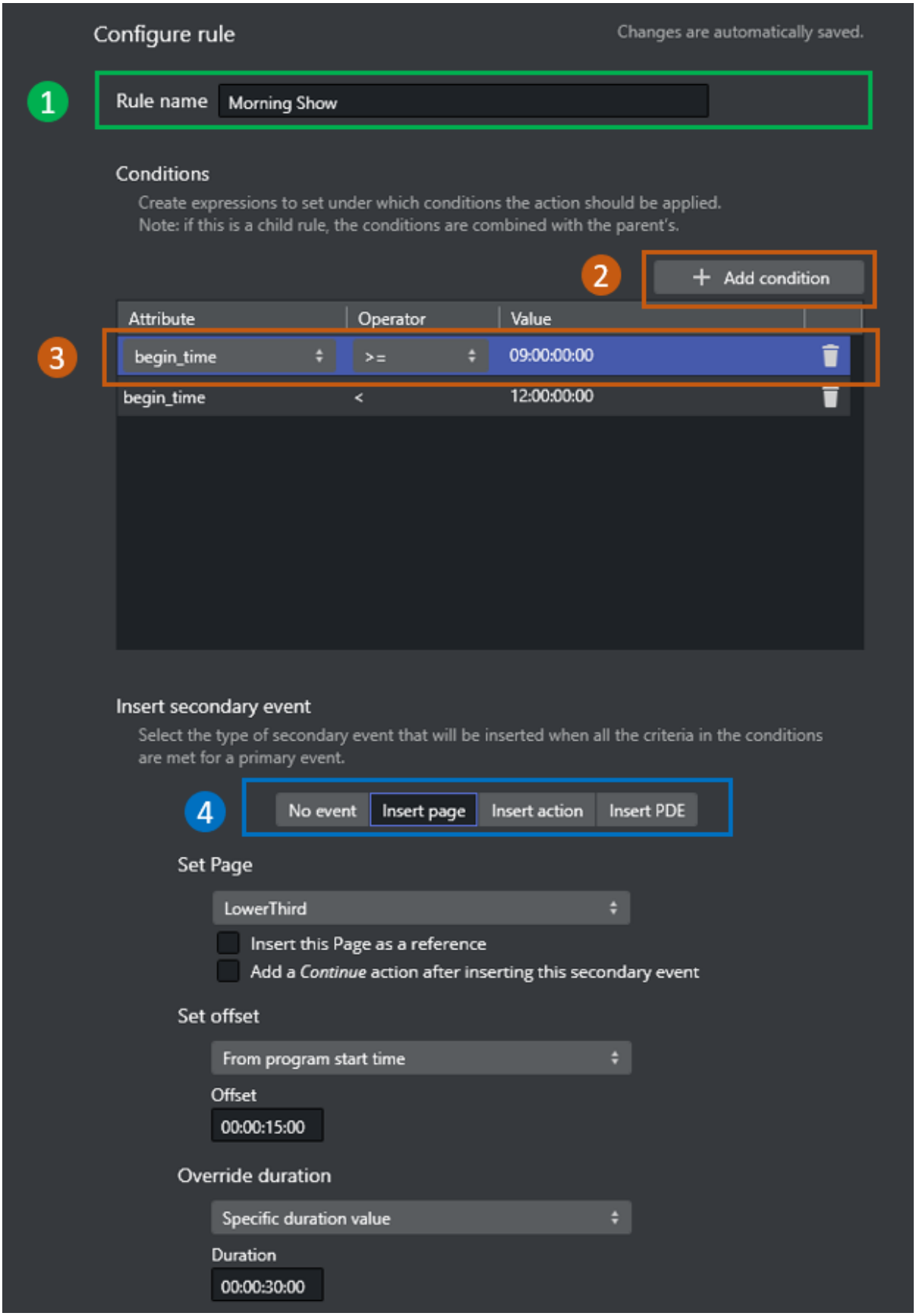
The Import/Export of rules is from/to XML-files.

- Since existing rules will be overwritten when importing rules, make sure you want to overwrite, or create a copy safely stored to disk as backup, before importing.

16.6.2 Defining Rules for Conditions and Actions

Once you have selected a rule, you can use the rules panel to create conditions and actions for the rule.

Copyright © 2024 Vizrt



Copyright © 2024 Vizrt

Referring to the screen-shot above:

1. Name the rule (if newly created).
2. Create **conditions**. You can have as many conditions as needed. Conditions are expressions that specify if the rule should be applied. Using the various condition operators you can answer questions such as: *if the next film is for the ages over 12, display a legal-age page as graphics 10 seconds into the film.*
3. Define the conditions of the rule. A rule is only applied if all the criteria in the conditions are met. Click the column **Attribute**, **Operator** or **Value** to select possible values from a dropdown-list or to change the actual **Value**. Condition types are explained in more detail in the next section, **Rules criteria and conditions**.
4. If all conditions for the rule are met, the rule will be acted upon and you can specify what should happen: insert **No event**, a specific **Page,** a predefined **Action** or a **Pilot Data Element**. Clicking on the **Insert page, Insert action** or **Insert PDE** button will enable a form where you can select and configure the appropriate element.

Options in the **Rules** UI panel:

- Select the **Insert this Page as a reference** checkbox to allow page to be added as a referenced page.
- Select the **Add a Continue action after inserting this secondary event** checkbox to add Continue action after the event added by rules. This option lets you set the offset time for the continue action.
- Select the type of offset in the **Set offset** combobox to set the value accordingly.
- Select how to **Override duration** in the combobox and set the duration for the page. If **Do not override** is selected, the call-back will be the origin page.

16.6.3 Rules Criteria and Conditions

Any number of conditions can be applied to a rule so that the rule is applied only when specific criteria are met. If there are no conditions, rules are applied on every primary event when a playlist is imported.

Conditions are made up of an attribute representing a column in the playlist table, an operator (greater than, less than, equal to, etc.) and a value (time, number, text). Only when the condition as defined is met will the rule be applied on a particular event. Likewise, when there are multiple conditions, only when all of them are met will the rule be applied to an event.

For any event dealing with time (for example 19:00:00), the Rules can look up elements that start after a given time (>) or at a given time and later (>=). In addition, the Rules can perform partial matches on strings which lets you find primary elements that are“right before”, “first one before this” or “current one at this time”.

Rules are executed in the order they are listed. However, the order of the rules has no impact on the final result as there is no adjustment for any pages already added. This means that when you are setting up rules you should make sure that the end-result you are seeking is not dependent on the order in which the rules are run.

While defining rules, it is possible that many of them share some common conditions. By having a parent rule defined with such conditions, we can define the remaining in child rules. The child rules will therefore only run if *all* the conditions in *parent rule* are met.

Copyright © 2024 Vizrt

Partial Match

In order to find partial matches such as “NEWS*” to match against “NEWSROOM” you must use the partial match operator *(~=)*. This allows for string based comparisons of partial hits (“contains”) and time-based comparisons (“currently playing”).

The partial match operator can also be negated with ***!~=*** which can be read as “does NOT partial match”.

The logic for time-based look-up using the partial match operator is a proper check of the “time” (for example at 19:00:00) and start-time of a primary element (for example at 18:45:00), its duration (for example 00:30:00) and detecting that the primary element is currently on air.

Condition Types

Operator	Description
=	equal to
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
<>	different from
~=	partial match
!~=	does NOT partial match
RegExp	a regular expression

Regular Expressions

You can also use regexp (regular expression) in conditions. This gives near unlimited flexibility in defining conditions and is a very powerful tool. It is advisable to test any regexp before using it in a rule. It is beyond the scope of this manual to explain regexp in detail. Please see one of the many books or internet resources that explain RegExp.

Some simple examples for the RegExp operator are shown below.

Copyright © 2024 Vizrt

Attribute	Operator	Value	Meaning
Description	RegExp	^[a|b|c]	Description begins with a or b or c
Description	RegExp	[a|b|c]$	Description ends with a or b or c
Description	RegExp	^a.*b$	Description starts with a and ends with b
Description	RegExp	^[a|b].*[d|e]$	Description starts with a or b and ends with d or e
Description	RegExp	^[a|b]|[d|e]$	Description starts with a or b or ends with d or e
Description	RegExp	^(?=.*a)(?=.*b).*	Description contains a and b
Description	RegExp	^(?=.*a)|(?=.*b).*	Description contains a or b
Description	RegExp	^((?!a).)*$	Description does not contain a
Description	RegExp	^(?=.*a)(?!.*b).*	Description contains a and does not contain b
Description	RegExp	\+	Description contains +

16.6.4 Back up and Restoring Rules

All rules and their associated conditions can be saved as XML files for later use.

To back up Rules

1. Select your channel.
2. Click the Rules button. The list of rules is displayed in the pane.
3. Click the **Export Rules** icon.
4. Enter a file name and click **Save**.

To restore Rules

1. Select your channel.
2. Click the Rules button. The list of rules is displayed in the right pane.
3. Click the **Import Rules** icon. You are asked to confirm the overwriting of any existing rules.
4. Click **Yes** to proceed.
5. Select the XML rules file and click **Open**.

Copyright © 2024 Vizrt

16.7 Resolving Conflicts

Viz Channel Branding ensures that there will not be conflicts or clashes if all necessary data is available to an event. The playlist should include the following data for every instance:

- Event ID
- Primary event begin time
- Primary event duration
- Secondary event offset
- Secondary event duration

Copyright © 2024 Vizrt

16.8 Generating The Channels File

Once you have finished all scheduling tasks, you should manually regenerate the channels file before proceeding to monitor your playlists in MCR mode.

16.8.1 To regenerate the channels.ini file

1. From the **main menu** select the **Tools** option.
2. From the Tools menu click **Generate Channels File**. You are prompted to allow Viz Channel Branding Client to overwrite the existing file.
3. Click **Yes**.

The channels file is normally located in *C:\ProgramData\Vizrt\Viz Channel Branding* and is named *channels.ini*. It is a normal text file and can be manually edited.

Copyright © 2024 Vizrt

17 Master Control Workflow

This chapter defines and describes tasks that are typically performed by the automation system after control of the playlist has been handed over by the scheduling system in the traffic department.

This section contains information on the following topics:

- Activate the Playlist
- Initialize the Playlist
- Run the Playlist
- Triggers
  - Primary Events
  - Secondary Events
- Monitoring
- Live Schedule Updates
- Fail-over Procedure

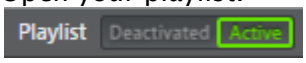
17.1 Activate The Playlist

When a playlist is activated for the currently selected channel, Media Sequencer can start to transfer full screen video clips from your media asset management system (for example Viz One) to Viz Engine for playout.

When activated, Media Sequencer is made aware of the playlist's schedule. Without the schedule, Media Sequencer will not be able to initialize the elements and run the playlist properly. Hence, once activated you can Initialize the Playlist.

A playlist can be automatically activated based on Rules or Master Control Workflow (day switch options are common for broadcast day playlists) or by manually clicking the **Activate** button when the playlist is open.

17.1.1 To manually activate a playlist

1. Select your channel.
2. Click the Playlists button.
   All existing playlists for the selected channel are displayed.
3. Open your playlist.
   

   ,
4. From the Playlist Toolbar, click the **Run Playlist** button.
5. Initialize the Playlist.
6. Run the Playlist.

17.2 Initialize The Playlist

All playlist pages (graphics including embedded video clips and imagery) that are initialized for the currently selected channel are loaded on the Viz Engine program for playout.

Copyright © 2024 Vizrt

When activated and initialized, Media Sequencer is aware of the playlist's schedule and is able to Master Control Workflow.

A playlist may be initialized based on Rules (this is common for continuous playlists), Day Switch Options (this is common for broadcast day playlists) or by manually clicking the **Initialize** button when the playlist is open and active.

17.2.1 To manually initialize a playlist

1. Select your channel.
2. Click the Playlists button. All existing playlists for the selected channel are displayed.
3. Open your playlist.
4. Master Control Workflow.

5. From the Playlist Toolbar, click **Initialize Playlist**.
6. Run the Playlist.

17.3 Run The Playlist

When in run mode, a playlist is activated and initialized for the currently selected channel. All full screen video clips and all pages (graphics including embedded video clips and imagery) are then available and loaded on the Viz Engine program for playout.

When in run mode, Media Sequencer has a playlist that is activated and initialized and the system can take elements on air.

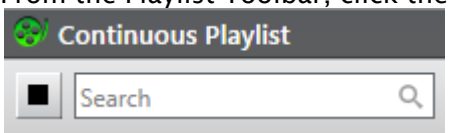
A playlist is almost always automatically run by Media Sequencer. If there are multiple broadcast day playlists, these are run according to **Day Switch Options**. A playlist can be manually run by an operator, but this is not recommended nor common practice.

> ⚠ **Note:** Only one playlist per channel can be in **Run** mode at a time.

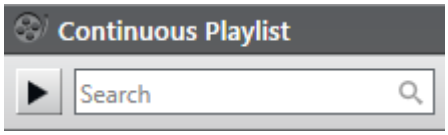
17.3.1 To manually run a playlist

1. Select your channel.
2. Click the Playlists button.
   All existing playlists for the selected channel are displayed.
3. Open your playlist.
4. Activate the Playlist.
5. Master Control Workflow.

6. From the Playlist Toolbar, click the **Run Playlist** button.



Copyright © 2024 Vizrt



When in Run mode, the film roll icon seen at the top of the playlist will turn green. When stopped, the icon will be grayed out.

If **manual intervention** is required, control the playout using the control buttons available on the **Playlist Toolbar**.

- **Take**: Takes the selected playlist element (or group of elements) on air.
- **Continue**: Continues the animation of the playlist element.
- **Take Out**: Takes the current playlist element off air.

17.4 Triggers

Triggers can be set on *Primary Events* and *Secondary Events* (pages and video clips - see below). Note that triggers are configured in Master Control Workflow. Triggers are only accepted when a Playlist is in Run mode.

17.4.1 Primary Events

Primary events are triggered by automation systems that send triggers to start playing out groups (primary event/element). All secondary events that exist under each group are automatically played out by the solution. Playout of the secondary events is relative to the time of the trigger of the primary event and based on the defined offset of the secondary events.

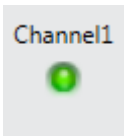
17.4.2 Secondary Events

Secondary events can be triggered by automation systems. This includes all types of secondary events such as pages with video clips and video clips with graphics overlay.

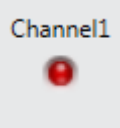
In the case of secondary events, Viz Channel Branding will simply act as a slave system listening for triggers to take events on air.

17.5 Monitoring

In most cases the solution requires little or no monitoring as the automation system usually controls the playout. In the Control Room, playout is typically monitored live on TV displays set up for that purpose. Playout can also be monitored on the dedicated Viz Engine machine (if installed).



Copyright © 2024 Vizrt



You can monitor the status of the Media Sequencer in the left pane. If the channel configuration is OK, the lights are green. If there is any problem, the lights are red. Hover your cursor over the red light to receive an indication of the problem. This will tell you if Media Sequencer and your configured playout devices are working properly.

17.6 Live Schedule Updates

If playlists are liable to be changed a great deal, it is recommended that updates be done in the automation system and updates to the solution be done automatically and frequently using direct integration against Integrations Hub component.

17.7 Fail-Over Procedure

In the case of fail-over, you need to be prepared beforehand to accommodate such a situation.

The prerequisite to backing up is to have an extra machine with the Viz Engine and Media Sequencer and configure either a *hot* or *cold* backup solution. A hot solution includes having either an engine or both MSE and engine actively play out the current content, while a cold solution could either be a dormant machine with multiple channels in waiting or a semi-hot solution where the automation and all other components are in a prepared backup chain for fail-over when needed. Each installation may have individual or distinct considerations and should be designed during the installation phase.

See Also

- Traffic Department Workflow

Copyright © 2024 Vizrt

18 Integration With Automation Systems

The content of a playlist is usually provided from an external source like an automation, traffic or scheduling system.

This chapter describes some common automation system integrations to Viz Channel Branding.

- Automation Systems
- Automation System Integration types

18.1 Automation Systems

An **Automation System** is usually located in the master control room, controlling all devices in the room in real-time. For example video servers, keyers, and Character Generators.

An integration to an automation system typically performs:

- **Triggering**: The automation system triggers Viz Channel Branding (via Media Sequencer) for starting either primary or secondary events. Triggering protocols TCP/IP, Intelligent Interface, VDCP, GPI, and REST are supported by Vizrt. It is important to understand that although Channel Branding manages the playlist and events, the actual triggering of actions will be done by Media Sequencer.
- **Schedules and last minute updates**: If an automation system provides an API or an option to automatically generate an update file, Viz Channel Branding can update its playlist using them.
  This is usually done using direct integration against Integrations Hub or via a schedule file that is ingested with the PlayList Importer tool.

18.2 Automation System Integration Types

Although the precise manner in which external automation systems deliver the schedule and triggering varies greatly, the flexible and open architecture of Channel Branding ensures that most commonly used automation systems are supported directly.

There are two main types of integration to automation systems:

- File based systems
- Direct integrations systems (API-based)

File based automation systems will typically deliver the schedule in a pre-defined Text, XML, Excel, CSV, database or proprietary binary format. The typical workflow is then to have a file or directory produced by the automation system watched (or have time-based tasks running) and use Playlist Importer to ingest the data that is pushed via Integrations Hub and updates the playlist as seen in the Viz Channel Branding Client.

Direct Integration systems are automation and schedule systems that have an API (Application Programming Interface) that Viz Channel Branding can interface with and read information from, which is sometimes bi-directional. The main advantage with direct integration systems is that the file-based workflow can be avoided. This normally leads to significant performance gains. Direct

Copyright © 2024 Vizrt

integration is also a more secure, clean and resilient setup than most file-based workflow systems and is the recommended integration type.

For an overview of the integration types and how to configure some of the more commonly used automation systems, see the following sections:

- Overview of Direct Integration Architecture
- Integration Module for Marina from Pebble Beach
- Integration Module for Neptune from Pebble Beach
- Integration module for Harris ADC
- ADC-100 by Harris Broadcast
- D-Series by Harris Broadcast

Note: Vizrt may support additional automation systems not listed here. Please contact your local Vizrt Support for more information.

Copyright © 2024 Vizrt

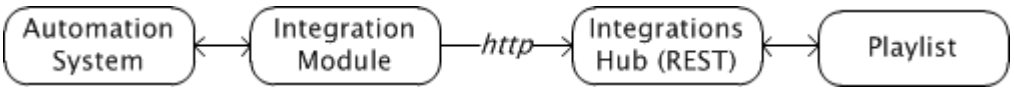
18.3 Overview Of Direct Integration Architecture

The architecture flow of an external system communicating with the Integrations Hub is called Direct Integration (DI). The Integrations Hub component is a REST web service that lets an external system fully query, add, update and remove contents of a playlist.

The preferred setup is for the external system to have a native driver for communicating with the Integrations Hub, using the REST interface to maintain sync between the systems. Since the Integrations Hub has an open and easily accessible REST interface, integration modules can be written in a wide range of programming and scripting languages.

18.3.1 Integration Module

An alternative approach is to use a custom **Integration Module** if the external system lacks the ability to implement a native driver or already has its own web service. The integration module will handle all details relating to interfacing with a specific automation system and forward the required data (schedules etc.) to the Integrations Hub. Although the specific integration module can be provided by Vizrt, it is also possible for vendors and third parties to develop integration modules.



See Also

- Introduction to REST (external Wikipedia page)
- Integrations Hub

Copyright © 2024 Vizrt

18.4 Integration Module For Marina From Pebble Beach

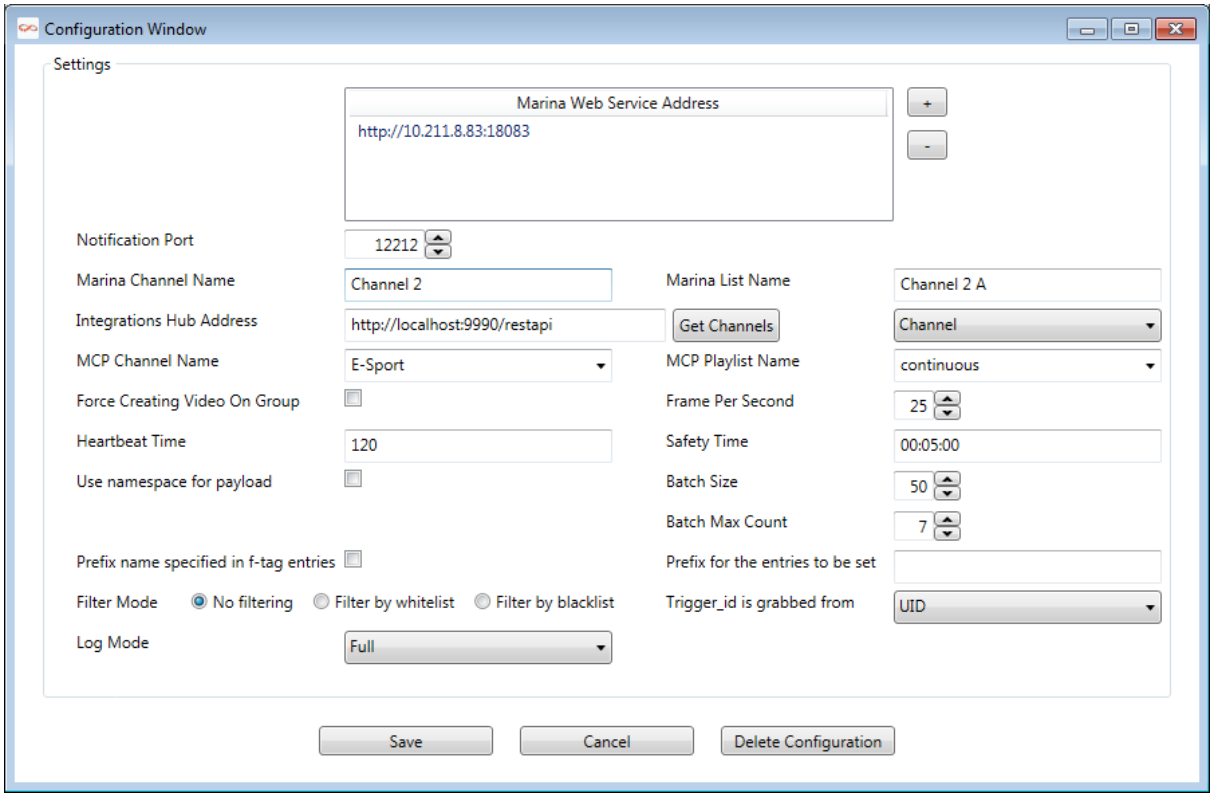
The integration for Pebble Beach's (PB) Marina system uses the PB Marina Integrations Module, as explained in the **Integrations Module** section of Overview of Direct Integration Architecture.

Before using the PB integration module, make sure that the Integrations Hub service is installed.

18.4.1 Configuring the Integration Module for PB Marina

Before you start the PB Marina integration module you must collect some details from your local Viz Channel Branding setup and PB Marina system. In particular, you need the network address (or IP-address) of Marina servers and the hostname where the Integrations Hub is installed. The required information is documented below.

- Start up the application
- Click the **New** button to create a new configuration. This will display the configuration window:



Copyright © 2024 Vizrt

Name	Definition
Marina Web Service Address	Use the + and - buttons to add (or delete) the address of PB Marina servers, including port number. Example: http://localhost:18083 . You can include both main and fallback Marina Web Service servers. Having several housekeeper addresses ready in advance makes it easy to quickly switch between them when needed. In addition, if a Marina server is not reachable due to Marina server problems or network issues, for example, the integration unit will make a note in the log and try the next one in the list in a rolling fashion.
Notification Port	The port that Marina Integrations opens to receive notification from Marina IS. The default value is 12212.
Integrations Hub address	The IP or hostname of the Integrations Hub module. Example: http://MyServer:9990/restapi.
Marina Channel Name	Define name for Channel as defined on the Marina system (applicable for newer Marina 2018+ versions).
Marina List Name	The Marina list name for the integration.
Channel / TxList DropDown	Choose what component type to use to verify channel declaration in the Marina system (Marina 2018+ can now optionally choose 'Channel').
MCP Channel Name	The channel name you defined on Viz Channel Branding side that should be used with this direct integration module.
MCP Playlist Name	The name of the playlist you want to sync data with. The format of playlist name is [dd_MM_yyyy] and you can also input "*continuous*" or "*running_playlist*". Use the playlist name `running_playlist` to sync data with the current running playlist.
Force Creating Video On Group	Enables the application to use Media Name to search for videos in Viz One and add to an event. For this feature to work, integration with Viz One must be enabled.
Frame Per Second	Used to calculate the begin-time of events that are received from Marina.
Heartbeat Time	(in seconds). Time watch that will wait for a notification from PB Marina. Otherwise, Marina Integration will collect the current playlist.

Copyright © 2024 Vizrt

Name	Definition
Use namespace for payload	Switch, either **on** or **off**. This option will depend on the current version / configuration of your PB Marina installation. If unchecked: If the PB Marina has AuxData node, the Marina Integration module will generate the element with attributes from the payload node inside AuxData. If checked: The Marina Integration module will look for a namespace in the payload received from Marina.
Safety Time	Specify as HH:mm:ss - this is the input period +/- from the current time to collect input from Marina. Example: If current time is 12:00:00 and Safety-Time is 00:05:00 then the integration module will only collect events that have been modified from Marina within 11:55:00 - 12:05:00.
Batch Size	Accepted values: between 1 and 50. Batch size will be used when requesting event data from Marina. If set to 50, the Integrations Module will ask for 50 events per set. If Marina has more than 50 events, the Integrations Module will ask multiple times until all events are received.
Batch Max Count	Optionally limit the number of elements to continuously retrieve from the Marina system by limiting the number of Batch requests. (0 = no limit)
Prefix name specified in f-tag entries	Switch. Handle non-conforming XML-formatted f-tag entries by prefixing them if switched on. See also next bullet.
Prefix for the names to be set	For the entries to be **used**: If prefixing f-tag entries, use the prefix specified.
Filter Mode	Select from **No Filtering** or filtering by **Whitelist** or by *Blacklist * as explained in the bullets below. Comma-separated values for whitelist or blacklist must be provided in the edit-box. · Filter by **whitelist**: Each element's type-value is verified against each value set in the editbox, if the type-value *match* any of the values then it is allowed through. · Filter by **blacklist**: Each element's type-value is verified against each value set in the editbox, if the type-value *does _**not*** match_ all of the values then it is allowed through.
Trigger_id is grabbed from	Select to grab value for trigger_id control data from ReconcileKey or the UID value from Marina payload.
Log Mode	Enable or Disable log messages. Verbosity can be specified as **Off** (no logs), **Basic** or **Full**. Log files will normally be located in *%ProgramData%\Vizrt\Viz Channel Branding\Marina Integration*.

Copyright © 2024 Vizrt

- When all required information is entered, press **Save** to save the settings and return to the main application window.
- Press **Start** to start the Marina Integration service.

Copyright © 2024 Vizrt

18.5 Integration Module For Neptune From Pebble Beach

Pebble Beach (PB) Neptune systems can use the PB Neptune Integrations Module, as explained in the **Integrations Module** section of Overview of Direct Integration Architecture.

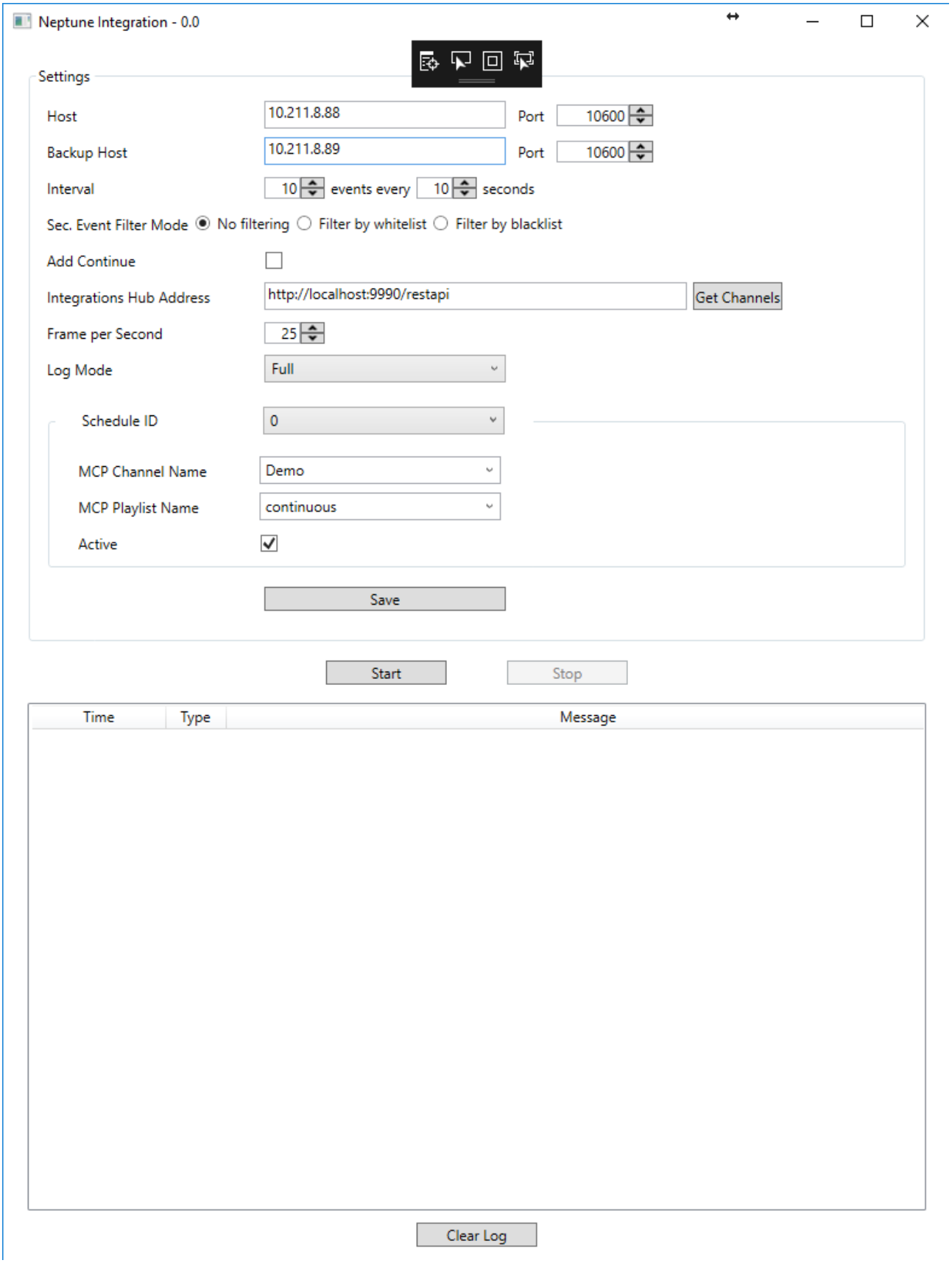
Before using the PB Neptune Integration Module, make sure that the Integrations Hub service is installed. In most cases, it is also required to have the Promo Server installed.

- Configuring the Integration Module for PB Neptune

18.5.1 Configuring the Integration Module for PB Neptune

Before you start the PB Neptune integration module, you must collect some details from your local Viz Channel Branding solution and PB Neptune system. In particular, you will need the network address (or IP-address) of Neptune servers and the hostname where the Integrations Hub is installed. The required information is shown in the screen-shot below.

Copyright © 2024 Vizrt



Copyright © 2024 Vizrt

Form Input	Description
Host	Neptune server host name/IP and port number.
Backup Host	Backup Neptune server host name/IP and port number.
Schedule ID	Index of Schedule from Neptune server, starting from 0, up to 20. Set the channel and playlist per schedule ID. You activate the schedule by selecting an active schedule (only schedule IDs that are active are displayed).
Interval	Number of events sent to MCP, and loop time in seconds. Default is to continuously ask for 20 events every 30 seconds.
Sec. Event Filter Module	Select either *No Filtering* or *Filter by whitelist* or by *Filter by blacklist*. Comma-separated values for whitelist or blacklist must be provided in the edit-box. • *Filter by whitelist:* Each element's type-value is verified against each value set in the edit box. If the type-value matches any of the values, then it is allowed through. • *Filter by blacklist:* Each element's type-value is verified against each value set in the edit box. If the type-value does not match all of the values, then it is allowed through.
Add Continue	Flag to insert 'Continue' action to each primary event.
Integration Hub Address	The IP or hostname of the Integrations Hub module. Example: http://MyServer/integrationshub/restapi
MCP Channel Name	The channel name you have defined that will be used with this direct integration module.
MCP Playlist Name	Name of the playlist you want to synch data with. The format of playlist name is `[dd_MM_yyyy]` and you can also input "continuous" or "running_playlist". Use the playlist name `running_playlist` to sync data with the *current* running playlist.
Frame per Second	Used to calculate begin-time of events received from Neptune.

Copyright © 2024 Vizrt

Form Input	Description
Log Mode	Enable or Disable log messages. Verbosity can be specified as **Off** (no logs), **Basic** or **Full**. Log files normally located in `%ProgramData%\Vizrt\Viz Channel Branding\Neptune Integration.`

- When all the required information is entered, press **Save** to save the settings.
- Press **Start** to start the Neptune Integration service.

Copyright © 2024 Vizrt

18.6 Integration Module For Harris ADC

The Harris automation system can use the Harris Integrations Module, as explained in the **Integrations Module** section of Overview of Direct Integration Architecture. The integration is based on the Harris ADC-12 Web API.

In most cases, the Promo Server must be installed before using the integration module.

> **Note:** Harris is now renamed Imagine Communications (http://www.imaginecommunications.com). This documentation uses the former Harris brand name.

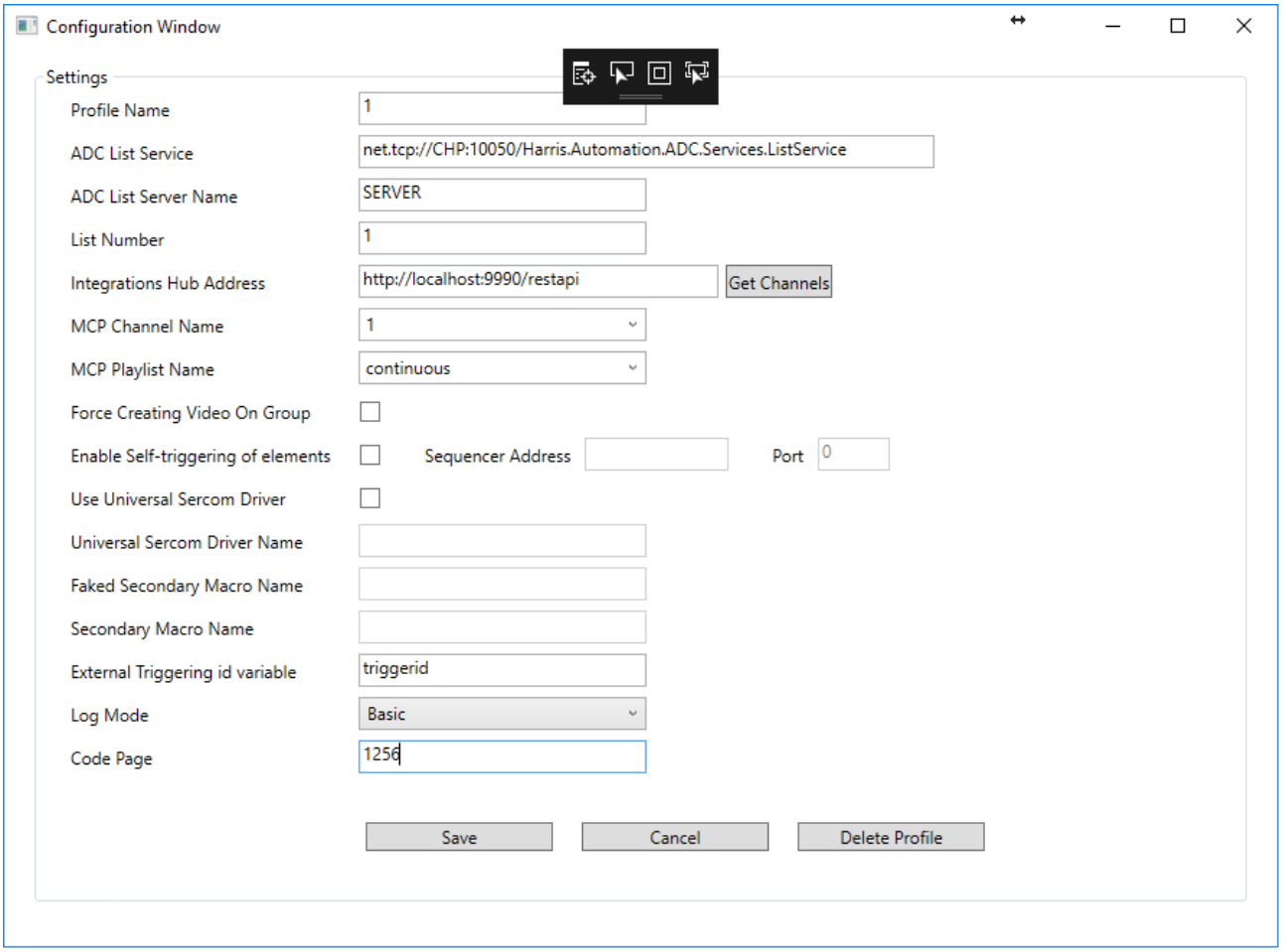
- Configuration and Startup
- Pushing Optional Metadata from Harris to the resulting playlist

18.6.1 Configuration and Startup

- Start the integration module application.

The Harris integration module will display the **Configuration window**:

Copyright © 2024 Vizrt



Detail	Description
Profile Name	You can have multiple settings profiles. If you run multiple instances of the Harris Integrations service, it is recommended to make configuration changes from one instance only to avoid overwriting configuration changes.
ADC List Service	Address of the ADC List. Change the IP as required. Ask your Harris system administrators for the correct address.
ADC List Server Name	The ADC List Server name that should be used in the integration.
List Number	Sequence of list that should be used in the integration. Example: If the actual Harris system has three list-names called "Playlist 1", "Playlist 2", "Playlist 3" and you want to use "*Playlist 2*" for the integration, specify **2** as the list-number.

Copyright © 2024 Vizrt

Detail	Description
Integrations Hub Address	Specify the URL of the REST service. Note: You can browse to this URL with a browser to check that the Integrations Hub is installed and working. Expected result answer is: *Integrations Hub.*
MCP Channel Name	The channel name you are setting up this integration module for.
MCP Playlist Name	The name of the playlist you want to sync data with. The format of playlist name is [dd_MM_yyyy] and you can also input "*continuous*" or "*running_playlist*". Use the playlist name `running_playlist` to sync data with the current running playlist.
Force Creating Video On Group	Enable the application to use Media Name to search for videos in Viz One and add to an event. For this feature to work, integration with Viz One must be enabled.
Sequencer Address	The Address of Media Sequencer that we shall send the trigger against. The TCP port which has to match with the setting in Channel Settings.
Port	The TCP port which has to match with the setting in Channel Settings.
Enable Self-triggering of elements	If checked, any primary event element notified to have their state changed to 'PLAY', it will be triggered against Media Sequencer via TCP connection. **Note:** The option 'Re-use ADCEventId as Trigger Id' will be checked automatically when this option is checked.
Use Universal Sercom Driver Name	If checked, Harris Integration will activate the Universal Sercom Driver Name and this Macro Name will also be activated in the process.
Universal Sercom Driver Name	This ID must match the ID column in Harris Automation. Ask your Harris administrator for the driver name ID.
Faked Secondary Macro Name	Define the macro used here in order to use a secondary event to re-label the Primary event's trigger ID.

Copyright © 2024 Vizrt

Detail	Description
Secondary Macro Name	Prefix of page-name information from Harris Automation, related to Macro definition in Harris system. The value can be semi-colon (';') separated and contain multiple values.
Log Mode	Enable or Disable log messages. Verbosity can be specified as **Off** (no logs), **Basic** or **Full**. The Log and Error Log files will normally be located in *%ProgramData%\Vizrt\Viz Channel Branding\Harris Integration*. You can also open the log directories in Windows Explorer by pressing **Show log folder** or **Show error folder**.
Code Page	The Description and CompileId fields returned from the Harris system are converted according to this defined code page. For example, code page with identifier *1256* will convert bytes to ANSI Arabic.

- When all the appropriate setting are configured press **Save**.
- Press **START**.

18.6.2 Pushing Optional Metadata from Harris to the resulting playlist

ADC systems using the LST12 format can optionally use the generic Harris "*Data*" field (max 4096 characters) to push over meta-data information. If the Harris Data-field contains entries of type "*value=somedata*<CRLF>" (or several formatted as "value1=somedata<CRLF>value2=somemoredata<CRLF>" etc.), the integrations module will parse these entries and, if a valid combination *fieldname=value* is found, push this payload to the Integrations Hub.

Copyright © 2024 Vizrt

18.7 ADC-100 By Harris Broadcast

This section describes how to configure the Automated Content Management and Distribution system (ADC-100) by Harris Broadcast for Viz Channel Branding. It describes the setup on the ADC side and contains information on the following topics:

18.7.1 General Information and Setup

An ADC-100 system can trigger Viz Channel Branding (by sending commands to Media Sequencer) through VDCP or Universal Sercom Device (USD) using either TCP/IP or VDCP.

> ⚠ **Note:** Both version 11 and 12 of the LST format are supported.

The advantages of VDCP are that it is:

- Standard
- Proved to work for many customers

The advantages of USD are that it:

- Can be used over TCP/IP thus saving serial ports and additional cabling.

This section contains information on the following topics:

- Triggering
- Limitations
- Connections
- ADC-100 to Media Sequencer: Connections
- Device Configuration
- Operation
- USD Device Configuration
- Notes
- Operation

Triggering

Triggering should be Primary/Secondary Events selected.

Primary Event Defaults should be Triggered (Ext) selected. The ADC-100 system reads playlist data via files in LST format (binary files).

Reconcile field value in the LST is read into the Event ID field in Viz Channel Branding and must be set in order to receive playlist updates. An empty or corrupted Reconcile value may omit that event from the update procedure.

Limitations

The LST format only allows for 32 character titles and misses other necessary fields such as local language titles.

Copyright © 2024 Vizrt

Allow replay – Media Sequencer virtually removes played events from its exposable list, so Vizrt events will be marked as MISSED on ADC-100 side although played successfully.

18.7.2 Triggering with VDCP

The standard configuration of triggering by a serial connection is using Video Disk Control Protocol (VDCP).

Connections

Use an RS422 board on the machine with Media Sequencer, for example: Blue Heat/PCI of Connect Tech Inc. The following custom cables are required to connect Media Sequencer to the ADC-100 system.

ADC-100 to Media Sequencer: Connections

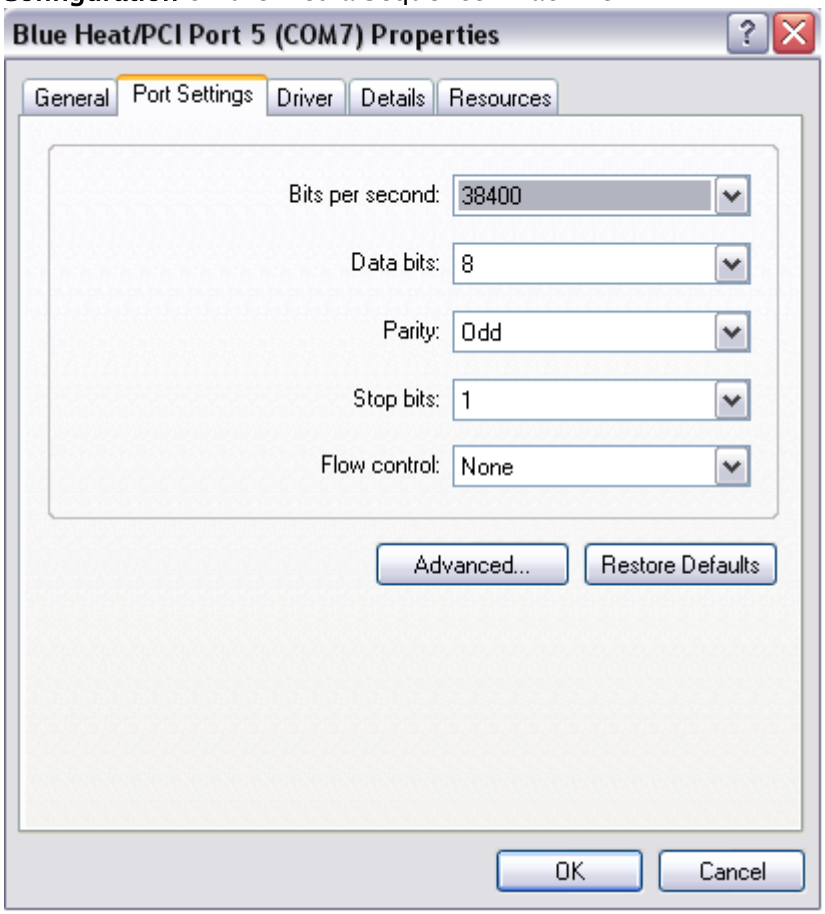
ADC-100	Media Sequencer Machine
2 (Rx -)	2 (Tx -)
3 (Tx +)	3 (Rx +)
7 (Rx +)	7 (Tx +)
8 (Tx -)	8 (Rx -)

Copyright © 2024 Vizrt

Device Configuration

To configure the software

- Make sure the COM port used has the following suggested parameters under **Windows Configuration** on the Media Sequencer machine:



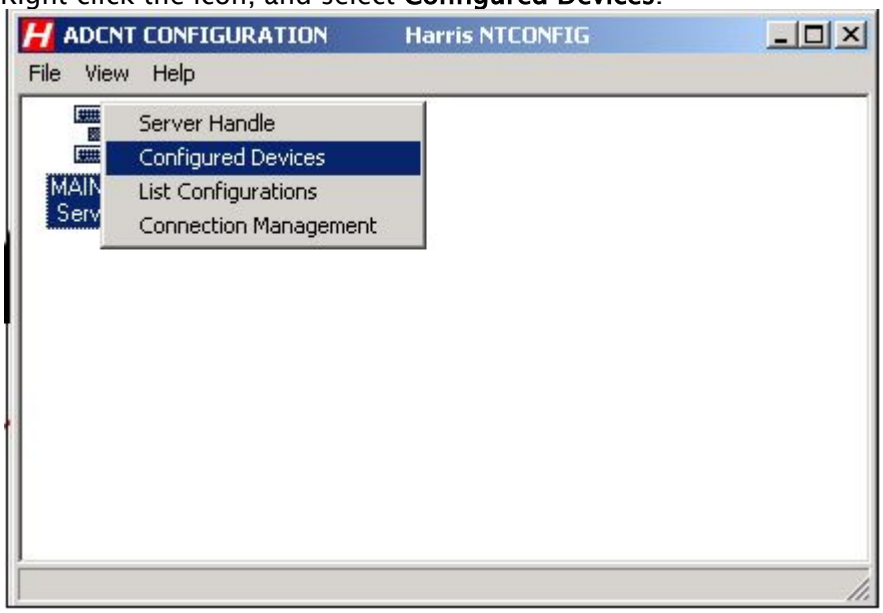
  - Bits per second: **38400**
  - Data bits: **8**
  - Parity: **Odd**
  - Stop bits: **1**
  - Flow control: **None**
- Enable the VDCP device channel settings and ensure configured properly with the correct port.

To configure the Viz driver

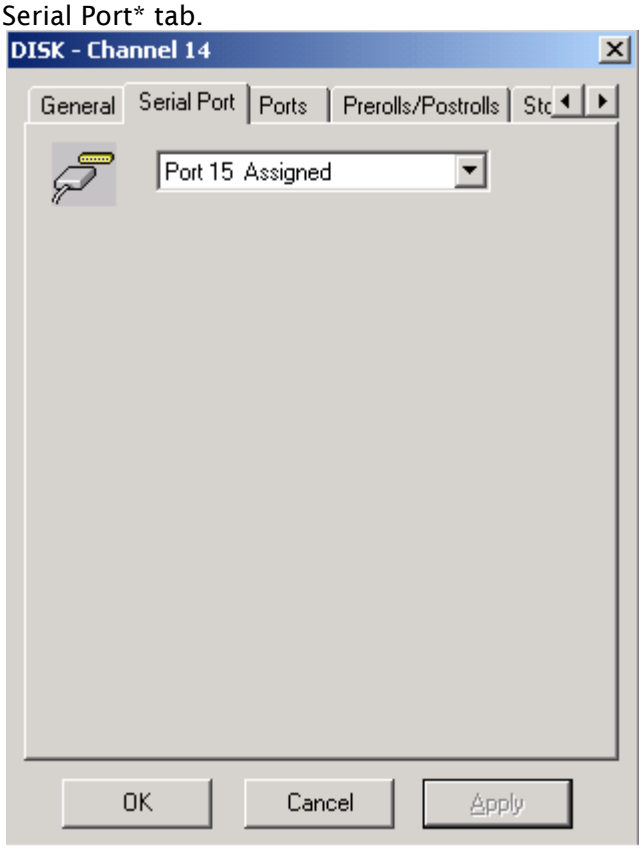
1. Launch the ADC-100 Configuration Utility and locate the icon that represents your Device Server.

Copyright © 2024 Vizrt

2. Right-click the icon, and select **Configured Devices**.



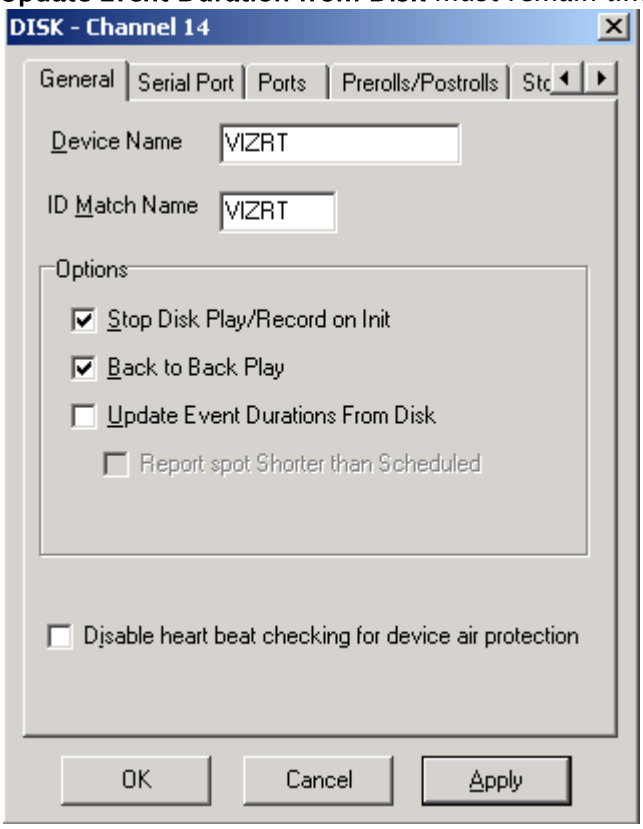
3. In the **Available** column on the left, locate the **Standard Video Disk** driver under the **Video Disk** category. Click and drag the driver to one of the channels on the **Configured** column that reads **NO DEVICE**.
4. Double-click on the Disk device that was just created. From the resulting dialog select the* Serial Port* tab.



5. Select the physical port the Media Sequencer is connected to. The Harris Device Server is now configured properly to communicate with Media Sequencer.

Copyright © 2024 Vizrt

6. Configure the following parameters on the General tab:
   - **Device Name** will appear in the **Device Status** and configuration windows on the client applications. The default name may be used, but it is recommended to use unique names to ease later identification.
   - **ID Match Name** will be used in the Transmission List's *ID Field *to let the system know that the event is intended for the Vizrt device.
   - **Stop Disk Play** can be selected as required.
   - **Back to Back Play** can be selected as required.
   - **Update Event Duration from Disk** must remain *unchecked*.



7. The Vizrt Standard Video Disk must also be assigned to a Transmission List to be scheduled to air.

> ⚠ **Note:** The Standard Video Disk driver cannot be used if you are using Extended IDs. In addition, Media Sequencer does not contain the notion of duration, so the parameter **Update Duration from Disk** must be *unchecked* to avoid excessive error messages when ADC driver asks for the IDs list.

Operation

Secondary A/V or Secondary Backtimed events can be used to execute the pre-loading and playout of media.

Copyright © 2024 Vizrt

Transmission : Hold

Event	Time	Device	Status	Protect	PStatus	Sec	Type	ID	Seg
1	12:33:54.05		SKIPPED				A	00 HELLO O	
2	12:34:08.04		RAN SHOR				A	10 Bus crash	
3	12:34:11.09		SKIPPED				A	10 Dutch arr	
4	12:34:15.13		DONE				A	10 Gaza Wra	
5	12:34:40.23	VIZRT:1	PLAY				A	ddd	
6	12:35:13.08	PR2A:2	CUED	PR1A:2	CUED		A	00 HELLO O	
7	00:00:00.00	VIZRT:2	CUED			sAV	PT	zzz	
8	12:35:33.13	PR2B:2	CUED	PR1B:2	CUED		A	10 Bus crash	
9	12:41:42.00	PR2A:1	STOP	PR1A:1	STOP		A	10 Dutch arr	
10	12:42:32.20	PR2B:1	STOP	PR1B:1	STOP		A	10 Gaza Wra	
11	12:43:58.05	PR2A		PR1B			A	00 HELLO O	
12	12:44:18.10	PR2A		PR1B			A	10 Bus crash	
13	12:50:26.22	PR2A		PR1B			A	10 Dutch arr	
14	12:51:17.17	PR2A		PR1B			A	10 Gaza Wra	

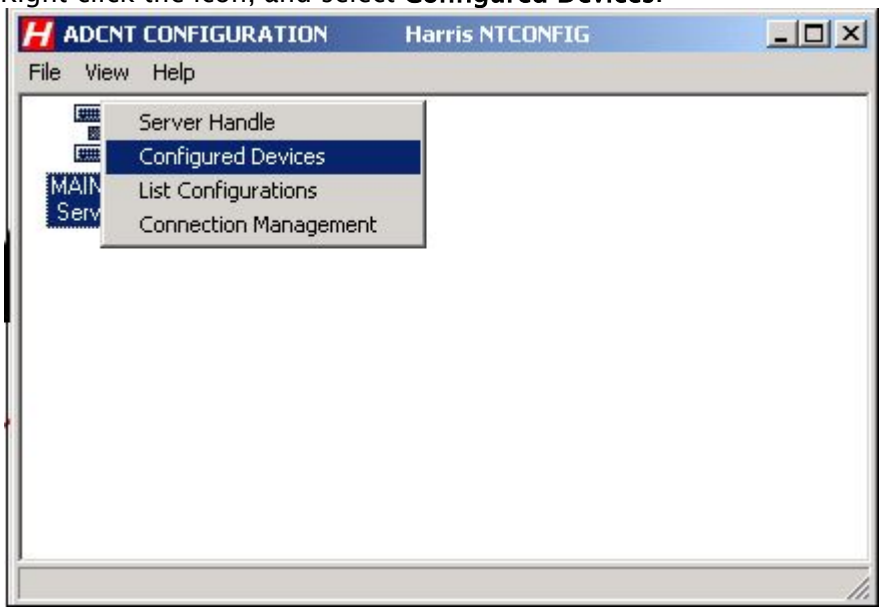
18.7.3 Triggering with USD

When configuring triggering by a Universal Sercom Device (USD) driver you can either configure it to use TCP or Serial Communication.

USD Device Configuration

To configure the Vizrt driver

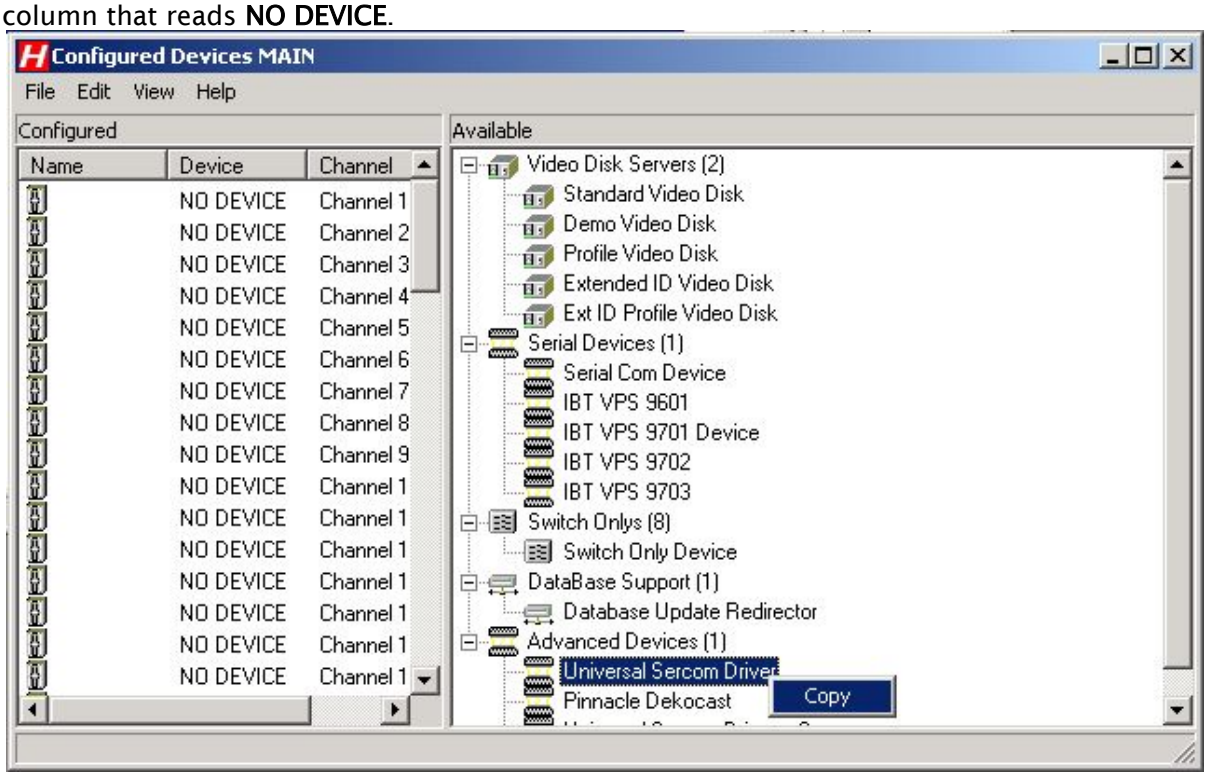
1. Launch the ADC-100 Configuration Utility and locate the icon that represents your Device Server.
2. Right-click the icon, and select **Configured Devices**.



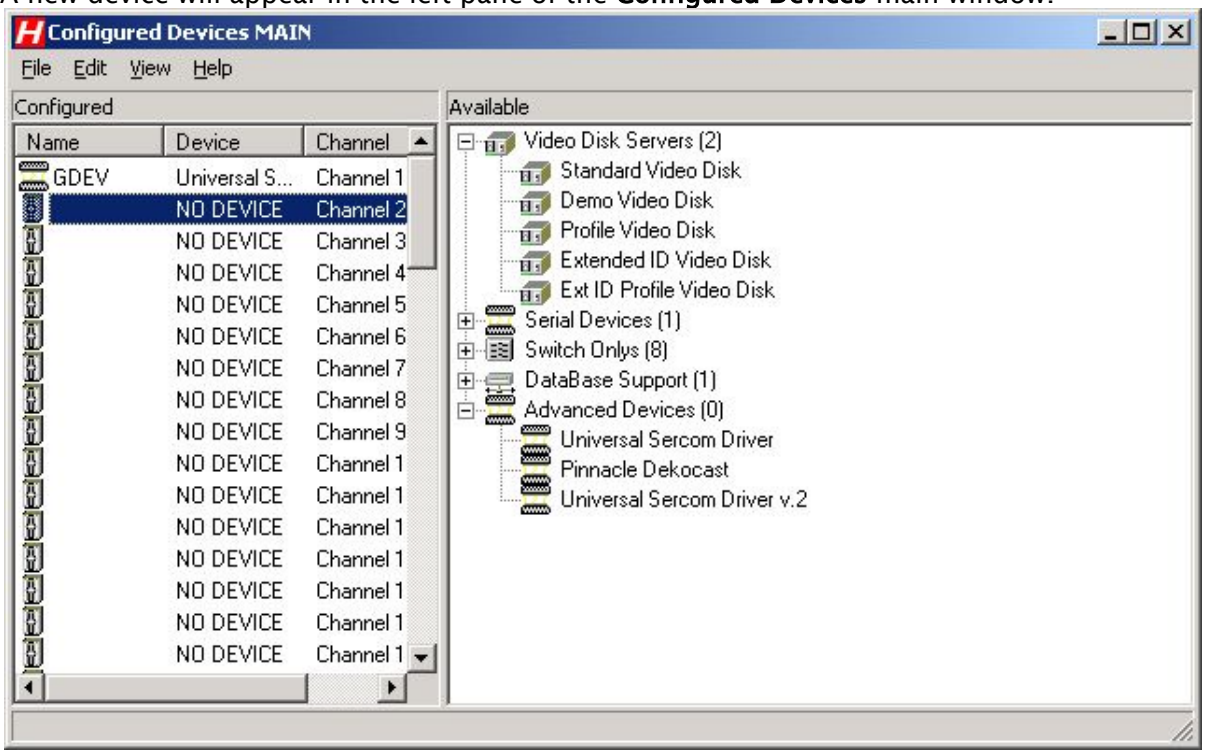
3. In the **Available** column on the right, under the **Advanced Devices** category, locate the **Universal Sercom Driver**. Click and drag the driver to one of the channels in the **Configured**

Copyright © 2024 Vizrt

column that reads **NO DEVICE**.

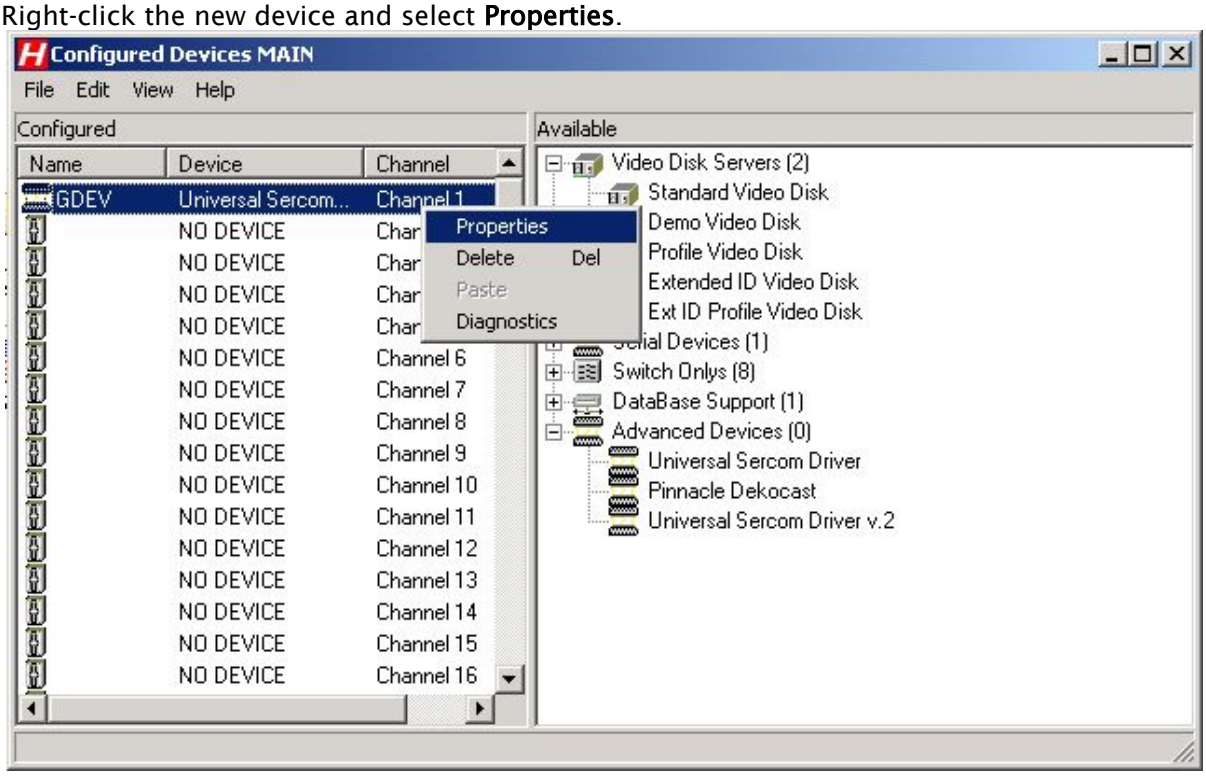


4. A new device will appear in the left pane of the **Configured Devices** main window.

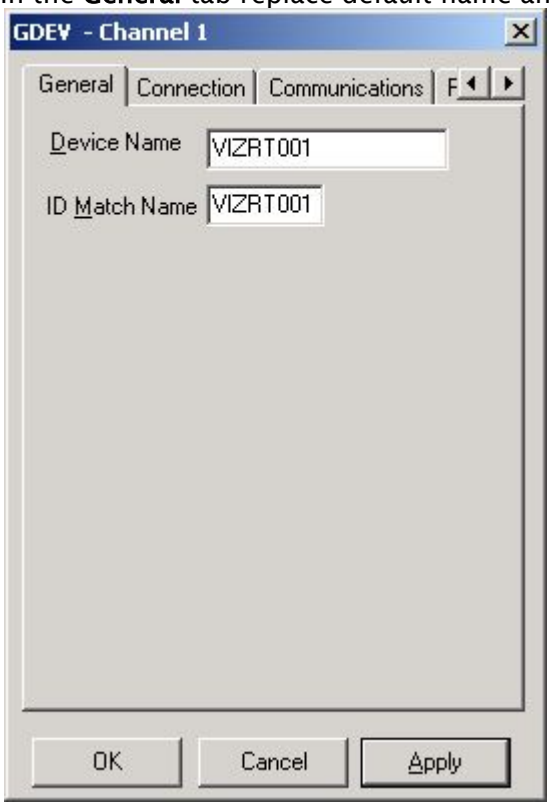


Copyright © 2024 Vizrt

5. Right-click the new device and select **Properties**.

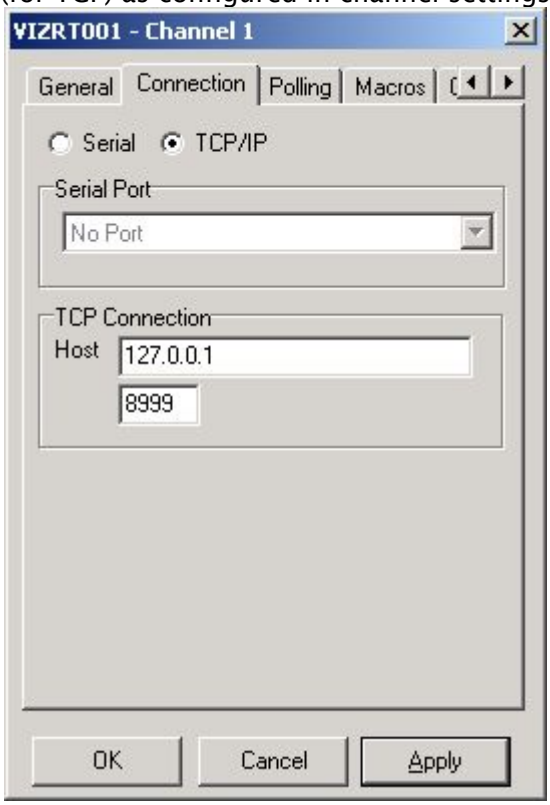


6. In the **General** tab replace default name and ID with **VIZRT001** and click **Apply**.



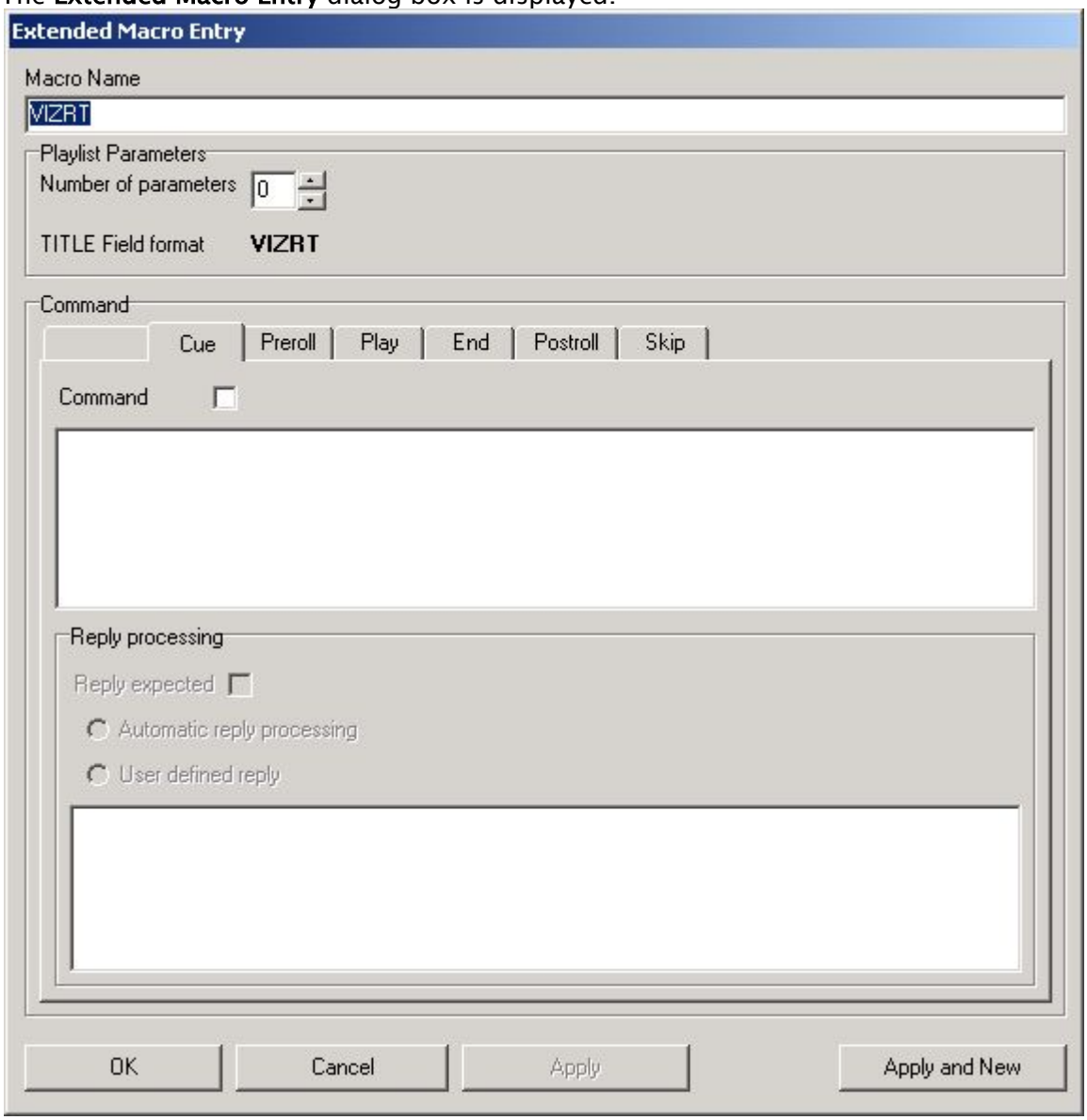
Copyright © 2024 Vizrt

7. In the **Connection** tab, insert Media Sequencer machine's IP/Host or Serial value and the Port (for TCP) as configured in channel settings for TCP or VDCP device settings.



Copyright © 2024 Vizrt

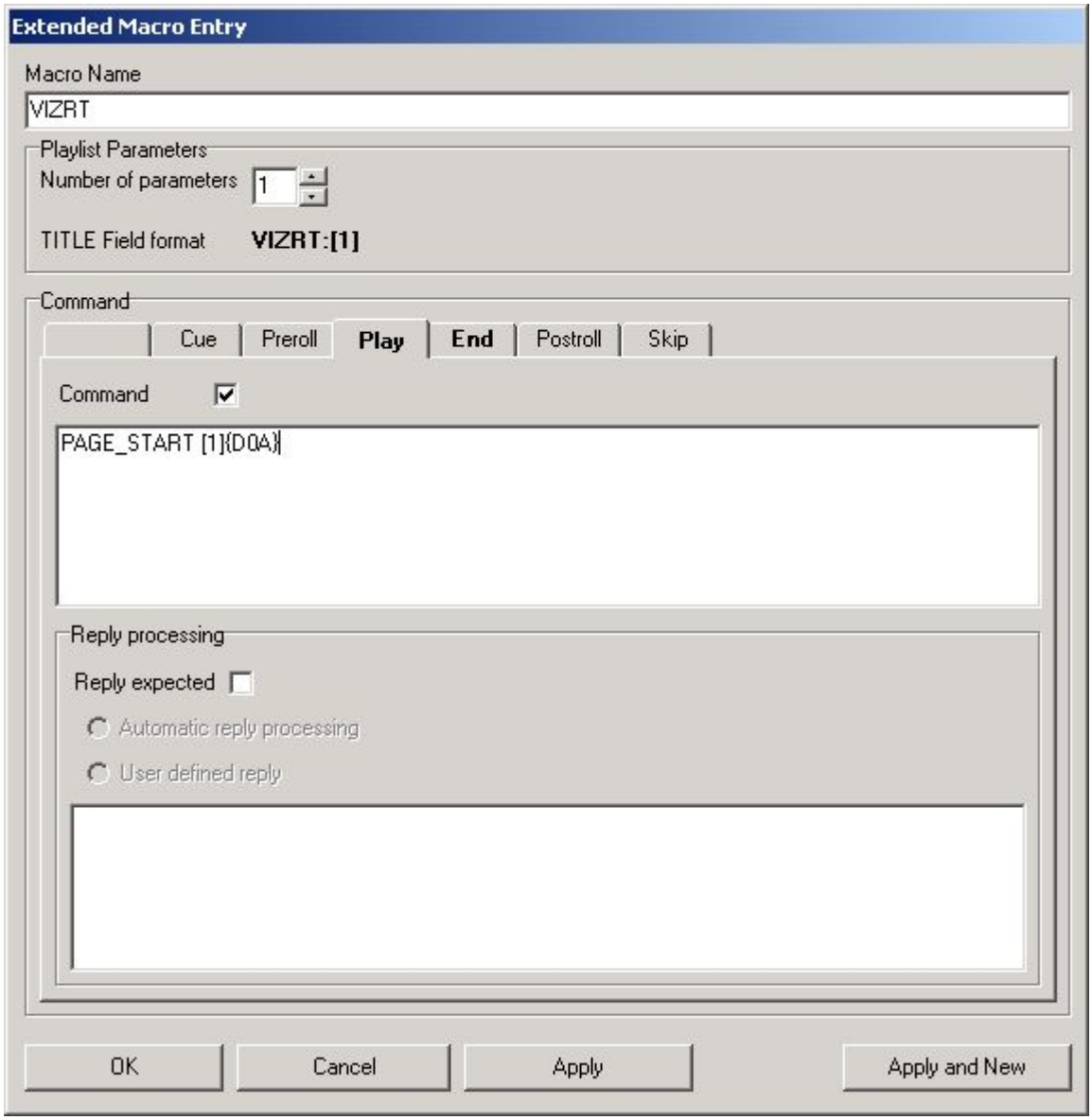
8. In **Macros** tab click the **Add** button to create a new macro.
   The **Extended Macro Entry** dialog box is displayed.



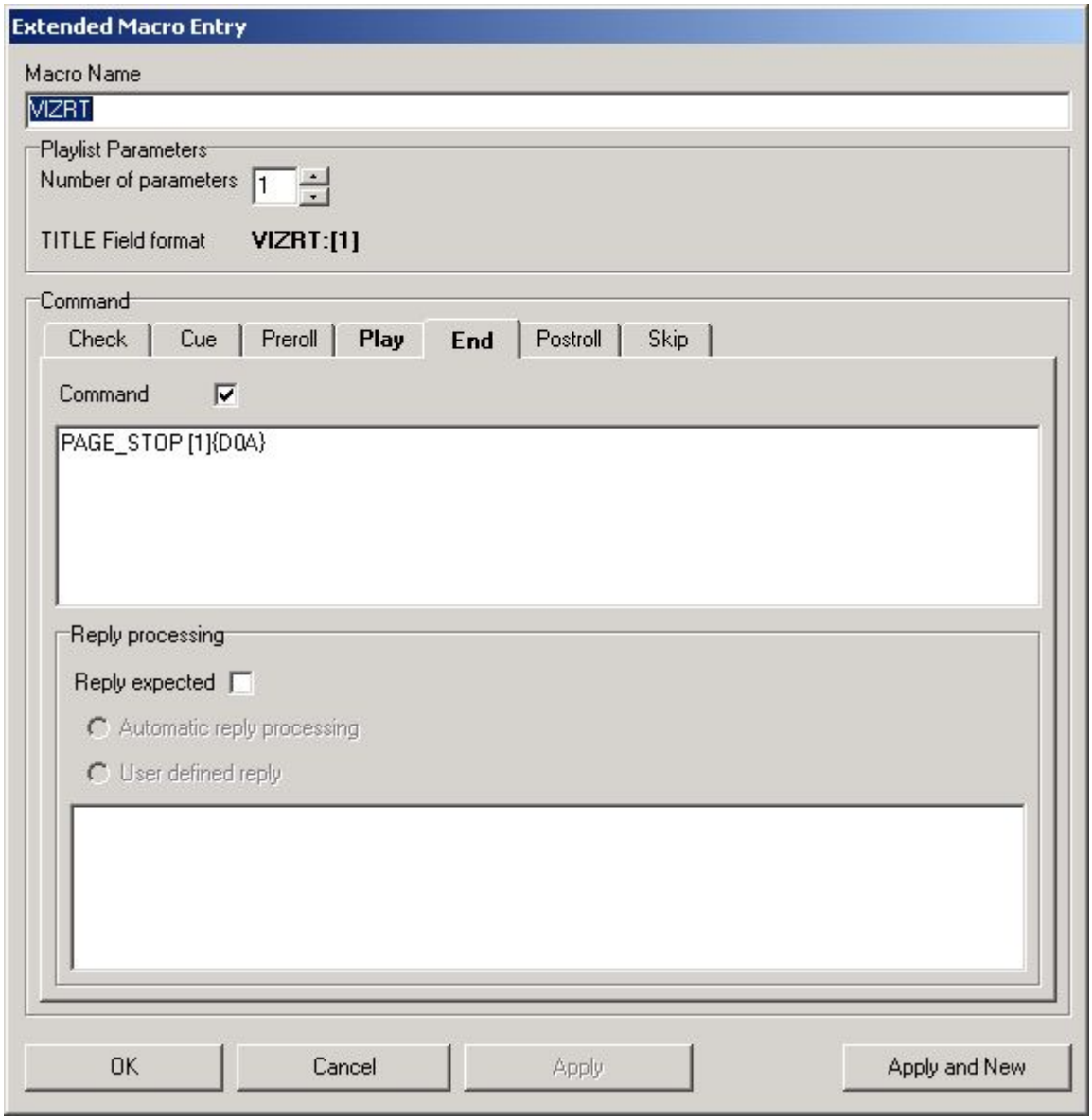
   a. In the **Cue** tab set the **Macro Name** to **VIZRT**.
   b. Set the **Number of Parameters** to **1**.
   c. Check the **Command** check box.
   d. In the **Command** field, enter the command to be used.

1. To edit the graphics stopping macro string, click the **End** tab.
2. Check the **Command** check box.
3. Set the **Number of Parameters** to **1**.
   Play will be triggered when an event starts. End will be triggered when an event ends.

Copyright © 2024 Vizrt



Copyright © 2024 Vizrt



Notes

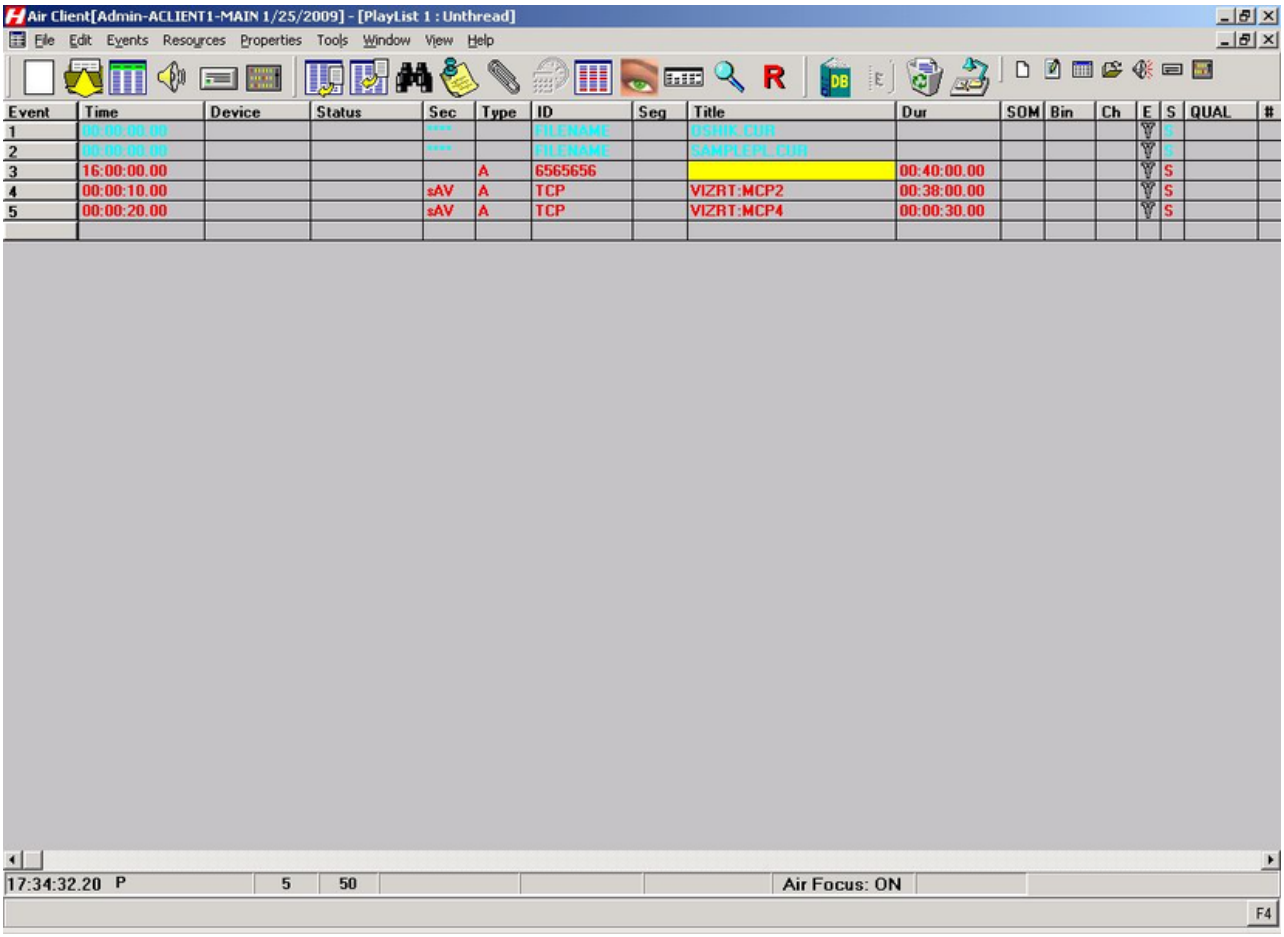
When an event is triggered (primary or secondary), the command **take id** is sent where id is the *Trigger ID* value in Viz Channel Branding and *ID field* value in ADC-100.

- **Harris playlist (**.LST) USD format:*
  - **TITLE field:** macro_name:trigger_id (Media ID). For example: VIZRT:MCP2
  - **ID field:** USD_NAME (device) name
  - **Reconcile Key field:** Template name
- **Harris macro format**: trigger_command [1]{D0A}. For example: *PAGE_START [1] {D0A}. {D0A}* is the char symbol for Enter.

Copyright © 2024 Vizrt

Operation

Viz uses A/V, Secondary A/V to execute the pre-loading and playout of media.



18.7.4 Automatic Playlist Updates Using RCV Files

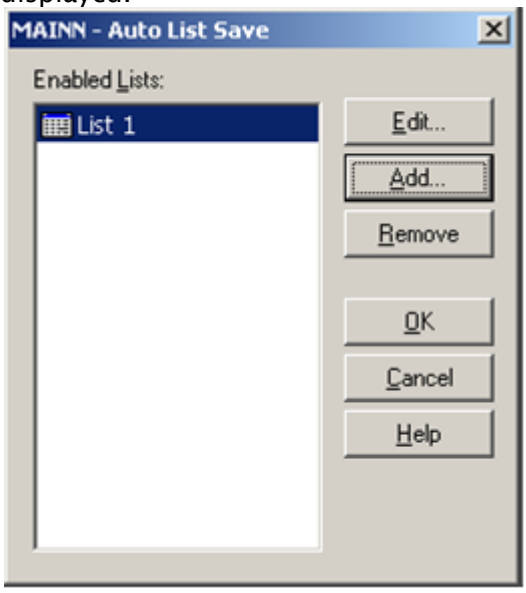
Auto List Save allows automatic list savings based on the number of list changes that occur on an active on-air transmission list.

Number of changes value indicates how often the list is automatically saved to an RCV file.

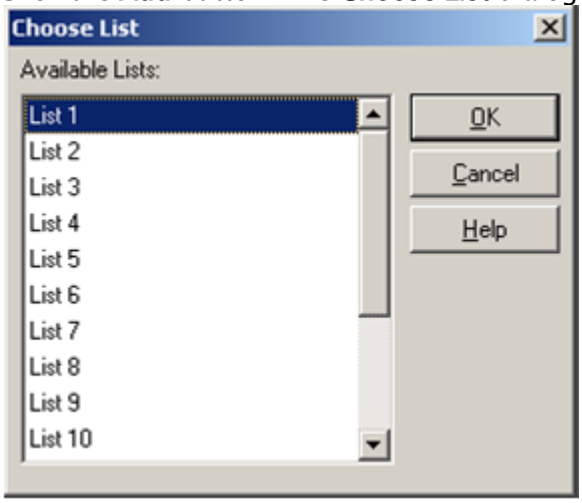
Copyright © 2024 Vizrt

To enable auto list save

1. From the main menu select **Properties > Auto List Save**. The **Auto List Save** dialog box is displayed.

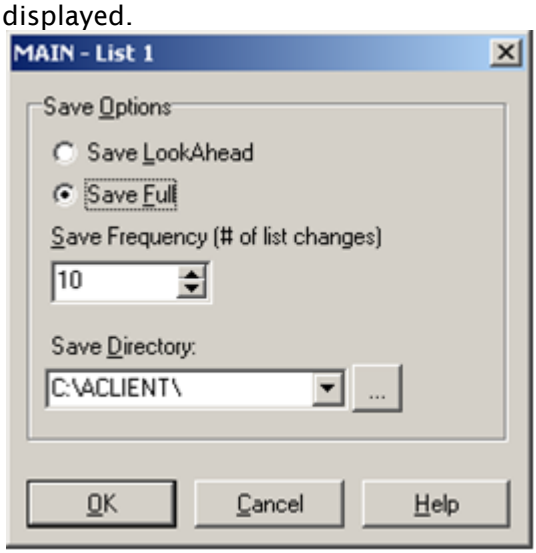


2. Click the **Add** button. The **Choose List** dialog box is displayed.



Copyright © 2024 Vizrt

3. Select a list from the Available Lists pane and click **OK**. The **Save Options** dialog box is displayed.



4. **Save LookAhead**: (Default: enabled) Check to enable save of the LookAhead list only.
5. **Save Full**: Check to enable save of the full list.
6. **Save Frequency**: (Default: 10) Specify the number of changes required before a list save is initiated. This is the value of how many events go by, NOT how many edits are performed.
7. **Save Directory**: Specify the save directory for the saved list. Click '...' to browse for a location. The import/update process will scan the specified folder for new RCV file updates.
8. Click **OK**. The list is added to the **Enabled Lists** pane.

Copyright © 2024 Vizrt

18.8 D-Series By Harris Broadcast

This section describes how to configure the Playout Automation for Enterprise Environments system (D-Series) by Harris Broadcast.

- General Information and Setup
- Updating Live
- Limitations

18.8.1 General Information and Setup

Setup on the solution side is similar to the ADC-100 setup; however, Universal Sercom Device is not available, and the LST file provided by the D-Series is not binary but text file based.

18.8.2 Updating Live

VICC: D-Series automatically generates text format *.VICC files into a drop-folder, a line per primary event (secondary events in the primary event's line). The import/update can scan that folder and update playlists accordingly.

18.8.3 Limitations

- VICC file size is limited so it may not contain all events in the playlist but only a portion of them.
- VICC file will not provide all the information available in scheduling system.
- Maximum number of playout channels exported = 25.
- Maximum number of schedule events exported per channel = 1000.
- Maximum number of as ran events exported per channel = 50.
- Maximum number of fields exported per event = 16.
- Maximum length of each event record = 256 characters.

Copyright © 2024 Vizrt

19 Viz One Integration

This chapter provides an overview necessary steps before using Viz One:

- Viz One and Viz Engine Configuration
- Viz Channel Branding Configuration
- Video Preview Configuration
- Installation Options

19.1 Viz One And Viz Engine Configuration

When integrating Viz One with Viz Engine you must:

- Allow Viz One to *transfer* assets to Viz Engine by installing the MediaFTP service on the Viz Engine.
- Allow Viz One to *monitor* Viz Engine's transferred assets by installing the FSMon service. For more information on how to install these services, see the *Viz Enigne Administrator Guide* on Integrating with Viz One.

Once Viz Engine has been configured to allow Viz One transfer and monitor assets to/on the Viz Engine, you can start to configure Viz One.

When configuring Viz One you must:

- Add a user to Viz One that will allow other applications to *search* for Viz One assets and metadata.
- Add a Viz Engine as a *video server* to Viz One.
- Define the *storage* and *access methods* Viz One requires in order to transfer and monitor assets to/on the Viz Engine.

  For more information on how to add users, and how to add and define Viz Engines as video servers, see the *Viz One Deployment Guide* on Integrating Video Production with Broadcast Graphics.

19.2 Viz Channel Branding Configuration

When integrating with Viz One you must:

- Allow solution to *resolve* the *schedule's house IDs* by matching them against Viz One assets. Once resolved, the playlist can be populated with assets from Viz One. See also Import Options on how to create placeholders in Viz One for missing House IDs.
- Allow solution to *search* for and *preview media assets* locally before adding it to a template/page and/or the playlist.
- Allow solution to *configure one or multiple device channels* (Viz Engine) with a Viz One connection for asset transfers to the same device channel.

Copyright © 2024 Vizrt

For more information on how to configure Viz Channel Branding's Viz One connection and playout devices, see the Viz One and Configuring Playout Devices sections.

19.3 Video Preview Configuration

If you are previewing proxy versions of video from Viz One using VCP's Newsroom Component (Timeline Editor) or Viz Engine (Viz Engine installed on your control client machine) you must install video codecs. These are not part of Vizrt's standard installation.

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

Note: Codecs are only required when local preview is done, as in the cases above. Playout of the high resolution versions do not require a codec installation.

The following procedures will guide you through the necessary installation steps:

- To install codecs for local preview
- To set preferred decoder

19.4 Installation Options

Codecs are available from several suppliers. The list below includes several suggestions:

- FFDShow MPEG-4 video decoder and Haali Media Splitter
- LAV Filters video decoder and splitter
- MainConcept video decoder and splitter

IMPORTANT! On Windows 7 machines, the default installation of the LAV filters results in the *LAV splitter* and the *Windows video decoder* being used. This can lead to problems using the Timeline Editor. This situation can be avoided by ensuring that the *LAV decoder* is selected using the Windows 7 filter tweaker tool. See **Set a Preferred decoder.**

Note: The display accuracy of the Timeline Editor can be effected by the type of codecs installed.

Some video decoders function in a way that limits the accuracy of the Timeline Editor when displaying frames. This must be taken into account in workflows that involve accurate placement of data elements, mark-in and mark-out points, and poster frames using the Timeline Editor.

If using the *FFDShow* video decoder, the displayed frame for the poster frame, mark-in and mark-out features may be up to 0.5 seconds away from the position of the marker.
Users that need a high level of frame accuracy should consider using other solutions.

Copyright © 2024 Vizrt

The video decoders from *MainConcept* and *LAV* operate in a different way; the Timeline Editor displays frames accurately when using either of these video decoders.

Previewing videos in the Timeline Editor works accurately, regardless of the video decoder used.

19.4.1 To install codecs for local preview

The example below sets up support for h.264 playback using the FFDShow MPEG-4 codec package and a Matroska Splitter from Haali.

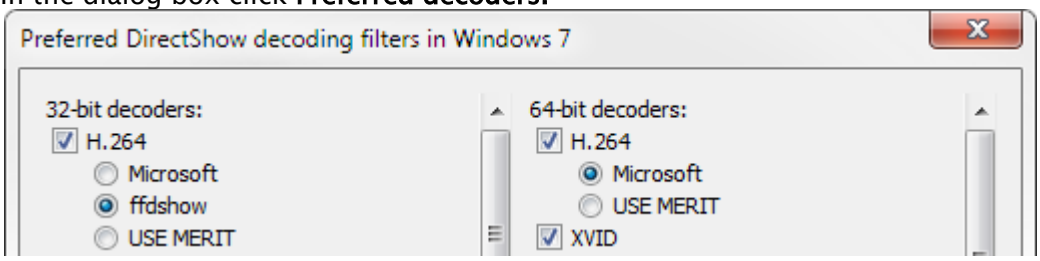
> ⚠ **Note:** You must have your own license for clip playback as FFDShow does not come with a decoding license.

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

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

19.4.2 To set preferred decoder

1. **Run** the **Windows 7 DirectShow Filter Tweaker**.
2. In the dialog box click **Preferred decoders.**



3. Set your **MPEG-4/H.264 32-bit decoder** to *FFDShow.*
4. *C*lick **Apply & Close**.
5. Click **Exit**.

See Also

- Working with Video Clips

Copyright © 2024 Vizrt

20 Appendix

The appendix covers the following topics:

- Upgrading Viz Channel Branding
- Vizrt Data Format (VDF)
- Page Content Filling Aliases
- Installing and Upgrading the Microsoft .NET library
- Enabling Windows Crash Dumps
- Configuring Import and Update Folders for Network Drives
- Promo Server Remote Configuration REST API
- Logging
- Glossary

Copyright © 2024 Vizrt

20.1 Upgrading Viz Channel Branding

Vizrt recommends that customers contact their local Vizrt representative before upgrading. Tools are available to assist upgrade from previous versions of Viz Channel Branding.

IMPORTANT! The **Upgrade Tool** shipped with Channel Branding will upgrade an existing installation from legacy Multichannel 2.6.2 or higher, and any Channel Branding release up to the current version. If upgrading from a version *older* than Multichannel 2.6.2 you will have to use the upgrade tools shipped with the Multichannel 2.6.2 installer.

To upgrade Viz Channel Branding for each version, follow the instructions below.

- Working with Upgrade Tool
- Upgrading from Legacy Viz Multichannel from 2.6.2+ to 4.x and Migrating to Viz Channel Branding 5.0+

20.1.1 Working with Upgrade Tool

Viz Channel Branding provides an update tool which can be found at *C:\Program Files\Vizrt\Viz Channel Branding\Upgrade Tool*.

1. Double-click the program to open it.
2. Fill out Media Sequencer IP address or hostname and click **Connect** to show a list of existing channels.
   The channels in the list which are possible to upgrade are shown with an **Upgrade** button.
3. Click **Upgrade** to begin.
4. The **Upgrade Status** will show a *Fully Upgraded* when the upgrading process is completed.

20.1.2 Upgrading from Legacy Viz Multichannel from 2.6.2+ to 4.x and Migrating to Viz Channel Branding 5.0+

The included Upgrade Tool include support for migrating the Viz Multichannel channels to Viz Channel Branding.

1. Double-click the program to open it.
2. Fill out Media Sequencer IP address or hostname and click **Connect**.
3. Switch to tab **Migrate from Viz Multichannel** to see list of available channels to migrate
4. Click **Upgrade and Migrate** on the relevant channels to migrate it over and upgrade it to 4.99.0 (preparation-version)
5. Switch tab to **Upgrade Viz Channel Branding channel data** to see list of channels in Channel Branding
6. Click **Upgrade!** on any channel that is listed and marked with such red button to finalize the migration and upgrading of the channel(s)

Copyright © 2024 Vizrt

20.2 Vizrt Data Format (VDF)

To implement an external update service, you need to parse and return a Vizrt Data Format (VDF) document to Media Sequencer. The document is posted to the update service by Media Sequencer.

Working with the data formats requires knowledge on how the data is structured when sent from Media Sequencer to Viz Engine. You can do this by, for example, looking at the Media Sequencer's console, and then work with the data that is sent.

Vizrt recommends that you contact your local Vizrt Support representative and obtain the current version of the VDF.

20.2.1 Example of a VDF document

```
<?xml version="1.0" encoding="utf-8"?>
<payload xmlns="http://www.vizrt.com/types">
<field name="foo">
    <value>bar</value>
    </field>
    <field name="otherfoo">
    <value>otherbar</value>
    </field>
</payload>
```

Field Names refer to Field Identifiers, a property available to all Viz Artist control plugins.

Copyright © 2024 Vizrt

20.3 Page Content Filling Aliases

Built-in aliases can typically be used in the Data Content Filling dialog and will outline their usage and function.

- Page Content Filling Aliases
- Page Content Filling Example
  - To use begin_date_formatted for current event
  - To use begin_date_formatted for next event
  - To use begin_date_formatted for event later tonight
  - To use begin_date_formatted for event tomorrow

20.3.1 Page Content Filling Aliases

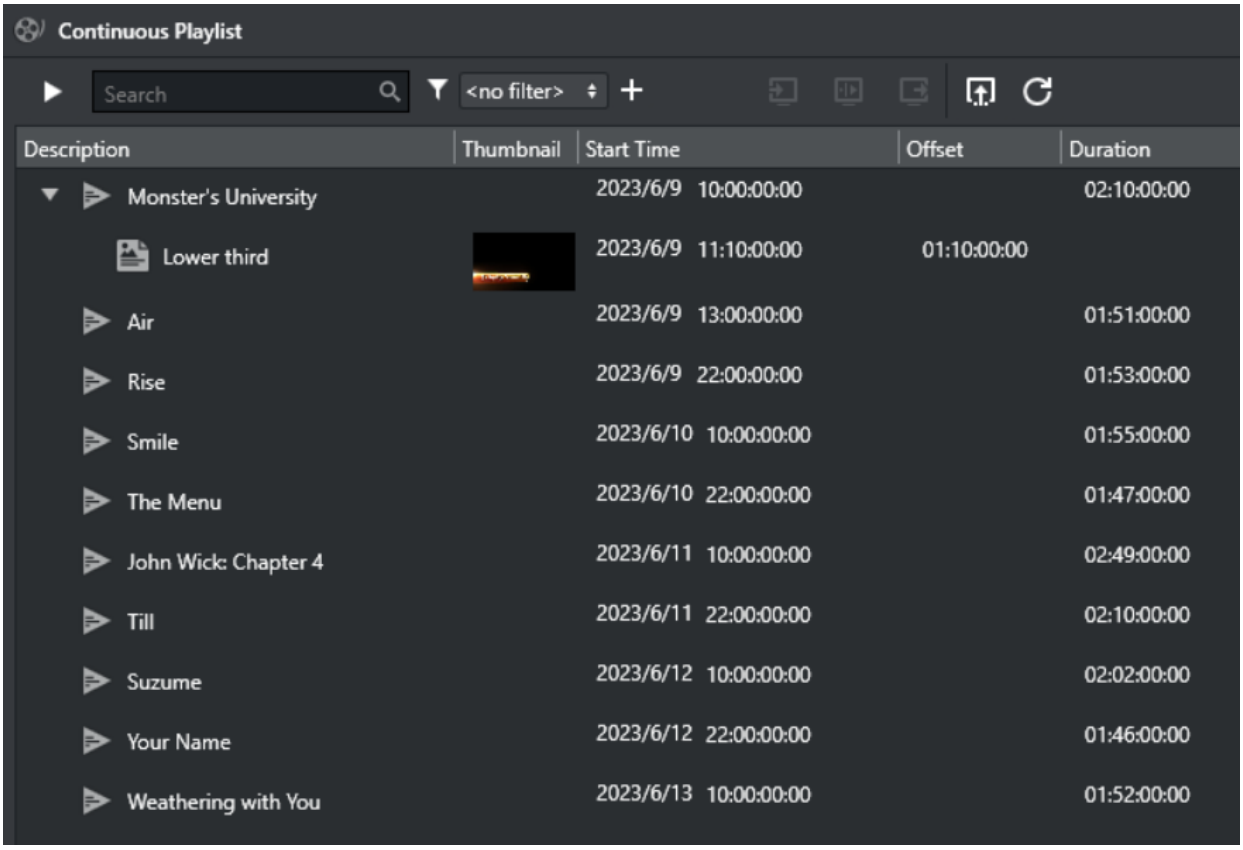
- **begin_date:** Date property extracted from the **begin_at** attribute, which refers to the *date* that the primary event is scheduled for play out.
- **begin_time:** Time property extracted from the **begin_at** attribute, which refers to the *time* that the primary event is scheduled for play out.
- **begin_date_formatted:** It is a way of representing date and uses **Present as** options.
- **Present as:** It has 4 options:
  - **Date:** The default setting for attribute *begin_date_formatted*. It uses *Now, Next*, *Today*, *Tonight* and *Tomorrow* states to define date. To represent a program which is scheduled for the days later, it symbolizes the date of corresponding playlist as: *yyyy/mm/dd.*
  - **WeekDay:** Uses *Now, Next*, *Today*, *Tonight* and *Tomorrow* states to define date. To represent program which is scheduled for the days later, it symbolizes the scheduled day of corresponding playlist as a number from 0 to 6, where 0 means Sunday, 1 means Monday and so on.
  - **Relative Day:** Uses *Now, Next*, *Today*, *Tonight* and *Tomorrow* states to define date. To represent program which is scheduled for the days later, it symbolizes total number of days from running playlist to selected playlist.
  - **Absolute Date:** Uses *yyyy/mm/dd* to represent date. And to represent a program which is scheduled for the days later it uses the same symbol: *yyyy/mm/dd.*
    - **Now**: Represents the program currently is playing.
    - **Next:** Represents the next program from currently selected program.
    - **Today:** Represents the program within the same playlist, where begin_time is less than 9 pm but doesn't indicate the program which is next from currently selected program.
    - **Tonight:** Represents the program within the same playlist, where begin_time is greater than 9 pm but doesn't indicate the program which is next from currently selected program.
    - **Tomorrow:** Represents the program from next day's playlist, which is scheduled to be played tomorrow.

Copyright © 2024 Vizrt

20.3.2 Page Content Filling Example

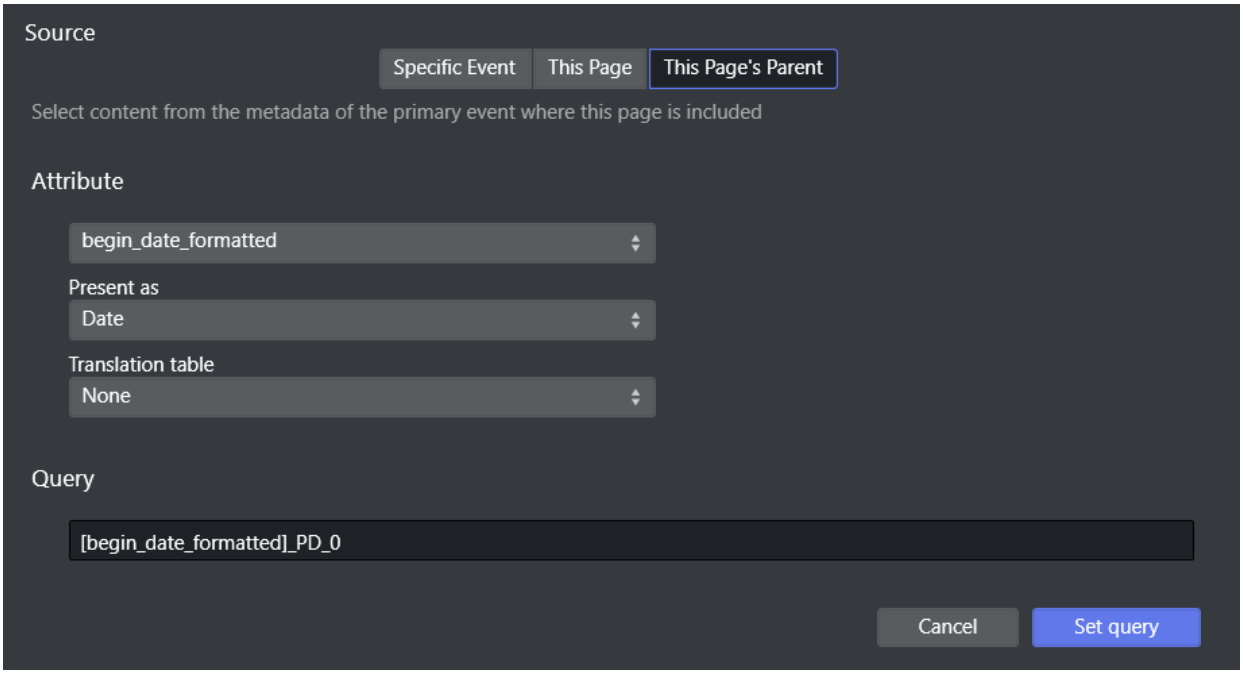
To use begin_date_formatted for current event

Consider an example Continuous Playlist as in the image below:

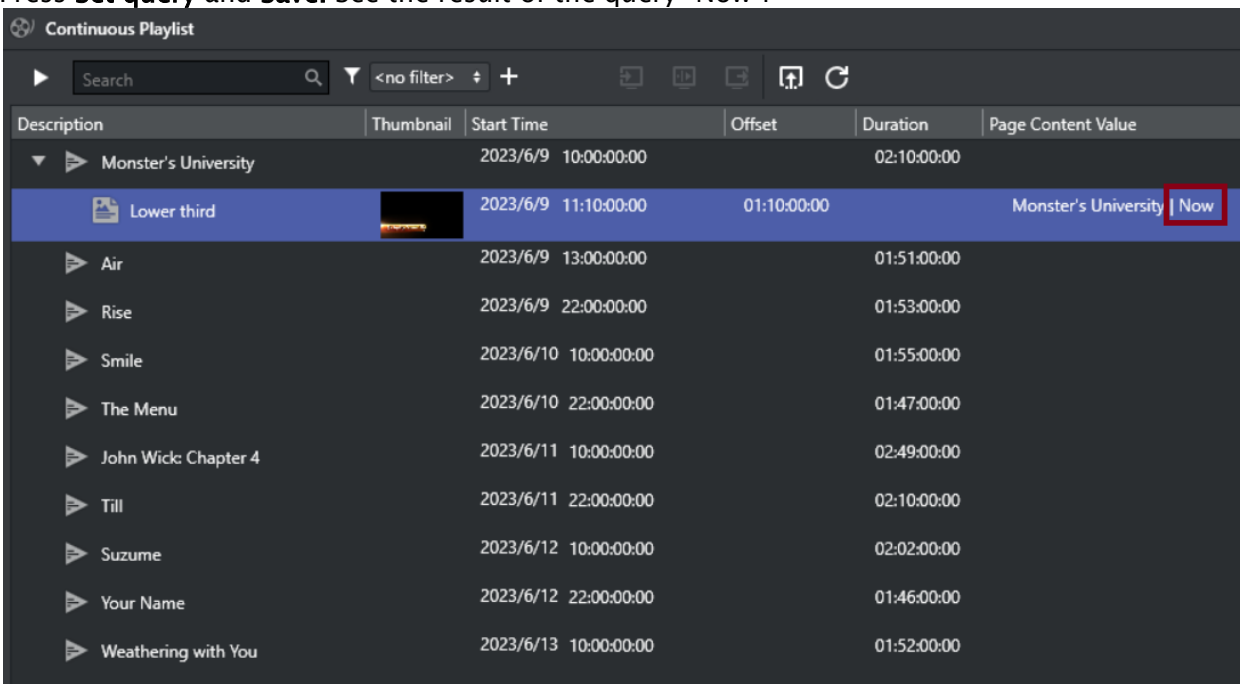


For page "Lower third", set query to get **begin_date_formatted** from current primary event:

Copyright © 2024 Vizrt



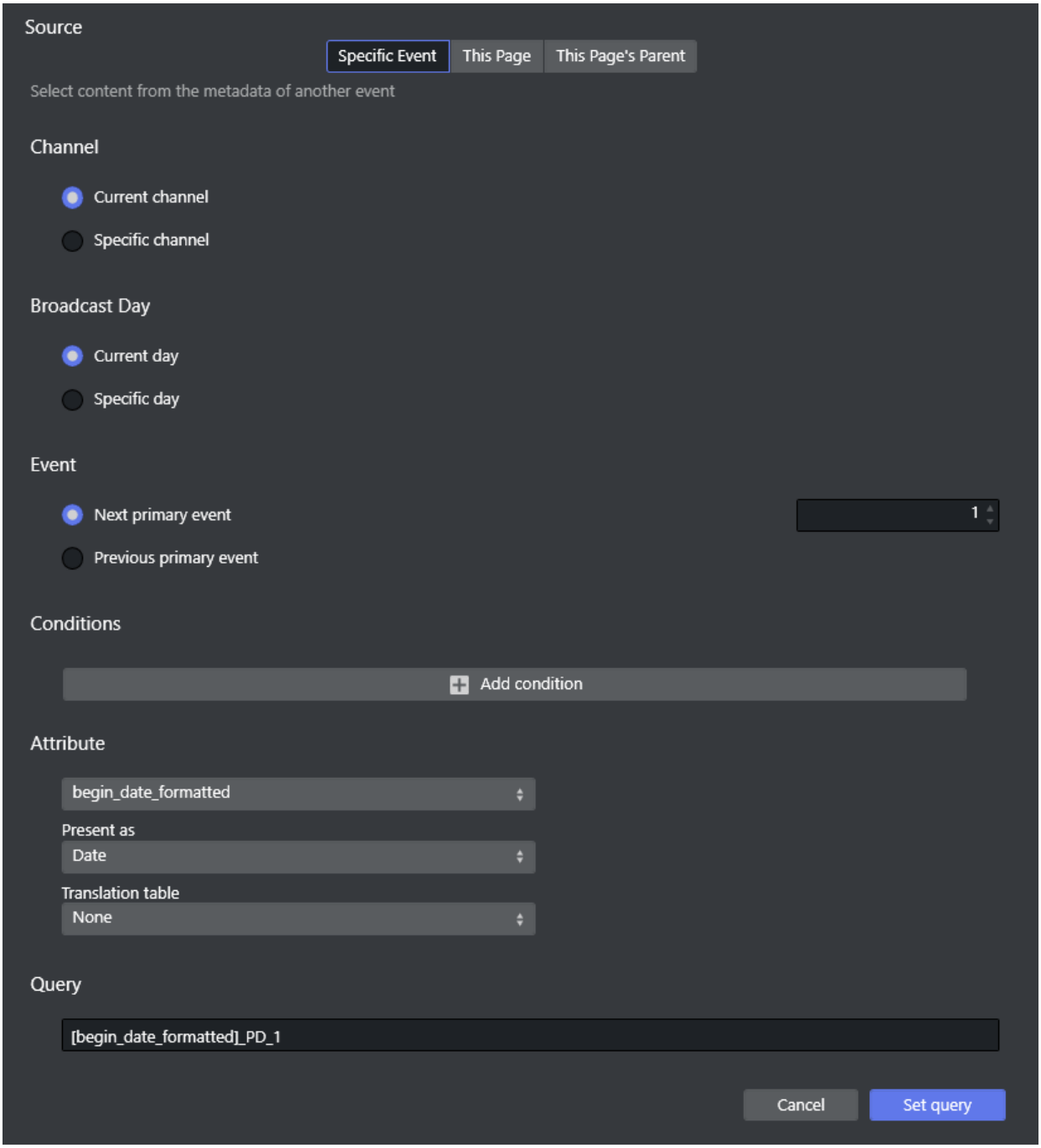
Press **Set query** and **Save.** See the result of the query "Now":



To use begin_date_formatted for next event

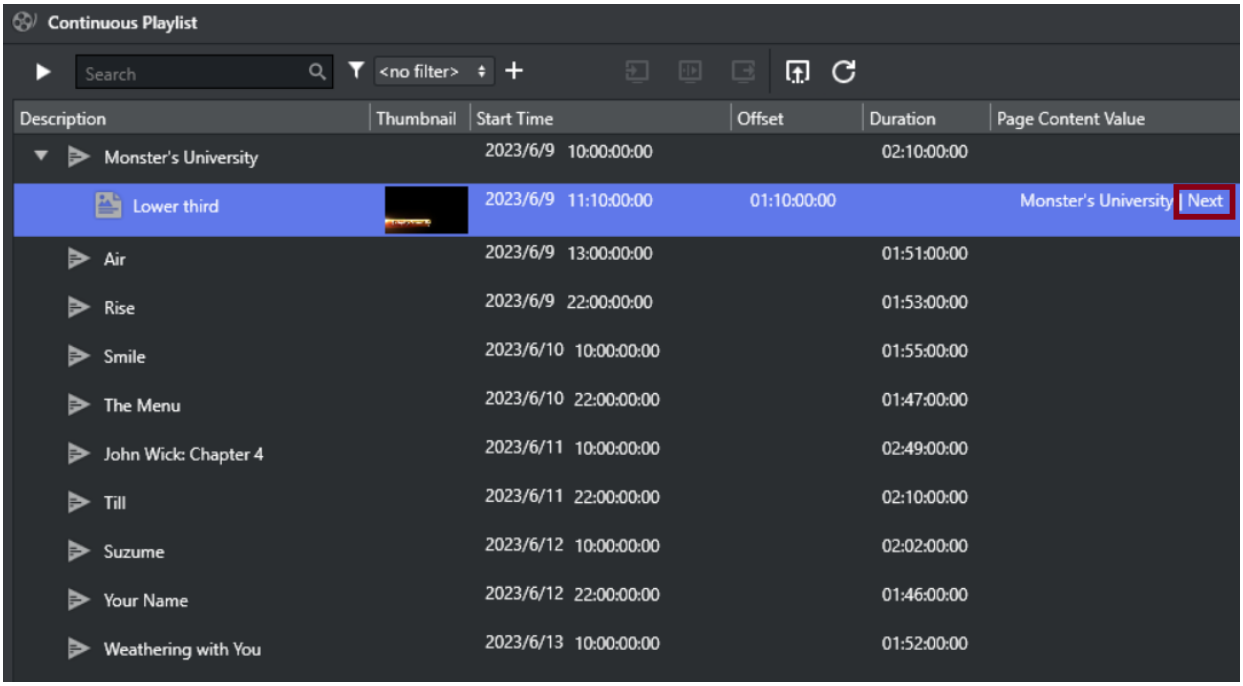
For page "Lower third", set query to get attribute **begin_date_formatted** from next primary event, which will get primary event with description "Air":

Copyright © 2024 Vizrt



Press **Set query** and **Save.** See the result of the query "Next":

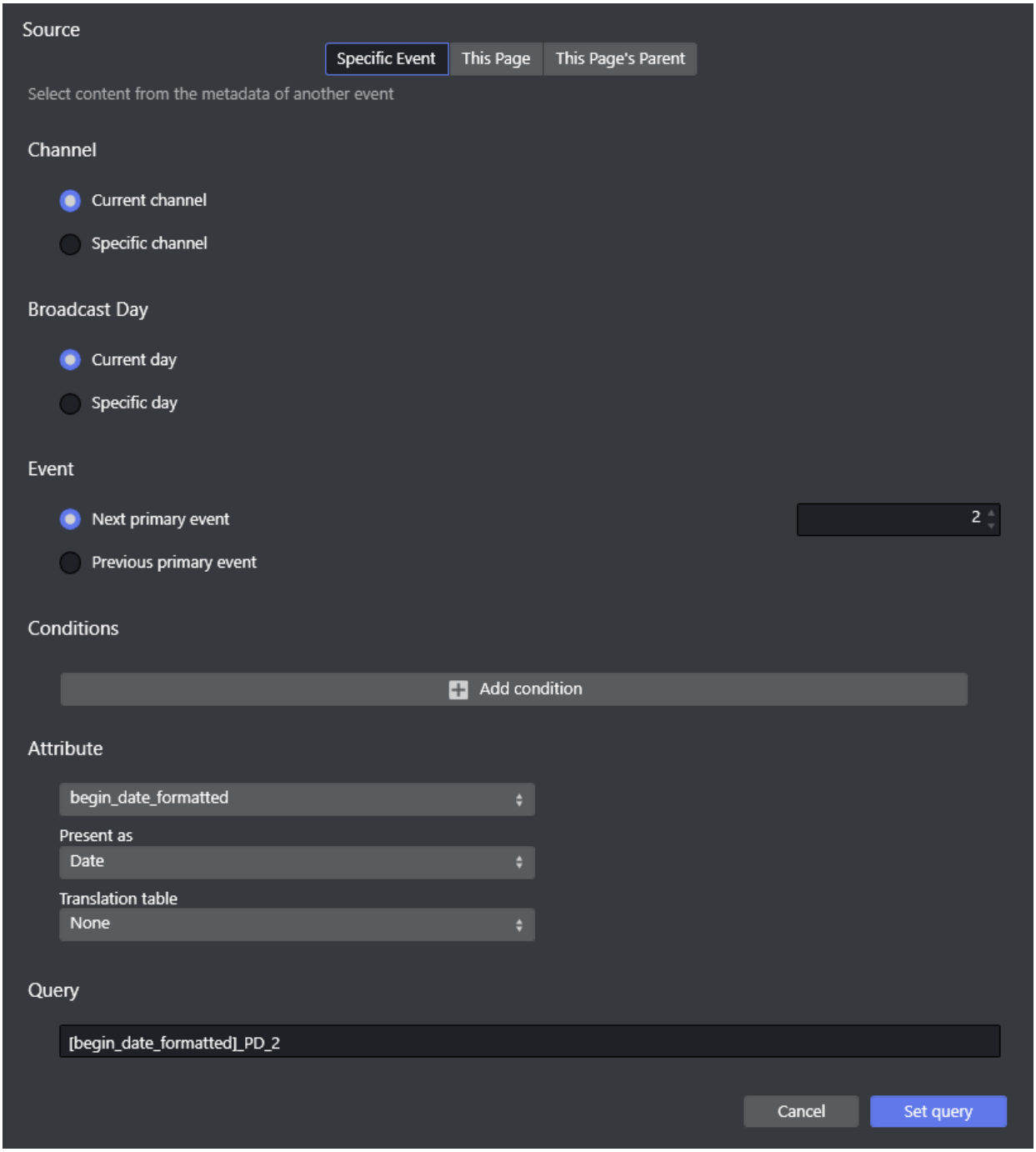
Copyright © 2024 Vizrt



To use begin_date_formatted for event later tonight

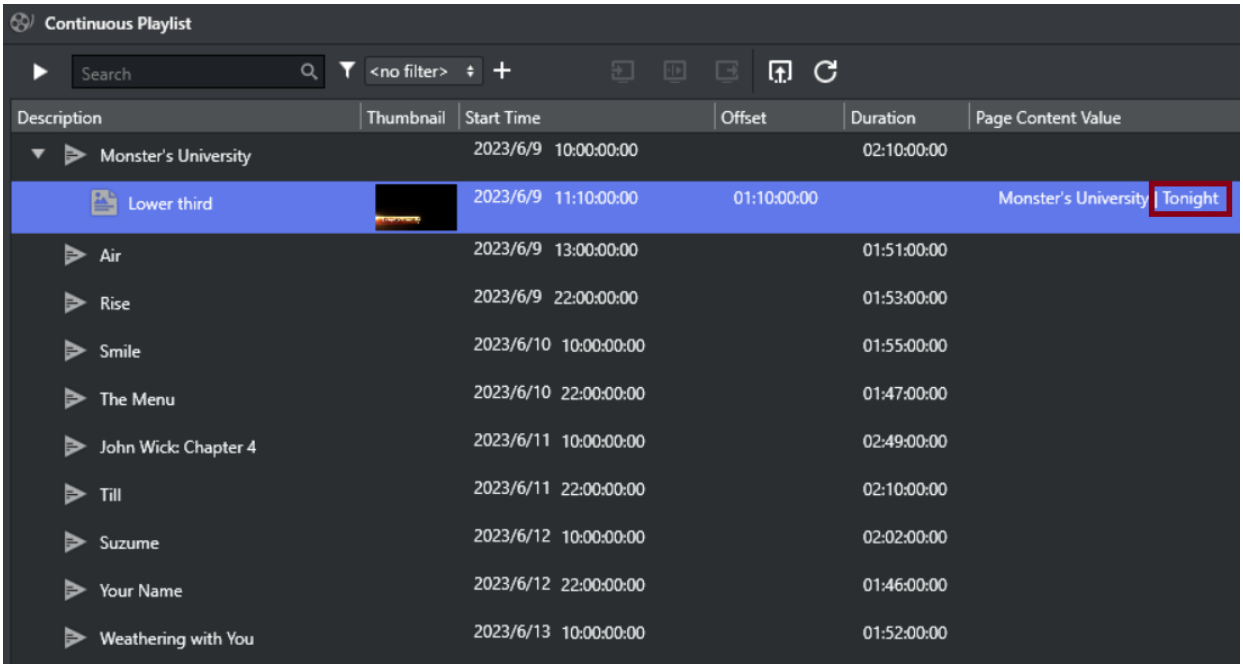
For page "Lower third", set query to get attribute **begin_date_formatted** from next 2 primary event, which will get primary event with description "Rise":

Copyright © 2024 Vizrt



Press **Set query** and **Save.** See the result of the query "Tonight":

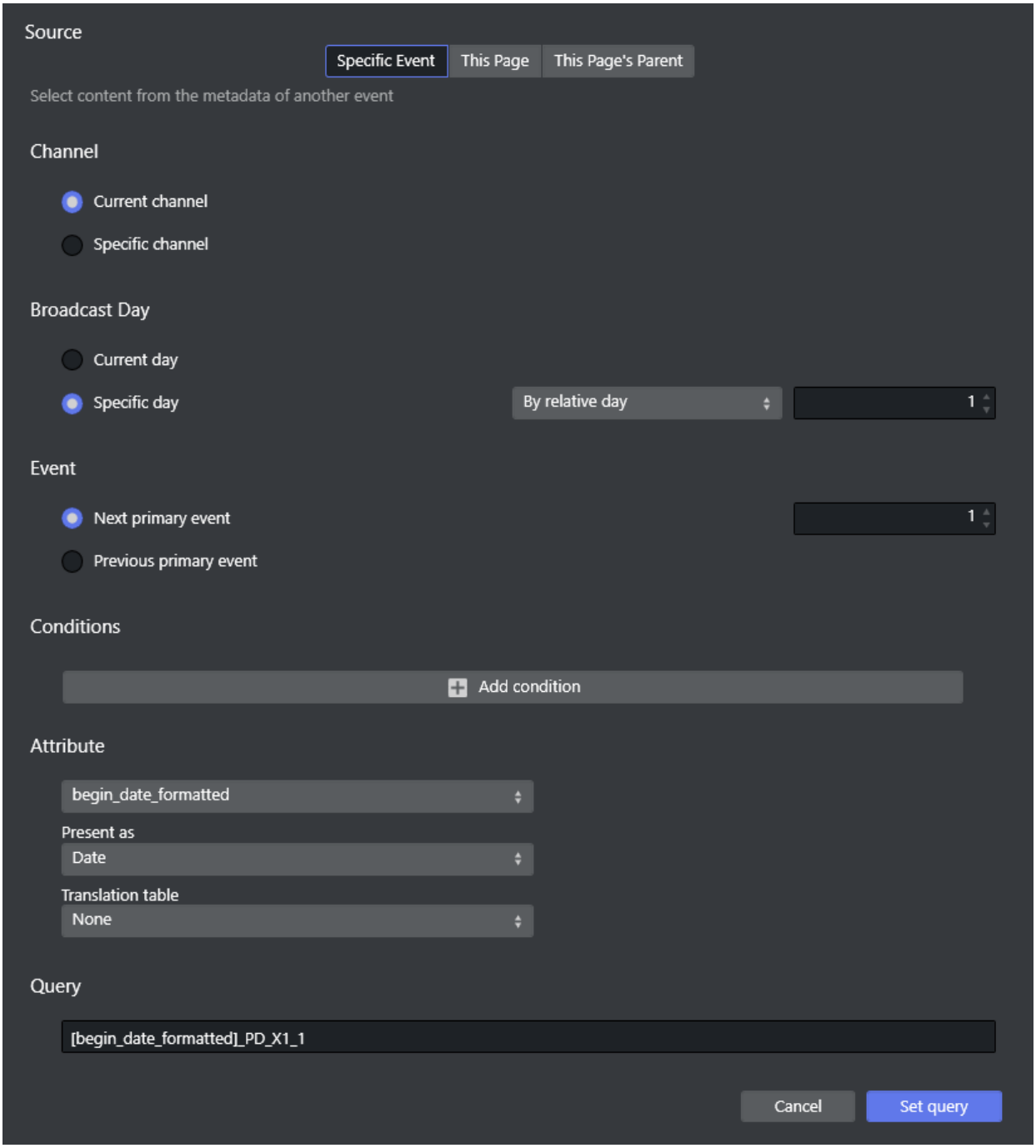
Copyright © 2024 Vizrt



To use begin_date_formatted for event tomorrow

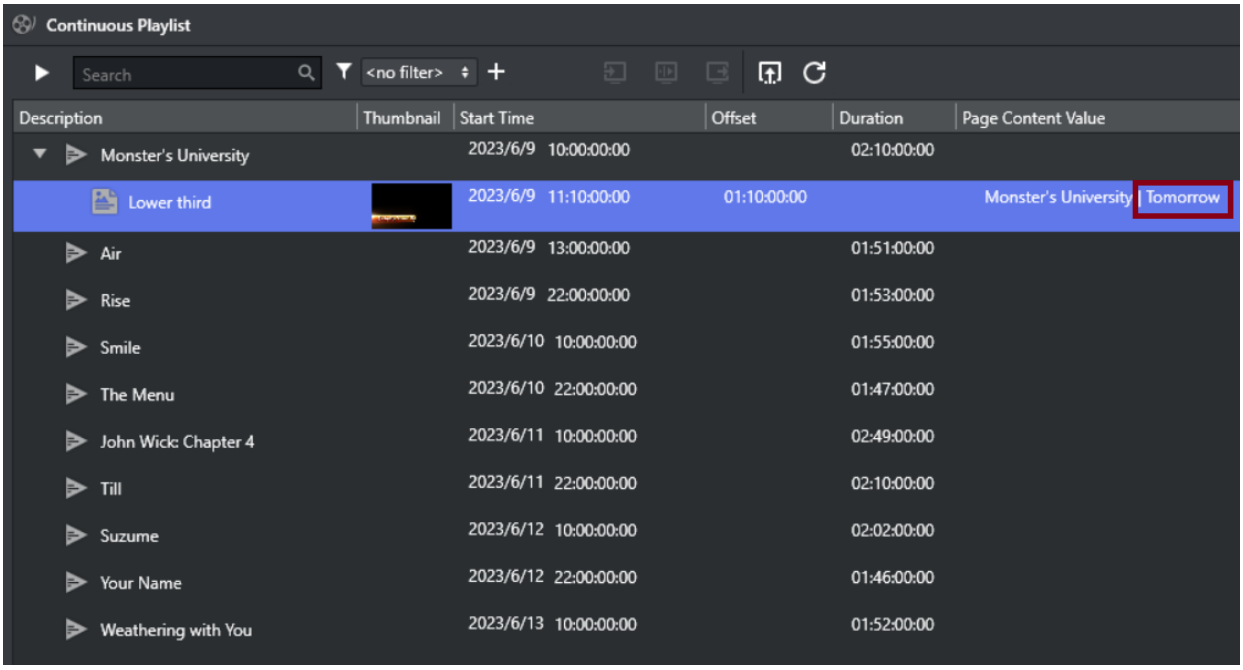
For page "Lower third", set query to get attribute **begin_date_formatted** from first primary event in the next day, which will get primary event with description "Smile":

Copyright © 2024 Vizrt



Press **Set query** and **Save.** See the result of the query "Tomorrow":

Copyright © 2024 Vizrt



For the states of the program which is scheduled for the days *later* than *tomorrow*, it depends upon the "**Present as**" settings.

Copyright © 2024 Vizrt

20.4 Installing And Upgrading The Microsoft .NET Library

Microsoft .NET is a software framework that runs primarily on Microsoft Windows. It includes a large library and provides language interoperability across several programming languages. Software that uses the .NET Framework Class Library, (as many programs from Vizrt), requires the .NET library. Most Windows machines have .NET installed.

Please consult the Microsoft .NET website for updates, installers and support for the .NET library framework. As a general rule, Vizrt software requires .NET version 4 or later. Always check your product Release Notes for the minimum required version of .NET for a specific Vizrt software release.

If an upgrade or installation of .NET is required go to http://www.microsoft.com/net/downloads.

You can check for the currently installed version of .NET with the command shell (Windows Start-button + Ctrl-R + cmd + ENTER) and give the command:

```
dir /b /ad /o-n %systemroot%\Microsoft.NET\Framework\v?.*
```

This will list out all the versions of .NET installed, latest first.

Copyright © 2024 Vizrt

20.5 Enabling Windows Crash Dumps

20.5.1 MS Windows Logging

It is recommended to enable Microsoft Windows to generate user-mode crash dumps.
This makes debugging easier, particularly if there are hardware or general windows issues affecting program behavior.

Enabling User-mode Dumps

This operations requires Windows 7 or higher.

To enable a user mode dump

- Configure the `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps` registry setting.

For more detailed information, see the Microsoft web information at Collecting User-Mode Dumps.

> ⚠ Refer to section Logging for more details about the logging options in relating to the solution directly.

Copyright © 2024 Vizrt

20.6 Configuring Import And Update Folders For Network Drives

A Windows service does not have access to network drives mapped with aliases such as M:\ when mapped using a regular domain user.

20.6.1 Working with Import and Update Folders

The following procedures describe how to configure this solution's components to use network drives for playlist imports and updates when running Media Sequencer as a Windows Service.

- To set playlist imports and update destination path for schedule files
- To enable automatic import
- To configure Media Sequencer network authentication
  - Microsoft Excel schedule files cannot be imported

To set playlist imports and update destination path for schedule files

1. Start Viz Channel Branding Client in **Scheduling** mode.
2. Open the **Configuration** settings.
3. Click Channel Settings **> Import/Export Settings > General Settings**.
4. Set the **Destination path** for the schedule files to import or update.

> ⚠ **Note:** For this case network paths, such as UNC paths (for example, \network\...) are supported; however, mapped drives are not.

To enable automatic import

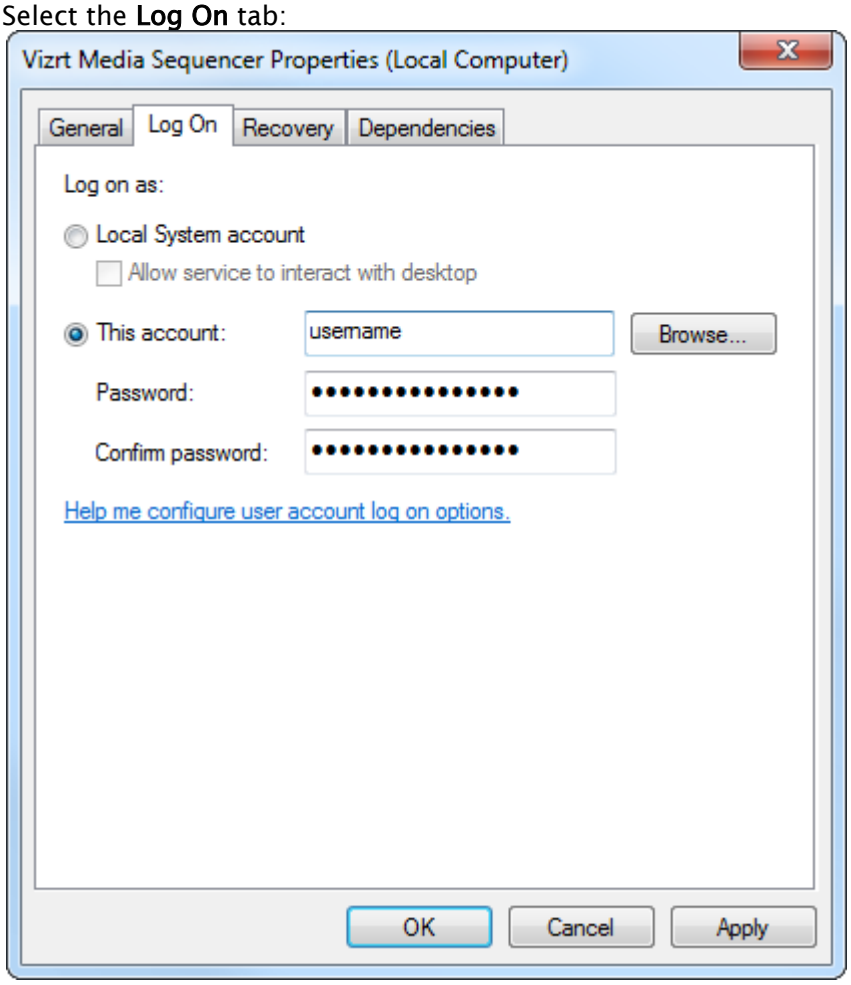
1. Click **Global Settings > Import/Export settings > Automatic Import**.
2. Set:
   - **Automatic import interval**
   - **Playlist importer folder path**
   - **Playlist importer log level**

To configure Media Sequencer network authentication

1. In Windows, click **Start**.
2. Open the **Control Panel**, select **System and Security > Administrative Tools**.
3. Open **Services**.
4. Select and right-click the **Vizrt Media Sequencer** service.
5. Select **Properties**.

Copyright © 2024 Vizrt

6. Select the **Log On** tab:



7. Select the **This account** radio button.
8. Enter a *username* and *password* that has access to the network share you want to connect to.
9. **Apply** and click **OK**.

> ⚠ **Note:** If Vizrt Media Sequencer service is already running, please *restart* the service for the new settings to take effect.

Troubleshooting

Microsoft Excel schedule files cannot be imported

> ⚠ **Note:** Microsoft Excel and other Microsoft Office applications are not designed to be run as a server application, and Microsoft does not support this usage.

1. Create a folder named Desktop in the following location:
   - 32-bit systems: *C:\Windows\System32\config\systemprofile*
   - 64-bit systems: *C:\Windows\SysWOW64\config\systemprofile*

Copyright © 2024 Vizrt

2. For more information, see the following knowledge base article on Arcana Development and/or the following Microsoft knowledge base article on *Considerations for server-side automation of Office*.

Copyright © 2024 Vizrt

20.7 Promo Server Remote Configuration REST API

The Promo Server is a windows service that is responsible for looking up information according to the Page Content Filling mapping for a page's tab-field. The configuration of the service defines what channels it should know about by linking the channel id and the address to an active Media Sequencer where the data is currently residing. The usage of the REST API allows for adjustment to this configuration while the service is running and there is no need to manually edit configuration files.

API Operations:

- GET API Root
- GET All Channels
- GET Channel
- GET All Machine Addresses under Channel
- GET Machine Address
- GET Status of Machine Address
- GET Global Settings
- POST Channel
- DELETE Channel
- PUT Channel
- POST Machine Address
- DELETE Machine Address
- PUT Global Settings

Copyright © 2024 Vizrt

20.7.1 GET API Root

The starting point for the structure of the REST API is found by calling the address and ip for the service. The port for Promo Server REST API is by default 9980.

Example request using curl against the service on a local machine would be: curl -X GET http://localhost:9980/

The rest of the guide assumes each request is prefixed with the localhost/ip and port as needed and only lists the end-points addresses.

Syntax

```
GET /
```

Request

```
GET / HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.promoserver+xml; type=root
<promoserver:root xmlns:promoserver="http://www.vizrt.com/promoserver">
  <promoserver:refLink rel="channels" type="application/vizrt.promoserver+xml;
type=channelsList" href="http://localhost:9980/config/channels" />
  <promoserver:refLink rel="globalsettings" type="application/vizrt.promoserver+xml;
type=globalsettingsList" href="http://localhost:9980/config/globalsettings" />
</promoserver:root>
```

Disallowed Actions

This end-point does not allow POST, PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

20.7.2 GET All Channels

Returns a list of channel-definitions.

Syntax

```
GET /config/channels
```

Request

```
GET /config/channels HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.promoserver+xml; type=channelsList
<promoserver:channelsList xmlns:promoserver="http://www.vizrt.com/promoserver">
    <promoserver:channelItem>
        <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/Test" />
        <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/Test" />
        <promoserver:refLink rel="machineaddresses" type="application/
vizrt.promoserver+xml; type=machineAddressesList" href="http://localhost:9980/config/
channels/Test/machineaddresses" />
        <promoserver:name>Test</promoserver:name>
    </promoserver:channelItem>
    <promoserver:channelItem>
        <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/TestCross" />
        <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/TestCross" />
        <promoserver:refLink rel="machineaddresses" type="application/
vizrt.promoserver+xml; type=machineAddressesList" href="http://localhost:9980/config/
channels/TestCross/machineaddresses" />
        <promoserver:name>TestCross</promoserver:name>
    </promoserver:channelItem>
</promoserver:channelsList>
```

Disallowed Actions

This end-point does not allow PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

20.7.3 GET Channel

Returns a definitions for a single specific channel.

Syntax

```
GET /config/channels/{channel}
```

Request

```
GET /config/channels/Test HTTP/1.1
```

Response - Channel returned

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.promoserver+xml; type=channel
<promoserver:channel xmlns:promoserver="http://www.vizrt.com/promoserver">
    <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/Test" />
    <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/Test" />
    <promoserver:refLink rel="machineaddresses" type="application/
vizrt.promoserver+xml; type=machineAddressesList" href="http://localhost:9980/config/
channels/Test/machineaddresses" />
    <promoserver:name>Test</promoserver:name>
    <promoserver:machineAddressInUse>localhost</promoserver:machineAddressInUse>
</promoserver:channel>
```

Error Response - Channel not found

```
HTTP/1.1 404 Not Found
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element Test
could not be found</string>
```

Disallowed Actions

This end-point does not allow POST, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

20.7.4 GET All Machine Addresses under Channel

Returns a list of machine addresses for this Channel.

Syntax

```
GET /config/channels/{channel}/machineaddresses
```

Request

```
GET /config/channels/Test/machineaddresses HTTP/1.1
```

Response - List of machine addresses

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.promoserver+xml; type=machineAddressesList
<promoserver:machineAddressesList xmlns:promoserver="http://www.vizrt.com/
promoserver">
  <promoserver:machineAddress>
    <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=machineAddress" href="http://localhost:9980/config/channels/Test/
machineaddresses/localhost" />
    <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=machineAddress" href="http://localhost:9980/config/channels/Test/
machineaddresses/localhost" />
    <promoserver:refLink rel="status" type="application/vizrt.promoserver+xml;
type=machineAddressStatus" href="http://localhost:9980/config/channels/Test/
machineaddresses/localhost/status" />
    <promoserver:name>localhost</promoserver:name>
  </promoserver:machineAddress>
  <promoserver:machineAddress>
    <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=machineAddress" href="http://localhost:9980/config/channels/Test/
machineaddresses/backuphost" />
    <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=machineAddress" href="http://localhost:9980/config/channels/Test/
machineaddresses/backuphost" />
    <promoserver:refLink rel="status" type="application/vizrt.promoserver+xml;
type=machineAddressStatus" href="http://localhost:9980/config/channels/Test/
machineaddresses/backuphost/status" />
    <promoserver:name>backuphost</promoserver:name>
  </promoserver:machineAddress>
</promoserver:machineAddressesList>
```

Copyright © 2024 Vizrt

Error Response - Channel not found

```
HTTP/1.1 404 Not Found
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element Test
could not be found</string>
```

Disallowed Actions

This end-point does not allow PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

20.7.5 GET Machine Address

Returns a specific machine address for the channel.

Syntax

```
GET /config/channels/{channel}/machineaddresses/{address}
```

Request

```
GET /config/channels/Test/machineaddresses/localhost HTTP/1.1
```

Response - Machine Address returned

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.promoserver+xml; type=machineAddress
<promoserver:machineAddress xmlns:promoserver="http://www.vizrt.com/promoserver">
  <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=machineAddress" href="http://localhost:9980/config/channels/Test/
machineaddresses/localhost" />
  <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=machineAddress" href="http://localhost:9980/config/channels/Test/
machineaddresses/localhost" />
  <promoserver:refLink rel="status" type="application/vizrt.promoserver+xml;
type=machineAddressStatus" href="http://localhost:9980/config/channels/Test/
machineaddresses/localhost/status" />
  <promoserver:name>localhost</promoserver:name>
</promoserver:machineAddress>
```

Error Response - Channel not found

```
HTTP/1.1 404 Not Found
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element Test
could not be found</string>
```

Copyright © 2024 Vizrt

Error Response – Address not found

```
HTTP/1.1 404 Not Found
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element localhost
could not be found</string>
```

Disallowed Actions

This end-point does not allow POST or PUT, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

20.7.6 GET Status of Machine Address

Returns the status for a specific machine address for the channel.

The following status codes can be returned for the system:

- OK - Connection successful to Media Sequencer
- CHANNEL_NOT_FOUND - Connection successful, but the channel could not be found
- NO_VALID_DATA_FOUND - Connection successful, but the storage of Viz Channel Branding could not be found
- CONNECTION_REJECTED - Connection was rejected
- CONNECTION_TIMED_OUT - Connection timed out

Syntax

```
GET /config/channels/{channel}/machineaddresses/{address}/status
```

Request

```
GET /config/channels/Test/machineaddresses/localhost/status HTTP/1.1
```

Response - Status returned

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.promoserver+xml; type=machineAddressStatus
<promoserver:machineAddressStatus xmlns:promoserver="http://www.vizrt.com/
promoserver">
  <promoserver:status>OK</promoserver:status>
</promoserver:machineAddressStatus>
```

Error Response - Channel not found

```
HTTP/1.1 404 Not Found
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element Test
could not be found</string>
```

Copyright © 2024 Vizrt

Error Response – Address not found

```
HTTP/1.1 404 Not Found
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element localhost
could not be found</string>
```

Disallowed Actions

This end-point does not allow POST, PUT or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

20.7.7 GET Global Settings

Returns the global settings for Promo Server.

Syntax

```
GET /config/globalsettings
```

Request

```
GET /config/globalsettings HTTP/1.1
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.promoserver+xml; type=globalSettingsList
<promoserver:globalSettingsList xmlns:promoserver="http://www.vizrt.com/promoserver">
  <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=globalSettingsList" href="http://localhost:9980/config/globalsettings" />
  <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=globalSettingsList" href="http://localhost:9980/config/globalsettings" />
  <promoserver:cacheTime>2</promoserver:cacheTime>
  <promoserver:machineAddressesCacheTime>30</promoserver:machineAddressesCacheTime>
</promoserver:globalSettingsList>
```

Disallowed Actions

This end-point does not allow POST or DELETE, and will return a 'HTTP/1.1 405 Method Not Allowed'.

Copyright © 2024 Vizrt

20.7.8 POST Channel

Adds a channel to Promo Server configuration.

The response is a list of all channels that are configured.

Syntax

Syntax

```
POST /config/channels
```

Request

```
POST /config/channels HTTP/1.1
Content-Type: application/vizrt.promoserver+xml; type=channel
<promoserver:channel xmlns:promoserver="http://www.vizrt.com/promoserver">
  <promoserver:name>TestNew</promoserver:name>
</promoserver:channel>
```

Response – New channel was added to the configured list of channels

```
HTTP/1.1  200 OK
Content-Type: application/vizrt.promoserver+xml; type=channelsList
<promoserver:channelsList xmlns:promoserver="http://www.vizrt.com/promoserver">
    <promoserver:channelItem>
        <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/Test" />
        <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/Test" />
        <promoserver:refLink rel="machineaddresses" type="application/
vizrt.promoserver+xml; type=machineAddressesList" href="http://localhost:9980/config/
channels/Test/machineaddresses" />
        <promoserver:name>Test</promoserver:name>
    </promoserver:channelItem>
    <promoserver:channelItem>
        <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/TestCross" />
        <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/TestCross" />
        <promoserver:refLink rel="machineaddresses" type="application/
vizrt.promoserver+xml; type=machineAddressesList" href="http://localhost:9980/config/
channels/TestCross/machineaddresses" />
```

Copyright © 2024 Vizrt

```
        <promoserver:name>TestCross</promoserver:name>
    </promoserver:channelItem>
    <promoserver:channelItem>
        <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/TestNew" />
        <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/TestNew" />
        <promoserver:refLink rel="machineaddresses" type="application/
vizrt.promoserver+xml; type=machineAddressesList" href="http://localhost:9980/config/
channels/TestNew/machineaddresses" />
        <promoserver:name>TestNew</promoserver:name>
    </promoserver:channelItem>
</promoserver:channelsList>
```

Error Response – Channel already exists

```
HTTP/1.1  409 Conflict
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element of name
TestNew already exists</string>
```

Error Response – Invalid channel name

```
HTTP/1.1  400 Bad Request
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Input content is
invalid: Channel name Test#2 is not valid</string>
```

Copyright © 2024 Vizrt

20.7.9 DELETE Channel

Deletes a channel from Promo Server configuration.

Syntax

```
DELETE /config/channels/{channelName}
```

Request

```
DELETE /config/channels/Test HTTP/1.1
```

Response – Channel deleted

```
HTTP/1.1 204 No Content
```

Error Response – Channel does not exist

```
HTTP/1.1 404 Not Found
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element Test
could not be found</string>
```

Copyright © 2024 Vizrt

20.7.10 PUT Channel

Edit channel configuration.

⚠ Channel Name can't be changed

⚠ New machine address must exist in channel configuration

Syntax

```
PUT /config/channels/{channelName}
```

Request

```
PUT /config/channels/Test HTTP/1.1
Content-Type: application/vizrt.promoserver+xml; type=channel
<promoserver:channel xmlns:promoserver="http://www.vizrt.com/promoserver">
  <promoserver:name>Test</promoserver:name>
  <promoserver:machineAddressInUse>localhost</promoserver:machineAddressInUse>
</promoserver:channel>
```

Response – Machine address used for channel updated

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.promoserver+xml; type=channel<?xml version="1.0"
 encoding="utf-8"?>
<promoserver:channel xmlns:promoserver="http://www.vizrt.com/promoserver">
    <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/Test" />
    <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=channel" href="http://localhost:9980/config/channels/Test" />
    <promoserver:refLink rel="machineaddresses" type="application/
vizrt.promoserver+xml; type=machineAddressesList" href="http://localhost:9980/config/
channels/Test/machineaddresses" />
    <promoserver:name>Test</promoserver:name>
    <promoserver:machineAddressInUse>localhost</promoserver:machineAddressInUse>
</promoserver:channel>
```

Copyright © 2024 Vizrt

Error Response – Machine address does not exists

```
HTTP/1.1 400 Bad Request
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Machine address
localhost is not configured for channel</string>
```

Error Response – Channel not found

```
HTTP/1.1 404 Not Found
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element Test
could not be found</string>
```

Error Response – Channel name can not be updated

```
HTTP/1.1 400 Bad Request
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Input content can'
t be used to update channel: Channel Name can't be changed</string>
```

Copyright © 2024 Vizrt

20.7.11 POST Machine Address

Adds a new machine address to a channel Promo Server configuration.

> **Note**: Machine address names are not case sensitive, i.e. localhost is the same as LOCALHOST

Syntax

```
POST /config/channels/{channelname}/machineaddresses
```

Request

```
POST /config/channels/Test/machineaddresses HTTP/1.1
Content-Type: application/vizrt.promoserver+xml; type=machineAddress
<promoserver:machineAddress xmlns:promoserver="http://www.vizrt.com/promoserver">
  <promoserver:name>secondarybackuphost</promoserver:name>
</promoserver:machineAddress>
```

Response – New machine address was added to the configured list of machine addresses for a channel

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.promoserver+xml; type=machineAddressesList
<promoserver:machineAddressesList xmlns:promoserver="http://www.vizrt.com/
promoserver">
  <promoserver:machineAddress>
    <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=machineAddress" href="http://localhost:9980/config/channels/Test/
machineaddresses/localhost" />
    <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=machineAddress" href="http://localhost:9980/config/channels/Test/
machineaddresses/localhost" />
    <promoserver:refLink rel="status" type="application/vizrt.promoserver+xml;
type=machineAddressStatus" href="http://localhost:9980/config/channels/Test/
machineaddresses/localhost/status" />
    <promoserver:name>localhost</promoserver:name>
  </promoserver:machineAddress>
  <promoserver:machineAddress>
    <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=machineAddress" href="http://localhost:9980/config/channels/Test/
machineaddresses/backuphost" />
```

Copyright © 2024 Vizrt

```
    <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=machineAddress" href="http://localhost:9980/config/channels/Test/
machineaddresses/backuphost" />
    <promoserver:refLink rel="status" type="application/vizrt.promoserver+xml;
type=machineAddressStatus" href="http://localhost:9980/config/channels/Test/
machineaddresses/backuphost/status" />
    <promoserver:name>backuphost</promoserver:name>
  </promoserver:machineAddress>
  <promoserver:machineAddress>
    <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=machineAddress" href="http://localhost:9980/config/channels/Test/
machineaddresses/secondarybackuphost" />
    <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=machineAddress" href="http://localhost:9980/config/channels/Test/
machineaddresses/secondarybackuphost" />
    <promoserver:refLink rel="status" type="application/vizrt.promoserver+xml;
type=machineAddressStatus" href="http://localhost:9980/config/channels/Test/
machineaddresses/secondarybackuphost/status" />
    <promoserver:name>secondarybackuphost</promoserver:name>
  </promoserver:machineAddress>
</promoserver:machineAddressesList>
```

Error Response – Machine address already exists

```
HTTP/1.1 409 Conflict
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element of name
localhost already exists</string>
```

Error Response – Channel not found

```
HTTP/1.1 404 Not Found
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element Test
could not be found</string>
```

Error Response – Invalid host name

```
HTTP/1.1 400 Bad Request
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Input content is
invalid: Machine address 'host#' is not a valid host name</string>
```

Copyright © 2024 Vizrt

20.7.12 DELETE Machine Address

Deletes a machine address under a specific channel.

Syntax

```
DELETE /config/channels/{channelName}/machineaddresses/{machineAddress}
```

Request

```
DELETE /config/channels/Test/machineaddresses/127.0.0.1 HTTP/1.1
```

Response – Machine Address deleted

```
HTTP/1.1 204 No Content
```

Error Response – Channel does not exist

```
HTTP/1.1 404 Not Found
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element Test
could not be found</string>
```

Error Response – Machine Address does not exist

```
HTTP/1.1 404 Not Found
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Element 127.0.0.1
 could not be found</string>
```

Copyright © 2024 Vizrt

20.7.13 PUT Global Settings

Edits global settings.

⚠ Cache Time must be an integer value higher then 0 and cannot be set null

Syntax

```
PUT /config/globalsettings
```

Request

```
PUT /config/globalsettings HTTP/1.1
Content-Type: application/vizrt.promoserver+xml; type=globalSettingsList
<promoserver:globalSettingsList xmlns:promoserver="http://www.vizrt.com/promoserver">
  <promoserver:cacheTime>5</promoserver:cacheTime>
  <promoserver:machineAddressesCacheTime>60</promoserver:machineAddressesCacheTime>
</promoserver:globalSettingsList>
```

Response – Successfully edited Global Settings

```
HTTP/1.1 200 OK
Content-Type: application/vizrt.promoserver+xml; type=globalSettingsList
<promoserver:globalSettingsList xmlns:promoserver="http://www.vizrt.com/promoserver">
  <promoserver:refLink rel="self" type="application/vizrt.promoserver+xml;
type=globalSettingsList" href="http://localhost:9980/config/globalsettings" />
  <promoserver:refLink rel="edit" type="application/vizrt.promoserver+xml;
type=globalSettingsList" href="http://localhost:9980/config/globalsettings" />
  <promoserver:cacheTime>5</promoserver:cacheTime>
  <promoserver:machineAddressesCacheTime>60</promoserver:machineAddressesCacheTime>
</promoserver:globalSettingsList>
```

Error Response – Invalid cache time

```
HTTP/1.1 400 Bad Request
Content-Type: application/xml; charset=utf-8
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Input content
can't be used to update global settings: Cache time for machine addresses must have
an integer value</string>
```

Copyright © 2024 Vizrt

20.8 Logging

- Daily Log (As Run)
- Log File Locations

20.8.1 Daily Log (As Run)

For information and troubleshooting purposes, a daily log file provides details on every event that was played. The file is located on the **Media Sequencer** machine at: `%ProgramData%\Vizrt\Media Sequencer`. Since the file is not set up as part of the installation process, it must be created as part of the installation process described in the Playout section.

The logging file naming convention is `ChannelName-YY-MM-DD.log`. Each file includes the events played on the calendar day specified, which may not necessarily correspond to a playlist. As such, the events in a playlist that starts at 7 AM and extends till the following day at 6 AM will be saved in two separate files - one for each calendar day.

An example of a log file's contents is shown below:

```
2007-12-18 12:06:36: Running: read on /VizRundown/channels/AD/rundowns/21_11_2005_AD/
data/PLAYLIST_DATA_ROMY_14_24_40_681_13_12_2007_823_IM_0
2007-12-18 12:06:41: Running: read on /VizRundown/channels/AD/rundowns/21_11_2005_AD/
data/PLAYLIST_DATA_ROMY_14_24_40_681_13_12_2007_823_IM_0
2007-12-18 12:07:15: Running: read on /VizRundown/channels/AD/rundowns/21_11_2005_AD/
data/PLAYLIST_DATA_ROMY_12_06_58_663_18_12_2007_22_PA_0
2007-12-18 12:08:28: Running: read on /VizRundown/channels/AD/rundowns/21_11_2005_AD/
data/PLAYLIST_DATA_ROMY_14_24_40_681_13_12_2007_823_IM_0
2007-12-18 12:09:34: Running: read on /VizRundown/channels/AD/rundowns/21_11_2005_AD/
data/PLAYLIST_DATA_ROMY_14_24_40_681_13_12_2007_823_IM_0
2007-12-18 12:09:39: Running: read on /VizRundown/channels/AD/rundowns/21_11_2005_AD/
data/PLAYLIST_DATA_ROMY_14_24_40_681_13_12_2007_823_IM_0
2007-12-18 12:09:46: Running: read on /VizRundown/channels/AD/rundowns/21_11_2005_AD/
data/PLAYLIST_DATA_ROMY_12_06_58_663_18_12_2007_22_PA_0
2007-12-18 12:16:50: Running: read on /VizRundown/channels/AD/rundowns/21_11_2005_AD/
data/PLAYLIST_DATA_ROMY_14_24_40_759_13_12_2007_715_IM_2
2007-12-18 12:16:53: Running: read on /VizRundown/channels/AD/rundowns/21_11_2005_AD/
data/PLAYLIST_DATA_ROMY_12_16_29_344_18_12_2007_319_DR_0
```

20.8.2 Log File Locations

As explained above, all log files in the list below are relative to the log file base directory of *C:\ProgramData\Vizrt\Viz Channel Branding*

- Viz Channel Branding Client: *.\Client*
- Playlist Importer : *.\Playlist Importer*
- Upgrade Tool: *.\Upgrade Tool*

Copyright © 2024 Vizrt

- Harris Integration Module: *.\Harris Integration*
- Pebble Beach Marina Integration Module: *.\Marina Integration*
- Pebble Beach Neptune Integration Module: *.\Neptune Integration*
- Promo Server: *No log-file on disk, use Windows Event Viewer*
- Integrations Hub: *No log-file on disk, use Windows Event Viewer*

Warning: Care should be taken to set up automatic tasks or manual routines to prune or delete old log files that are no longer needed, otherwise the log files could eventually fill the disk.

Copyright © 2024 Vizrt

20.9 Glossary

20.9.1 Viz Channel Branding Terminology

A number of glossary items in the table below refer to one another. If a definition contains an unfamiliar term, check for its explanation in the same table.

Term	Definition
Action	An action can be a GPO command, Viz Engine command or a Viz Ticker command. See Actions.
Automation system	The system that manages the assemblage of broadcast systems, meaning the camera, graphics, prompters, etc. for broadcast. During playout, the system sends control commands to Viz Channel Branding.
Broadcast day playlist	Viz Channel Branding supports two types of playlists; **broadcast day** and **continuous playlists**. A broadcast day playlist must be switched based on day switch settings.
Channel	A channel is a defined broadcast channel. A channel holds all the information needed for playout of that channel's program content. This involves how to import and map playout schedule(s) to a single continuous playlist or multiple broadcast day playlists.
Control Room	Room serving as an operations center where a facility or service can be monitored and controlled. A control room generally serves a single show at a time.
Continuous playlist	Viz Channel Branding supports two types of playlists: **continuous** and **broadcast day playlists**. A continuous playlist does not have to be switched based on day switch settings.
Device channel	A device channel is basically a list of playout devices assigned to a Channel. This can be one or multiple Viz Engines for graphics playout and video devices for full screen video playout.
DVE	Digital Video Effects. Video that is played through the Viz Engine and that can have 2D animation.
Event	A playlist element. Can be a primary or secondary event.

Copyright © 2024 Vizrt

Term	Definition
Event ID	For primary events, this field serves as the index on which the playlist updates are done. This identifies the primary event, and is the key to knowing which event is to be updated.
Master Control Room	The Master Control Room (MCR) is the technical hub of a broadcast operation, commonly found among most over-the-air television stations and networks. It is distinct from production control rooms in television studios where activities such as switching from camera to camera are coordinated.
Media Sequencer	Media Sequencer (MSE) is a central software component used by Vizrt client applications. The Media Sequencer is a framework for defining and executing media elements.
Offset	The delay between a primary event starting and its held secondary events starting.
Page	A page is an instance of a template containing data (such as imagery, text, video), and is based on a graphics template. Once added to a playlist, pages are considered secondary events.
Playout device	A playout device is either a Viz Engine dedicated to graphics and video clip playout, or a video device dedicated to full screen video playout, among other things (see device channel).
Playlist	A list of programs, typically for a single day. An external playlist can be imported from an automation or scheduling system. A playlist is composed of primary and secondary events.
Primary Event	A program in a playlist; appears as the parent element. Pages and Viz Commands can be added as secondary events.
Rule	For imported (and subsequently updated) playlists, you can create sets of rules that govern how templates can be automatically associated with certain events within a playlist. Rules are useful when external playlists do not already contain secondary events. Rules are individually defined for each template. Both master and editable templates support rules. Rules can also be applied to Viz Commands (see Working with Actions). Rules are applied when importing a schedule or updating a playlist of external origin.
Scene	A graphic scene that has been built in Viz Artist.

Copyright © 2024 Vizrt

Term	Definition
Scheduling department	See traffic department.
Scheduling system	The system used by the traffic department (scheduling department) for scheduling primary events - advertisements in particular - of program material for the broadcast day. The result of this scheduling is a daily schedule. Some scheduling systems also include information for secondary events. Note that a scheduling system is also known as a traffic system.
Secondary Event	A child element of a primary event in a playlist. Can be a Page or a Viz Command.
Template	A graphic scene that has been built in Viz Artist, after it has been imported into Viz Channel Branding. A template is used to create pages that can be added to the playlist. Once added to a, pages are considered secondary events.
Traffic department	The department where the initial insertion of primary and secondary events into the schedule takes place. Once complete, the schedule is transferred to the automation system and Viz Channel Branding in the Master Control Room for the actual playout. Note that the traffic department is also called the scheduling department.
Trigger	A control command initiated from the automation system that tells Viz Channel Branding to execute a certain command on a playlist event, such as LOAD, PLAY or STOP. The automation system uses a known protocol, such as VDCP, CII, GPI etc. to send the command.
Viz Engine	A combination of hardware and software used to render graphics and to play out media such as video.

20.9.2 Abbreviations

The abbreviations below are used in this guide:

Abbreviation	Definition
ADC	Automated Content Management and Distribution
PB	Pebble Beach
PCF	Page Content Filling

Copyright © 2024 Vizrt

Abbreviation	Definition
PLHM	Playlist History Manager
PLI	Playlist Importer
SCC	Schedule Collector
QS	Quick Scheduler
PF	Promo Factory

Copyright © 2024 Vizrt