# Viz Multichannel

Version 4.1

## Disclaimer

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

## Technical Support

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

## Created on

2019/05/26

# Contents

# 1 Introduction

This is the User Guide for Viz Multichannel.

- Related Documents
- Feedback and Suggestions

## 1.1 Related Documents

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

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

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

## 1.2 Feedback And Suggestions

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

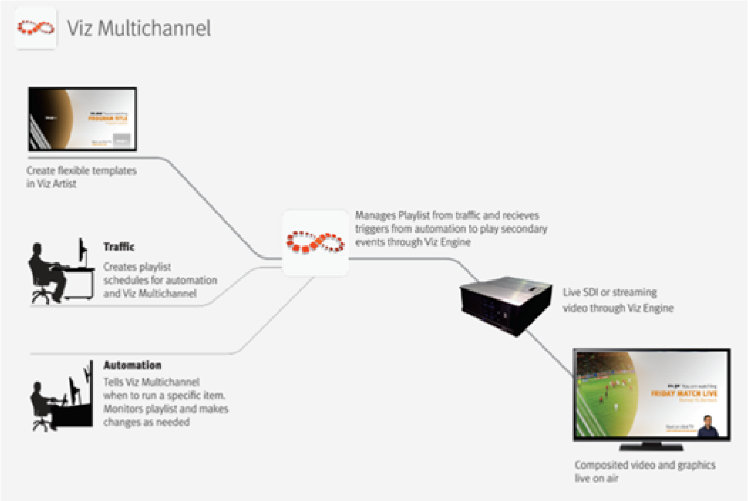
# 2 Overview

Viz Multichannel (VMC) provides high-quality Vizrt graphics for channel branding. The application simplifies the playout of playlists, potentially as a fully automated workflow that integrates with existing channel traffic and scheduling solutions.

Viz Multichannel combines pre-scheduled playlist operations with the ability to add real-time 2D and 3D graphic effects. In complex environments, Viz Multichannel helps with centralizing entire workflows to one or a few desktops.

- VMC-Introduction
- All-in-One Workflow
- Graphics Workflow
- Viz Multichannel Terminology

## 2.1 VMC–Introduction


Viz Multichannel
Create flexible templates in Viz Artist
Traffic
Creates playlist schedules for automation and Viz Multichannel
Automation
Tells Viz Multichannel when to run a specific item. Monitors playlist and makes changes as needed
Manages Playlist from traffic and recieves triggers from automation to play secondary events through Viz Engine
Live SDI or streaming video through Viz Engine
FRIDAY MATCH LIVE
Composited video and graphics live on air

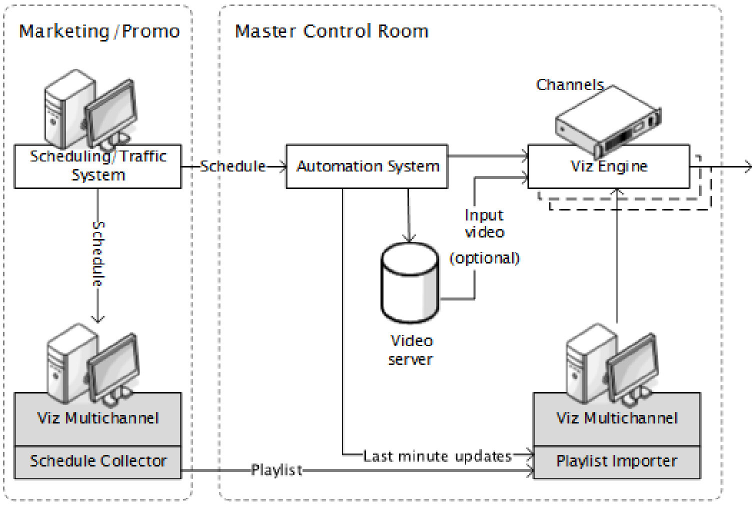
For seamless integration with existing hardware, the solution employs standard rack mountable PCs. Updates and upgrades are easily applied without requiring modifications to the solution as a whole.

Some key benefits:

- Translation of 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 Multichannel terminal
- Automatically adapt changes in the playlist, up to the very last minute

## 2.2 All-In-One Workflow



Viz Multichannel 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. 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 Multichannel playlist, including a graphic layer, is automatically created. If the playlist is *continuous*, the playlist will be updated to reflect the new data received.

*Pages* are graphic elements ready for playout that can be manually or automatically added to the Multichannel playlist. Pages are created from *Templates* that a graphic artist creates in Viz Artist. The templates (and therefore, pages created from templates) can have fields or elements that Multichannel 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?”.

Adding a predefined *page* to an event in the playlist can either be achieved by dragging the page from the Templates and Pages list and dropping it onto a primary event in the playlist or by allowing the graphic elements to be created automatically, based on predefined Rules. The latter is especially convenient for non-technical personnel in traffic departments who can prepare pages for playout without actually having to work with them. 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.

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

1. The first layer is Viz Multichannel’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 Multichannel 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 Multichannel also provides an interface mechanism that allows broadcast systems to exchange mission-critical data.

Last minute changes to existing playlists are common in everyday broadcasting: sports events go overtime, live shows do not keep their planned time schedule, and breaking news events demand their own space in the playlist.

Viz Multichannel 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.

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

Multichannel works with relative time codes. Offsets for graphics are given, based on the timecode of a specific event rather than an absolute time of day. Updates in the playlist do not affect the secondary events (e.g. graphics, video clips or a combination of the two) in the playlist, simply because the relative start point of such an event in relation to the primary event always stays the same.

Viz Multichannel integrates with Viz One that enables the operator to search and browse for Clips stored in the Viz One from within the Viz Multichannel 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.

Finally, the playlist can be populated with graphics effects from all sources that follow the ODBC standard.

Viz Multichannel allows the promotion of programs and shows across multiple channels. This is made possible by accessing the playlist of other channels and cross referencing the content with the primary channel.

Viz Multichannel can also make use of Viz Engine for local preview of graphics and graphics with embedded video clips. The preview functionality is an essential tool for operators in the master control room to monitor graphics before taking it to air.

## 2.3 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.

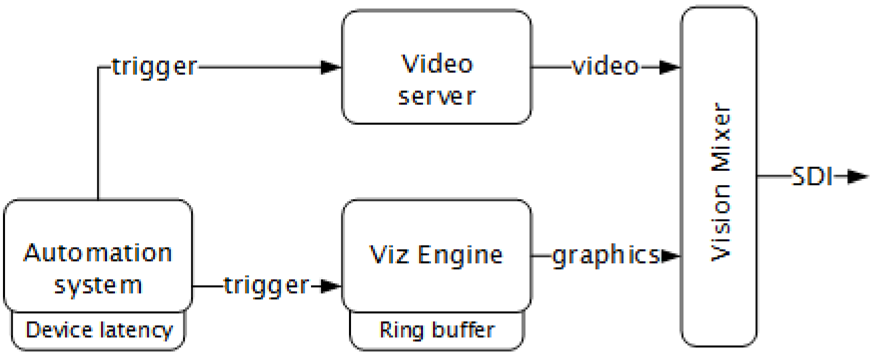
Opposite to upstream mixing you have **downstream** mixing. Downstream mixing means that graphics and video are composited and mixed by Viz Engine. This can be live video or clips treated as DVE for graphics insertion and/or squeeze backs.

⚠ **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.

### 2.3.1 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.

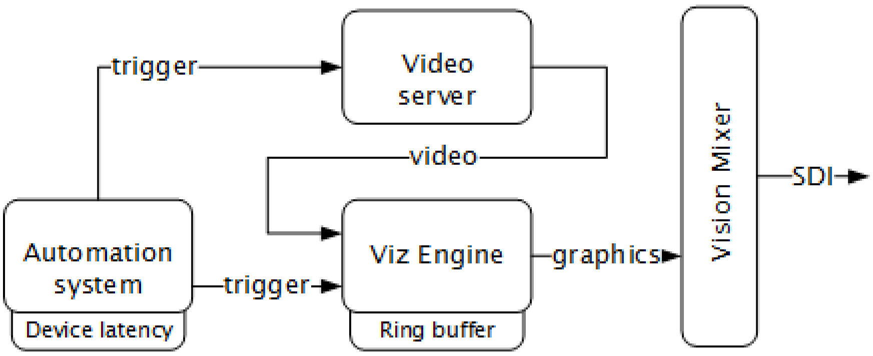


Therefore most automation systems, including Media Sequencer, are able to adjust and cater for latency introduced by devices such as Viz Engine due to the buffer it needs to render graphics in real-time.

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.

### 2.3.2 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.

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.

## 2.4 Viz Multichannel 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 Multichannel. |
| Broadcast day playlist | Viz Multichannel supports two types of playlists; **broadcast day** and **continuous playlists**. A broadcast day playlist must be switched based on day switch settings. |

| Term | Definition |
|---|---|
| Channel | A channel is a defined broadcast channel. In Viz Multichannel 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. It also includes how Viz Multichannel is configured in order to cater for the channel’s setup requirements (see device channel). |
| Control Room | Room serving as an operations centre where a facility or service can be monitored and controlled. A control room generally serves a single channel. |
| Condition | Applied on rules, a condition relates to a primary events. When defining a condition, one or more attributes must be selected: the user enters a value which the primary event attribute must meet. If the condition is met, then the associated template will be shown. |
| Continuous playlist | Viz Multichannel 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 (see playout device). |
| 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. |
| 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 know which event is to be updated. |
| Item | An item (or element) in a playlist. Can be a primary or secondary event. |
| Master Control Room | The Master control room (MCR) is the technical hub of a broadcast operation common among most over-the-air television stations and networks. It is distinct from production control rooms in television studios where the activities such as switching from camera to camera are coordinated.Typically there will be Multichannel client for last-minute changes and monitoring. |

| Term | Definition |
|---|---|
| Media Sequencer | Media Sequencer 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 the secondary events and its respective primary event. |
| Page | A page is an instance of a template containing data (e.g. imagery, text, video), and is based on a graphics template. Pages, when added to a Viz Multichannel playlist, 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 and so on (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 Viz Multichannel playlist is composed of primary and secondary events, and can be viewed in the right pane. |
| Playlist Element | A primary or secondary event in a playlist. |
| Primary Event | A program in a playlist. Appears as the parent element. Templates 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 defined on a per-template basis. Both master and editable templates support rules. In addition, 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 built in Viz Artist, before being imported into Viz Multichannel. |
| Scheduling department | See traffic department. |

| Term | Definition |
|---|---|
| Scheduling system | The system used by the traffic department for scheduling primary events of program material, and in particular advertisements, for the broadcast day. The result of this scheduling is a daily schedule which is the input for Viz Multichannel.Depending on the system, it can 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 template or a Viz Command. |
| Template | A graphic scene built in Viz Artist, after being imported into Viz Multichannel. A template is used to create pages (an instance of the template) that are added to the playlist. Pages, when added to a Viz Multichannel playlist, are considered secondary events. |
| Traffic department | The department where the initial insertion of primary and secondary events into the schedule takes place. When ready, it will be transferred to the automation system and Viz Multichannel in the Master Control Room for the actual playout. |
| Trigger | A control command initiated from the automation system telling Viz Multichannel to execute a certain command on a playlist event, such as LOAD, PLAY, 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. |

# 3 Requirements

There are compatibility requirements, access policies and rights for the software components in a Viz Multichannel 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.

## 3.1 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 in 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 an USB or a parallel port dongle on the renderer machine is required.

## 3.2 Hardware

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

For older hardware that are used with newer versions of Vizrt software, such as Viz Engine, it is always recommended to check against the current hardware specifications for the new software version to make sure that the latest software can run on the old hardware specification.

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 in the Media Sequencer servers.

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

## 3.3 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 (RN) for late minute changes or requirements, if any. If there are differences in the hardware or software requirements listed here and in the RN then the Release Notes supersedes the manual.

## 3.4 Viz Multichannel

The Viz Multichannel 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<br>Microsoft .NET Framework 4.6.1 or higher.<br>Microsoft Visual C++ 2010 Redistributable Package (free download from http://www.microsoft.com/en\-us/download/details.aspx?id=5555)<br>Viz Engine 3.6 or higher. Viz Engine 3.8.3 recommended.<br>If you are using a local Viz Engine to preview video clips from Viz One, Viz Engine must be installed with video codecs: (Windows XP = MatroskaSplitter, ffdshow, Win7DSFilterTweaker // Windows 7 = K-lite Codec Pack full)<br>Viz One 5.11 or higher if using Viz One integration. |

| Component | Requirement |
|---|---|
| Executable(s) | VizRundown.exe<br>PlaylistImporter.exe Version 1<br>PlaylistImporter.exe Version 2<br>viz.exe (Viz Engine executable)<br>PromoServer.exe<br>UpgradeTool.exe<br>HarrisIntegration.exe<br>MarinaIntegration.exe<br>NeptuneIntegration.exe |
| Service(s) | Viz Multichannel Schedule Collector<br>Viz Multichannel Promo Server<br>Viz Multichannel Integrations Hub |
| Operating system | Windows 10 32-bit or 64-bit<br>Windows 7 32-bit or 64-bit<br>Windows Server 2008 R2 |

## 3.5 Integration With Viz One

Multichannel integrates with the Viz One MAM system. Viz One version 5.8 (and later versions) require Multichannel version 2.11 or later.

## 3.6 Network Ports And Connections Ports And Connections

| Application (From) | Application (To) | Port |
|---|---|---|
| Viz Multichannel (VizRundown.exe) | Media Sequencer | 8580, 8594 |
| Viz Multichannel | Viz Engine | 6100, 6800 |
| Viz Multichannel | Viz One | 80 |
| Viz Multichannel | Promo Server | 9005 |
| Viz Multichannel | Preview Server | 5400, 21098 |

| Application (From) | Application (To) | Port |
|---|---|---|
| PlayList Importer v1 | Media Sequencer | 8594 |
| Integration Modules, Quick Scheduler Server | Integrations Hub | 9990 |
| PlayList Importer v2 | Integrations Hub | 9991 |
| Quick Scheduler | Quick Scheduler Server | 9991 |
| Harris Integration | Harris Automation | 10050 |
| Marina Integration | Marina Automation | 18083 |

Ports in table above are TCP. Read the *From* and *To* in the table as initiator and receiver. Example: Multichannel uses TCP-port 8580 and 8594 when exchanging data with the Media 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 Multichannel Marina Integration sends traffic on port 18083 to the Marina Automation and uses port 12212 bi-directional.

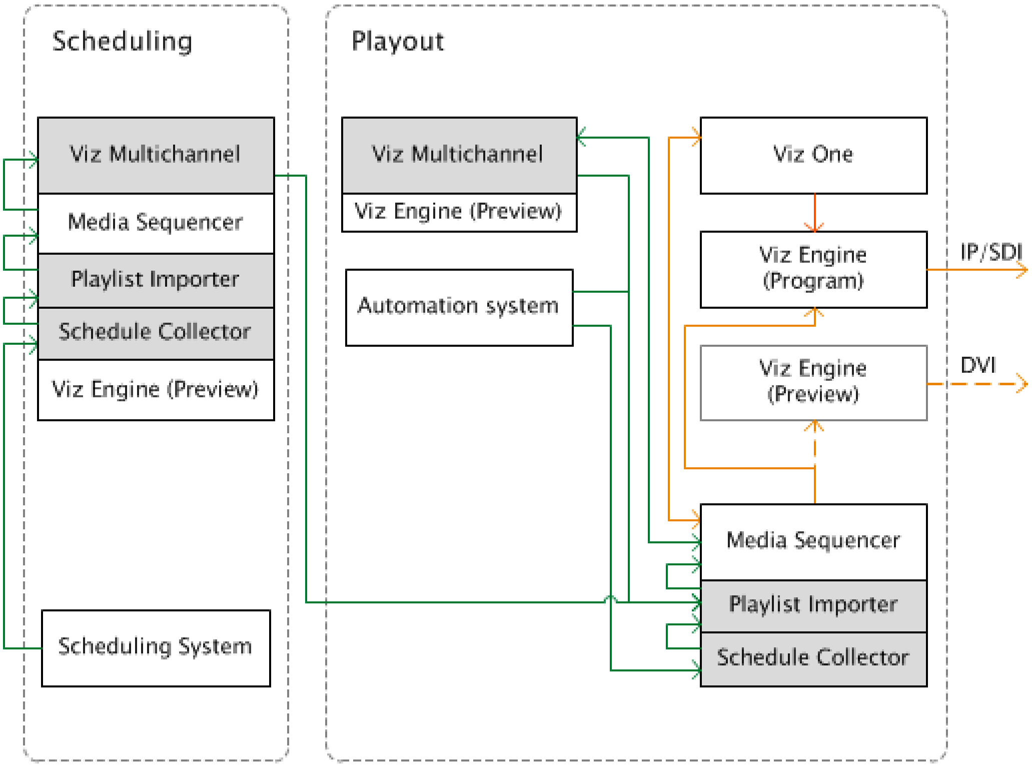
⚠ **IMPORTANT!** Some ports are configurable. For instance the Viz Engine uses port 6100 (and additional port 6800 for dual-channel engines) but can be configured to use other ports.

# 4 Setup And Configuration

When setting up Viz Multichannel you will install other software components except those parts bundled with Viz Multichannel. Hence, this manual will refer to those parts, but for actual help on installation it will refer you to the respective manuals.

When planning the setup and configuration there are in particular two parts that should be carefully considered before you install and configure Multichannel:

- Many Viz Multichannel systems are connected to an automation system that automatically provides the schedule from a scheduling department. Other installations read the schedule from a text file, a database, XML, Excel or by other means. Multichannel refers to the received schedule (i.e. what to do and when to do it) as a playlist. At any given time, Multichannel must have a valid, active and running playlist for the planned automated actions to be performed. How will the schedule be provided and how will it be ingested?
- 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. This is called externally triggered events. In some cases the automation system will only trigger main events (example: a film) and then graphics will be applied by a time offset from the main event.



Typically Viz Multichannel is installed in the following locations:

- Viz Multichannel's Schedule Collector and Playlist Importer on the Media Sequencer machine in the **server room**. Media Sequencer is in most cases installed on the Viz Engine (program) machine.

- Viz Multichannel, it’s Schedule Collector and Playlist Importer on a client machine in the **scheduling center**
- Viz Multichannel on a client machine in the **master control room**

This section contains information on the following topics:

- Bundle Installer
- Scheduling
- Playout
- Running Viz Multichannel

## 4.1 Bundle Installer

The bundle installer is a convenient “wrapped” executable that can be used to install one or all of the separate components (programs) that are part of Viz Multichannel. It is typically distributed as:

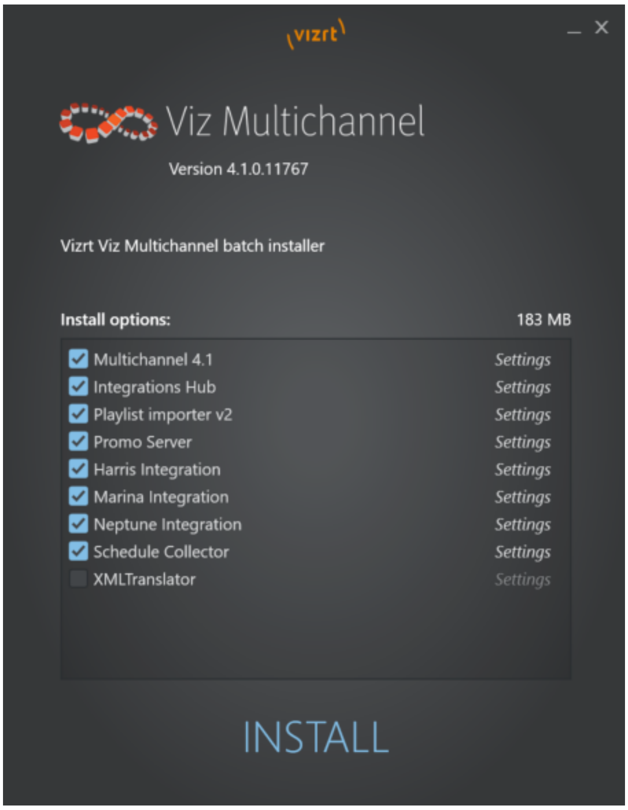
```
MultiChannelBundleInstaller-<VERSION-NUMBERS>.exe
```

Start the installer and you can select the components you require for install, repair or deletion. The installer will also do the required basic configuration and starting service processes. The bundle (wrapped) installer is the most convenient installation method and as such recommended. See below if you require the separate MSI components.

⚠ **IMPORTANT!** Before you can install the **Integrations Hub** component, either with the Bundle installer or using the separate MSI installer, the server must have the requirements installed and configured before starting the installation process. If not, the installation process will fail with an error message for the Integrations Hub component, the other components will install correctly. For detailed instructions, see Installation, Requirements and Configuration check in the Integrations Hub chapter.

If you need to install from separate MSI installers, see Unbundling the Wrapped Installer below.

Start the bundle installer by double-clicking the installer and click install after selecting the components:



Viz Multichannel

Version 4.1.0.11767

Vizrt Viz Multichannel batch installer

Install options: 183 MB

| | |
|---|---|
| ☑ Multichannel 4.1 | Settings |
| ☑ Integrations Hub | Settings |
| ☑ Playlist importer v2 | Settings |
| ☑ Promo Server | Settings |
| ☑ Harris Integration | Settings |
| ☑ Marina Integration | Settings |
| ☑ Neptune Integration | Settings |
| ☑ Schedule Collector | Settings |
| □ XMLTranslator | Settings |

### 4.1.1 Unbundling the Wrapped Installer

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

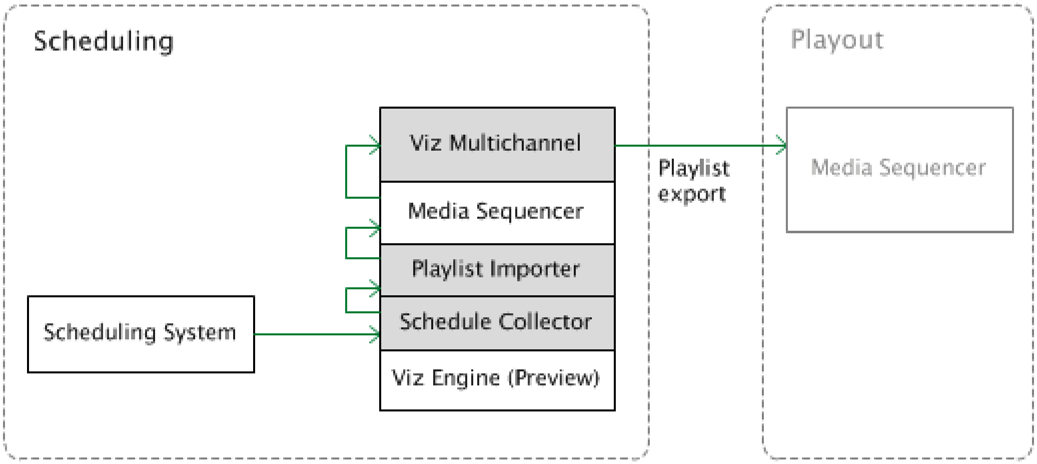
```
MultiChannelBundleInstaller-<VERSION-NUMBER>.exe /msi
```

This will create a new directory named after the installer containing 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.

## 4.2 Scheduling



A typical installation includes one Viz Multichannel and a local Viz Engine. In addition you typically install Viz Multichannel's Schedule Collector (SCC) and Playlist Importer (PLI) in order to collect and import playlists through mapping settings into Viz Multichannel.

Viz Engine is used for preview and is typically a VGA version without SDI out capabilities. This setup allows the Scheduling Center to prepare and preview playlists before sending them to playout by the program Viz Engine(s).

The following software components are in use:

- Viz Multichannel client
- Viz Multichannel's Schedule Collector (SCC)
- Viz Multichannel's Playlist Importer (PLI)
- Viz Engine (local preview)
- Media Sequencer

### 4.2.1 Installation for Scheduling

The machine in the Scheduling must have the following software components installed:

- Viz Multichannel client
- Viz Multichannel's Schedule Collector
- Viz Multichannel's Playlist Importer
- Media Sequencer
- Viz Engine (preview)

> ⚠ **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.

#### To Install Viz Multichannel Client

1. Start the Viz Multichannel installer
2. Click **Next**

3. Select **Complete** and click **Next**
   Select **Custom** if a different *install path* is needed.
4. Click **Install**
5. Click **Finish**

## To Install Media Sequencer

1. Start the Media Sequencer installer
2. Click **Next**
3. Click **Next**
   Select *features* and/or a different *install path* if needed.
4. Click **Install**
   Put *shortcuts* on the desktop if needed.
5. Click **Finish**

## To Install Viz Engine

Viz Engine is used for local preview, hence, you only need to install a **VGA** version.

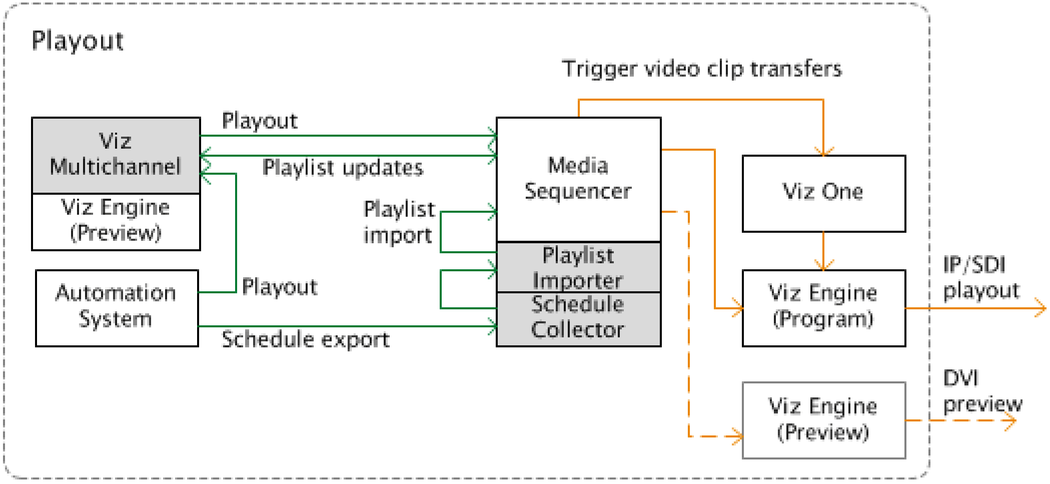
1. Start the Viz Artist/Viz Engine installer
2. Click **Next**
3. Check the *I accept the terms in the License Agreement* check box and click **Next**
4. Click **VGA**
5. Click **Next**
   Select *features* and/or a different *install path* if needed.
6. Click **Install**
7. Click **Finish**

**See Also**

- Playout
- Vizrt’s Documentation Center

## 4.3 Playout

### 4.3.1 Playlist Importer v1



⚠ The schematic above refers to Playlist Importer v1. As of Viz Multichannel 4.1, Playlist Importer v2 is introduced. See Playlist Importer v2 below.

### 4.3.2 Playlist Importer v2

Viz Multichannel 4.1, introduces Playlist Importer (PLI) v2. PLI v2 first pushes data to the Integrations Hub v2, that in turns pushes it to Media Sequence Engine (MSE). In the illustration above, playlists from Automation always go via SCC. With PLI v2, this is only sometimes true, as most communication is through Integrations Hub v2.

### 4.3.3 Component Placement

A second Viz Multichannel application is typically installed in the master control room (MCR) for monitoring purposes and manual update of data. It can also be used for manual playout. This installation also includes a local Viz Engine for preview purposes.

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

The machine in the MCR must have the following software components installed:

- Viz Multichannel client
- Viz Engine (local preview)
  Viz Multichannel’s Schedule Collector and importer must be installed alongside the Media Sequencer. Note that you may install the Media Sequencer, Schedule Collector and Playlist Importer on the Viz Engine program machine.

The Schedule Collector collects and outputs scheduled events, typically from the automation system, which the Playlist Importer imports into the Media Sequencer for the Viz Multichannel operator in the MCR to see. Final mapping of events into the Vizrt playlist is done using the Viz Multichannel client.

Video clip transfers from Viz One to the Viz Engine program machine are initiated by the Media Sequencer once the playlist is activated.

The following software components are in use:

- Viz Multichannel's Schedule Collector (SCC)
- Viz Multichannel's Playlist Importer (PLI)
- Viz Engine (program and preview)
- Media Sequencer

> ⚠ **Note:** Collection of events from scheduling systems are currently not supported by the Schedule Collector. This is done using the Viz Multichannel client in *scheduling mode*.

This section presents:

- Installation of Viz Multichannel
- Installation of Viz Engine Machines for Playout
- Configuration of Channels with channels.ini
- Enabling Logging

**See Also**

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

## 4.3.4 Installation of Viz Multichannel

Viz Multichannel in the master control room must be connected to Viz Engines for program and preview output; however, a Viz Engine for local preview is often needed as well. Ensure that you have Viz Engines ready before starting the installation steps below.

A Viz Engine is also used for local preview by the Page Editor, hence, you need to install a **VGA** version.

> ⚠ **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

### To Install the Viz Multichannel client

1. Start the Viz Multichannel installer
2. Click **Next**
3. Select **Complete** and click **Next**
   Select **Custom** if a different *install path* is needed.
4. Click **Install**
5. Click **Finish**

### To Install Viz Engine

1. Start the Viz Artist/Viz Engine installer
2. Click **Next**
3. Check the *I accept the terms in the License Agreement* check box and click **Next**
4. Click **VGA**
5. Click **Next**
6. Select *features* and/or a different *install path* if required.
7. Click **Install**
8. Click **Finish**

## 4.3.5 Installation of Viz Engine Machines for Playout

It is recommended to install Viz Multichannel’s Schedule Collector, Playlist Importer, 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.

### To Install the Schedule Collector

1. Start the Viz Multichannel Schedule Collector installer
2. Click **Next**
3. Click **Next**
   Select a different *install path* if you need to.
4. Click **Install**
5. Click **Finish**

### To Install the Playlist Importer

1. Start the Viz Playlist Importer installer
2. Click **Next**
3. Click **Next**
   Select a different *install path* if you need to.
4. Click **Install**
5. Click **Finish**

### To Install Media Sequencer

1. Start the Media Sequencer installer
2. Click **Next**
3. Click **Next**
   Select *features* and/or a different *install path* if needed.
4. Click **Install**
   Put *shortcuts* on the desktop if needed.
5. Click **Finish**

### To Install Viz Engine (program and preview)

Viz Engines used as your program and preview output channels can easily switch roles, hence, you should always have two **Video** versions installed.

1. Start the Viz Artist/Viz Engine installer
2. Click **Next**
3. Check the *I accept the terms in the License Agreement* check box and click **Next**
4. Click **Video**
5. Click **Next**
   Select *features* and/or a different *install path* if needed.
6. Click **Install**
7. Click **Finish**

**See Also**

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

## 4.3.6 Configuration of Channels with channels.ini

For the Master Control Room machine(s) using the MCR-mode of Multichannel 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 Multichannel GUI in MCR-mode will abort startup with an error message.

### To Configure channels.ini

1. In Windows Explorer, locate the Viz Multichannel's directory where the *channels.ini* is:
   - For Windows XP this would normally be:
   - `C:\Documents and Settings\All Users\Application Data\Vizrt\Multichannel\channels.ini`
   - For Windows 7 or Windows Server 2008 R2 and higher (64-bit Windows) this would normally be:
   - `%ProgramData%\vizrt\Multichannel\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 textfile so make sure you save the file as type text.
5. Add `-msehost none` to your Viz Multichannel shortcut as explained in Startup Parameters.

**See Also**

- Channel Settings

### 4.3.7 Enabling Logging

Logging will help you troubleshoot issues that may occur. Note that the log produced by Viz Multichannel only reflects what the Media Sequencer attempted to play out (as opposed to what was actually played out).

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

#### To Enable Logging

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

**See Also**

- Logging

## 4.4 Running Viz Multichannel

This section contains information on the following topics:

- Media Sequencer
- Startup Parameters
- Time Format and Timezones

### 4.4.1 Media Sequencer

There should **always** be a connection to the Media Sequencer(s) when running Viz Multichannel. 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 Windows Services component and make sure Sequencer is running and set to autostart. You can browse the complete Media Sequencer Documentation (assuming it's running on standard port) by browsing to:

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

> **Note:** When running Viz Multichannel in Master Control Room (MCR) mode you will need to configure your `channels.ini` file or else Viz Multichannel will not start in MCR mode.

Also, note that the Playlist Importer by default runs on the Media Sequencer. Hence, in order for any schedule to be imported correctly your Viz Multichannel client must set the correct Startup Parameters for your Media Sequencer.

### 4.4.2 Startup Parameters

| Parameter | Description | Example |
|---|---|---|
| `-channelini` | A path of `channel.ini` file that we want to use for MCR mode. If no `-channelini` is specified, MCP will read `channel.ini` file from the default path, `[ProgramData]\Vizrt\Multichannel\channels.ini` | `-channelini "C:\channel_1.ini"` |
| `-msehost` | Hostname or IP address for Media Sequencer. If `none` is used MCP will work in MCR mode, reading each channel's Media Sequencer from `channels.ini`. See 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` |
| `-vizdb` | host:db:user:pw Configure the Viz Engine 3.x database login for the local preview Engine. | `-vizdb localhost:VizDb:Guest` |

### 4.4.3 Time Format and Timezones

All "internal" time handling calculations and operations, i.e. the time *reference* in the Media Sequencer, is **UTC**- Universal Time Code. This is always true, no matter what timezone and locale settings are set on your Windows servers and PCs. This is a reference time only - your actual Window PC clients running the User Interface can and often will have different time-settings. This can easily get confusing, so please keep in mind the following rules:

- The Media Sequencer reference time is UTC.
- The Multichannel UI will show the *client* machine time. If you click on the time-display in the Multichannel UI it will show the Media Sequencer's time and local offset, if any.
  When importing and handling time-dependent items in Multichannel please make sure that:

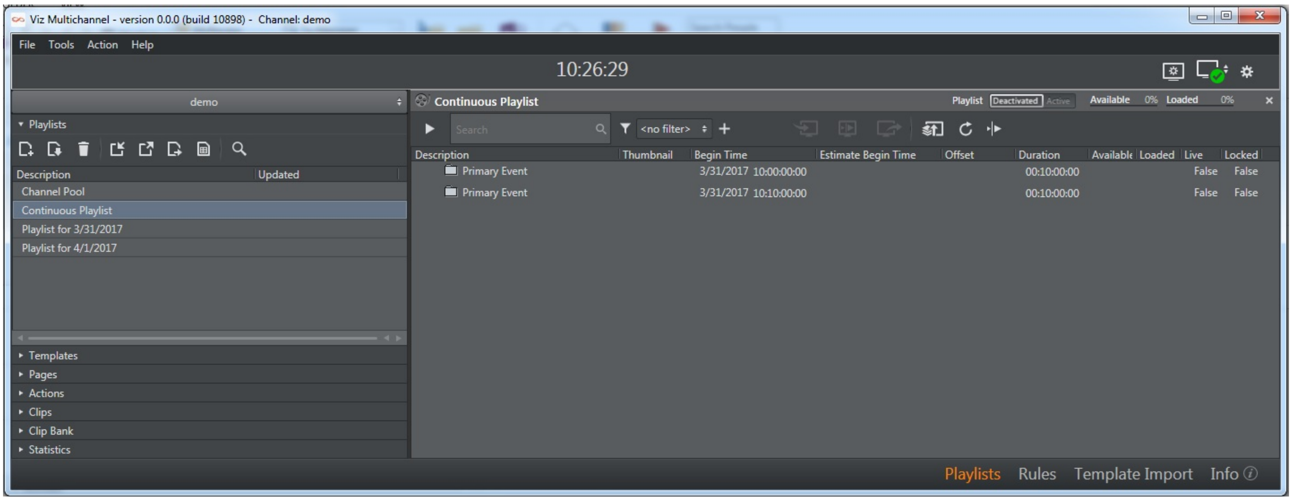
- Logic in rules should use the channel's timezone setting, see **Timezone Settings**.
- Logic for Page Content Filling (PCF) should use the playout channels timezone settings.
  A time-code in the **Rules** section always have to follow full time-code format, i.e. the value you write by hand has to be **HH:mm:ss:00**. The value that *begin_time* contains and what it will push to on-air when using it as an attribute to grab data from, is **HH:mm:ss**.

**See Also**

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

# 5 User Interface

This chapter is a reference chapter for the user interface. It contains information relevant for the Viz Multichannel client when used in scheduling and master control room (MCR) mode.



| Description | Updated |
|---|---|
| Channel Pool | |
| Continuous Playlist | |
| Playlist for 3/31/2017 | |
| Playlist for 4/1/2017 | |

| Description | Thumbnail | Begin Time | Estimate Begin Time | Offset | Duration | Available | Loaded | Live | Locked |
|---|---|---|---|---|---|---|---|---|---|
| Primary Event | | 3/31/2017 10:00:00:00 | | | 00:10:00:00 | | | False | False |
| Primary Event | | 3/31/2017 10:10:00:00 | | | 00:10:00:00 | | | False | False |

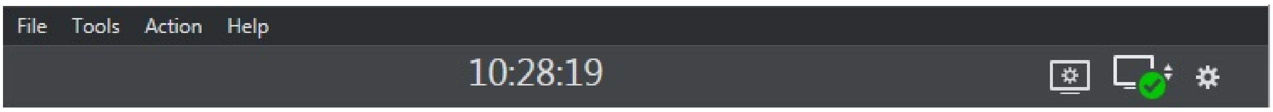
The main window is divided into the following sections:

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 always will be visible) using the **Action** menu page options.
2. Select panel to work on: **Playlists, Templates, Pages, Actions** 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 rest of this section contains presents:

- Main Menu and Toolbar
- Information Panel
- Channels
- Playlists
- Templates and Pages
- Clips
- Actions
- Import Scene
- Rules
- Page Editor
- Timeline Editor
- Combo Page Editor

## 5.1 Main Menu And Toolbar



From the top the Viz Multichannel user interface offers you a main menu and below it a dynamic (configurable) toolbar.

A dynamic toolbar means that you can configure which icons (representing actions) that should be available in the toolbar. You configure which icons to display under **Actions** from the main menu.

The actions themselves will be available as keyboard shortcuts, for example **F9** and **Ctrl-F*9*** to take or take-out the Logo-page, even if the icons representing the actions are not displayed.

- Main Menu
- Toolbar

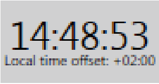
### 5.1.1 Main Menu

- File
  - **Exit**: Closes the application
- Tools
  - **Analyze logs**: allows you to open and analyze the As Run Logs. The As Run logs offers an easily readable overview of all elements run by the system. The logs are created by the Media Sequencer.
  - **Reports Manager**: allows you to create reports on multiple channels.
  - **Generate Channels File**: Generates a file containing information about channels file. This is required when running Viz Multichannel in MCR mode (see Master Control Workflow).
- Action
  - **Edit Mode**: enable edit mode
  - **Cleanup Engines**: clears all loaded graphics from memory on the program and preview renderer's for the output profile currently in use. It should be used before initializing a new playlist or in order to re-initialize the same playlist into the renderer’s memory. Note that cleanup commands will affect all clients that are connected to the same Media Sequencer, and using the same output profile (see Clear Layers).
  - **Clear Layers**: allows you to clear all or specific layers (i.e. front, middle or back) for your channel’s configured Viz Engines. Note: This will still leave the graphics in memory (see Toolbar and Cleanup Engines).
  - **Initialize**: use this option to manually initialize either the currently selected Channel or the Channel Pool. Initializing will fetch and load resources to the memory of the Viz Engine ensuring resources such as graphics are ready for playout. If not initialized, the Viz Engine will still load resources when needed, but pre-fetching can in certain situations be appropriate for flicker and artifact-free playout.
  - **Logo Page**: allows you to take or take out the defined logo page.
  - **Error Page**: allows you to take or take out the defined error page.

- **Video-only Page**: allows you to take or take out the defined video-only page.
- **Ticker**: allows you to import or export Viz Ticker3D playlists (see Working with Ticker Actions).
- **Actions**: allows you to import or export actions from an XML-file.
- **Template and Pages**: allows you to import or export template and pages to XML-files.
- **Video bypass**: When enabled, only video is shown. When disabled, video and graphics are shown.

- Help
  - **Help (F1)**: Launches the documentation in your preferred browser.
  - **About**: Shows the version of Viz Multichannel.

### 5.1.2 Toolbar

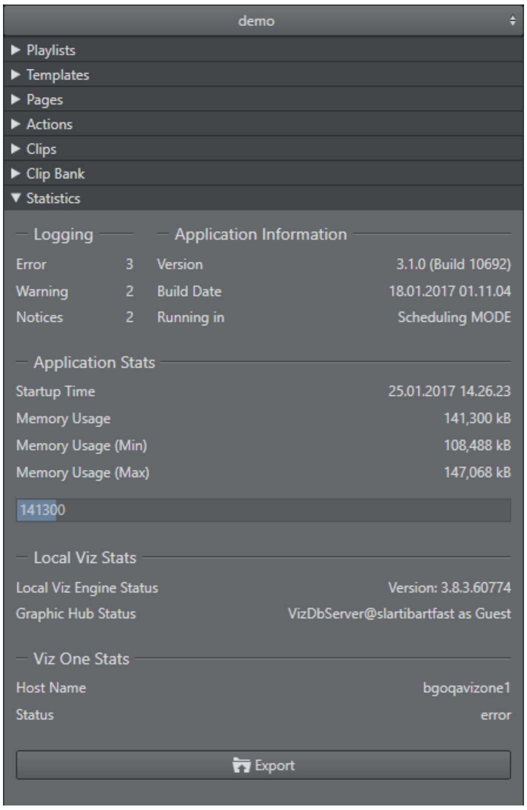
| Icon | Shortcut | Description |
|---|---|---|
| ⚙ | Configuration menu | |
| | Local Viz Engine status. Sub-Commands that can be selected: Show Commands, Show Status and Restart Viz. Note that these commands will be sent to the local Viz Engine, normally used for preview. For more information about 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. | |
|  14:48:53<br>Local time offset: +02:00 | Shows the time of the currently selected channel's Media Sequencer server, with the local client (Multichannel 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. | |
| | F9 | Take **in** the defined Logo Page. |

| | | |
|---|---|---|
| | Ctrl-F9 | Take **out** the defined Logo Page. |
| | F10 | Take **in** the defined Error Page. |
| | Ctrl-F10 | Take **out** the defined Error Page. |
| | F11 | Take **in** the Video Only Page. |
| | Ctrl-F11 | 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 the Viz Engine from impacting the video signal. Important: for this to work, the 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 the Viz Engine from impacting the video signal. Important: for this to work, the Viz Engine must be properly configured. See the Matrox and Watchdog sections in the *Viz Engine Administrator Guide*. | |

## 5.2 Information Panel

The Information panel is an overview dashboard, also useful for debugging.



Logging

Error 3
Warning 2
Notices 2

Application Information

Version 3.1.0 (Build 10692)
Build Date 18.01.2017 01.11.04
Running in Scheduling MODE

Application Stats

Startup Time 25.01.2017 14.26.23
Memory Usage 141,300 kB
Memory Usage (Min) 108,488 kB
Memory Usage (Max) 147,068 kB

141300

Local Viz Stats

Local Viz Engine Status Version: 3.8.3.60774
Graphic Hub Status VizDbServer@slartibartfast as Guest

Viz One Stats

Host Name bgoqavizone1
Status error

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

## 5.3 Channels

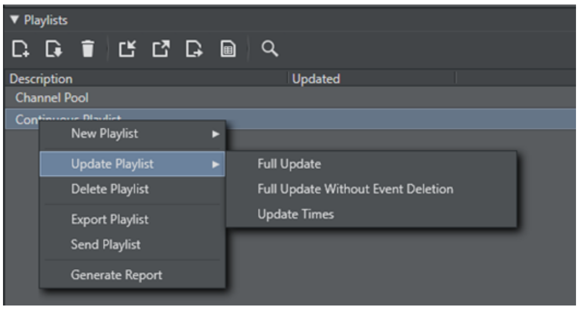
At the top of the left pane area is the channel selection combo-box. The channel selection combo-box lists all your different channels and allows you to switch between them. You can also click on the channel icon above the combo-box in order to switch to the corresponding channel.

### 5.3.1 Channel Selection Combo Box

**See Also**

- Working with Channel Settings

## 5.4 Playlists



The Playlists pane displays the playlist(s) created for the selected channel. Viz Multichannel supports two types of playlists; continuous and broadcast day playlist. In addition it features a location to store prepared non-scheduled events named Channel Pool.

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

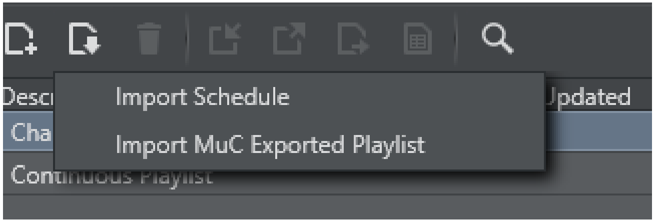
Right-clicking on a playlist in the left pane displays a context menu of playlist commands:

- **Update Playlist**: allows you to select one of the following update options:
  - Full Update: updates the entire playlist, primary and secondary events.
  - Full Update Without Event Deletion: updates the entire playlist, primary and secondary events; however, it will not delete already added events.
  - Update Times: Allows you to only update the event times according to the imported schedule.
- **Delete Playlist**: Deletes the playlist.
- **Export Playlist**: Exports the playlist to a native Viz Multichannel XML format. You can, as part of the export, also export the Graphics and Data.
- **Send Playlist**: Sends the playlist to the master control room.
- **Generate Report**: Generates an HTML report of the selected playlist.

## 5.4.1 To Import a MuC Exported Playlist

A playlist that has previously been exported from MuC as an xml file, can be imported back using the **Import MuC Exported Playlist** menu item.
This will import the xml playlist to MuC without performing any mapping settings. Playlist data is set as it appears in the imported file.
If the contained broadcast date already exists, the existing playlist won't  be replaced with the imported data. You must first delete the existing playlist.

- Click the Import icon and select **Import MuC Exported Playlist**



This section presents:

- Channel Pool
- Playlist Toolbar
- Playlist Keyboard and Mouse Shortcuts
- Playlist Columns
- Playlist Context Menu
- Playlist Filters

## 5.4.2 Channel Pool

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

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

A separate channel pool is added for each channel.

## 5.4.3 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 been stopped |

| Icon | Description |
|---|---|
| Deactivated Active<br>Deactivated Active | When a playlist is **active** it is added to the Media Sequencer. Once added to the Media Sequencer it will start to check that all resources (e.g. videos and imagery) are available (see Available status) to Viz Engine for playout. This means, amongst other things, that videos are transferred from e.g. Viz One to Viz Engine |
| Available 0%<br>Available 100% | **100% available** means that full screen video clips are available to Viz Engine for playout |
| Loaded 0%<br>Loaded 100% | **100% loaded** means that all pages (i.e. 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 |
| Search | Search enables you to search the description field of your playlist elements. To see the next result, press Enter |
| <no filter> ▼ | Predefined filters allow you to filter the playlist. Click the funnel icon to create, edit or delete filters. See how 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's. When all is done, the loaded status is set to 100%.See Playlist Context Menu on how to initialize the playlist and individual elements from the context menu. |
| ↻ | Refreshes the data in use by the pages |

### 5.4.4 Playlist Keyboard and Mouse Shortcuts

| Function | Key | Mouse |
|---|---|---|
| Moves one or multiple 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 | |
| Collapses the group | Arrow + Left | Double-click left mouse button |
| Expands the group | Arrow + Right | Double-click left mouse button |
| Expands all groups | Ctrl + Numpad + | |
| Collapses all groups | Ctrl + Numpad - | |
| Selects all events | Ctrl + A | |
| Opens the search | Ctrl + F | |
| Selects individual events | Ctrl | Left mouse button |
| Removes the selected event(s) | Delete | |
| Makes an event’s Description field editable | F2 | Click left mouse button x 2 on the Description field |
| Moves the focus to the top of the playlist | Home | |
| Moves the focus to the end of the playlist | End | |

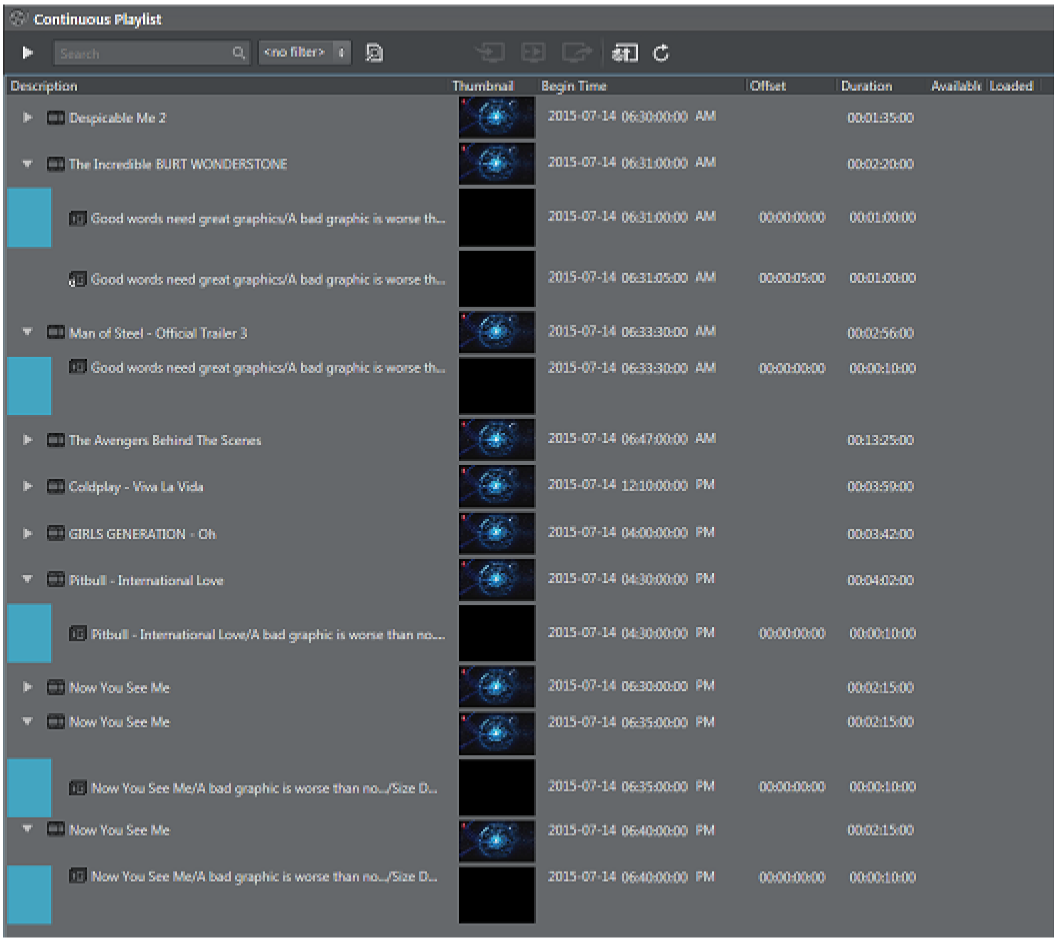
| Function | Key | Mouse |
|---|---|---|
| Moves the focus one section down | Page Down | |
| Moves the focus one section up | Page Up | |
| Selects events (up or down) | Shift | Left mouse button |
| Displays the context menu | | Right mouse button |

### 5.4.5 Playlist Columns

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

To **edit the playlist columns** you have to check - for each column - the **Edit Events** check-box in your mapping table (see Table Columns).

## Right pane showing playlist


Continuous Playlist

| Description | Thumbnail | Begin Time | Offset | Duration | Availabl | Loaded |
|---|---|---|---|---|---|---|
| Despicable Me 2 | | 2015-07-14 06:30:00:00 AM | | 00:01:35:00 | | |
| The Incredible BURT WONDERSTONE | | 2015-07-14 06:31:00:00 AM | | 00:02:20:00 | | |
| Good words need great graphics/A bad graphic is worse th... | | 2015-07-14 06:31:00:00 AM | 00:00:00:00 | 00:01:00:00 | | |
| Good words need great graphics/A bad graphic is worse th... | | 2015-07-14 06:31:05:00 AM | 00:00:05:00 | 00:01:00:00 | | |
| Man of Steel - Official Trailer 3 | | 2015-07-14 06:33:30:00 AM | | 00:02:56:00 | | |
| Good words need great graphics/A bad graphic is worse th... | | 2015-07-14 06:33:30:00 AM | 00:00:00:00 | 00:00:10:00 | | |
| The Avengers Behind The Scenes | | 2015-07-14 06:47:00:00 AM | | 00:13:25:00 | | |
| Coldplay - Viva La Vida | | 2015-07-14 12:10:00:00 PM | | 00:03:59:00 | | |
| GIRLS GENERATION - Oh | | 2015-07-14 04:00:00:00 PM | | 00:03:42:00 | | |
| Pitbull - International Love | | 2015-07-14 04:30:00:00 PM | | 00:04:02:00 | | |
| Pitbull - International Love/A bad graphic is worse than no... | | 2015-07-14 04:30:00:00 PM | 00:00:00:00 | 00:00:10:00 | | |
| Now You See Me | | 2015-07-14 06:30:00:00 PM | | 00:02:15:00 | | |
| Now You See Me | | 2015-07-14 06:35:00:00 PM | | 00:02:15:00 | | |
| Now You See Me/A bad graphic is worse than no.../Size D... | | 2015-07-14 06:35:00:00 PM | 00:00:00:00 | 00:00:10:00 | | |
| Now You See Me | | 2015-07-14 06:40:00:00 PM | | 00:02:15:00 | | |
| Now You See Me/A bad graphic is worse than no.../Size D... | | 2015-07-14 06:40:00:00 PM | 00:00:00:00 | 00:00:10:00 | | |

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.

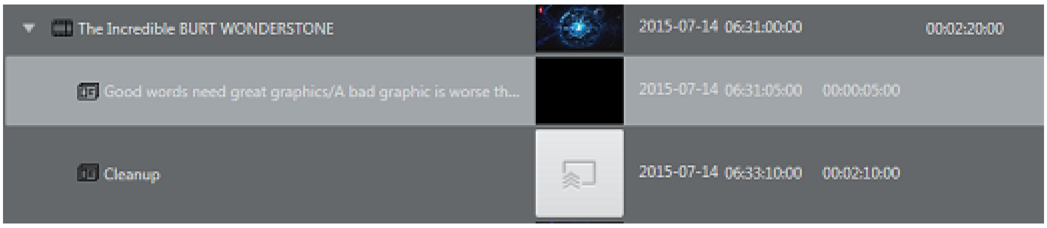
> ⚠ **Note:** If there are any errors found in the playlist, the primary element(s) that is causing the error will be shown in red. Hover the mouse over the red “no-entry” icon sign for the element(s) in error to find out what.

The default columns available are:

- **Available**: Shows if a video clip is available to your playout device.
- **Begin Time**: Shows the element’s begin time (see also Duration and Offset, and External Data).
- **Channel**: Shows the name of the currently selected device channel for any given element in the playlist (see section *Configuring Device Channels*).
- **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.

- **Duration**: Shows the duration of the element (see also Begin Time and Offset).**End Time**: Shows the end time (Begin Time + Duration) for each element.
- **Event ID**: For playlist updates. It should be unique since it is the events identifier. If it is not unique, you must construct a unique identifier with other properties toward the internal property `name`.
- **Layer**: Shows which logical layer the element will be playout out on (i.e. front, middle, or back). For pages the middle layer is the default. For full screen video clips, the back layer is the default.
- **Loaded**: Shows the loaded status of pages on the Viz Engine program device channel.
- **Mark In**: Shows the mark in point for a video clip set by the schedule.
- **Mark Out**: Shows the mark out point for a video clip set by the schedule.
- **Media ID**: Shows the ID of the element.
- **Offset**: Shows the offset of the secondary event relative to its primary event (see also Begin Time and Duration).
- **Page Name**: Shows the name of the page (i.e. secondary event).
- **Template Description**: Displays the template name as saved in Viz Artist.
- **Template ID**: Displays the template name as saved in Viz Artist.
- **Thumbnail**: Shows a snapshot/thumbnail of the template used in a Secondary Event.

Elements in the playlist that are **inactivated** will be show with a color background (gray background) as illustrated in the screenshot below.



| | | | |
|---|---|---|---|
| The Incredible BURT WONDERSTONE | 2015-07-14 06:31:00:00 | | 00:02:20:00 |
| Good words need great graphics/A bad graphic is worse th... | 2015-07-14 06:31:05:00 | 00:00:05:00 | |
| Cleanup | 2015-07-14 06:33:10:00 | 00:02:10:00 | |

**Note:** Elements in the Multichannel playlist can be de-activated by external events or applications. For example by the Media Sequencer because required resources are not available for the element.

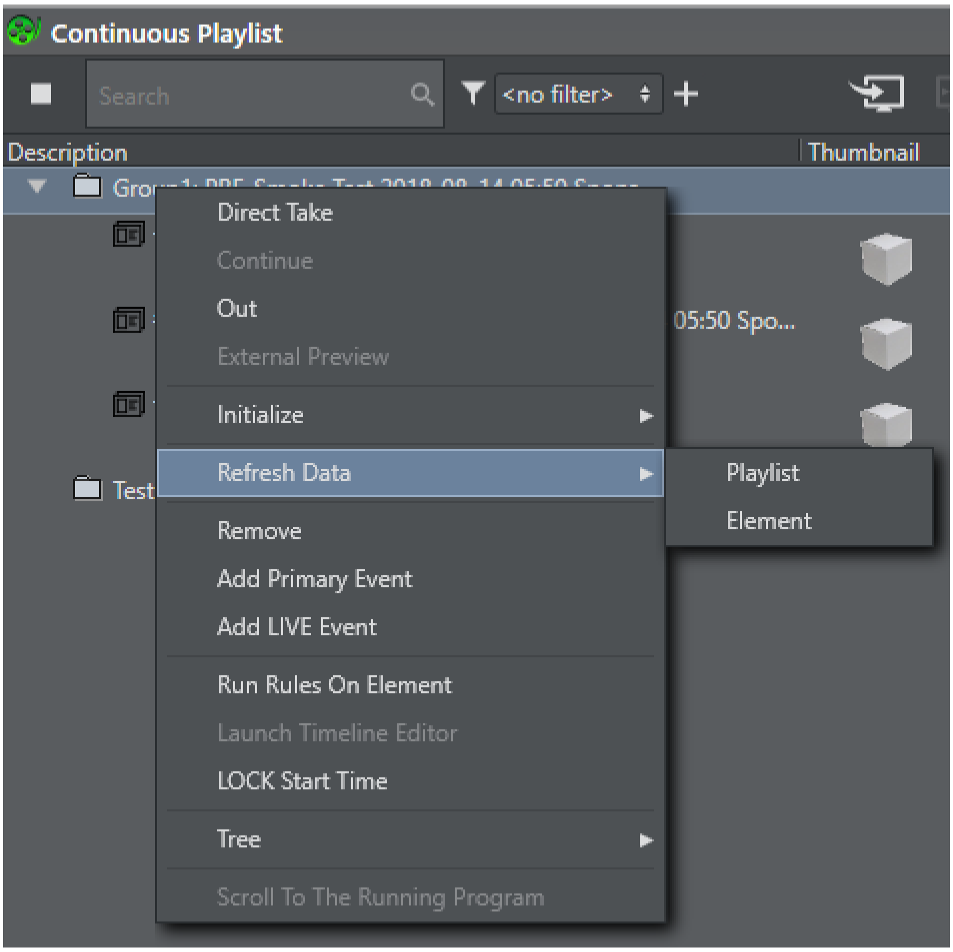
## 5.4.6 Playlist Context Menu

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*).
- **Enable** or **Disable Element**: *Only* for secondary events. Gives you the possibility to either enable or disable a secondary element.
- **Re-try creating page from this element**: This menu item will only shown for elements with errors. An example error condition that could trigger this is: A request to create a page is

pushed via the Integrations Hub, but the page is not found in the Page Pool at creation time. A stub-version containing meta-information will be created. If the requested page is later created in the Page Pool you can use this menu item on the element and store the page-relevant data under the element.

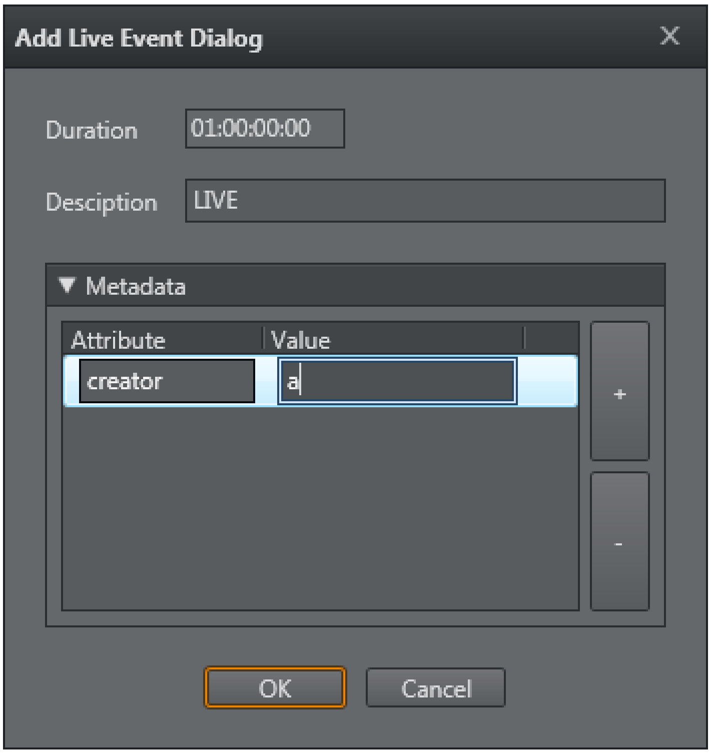
- **Initialize** (see section *Playlist Toolbar*).
    - **Playlist**: Initializes the entire playlist.
    - **Element**: Initializes the selected playlist element.
- **Refresh Data**:



Request a refresh of automatic page content or primary event VizOne metadata, depending on the element selected and refresh option, as can be found on the picture above:

- **Playlist**: Clicking on this option, it will iterate through entire playlist and update all elements.
- **Element**: Clicking on it will update page data or VizOne metadata only for the selected element of the playlist.

- **Remove**: Removes an element from the playlist. This action cannot be undone.
- **Add Primary Event**: Adds a primary event to the playlist.
- **Add Live Event**: Adds a live event



Duration 01:00:00:00

Desciption LIVE

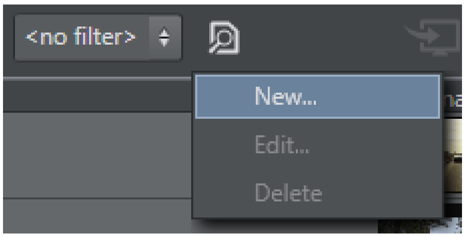
| Attribute | Value |
|---|---|
| creator | a |

- **Run Rules On Element**: enabled for group and video primary events only. When selected, it will run defined Rules 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 as indicated in the following picture:
- **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 showing secondary events
  - **Collapse All**: Collapses all primary events hiding secondary events
  - **Hide Empty Groups**: Hides all empty groups
  - **Wrap Text**: Wraps the text according to the 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.

### 5.4.7 Playlist Filters

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

### To Add a Playlist Filter

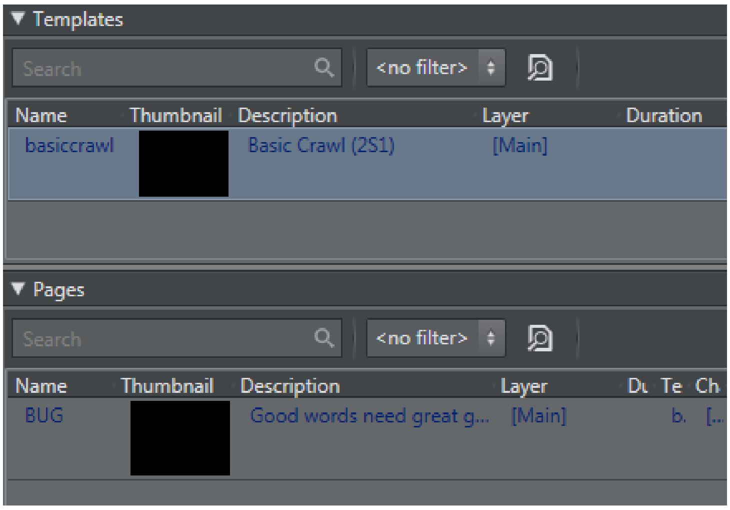


1. Click the **funnel** icon, and from the menu click **New...**.
    - This will open the Edit filter window.
2. Enter a **Filter name**.
3. Select whether the filter should apply if:
    - All conditions match, or
    - One of the conditions matches, or
    - None of the conditions match.
4. Check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

### To Manage a Playlist Filter

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

## 5.5 Templates And Pages



| Name | Thumbnail | Description | Layer | Duration |
|---|---|---|---|---|
| basiccrawl | | Basic Crawl (2S1) | [Main] | |

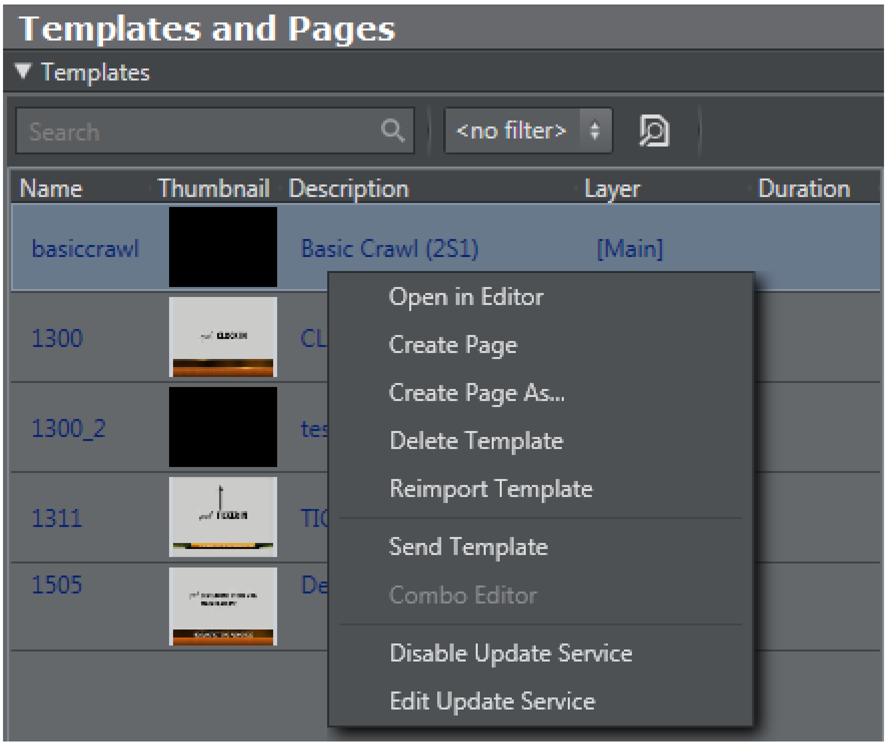
| Name | Thumbnail | Description | Layer | Du | Te | Ch |
|---|---|---|---|---|---|---|
| BUG | | Good words need great g... | [Main] | | b. | [.. |

In the Templates and Pages panel, you can view all imported templates, and pages based on those templates. You can also filter your templates and pages view, and add content to pages using the Page Content Filling Window.

- Template Context Menu
- Page Context Menu
- Page Content Filling Window
- Database Connection Window
- Template and Page Filters

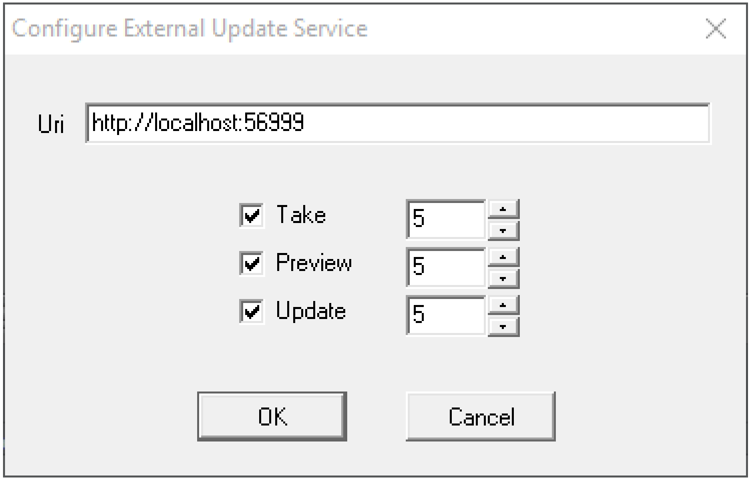
## 5.5.1 Template Context Menu

Right-clicking a **template** opens the 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. The page can be opened in the template editor for editing.
- **Delete Template**: Deletes the selected template/data.
- **Reimport Template**: Will scan the scene and re-import the selected template. The resulting object will be recreated. When reimporting, you will first be asked if you want to do this and then a reminder will be shown: existing pages description might not reflect the true data content until refreshed by update.
- **Add To Favorites**: Adds the template to your pre-defined Favorites filter.
- **Send Template**: Sends the selected template(s) to the master control room. This option is only enabled if the **Remote Host** is defined in the General Settings.
- **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 the Combo Page Editor section.

- **Enable Update Service**: Enables you to assign an update service for the selected template.



Configure External Update Service

Uri: http://localhost:56999

☑ Take: 5

☑ Preview: 5

☑ Update: 5

- **Disable Update Service**: Disables the update service for the selected template.
- **Edit Update Service**: Edit the update service for the selected template by setting the service URI. A timeout on the available commands (i.e. take, preview and update) may be set as well. This will delay the given command sent by the Media Sequencer to the Viz Engine.

## 5.5.2 Page Context Menu

Right-clicking a **page** opens the following context menu:

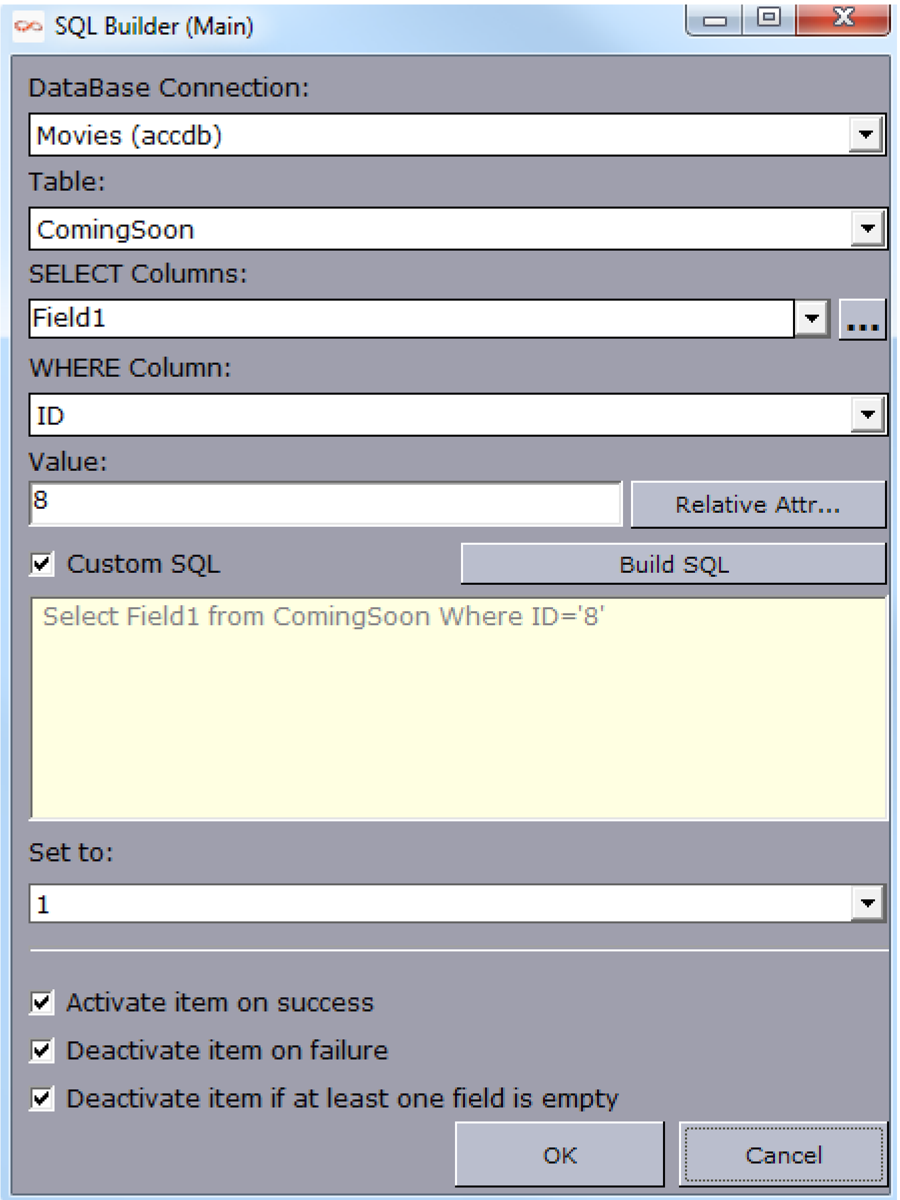
- **Open in Editor**: Opens the template editor which allows you to update the page or save it as a new page based on the same template.
- **Save Page as**: Creates a page based on an existing page. The page can be opened in the template editor for editing.
- **Rename**: Change name of selected page. 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 To add a template or page filter.
- **Send Page**: Sends the page to the defined remote hosts, see General Settings.
- **Reload Page**: Reloads the scene referenced by the page on the device defined as the program Viz Engine, see Configuring Device Channels.
- **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.
  Right-clicking the **column headers** allows you to customize your templates and pages view.

- **Channel:** Allows you to set a default playout channel (see Configuring Device Channels) for a given page.
- **Description**: Shows the template or page description.
- **Duration**: Shows the duration of the template or page. Click the area and type to set a default duration for the page.
- **TTL**: Time To Live. This header is available for Pages pool *only*. It allows to set an “expire” (TTL) date for a page. Page(s) that has a TTL-date equal to (or older) than the current date will be removed when PLHM is run.
- **Layer**: Defines the layer the scene should play in. Available options are Front, Middle, and Back. Default is Middle.
- **Name**: Shows the name of the template or page.
- **Template Name**: For Pages an additional column shows the template associated with a given page.
- **Thumbnail**: Shows a thumbnail image of the template or page.
- **Store as Default**: Allows you to arrange and store your columns as a default column arrangement.
- **Use Default**: Reverts to your current default column arrangement.
- **Auto Fit Columns**: Automatically fits the columns to the available space.

### 5.5.3 Page Content Filling Window

This is described in the section Page Content Filling.

### 5.5.4 Database Connection Window



DataBase Connection: Movies (accdb)

Table: ComingSoon

SELECT Columns: Field1

WHERE Column: ID

Value: 8

☑ Custom SQL

```
Select Field1 from ComingSoon Where ID='8'
```

Set to: 1

☑ Activate item on success

☑ Deactivate item on failure

☑ Deactivate item if at least one field is empty

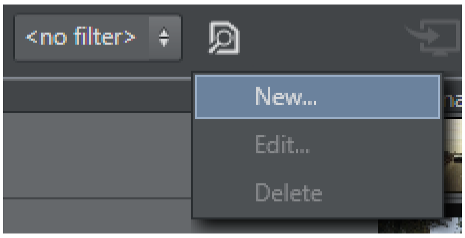
- **Database Connection:** Allows you to select from your list of pre-configured Database Connections.
- **Table**: Allows you to select a table from your selected database.
- **SELECT Columns**: Allows you to select a column from your selected table.
- **WHERE Column**: Allows you to select from where the value should be selected.
- **Value**: Allows you to set which value that should be selected from the WHERE column.
- **Relative Attr...**(button): Allows you to select a relative attribute from the Page Content Filling Window.
- **Custom SQL**: Allows you to write a custom SQL query with multiple dynamic mappings (e.g. `Select Field1 from ComingSoon Where Field4='PG-13' AND Field2='23.12.2013'`).

- **Build SQL** (button): Builds the query based on your previous selections.
- **Set to:** Allows you to select which page field that should be updated by the query. Ensure that the SELECT columns match the page fields!
- **Activate item on success:** Ensures that playlist items will be marked as active if the query is successful and data fields have been populated. Leave it unchecked if you do not want the success of the query to have any effect on the status of a playlist item.
- **Inactivate item on failure:** Ensures that playlist items will be marked as inactive if the query fails and data fields have not been populated. Leave it unchecked if you do not want the failure of the query to have any effect on the status of a playlist item.
- **Deactivate item if at least on field is empty:** Ensures that playlist items will be marked as inactive when one of the fields the query returns is empty. Leave it unchecked if you do not want an empty field to have any effect on the status of a playlist item.

## 5.5.5 Template and Page Filters

### To Add a Template or Page Filter

The Templates and Pages panel allows you to add filters to quickly sort the list of templates or pages.



1. Click the **funnel** icon, and from the menu click **New...**.
   - This will open the Edit filter window.
2. Enter a **Filter name**.
3. Select if the filter should apply when:
   - All conditions match, or
   - One of the conditions matches, or
   - None of the conditions match.
4. Continue to define your filter by adding one or multiple conditions.
   - As you define your conditions the list of templates or pages will be filtered, helping you to verify that the filter is configured correctly.

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

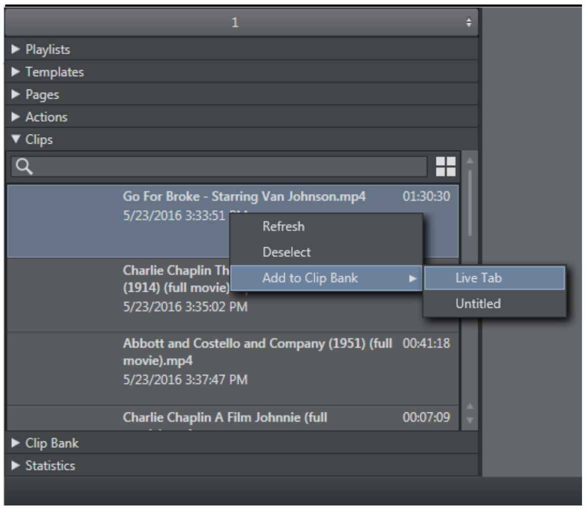
### To Manage a Template or Page Filter

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

**See Also**

- Import Scene
- Working with Templates and Pages

## 5.6 Clips



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 operators 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

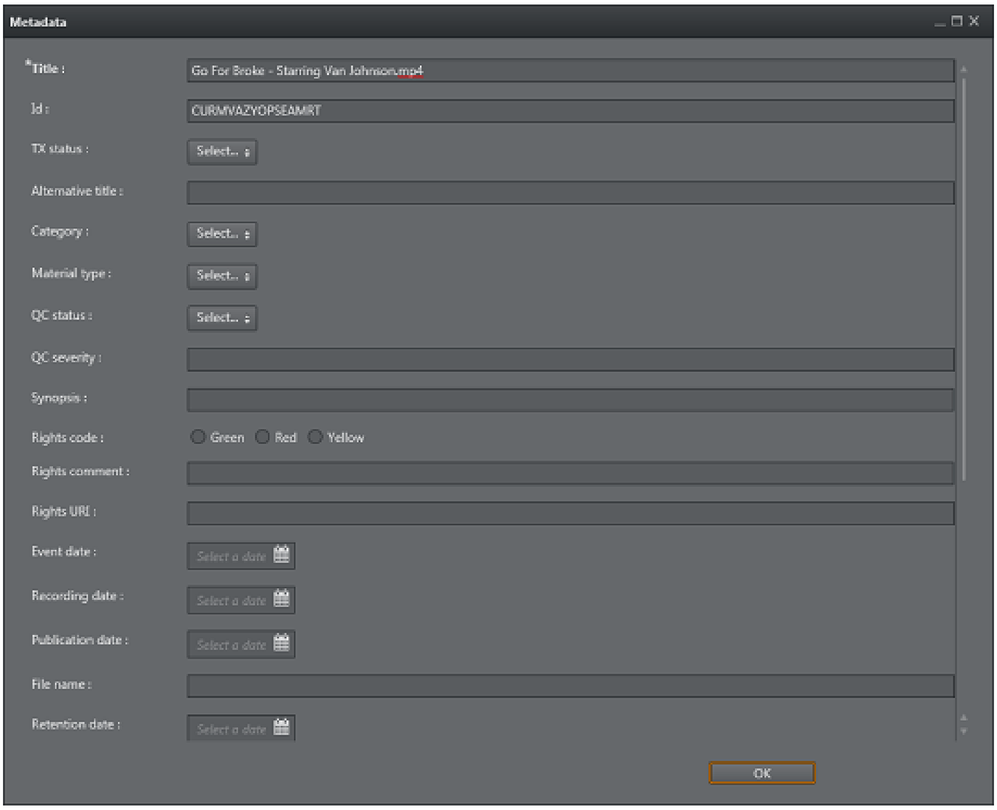
### 5.6.1 Clips Context Menu

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

- **Refresh**: Reload all clips from Viz One.
- **Deselect**: Deselect a clip
- **Add to Clip Bank**: add the selected Viz One clip into the Multichannel Clip Bank. You can select which clip bank tab the clip should be added to.

### 5.6.2 Clip Metadata

Click on a clip to view or edit metadata:



Metadata

*Title : Go For Broke - Starring Van Johnson.mp4

Id : CURMVAZYOPSEAMRT

TX status : Select...

Alternative title :

Category : Select...

Material type : Select...

QC status : Select...

QC severity :

Synopsis :

Rights code : ○ Green ○ Red ○ Yellow

Rights comment :

Rights URI :

Event date : Select a date

Recording date : Select a date

Publication date : Select a date

File name :

Retention date : Select a date

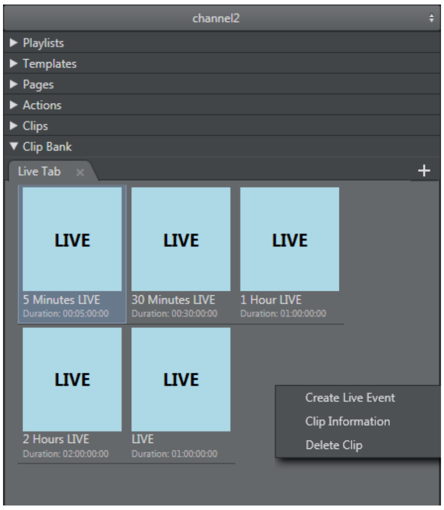
OK

## See Also

- Viz One Integration

# 5.7 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 into the running playlist. When you first create a channel, the default clip tab name "LIVE" with 4 default LIVE events will be created.

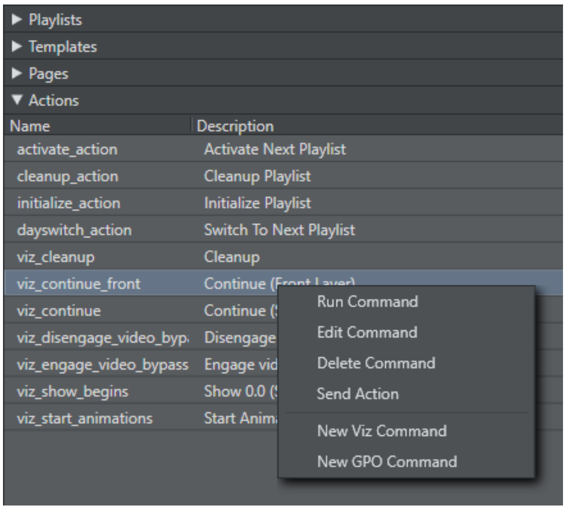


You can add a new tab by clicking the "+" sign on the top right of Clip Bank tab.

Right-Click for the options menu where you can:

- Create Live Event
- Clip Information
- Delete Clip

## 5.8 Actions



| Name | Description |
|---|---|
| activate_action | Activate Next Playlist |
| cleanup_action | Cleanup Playlist |
| initialize_action | Initialize Playlist |
| dayswitch_action | Switch To Next Playlist |
| viz_cleanup | Cleanup |
| viz_continue_front | Continue (Front Layer) |
| viz_continue | Continue ( |
| viz_disengage_video_byp | Disengage |
| viz_engage_video_bypass | Engage vid |
| viz_show_begins | Show 0.0 ( |
| viz_start_animations | Start Anim |

Viz Multichannel’s Actions pane allows you to define General Purpose Output (GPO) commands, Viz Ticker commands and Viz Engine commands. GPO commands are handled by Media Sequencer. Both Viz Ticker and Viz Engine commands can be used in Viz Multichannel to automate certain tasks. The commands behave as secondary events, in that you can also apply rules and schedule them in the playlist. Ticker actions are accessible from the Actions panel seen in the left pane. Ticker actions integrates tickers from Viz Ticker. Once the ticker has been created, it can be imported into Viz Multichannel.

Actions can for example include:

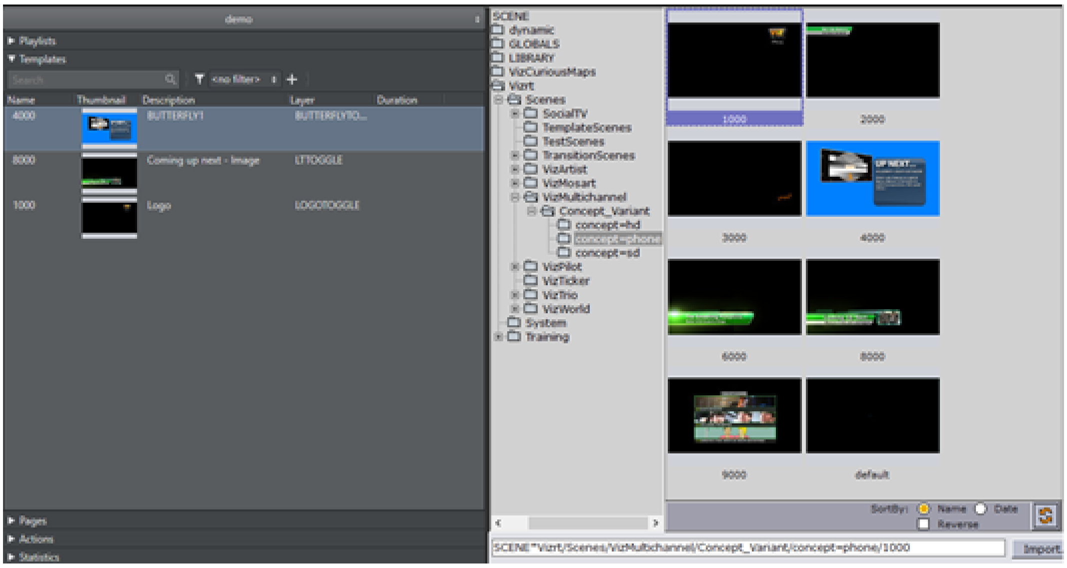
- Playing specific data elements on air
- Clearing specific render layer
- Sending Stop, Continue, Take and other such commands to a specific render layer
- Associated with playlists as secondary events

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

### See Also

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

## 5.9 Import Scene



| Name | Thumbnail | Description | Layer | Duration |
|---|---|---|---|---|
| 4000 | | BUTTERFLY1 | BUTTERFLYTO... | |
| 8000 | | Coming up next - Image | LTTOGGLE | |
| 1000 | | Logo | LOGOTOGGLE | |

SCENE*Vizrt/Scenes/VizMultichannel/Concept_Variant/concept=phone/1000

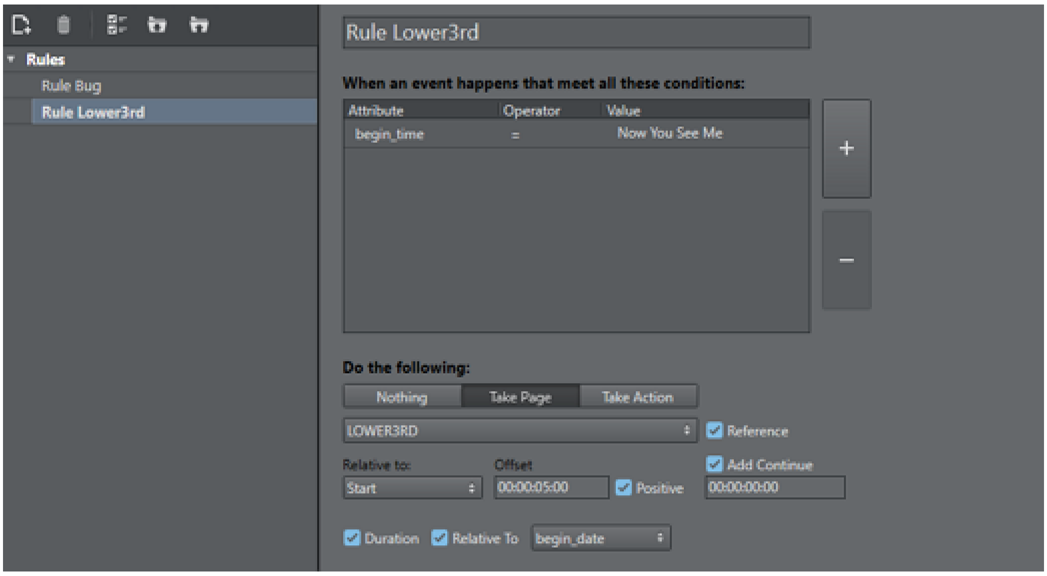
- Click the **Scenes** button on lower right to view the contents of your Viz Graphics Hub and select scenes to import.
- Double-click scenes to import or select several by clicking and then import.

> To view the content you need to have a **working Viz Graphics Hub connection**. For information on how to configure your local Viz Engine Database connection settings see the Viz Engine Administrator Guide

### See Also

- Templates and Pages
- Working with Templates and Pages

## 5.10 Rules



Rule Lower3rd

When an event happens that meet all these conditions:

| Attribute | Operator | Value |
|---|---|---|
| begin_time | = | Now You See Me |

Do the following:

LOWER3RD ☑ Reference

Relative to: Start

Offset: 00:00:05:00 ☑ Positive

☑ Add Continue 00:00:00:00

☑ Duration ☑ Relative To begin_date

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

- **Description**: 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 to meet for the event to happen.
- **Take Page or Take Action:** Sets the page or action on which the rule will be applied.
- **Reference**: If required, associates the page, Viz or GPI command as referenced (see Referenced Pages and Referenced Actions).
- **Relative To**: Allows you to set the time relative to the offset that a page or action will be scheduled. Your options are:
  - **Start**: Schedules the page or action relative to the event’s start time.
  - **End**: Schedules the page or action relative to the event's end time.
  - **Program Duration**: Schedules the page or action relative to the event’s 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.
  - **Relative To**: set the duration of page from rule.
- **Offset**: Allows you to set the offset time. Offsets may be a positive or negative number (see **Positive** option). The format is `hh:nn:ss:ff`.
  - Configuring a negative value will push the page into another primary event, and not the one it originally resided under. This will correctly place the page with the proper offset to the original primary event.
    For example, if a 1 minute count-down timer page is added that counts until a specific program starts (with an unknown number of programs before), you can configure the offset to be 1 minute and uncheck the **Positive** option in order for it to be a negative value. The page will then be inserted correctly at the correct offset, no matter how many previous primary events there might be.

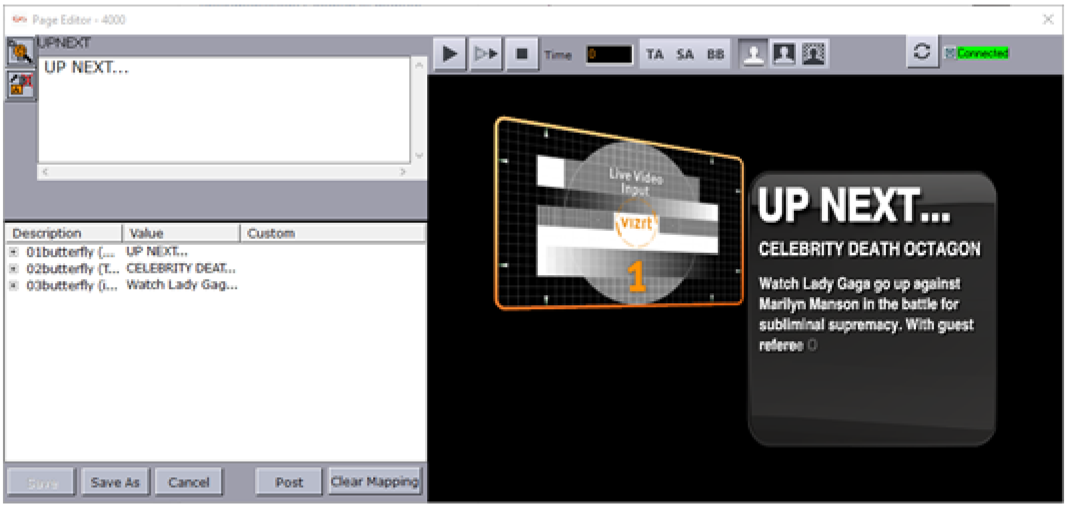
- **Positive**: Determines if the number is positive (checked) or negative (not checked). A page may be added at 00:02:00:00 (negative) which allows it to be added before a primary event is started (as a count-down timer).
- **Add Continue**: Allows you to select the check box if the page (i.e. the graphics) has been designed with a stop point, and a Continue action is needed for full execution. Enabling **Add Continue** you can set the offset time for the Continue action, if required by the previous step.
- **Duration**: If the **Duration**-option check box is selected (activated), then the related *Relative To* option check box can optionally be selected as well.

> ⚠ **Note:** If both **Duration** and **Relative To** check-boxes are selected you will not be able to input a fixed duration. Instead, you must select the appropriate attribute from the drop-down list. Each primary event will then have different duration based on the value of the selected attribute.

### See Also

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

## 5.11 Page Editor



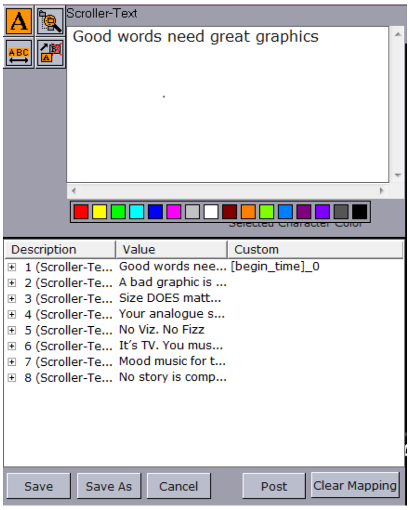
| Description | Value | Custom |
|---|---|---|
| ⊞ 01butterfly (... | UP NEXT... | |
| ⊞ 02butterfly (T... | CELEBRITY DEAT... | |
| ⊞ 03butterfly (i... | Watch Lady Gag... | |

Viz Multichannel basically offers two types of content editors; a page editor and a Timeline Editor. The page editor, which this section will describe, allows you to edit secondary events such as graphics and graphics with embedded video clips. Depending on the design of the graphics and the functionality it exposes the available options for editing the content will change.

This section contains information on the following topics:

- Page Editor Tab Fields
- Graphics Control Buttons
- Post Rendering

## 5.11.1 Page Editor Tab Fields



Good words need great graphics

| Description | Value | Custom |
|---|---|---|
| ⊞ 1 (Scroller-Te... | Good words nee... | [begin_time]_0 |
| ⊞ 2 (Scroller-Te... | A bad graphic is ... | |
| ⊞ 3 (Scroller-Te... | Size DOES matt... | |
| ⊞ 4 (Scroller-Te... | Your analogue s... | |
| ⊞ 5 (Scroller-Te... | No Viz. No Fizz | |
| ⊞ 6 (Scroller-Te... | It's TV. You mus... | |
| ⊞ 7 (Scroller-Te... | Mood music for t... | |
| ⊞ 8 (Scroller-Te... | No story is comp... | |

The panel at the left of **Page Editor** shows the available tab fields (editable elements of the graphics) for the page or template currently loaded.
Click on a tab field to display the available options.

When a tab field for video elements is selected you will be able to search for, filter, and add Clips to your graphics. A page containing video elements will automatically search for the video and preview it.

- **Save**: Overwrites the page using the existing save name.
- **Save As**: Saves the template or page as a new element.
- **Cancel**: Cancels all changes.
- **Post**: Switches to Post Rendering mode (only visible in page editing mode).
- **Clear Mapping**: Clicking this button will remove any Page Content Filling mapping done for each tab-field as well as any DB mapping for the page. You must press **Save** afterwards to actually have the page saved.

## 5.11.2 Graphics Control Buttons

The preview window shows a WYSIWIG (what you see is what you get) representation of the graphics and video clip. 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 the Page Editor Tab Fields section). For the full screen video preview there is also a Timeline Editor.

**Preview window buttons**

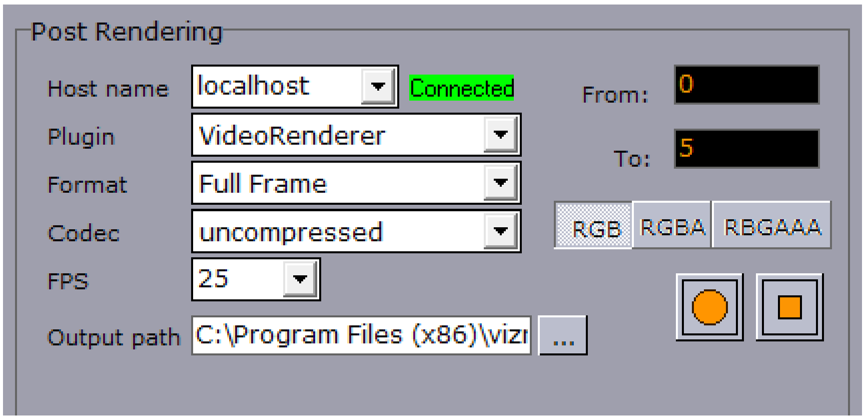
| Icon | Description |
|---|---|
| ▶ | Starts the scene animation currently loaded in the local preview |
| ⏯ | Continues the scene animation currently loaded if it contains any stop points |

| Icon | Description |
|---|---|
| ■ | Stops the scene animation |
| Time | With the Time slider it is possible to scrub the timeline manually by clicking in the value field, keeping the button pressed and moving the mouse horizontally back and forth. The cursor will change to an arrow to indicate that the time value can be changed |
| TA | Show or hides the Title Area in the Preview window |
| SA | Shows or hides the Safe Area in the Preview window |
| BB | Shows or hides the Bounding Box for the tab field currently selected in the Preview window |
| 👤 👤<br>👤 | The three buttons can be used to preview graphics in three different modes. The first shows an RGB version of the graphics with a default background color (black). The second shows the Key version of the graphics. The third shows the graphics keyed on top of a still image. Currently not supported by Viz Multichannel 2.7 |
| ⟳ | Press the icon to refresh your view |
| ✓ | This icon indicated the Viz Engine connection; a green check mark indicates that the Engine is connected properly.You can click the drop-down for further actions:- Show Commands: Display the Viz Engine console window where you can observe all commands sent to the Engine and replies sent from the Engine.- Show Status: get the Viz Engine status- Restart Viz: will try to restart the Viz Engine. This is a disruptive action.- Reconnect Viz: reconnect the Viz Engine |

**See Also**

- Timeline Editor

### 5.11.3 Post Rendering



Post Rendering

Host name: localhost — Connected

Plugin: VideoRenderer

Format: Full Frame

Codec: uncompressed

FPS: 25

Output path: C:\Program Files (x86)\vizr

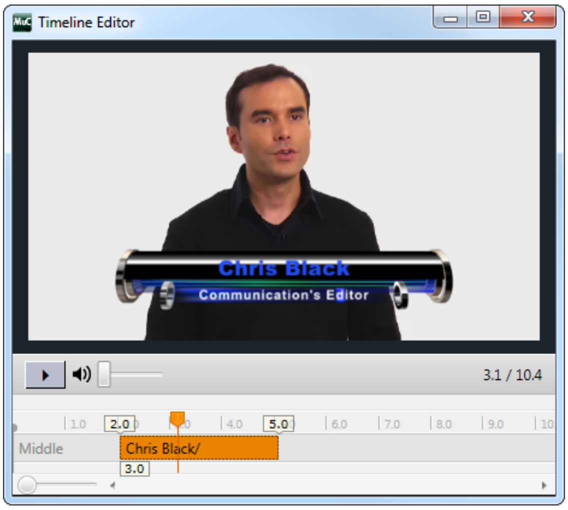
From: 0

To: 5

The Post option in the Page Editor allows you to generate videos and images of pages based on standalone scenes (i.e. not transition logic based scenes).

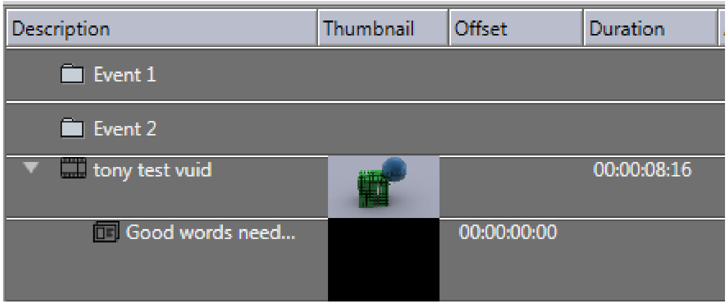
- **Host name**: Host name of the rendering engine. If connected the different post rendering options are made available.
- **Plugin**: Sets the rendering engine to generate either video or images.
  - Available options are: Video-renderer, JpegRenderer, PNGRenderer, QuicktimeRenderer, TgaRenderer, and TiffRenderer.
  - The Video-renderer option creates one video file whereas any of the image rendering options create multiple images (based on the configured frame rate).
- **Format**: Sets the output format. Formats vary depending on the option selected.
  - Available formats are: *Full Frame, Fields Top, Fields Bottom, Full Frame Skip, Full Frame/ Interlaced Top*, and *Full Frame/ Interlaced Bottom*.
  - The Video-renderer option supports all formats, whereas the image rendering options do not support the interlace (top/bottom) formats.
- **Codec**: Sets the codec device to be used. The codec option is available only for creating videos. If the plugin is set to image renderer, this option is unavailable.
- **FPS**: Sets the numbered of frames to be rendered per second. This option is also only available for the Video-renderer option.
  - Available frame rates are: *25, 30, 50, 60, 29.97, 59.94*.
  - Setting a frame rate for image rendering is not available. Default is 25 FPS. Hence, rendering a scene for five seconds will result in 250 images.
- **Output Path**: Requires a full path where the created file will be saved. Default path is, `%Program Files%\vizrt\Viz3\trioclip`.
- **From**: Sets the start time in seconds.
- **To**: Sets the end time in seconds.
- **RGB/RGBA/RGBAA**: Sets the pixel format to RGB, RGBA and RCBAA, respectively.
- **Record button (circle)**: Starts the rendering process.
- **Stop button (square)**: Stops the rendering process before the configured stop time set by the *To* field.

## 5.12 Timeline Editor



Viz Multichannel offers two types of content editors; a Page Editor and a Timeline Editor. The Timeline Editor, which this section describes, allows you to edit secondary events such as video clips and video clips with embedded graphics.

### 5.12.1 To Add Graphics to the Video Timeline



| Description | Thumbnail | Offset | Duration |
|---|---|---|---|
| Event 1 | | | |
| Event 2 | | | |
| tony test vuid | | | 00:00:08:16 |
| Good words need... | | 00:00:00:00 | |

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, they will not show up 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.

### 5.12.2 To 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 from the appearing context menu select **Remove**.

**See Also**

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

## 5.13 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.

### 5.13.1 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**

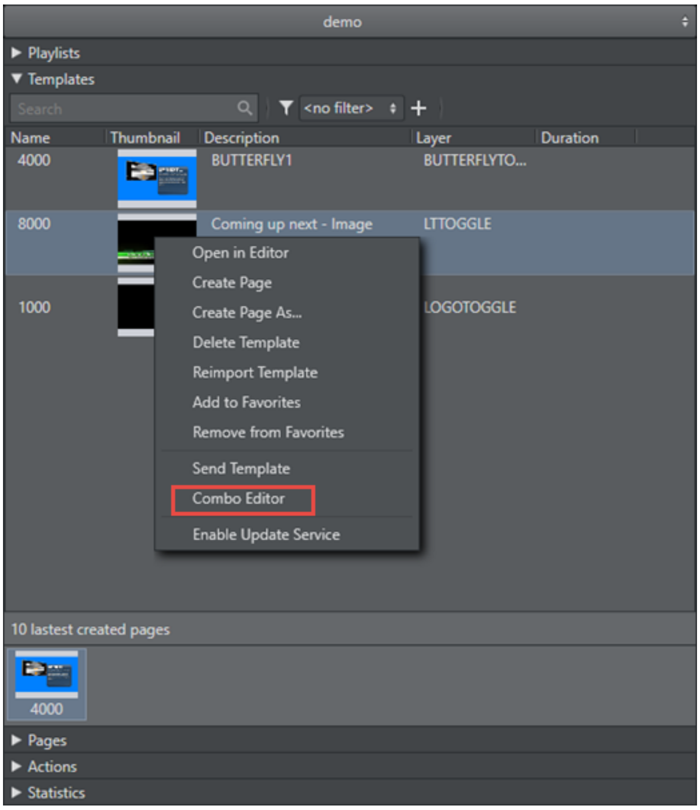
- A Transition Logic scene is not a single scene, but a set of Viz graphic scenes that consist of a background scene that may have multiple layers of graphics that can be on air at the same time and independently controlled.
- Each layer in the background scene may have multiple referring foreground 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.

## 5.13.2 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.

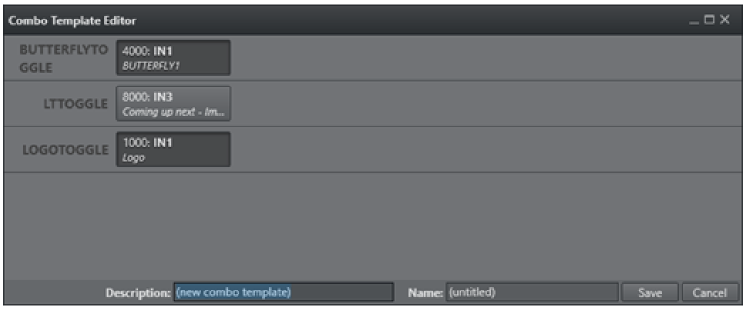
To create a combo template:

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



| Name | Thumbnail | Description | Layer | Duration |
|---|---|---|---|---|
| 4000 | | BUTTERFLY1 | BUTTERFLYTO... | |
| 8000 | | Coming up next - Image | LTTOGGLE | |
| 1000 | | | LOGOTOGGLE | |

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 selectable boxes.



| | |
|---|---|
| BUTTERFLYTOGGLE | 4000: IN1<br>BUTTERFLY1 |
| LTTOGGLE | 8000: IN3<br>Coming up next - Im... |
| LOGOTOGGLE | 1000: IN1<br>Logo |

Description: (new combo template) Name: (untitled)

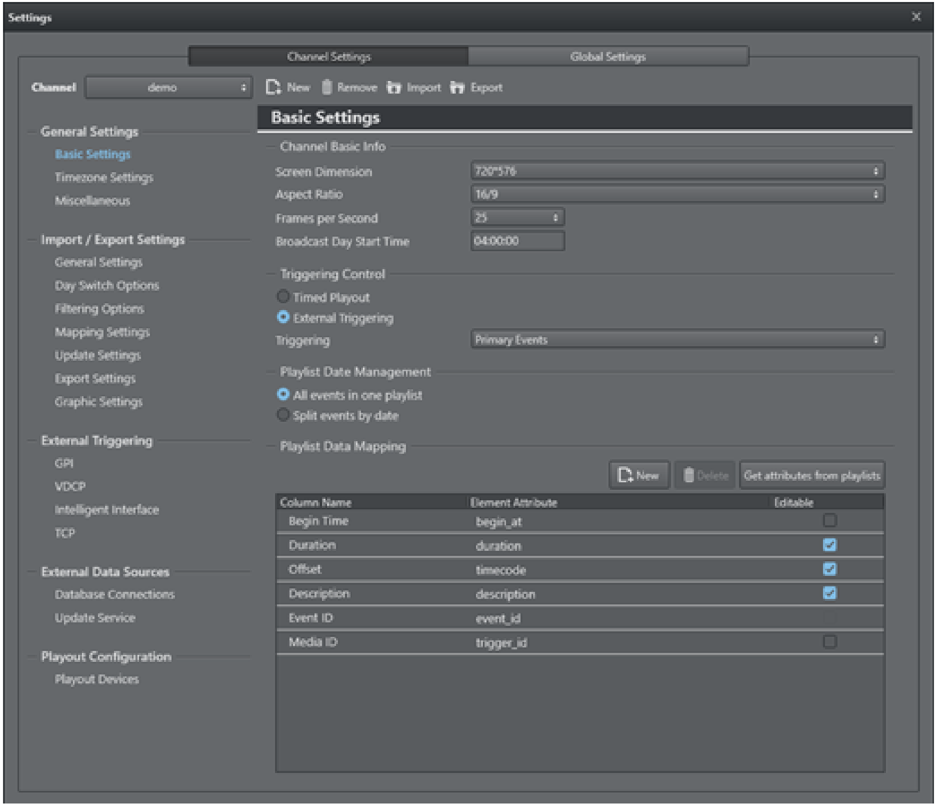
1. In the **Combo Template Editor**, select the TL state that should be activated when pages based on the TL template are shown on air.
Selected states are highlighted in orange.
2. Give the new combo template a text description and a unique name.

3. Press **Save**.

# 6 Configuration Interface

Viz Multichannel settings are split in two distinct categories: **Global** and **Channels** where *channels* are the various channels that have been defined by the user in Viz Multichannel.

- **Global settings** apply to all channels.
- **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 your channel settings, ensure that you have selected the required channel beforehand. Most settings are set using the **Settings** window.



Channel: demo

Screen Dimension: 720*576

Aspect Ratio: 16/9

Frames per Second: 25

Broadcast Day Start Time: 04:00:00

○ Timed Playout

● External Triggering

Triggering: Primary Events

● All events in one playlist

○ Split events by date

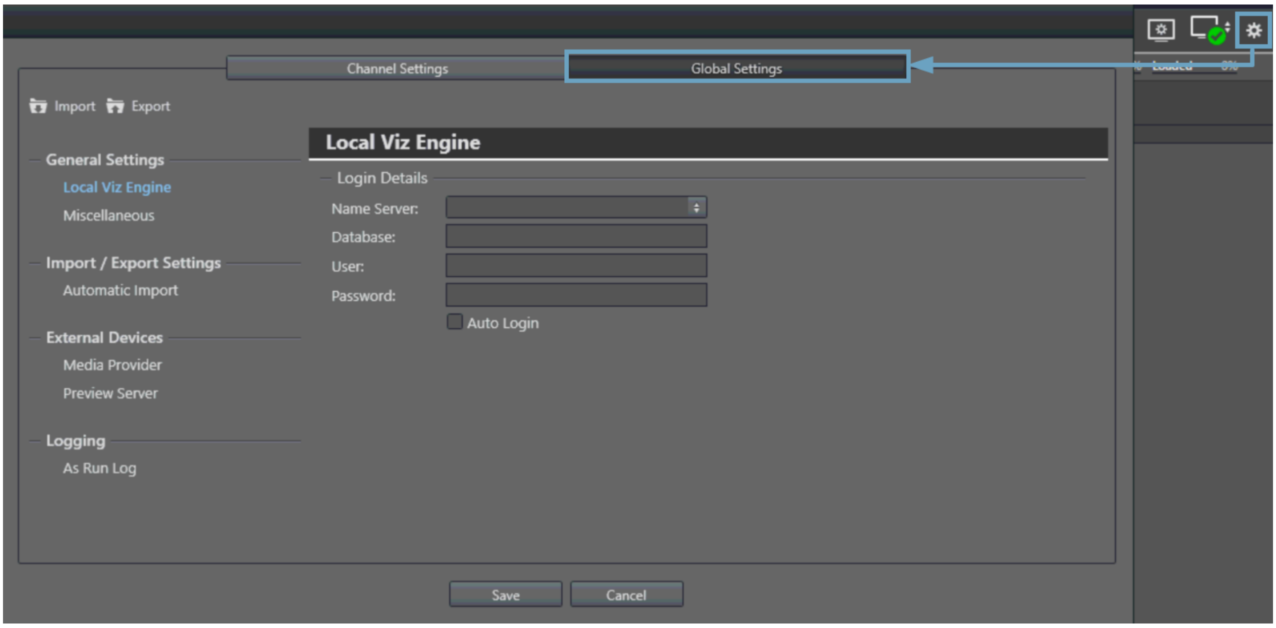
| Column Name | Element Attribute | Editable |
|---|---|---|
| Begin Time | begin_at | ☐ |
| Duration | duration | ☑ |
| Offset | timecode | ☑ |
| Description | description | ☑ |
| Event ID | event_id | |
| Media ID | trigger_id | ☐ |

This section presents the following topics:

- Global Settings
- Local Settings
- Working with Channel Settings
- Channel Settings
- Video Configuration
- Playout Configuration

## 6.1 Global Settings

In addition to Channel Settings Viz Multichannel has 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

## 6.1.1 General settings

### Local Viz Engine

The local Viz Engine will usually run on the same machine as the client application. In this settings section you provide Viz Multichannel with the information of the connection and authorization parameters (see screenshot above).

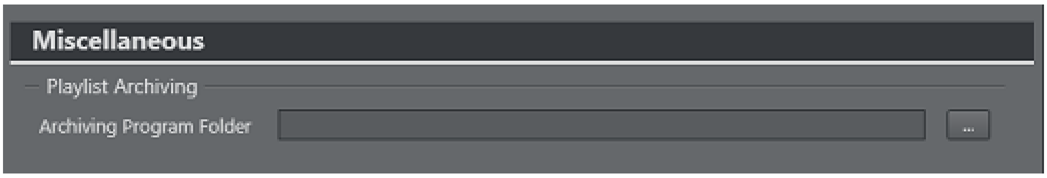
**Name server**: <viz engine server name> Can be localhost or a different PC where viz engine installed.

**Database**: Graphics Hub (GH) database name.

**User** + **Password**: GH database user and password.

**Auto Login** checkbox: Next time system will not ask for user name/password, instead system will use stored credentials for log-in.

### Miscellaneous



Miscellaneous

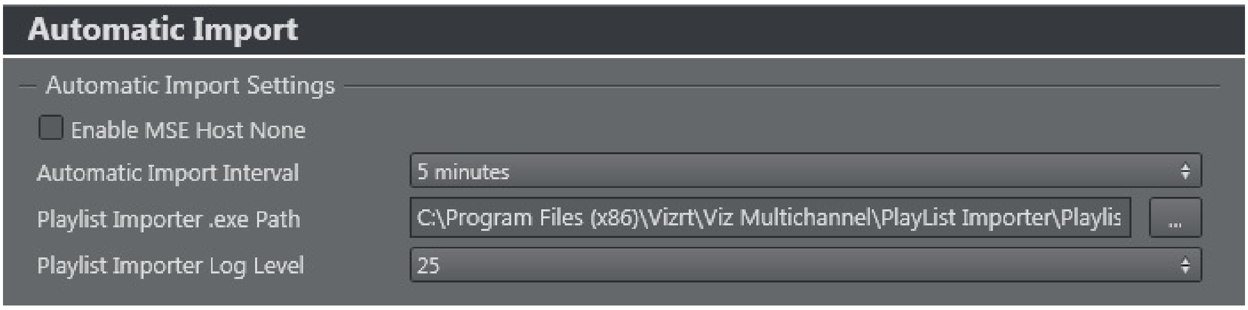
Playlist Archiving

Archiving Program Folder

Archiving Program Folder: Location of the Playlist History Manger (PLHM).
The default location is `C:\Program Files (x86)\Vizrt\VizMultichannel\Tools`

## 6.1.2 Import / Export Settings

### Automatic Import



Automatic Import

Automatic Import Settings

□ Enable MSE Host None

Automatic Import Interval: 5 minutes

Playlist Importer .exe Path: C:\Program Files (x86)\Vizrt\Viz Multichannel\PlayList Importer\Playlis

Playlist Importer Log Level: 25

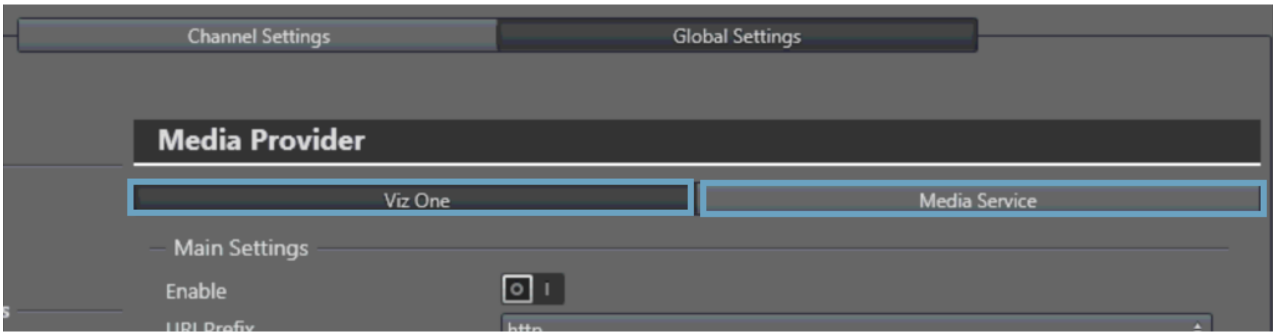
This setting is a global setting for all channels. To automatically import a schedule for a specific channel, you need to enable automatic import for *each* affected channel (see **General Settings**).

- **Enable MSE Host None** checkbox:
  *Selected*, enable automatic imports for all channel in `channel.ini` file.
  *Unselected*, disable automatic imports for all channel in `channel.ini` file
- **Automatic Import Interval**:
  Select the interval at which automatic imports of schedules will occur.
- **Playlist Importer .exe Path**: The file path, on the remote machine where the Playlist Importer is installed.
  Select `PlaylistImporter.exe` or similar batch file.
  By default, Playlist Importer is installed on the Media Sequencer machine.
- **Playlist Importer Log level**: Sets the log level of the Playlist Importer.
  See the Playlist Importer’s **Startup Parameters.**

## 6.1.3 External Devices

### Media Provider

You can select from either *Viz One* or *Media Service* as a provider of media assets.



Channel Settings | Global Settings

Media Provider

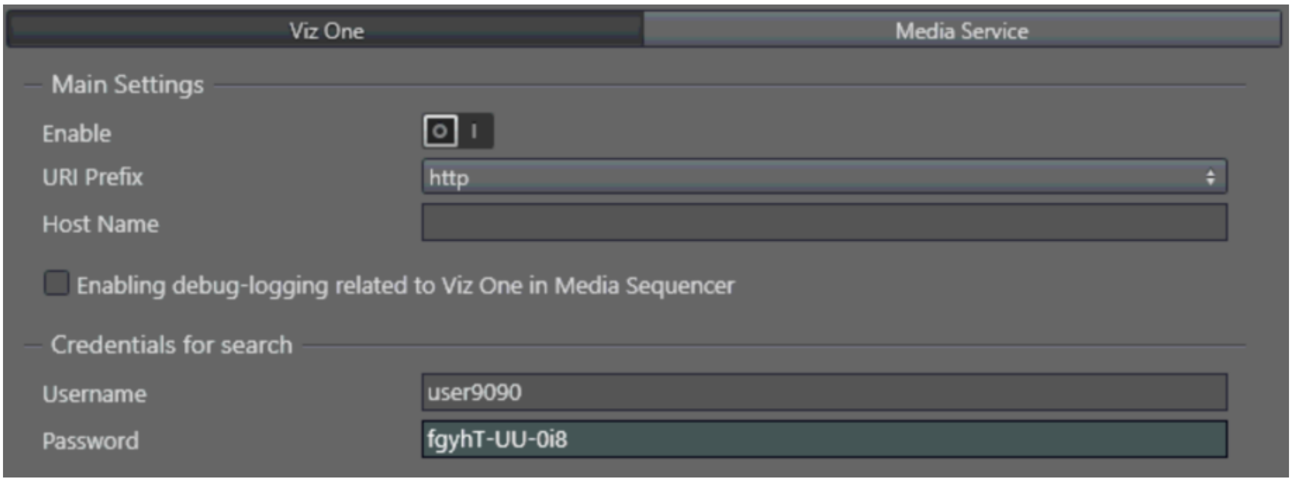
Viz One | Media Service

Main Settings

Enable

URI Prefix: http

### Viz One



Viz One | Media Service

Main Settings

Enable

URI Prefix: http

Host Name:

□ Enabling debug-logging related to Viz One in Media Sequencer

Credentials for search

Username: user9090

Password: fgyhT-UU-0i8

These global Viz One settings are used by *all* channels sharing the same Media Sequencer, defining the Viz One connection to the Media Sequencer.
Connected to Viz One, Media Sequencer can then request transfer of media stored in Viz One, 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 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.

- **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.

## To Configure a Viz One Connection

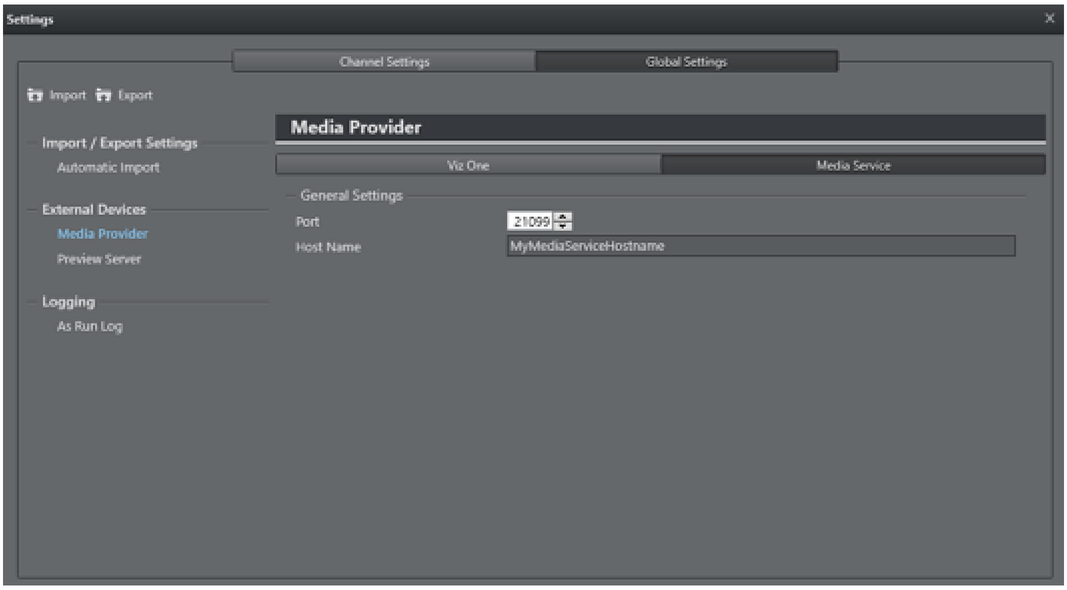
1. Make sure that **Viz One** is enabled
2. Enter the Viz One **host name**
3. Enter **Username** and **Password**
4. Click **Save**.
   You can now select the **Video Clips** options and search for a clip in Viz One (to verify the connection).

### See Also

- Viz One Integration

### Media Service

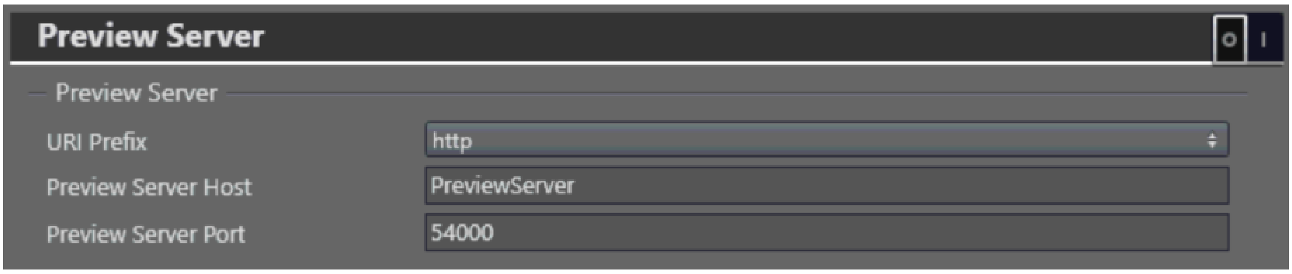
Media Provider is a simple REST-based service enabling Multichannel to search and use video resources from a Viz Engine clip folder. The Media Provider standard TCP/IP port is **21099**. The port-number is configurable and must match the IP port used by the Media Service server.



Port: 21099

Host Name: MyMediaServiceHostname

## Preview Server



URI Prefix: http

Preview Server Host: PreviewServer

Preview Server Port: 54000

With the Preview Server, Viz Multichannel can request frames of playlist elements that create a visual references in the playlist or when adding graphics to secondary event video clips.

This configuration requires a running Preview Server. Note that the Preview Server setup must be installed on a *separate* machine.

The Preview Server itself 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 has an On/Off switch. You can configure settings with the 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 Preview Server

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

## 6.1.4 Logging

The **As Run Log** logs 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.

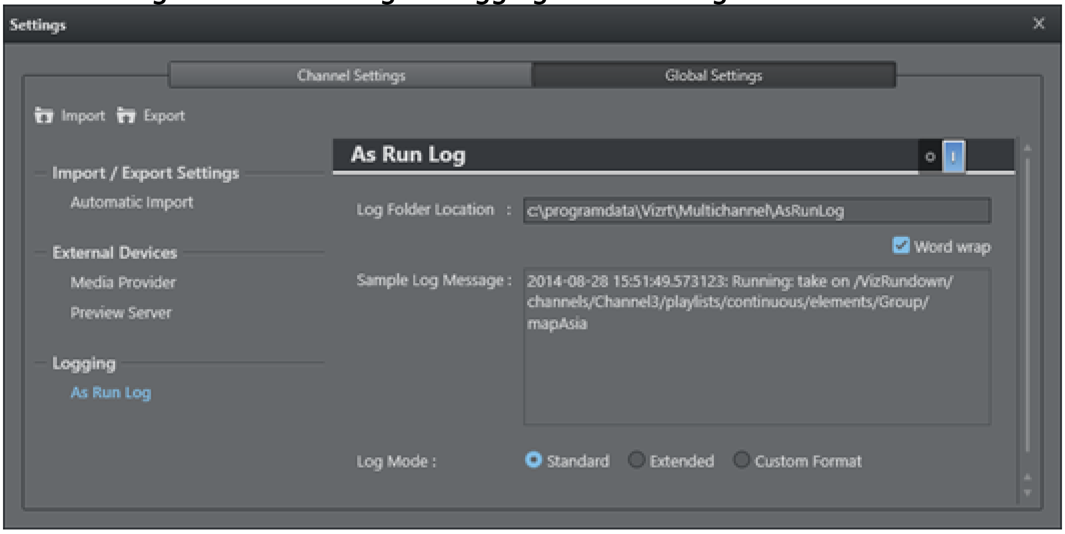
For more information (e.g. how to customize log entries), please see the *Media Sequencer* manual's As Run Log section.

⚠ **Note:** The log does not guarantee that something went On Air.

### To Enable As Run Log

When running Viz Multichannel in MCR mode, you have in most cases the Media Sequencer running on a separate machine. Most often on the same machine as your Viz Engine.

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



As Run Log

Log Folder Location : c:\programdata\Vizrt\Multichannel\AsRunLog

☑ Word wrap

Sample Log Message : 2014-08-28 15:51:49.573123: Running: take on /VizRundown/channels/Channel3/playlists/continuous/elements/Group/mapAsia

Log Mode : ● Standard ○ Extended ○ Custom Format

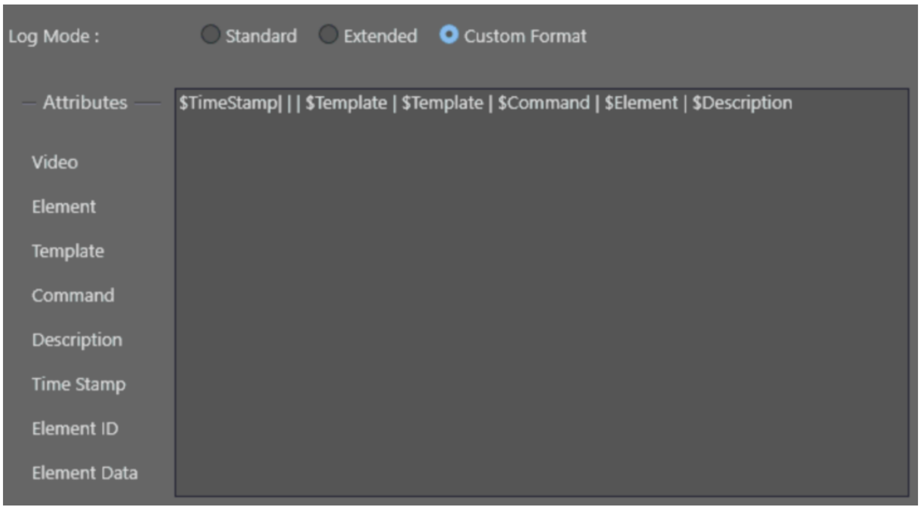
2. Select **ON** to enable logging.
3. Enter the **Log Folder Location** if required, or use the default location.
4. Select **Standard** or **Extended (JSON** or **Custom Format**).
   - (Optional) **Log Mode** is *Extended* or *Custom Format*

     Perform customization. See below.
5. Click **Save**.

### Custom Format

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

For example, you can use **Custom Format** for creating a specific As Run log format. When **Custom Format** is chosen, you can drag the available formatting keys (**Attributes**) from the left side of the panel to the **Sample Log** area to create a log-style.



Log Mode : ○ Standard ○ Extended ● Custom Format

$TimeStamp| | | $Template | $Template | $Command | $Element | $Description

### See Also

- Local Settings
- Working with Channel Settings
- Channel Settings

- Video Configuration

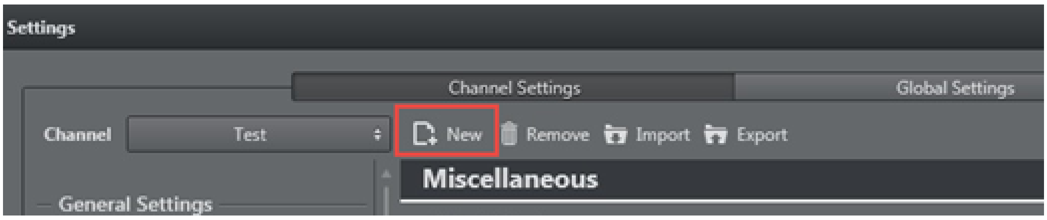
## 6.2 Working With Channel Settings

This section and its sub-sections explain how to **configure a channel.** It will give you an ordered overview of the different configuration topics that will help you configure the system more quickly.

The following sections contain information on how to **add**, **remove**, **export** or **import** channel settings:

- 
    - To Add a New Channel
    - To Remove a Channel
    - To Export a Channel as an XML File
    - To Import a Channel from an XML File

### 6.2.1 Creating a Channel



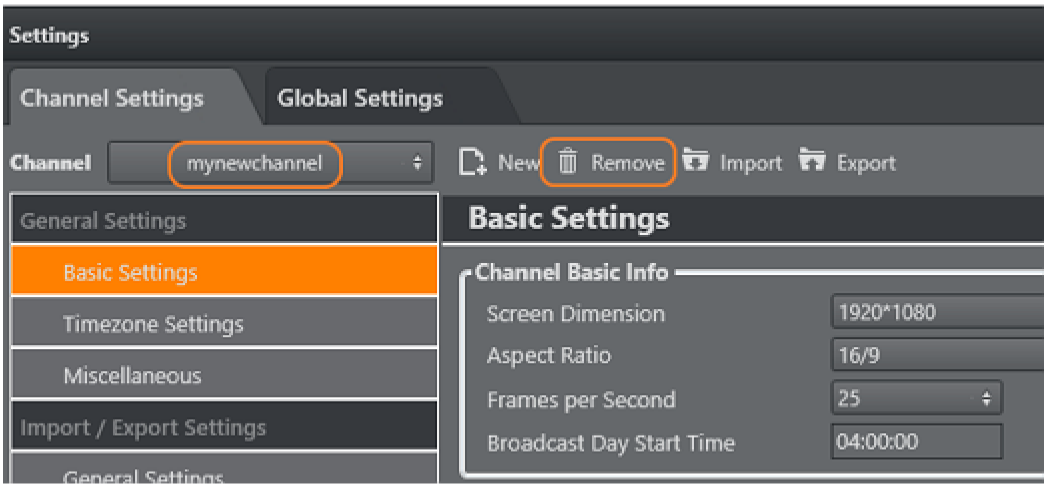
Create a new channel by pressing the **New** icon and give the channel a name in the dialog.

#### To Add 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**.
4. Proceed to configure the Channel Settings.

## 6.2.2 Removing a Channel

### To Remove a Channel



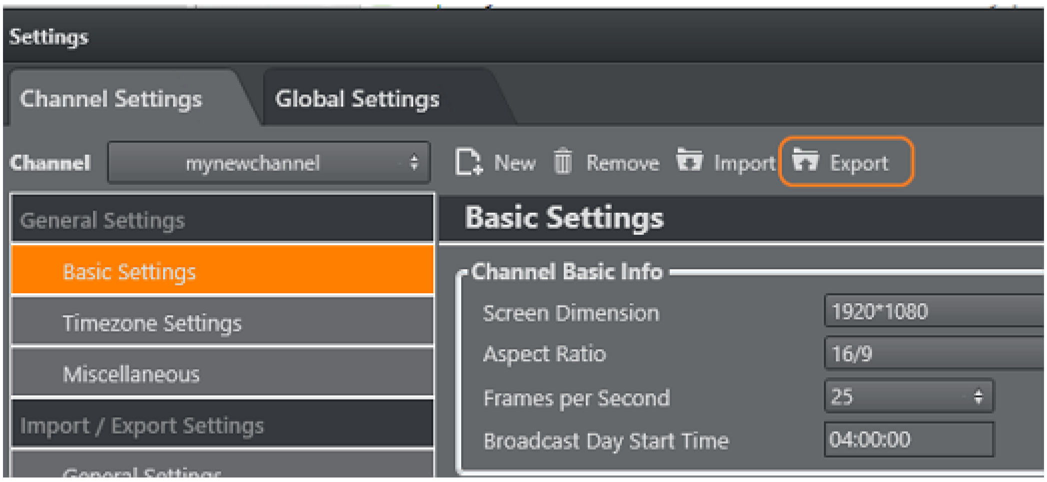
Channel: mynewchannel

Screen Dimension: 1920*1080
Aspect Ratio: 16/9
Frames per Second: 25
Broadcast Day Start Time: 04:00:00

1. Open the **Settings** window.
2. From the **Channels** combo box, select the channel you wish to remove.
3. Select **Remove**
   You are asked if you are sure you want to remove the selected channel.
4. Click **Yes**.
   The last removed channel is automatically saved to a `<ChannelName>.bkp` file, which can later be re-imported back into Viz Multichannel. The file is located in the **backup channel** folder found in the Viz Multichannel program folder.

## 6.2.3 Exporting a Channel

### To Export a Channel as an XML File



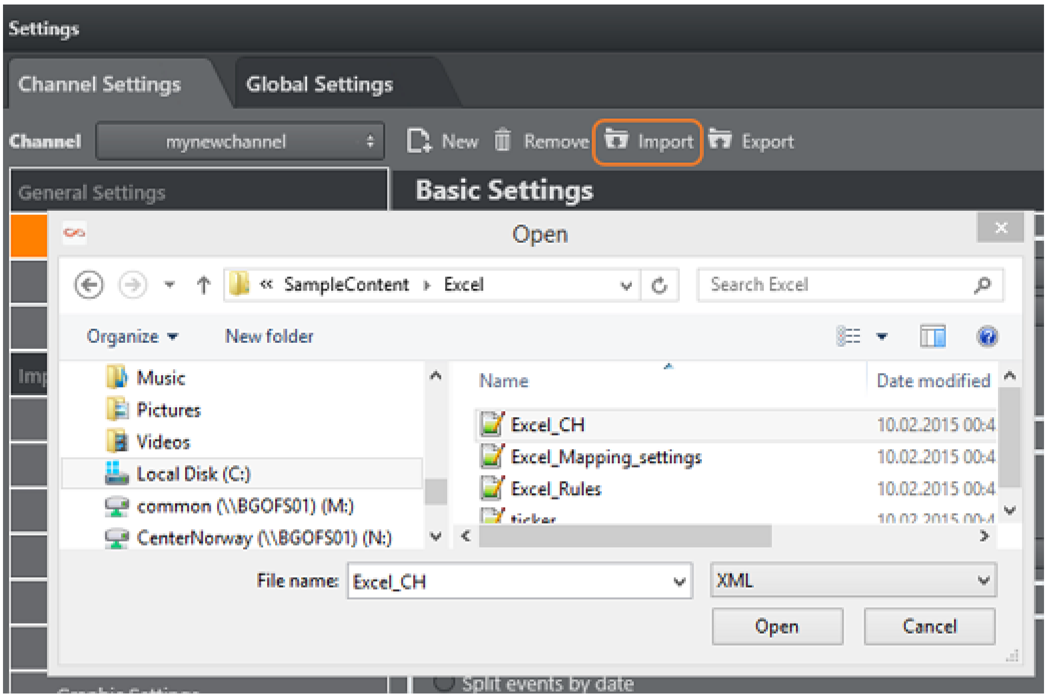
Channel: mynewchannel

Screen Dimension: 1920*1080
Aspect Ratio: 16/9
Frames per Second: 25
Broadcast Day Start Time: 04:00:00

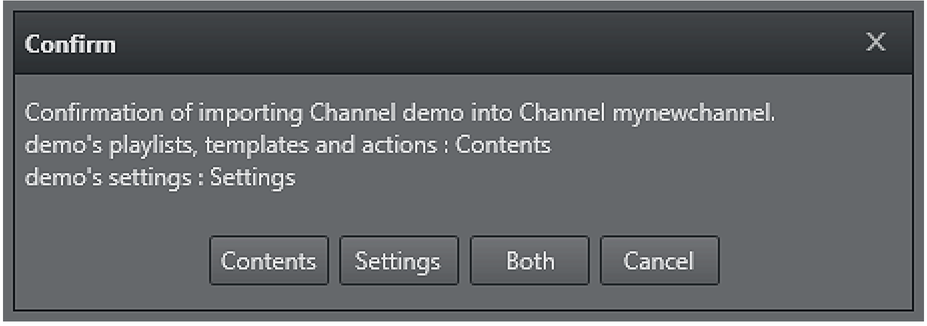
1. Open the **Settings** window.
2. From the **Channels** combo box, select the channel you wish to export.
3. Select **Export**.
   You are asked if you want to include the playlists and templates along with the selected channel’s settings.
4. Click **Yes** or **No**, as required.
5. Enter a file name and click **Save**.

## 6.2.4 Importing a Channel

### To Import a Channel from an XML File



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



Confirmation of importing Channel demo into Channel mynewchannel.
demo's playlists, templates and actions : Contents
demo's settings : Settings

4. 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.

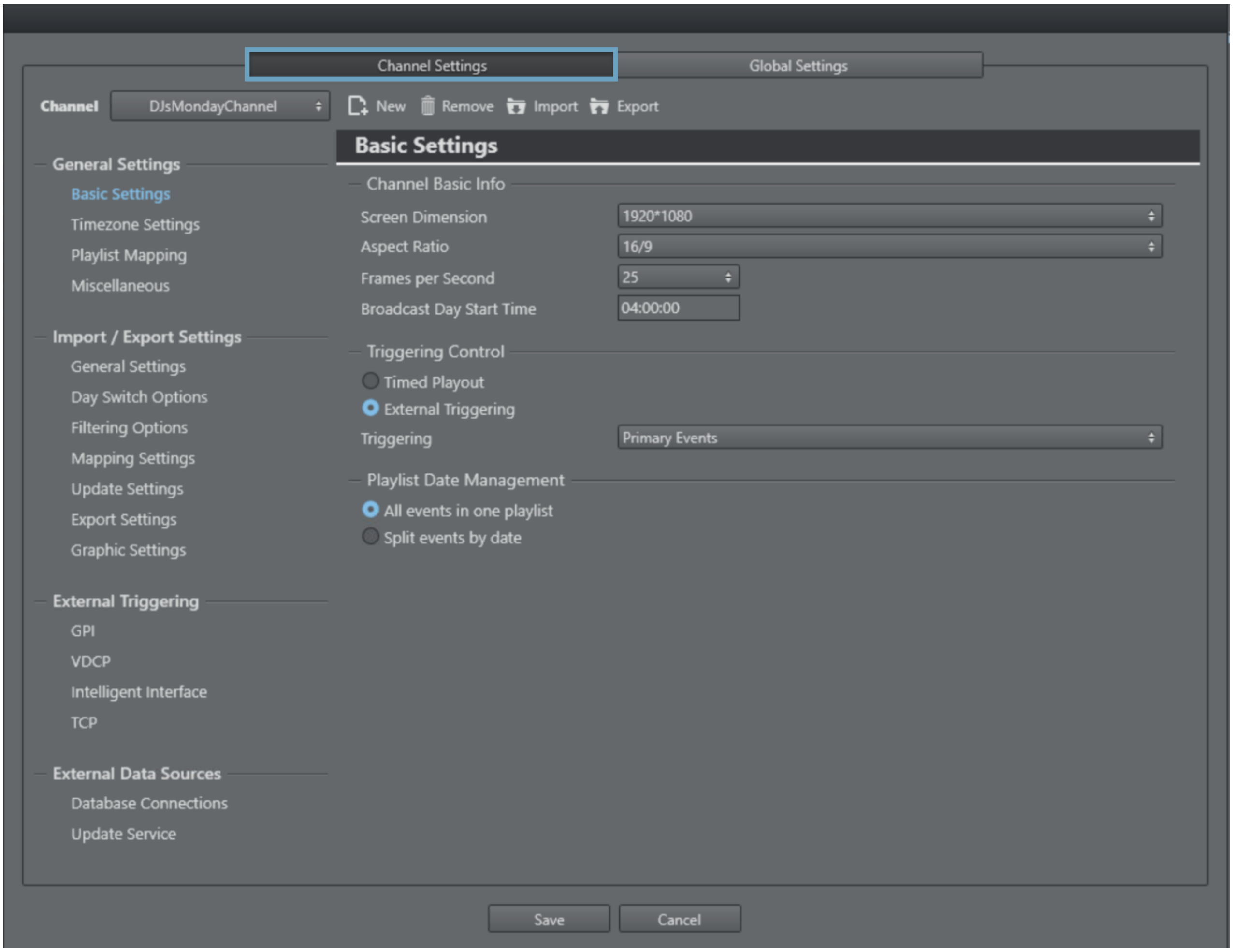
**See Also**

- Global Settings
- Local Settings
- Channel Settings
- Video Configuration

---

## 6.3 Channel Settings

In addition to Global Settings, Viz Multichannel has channel-specific settings, affecting a single channel.



Channel: DJsMondayChannel

**Basic Settings**

Channel Basic Info

| | |
|---|---|
| Screen Dimension | 1920*1080 |
| Aspect Ratio | 16/9 |
| Frames per Second | 25 |
| Broadcast Day Start Time | 04:00:00 |

Triggering Control

○ Timed Playout
● External Triggering

| | |
|---|---|
| Triggering | Primary Events |

Playlist Date Management

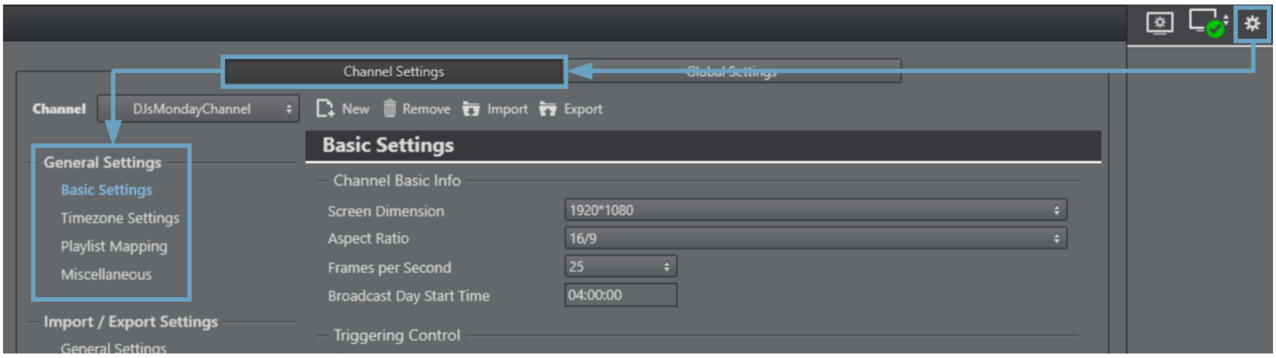
● All events in one playlist
○ Split events by date

Each of the settings groups are presented below:

- General Settings
- Import / Export Settings
- External Triggering
- External Data Sources

## 6.3.1 General Settings

You locate channel-specific general settings from the cog wheel icon, and selecting the **Channel settings** tab.



Screen Dimension 1920*1080

Aspect Ratio 16/9

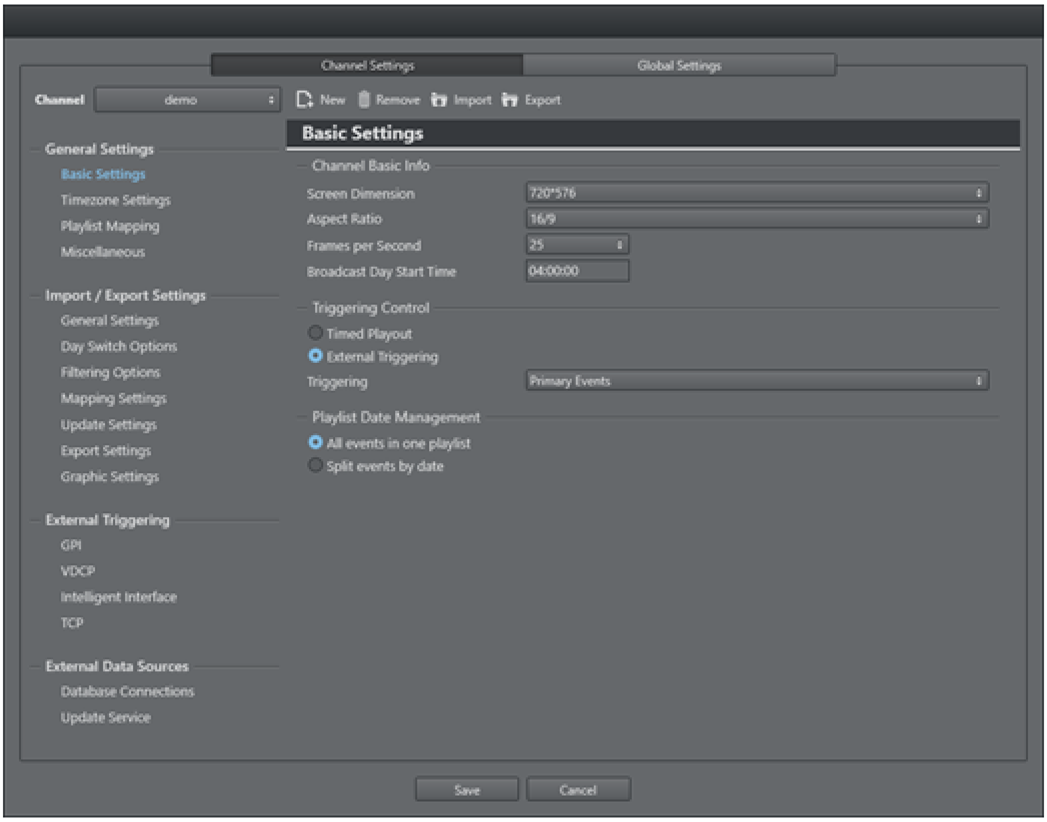
Frames per Second 25

Broadcast Day Start Time 04:00:00

The **General Settings** group comprises:

- Basic Settings
- Timezone Settings
- Playlist Mapping
- Miscellaneous

## Basic Settings



Channel: demo

Basic Settings

Channel Basic Info

Screen Dimension: 720*576

Aspect Ratio: 16/9

Frames per Second: 25

Broadcast Day Start Time: 04:00:00

Triggering Control

○ Timed Playout

● External Triggering

Triggering: Primary Events

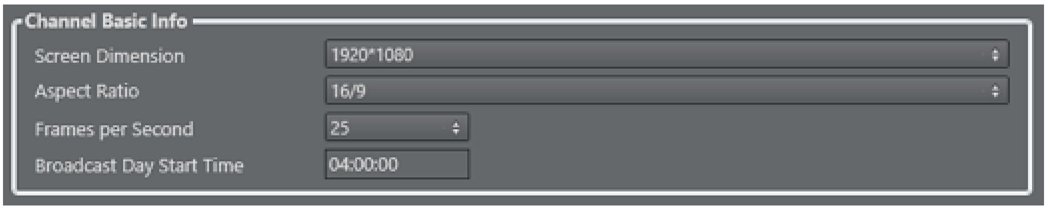
Playlist Date Management

● All events in one playlist

○ Split events by date

- Channel Basic Info
- Triggering Control
- Playlist Date Management

## Channel Basic Info


Channel Basic Info

Screen Dimension: 1920*1080

Aspect Ratio: 16/9

Frames per Second: 25

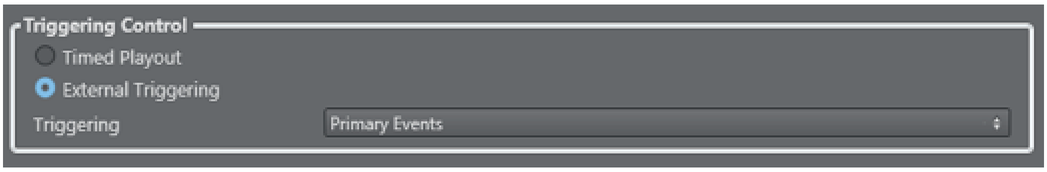
Broadcast Day Start Time: 04:00:00

Format settings are most relevant for Viz Multichannel in master control room (MCR) mode.

- **Screen Dimension**: Defines the screen dimension. Options are:
  - SD NTSC: 720 x 486
  - SD PAL/SECAM: 720 x 576
  - HD 720: 1280 x 720
  - HD 1080: 1920 x 1080 (current the default value, user changeable)
- **Aspect Ratio**: Defines the aspect ratio. Options are 4:3 and 16:9. This setting also affects the aspect ratio of the template editor.

- **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 secondary events only (i.e. when an external system controls the primary events). In such situations it is important that this setting is correct, or else 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

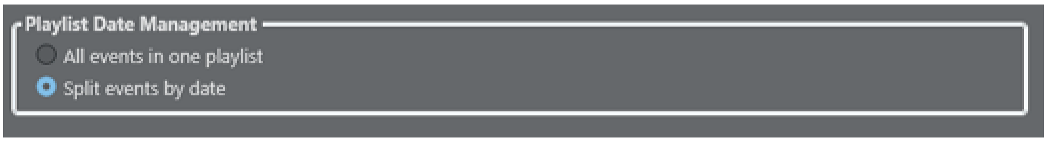


Triggering Control
○ Timed Playout
● External Triggering
Triggering: Primary Events

Triggering control is only relevant for Viz Multichannel in master control room (MCR) mode.

- **Timed Playout**: Allows the **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.
  - If the **primary events** option is selected, the Media Sequencer will trigger the secondary events based on offsets to the timed primary events.
  - If the **secondary and video events** option is selected, the automation system is in full control of the playlist and the Media Sequencer will act as a slave. Note that this using this option and assets from Vizrt's Viz One system as Primary Events, the effect is that the video of the primary Primary and all other Secondary types can be played out in combination and controlled from Automation.

## Playlist Date Management



Playlist Date Management
○ All events in one playlist
● Split events by date

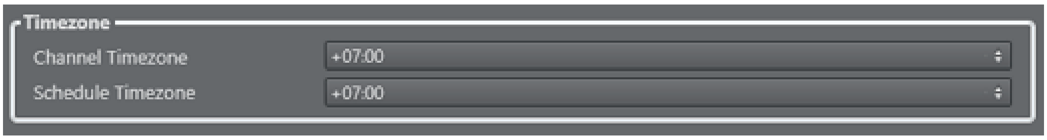
- **All events in one playlist**: When selected, Viz Multichannel will only allow one continuous playlist exist per channel (recommended). Regardless of time and on-air date, all events are included in the imported playlist.
- **Split events by date**: Schedules received from automation systems *are split by date*. This means that Viz Multichannel usually gets one long sequence of events that may belong to multiple playlists, typically for the current and the next day. Selecting this option causes Viz Multichannel to separate the events based on date and begin time, according to the **Broadcast Day Start Time** settings.

> ⚠ **Note:** The **Split events by date** option is only valid when reading the schedule's date from within the source file contents (and not the file name). If the `filedate` field is mapped to read values from `filename (=` **$Filename**`)`, this feature will not work.

## Timezone Settings

- Timezone
- DST Setting
- Dynamic DST Setting
- Explicitly defined DST Setting

### Timezone

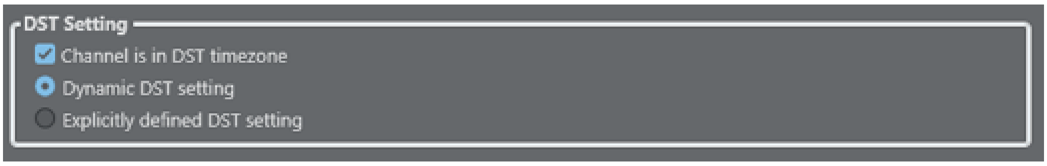


Timezone

| | |
|---|---|
| Channel Timezone | +07:00 |
| Schedule Timezone | +07:00 |

- **Channel Timezone**: Defines the timezone of the area the channel is being broadcasted into. Options are: -12:00 to +14:00.
- **Schedule Timezone**: Defines the timezone of the schedule being imported.

### DST Setting



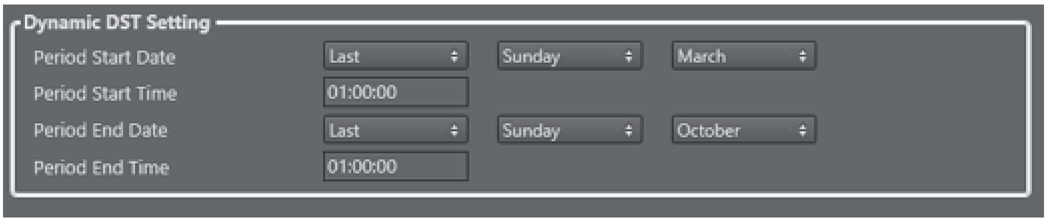
DST Setting

☑ Channel is in DST timezone
● Dynamic DST setting
○ Explicitly defined DST setting

- **Channel is in DST timezone**: Check this option when you want to enable the Day-light saving function and select one of the Day-light saving modes:
  - Dynamic DST Setting
  - Explicitly defined DST Setting

#### Dynamic DST Setting

This configuration panel opens when **DST Setting > Channel is in DST timezone** (above) is set to *Dynamic DST setting.*



Dynamic DST Setting

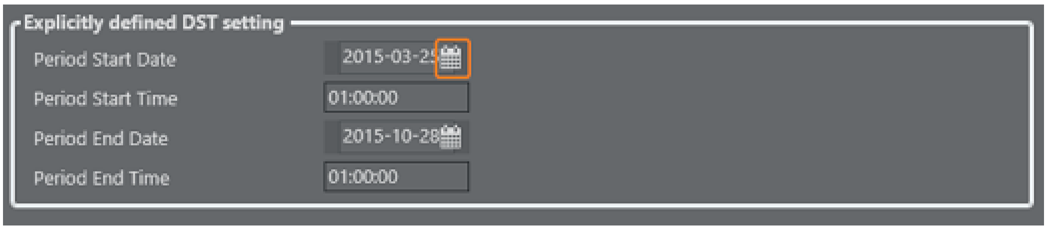
| | | | |
|---|---|---|---|
| Period Start Date | Last | Sunday | March |
| Period Start Time | 01:00:00 | | |
| Period End Date | Last | Sunday | October |
| Period End Time | 01:00:00 | | |

- **Period Start Date**: Specifies the *date* that starts the Day-light saving period.
- **Period Start Time**: Specifies the *time* that starts the Day-light saving period.
- **Period End Date**: Specifies the *date* that ends the Day-light saving period.
- **Period End Time**: Specifies the *time* in local non- DST time that ends the Day-light saving period.

> **Note:** Invert months for southern hemisphere settings.

### Explicitly Defined DST Setting

This configuration panel opens when **DST Setting > Channel is in DST timezone** (above) is set to *Explicitly defined DST setting*



Explicitly defined DST setting

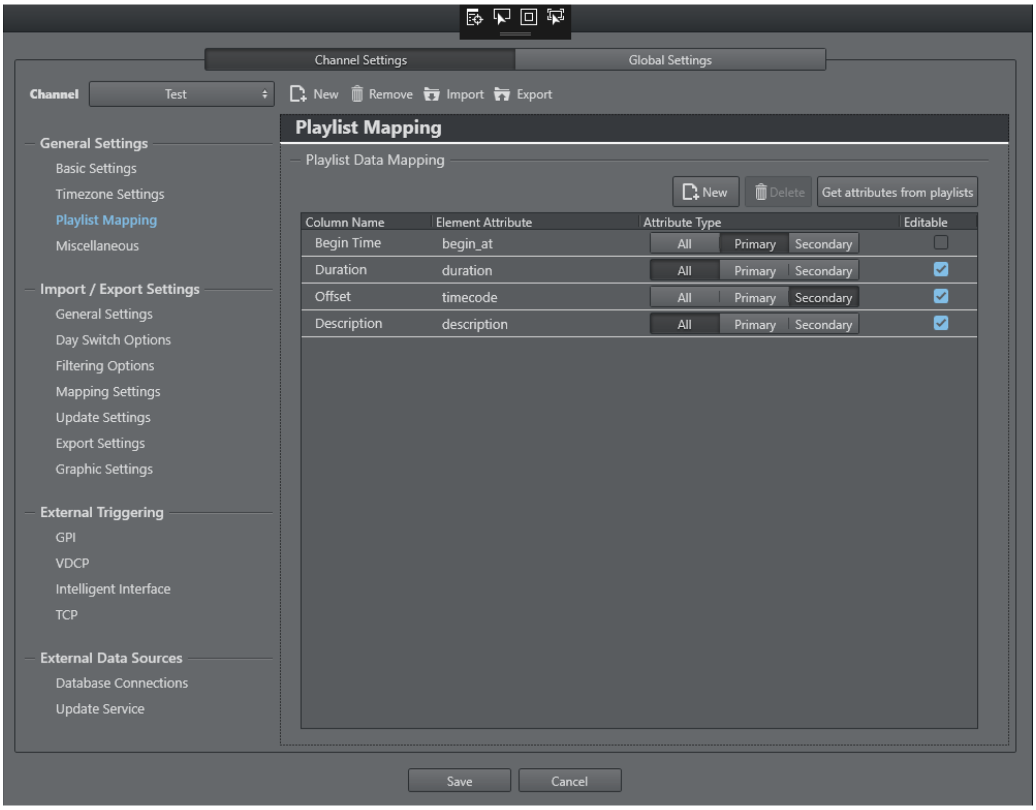
| | |
|---|---|
| Period Start Date | 2015-03-29 |
| Period Start Time | 01:00:00 |
| Period End Date | 2015-10-28 |
| Period End Time | 01:00:00 |

> **Note:** For nations with special DST handling. Press the calendar icon to selected dates from a popup.

- **Period Start Date**: Specifies the *date* that starts the Day-light saving period
- **Period Start Time**: Specifies the *time* that starts the Day-light saving period
- **Period End Date**: Specifies the *date* that ends the Day-light saving period
- **Period End Time**: Specifies the *time* in local non-DST time that ends the Day-light saving period

## Playlist Mapping

Playlist mappings define how the current playlist (rundown) columns are mapped to attributes.



Playlist Mapping

Playlist Data Mapping

| Column Name | Element Attribute | Attribute Type | Editable |
|---|---|---|---|
| Begin Time | begin_at | All Primary Secondary | ☐ |
| Duration | duration | All Primary Secondary | ☑ |
| Offset | timecode | All Primary Secondary | ☑ |
| Description | description | All Primary Secondary | ☑ |

- **Column Name**: Specify what attribute name will be shown on the playlist.

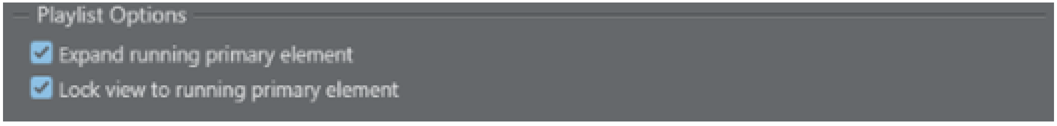
- **Element Attribute**: All attributes in the **Element Attribute** column are used in Page Content Filling Dialog, Rules condition and Filtering Options in **Import/Export Settings**.
- **Attribute Type**: Indicates whether an attribute belongs to primary event, secondary event or both.
- **Get attributes from playlist**: Examine attributes from 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 a blank **Column Name.** If you want to show this attribute in the playlist column (option **Use External Data** menu when right-clicking on playlist column), you have to first fill in a **Column Name**

## Miscellaneous

- Playlist options
- General Playout Options
- Edit Options
- Scene Import
- Playlist Archiving
- Viz One Integration
- Default Scenes
- Reporting

## 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 (i.e. primary element) to the top whenever the primary element is changed.



Playlist Options

☑ Expand running primary element

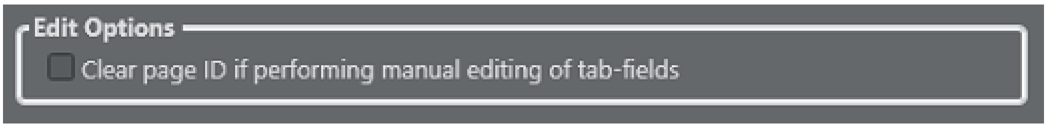
☑ Lock view to running primary element

## General Playout Options

- **Allow item replay**: When checked the automation system requests the replay of an item in a playlist (same Media ID), Viz Multichannel will go back and play the requested item (even if marked as played). When unchecked, it prevents an event from being played twice, but looks for the next event with the requested ID marked as not played.

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

## Edit Options

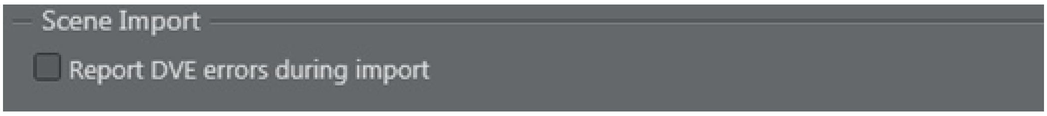


Edit Options

☐ Clear page ID if performing manual editing of tab-fields

- **Clear ID after editing secondary event**: When checked the secondary event ID is removed from the edited element in the playlist. This allows you to receive an updated schedule without overwriting already edited secondary events in your playlist.

## Scene Import



Scene Import

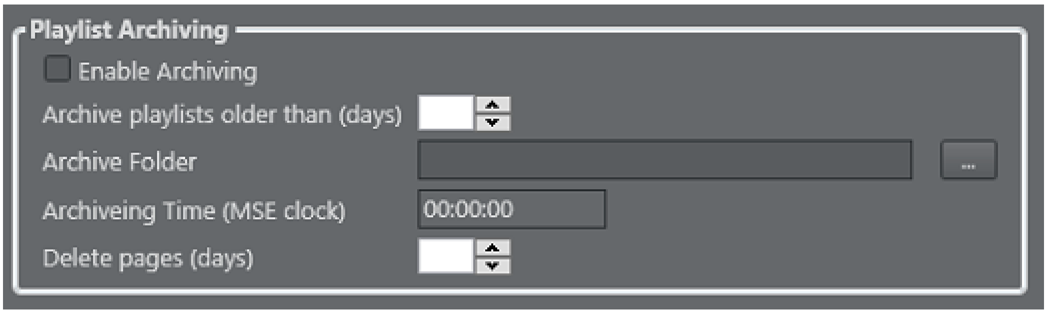
□ Report DVE errors during import

- **Report DVE errors during import**: When checked, if in the Basic Settings you selected to *Play Video Through the Viz Engine*, any Viz Artist scenes being imported as Viz Multichannel templates that were built without DVE support will be reported. Therefore, if you selected to play video through the Viz Engine, it is recommended that you select this option, since DVE support is important.
  If you do not play video through the Viz Engine, you can leave this setting unchecked, or if you do not wish to see this error highlighted.

## Playlist Archiving

Playlist archiving is used to archive and delete old playlists and pages, keeping only relevant data in the system.

These settings are only applicable for playlists that are split on days, and *not* for a continuous playlist.



Playlist Archiving

□ Enable Archiving

Archive playlists older than (days)

Archive Folder

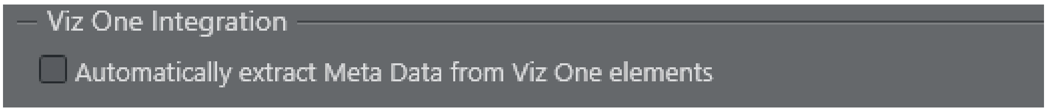
Archiveing Time (MSE clock) 00:00:00

Delete pages (days)

- **Enable Archiving**: When checked, archiving is enabled. When unchecked, no archiving will occur. Archiving will remove old playlists from the system and archive them in files.
- **Archive playlists older than (days)**: Enter the number of days after which a playlist will be archived.
  Example: If today is 20/07/2014 and playlists are older than 2 days, playlists dated 17/07/2014 and older are archived.
- **Archive Folder**: Enter the folder path in which older playlists are archived.
- **Archiving Time (MSE clock)**: Enter the time at which archiving will take place each day. The action will occur according to the clock of the machine on which Media Sequencer is installed.
  Example: If set to 00:00:00, Playlist History Manager will run every midnight.
- **Delete pages (days)**: Enter the number of days after which pages will be deleted from the channel's storage.

Example: If today is 20/07/2014 and pages are older than 2 days, pages modified on date 17/07/2014 and older are deleted. Note, *date modified* is used, not *date created*.

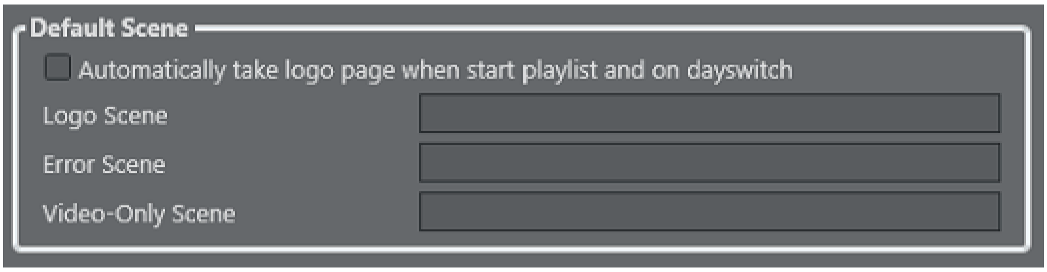
## Viz One Integration


Viz One Integration

□ Automatically extract Meta Data from Viz One elements

- **Automatically extract Meta Data from Viz One elements**: If this option is checked, Multichannel will add metadata from the Viz One item as primary event attributes while building video element when importing a schedule.

## Default Scenes


Default Scene

□ Automatically take logo page when start playlist and on dayswitch

Logo Scene

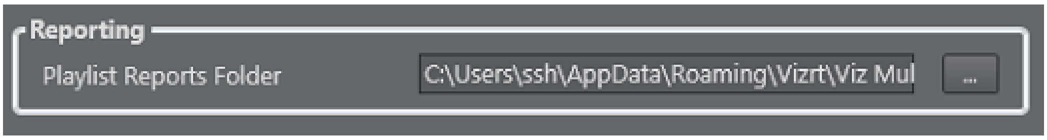
Error Scene

Video-Only Scene

- **Automatically take logo page when start playlist and on dayswitch**: Once the day switch occurs, the logo scene goes on air.
- **Logo Scene**: Enter the scene name of the logo scene that will be played as necessary.
- **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 in the Basic Settings you selected to *Play Video Through the Viz Engine*, enter the scene name of the video-only scene that will be played when Viz Multichannel has no templates or playlists in memory. If you have a Matrox X.mio 2 plus card this is not needed.

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

## Reporting

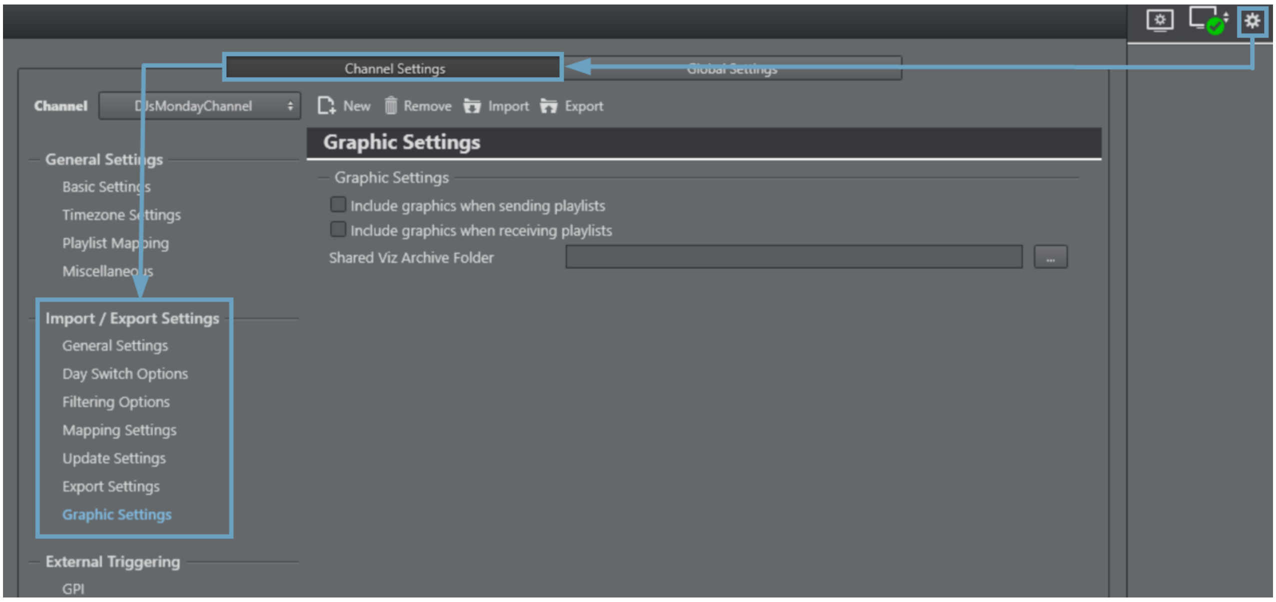

Reporting

Playlist Reports Folder C:\Users\ssh\AppData\Roaming\Vizrt\Viz Mul

- **Playlist Reports Folders**: Enter the folder path where playlist reports are saved. For more information on how to generate reports, see the Channel and Playlist Reports section.

## 6.3.2 Import / Export Settings

You locate channel-specific import and export settings from the cog wheel icon, and selecting the **Channel settings** tab.



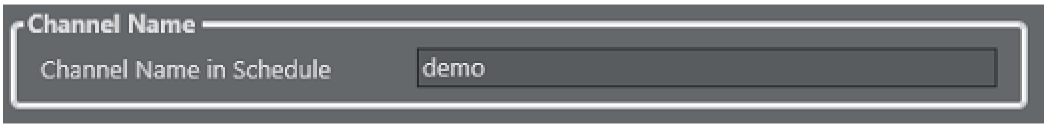
The **Import and Export** settings group comprises:

- General Settings
- Day Switch Options
- Filtering Options
- Mapping Settings
- Secondary Mapping Settings
- Update Settings
- Export Settings
- Graphic Settings

### General Settings

- Channel Name
- Remote Hosts
- Automatic Import Settings
- Import Options
- Page Content Filling
- Translation Tables

### Channel Name

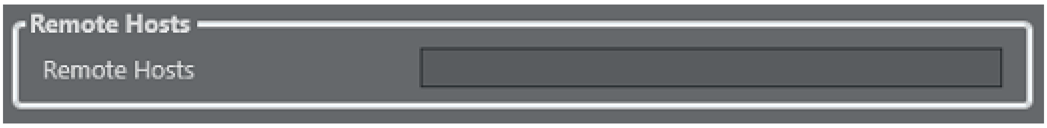


Channel Name

Channel Name in Schedule: demo

- **Channel Name in Schedule**: Sets the channel name that corresponds with the channel name given by the imported schedule. Useful if multiple channel's schedules can be found in import or update folders.

## Remote Hosts

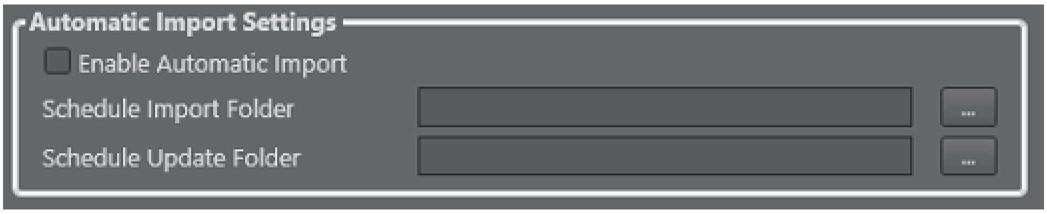


Remote Hosts

Remote Hosts

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

## Automatic Import Settings



Automatic Import Settings

□ Enable Automatic Import

Schedule Import Folder

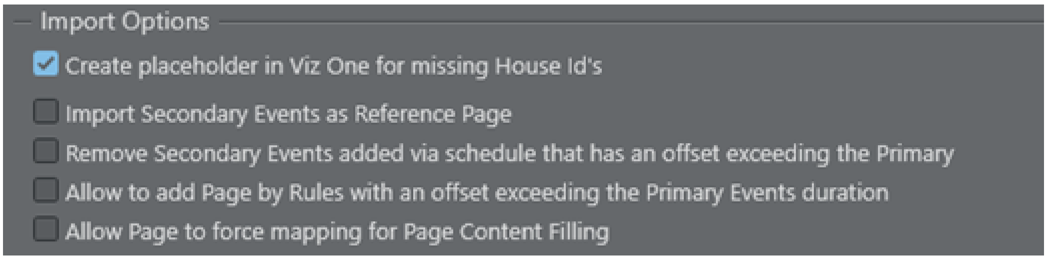
Schedule Update Folder

For information on how to set global import settings (i.e. import interval and import path), see the global Automatic Import settings. Also, see the Playlist Importer section for installation, configuration and use of the Playlist Importer.

- **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 are defined as **separate folders**.

## Import Options



Import Options

☑ Create placeholder in Viz One for missing House Id's

□ Import Secondary Events as Reference Page

□ Remove Secondary Events added via schedule that has an offset exceeding the Primary

□ Allow to add Page by Rules with an offset exceeding the Primary Events duration

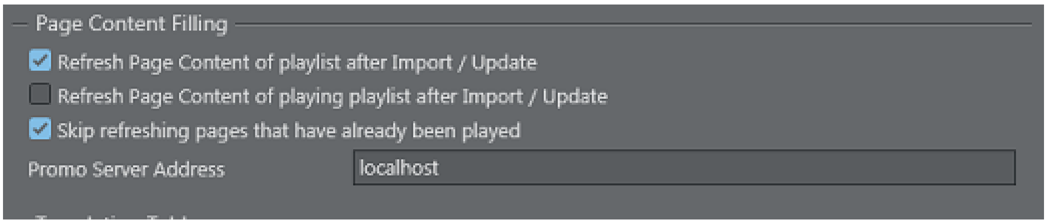
□ Allow Page to force mapping for Page Content Filling

- **Create placeholder in Viz One for missing House Ids**: If enabled then any request towards a given House Id that is not found in the Viz One system will issue a second query against Viz One to create a placeholder element (and retrieve the resulting payload for playlist composite group purposes). If placeholders are not required, disable this setting. Setting is by default enabled.

- **Import Secondary Events as Reference**: When checked, all secondary events associated to pages in Viz Multichannel will be imported as Referenced Pages. When unchecked, pages will be instances of the original page.
- Self-explanatory options:
    - Remove secondary events that has an offset exceeding the primary events duration.
    - Allow to add Page by Rules with an offset exceeding the Primary Event’s duration.
    - Allow page to force mapping for data content filling.

## Page Content Filling

In order to use Page Content Filling the Promo Server component must be installed and configured, see **Promo Server Installation and Configuration** in the Promo Server chapter.



Page Content Filling

☑ Refresh Page Content of playlist after Import / Update

☐ Refresh Page Content of playing playlist after Import / Update

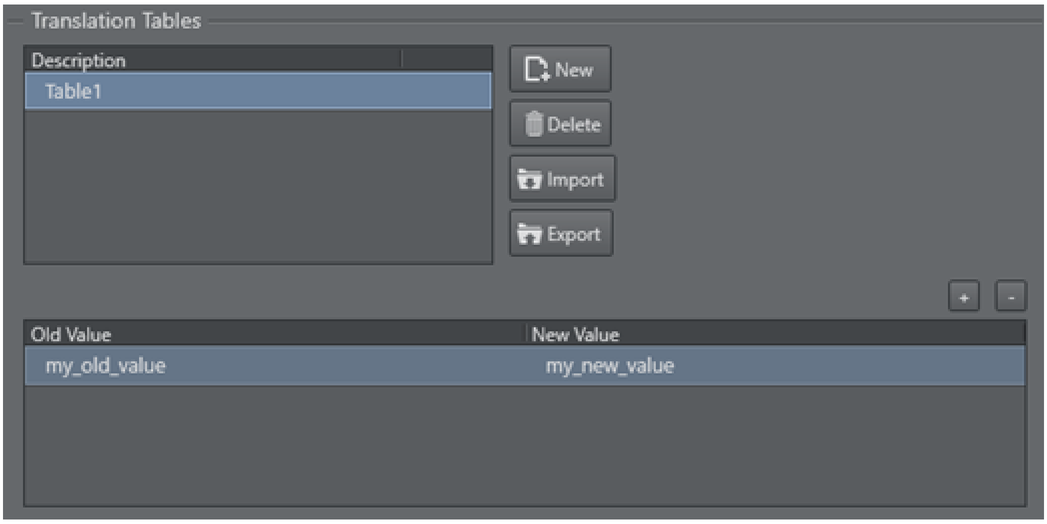
☑ Skip refreshing pages that have already been played

Promo Server Address: localhost

Options:

- Refresh Page Content of playlist after Import / Update. Is by default enabled.
- Refresh Page Content of **playing** playlist after Import / Update. Only useful if segmenting playlist per broadcast days and an update of a coming days playlist should impact the currents day's data content.
- Skip refreshing pages that have already been played: If this option is checked, the system will skip page content filling for pages that have already been played. If this option is unchecked, the system will do page content filling for all pages that has lookup rules or database connection set. The default value is 'Checked'.
- Setting defining the Promo Server's network address.

## Translation Tables

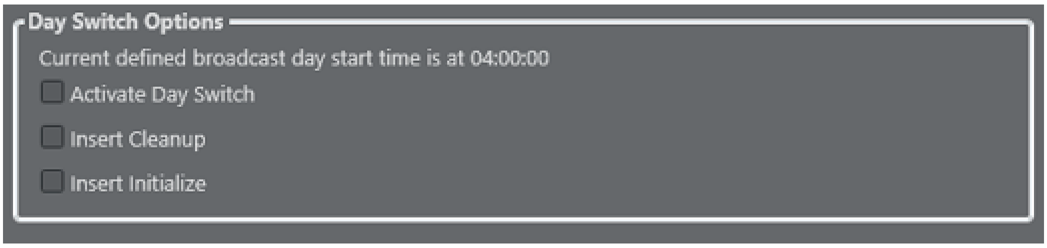


| Description |
|---|
| Table1 |

| Old Value | New Value |
|---|---|
| my_old_value | my_new_value |

Translation Tables are global for translating values when doing Content Filling. You can have several tables, but only one can be active at any given time. Use **New** to create a table and add *Old* and *New* translations as required. Use **Import** and **Export** buttons to load or save translation tables from/to XML-files.

## Day Switch Options



Current defined broadcast day start time is at 04:00:00

☐ Activate Day Switch

☐ Insert Cleanup

☐ Insert Initialize

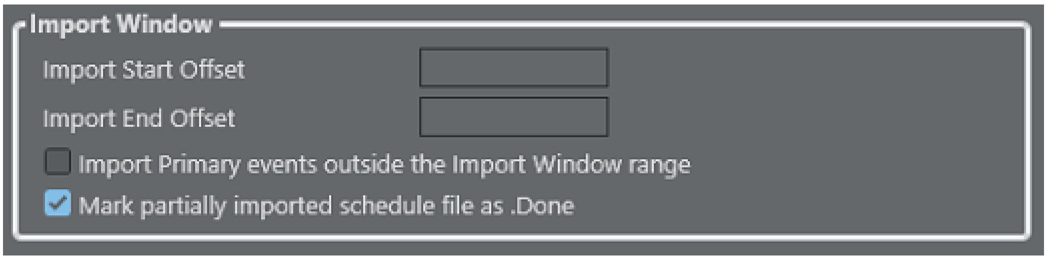
- **Activate Day Switch**: Activates day switch for split playlists. The default Day-Switch is 04:00 but this can be changed as needed in Channel Settings.
- **Insert Cleanup**: Cleans up the program Viz Engine a set *amount of time before the day switch* takes place.
- **Insert Initialize**: Initializes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

> ⚠ **Note:** Time is relative to the day switch action inserted, that in itself is inserted as **the last** element of the day. E.g. 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.

## Filtering Options

- Import Window
- Primary Event Filtering

## Import Window


Import Window

Import Start Offset

Import End Offset

□ Import Primary events outside the Import Window range

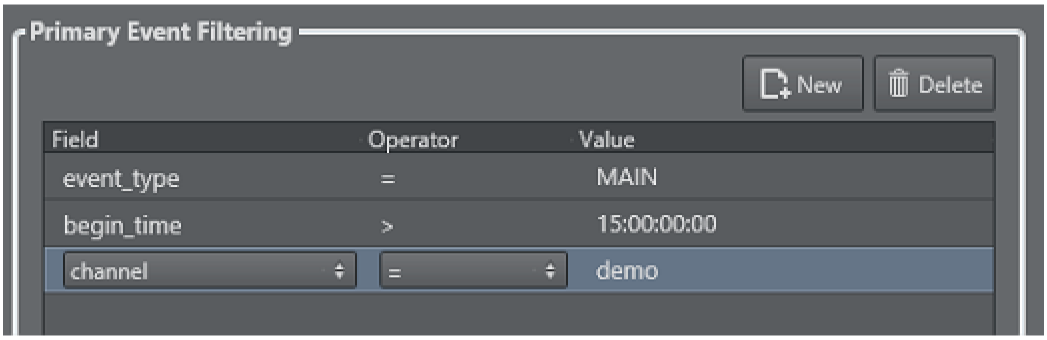
☑ Mark partially imported schedule file as .Done

The import window defines how much, in time, of the playlist that should be imported at any given time. This is useful in order to control the amount of secondary and, if needed, primary events that should be imported.

- **Import Start Offset**: Sets the relative offset for when to start importing elements according to their start time. Format is HH:MM:SS (hours, minutes and seconds). If the current time is 12:30:00 and the start offset is set to 01:00:00 it means only events with a start time set to 11:30:00 and forward will be imported.
- **Import End Offset**: Sets the relative offset time for when to end the import. Format is HH:MM:SS (hours, minutes and seconds).
- **Import Primary events outside the Import Window range**: Imports the primary elements (groups), but will not try to apply rules or create any secondary elements. This setting should be used in conjunction with the offset settings.
- **Mark partially imported schedule files as .Done**: Tags the already processed schedule file as *.done* such that it is filtered out during next import or update. Files that failed to import are tagged as *.fail*.
  Setting an import interval allows for improved control over what is imported and reduces the overall load on the system.

Some channels may also benefit from setting a small import window in order to set a more frequent Automatic Import Interval - see Automatic Import.

## Primary Event Filtering



| Field | Operator | Value |
|---|---|---|
| event_type | = | MAIN |
| begin_time | > | 15:00:00:00 |
| channel | = | demo |

This section allows you to filter the import process, so that source primary events must meet certain criteria in order to make it into the playlist.

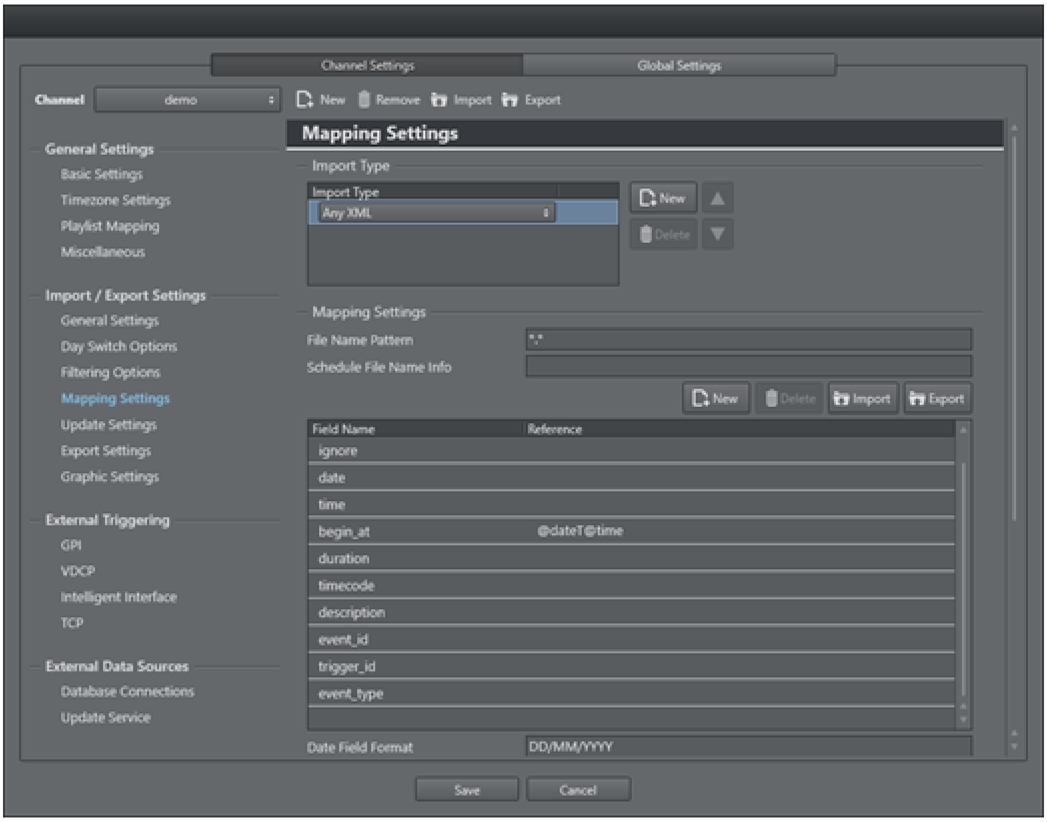
## To Create a Primary Event Filter

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 Secondary Mapping Settings.
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`.
   For a list of operator options, see:

| Operator | Description |
|---|---|
| **=** | equal to |
| **>** | 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 |

## Mapping Settings



Import Type: Any XML

File Name Pattern: *.*

Schedule File Name Info:

| Field Name | Reference |
|---|---|
| ignore | |
| date | |
| time | |
| begin_at | @dateT@time |
| duration | |
| timecode | |
| description | |
| event_id | |
| trigger_id | |
| event_type | |

Date Field Format: DD/MM/YYYY

The primary mapping settings can map from scheduling and/or automation systems. Depending on the local setup, some of these configuration panels are presented:

- Mapping Settings
- Table Columns
- Primary Events Ignore Settings for Page Content Filling
- LST Import Settings
- Excel Import Settings
- Any Text Import Settings
- XML Import Settings
- Omnibus Import Settings

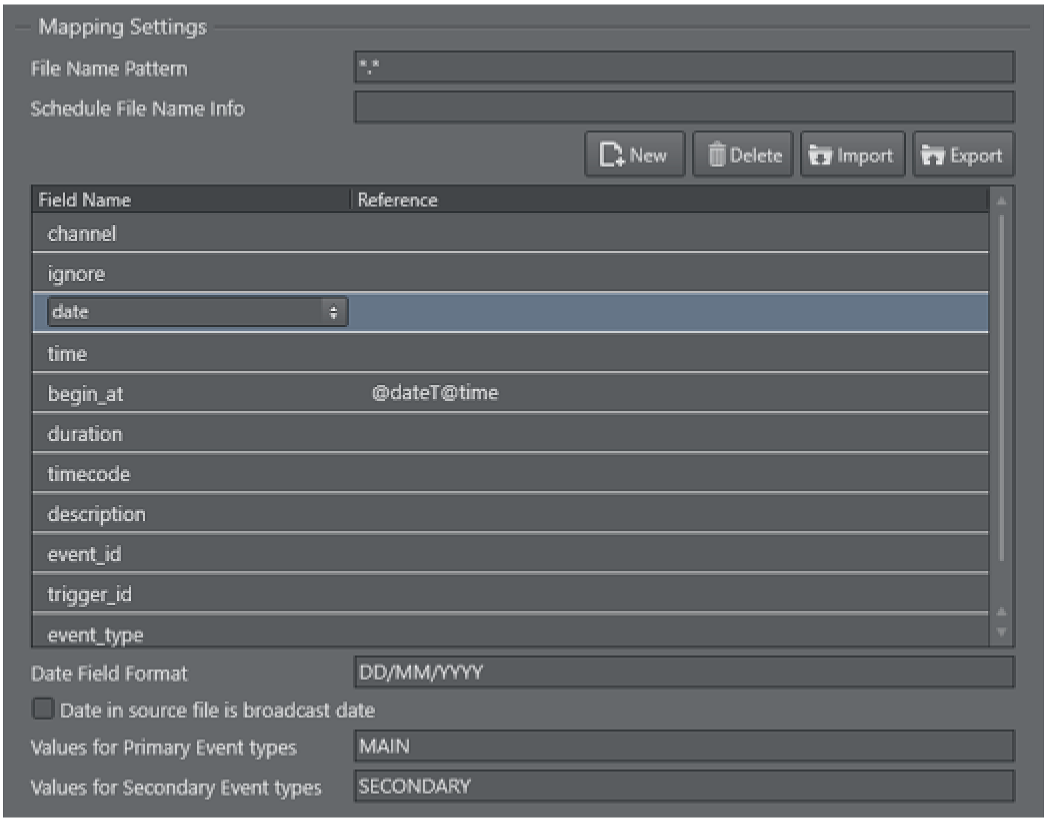
## Mapping Settings

- **Import Type**: Select the type of external playlist.
- **File name pattern**: Enter the specific file format to search for in the import/update folders, e.g. `*.xml`, `*.lst`, `*10.txt`, etc.
- **Schedule File Name Info**: When Reading Channel and Date Values from Filenames, indicate the format that Viz Multichannel 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 occurrences of the letter C are as many times as there are characters in the channel name. X can be used in order to skip characters.

- `Channel10_20100824.txt` = `CCCCCCCC_YYYYMMDD`.
- `20130314_TV2Test.xml` = `YYYYMMDD_XXXCCCC`.

## Table Columns



Mapping Settings

File Name Pattern: *.*

Schedule File Name Info:

| Field Name | Reference |
|---|---|
| channel | |
| ignore | |
| date | |
| time | |
| begin_at | @dateT@time |
| duration | |
| timecode | |
| description | |
| event_id | |
| trigger_id | |
| event_type | |

Date Field Format: DD/MM/YYYY

☐ Date in source file is broadcast date

Values for Primary Event types: MAIN

Values for Secondary Event types: SECONDARY

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

- **Field Name**: Is Viz Multichannel's internal field name. 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. For further information, see the Basic Mapping Table Field Names section for information on required and optional field names, and how to use additional field names.
- **Reference**: Is the 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).

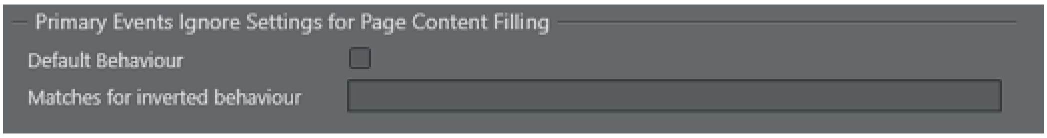
> ⚠ **IMPORTANT!** Field names and reference values vary depending on the automation system and data formats in use. For further information, see the Import Guide.

- **Date Field Format**: This field is optional. If defined, Viz Multichannel will expect to find dates in the source file based on the format provided here.

- **Date in source file is broadcast date**: Check this check 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, indicate the text within the fields that denote primary events. If there is more than one type of denoting text, separate with commas. For example `Program,Promo,Commercial`. This value will be called from within the `event_type` field.
- **Values for Secondary Event Types**: Indicate the text within the fields that denote 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 `seconary_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. These field with errors must be corrected.

## Primary Events Ignore Settings for Page Content Filling



Primary Events Ignore Settings for Page Content Filling

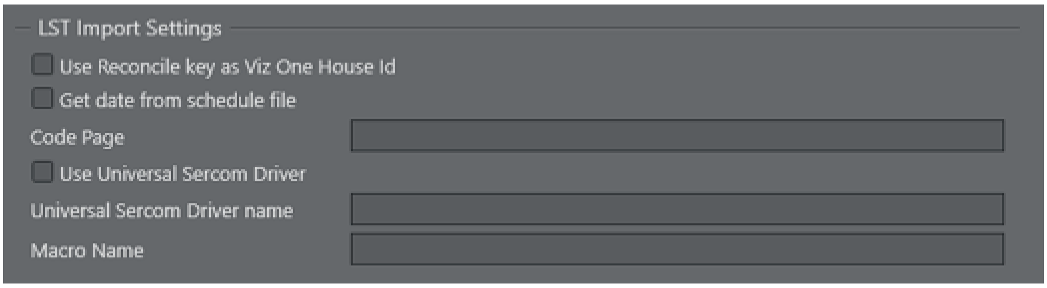
Default Behaviour □

Matches for inverted behaviour

- **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 **not to 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, e.g.: “LIVE”.
- `*LIVE`- any string that **ends** with “LIVE” is a hit, e.g.: “MUSICLIVE”, “NOTLIVE” and “LIVE”.
- `LIVE*`- any string that **starts** with “LIVE” is a hit, e.g.: “LIVESHOW”, “LIVEEVENT” and “LIVE”.
- `*LIVE*`- any string that **contains** “LIVE” is a hit, e.g.: “VERYLIVEMUSIC”, “MUSICLIVE”, “NOTLIVE”, “LIVESHOW”, “LIVEEVENT” and “LIVE”.

## LST Import Settings



LST Import Settings

□ Use Reconcile key as Viz One House Id

□ Get date from schedule file

Code Page

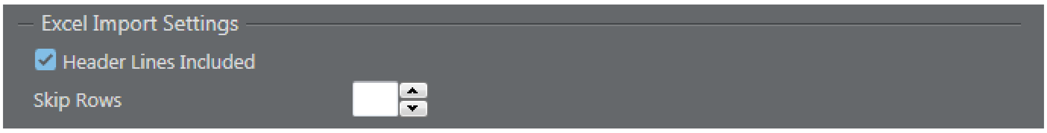
□ Use Universal Sercom Driver

Universal Sercom Driver name

Macro Name

- **Use Reconcile key as Viz One House Id**: After a playlist has been imported, the reconcile keys will be 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, Viz Multichannel will retrieve the schedule date 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**: Set the macro name used in the Harris system.

## Excel Import Settings



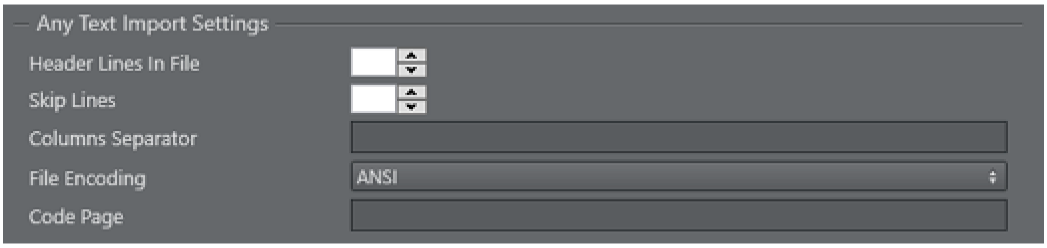
Excel Import Settings

☑ Header Lines Included

Skip Rows

- **Header Lines Included**: If checked, the header lines will *not* be read, and reading will start from the first row of data only. If left unchecked, first line will be read. Only relevant if the import type is Excel (`*.xls` or `*.xlsx`).
- **Skip Rows**: If the Excel file contains any blank lines after the header-line that need to the skipped, set the count here.

## Any Text Import Settings



Any Text Import Settings

Header Lines In File

Skip Lines

Columns Separator

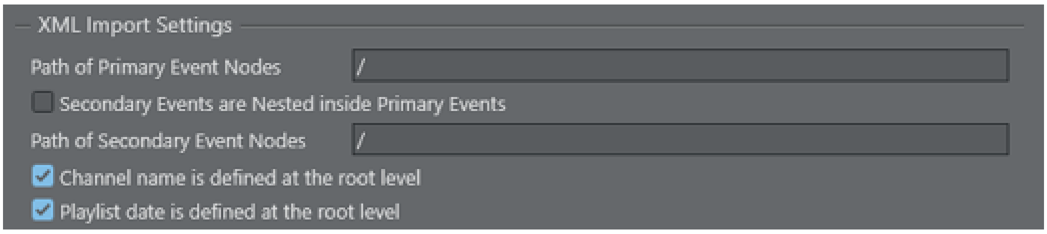
File Encoding: ANSI

Code Page

- **Header Lines in File**: Enter the number of header lines in the Any Text file.

- **Skip Lines**: Enter the number of lines to skip before beginning to read the import source file for data.
- **Columns Separator**: Enter the text symbol (e.g. `*`) that separates the columns in the text file.
- **File Encoding**: Sets the expected file encoding of the imported text file. Alternatives are ANSI, UTF-8, UTF-8 without BOM, and UTF-16.
- **Code Page**: Sets the expected code page of the imported text file (e.g. `28591` for ISO 8859-1 Latin 1 or `874` for ISO/IEC 8859-11 Latin/Thai).

## XML Import Settings



XML Import Settings

Path of Primary Event Nodes: /

□ Secondary Events are Nested inside Primary Events

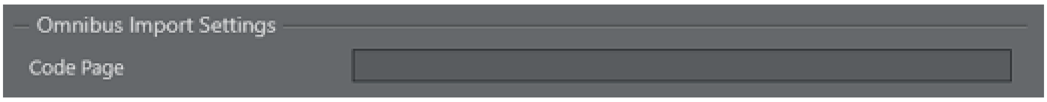
Path of Secondary Event Nodes: /

☑ Channel name is defined at the root level

☑ Playlist date is defined at the root level

- **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.

## Omnibus Import Settings



Omnibus Import Settings

Code Page:

- **Code Page**: Sets the expected code page of the imported text file (e.g. `28591` for ISO 8859-1 Latin 1 or `874` for ISO/IEC 8859-11 Latin/Thai).

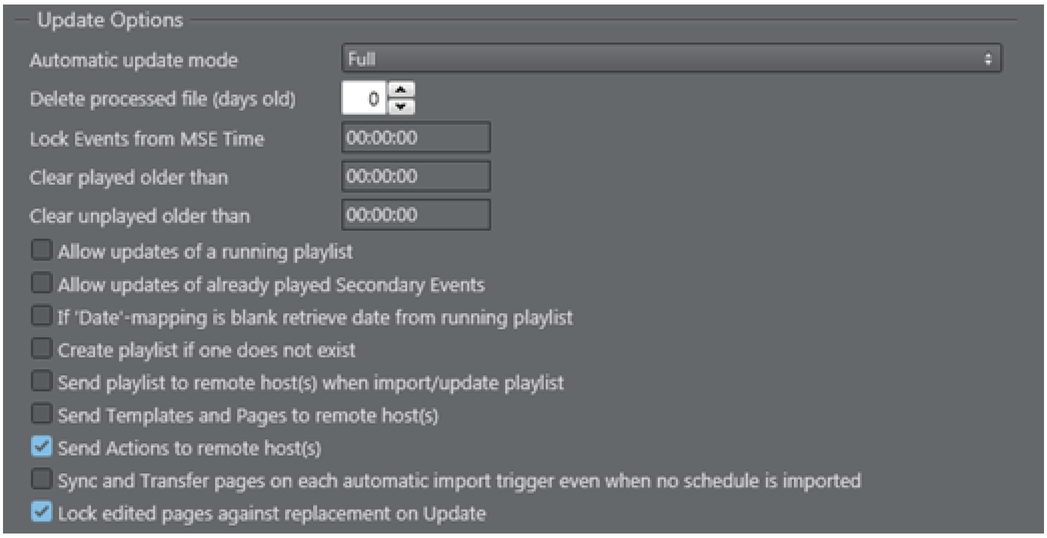
## Secondary Mapping Settings

Secondary mapping settings are usually for automation systems for automatic updates. They are used when Viz Multichannel has to read playlists from two different sources. The configuration methodology is the same as described in the Mapping Settings section.

## Update Settings

- Update Options
- Full Update Options for Secondary Events
- Update Times Options for Secondary Events

## Update Options



Update Options

Automatic update mode: Full

Delete processed file (days old): 0

Lock Events from MSE Time: 00:00:00

Clear played older than: 00:00:00

Clear unplayed older than: 00:00:00

☐ Allow updates of a running playlist

☐ Allow updates of already played Secondary Events

☐ If 'Date'-mapping is blank retrieve date from running playlist

☐ Create playlist if one does not exist

☐ Send playlist to remote host(s) when import/update playlist

☐ Send Templates and Pages to remote host(s)

☑ Send Actions to remote host(s)

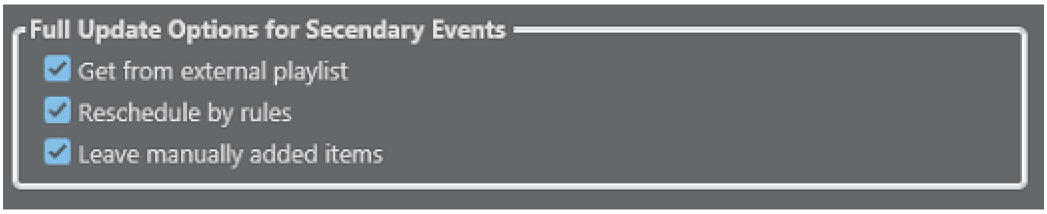
☐ Sync and Transfer pages on each automatic import trigger even when no schedule is imported

☑ Lock edited pages against replacement on Update

- **Automatic Update Mode**: Options are:
  - **Full**: Merging of the playlist as it is in Viz Multichannel with changes that have been made to the original source file.
  - **Times**: Update of primary event begin time, duration and from changes that have been made to the original source file. Any primary event additions or deletions are ignored. Only the times are updated.
  - **Full update without event deletion**: Merging of the playlist as it is in Viz Multichannel with changes that have been made to the original source file, without deleting any (primary) elements.
- **Delete Processed Files (days old)**: Enter the number of days after which processed playlist files (`*.done`) will be deleted.
- **Lock Events from MSE Time**: Sets an interval (hh:mm:ss) during which scheduled events will not be updated. The interval starts as of the next update, whether scheduled or manual, and is timed according to the Media Sequencer clock. The default Lock-Time is 00:01:00.
- **Clear played older than (hh:mm)**: When updating the playlist, Multichannel will remove primary and secondary events which have not been played and are older than the time indicated in this option..
- **Allow updates for a running playlist**: When checked, playlists that are currently active will be updated. When unchecked, only playlists that are not playing will be updated.
- **Allow updates of already played Secondary Events:** When checked all secondary events will be updated. If unchecked then the secondary events that already have been played will not be updated.
- **If ‘Date’-mapping is blank retrieve date from running playlist**: When checked, Viz Multichannel will try to get the date from the running playlist when the “Date”-mapping is blank.
- **Create playlist if one does not exist**: If you update a playlist whose date does not exist, Viz Multichannel imports/creates the playlist for you.

- **Send Playlist to remote hosts when import/update playlist**: Sends the playlist to remote Media Sequencer hosts (see Remote Hosts).
- **Send Actions to remote host(s)**: Sends actions to remote Media Sequencer hosts after import or update playlist (see Remote Hosts). If the remote host has an action with the same name, it will skip replacing it. Default value is false.
- **Sync and Transfer pages on each automatic import trigger even when no schedule is imported.** Although there is no schedule in the import/update folder, templates and pages are sent to Remote Hosts via the Playlist Importer.
- **Lock edited pages against replacement on updates**: self explanatory.

## Full Update Options for Secondary Events



Full Update Options for Secondary Events

☑ Get from external playlist

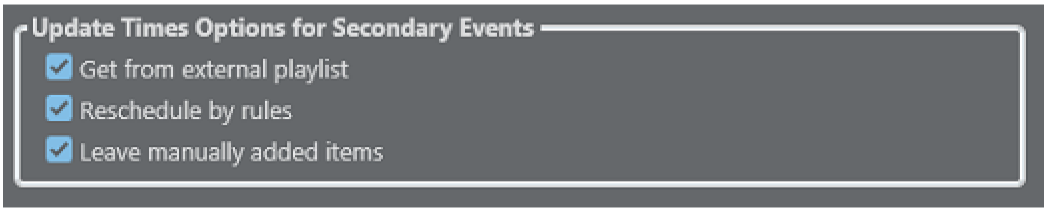
☑ Reschedule by rules

☑ Leave manually added items

These settings are valid both for manual and automatic updates and are checked by default.

- **Get from external playlist**: When checked, secondary events will also be updated from source files when performing full updates. When unchecked, they won't.
- **Reschedule by rules**: When checked, secondary events will be updated according to rules (see Working with Rules), when performing full updates. When unchecked, they won't.
- **Leave manually added items**: When checked, secondary events added manually in Viz Multichannel will not be affected, when performing full updates. When unchecked, they will.

## Update Times Options for Secondary Events



Update Times Options for Secondary Events

☑ Get from external playlist

☑ Reschedule by rules

☑ Leave manually added items

These settings are valid both for manual and automatic updates.

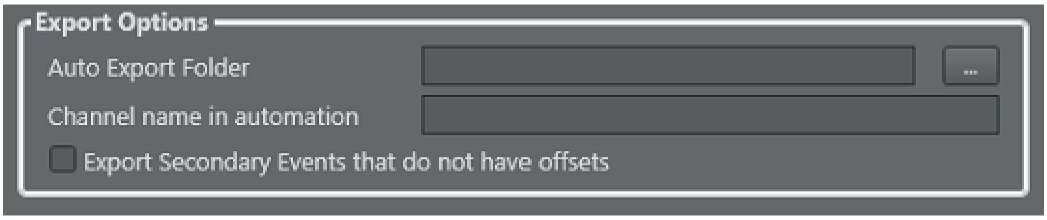
- **Get from external playlist**: When checked, secondary events will also be updated from source files when performing update times. When unchecked, they won't.
- **Reschedule by rules**: When checked, secondary events will be updated according to rules (see Working with Rules), when performing update times. When unchecked, they won't.
- **Leave manually added items**: When checked, secondary events added manually in Viz Multichannel will not be affected, when performing update times. When unchecked, they will.

## Export Settings

- Export Options

- Secondary Event ID Assignment
- Harris (LST)

## Export Options



Export Options

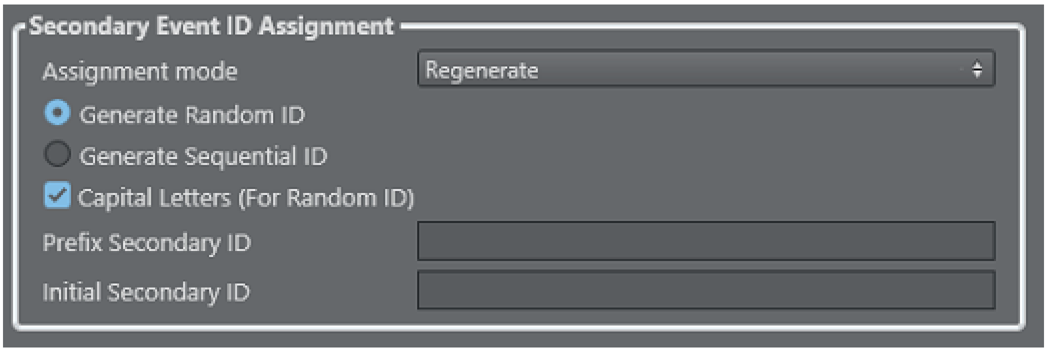
Auto Export Folder ...

Channel name in automation

☐ Export Secondary Events that do not have offsets

- **Auto Export Folder**: Enter the folder path where automatic exports are to be stored.
- **Channel Name in Automation**: Enter a channel name that will be recognized by the automation. Viz Multichannel uses this for all export types except XML. If you leave this field blank, the channel name will be the *Channel Name in Schedule* under the General Settings.
- **Export Secondary Events that do not have offsets**: When checked secondary events that do not have offsets are included in the exported file.

## Secondary Event ID Assignment



Secondary Event ID Assignment

Assignment mode: Regenerate

● Generate Random ID

○ Generate Sequential ID

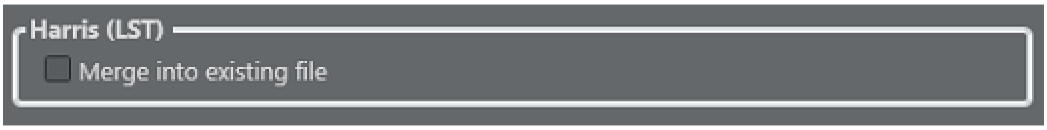
☑ Capital Letters (For Random ID)

Prefix Secondary ID

Initial Secondary ID

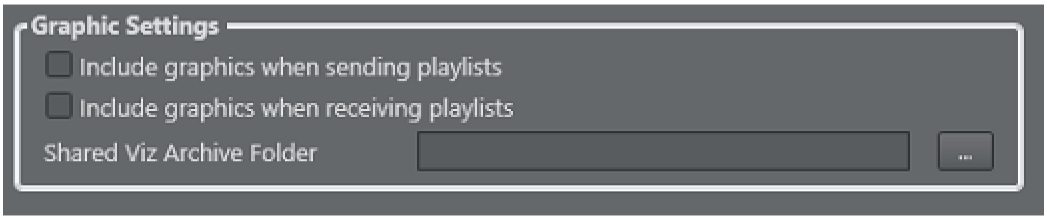
- **Assignment Mode**: Generation of IDs for secondary events during export to an LST file. Options are:
    - **Automatic**: Generates IDs on secondary events that do not have one.
    - **Manual**: Takes no action.
    - **Regenerate**: Generates IDs on all secondary events, overwriting existing ones.
- **Generate Random ID/Generate Sequential ID**: Select Random ID for IDs to be assigned randomly by the system. Select Sequential ID for the numbers to run sequentially from the playlist's first item onward.
- **Capital Letters (for Random ID)**: This option is only relevant if you select Random IDs in the previous option. Selecting capital letters will ensure that the letter portion of the generated ID is always in capitals. Otherwise the letters will be a mix of lower and upper case.
- **Prefix Secondary ID**: This option is only relevant if you are generating sequential IDs. Enter the string that you want as the ID's prefix.
- **Initial Secondary ID**: This option is only relevant if you are generating sequential IDs. Enter the first number of the ID list.

### Harris (LST)



- **Merge Into Existing File**: If left unchecked, when you export, Viz Multichannel exports the selected playlist from scratch. If checked, you are prompted to select an existing file, and any new secondary events are added to that file. The resulting file replaces the old one, which is saved as a backup with the extension{{.orig}}.

### Graphic Settings

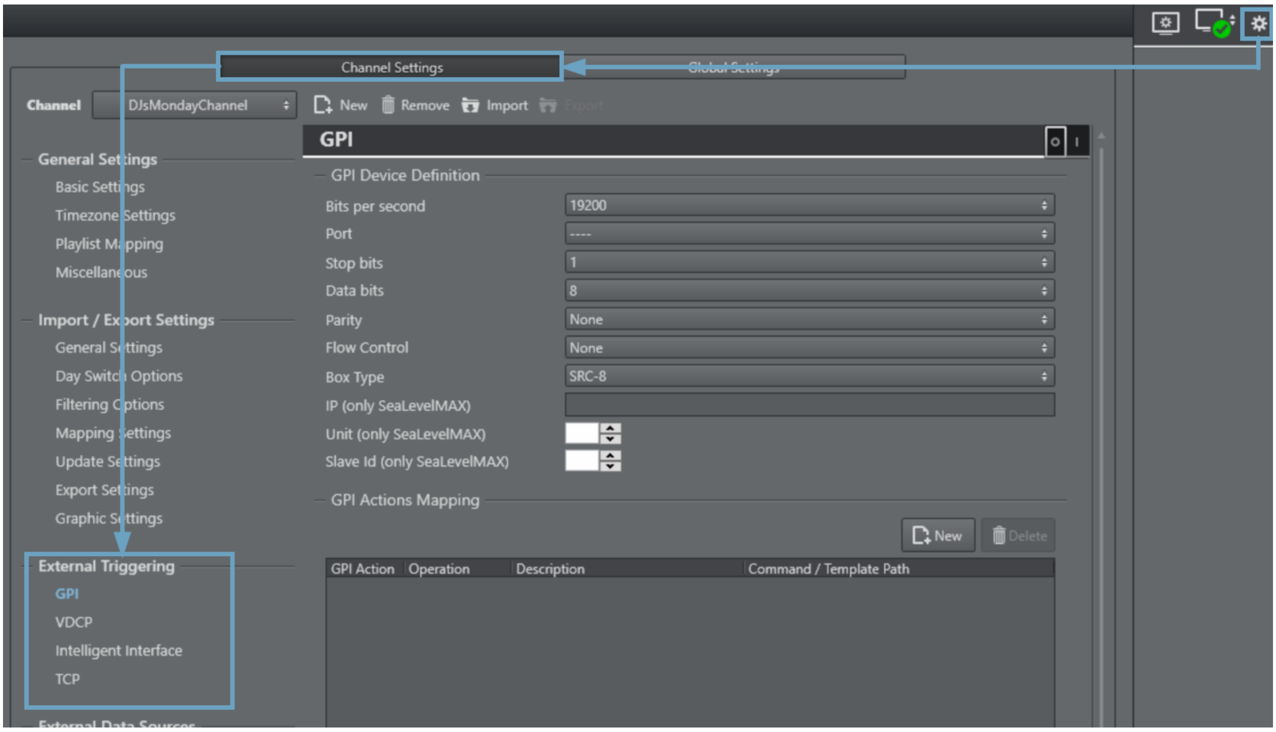

Graphic Settings
□ Include graphics when sending playlists
□ Include graphics when receiving playlists
Shared Viz Archive Folder
...

- **Include graphics when sending playlists**: When checked, exported playlists will include Viz Multichannel templates converted back into Viz Artist archive files (`.via` or{{.eva}}). When unchecked, no graphics will be included.
- *Include graphics data when receiving playlists: *When checked, Viz Multichannel will search for and import playlists along with associated graphics. When unchecked, only the playlist events will be imported.
- **Shared Viz Archive Folder**: Enter the folder path to the shared folder where the Channel Name (defined in the General Settings) will get playlists when Sending Playlists to the Master Control.

## 6.3.3 External Triggering

You locate channel-specific external triggering settings from the cog wheel icon, and selecting the **Channel settings** tab.

External triggering is relevant to the *control Room environment* as the protocols involved may be used in the ongoing broadcast process.
(For a *Traffic Department*, external triggering is usually *not* relevant).



Channel: DJsMondayChannel

Bits per second: 19200

Port: ----

Stop bits: 1

Data bits: 8

Parity: None

Flow Control: None

Box Type: SRC-8

IP (only SeaLevelMAX):

Unit (only SeaLevelMAX):

Slave Id (only SeaLevelMAX):

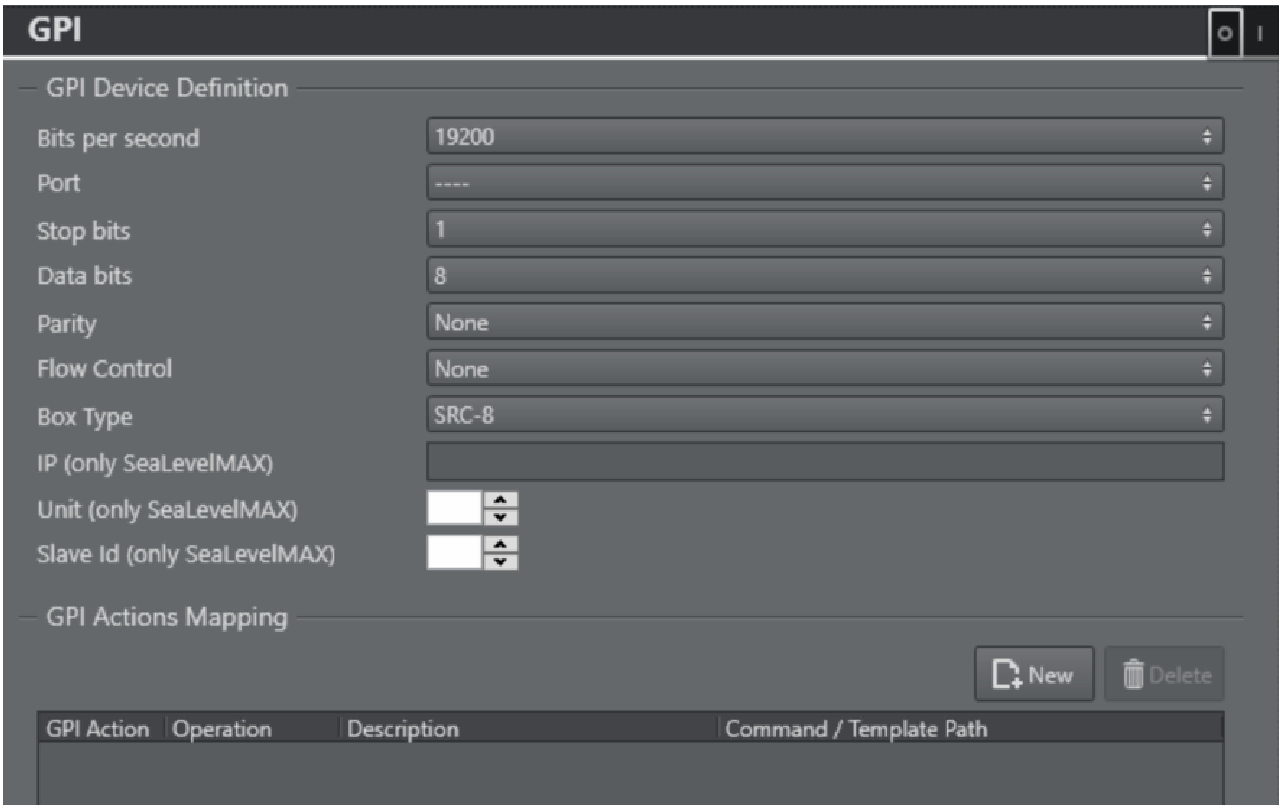
| GPI Action | Operation | Description | Command / Template Path |
|---|---|---|---|

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
- TCP

## GPI



GPI

GPI Device Definition

Bits per second: 19200

Port: ----

Stop bits: 1

Data bits: 8

Parity: None

Flow Control: None

Box Type: SRC-8

IP (only SeaLevelMAX):

Unit (only SeaLevelMAX):

Slave Id (only SeaLevelMAX):

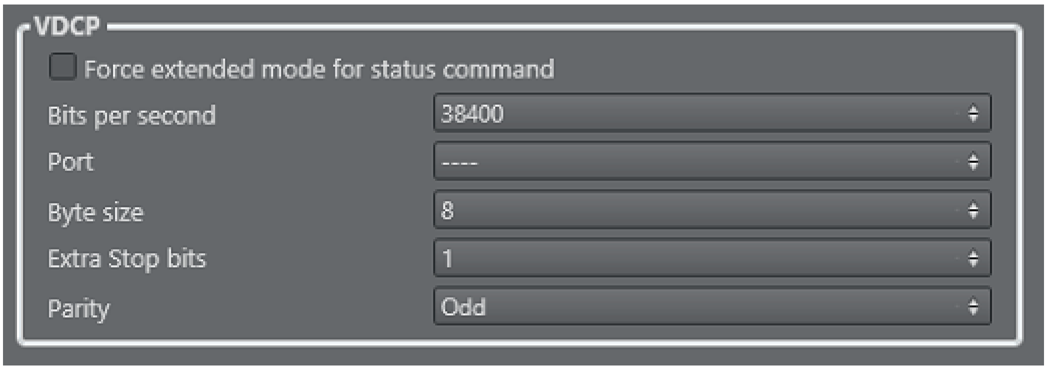
GPI Actions Mapping

| GPI Action | Operation | Description | Command / Template Path |
|---|---|---|---|

GPI is a type of trigger that does not relate directly to a playlist event. From a list of available commands, you can define what Viz Multichannel will do when it receives any one of them 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 the Viz Commands, editable templates and general actions such as show logo, remove logo, stop active playlist, etc. are available as options.
- **Description** can be defined freely.
- **IP/Unit/Slave Id**: only for SeaLevelMAX.

## VDCP



VDCP

□ Force extended mode for status command

Bits per second: 38400

Port: ----

Byte size: 8

Extra Stop bits: 1

Parity: Odd

VDCP is one of the protocols that Viz Multichannel uses to communicate with external automation systems. Use UI button switch to *ON* if using the VDCP protocol, otherwise off.

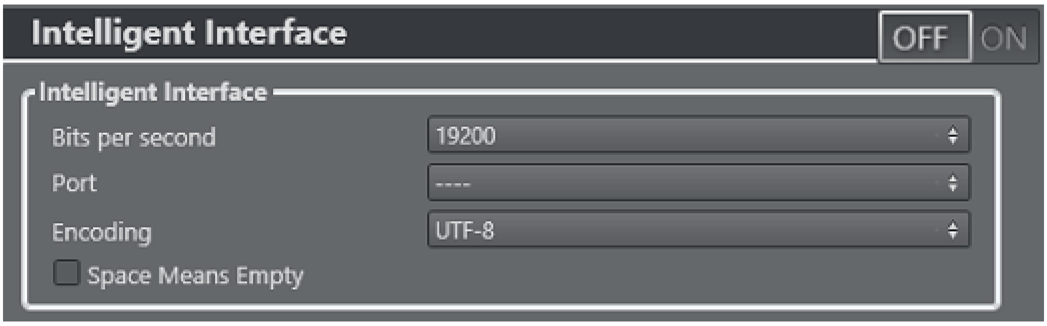
- **Force extended mode for status command** check box: if enabled will include all IDs with length greater than 8 characters in "PORT STATUS REQUEST" and "SYS STATUS REQUEST" VDCP commands.
- **Bits per second**: Default is **38400**.
- **Port**: Select the relevant COM port on the Media Sequencer machine.
- **Byte size**: Default is **8**.
- **Stop bits**: Default is **1**.
- **Parity**: Default is **Odd**.

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

**See Also**

- Media Sequencer documentation regarding **vdcp_client_actor**.

## Intelligent Interface



Intelligent Interface

Intelligent Interface

Bits per second: 19200

Port: ----

Encoding: UTF-8

□ Space Means Empty

Intelligent Interface is one of the protocols that Viz Multichannel uses to communicate with external automation systems.

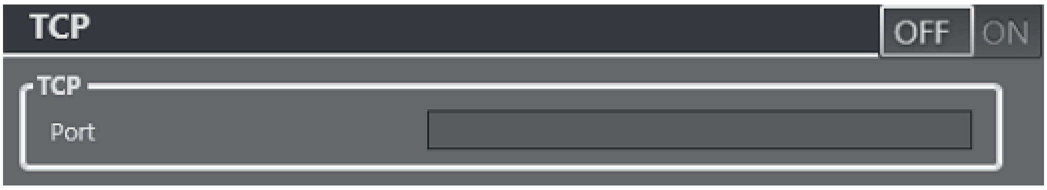
- **Bits per second**: Default is **19200**.

- **Port**: Select the relevant COM port on the Media Sequencer machine.
- **Encoding**: Default is **UTF-8.**
- **Space Means Empty**: Default is **No.**

**See Also**

- Media Sequencer documentation regarding **Intelligentif actor**.

### TCP



TCP
OFF ON
TCP
Port

If you want to receive TCP triggers, you must set a port that the Media Sequencer will listen to for that purpose.

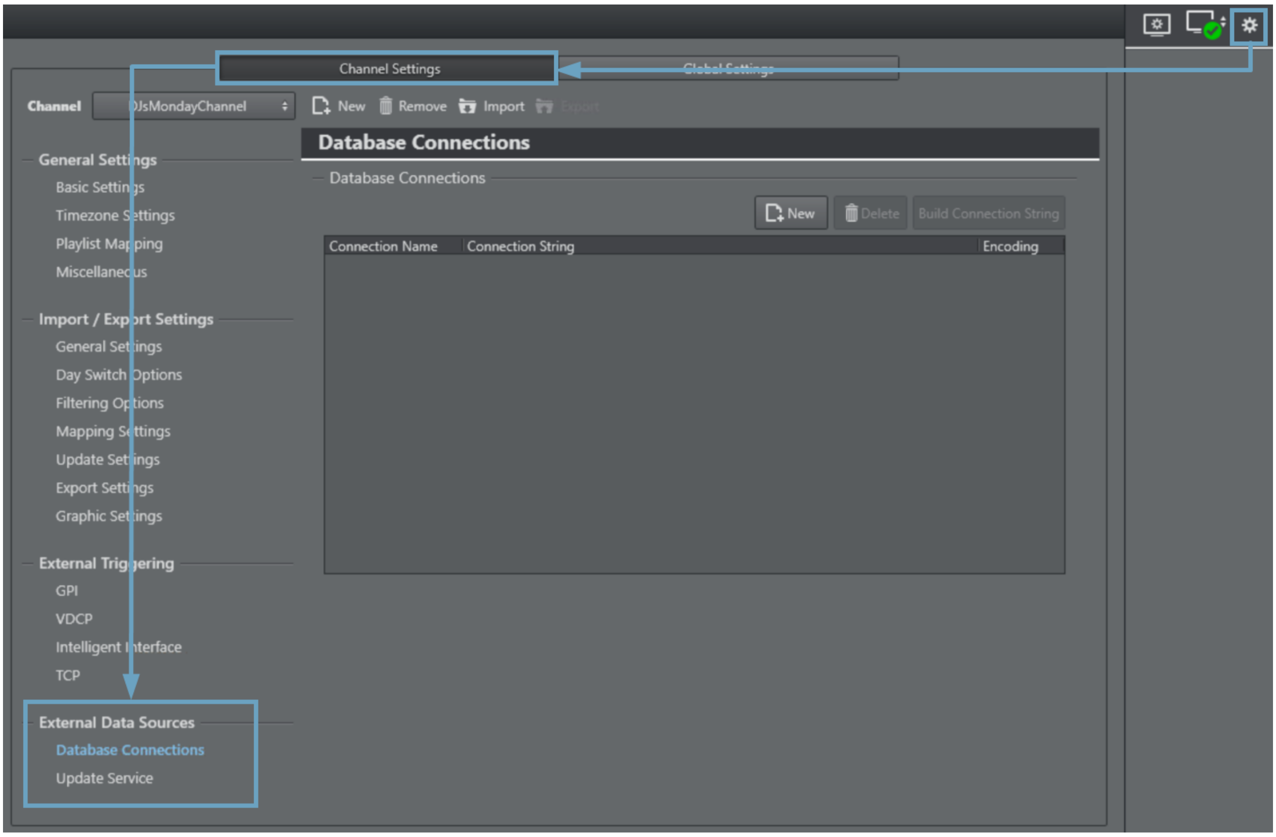
- **Port**: Enter the port number. This option is disabled when the field is left empty.

**See Also**

- Media Sequencer documentation on **pb_listener** actor.

## 6.3.4 External Data Sources

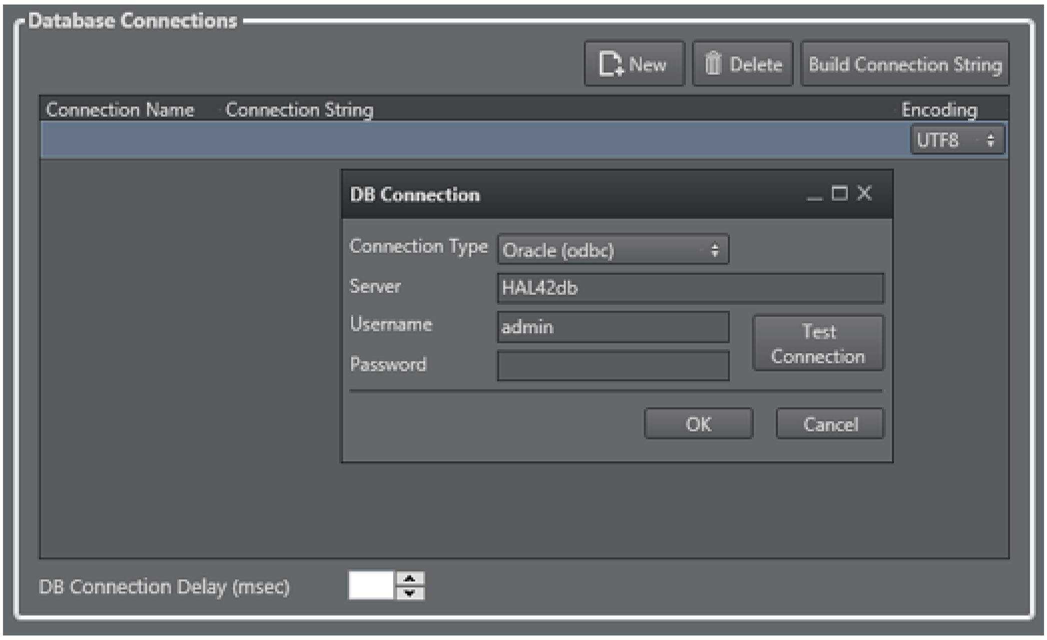
You locate channel-specific general settings from the cog wheel icon, and selecting the **Channel settings** tab.



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 an Microsoft Access Database
        - To Map from a Microsoft SQL Server Database
        - To Map from a MySql Database
        - To Map from an Oracle Database
        - To Map from a Text File
- Using ODBC 64-bit Drivers
- Update Service

## Database Connections



Encoding: UTF8

Connection Type: Oracle (odbc)

Server: HAL42db

Username: admin

Password:

This section describes the Viz Multichannel's 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

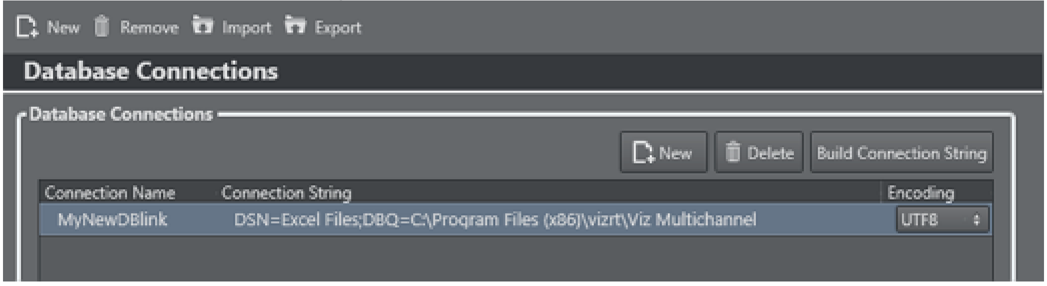
Before populating page content from database look-ups you must first configure the connection to the external data source. You can then proceed with mapping according to your data source.

> ⚠ **IMPORTANT!** If using ODBC in a 64-bit Windows environment, please be aware that there are small driver and driver-name changes to be aware of, see Using ODBC 64-bit Drivers for further information. The driver names and connection strings below are for the 32-bit drivers. The procedures will be the same for 64-bit systems, but the driver names/ connection-strings could be slightly changed.

### To Configure The Connection To An External Data Source

1. In the Settings dialog, click **Database Connections**.
2. Click the **New**.
3. Enter a name for the connection.

4. Enter the connection string provided by your DBA.



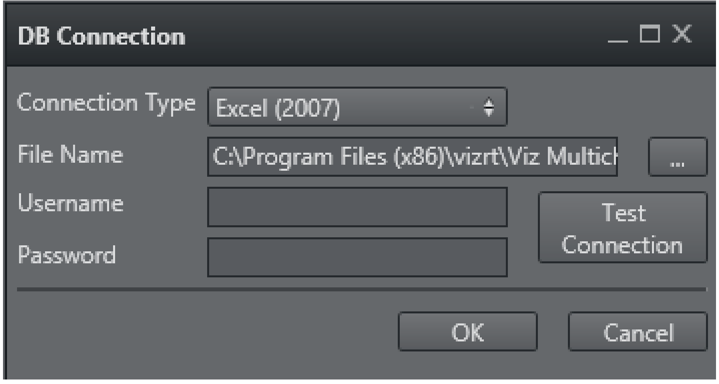
| Connection Name | Connection String | Encoding |
|---|---|---|
| MyNewDBlink | DSN=Excel Files;DBQ=C:\Program Files (x86)\vizrt\Viz Multichannel | UTF8 |

5. Click **OK**.

### To Map From A Microsoft Excel Spreadsheet

**Note**: The file should only contain data of type *text*.

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



Connection Type: Excel (2007)
File Name: C:\Program Files (x86)\vizrt\Viz Multicl
Username
Password

1. For Connection Type, select **Excel 2003** or **Excel 2007**.
2. Click the ellipsis to select the desired Excel file.
3. Click the **Test Connection** button.
4. If the test is successful, click **OK**.

```
DSN=Excel Files;DBQ=C:\Temp\Movies.xls;UID=<username>; PWD=<password>;
```

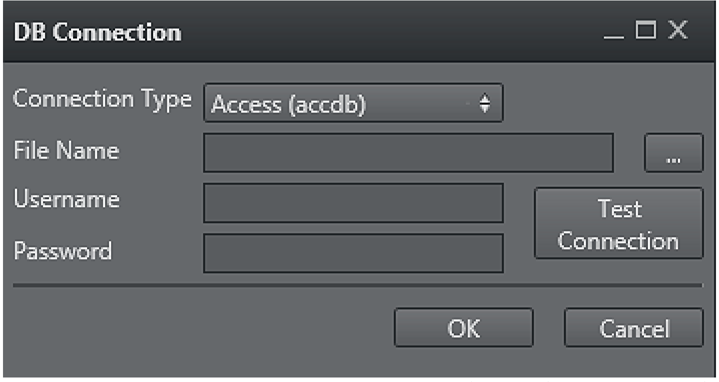
**Note**: An Excel spreadsheet is available in the SampleContent folder (e.g. `C:\Program Files (x86)\vizrt\Viz Multichannel\SampleContent\Database`).

### To Map From An Microsoft Access Database

**Note**: The file should only contain data of type *text*.

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

2. Click the **Build Connection String** button.
   The DB Connection window is displayed.



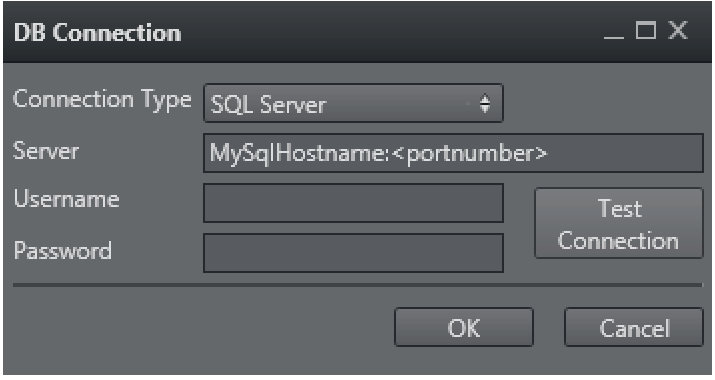
Connection Type Access (accdb)

3. For Connection Type, select **Access (accdb)** or **Access (mdb)**.
4. Click the ellipsis to select the desired Access 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)};DBQ=C:\Temp\Movies.mdb;UID=<username>;
PWD=<password>;
```

### To Map From A Microsoft SQL Server Database

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



Connection Type SQL Server

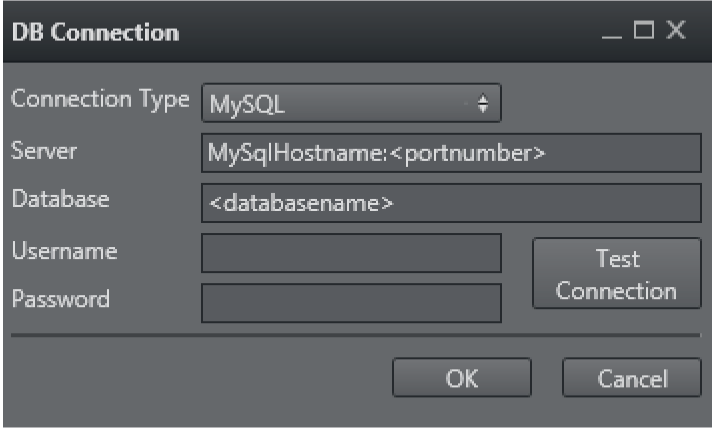
Server MySqlHostname:<portnumber>

3. For Connection Type, select **SQL Server**.
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 the **Test Connection** button.
7. If the test is successful, click **OK**.

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

### To Map From A MySql Database

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



DB Connection

Connection Type: MySQL

Server MySqlHostname:<portnumber>

Database <databasename>

Username:

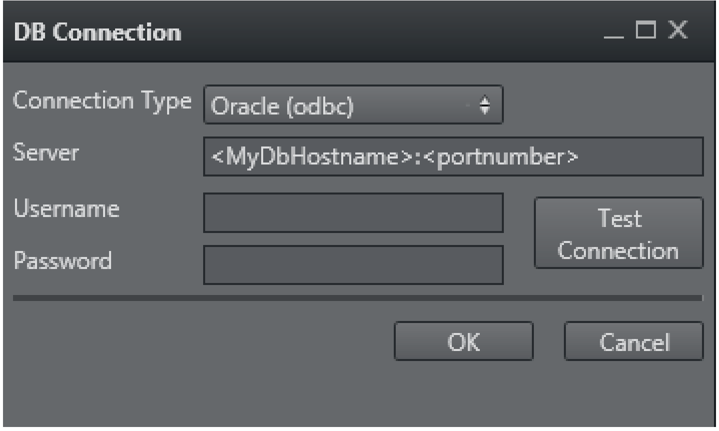
Password:

3. For Connection Type, select **MySql**.
4. Enter IP address or hostname of the database.
5. Enter the name of the database.
6. Enter the username and password for the database.
7. Click the **Test Connection** button.
8. If the test is successful, click **OK**.

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

### To Map From An Oracle Database

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



DB Connection

Connection Type: Oracle (odbc)

Server <MyDbHostname>:<portnumber>

Username:

Password:

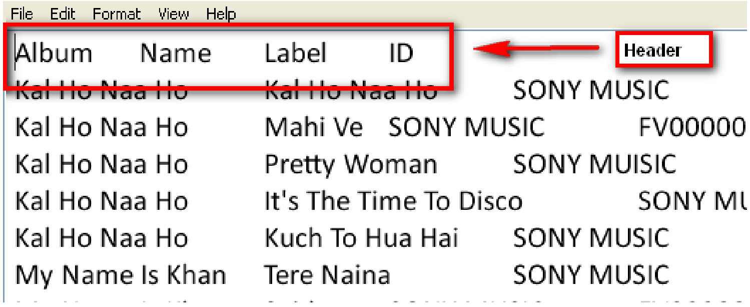
3. For Connection Type, select **Oracle (native)** or **Oracle (odbc)**.
4. Enter the database's server name, using the following format: *<hostname>:<port>* or <IP *address>:<port>*.

5. Enter the database’s username and password.
6. Click the **Test Connection** button.
7. If the test is successful, click **OK**.

```
DRIVER={Microsoft ODBC for Oracle};SERVER=<hostname>:<port>;UID=<username>;
PWD=<password>;
```

### To Map From A Text File

1. Perform the procedure To Configure the Connection to an External Data Source.
2. Create a folder for the text file database, for example `C:\Channel32DB`.
3. 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://www.connectionstrings.com/Providers/microsoft-text-odbc-driver
   - http://msdn.microsoft.com/en-us/library/ms709353(v=VS.85).aspx
4. In `Channel32DB.txt` enter a database header for mapping.



```
Album       Name        Label       ID
Kal Ho Naa Ho           Kal Ho Naa Ho           SONY MUSIC
Kal Ho Naa Ho           Mahi Ve   SONY MUSIC            FV00000
Kal Ho Naa Ho           Pretty Woman            SONY MUISIC
Kal Ho Naa Ho           It's The Time To Disco          SONY MU
Kal Ho Naa Ho           Kuch To Hua Hai         SONY MUSIC
My Name Is Khan         Tere Naina              SONY MUSIC
```

5. 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.

   ```
   {{MaxScanRows=0}}{{CharacterSet=ANSI}}
   ```

6. In the **Database Connections** panel, enter a connection string for the text file database, in the following format

```
Driver={Microsoft Text Driver (*.txt; *.csv)};Dbq=C:
\Channel1_DB;Extensions=asc,csv,tab,txt; If using the Microsoft Access 64-bit
ODBC drivers the format would be: Driver={Microsoft Access Text Driver (*.txt,
*.csv)};Dbq=C:\Channel1_DB;Extensions=asc,csv,tab,txt;
```

> **Note:** For further connection string detail, see: http://www.connectionstrings.com/textfile.

7. Click **OK**.

## Using ODBC 64-bit Drivers

When configuring and using ODBC in a 64-bit environment you need to have 64-bit ODBC drivers installed or bridge 32-bit drivers in a 64-bit system.

A solution is to install the Microsoft native 64 drivers that is part of the Microsoft Access Database Engine 2010 Redistributable driver pack, available at (external link): http://www.microsoft.com/downloads/details.aspx?familyid=C06B8369\-60DD\-4B64\-A44B\-84B371EDE16D&displaylang=en

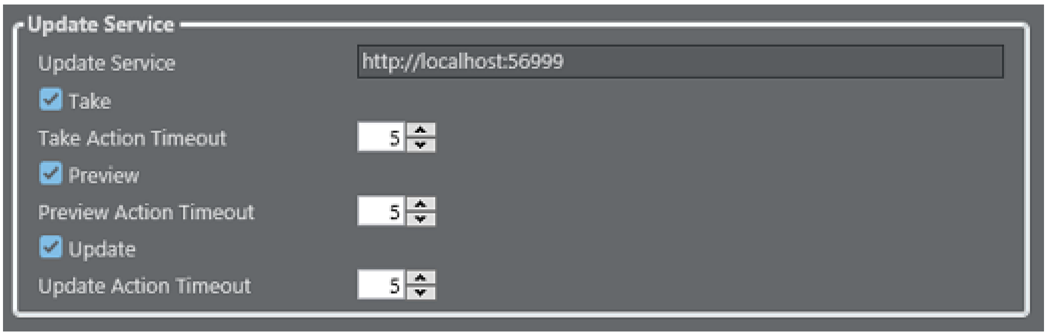
However, please note that all 64-bit ODBC drivers from the Microsoft Access Database Engine 2010 Redistributable have slightly changed their driver names to differentiate them from their 32-bit counterparts.

The connection string for ODBC text and csv files will be:

```
Driver={Microsoft Access Text Driver (*.txt, *.csv)};
```

See www.microsoft.com for further information.

## Update Service



Update Service

Update Service: http://localhost:56999

☑ Take

Take Action Timeout: 5

☑ Preview

Preview Action Timeout: 5

☑ Update

Update Action Timeout: 5

Update Service is an option that makes use of 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*.

The Update Service, when configured and enabled on one or more templates, will activate the Media Sequencer (MSE) to send an **XML Payload** when a Page is put in pre-queue of being triggered, to a custom web service. This web service can modify the content of the payload and send it back. When Media Sequencer Engine receives the payload it will parse it and put the data into the page's tab field.

The typical use case could for example be for sport-results or latest headlines from a feed. Since it relies on a custom web service the solution can be tailored to be unique. This update service is *not* an existing server solution, it is a path for external modules to be able to integrate and function with Multichannel and the Media Sequencer.

A **timeout** of N seconds on the available actions (**Take**, **Preview** and **Update**) may be set. This is to make sure a reply is given back during the configured timeout (default is 5 seconds) for a given action. If no reply is received within the time-frame the action is not sent/issued, even though a reply is received later.

**See Also**

- Playlists

## 6.4 Local Settings

Some local Viz Multichannel client settings are configured directly in Viz Multiplay's configuration files, rather than through the Configuration menus.

### 6.4.1 Local Viz Engine Characteristics

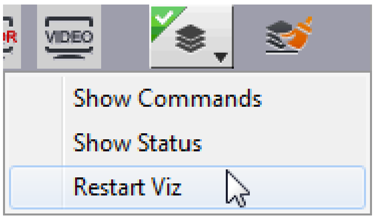
You can modify some local Viz Engine renderer settings in the config file `VizRundown.ini`, located in your Viz Multichannel program folder.

- **Viz Engine Connection Timeout**: Viz Engine connection timeout, in milliseconds (for example *30000*).
  If Viz Engine is not sending a response within the set amount of milliseconds, an error is displayed in the **Information** panel with status. The **Viz Engine connection** icon turns red.
  At timeout you must restart it.
  By default the connection timeout is set to 30000 ms (30 sec).
  See also how the procedure To Restart the Viz Engine Connection below.
- **Scenes in Viz Engine Memory**: Maximum number (`0-8`) of Viz Artist scenes to reside in the Viz Engine preview memory.
  These function as a cache for previewing templates. If a lot of manual template previewing and editing is expected, this number can be higher.
  0 means unlimited. The recommended default is 8.
- **Renderer Force Sleep**: Set this option (`true/false`) if an NVIDIA G-force graphics card is installed on the Viz Multichannel client machine.
  Otherwise the machine might, in some cases, use excessive CPU power.

```
[LocalVizEngine] VizEngineConnectionTimeout=30000 ScenesInVizEngineMemory=8
 IsRendererForceSleep=true
```

#### To Restart the Viz Engine Connection

- Click the **Viz Engine connection** icon and from the context menu and select **Restart Viz**.



**See Also**

- Global Settings
- Working with Channel Settings
- Channel Settings

- Video Configuration

## 6.5 Video Configuration

In order for video to pass through Viz Engine when no scenes are loaded you need to activate the Matrox Watchdog functionality and enable video to loop through the Matrox X.mio or X.mio 2 board.

⚠ **Note:** For Viz Engine 3.6 and X.mio 2 plus users the below procedure is not relevant.

### 6.5.1 To 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 from and to 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
- Local Settings
- Working with Channel Settings
- Channel Settings

## 6.6 Playout Configuration

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

For your *graphics output* you assign Viz Engines. 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 the Graphics Workflow sections on Upstream and Downstream workflows.

- Forked Execution
- Configuring Device Channels
- Configuring Playout Devices

### 6.6.1 Forked Execution

Forked execution allows you to setup 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**

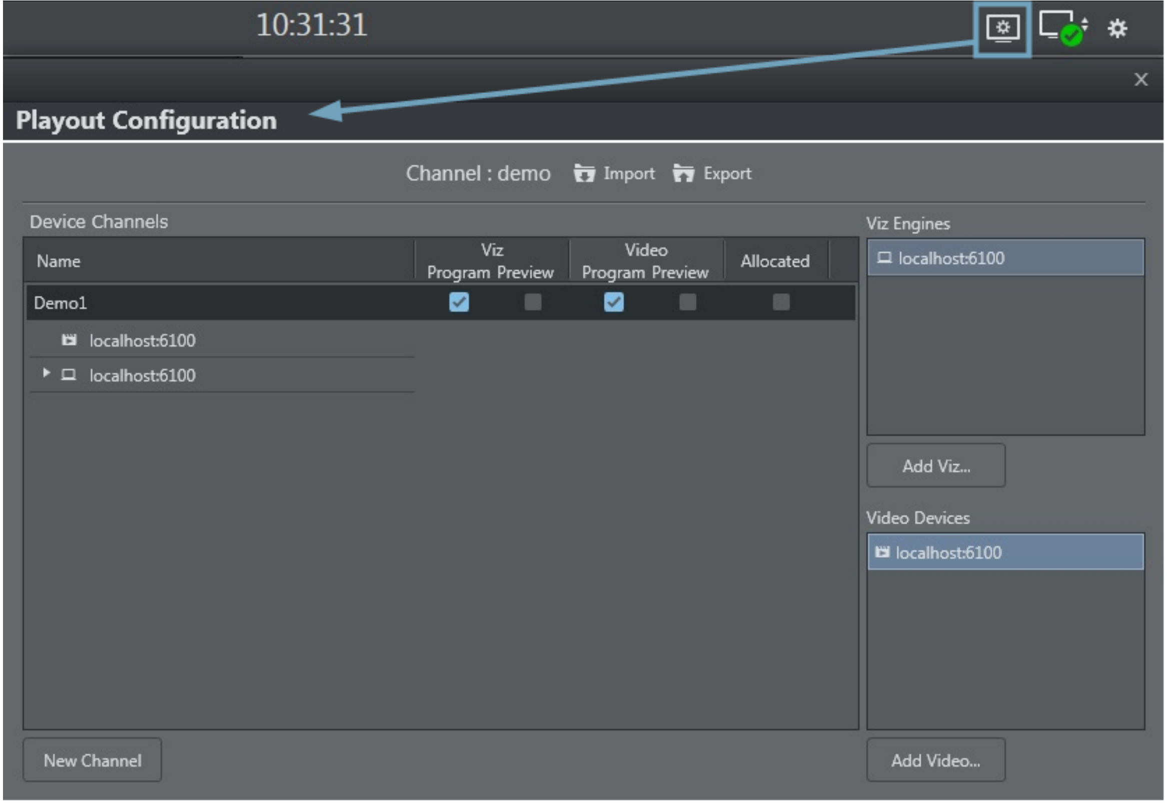
- *Viz Artist  User Guide* section on *Transition Logic*

### 6.6.2 Configuring Device Channels

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).

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:

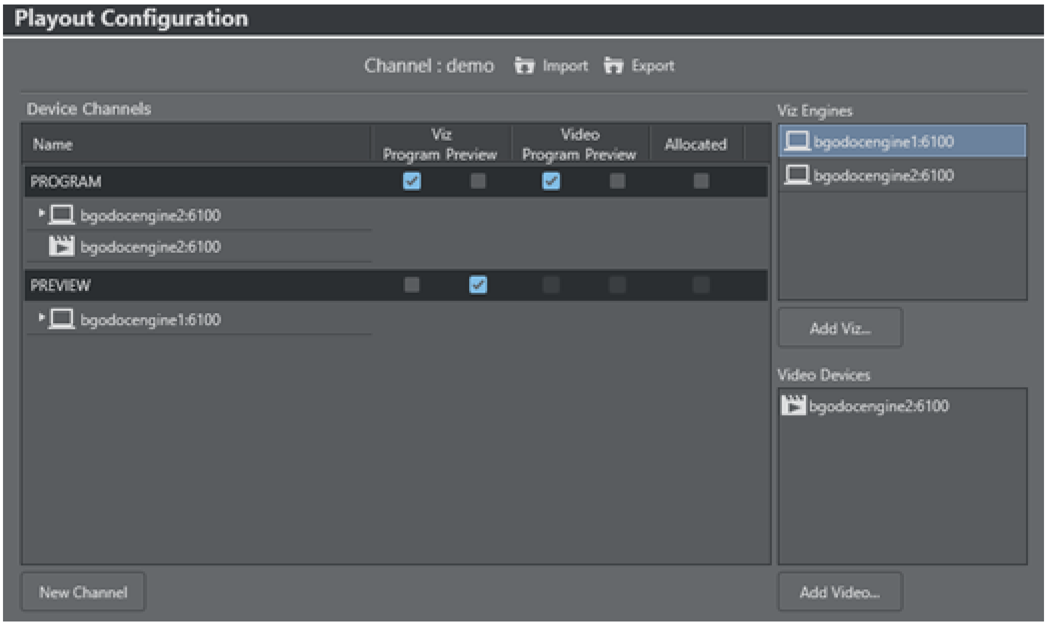


You can use the **Import** and **Export** buttons to import (or save) a configuration to an XML-file for safekeeping. Importing a configuration from an XML-file makes it easy to quickly change configuration parameters.

This section contains the procedures:

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

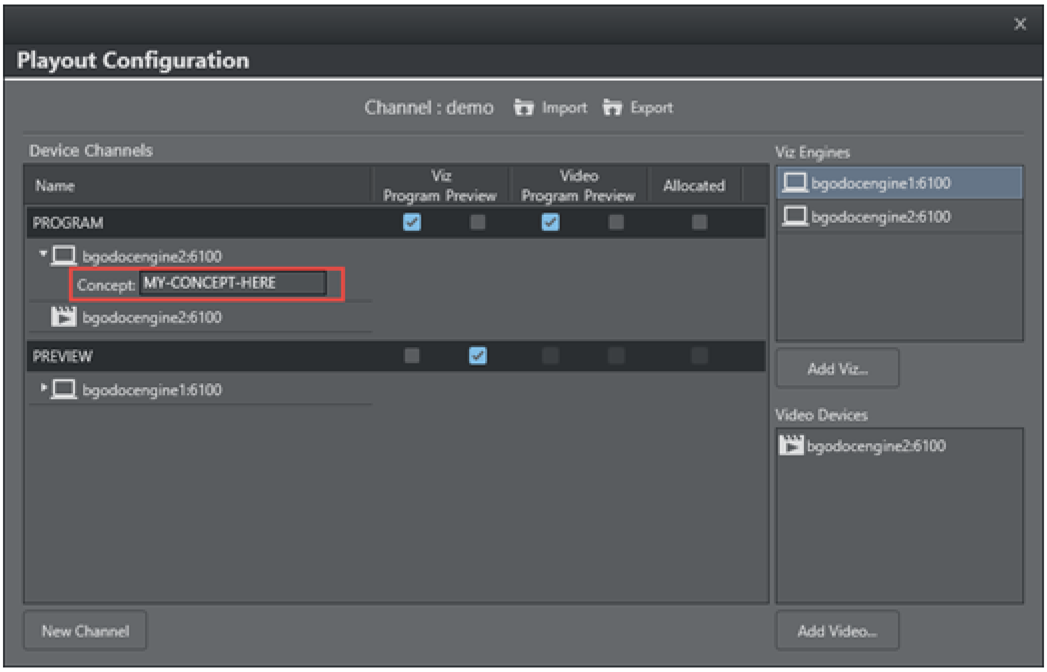
## To Add a Playout Device to the Device Channels List



| Name | Viz Program | Viz Preview | Video Program | Video Preview | Allocated |
|---|---|---|---|---|---|
| PROGRAM | ☑ | □ | ☑ | □ | □ |
| bgodocengine2:6100 | | | | | |
| bgodocengine2:6100 | | | | | |
| PREVIEW | □ | ☑ | □ | □ | □ |
| bgodocengine1:6100 | | | | | |

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

## To Add a Concept Override for a Channel’s Playout Device



| Name | Viz Program | Viz Preview | Video Program | Video Preview | Allocated |
|---|---|---|---|---|---|
| PROGRAM | ☑ | □ | ☑ | □ | □ |
| bgodocengine2:6100<br>Concept: MY-CONCEPT-HERE | | | | | |
| bgodocengine2:6100 | | | | | |
| PREVIEW | □ | ☑ | □ | □ | □ |
| bgodocengine1:6100 | | | | | |

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

⚠ **Caution:** Note that 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.**

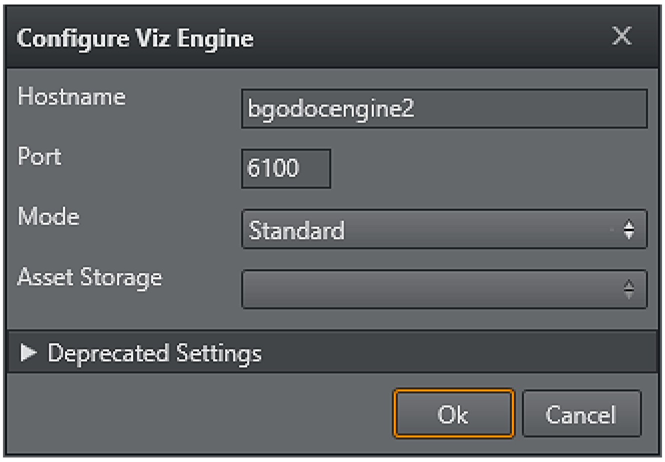
## 6.6.3 Configuring Playout Devices

Playout devices are physical output devices that either playout program or preview content of graphics or video. These devices are added to *device channels* for the purpose of logically grouping playout devices into functional areas of responsibility; they act as program or preview channels for either graphics or video.

This section contains the procedures:

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

### To Configure a Graphics Device



Configure Viz Engine

Hostname: bgodocengine2

Port: 6100

Mode: Standard

Asset Storage:

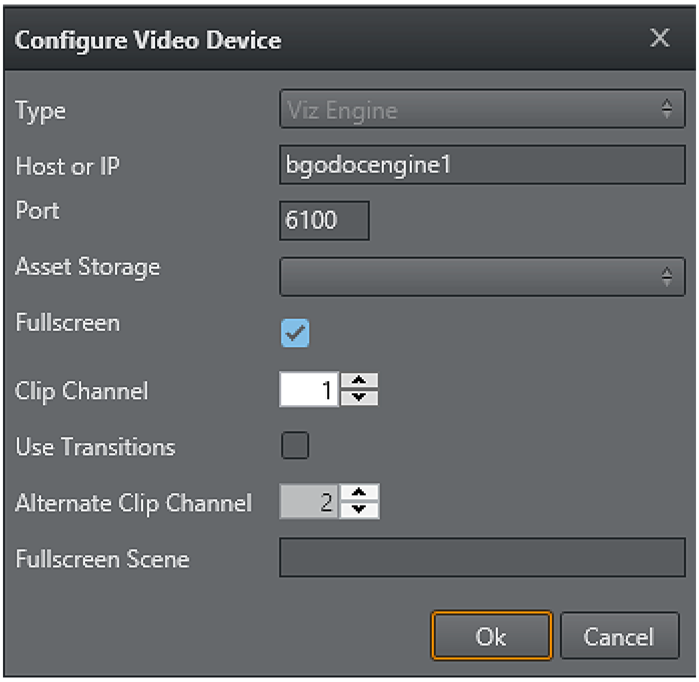
1. Click the **Add Viz** button to open the Configure Viz Engine dialog box.
    a. Enter the hostname and port

   b. Default port for Viz Engine is 6100.
2. *Optional*: Select **Mode**
   a. **Scene Transitions**: See how To Enable Scene Transitions
   b. **Still Preview**: See how 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 Multichannel.

## To Configure a Video Device

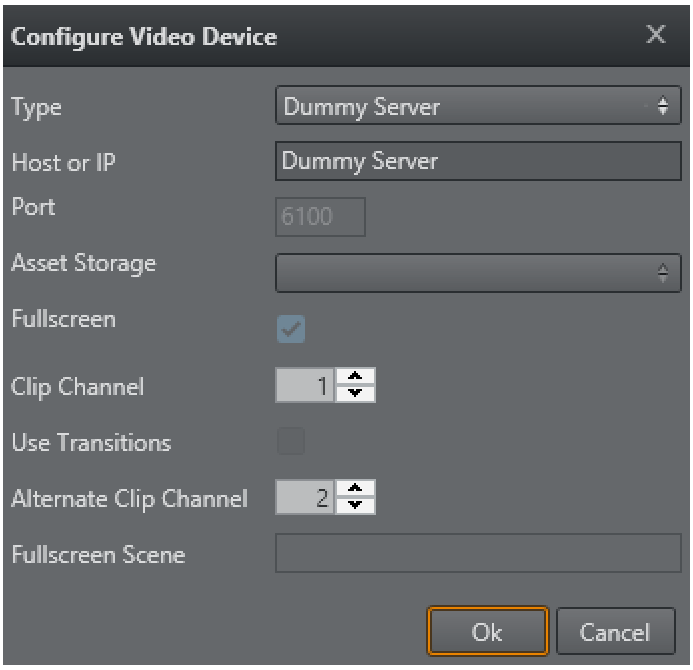


Configure Video Device

| | |
|---|---|
| Type | Viz Engine |
| Host or IP | bgodocengine1 |
| Port | 6100 |
| Asset Storage | |
| Fullscreen | ☑ |
| Clip Channel | 1 |
| Use Transitions | ☐ |
| Alternate Clip Channel | 2 |
| Fullscreen Scene | |

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.

## To Configure a Dummy Video Device



Configure Video Device

| | |
|---|---|
| Type | Dummy Server |
| Host or IP | Dummy Server |
| Port | 6100 |
| Asset Storage | |
| Fullscreen | ☑ |
| Clip Channel | 1 |
| Use Transitions | □ |
| Alternate Clip Channel | 2 |
| Fullscreen Scene | |

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

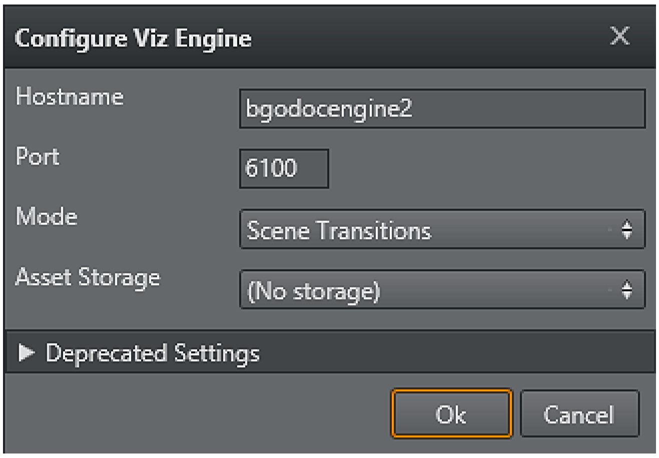
- Right-click the video device and select **Delete** from the context menu, or simply select it and press the **Delete button.**

## 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
  - **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 are for example, wipes, dissolves and alpha fades.



Configure Viz Engine

Hostname: bgodocengine2

Port: 6100

Mode: Scene Transitions

Asset Storage: (No storage)

1. Configure the Viz Engine settings as described in how To 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:** To see the effects, the program channel must be configured and on-air.

**See Also**

- Configuring Device Channels

# 7 Import Guide

This section describes the supported import modules and how to setup Viz Multichannel to import schedules and thereby create playlists for the client.

To help in filling out the Mapping Settings and Secondary Mapping Settings, the following topics serve to explain how to perform various types of imports into Viz Multichannel:

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

**See Also**

- Page Content Filling

## 7.1 Basic Mapping Table Field Names

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

> ⚠ **Note:** Field Name variables can only use ASCII characters (a-z/A-Z). If underscores are used the variable name *must* be bracketed, like this [`field_name_variable`].

### 7.1.1 Required Field Names

If the following 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 the *Channel Name in Schedule* blank.
- **date**: Is the date the primary event is supposed to be played out.
- **time**: Is the time the primary event is supposed to be played out.
- **begin_at**: Is the combination of date and time the primary event is supposed to be played out.
- **event_id**: Is the 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 (e.g. *ReconcileKey* and *Segment*).

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

### 7.1.2 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.
- **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 Multichannel. See Working with Templates and Pages.
- **secondary_timecode**: Offset relative to the primary event's start time.
- **secondary_duration**: Duration of the secondary event.
- **house_id**: Map the house ID to the `house_id` keyword. This will allow Viz Multichannel client to resolve assets on Viz One (see the Viz One Integration chapter). Will create a composite group in your playlist (i.e. a primary element containing a video asset). 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 user click on Continue from LIVE show button. When playlist is resumed from live event earlier or later than plan, begin time of events following live event are shifted to be taken on air immediately. Begin time of events will be adjusted up until the first locked event.

### 7.1.3 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 other format than the LST Import Type.

## 7.2 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 an *fixed width information* structure.

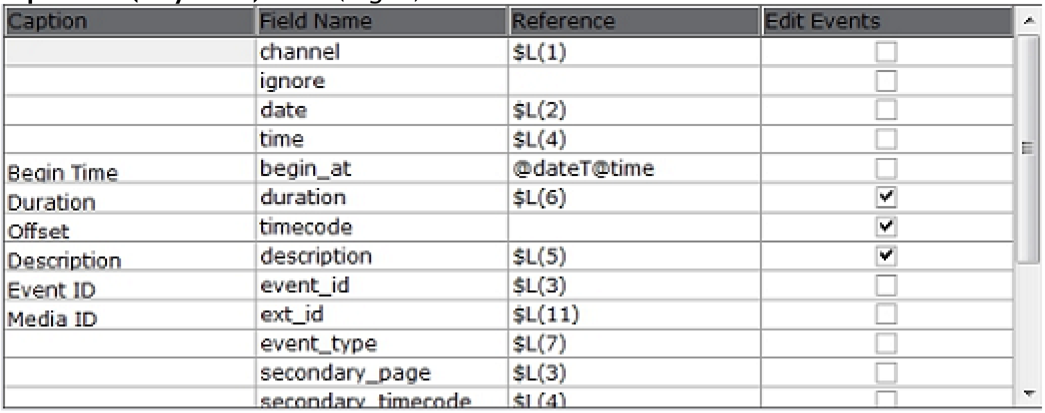
In a *delimited field structure* file, each line contains a list of values, delimited by a configured character like a comma or tab. In an *fixed width information* file, each line contains a list of values, and each value is stored in a given place, like the first to tenth characters.

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

Before performing one of the procedures below you need to make sure a **schedule** is available in the required format (i.e. Any Text) and that you have created a channel**.**

## 7.2.1 To Import Delimited Files

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



| Caption | Field Name | Reference | Edit Events |
|---|---|---|---|
| | channel | $L(1) | ☐ |
| | ignore | | ☐ |
| | date | $L(2) | ☐ |
| | time | $L(4) | ☐ |
| Begin Time | begin_at | @dateT@time | ☐ |
| Duration | duration | $L(6) | ☑ |
| Offset | timecode | | ☑ |
| Description | description | $L(5) | ☑ |
| Event ID | event_id | $L(3) | ☐ |
| Media ID | ext_id | $L(11) | ☐ |
| | event_type | $L(7) | ☐ |
| | secondary_page | $L(3) | ☐ |
| | secondary_timecode | $L(4) | ☐ |

6. In the 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 (e.g. `demo`) in the list, *$L(2)* the second (e.g. `20100923`), *$L(3)* the third (e.g. `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 (e.g. `20100923_demo`) and click **Open**
   - This will import the schedule and create a playlist in Viz Multichannel
     Note that you may not have all the **secondary elements** (i.e. pages) in your playlist. Check the **schedule file** for secondary elements and **create** the pages required and perform an **update** of your playlist.

1. Right-click your playlist (e.g. Continuous Playlist) and select **Update Playlist > Full Update**
2. Select your schedule file (e.g. `20100923_demo`) and click **Open**

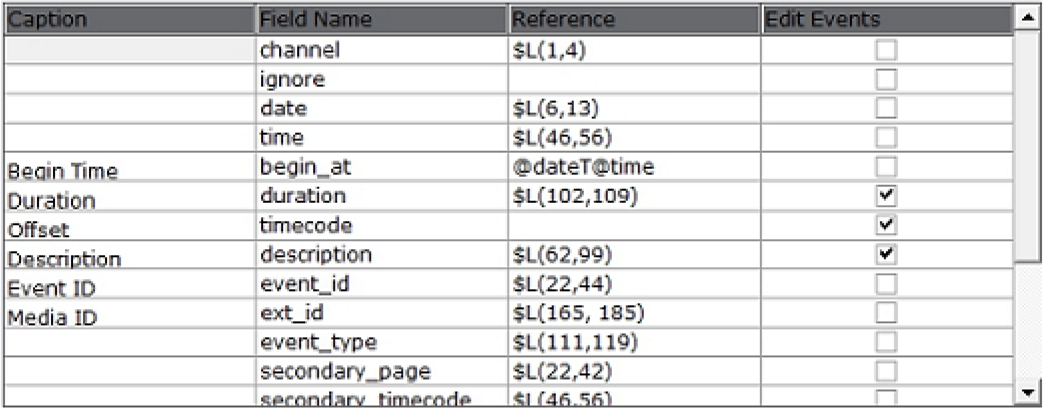
## 7.2.2 To Import Delimited Files with Header Lines

1. If a delimited file contains header lines, configure the number of header lines for Viz Multichannel to recognize them as such, with the setting **Header lines in file** under the Excel Import Settings.
2. For fields read from the header lines, link the field's line and location counters in the file, in the format *$H(line, location)*, where *H* means header.

> ⚠ **Example:** Map *$H(1,3)* for the third value in the first header line, *$H(4,2)* for the second value in the forth header line, etc.

## 7.2.3 To Import Fixed Width Files

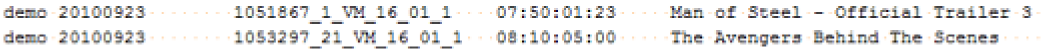
1. Open the **Settings** window.
2. Select the **General** Settings section and set the **Channel Name in Schedule.**
    - This procedure uses channel *demo* as seen in the example imagery below.
3. Select the **Mapping Settings** section and set **Import Type** to **Any Text**
4. Set the *File name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



| Caption | Field Name | Reference | Edit Events |
|---|---|---|---|
| | channel | $L(1,4) | □ |
| | ignore | | □ |
| | date | $L(6,13) | □ |
| | time | $L(46,56) | □ |
| Begin Time | begin_at | @dateT@time | □ |
| Duration | duration | $L(102,109) | ☑ |
| Offset | timecode | | ☑ |
| Description | description | $L(62,99) | ☑ |
| Event ID | event_id | $L(22,44) | □ |
| Media ID | ext_id | $L(165, 185) | □ |
| | event_type | $L(111,119) | □ |
| | secondary_page | $L(22,42) | □ |
| | secondary_timecode | $L(46,56) | □ |

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.

## 7.2.4 Sample content – fixed width (indexed)



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

> ⚠ **Example:** *$L(1,2)* maps the first value (e.g. `demo`) in the list, *$L(4,11)* the second (e.g. `20100923`), *$L(20,40)* the third (e.g. `1051270_23_VM_16_01_1`), and so on.

1. If the imported file contains header lines, configure the number of the header lines for Viz Multichannel to recognize them as such, with the setting **Header lines in file** in the **Mapping Settings**.

2. For fields read from the header lines, link the field's line counter and location indexes in the file, in the format *$H(line, start-index, end-index)*.

   > ⚠ **Example:** Map *$H(2,5,10)* if the field is in the second header line's 5th-10th characters, etc.

## 7.3 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**.

### 7.3.1 Map Primary and Secondary Events

#### To Map Primary and Secondary Events

1. Start Viz Multichannel
2. Open the **Settings** window
3. Select your channel
4. Select **Primary Mapping Settings**
   See examples below on how to set the paths before
5. Under the **XML Import Settings** set the parameters **Path Of Main Event Nodes** and **Path Of Secondary Event Nodes** to the main events' 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 "/".
6. Check the **Secondary events located inside main events** check box if the **Secondary events** are nested inside primary events
7. Set the parameters **Channel node location** and **Date node location** to value **XML root** or **Secondary Events Nested inside Primary Events** according to the XML format.
8. Map each field relatively 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.

### 7.3.2 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. Your mapping for both events under the **XML Import Settings** section will be as follows:

- **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>
```

### 7.3.3 Secondary Events Nested Inside Primary Events

The following example shows a two-level XML where the secondary events are nested inside the primary events. Your mapping for both events under the **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>
```

## 7.3.4 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.

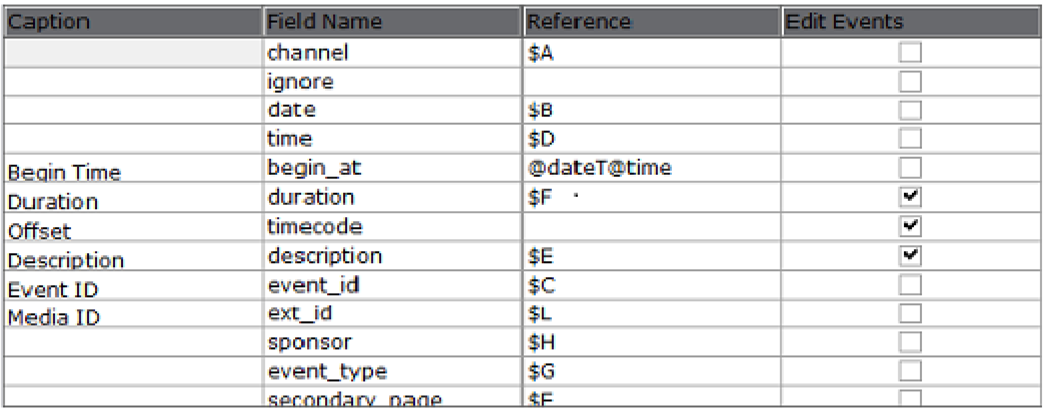
> ⚠ **Note:** Having the playlist date separate **limits the playlist to one broadcast day**.

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 **Primary Mapping Settings** section will be as follows:

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

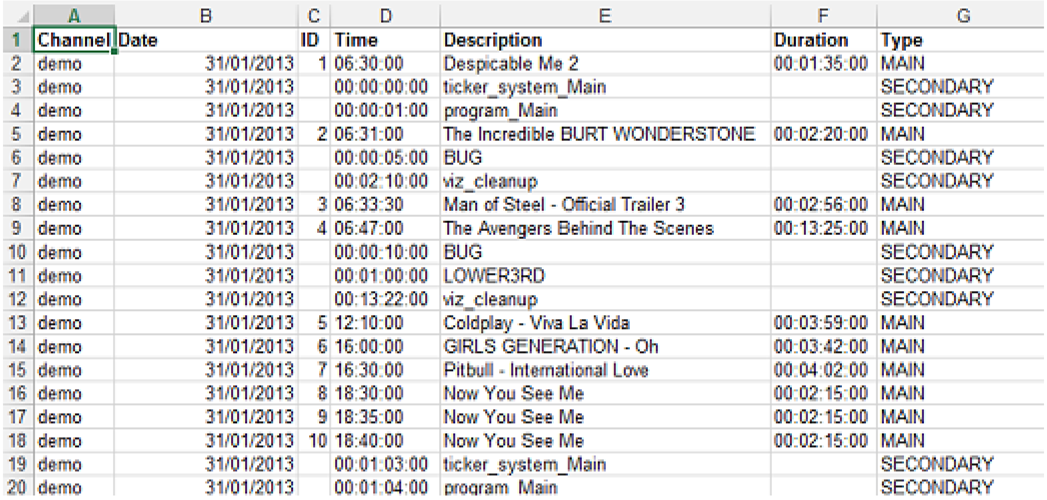
```
<playlist>
   <general_info>
     <channel_name>…</channel_name>
     <date>…</date>
   </general_info>
     <event>
    <begin_time time=”…” />
    <duration time=”…” />
   <title>…</title>
   <type>MAIN</type>
   </event>
</playlist>
```

## 7.4 Importing From Excel



| Caption | Field Name | Reference | Edit Events |
|---|---|---|---|
| | channel | $A | □ |
| | ignore | | □ |
| | date | $B | □ |
| | time | $D | □ |
| Begin Time | begin_at | @dateT@time | □ |
| Duration | duration | $F | ☑ |
| Offset | timecode | | ☑ |
| Description | description | $E | ☑ |
| Event ID | event_id | $C | □ |
| Media ID | ext_id | $L | □ |
| | sponsor | $H | □ |
| | event_type | $G | □ |
| | secondary_page | $F | □ |

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



| | 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 | Man 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 |
| 17 | demo | 31/01/2013 | 9 | 18:35:00 | Now You See Me | 00:02:15:00 | MAIN |
| 18 | demo | 31/01/2013 | 10 | 18:40:00 | Now You See Me | 00:02:15:00 | MAIN |
| 19 | demo | 31/01/2013 | | 00:01:03:00 | ticker_system_Main | | SECONDARY |
| 20 | demo | 31/01/2013 | | 00:01:04:00 | program_Main | | SECONDARY |

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

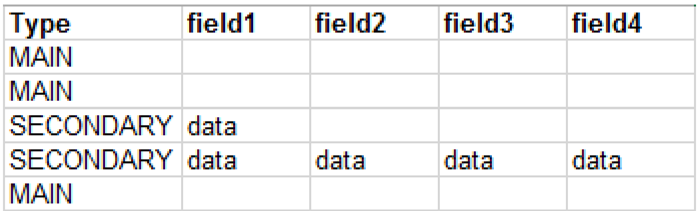
When working with Excel files the first row should always be your header row, and needs to span all columns in use. Also, the *Header Lines Included* option must be checked.

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`

You can also take a portion of the column with the following notation: `&(COLUMN,"DELIMITER",FROM,COUNT)`. If the delimiter is empty, it will split into characters. For example, if `L` is `31.12.2006`.

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

### 7.4.1 Secondary Fields in Excel



| Type | field1 | field2 | field3 | field4 |
|---|---|---|---|---|
| MAIN | | | | |
| MAIN | | | | |
| SECONDARY | data | | | |
| SECONDARY | data | data | data | data |
| MAIN | | | | |

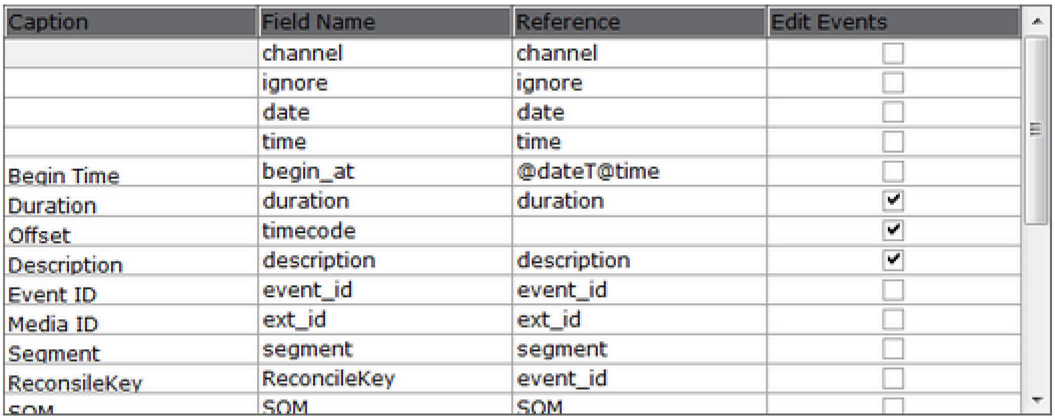
Data can be added to the schedule file that can be used by Viz Multichannel to fill pages with data. In order to achieve this you need to add in additional columns after the other columns in the Excel sheet.
If you are using column headers you also need to define column headers for the additional columns (e.g. field1, field2, and so on).

Viz Multichannel will start reading the fields, one by one, from that column until it gets to the first empty column header.

**IMPORTANT!** To use the Excel Import Type you need to install Microsoft® Excel on the machines that imports the schedule file.

**Note:** When importing from Excel files, there is a limitation of 100 columns.

## 7.5 Harris ADC-100



| Caption | Field Name | Reference | Edit Events |
|---|---|---|---|
| | channel | channel | □ |
| | ignore | ignore | □ |
| | date | date | □ |
| | time | time | □ |
| Begin Time | begin_at | @dateT@time | □ |
| Duration | duration | duration | ☑ |
| Offset | timecode | | ☑ |
| Description | description | description | ☑ |
| Event ID | event_id | event_id | □ |
| Media ID | ext_id | ext_id | □ |
| Segment | segment | segment | □ |
| ReconsileKey | ReconcileKey | event_id | □ |
| SOM | SOM | SOM | □ |

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

Using *.rcv files is basically only used when it is required to receive last-minute updates from the Harris automation system.

Before performing the procedure below you need to make sure a **schedule** is available in the required format (i.e. LST (Harris ADC-100) and that you have created a **channel**).

**Note:** Base your mapping on the sample content. See the directory `c:\Program Files (x86)\Vizrt\Viz Multichannel\SampleContent\LST`

### 7.5.1 To Import LST Format Files

⚙

1. Open the **Settings** window.
2. Select the **General Settings** section and set the **Channel Name in Schedule.**
   - This procedure uses channel _demo _as seen by the name of the schedule file.
3. Select the **Mapping Settings** section and set **Import Type** to **LST (Harris ADC-100).**
4. Use the **Mapping Table** to map the **Field Name** and **Reference** fields.
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 (e.g. `19_07_2011_demo`) and click **Open.**
   This will import the schedule and create a playlist in Viz Multichannel.
   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 > Full Update.**
   b. Select your schedule file (for example `19_07_2011_demo`) and click **Open.**

### 7.5.2 Mapping Table

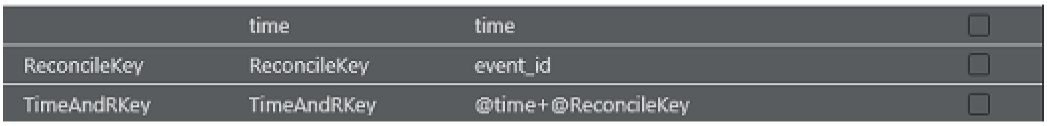
The mapping table for *.*lst* and *.*rcv* files must be as follows:

| Caption | Field Name | Reference |
|---|---|---|
| channel | channel | |
| ignore | ignore | |
| date | date | |
| Begin Time | begin_at | time |
| Duration | duration | duration |
| Offset | timecode | |
| Description | description | description |
| Event ID | event_id | event_id |
| Media ID | trigger_id | trigger_id |

| Caption | Field Name | Reference |
|---|---|---|
| Segment | segment | segment |
| ReconcileKey | ReconcileKey | event_id |
| SOM | SOM | SOM |
| Effect1 | effect1 | effect1 |
| Effect2 | effect2 | effect2 |
| Effect3 | effect3 | effect3 |
| secondary_page | secondary_page | |
| secondary_timecode | secondary_timecode | |
| secondary_event_id | secondary_event_id | |
| secondary_ext_id | secondary_ext_id | |
| event_type | event_type | |
| overlay_type | overlay_type | |

## 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. For combination-mapping, you need to use concatenation of the fields as **textFieldName**
To configure this navigate to **Multichannel > Settings > Channel Settings > Mapping Settings**



| | | | |
|---|---|---|---|
| | time | time | □ |
| ReconcileKey | ReconcileKey | event_id | □ |
| TimeAndRKey | TimeAndRKey | @time+@ReconcileKey | □ |

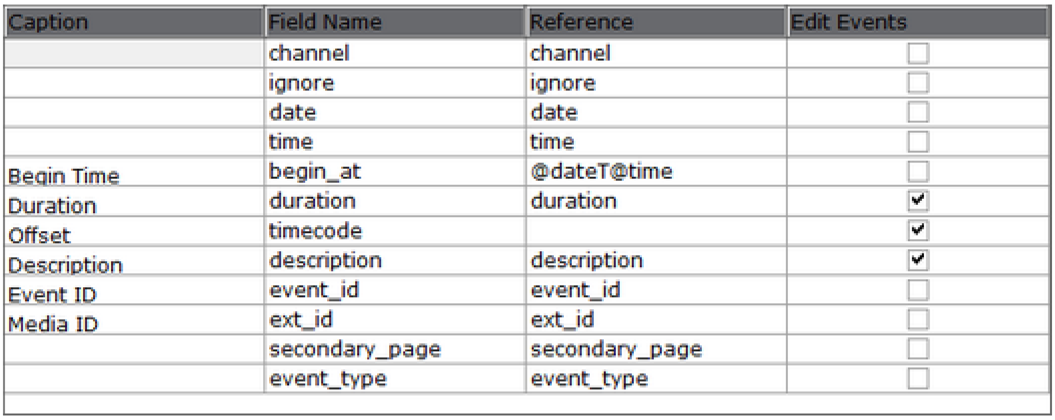
An example mapping settings can be found at `<Multichannel installation directory>\SampleContent\LST\LST_Combination_Mapping_Settings.xml`
In the screenshot above we have defined a new unique instruction named *TimeAndRKey* which concatenates fields *time* and *ReconcileKey*.

**To Support Concatenation**:

- Define "Field Name" *time* and *ReconcileKey* as: *time = time* and *ReconcileKey = event_id*
- Define new combination mapping field *TimeAndRKey* as *TimeAndRKey = @time+@ReconcileKey*

**Note**: The fields that are used for concatenation (time and ReconcileKey) should be defined before the concatenating field(TimeAndRKey)

## 7.6 Omnibus



| Caption | Field Name | Reference | Edit Events |
|---|---|---|---|
| | channel | channel | ☐ |
| | ignore | ignore | ☐ |
| | date | date | ☐ |
| | time | time | ☐ |
| Begin Time | begin_at | @dateT@time | ☐ |
| Duration | duration | duration | ☑ |
| Offset | timecode | | ☑ |
| Description | description | description | ☑ |
| Event ID | event_id | event_id | ☐ |
| Media ID | ext_id | ext_id | ☐ |
| | secondary_page | secondary_page | ☐ |
| | event_type | event_type | ☐ |

The Omnibus import module supports the *.osc file format. Each element starts with an `ITEM` line and ends with an `end` line. In between the element's fields are one in a line.

**Note**: It is **recommended** to base your mapping on the sample content.

```
ITEM
type 1
mode 2
channel_id demo
title Friends
clip CGIUJ101HS11A
preset_date 01/06/11
preset_time 06:30:00
in_src 982501
out_src 982751
user_data 0001-06_0000002
end
ITEM
mode 3
preset_time 00:00:00:00
overlay 1
clip 1190
title BUG_ON
type 1
end
```

As depicted above the rest of the fields are in a `<key><value>` format (e.g. `title Friends`).

## 7.7 Reading Channel And Date Values From Filenames

Whenever your collected schedule does not contain channel and date values you can extract that information based on the schedule's file name.

Viz Multichannel can read these two fields from the imported filename when using Importing from Text, Importing from XML, Importing from Excel, or Harris ADC-100.

In order to read these fields from the filename, set their mapping to *$Filename* and define the filename format in the **Schedule File Name Info** parameter in the **Mapping Settings**.

The file format should contain D/M/Y to represent the day/month/year 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 D/M/Y/C.

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

## 7.8 Merging Fields

The correct format for merging (joining) fields is:

```
freetext<at:var at:name="FIELDNAMEfreetext" />FIELDNAMEfreetext…
```

where *free-text* can be empty (`FIELDNAME@FIELDNAME`).

⚠ **Example:**`@Hour:@Minute:@Second`

The mapping is divided into two: one for each field reading its values, and one for concatenating (merging) them.

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

### 7.8.1 Example 1 for fixed width Any Text import type

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

### 7.8.2 Example 2 for delimited Any Text import type

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

- **Begin**= `@StartHour:StartSecond`

## 7.8.3 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`

# 8 Schedule Collector

The Schedule Collector communicates with the scheduling or the automation system, queries the current running automation-system’s schedule, and checks if anything is different from the last time queried. If a change is found, it creates an updated file for the Playlist Importer to update Viz Multichannel. The Schedule Collector then triggers the Playlist Importer to ingest the actual schedule.

Optionally, Viz Multichannel can be down and the live update functionality still runs (Media Sequencer and the Schedule Collector are prerequisites in this scenario).

There are minor configuration differences between one automation system and the next. In most cases, configuration can be done directly from the application. There are a few exceptions, where parameters can only be changed in the `.config` file.

Configuration is usually done only once, allowing the Schedule Collector to simply work in the background afterwards.
However, if you need to do **changes** simply **stop** it, **configure** it, and then **start** it again. The procedures for each automation system are described in the respective section below.

⚠ **Note:** The Schedule Collector runs as a service. To start, stop, and configure the service use the system tray icon.

## 8.1 Installation And Startup

Viz Multichannel’s Schedule Collector is a separate installer that installs itself as a service, and once installed is available from the system tray. To see the tray icon you need To run Schedule Collector once.

### 8.1.1 To Install Schedule Collector

1. Double-click the installer: `VizMultichannel_SCC<version>.msi` (For example: `VizMultichannel_SCC<XXX>.<bbbb>.msi` where `<XXX>` is the release number and *<bbbb>* is the build number).
2. Click **Next**.
3. *Optional*: Change the default installation path.
   - Windows XP: `%Program Files%\vizrt\Viz Multichannel\Tools\Schedule Collector`
   - Windows 7: `%ProgramData%\Vizrt\Multichannel\ScheduleCollector`
4. Click **Next**.
5. Click **Install**.
6. Click **Finish**.

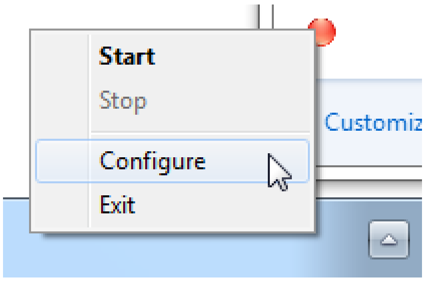
### 8.1.2 To Run Schedule Collector

1. Open the **Windows Start menu**.
2. Run **Viz Multichannel - Schedule Collector Monitor** found under Vizrt -> Viz Multichannel.

## 8.2 General Configuration

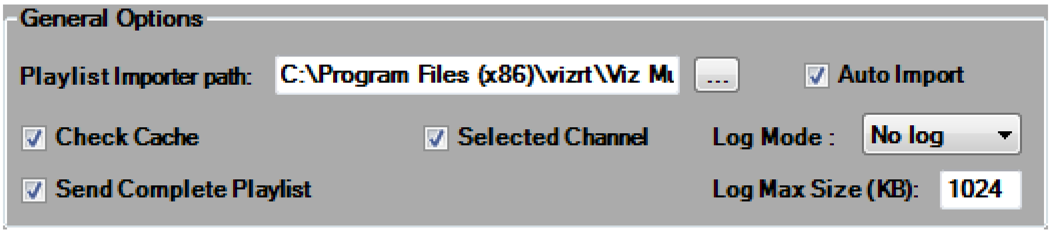
This section describes how to start configuring the Schedule Collector, and how to configure general options that apply to all automation systems.

### 8.2.1 To Configure Schedule Collector



- From the system tray, right-click the Schedule Collector icon and from the appearing context menu select **Stop** and then **Configure**.

### 8.2.2 To Configure General Options for Omnibus



General Options

Playlist Importer path: C:\Program Files (x86)\vizrt\Viz Mu ... ☑ Auto Import

☑ Check Cache ☑ Selected Channel Log Mode : No log

☑ Send Complete Playlist Log Max Size (KB): 1024

1. *Optional*: Enter the **Playlist Importer Path** if *Auto Import* will be enabled.
   Enter the path on the Viz Multichannel machine (usually the master control room client) where the Playlist Importer application is located.
2. *Optional*: Enable **Check Cache**.
   This will improve performance by updating changes only.
3. *Optional*: From the **Log Mode** drop-list, select if logs should be generated.
   - **No log**: No log is created (default).
   - **Basic log**: Logs basic activities.
   - **Full log**: Logs sent and received message contents. This option should only be used for troubleshooting. Deactivate once no longer needed.
4. *Optional*: Enable **Selected Channel**.
   - This will import only for the selected channel (defined in the Schedule Details section).
   - To import for all channels, uncheck it.
5. *Optional*: Enable **Send Complete Playlist** (*OmnibusBXF* only).
   If working in MCR mode (see Master Control Workflow), in the `ScheduleCollector.exe.config` file add the value `-msehost none` to `General.ImporterParameters`.

```
<add key="General.ImporterParameters" value="-msehost none" />
```

## 8.3 Integrating To Schedule Collector

See the section Schedule Collector Integrations for details of the additional configuration of the respective connections:

- Harris
- Multi DB
- Omnibus
- PBS
- SGT
- BXF Support

## 8.4 Schedule Collector Integrations

- DB
- Harris
- Multi DB
- VMC-Omnibus
- Omnibus BXF iTX Automation
- PBS
- SGT
- BXF Support

### 8.4.1 DB

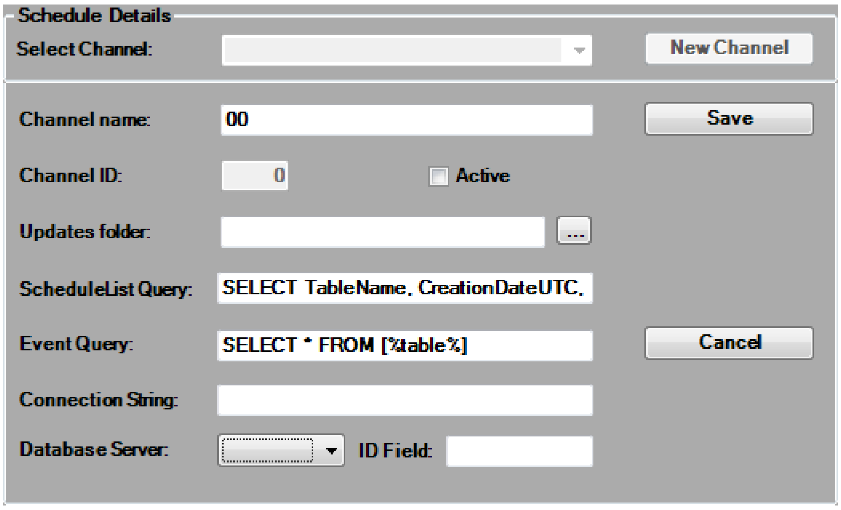
This section contains information on how to configure your DB setup for connecting to a Microsoft SQL, MySQL or Oracle database.

For information on how to configure **general options**, see the *General* section of Schedule Collector.

#### To Configure Connection Settings for DB

1. Open Schedule Collector.
2. Set **Automation** system to **DB.**
3. Set Heartbeat Interval (default is every 60 seconds) for how often the Schedule Collector should send a heartbeat query.

### To Configure Schedule Details for DB



Schedule Details

Select Channel:

Channel name: 00

Channel ID: 0

☐ Active

Updates folder:

ScheduleList Query: SELECT TableName, CreationDateUTC,

Event Query: SELECT * FROM [%table%]

Connection String:

Database Server:

ID Field:

1. Open Schedule Collector.
2. Click the **New Channel** button.
3. Set **Channel Name**.

   > **Note:** *Channel ID* is automatically set.

4. Set to **Active** to activate this channel setting.
5. Set the **Updates Folder** where the Schedule Collector will create its playlists.
6. Enter the **ScheduleList Query**.
   - This query must query the database for available schedule table names, creation dates and when last modified. The selected table name is then used by the **Event Query**.
   - Use the following syntax in your SQL statement (e.g. `SELECT TableName, CreationDateUTC, LastModificationTimeUTC FROM [table name]`).
7. Enter the **Event Query**.
   - This query must query the data from the selected schedule table name provided by the *ScheduleList Query*.
   - Use the following syntax in your SQL statement (e.g. `SELECT * FROM [%table%]`), where `[%table%]` is required.
8. Enter the database **Connection String**.
9. Select a **Database Server**:
   - Microsoft SQL
   - MySQL
   - Oracle
10. Enter the **ID Field** that stores the schedule table name.
11. Click **Save**.

## 8.4.2 Harris

This section contains information on how to configure your D-Series system from Harris Broadcast.

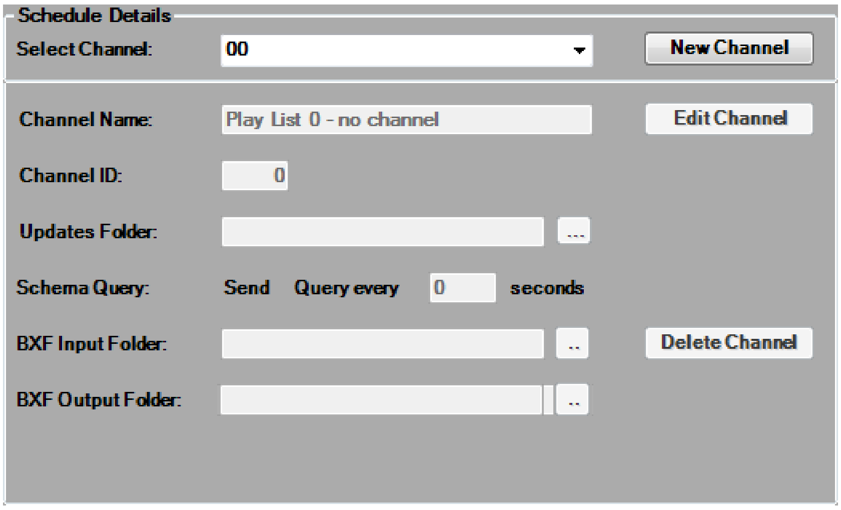
## To Configure Connection Settings for Harris

1. Open Schedule Collector.
2. Set **Automation** system to **Harris**.
3. Open Schedule Collector’s configuration file
    - Windows XP: `%ProgramFiles%\vizrt\Viz Multichannel\Tools\Schedule Collector\ScheduleCollector.config`
    - Windows 7: `%ProgramData%\Vizrt\Multichannel\ScheduleCollector\ScheduleCollector.config`
4. Set `General.ExtraConfigFiles` to **Bxf.config.xml**.

```
<add key="General.ExtraConfigFiles" value="Bxf.config.xml"/>
```

5. Save and close the file.

## To Configure Schedule Details for Harris



Schedule Details

Select Channel: 00

Channel Name: Play List 0 - no channel

Channel ID: 0

Updates Folder:

Schema Query: Send Query every 0 seconds

BXF Input Folder:

BXF Output Folder:

1. Open Schedule Collector.
2. Click the **New Channel** button.
3. Enter the **Channel Name** as defined by the *Harris automation system*.
4. Enter the unique **Channel ID** for the channel as defined by the *Harris automation system*.
    - Ensure that Channel ID is identical to Viz Multichannel’s **Channel Name in Schedule.**
5. Enter the **Updates Folder** path.
    - This is the path where updates will be received. Make sure you select an existing folder as Schedule Collector will not create on for you.
6. For the **Schema Query**, under **Query Interval**, enter the desired interval in seconds.
7. In the **BXF Input Folder** field, enter the path for protocol input files. This must be an existing folder, and must match the corresponding folder in the Harris automation system.
8. In the **BXF Output Folder** field, enter the path for protocol output files. This must be an existing folder, and must match the corresponding folder in the Harris automation system.
9. Click **Save**.
10. Repeat this procedure for each Harris channel, as required.

### 8.4.3 Multi DB

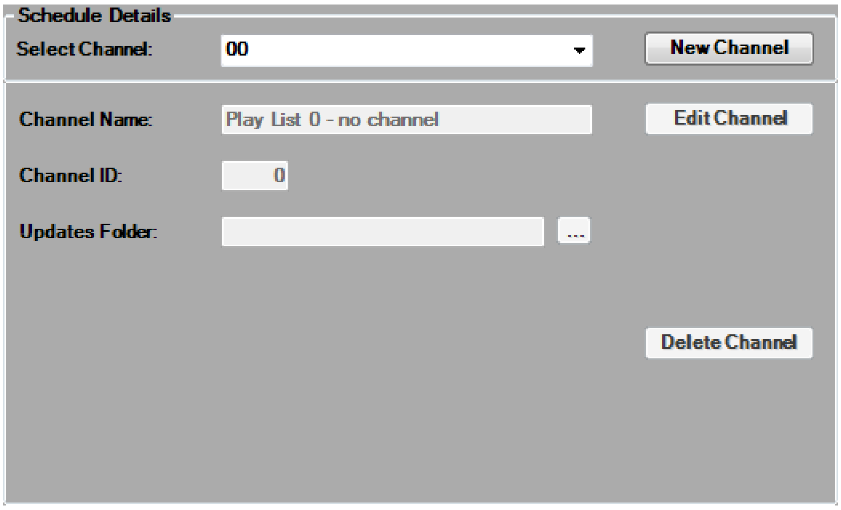
This section contains information on how to configure your DB setup for connecting to a Microsoft SQL, MySQL or Oracle database in case you have multiple database instances running on your server (and you do not want to use the default database).

For information on how to configure **general options**, see the *General* section of Schedule Collector.

#### To Configure Connection Settings for Multi DB

1. Open Schedule Collector.
2. Set **Automation** system to **Multi DB**.
3. Set **Heartbeat Interval** (default is every 60 seconds) for how often the Schedule Collector should send a heartbeat query.

#### To Configure Schedule Details for Multi DB



Schedule Details

Select Channel: 00

Channel Name: Play List 0 - no channel

Channel ID: 0

Updates Folder:

1. Create an XML file and name it **Extra.config.xml**.
2. Add the XML file to the **SCC folder** in your Viz Multichannel installation directory.
3. Use the **sample XML** below to configure a connection to your multi DB instance. Example:

```
<MultiDBConfigs>
  <QueryIDField>EventUniqueID</QueryIDField>
  <ConnectionString>Data Source=hostname;Initial
Catalog=FLORICAL_SCHEDULES_MAX;User Id=name;Password=123456;</ConnectionString>
  <QueryForOnAirEvent>SELECT * FROM [FLORICAL:ARL:MAX1:2011:02:25]</
QueryForOnAirEvent>
  <QueryForEvent>SELECT * FROM [FLORICAL:ARL:MAX1:2011:02:25]</QueryForEvent>
</MultiDBConfigs>
```

4. Set **QueryIDField** to be the column name in the database that stores Event IDs.
5. Set **ConnectionString** to be specify the database instance you are connecting to.
6. Set **QueryForOnAirEvent** to query for on-air events.
7. Set **QueryForEvent** to query for a list of events to create the playlist.

8. Open Schedule Collector.
9. Click the **New Channel** button.
10. Set **Channel Name**.

> ⚠ **Note:** *Channel ID* is automatically set.

11. Set to **Active** to activate this channel setting.
12. Set the **Updates Folder** where the Schedule Collector will create its playlists.
13. Enter the database **Connection String**.
14. Set the **Small** and **Big intervals** for how many and how often Schedule Collector should query the database for new or updated events.
15. Select a **Database Server**:
    - Microsoft SQL
    - MySQL
    - Oracle
16. Click **Save**.

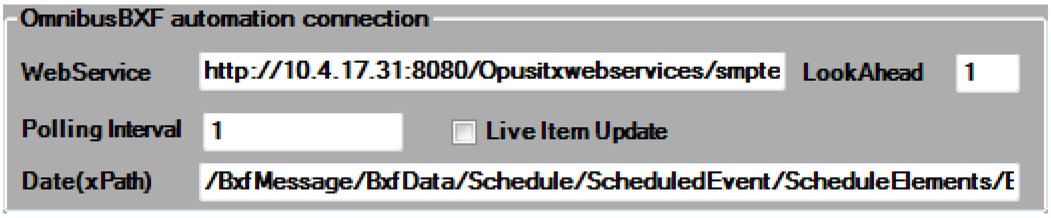
## 8.4.4 VMC-Omnibus

You can configure your Omnibus systems from Miranda Technologies; Automation can be configured in one of two ways:

- Omnibus General Automation
- Omnibus BXF iTX Automation

### Omnibus General Automation

### To Configure Connection Settings for Omnibus



OmnibusBXF automation connection

WebService: http://10.4.17.31:8080/Opusitxwebservices/smpte

LookAhead: 1

Polling Interval: 1

□ Live Item Update

Date(xPath): /BxfMessage/BxfData/Schedule/ScheduledEvent/ScheduleElements/E

1. Open Schedule Collector.
2. Set **Automation** system to **Omnibus**.
3. Enter the Omnibus machine’s **Host Address**.
4. Enter the listener **Port** number, default is `10540`.
5. In the **Views List** field, enter a comma-separated list of the views defined in the Omnibus Adaptor for the relevant channels.

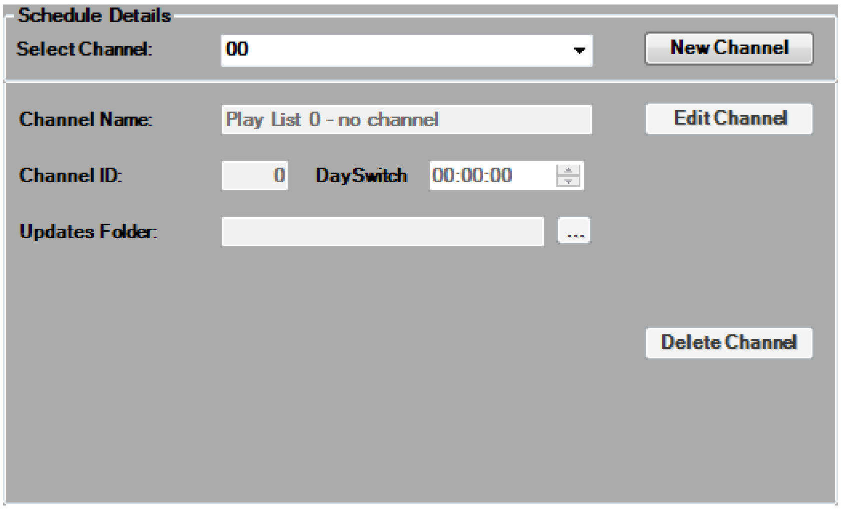
### To Configure Update Settings for Omnibus

1. Open Schedule Collector’s configuration file
    - Windows XP: `%ProgramFiles%\vizrt\Viz Multichannel\Tools\Schedule Collector\ScheduleCollector.config`

- Windows 7: `%ProgramData%\Vizrt\Multichannel\ScheduleCollector\ScheduleCollector.config`

2. Set `General.OmnibusConnectionMode` to **IntervalsConnection** or **ConstantConnection**.
    - Constant connection: Schedule Connector stays connected until you choose to disconnect.
    - Intervals connection: Schedule Connector connects, gets the full schedule and disconnects for every interval.
3. Set `General.CacheIgnoreFields` to **story/mosExternalMetadata/mosPayLoad/currentTime,story/mosExternalMetadata/mosPayLoad/status**.
4. Save and close the file.

## To Configure Schedule Details for Omnibus



Schedule Details

Select Channel: 00

Channel Name: Play List 0 - no channel

Channel ID: 0

Day Switch: 00:00:00

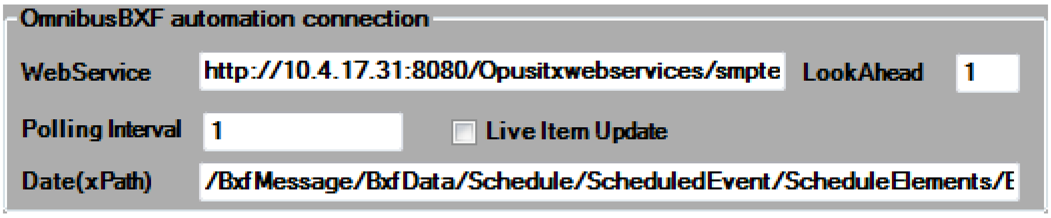
Updates Folder:

1. Open Schedule Collector.
2. Click the **New Channel** button.
3. Enter the **Channel Name** as defined by the *Omnibus Adaptor*.
4. Enter the unique **Channel** and **Backup Channel IDs** for the channel as defined by the *Omnibus Adaptor*.
    - Ensure that Channel ID is identical to Viz Multichannel's **Channel Name in Schedule** (see the *General Settings* section).
5. Enter the **Updates Folder** path.
    - This is the path where updates will be received. Make sure you select an existing folder as Schedule Collector will not create on for you.
6. Click **Save**.

> ⚠ **Note:** Repeat the above procedure for each Omnibus channel, as required.

## 8.4.5 Omnibus BXF iTX Automation

### To Configure Connection Settings for Omnibus BXF



OmnibusBXF automation connection

WebService: http://10.4.17.31:8080/Opusitxwebservices/smpte

LookAhead: 1

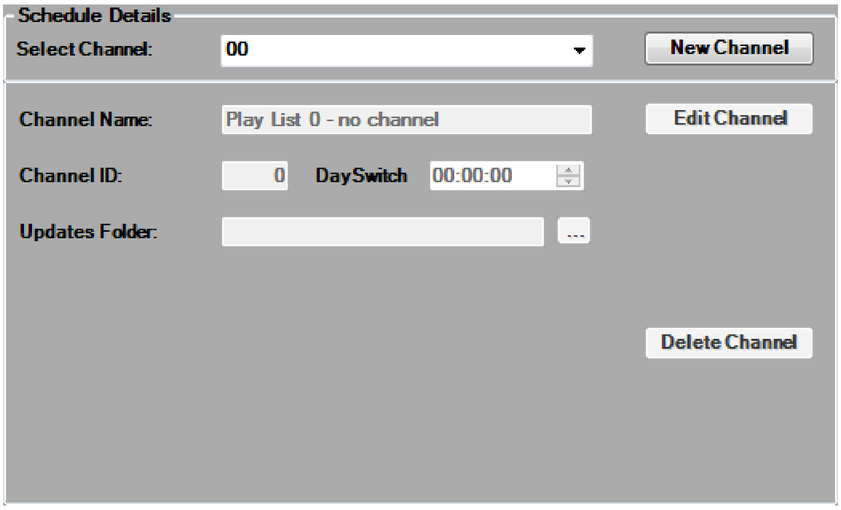
Polling Interval: 1

□ Live Item Update

Date(xPath): /BxfMessage/BxfData/Schedule/ScheduledEvent/ScheduleElements/E

1. Open Schedule Collector.
2. Set **Automation** system to **OmnibusBXF**.
3. Enter the link to your **Web Service**.
4. Enter the number of days to get the playlists for in the **LookAhead** field.
5. Enter a value to determine the **Polling Interval**, in seconds, to poll the automation system for changes.
6. *Optional*: Enable **Live Item Update** if you want to receive live updates from the your web service.
7. Enter the **Date(xPath)** path in the XML that defines the event date.
    - Example: `/BxfMessage/BxfData/Schedule/ScheduledEvent/ScheduleElements/EventData/StartDateTime/SmpteDateTime/@broadcastDate`
    - See *Playlist Structure* for a more complete XML example.

### To Configure Schedule Details for Omnibus BXF



Schedule Details

Select Channel: 00

Channel Name: Play List 0 - no channel

Channel ID: 0

DaySwitch: 00:00:00

Updates Folder:

1. Click the **New Channel** button.
2. Enter the **Channel Name** as defined by the *Omnibus Adaptor*.
3. Enter the unique **Channel ID** for the channel as defined by the *Omnibus Adaptor*.
    - Ensure that Channel ID is identical to Viz Multichannel’s **Channel Name in Schedule.**
4. In the *Day Switch *field, enter the day switch as defined in the Day Switch Options.
5. Enter the **Updates Folder** path.

   - This is the path where updates will be received. Make sure you select an existing folder as Schedule Collector will not create on for you.

6. Click **Save**.
7. Repeat this procedure for each Omnibus BXF channel, as required.

## Playlist Structure

```
<BxfMessagae>
    <BxfData>
        <Schedules>
            <Channel><Name>AChannel</Name></Channel>
                <Scheduled>
                    <EventData>
                        <EventId>id</EventId>
                        … (other info)
                    </EventData>
                </Scheduled>
            </Schedules>
        </BxfData>
</BxfMessagae>
```

Channel name and event ID are highlighted, and checked by the Schedule Collector. All the rest of the event information and is checked in Viz Multichannel according to the mapping.

Another alternative is to use different field names instead of the Schedules/Scheduled/EventData is Schedule/ScheduledElements/EventData. However, it is worthwhile to run tests with examples to verify first. Sample structures can be found in Harris documentation.

An example of the second type with primary/secondary events:

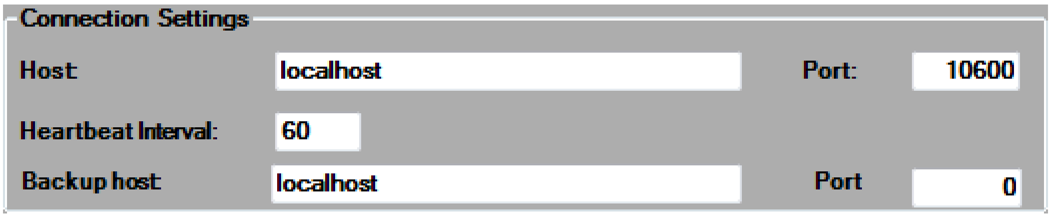
```
<BxfMessagae>
<BxfData>
    <Schedule>
    <Channel><Name>AChannel</Name></Channel>
    <ScheduledEvent>
        <ScheduleElements>
            <EventData eventType=”Primary-ProgramHeader”> //Primary event
                <EventId>
                    <EventId>id</EventId>
                </EventId> //GUID
                <PrimaryEvent>
                    <ProgramEvent>
                        <SegmentNumber>1</SegmentNumber>
                        <ProgramName>Eve News</ProgramName> //Title
                    </ProgramEvent>
                </PrimaryEvent>
                <StartDateTime></StartDateTime> //Smpte date-time format
                <LengthOption>
                    <Duration></Duration> //Smpte time format
                </LengthOption>
                <StartMode>Fixed</StartMode>
                <EndMode>Duration</EndMode>
                <PrivateInformation></PrivateInformation> //Custom data area. E.g.
you can place reconcilekey or anything you like
            </EventData>
            <Content></Content> //SOM, Duration, and other info
            </ScheduleElements><ScheduledElements>
            <EventData eventType=”NonPrimary”> //Secondary event
                <NonPrimaryEvent></NonPrimaryEvent> //Similar to primary – holds id,
offset details, other info
            </EventData>
            </SceduledElements></ScheduledEvent>
        </Schedule>
</BxfData>
</BxfMessagae>
```

An example for the SMPTE date-time format:

```
<StartDateTime nominalFlag="false">
    <SmpteDateTime broadcastDate="2006-10-01">
        <SmpteTimeCode>15:52:30:00</SmpteTimeCode>
    </SmpteDateTime>
</StartDateTime>
```

## 8.4.6 PBS

### To Configure Connection Settings for PBS



Connection Settings

Host: localhost

Port: 10600

Heartbeat Interval: 60

Backup host: localhost

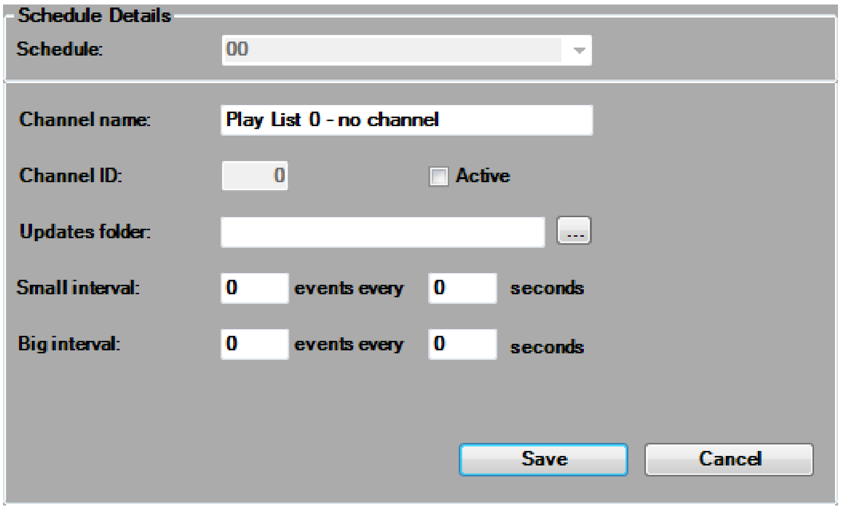
Port 0

1. Open Schedule Collector.
2. Set **Automation** system to **PBS**.
3. Enter the PBS machine's **Host**.

> ⚠ **Note:** The host machine will always be given priority in connections.

4. Enter the listener **Port** number, default is `10600`.
5. Enter the **Heartbeat Interval time**, in seconds.
   - This allows the Schedule Collector to indicate to the PBS automation system it is still "alive", even if no updates have been requested.
6. Enter the PBS backup machine's **Backup host.**

### To Configure Schedule Details for PBS



Schedule Details

Schedule: 00

Channel name: Play List 0 - no channel

Channel ID: 0

□ Active

Updates folder:

Small interval: 0 events every 0 seconds

Big interval: 0 events every 0 seconds

From the **Schedule** drop-list, select a schedule.

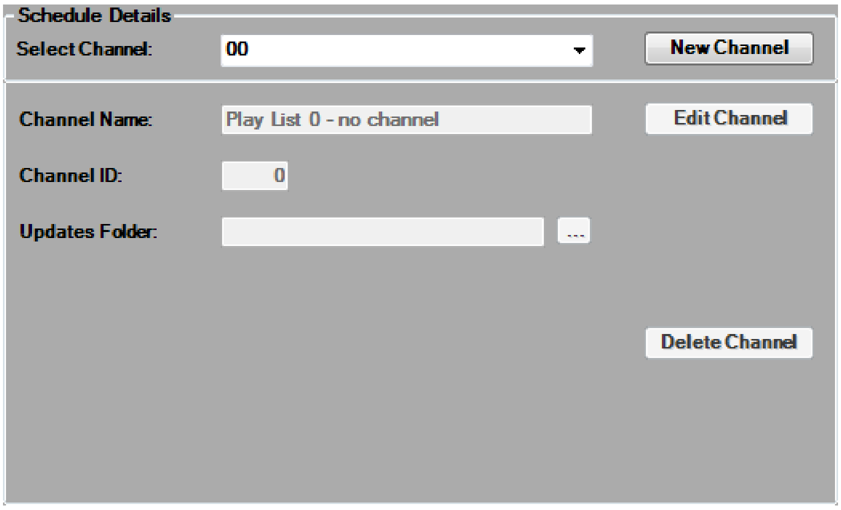
1. Click **Edit Channel.**
2. Enter the **Channel name** as defined by the *PBS automation system.*
3. Enter the unique **Channel ID** for the channel as defined by the *PBS automation system.*
   - Ensure that Channel ID is identical to Viz Multichannel's **Channel Name in Schedule.**
4. *Optional*: Enable **Active.**
   - This will enable automatic updates.

5. Enter the **Updates folder** path.
    - This is the path where updates will be received. Make sure you select an existing folder as Schedule Collector will not create on for you.
6. For minor but frequent updates, under **Small interval**, enter the number of events to be received within an indicated time frame. Zero events and/or zero seconds disables this update option.
7. For major but infrequent updates, under **Big interval**, enter the number of events to be received within an indicated time frame. If also updating minor events, ensure that the Big Interval is greater than the Small Interval. Zero events and/or zero seconds disables this update option.
8. Click **Save**.

> ⚠ **Note:** Repeat the above procedure for each PBS channel, as required.

## 8.4.7 SGT

### To Configure Connection Settings for SGT



Schedule Details

Select Channel: 00

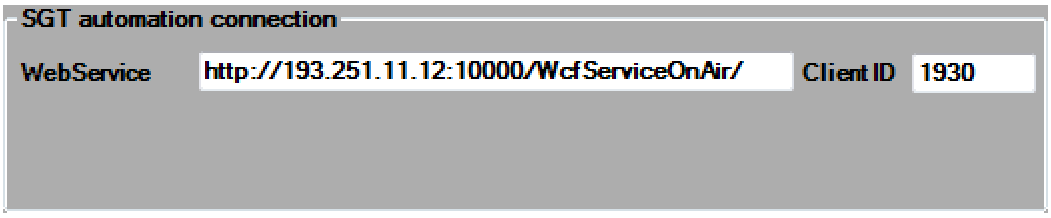
Channel Name: Play List 0 - no channel

Channel ID: 0

Updates Folder:

1. Open Schedule Collector.
2. Set **Automation** system to **SGT**.
3. Enter the link to your **Web Service**.
4. Enter the **Client ID**.

### To Configure Schedule Details for SGT



SGT automation connection

WebService http://193.251.11.12:10000/WcfServiceOnAir/ Client ID 1930

1. Open Schedule Collector.
2. Click the **New Channel** button.
3. Enter the **Channel Name** as defined by the *SGT automation system*.
4. Enter the unique **Channel ID** for the channel as defined by the *SGT automation system*.
    - Ensure that Channel ID is identical to Viz Multichannel's **Channel Name in Schedule.**
5. Enter the **Updates Folder** path.
    - This is the path where updates will be received. Make sure you select an existing folder as Schedule Collector will not create on for you.
6. Click **Save**.

## 8.4.8 BXF Support

All messages come with root tag of `<bxfmessage>`. This root message comes with various information, like origin m+target machine, user name and others. Viz Multichannel supports these fields, and rejects messages aimed to other machines, but does not require them.Viz Multichannel can also check that this info matches the bxf file name (also according to the spec) but this is also not a requirement.

One important field is `messageType`. Viz Multichannel identifies `Heartbeat/Information/Reply/Acknowledgement/` from it. If the message is `Heartbeat/Reply/Acknowledgement` then the field `originId` must match an ID of a `Heartbeat/Query` that Viz Multichannel has sent to the automation. When Viz Multichannel replies to such messages, it also includes the correct `originId` in the reply.

In short, every message (from both sides) is automatically answered by `Acknowledge`. Viz Multichannel receives information messages, or sends a `Query` and receives a `Reply` (both with playlist updates). Viz Multichannel also handle `Heartbeats`.

The following types of messages are handled:

- Viz Multichannel receives `Hearbeat` -> Viz Multichannel returns `Acknowledge`.
- Viz Multichannel sends `Heartbeat` -> Viz Multichannel accepts its `Acknowledge`.
- Viz Multichannel sends `Query` for update -> Viz Multichannel accepts `Acknowledge` for it.
- Viz Multichannel receives `Information` with playlist -> Viz Multichannel sends `Acknowledge` for it (and processes the playlist).
- Viz Multichannel receives `Reply` with playlist (answer to Query Viz Multichannel sent) - > Viz Multichannel sends `Acknowledge` for it (and processes the playlist).
- Viz Multichannel receives `Query` -> Viz Multichannel sends `Acknowledge` (reject).
- Viz Multichannel receive any other message -> Viz Multichannel sends `Acknowledge` (reject).

# 9 Playlist Importer

The Playlist Importer (PLI) includes all import logic of Viz Multichannel. The Playlist Importer can be run automatically by the Media Sequencer, or by the Schedule Collector. The purpose of the Playlist Importer is to convert a schedule, received for example from an automation system, to a native Media Sequencer playlist format using mapping settings configured by the Viz Multichannel client.

The normal workflow for v1 is that Viz Multichannel connects to the Media Sequencer (MSE), while v2 connects against global integration point Integrations Hub. If using v2 there is a need to install Integrations Hub alongside the Media Sequencer on the Engine machine.

> ⚠ From release v4.1 the 64-bit version (referred to as v2 in below notes) is the preferred version of the Playlist Importer. The 32-bit version is deprecated and may be removed in coming releases.

## 9.1 Installation And Configuration

Viz Multichannel's Playlist Importer is a separate installer and can be installed or upgraded without upgrading the core of the full Viz Multichannel installation.
For detailed information on how to install Viz Multichannel, see the Setup and Configuration chapter.

> ⚠ **Note:** The PlaylistImporter (PLI) installer is typically named (v1) `VizMultichannelPlayListImporter-VERSION.exe` and (v2) `VizMultichannel_PlaylistImporterV2-VERSION.exe` where *VERSION* indicates the version/release and build numbers of the software.
> There will also be a corresponding <identical_filename>.MD5 file that can be used to check that the downloaded installer has a correct md5-checksum.
> It is recommended to always check that the downloaded installers have the correct checksum.
> Normally installers will be downloaded with ftp from download.vizrt.com using your customer support login credentials.

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

> ⚠ **IMPORTANT!** The location of the *PlaylistImporter.exe* program must be specified in **Settings > Channel > Global Settings > Import/Export Settings > Automatic Import > Playlist Importer Folder Path**

For this option to work, the `PlaylistImporter.exe` file must be installed on the Media Sequencer machine. Usually Viz Multichannel is *not* installed on the Media Sequencer machine, so you *must install the* Playlist Importer *there*. The normal installation directories for Play List Importer is:

Playlist Importer v1: `C:\Program Files (x86)\Vizrt\Viz Multichannel\PlayList Importer`

Playlist Importer v2: `C:\Program Files\Vizrt\Viz Multichannel\Tools\PlaylistImporterV2`

Play List Importer v2 is a 64 bit utility. The earlier (v1) Play List Importer was 32 bit.

### 9.1.1 PlaylistImporter Log File

The PlaylistImporter (PLI) application writes a log file when executing. The log file is normally written to the directory: %ProgramData%\Vizrt\Multichannel\PlaylistImporterV2

**Note:***%ProgramData%* is an environment variable, the fully expanded path will often be `C:\ProgramData\Vizrt\Multichannel\PlaylistImporterV2`

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 Multichannel configuration settings found in **Settings > Channels > Global Settings**.

For changes in the log-level to take effect, a playlist must be stopped and re-activated.

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

## 9.2 Viz Multichannel Startup Parameters

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

| 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` |
| `-msehost` | Host name or IP address for Media Sequencer. When host name is "none" Playlist Importer will read each channel's Media Sequencer from `channels.ini`.<br>When no parameter is provided, Playlist Importer will attempt to connect to *localhost*. | `-msehost none-msehost 10.11.10.11-msehost localhost` |

| Parameter | Description | Example |
|---|---|---|
| `--refresh` | PLI will only refresh current running playlist. It will not import/ update the playlist | `--refresh` |
| -viz2 | Allows you to use Viz Engine 2.x. Viz 3.x is default. Version 3 is recommended | |

## 9.3 Automatic Import Of Files

Automatically imported files, using 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.

## 9.4 Troubleshooting

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

# 10 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 Multichannel.

In addition, the Integrations Hub provides a REST-based interface for services and components to integrate with Viz Multichannel.

The Integrations Hub offers Direct Integration modules provided by Vizrt, and the REST interface enables customers and 3rd party system vendors and integrators to create additional services using this interface.

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

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 introduction article REST (external link) can be a nice starting point.

This section presents:

- Installation, Requirements and Configuration Check
- Selecting the Integrations Hub Media Sequencer

> ⚠ For REST documentation, see sections
> - REST Interface for Integrations Hub Version 1.0
> - REST Interface for Integrations Hub Version 2.0

## 10.1 Installation, Requirements And Configuration Check

The Integrations Hub service requires .NET Framework 4.6.1 or higher. If the server is behind a corporate firewall *in a secure environment*, is it recommended to change User Account Control (UAC ) Settings to *off* and also turn *off* Windows Firewall.

To Install the Integrations Hub

1. Locate the current Integrations Hub installer from your Vizrt representative or download.vizrt.com - `VizMultichannel_IntegrationsHub-VERSION.exe` here `VERSION` will be numbers indicating the version of the software.
2. Run the installer software, selecting the default installation directory.

   > ⚠ **Important!**
   > The Integrations Hub run 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 service is open at 9990 port.

3. Check the Integrations Hub installation:
   Browse to http://localhost:9990/restapi
   Expected response: HTTP-answer listing all channels

No further configuration of the Integrations Hub is required for normal operation unless the Integrations Hub should connect to different Media Sequencer host than *localhost.*

> **Note:** Status message from Integrations Hub will be stored in 'infoIntegrationsHub' attribute of an event.

---

## 10.2 Selecting The Integrations Hub Media Sequencer

By default, the Integrations Hub service connects to the local Media Sequencer, i.e. the Media Sequencer running at localhost = 127.0.0.1. If required, follow 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 Multichannel\Tools\Integrations Hub _

   > `\ProgramData` is normally a hidden Windows directory so you have 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:

   ```
   <appSettings>
        <add key="MseHost" value="localhost" />
        <add key="ContentFillingWaitTime" value="2" />
        <add key="ChannelCacheTime" value="30"/>
    </appSettings>
   ```

   **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, options set in the Multichannel UI could be up to 60 seconds delayed before acted upon.
   Use of cache can improve throughput in an application, but could have hard to catch side effects due to delay, so always consider your options carefully.

4. Save the file.
5. Restart the windows service.

**See Also**

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

## 10.3 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.

> Version 2.0 of the REST Interface for Integrations Hub is 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.

### 10.3.1 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>
```

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

## 10.3.3 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="application/vizrt.integrationshub+xml; type=pagepool" />
</pagepool>
```

## 10.3.4 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="application/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>
    <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>
```

## 10.3.5 GET Channel Pool Playlist 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="application/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" />
    <begin_at />
    <duration>600</duration>
    <event_id />
    <trigger_id />
  </group>
</playlist>
```

## 10.3.6 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="application/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="entry#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>
```

## 10.3.7 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">
      <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>
```

## 10.3.8 GET Pages from a Group (Version 1.0)

### GET Pages from a Group

To obtain pages 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]_pool/[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>
```

## 10.3.9 POST Channel (Version 1.0)

### POST Channel

A Channel can only be added via the Viz Multichannel client.

### Syntax

```
POST /restapi/
```

### Request

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

### Reponse

Returns a **403 Forbidden** status message.

## 10.3.10 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" />
</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/MovieChannel/
channel_pool</title>
  <description>Channel Pool</description>
  <name>channel_pool</name>
  <ischannelpool>true</ischannelpool>
  <link href="http://localhost:9990/restapi/MovieChannel/channel_pool" rel="self"
 type="application/vizrt.integrationshub+xml; type=playlistpostreply" />
  <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="self" 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="take_duration">
        <value>50</value>
      </attribute>
      <attribute name="name">
        <value>page1</value>
      </attribute>
    </attributes>
    <fields />
  </page>
</playlistpostreply>
```

## 10.3.11 POST Playlist to a Channel (Version 1.0)

### POST Playlist to a Channel

Playlists can only be added via the Viz Multichannel client.

### Syntax

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

### Request

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

### Reponse

Returns a **403 Forbidden** status message.

## 10.3.12 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.

## 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="2014-06-13T01:00:00.000" />
  <group name="group2" event_id="2" description="group 2" duration="3600"
 begin_at="2014-06-13T00:00:00.000" />
</payload>
```

### 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="self" type="application/vizrt.integrationshub+xml; type=group" />
    <begin_at>2014-06-13T01:00:00.000</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="self" type="application/vizrt.integrationshub+xml; type=group" />
    <begin_at>2014-06-13T00:00:00.000</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>
```

## 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="2014-06-13T01:00:00.000">
    <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="self" type="application/vizrt.integrationshub+xml; type=group" />
    <begin_at>2014-06-13T01:00:00.000</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>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

```
<group name="group1"
isLive="true"
isLocked="true"
event_id="1"
description="group 1"
duration="3600"
begin_at="2016-06-13T01:00:00.000"
/>
```

## 10.3.13 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="take_duration">
        <value>30</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 />
   </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>
```

## 10.3.14 DELETE All Channels (Version 1.0)

### DELETE All Channels

Channels can only be deleted via the Viz Multichannel client.

### Syntax

```
DELETE /restapi/
```

### Request

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

### Response

Returns a **403 Forbidden** status message.

## 10.3.15 DELETE Channel (Version 1.0)

### DELETE Channel

A Channel can only be deleted via the Viz Multichannel client.

### Syntax

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

### Request

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

### Reponse

Returns a **403 Forbidden** status message.

## 10.3.16 DELETE Playlist (Version 1.0)

### DELETE Playlist

Playlists can only be deleted via the Viz Multichannel client.

### Syntax

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

### Request

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

### Reponse

Returns a **403 Forbidden** status message.

## 10.3.17 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.

## 10.3.18 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
```

### Reponse

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.

## 10.3.19 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

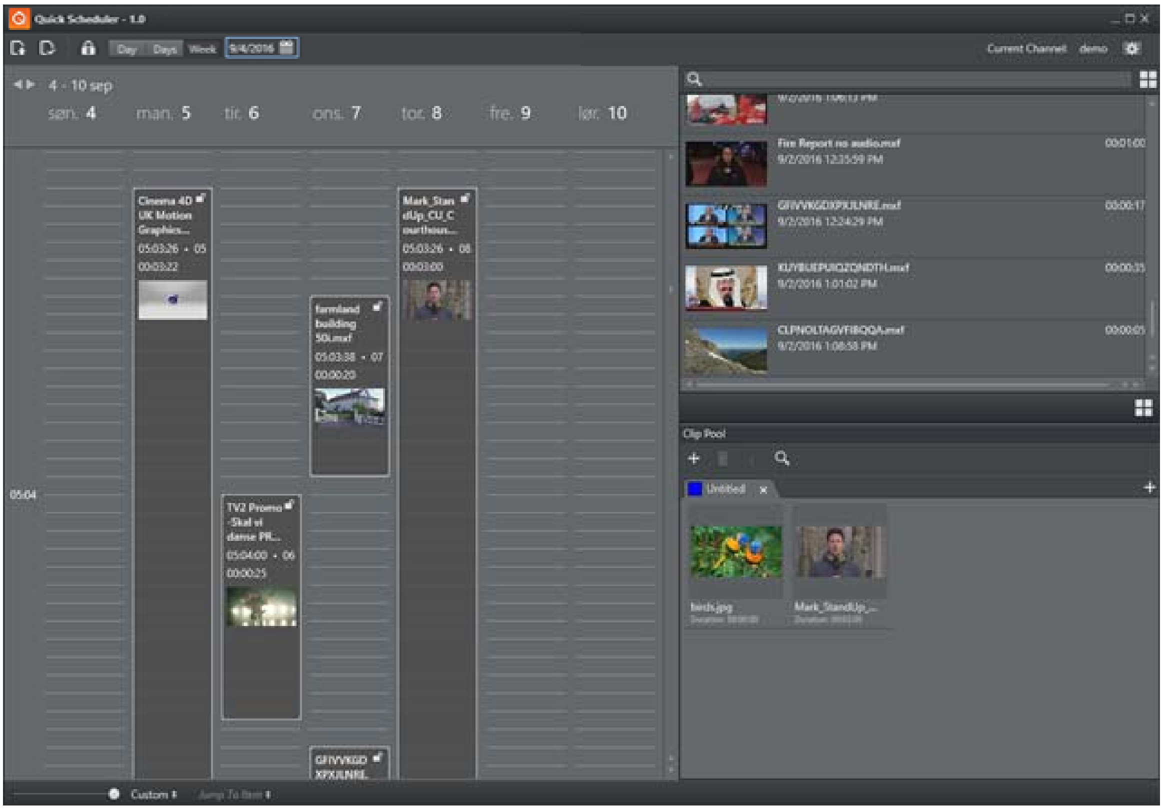
```
GET /GetEventsByEventId/test/continuous/070bf70c-0dbd-4fa6-a609-71bd34d6361c HTTP/1.1
```

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

# 11 Quick Scheduler

Quick Scheduler (QS) is a tool for creating schedules. A *schedule* in this context can be understood as a playlist with items or complete run-downs.
Schedules created with Quick Scheduler can be published to and used by Viz Multichannel using the Quick Scheduler Server.



- Installation and Startup
- Configuration
- Using Quick Scheduler

## 11.1 Installation And Startup

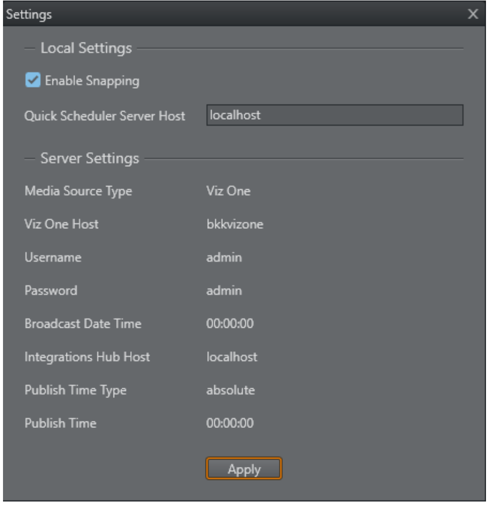
1. Install Quick Scheduler using the MSI installer or install QS as part of a bundled (wrapped) installer by fetching the file
   `VizMultichannel_QuickScheduler_<VERSION>.msi`
   from https://download.vizrt.com
2. Depending on your Windows version, start Quick Scheduler by navigating to
   **[Windows Start button] > Vizrt > Viz Multichannel - Quick Scheduler**

## 11.2 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 of Quick Scheduler Server.

  Other settings are read only. They are settings from Quick Scheduler Server side.



Settings

Local Settings

☑ Enable Snapping

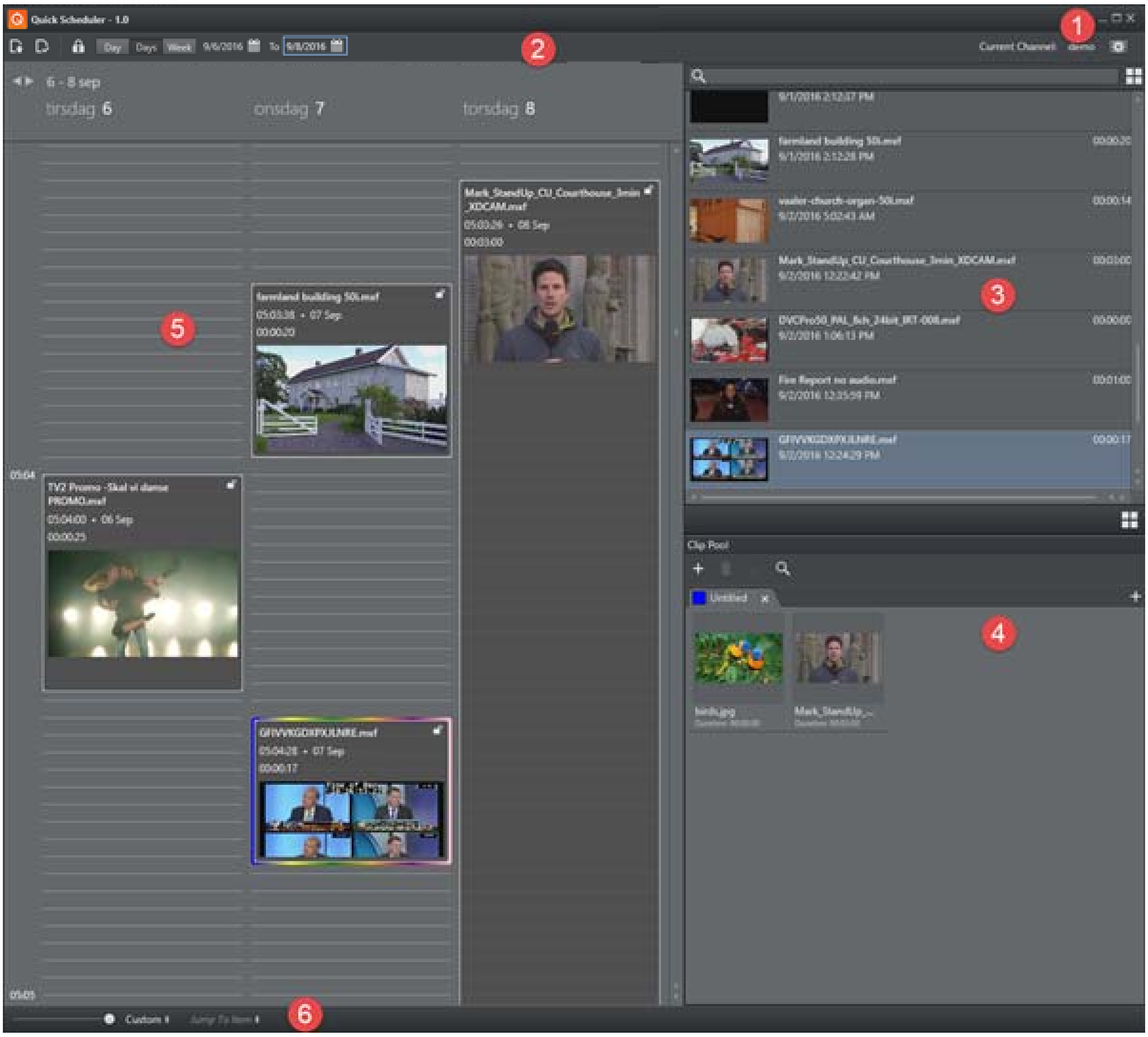
Quick Scheduler Server Host: localhost

Server Settings

| | |
|---|---|
| Media Source Type | Viz One |
| Viz One Host | bkkvizone |
| Username | admin |
| Password | admin |
| Broadcast Date Time | 00:00:00 |
| Integrations Hub Host | localhost |
| Publish Time Type | absolute |
| Publish Time | 00:00:00 |

## 11.3 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.



[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

[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:

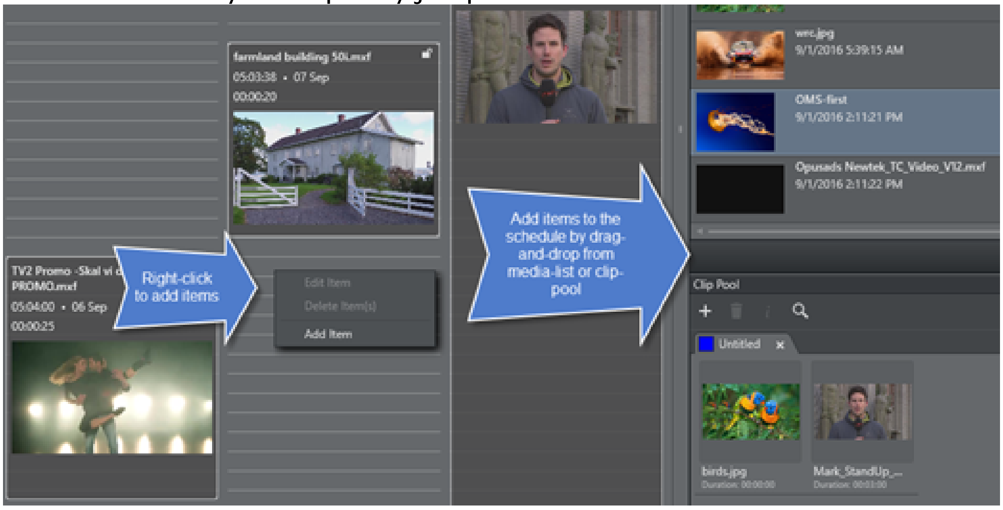
**Ctrl + Wheel UP**: Zoom in

**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 the *dynamic* (the list is updated depending on what is in the Schedule) **Jump to Item**. This allows you to quickly jump to items in the Schedule.

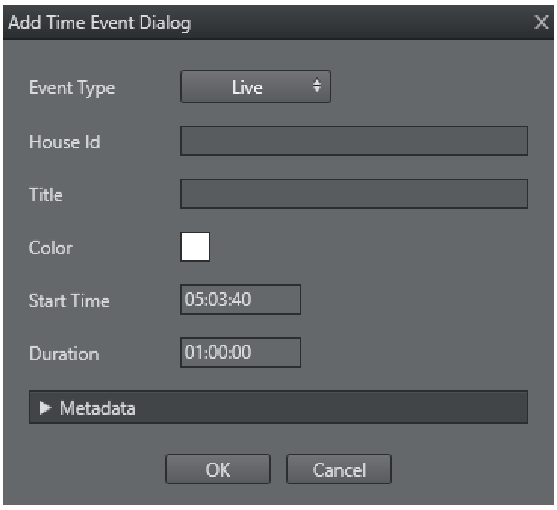


Right-click to add items

Add items to the schedule by drag-and-drop from media-list or clip-pool

## 11.3.1 To Add a New Item to the Schedule

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



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

### 11.3.2 To Modify Metadata for an Item in the Schedule

Right-click an existing item in the schedule and select **Edit item** to view and possibly 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**.

> ⚠ Options presented and save option depends on you having the required credentials for saving to Viz One

### 11.3.3 To Export a Schedule

Use the **Export** button in the main menu to export a schedule to a XML file.
**An example saved Schedule**

```
<?xml version="1.0" encoding="utf-8"?> <schedule> <event>
<id>1092781a-6d23-4504-956d-66d472991c90</id> <title>Opusads Adobe Adobe Snakebite
Mar2016_Summit_Grade_v2.mxf</title> <startdate>2016-09-06</startdate> <starttime>05:
00:00</starttime> <houseId>ZBCRUPDJOOFINWFC</houseId> <duration>00:01:00</duration>
</event> </schedule>
```

# 12 Page Content Filling

Page content can come from multiple sources. You are advised to use Promo Server for content filling. In addition, Viz Multichannel supports the following ways of adding content to pages:

- Manual Input
- Referenced Pages
- Page Content from Playlist Metadata
- Page Content from Database Lookups
- Page Content from External Update Services
- Page Content filling using Promo Server

## 12.1 Manual Input

In this mode, pages are filled with content without any connection to a schedule, playlist, database or web service (using update service). Usually, you open a page and fill it with data. You can also reference pages that are already filled with data.

For details of UI fields, see the section Dialog Menu and Explanations of Options. If you add pages as referenced pages, it allows you to edit one page that will change all other referenced pages.

### 12.1.1 To Edit Page Fields

1. Preview the page.
2. Click a field or press the Tab button to move between fields.
3. Edit the field as necessary and move on to the next field.
4. When you are finished editing, click **OK**.

**See Also**

- Working with Rules
- Dialog Menu and Explanations of Options

## 12.2 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 associated by reference to primary events in the playlist. When the content of the page has changed, either in the page 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.

### 12.2.1 To Create a Referenced Page

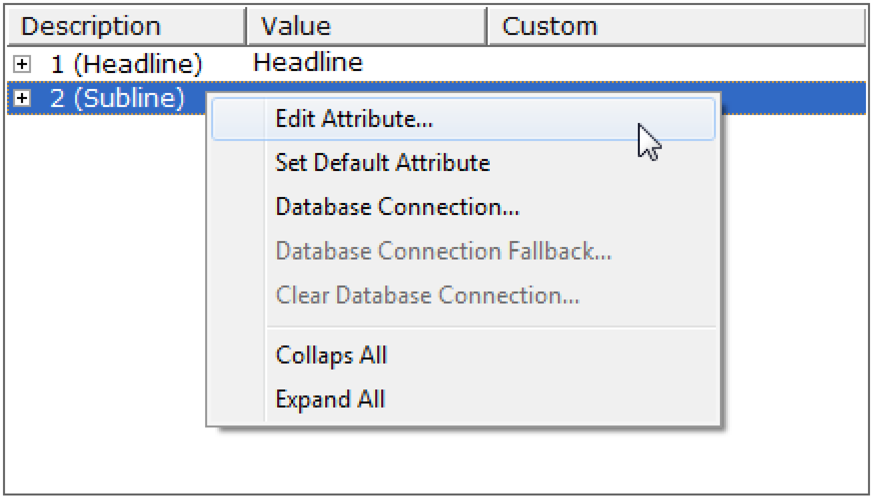
1. Preview the page
2. Edit the page
3. Press and hold **Alt + drag and drop** the page to the playlist
   - The page will receive the following icon:

## 12.3 Page Content From Playlist Metadata

Pages can automatically retrieve information from primary event meta data 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 effect all referenced occurrences of that page in a playlist. Changes to a page in the Playlists view will only have an effect on page.

### 12.3.1 To Populate a Page based on Playlist Fields

1. Select your channel.
2. Open the Page Pool or the Playlists view.
3. Double-click a page.
   - This will open the Page Editor.
   - Select a page from the Templates and Pages view to effect all subsequent occurrences of a page in the playlist.
   - Select a page from the Playlists view to effect only this occurrence.



| Description | Value | Custom |
|---|---|---|
| ⊞ 1 (Headline) | Headline | |
| ⊞ 2 (Subline) | | |

4. Right-click a field and from the appearing context menu select **Edit Attribute...**
   - This will open the Page Content Filling Dialog Menu.
5. Make your configurations.
6. Click **OK**.

## 12.4 Page Content From Database Lookups

Pages can automatically retrieve information from a range of Database Connections, using the Promo Server to carry out the queries.

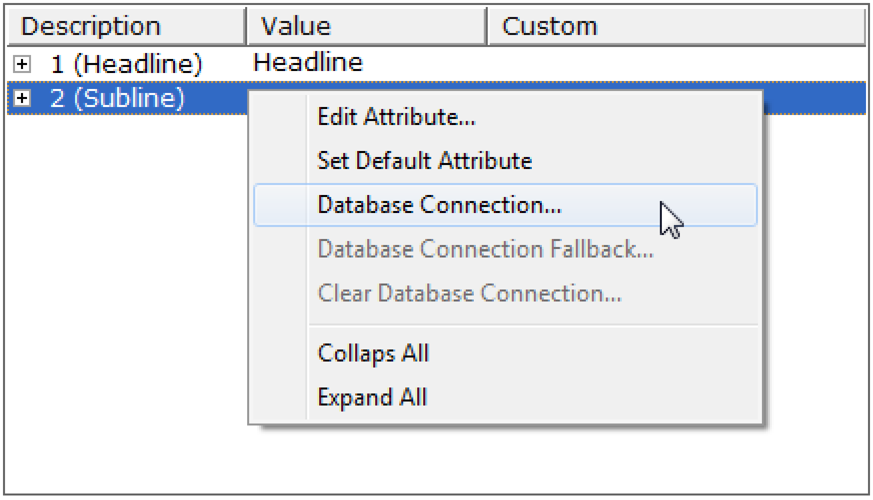
When working with databases it is important to always make sure the database columns **match the data type** of a page's field. If a database column is of type *Integer*, and your field is of type *String* you will not be able to make any updates.

Note that changes to a page in the Page Pool view will effect all referenced occurrences of that page in a playlist. Changes to a page in the Playlists view will only have an effect on page.

### 12.4.1 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 will open the Page Editor.



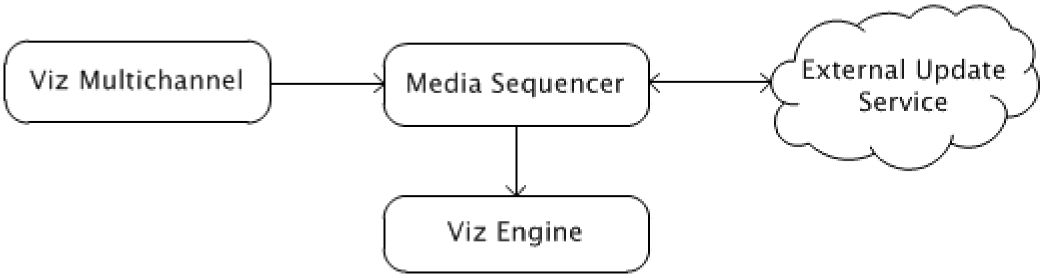
4. Right-click a field and from the appearing context menu select **Database Connection** or **Database Connection Fallback**
    - This will open the **Database Connection** window.
    - The *fallback* connection allows you to map up a second database, supporting automatic fallback if the first database lacks the Id supplied in the query.
5. Make your configurations.
6. Click **OK**.

### 12.4.2 To Clear a Database Connection

1. Select your channel.
2. Open the Page Pool or the Playlists view.
3. Double-click a page.

   - This will open the Page Editor.
4. Right-click a field and from the appearing context menu select **Clear Database Connection**
5. Click **Yes**.
   - The connection to the database is cleared for this page only.

## 12.5 Page Content From External Update Services



Page content filling can be done by pointing your template (without a script) to an external update service (e.g. 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 on the Media Sequencer that should be triggered during an update (read, take etc.) and set the timeout for each action.

⚠ **Note:** Using an update service currently does not allow updates at regular intervals.

For interested developers and 3rd party integrators Vizrt have developed a small stub-implementation for an external update service called ExternalUpdateServiceSample. This project is available as source code without cost upon request and can be used as a starting point for developing external update services.

**See Also**

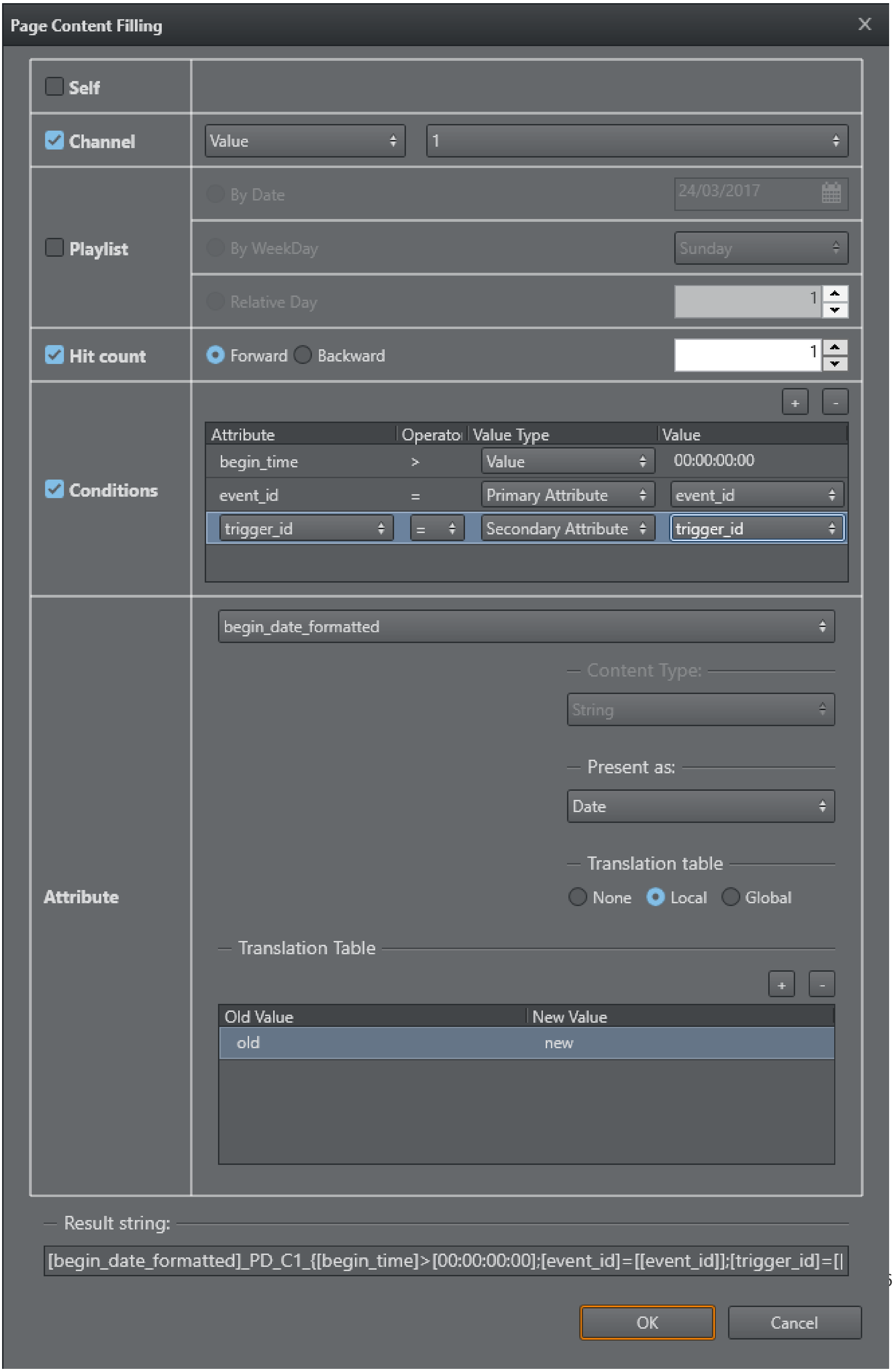
- Vizrt Data Format (VDF)

## 12.6 Page Content Filling Using Promo Server

Page content can be filled or updated by using the Promo Server. Promo Server is particularly effective and recommended in a cross-promotion usage scenario. Cross-promotion is when one channel needs to fill templates or pages with data fetched from other channels.

## 12.7 Dialog Menu And Explanations Of Options

This section provides guidelines and sample values, when working with the **Page Content Filling** dialog menu.



Page Content Filling

□ Self

☑ Channel: Value, 1

□ Playlist
- ○ By Date: 24/03/2017
- ○ By WeekDay: Sunday
- ○ Relative Day: 1

☑ Hit count: ● Forward ○ Backward, 1

☑ Conditions

| Attribute | Operato | Value Type | Value |
|---|---|---|---|
| begin_time | > | Value | 00:00:00:00 |
| event_id | = | Primary Attribute | event_id |
| trigger_id | = | Secondary Attribute | trigger_id |

Attribute: begin_date_formatted

Content Type: String

Present as: Date

Translation table: ○ None ● Local ○ Global

Translation Table

| Old Value | New Value |
|---|---|
| old | new |

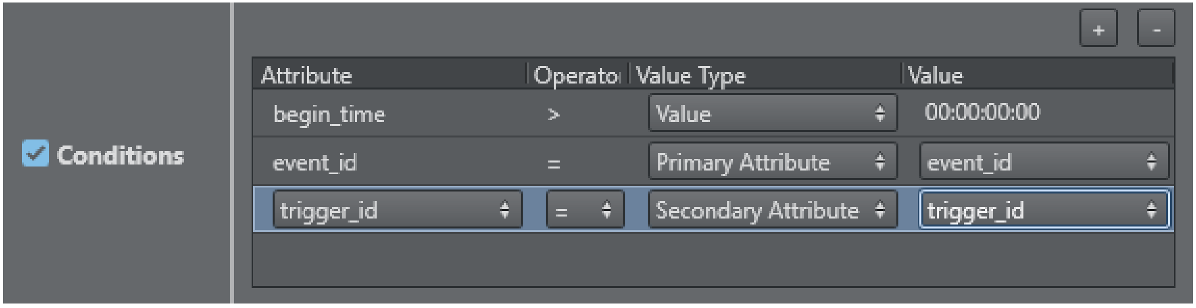
Result string:

[begin_date_formatted]_PD_C1_{[begin_time]>[00:00:00:00];[event_id]=[[event_id]];[trigger_id]=[[

- **Self**: Gets information from the primary event the page resides under. This will disable the Channel, Playlist, Item and Conditions options.
- **Channel**: Allows you to select the relevant channel if you want to populate your pages with primary event content from another channel. Can have one of 3 values.
  - **Value**: Select the relevant channel from the list. Channels are configured in Promo Server configuration (*see* Promo Server Installation and Configuration)
  - **Primary Attribute**: Channel name derived from the specific attribute value of primary event.
  - **Secondary Attribute:** Channel name derived from the specific attribute value of page itself.
- **Playlist**: Allows you to select another playlist than the current. If you are working on the current playlist do not select this option. For selecting other playlists than the current, your options are:
  - **By Date**: Reads from the playlist of the date selected.
  - **ByWeek Day**: Reads from the next day selected.
  - **By Relative Day**: Reads from the playlist as many days in the future as selected. Select **1** for the following day, **2** for two days after the current date's playlist, etc.
- **Hit count**: Allows you to select an item from another playlist than the current. If you are working on the current playlist do not select this option.

> ⚠ **Note:** The old behavior of selecting an item by time or relative to a current event has been replaced by the **Hit count** option. Unlike the **Item** option, **Hit count** does not allow for negative (i.e. relative).
> Also, 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_` indicates it is to check data from the page (itself) and not the parent (primary element). This allows you to use secondary data (e.g. `secondary_begin_time`) to find relevant primary event data, giving more dynamic conditions.



☑ Conditions

| Attribute | Operato | Value Type | Value |
|---|---|---|---|
| begin_time | > | Value | 00:00:00:00 |
| event_id | = | Primary Attribute | event_id |
| trigger_id | = | Secondary Attribute | trigger_id |

  - **Value Type** which has 3 options:
    **Value**: Compares attribute value with the condition value.
    **Primary Attribute**: Compare attribute value with the specific attribute value of primary event.
    **Secondary Attribute**: Compare attribute value with the specific attribute value of page itself.
- **Attribute**: Allows you to select an attribute from a located primary event or the current secondary event. The fields available in the list depends on what is defined in the Mapping Table under General Settings. Translation tables can be used to define arbitrary value-translations, via a value=value-table. There are two translations tables:

    - *TranslationTable*: settings per page.
    - *GlobalTranslationTable*: As the name implies, a Global Translation Table. Can be configured in Channel Settings under Import/Export Settings > General Settings > Translation Tables.
- **Result string**: Displays the resulting string based on your configuration of the respective fields. It can be edited.

**See Also**: Page Content Filling Aliases

# 13 Promo Server

Promo Server is a service process that enables efficient handling of page-filling and playlist data transfers, especially for cross-promotional purposes. Promo Server runs as a Windows service process. After installation and configuration you do not interact with the Promo Server directly as it neither requires nor have a user interface. You will use the Multichannel Client Configuration settings to configure server address of the Promo Server and direct Multichannel to use Promo Server.

Cross-promotion is typically used in scenarios where one or more channels in Viz Multichannel need to access information about events in other channels in order to fetch data that are used for page-filling. A typical example could be that the channel MUSIC needs to know what events are scheduled on channel MOVIES in order to answer such questions 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 questions.

In order to use Promo Server you will need to:

1. Install Promo Server on a server. Preferably on a separate server for security and efficiency, but it could also be on a shared server.
2. Configure Promo Server by editing the *PromoServer.config* configuration file. This step basically tells Promo Server: what are the *channel names* and what are the *network address* of the Media Sequencer servers handling these channels.
3. Tell Viz Multichannel to use the Promo Server you are using by configuring and selecting it in Viz Multichannel’s configuration settings (ref. **Settings > General Settings > Page Content Filling**).
   The details on how to do this are explained in the following sections:

- Promo Server Installation and Configuration
- Promo Server Logs and Messages
- Configuring Multichannel for Promo Server
- Promo Server Web Interface

## 13.1 Promo Server Installation And Configuration

### 13.1.1 Installing the Promo Server

1. Locate the current Promo Server installer from your Vizrt representative or from the ftp server download.vizrt.com. The Promo Server is a separate installer image named `VizMultichannel_PromoServer-VERSION.exe` where *VERSION* will be numbers indicating the release/version of the software. Double-click on the installer to start the installation.
2. Press **Next** and on the next page **Next** to install Promo Server to the default installation directory.
3. Press the **Install** button and wait until the software is installed. You can now exit the installer by pressing **Finish**.

> **IMPORTANT!** The Promo Server run as a Windows Service process. It is recommended that the Promo Server service process is enabled to start automatically and that the Promo Server is run as an Administrator servicelevel process. See the next chapter for details.

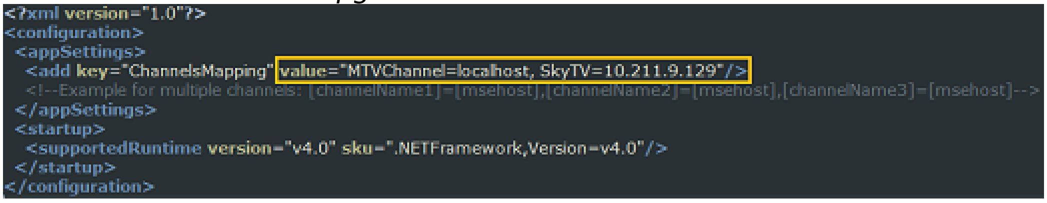
### 13.1.2 Configuring the Promo Server

Promo Server must be told which target channels to serve and on which server (i.e. network address) these channel(s) reside. This is done by editing the XML textfile *PromoServer.config*

> **IMPORTANT!** The *PromoServer.config* is an XML-formated textfile so make sure to save the file in text format (not Word, RTF or any other special format) after editing. You can use for instance Notepad to do the editing.

To configure Promo Server follow these steps:

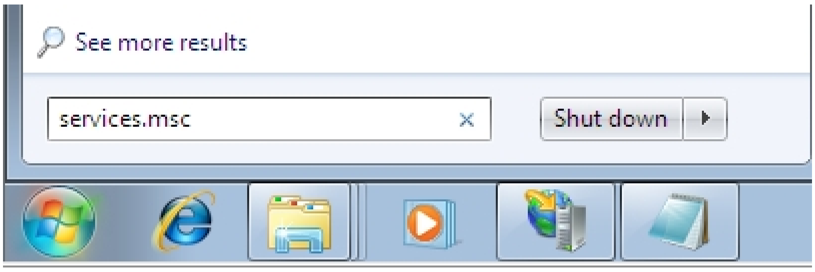
1. Use Windows Explorer to locate the directory *%ProgramData%\Vizrt\Multichannel\PromoServer* and locate the config-file. Normally: *c:\ProgramData\Vizrt\Multichannel\PromoServer.config*
2. Open PromoServer.config in a text editor
3. For key **ChannelsMapping** change the attribute name **value** to be the target channelname and either IP-address or hostname of the server. If you have several channels, each channel is separated by a “,” (comma) character. Example format: *value=”MUSIC=192.168.0.10, MOVIE=192.168.0.11*" , see the image below for an example.
4. Save the *PromoServer.config* file as a normal textfile.



```
<?xml version="1.0"?>
<configuration>
 <appSettings>
  <add key="ChannelsMapping" value="MTVChannel=localhost, SkyTV=10.211.9.129"/>
  <!--Example for multiple channels: [channelName1]=[msehost],[channelName2]=[msehost],[channelName3]=[msehost]-->
 </appSettings>
 <startup>
  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.0"/>
 </startup>
</configuration>
```

After you have configured (or changed) the *PromoServer.config* file you need to restart the Promo Server service process. To restart the Promo Server and to configure the service-level follow these steps:

Start Windows Services administrator utility. On Windows 7 like systems: Press Windows **Start** menu button, type **services.msc** in the search bar and press the **Enter** button to start the program.



1. In the Services window, scroll down in the list to locate the **Vizrt Viz Multichannel Promo Server** service.

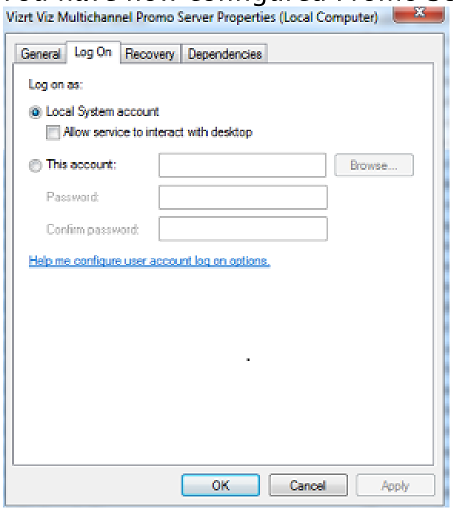
2. Right-click on the **Vizrt Viz Multichannel Promo Server** item and in the context menu select **Restart**.
3. Wait until the Service Window confirmation window disappear.
4. Check that Promo Server is indeed running and that the option to have it start automatically at server startup is selected.

### 13.1.3 Checking Promo Server Service has Administrator Privileges

In order to ensure that the Promo Server runs with administrator privileges follow the following steps.

> **Note:** This procedure normally only needs to be performed once

1. Start Windows services.msc utility as described in the previous procedure.
2. Locate the **Vizrt Viz Multichannel Promo Server** item, right-click and in the context menu select **Properties**.
3. In the properties window set the Administrator username and password or a user account with Administrator privileges and press **OK.**
4. Check Promo Server logs
   You have now configured Promo Server.



Vizrt Viz Multichannel Promo Server Properties (Local Computer)

Log on as:

● Local System account

□ Allow service to interact with desktop

○ This account:

Password:

Confirm password:

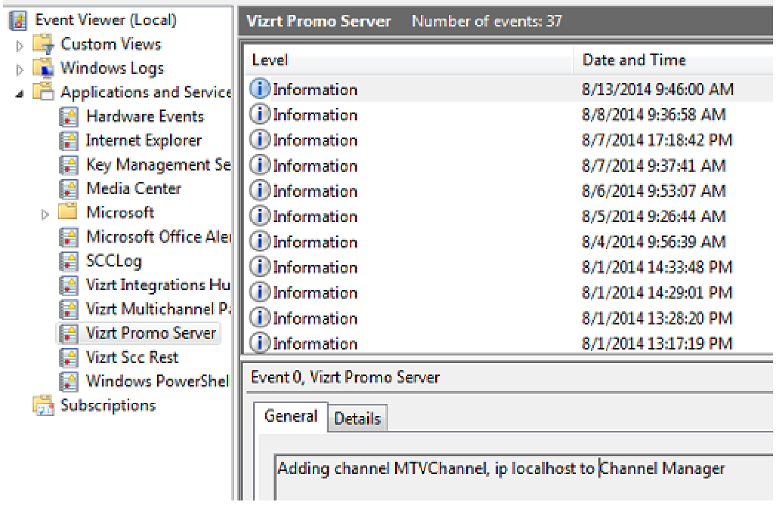
Help me configure user account log on options.

## 13.2 Promo Server Logs And Messages

In order to check Promo Server logs perform the following steps. Note: procedure is for Windows 7, minor differences can be expected depending on your Windows version, but 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 you wish to view the log for, in this case Vizrt Promo Server as illustrated in the image below:

**Note:** Status message from Promo Server will be stored in 'infoPromoServer' attribute of an event.



Vizrt Promo Server Number of events: 37

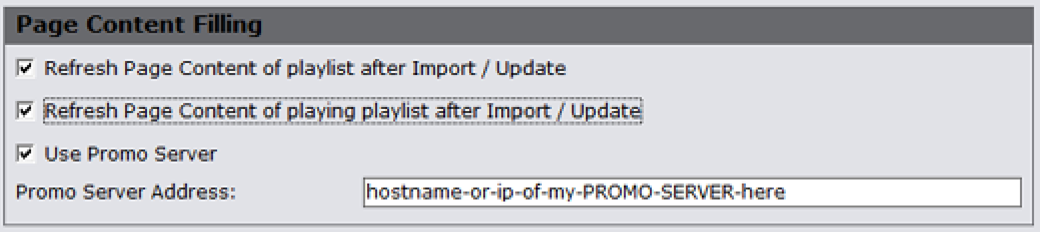
| Level | Date and Time |
|---|---|
| Information | 8/13/2014 9:46:00 AM |
| Information | 8/8/2014 9:36:58 AM |
| Information | 8/7/2014 17:18:42 PM |
| Information | 8/7/2014 9:37:41 AM |
| Information | 8/6/2014 9:53:07 AM |
| Information | 8/5/2014 9:26:44 AM |
| Information | 8/4/2014 9:56:39 AM |
| Information | 8/1/2014 14:33:48 PM |
| Information | 8/1/2014 14:29:01 PM |
| Information | 8/1/2014 13:28:20 PM |
| Information | 8/1/2014 13:17:19 PM |

Event 0, Vizrt Promo Server

Adding channel MTVChannel, ip localhost to Channel Manager

## 13.3 Configuring Multichannel For Promo Server

1. Start Multichannel and select the Open **Settings** icon in the toolbar.
2. Under the Import/Export heading select **General Settings**
3. In the **Page Content Filling** header check **Use Promo Server** and in the parameter field specify either the hostname or the IP-address of the Promo Server.



Page Content Filling

☑ Refresh Page Content of playlist after Import / Update

☑ Refresh Page Content of playing playlist after Import / Update

☑ Use Promo Server

Promo Server Address: hostname-or-ip-of-my-PROMO-SERVER-here

4. Press **OK** to save the changed settings.

**Note:** Only the Use Promo Server and the field to specify the Promo Server hostname address in the image above is specific for Promo Server. The *Refresh...* settings is definitions for when Page Content Filling should be run, for a normal schedule import, be it using the standalone Promo Server service or PlayListImporter.

## 13.4 Promo Server Web Interface

The Multichannel Promo Server has a web-interface. This section details the required format of a typical API call for interfacing towards the Promo server services.

### 13.4.1 Required Namespace

Namespace of payload is: "http://www.vizrt.com/types".

### 13.4.2 Required Data Format

The payload must contains a "header" node which has information of channel, playlist, group and page names. Example payload:

```
<header>
    <channel>1</channel>
    <playlist>continuous</playlist>
    <group>523d0bb9-2b6c-4278-b5ff-1f594b3d6e17</group>
    <page>4923de46-ff7c-452d-b7e5-79bec7d2bfe2</page>
</header>
```

If the request payload is invalid (not correct namespace, no content, no header information) the response will be HTTP/1.1 400 Bad Request

### 13.4.3 Example API Call (POST)

*POST /PayloadParserService/ResolvePayload HTTP/1.1*

```
<payload xmlns="http://www.vizrt.com/types">
  <field name="pop_testing">
    <value>Good words need great graphics</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <field name="2">
    <value>A bad graphic is worse than no graphic</value>
    <field name="kerning">
      <value>0</value>
    </field>
    <lookuprules xmlns="http://www.vizrt.com/2014/
multichannel">[description]_0_IsString</lookuprules>
  </field>
  <field name="3">
    <value>Size DOES matters! Superb HD graphics from Vizrt</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <field name="4">
    <value>Your analogue station just entered a new era - history</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <field name="5">
    <value>No Viz. No Fizz</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <field name="6">
    <value>It...s TV. You must VIZualize it</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <field name="7">
    <value>Mood music for the eyes. Vizrt</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <field name="8">
    <value>No story is complete without a Viz graphic</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <header>
    <channel>1</channel>
    <playlist>continuous</playlist>
    <group>523d0bb9-2b6c-4278-b5ff-1f594b3d6e17</group>
    <page>4923de46-ff7c-452d-b7e5-79bec7d2bfe2</page>
  </header>
  <entry name="dblink" xmlns="http://www.vizrt.com/2014/multichannel" />
</payload>
```

### 13.4.4 Expected Response

```
HTTP/1.1 200 OK
Content-Type: application/octet-stream
<payload xmlns="http://www.vizrt.com/types">
  <field name="pop_testing">
    <value>Good words need great graphics</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <field name="2">
    <value>Primary Event</value>
    <field name="kerning">
      <value>0</value>
    </field>
    <lookuprules xmlns="http://www.vizrt.com/2014/
multichannel">[description]_0_IsString</lookuprules>
  </field>
  <field name="3">
    <value>Size DOES matters! Superb HD graphics from Vizrt</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <field name="4">
    <value>Your analogue station just entered a new era - history</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <field name="5">
    <value>No Viz. No Fizz</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <field name="6">
    <value>It...s TV. You must VIZualize it</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <field name="7">
    <value>Mood music for the eyes. Vizrt</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <field name="8">
    <value>No story is complete without a Viz graphic</value>
    <field name="kerning">
      <value>0</value>
    </field>
  </field>
  <header>
    <channel>1</channel>
    <playlist>continuous</playlist>
    <group>523d0bb9-2b6c-4278-b5ff-1f594b3d6e17</group>
    <page>4923de46-ff7c-452d-b7e5-79bec7d2bfe2</page>
  </header>
  <entry name="dblink" xmlns="http://www.vizrt.com/2014/multichannel" />
  <promoserverinfo xmlns="http://www.vizrt.com/2014/multichannel">
    <status>ok</status>
  </promoserverinfo>
</payload>
```

# 14 XML Translator

The XML Translator is a command-line tool for transforming a schedule-file in XML (BXF) format to a format Multichannel can import. Content is not changed, the tool just transforms data format 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

## 14.1 Installing The XML Translator

You can install XML translator with the bundle installer, for details see Bundle Installer in the Setup and Configuration chapter.

Alternatively, you can install the tool separately from the MSI installer:

1. Download the installer from ftp.vizrt.com or extract the MSI from the bundle installer.
2. Click `VizMultichannel_XMLTranslator_VERSION.msi` and follow the on-screen instructions. Normally, the XML translator will be installed to:

```
%PROGRAMFILES%\Vizrt\Viz Multichannel\XMLTranslator\XMLTranslator.exe
```

## 14.2 Using The XML Translator

Open a command window (**Windows-button+R CMD <enter>**) to use the XML translator. If you need admin rights for any part when using the tool start CMD as Administrator then to start an elevated session, **right-click CMD** and select **Run as Administrator**. In the following we assume you have XML translator program in the Windows PATH, otherwise you will have to provide the full path 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:

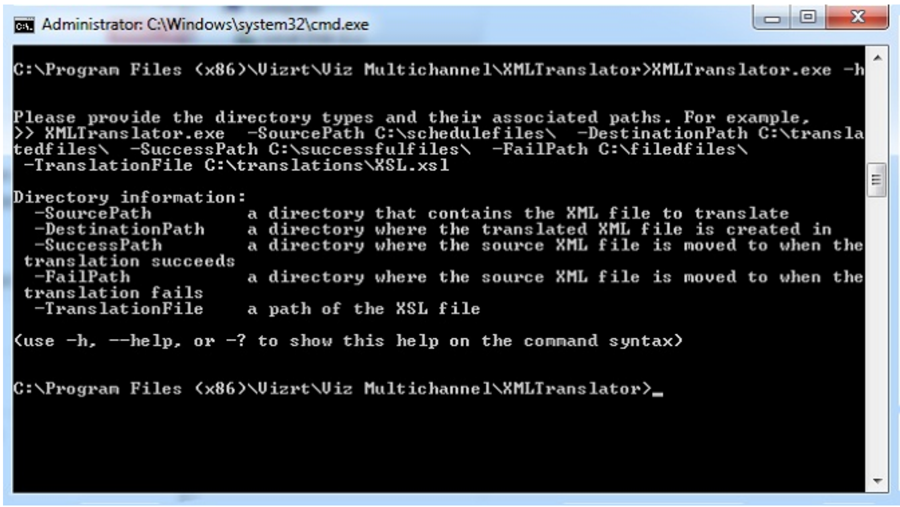
- **SourcePath** the directory that contains the XML file to translate
- **DestinationPath** a directory where the translated XML file is created in

- **SuccessPath** a directory where the source XML file is moved to when translation succeeds
- **FailPath** a directory where the source XML file is moved to if the translation fails
- **TranslationFile** path to the XSL file defining the transformations

A typical run (the command should be entered on a single line) could be:

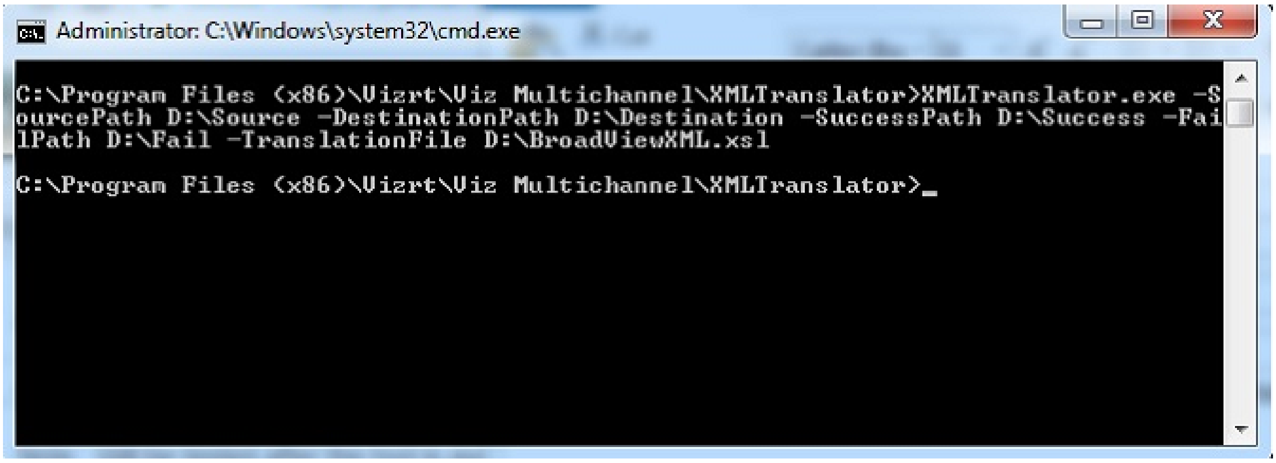
```
XMLTranslator.exe -SourcePath D:\Source -DestinationPath D:\Destination -SuccessPath
D:\Success -FailPath D:\Fail -TranslationFile D:\BroadViewXML.xsl
```

**Example invocation: help**



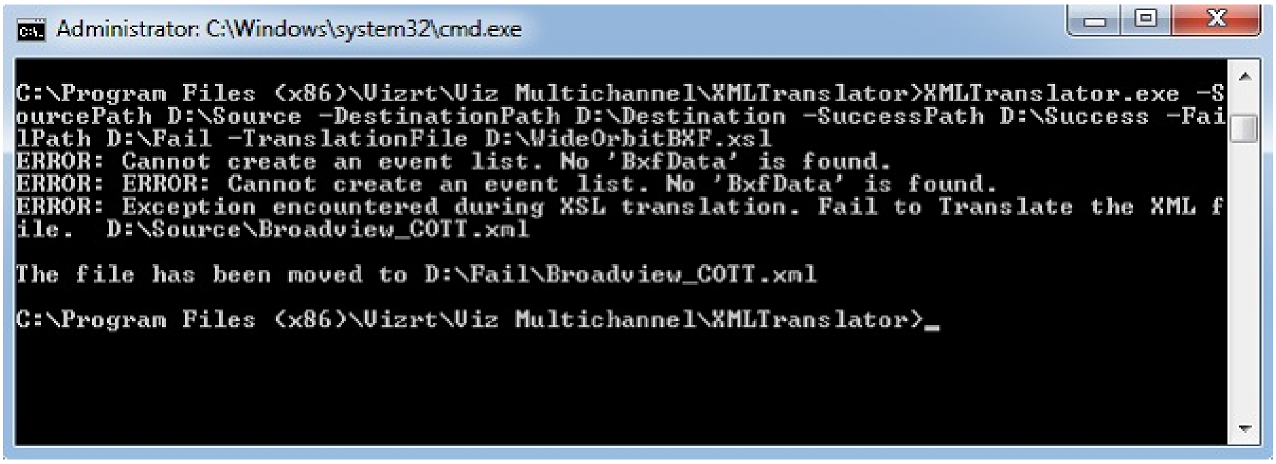
```
C:\Program Files (x86)\Vizrt\Viz Multichannel\XMLTranslator>XMLTranslator.exe -h

Please provide the directory types and their associated paths. For example,
>> XMLTranslator.exe  -SourcePath C:\schedulefiles\  -DestinationPath C:\transla
tedfiles\  -SuccessPath C:\successfulfiles\  -FailPath C:\filedfiles\
 -TranslationFile C:\translations\XSL.xsl

Directory information:
  -SourcePath          a directory that contains the XML file to translate
  -DestinationPath     a directory where the translated XML file is created in
  -SuccessPath         a directory where the source XML file is moved to when the
 translation succeeds
  -FailPath            a directory where the source XML file is moved to when the
 translation fails
  -TranslationFile     a path of the XSL file

(use -h, --help, or -? to show this help on the command syntax)

C:\Program Files (x86)\Vizrt\Viz Multichannel\XMLTranslator>_
```

**Example invocation: success**



```
C:\Program Files (x86)\Vizrt\Viz Multichannel\XMLTranslator>XMLTranslator.exe -S
ourcePath D:\Source -DestinationPath D:\Destination -SuccessPath D:\Success -Fai
lPath D:\Fail -TranslationFile D:\BroadViewXML.xsl

C:\Program Files (x86)\Vizrt\Viz Multichannel\XMLTranslator>_
```

**Example invocation: fail**



```
C:\Program Files (x86)\Vizrt\Viz Multichannel\XMLTranslator>XMLTranslator.exe -S
ourcePath D:\Source -DestinationPath D:\Destination -SuccessPath D:\Success -Fai
lPath D:\Fail -TranslationFile D:\WideOrbitBXF.xsl
ERROR: Cannot create an event list. No 'BxfData' is found.
ERROR: ERROR: Cannot create an event list. No 'BxfData' is found.
ERROR: Exception encountered during XSL translation. Fail to Translate the XML f
ile.  D:\Source\Broadview_COTT.xml

The file has been moved to D:\Fail\Broadview_COTT.xml

C:\Program Files (x86)\Vizrt\Viz Multichannel\XMLTranslator>_
```

## 14.3 External References

- Wikipedia: Extensible Markup Language: XML
- Wikipedia: Extensible Stylesheet Language Transformations: XSLT
- Wikipedia: Broadcast Exchange Format: BXF

# 15 Traffic Department Workflow

This is a collection of common Traffic Department procedures performed before handing playlist control to the automation system in master control.

This chapter contains information on the following topics:

- Working with Templates and Pages
- Working with Actions
- Working with Ticker Actions
- Working with Playlists
- Working with Video Clips
- Working with Rules
- Resolving Conflicts
- Sending Playlists to the Master Control
- Generating the Channels File

**See Also**

- Master Control Workflow

## 15.1 Working With Templates And Pages

The first step in populating a playlist with graphics is importing Viz Artist scenes in to Viz Multichannel. These scenes become templates, saved in Multichannel 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.

This section contains information on the following topics:

- Importing Scenes
- Creating Pages
- Previewing Pages
- Editing Page Duration
- Sending Templates and Pages

**See Also**

- Page Content Filling

### 15.1.1 Importing Scenes

Templates are scenes imported from Viz Artist, and while the scene cannot be edited one can adjust the information in the tab fields as well as create pages (i.e. an instance of the template) that can be edited (see Creating Pages) from this Template.

### To Import a template

1. Click the Import Scene 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).

## 15.1.2 Creating Pages

A page is an instance of a template that is made editable. Meaning you can, if the template contains editable elements, add text, images, and videos to it before you add 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 and Pages button (left pane).
   All templates are displayed in the left pane.
3. Right-click the template and from the appearing 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

## 15.1.3 Previewing Pages

You can preview all pages, regardless of whether or not they are editable.

### To Preview a Page

1. Select your channel
2. Click the Templates and Pages button (left pane).
   All templates and pages are shown.
3. Double-click the page you wish you preview.
   The page is shown in the preview window.

### 15.1.4 Editing Page Duration

You may edit the duration field for a page, setting a default duration, before you add it to your playlist.

#### To Edit Page Duration

1. Select your channel.
2. Click the Templates and Pages 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**

- Resolving Conflicts

### 15.1.5 Sending Templates and Pages

Templates and pages can be manually or automatically sent to the MCR setup or other machine for backup purpose. Both options require that the remote machines Media Sequencer host is defined in the **Remote Hosts** section under General Settings.

#### To Manually Send Templates and Pages to a Remote Host

1. Select your channel.
2. Click the Templates and Pages 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.
   Whenever 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

## 15.2 Working With Actions

Actions, such as *Viz Commands* and *GPI Functions*, can automate certain tasks. For example, Viz commands can be used in Viz Multichannel 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.

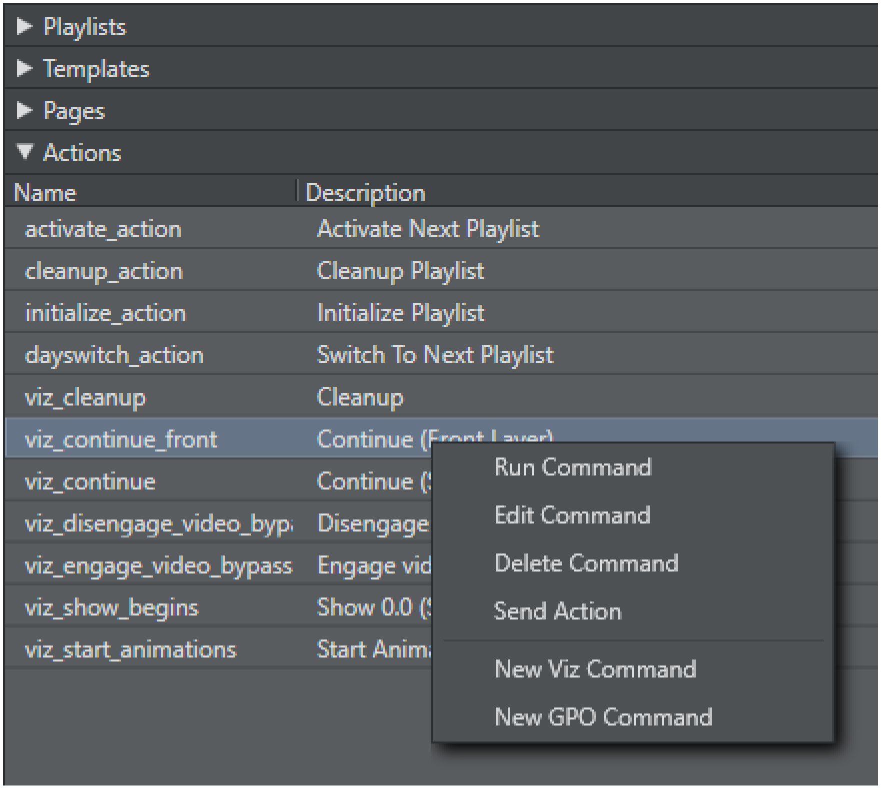
All commands must be first defined in Viz Multichannel in order for them to then be available as secondary events

- Defining Actions
- Referenced Actions

## 15.2.1 Defining Actions

### To Define Secondary Event Actions

1. Click the Actions button
   - The list of predefined commands are displayed



| Name | Description |
|---|---|
| activate_action | Activate Next Playlist |
| cleanup_action | Cleanup Playlist |
| initialize_action | Initialize Playlist |
| dayswitch_action | Switch To Next Playlist |
| viz_cleanup | Cleanup |
| viz_continue_front | Continue (Front Layer) |
| viz_continue | Continue ( |
| viz_disengage_video_byp | Disengage |
| viz_engage_video_bypass | Engage vid |
| viz_show_begins | Show 0.0 ( |
| viz_start_animations | Start Anim |

2. Right-click one of the actions and from the appearing context menu select **New Viz Command** or **New GPO Command**
3. In the appearing 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 (e.g. `viz_newcommand`) - that is verified unique upon generation - and may be edited 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 that is included with your Viz Engine/Viz Artist installation.

**See Also**

- Working with Ticker Actions

### 15.2.2 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 this use case, an action is associated by reference to primary events in the playlist. When the action has 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:

## 15.3 Working With Ticker Actions

Ticker actions are used to automate the playout and control of tickers.

In order to automate playout of tickers you first need to export available ticker actions from your Viz Multichannel client in the MCR. When ticker actions are exported you can import them to your Viz Multichannel client in the scheduling department. This will allow you to schedule ticker actions as secondary events and test playout of tickers from scheduling.

> **Note:** As of Viz Multichannel 2.8 and Media Sequencer 2.0.1 and higher, a **Take** will put the ticker in state ON in Viz Ticker3D, and a **Take Out** will set it in state OFF. This also allows you to set Duration for a Ticker Action command.

This section contains information on 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

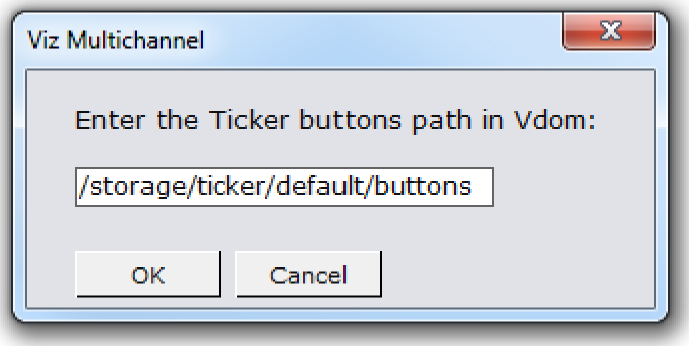
## 15.3.1 Export Ticker Actions

This procedure takes the ticker action’s structure in the Media Sequencer and virtually maps its XML structure (i.e. without the logic) for use by Viz Multichannel in scheduling mode.

The export is performed on the Media Sequencer where the real ticker resides (i.e. Viz Multichannel in MCR mode).

### To Export Ticker Actions

1. Click the Actions menu on the Main Menu.
2. Select **Ticker** and then **Export...**



Enter the Ticker buttons path in Vdom:

/storage/ticker/default/buttons

3. Use the default path (or select another) and click **OK**.
   You may enter the Viz Ticker buttons path in the Media Sequencer’s VDOM structure.
4. Enter a file name and click **Save**.

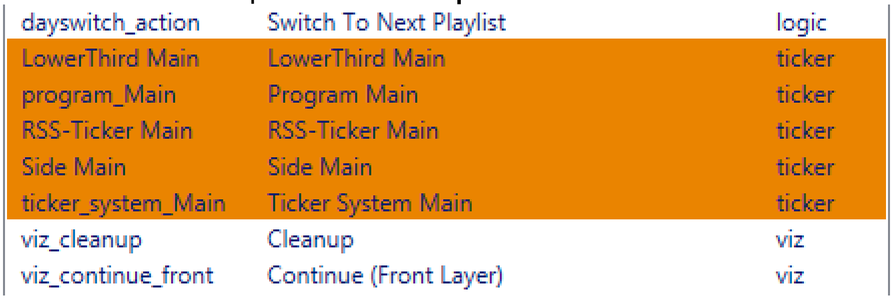
## 15.3.2 Import Ticker Actions

This procedure imports only a virtual version of the ticker actions (i.e. without its logic). Before you can import ticker actions you must Export Ticker Actions.

### To Import Ticker Actions

1. Click the Actions menu on the Main Menu.
2. Select **Ticker** and then **Import...** .

3. Select the file to import and click **Open**.



| | | |
|---|---|---|
| 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 |

The ticker actions are displayed in the left pane.

> ⚠ **IMPORTANT!** The Ticker actions that are read from the imported Ticker XML-file needs to have **ON** or **OFF** (capital or lowercase will both work) in the ticker button-name for the action to be imported correctly. For example will ticker-buttons named *All ON*, *All of* and *Command ON* all import correctly, while buttons named for example *Engine IN* or *Engine OUT* will **not** import correctly.

## 15.3.3 Scheduling Ticker Actions Manually

Ticker actions, as for pages and other actions, can be manually added to the playlist as secondary events.

### To Manually Schedule Tickers

- Follow the same procedure To Manually Add a Page to the Playlist.

## 15.3.4 Scheduling Tickers Actions Automatically

Ticker actions, as for pages and other actions, can be automatically added to the playlist as secondary events the same way as for pages. In addition you can add them through import rules.

Note that all actions (as for pages) can be mapped to via a schedule by referring to the action’s name while mapped as a secondary event (see the Table Columns’ *secondary_page* setting).

### To Automatically Schedule Tickers

- Use the name of the ticker actions in your schedule.

### To Define a Rule on a Ticker Command

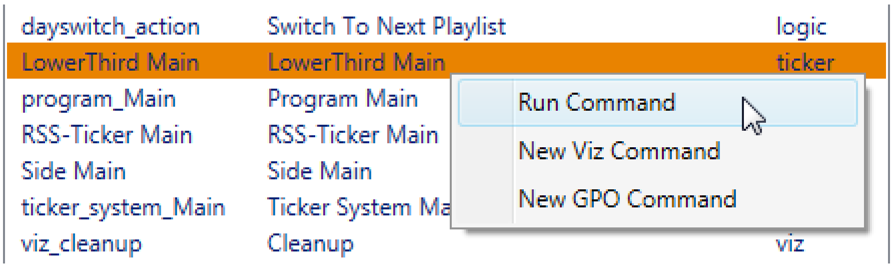
1. Select your channel.
2. Click the Rules button.
3. Continue with the procedures in the Working with Rules section.

## 15.3.5 Playing Tickers Manually

This procedure is performed on Viz Multichannel in MCR where the real tickers reside.

### To Manually Play Tickers

1. Select your channel.
2. Click the Actions button.



| | | |
|---|---|---|
| dayswitch_action | Switch To Next Playlist | logic |
| LowerThird Main | LowerThird Main | ticker |
| program_Main | Program Main | |
| RSS-Ticker Main | RSS-Ticker Main | |
| Side Main | Side Main | |
| ticker_system_Main | Ticker System Ma | |
| viz_cleanup | Cleanup | viz |

3. Right-click the command which you want to play and select **Run Command**.

## 15.4 Working With Playlists

Viz Multichannel’s main task as an application 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. When creating a continuous playlist you cannot add playlists per broadcast day, and conversely.

Each new broadcast day playlist will have a different date, regardless of the day you created it on. For example, if you create two playlists today, the first one will be dated today by default, and the following one will be dated tomorrow.

This section contains information on the following topics:

- Importing 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
- Exporting a Playlist
- Playlist State Definitions
- Playlist Errors

**See Also**

- Master Control Workflow on how to activate, initialize and manually take events on air
- Working with Actions as secondary event actions in a playlist

### 15.4.1 Import a Schedule

Viz Multichannel can import a schedule from an external source. However, there are certain parameters to configure the first time you import it.

Schedules are generally composed by the traffic departments. Many scheduling systems can create schedules with both primary events and secondary events, i.e. schedule plus associated pages. Viz Multichannel also supports user-defined rules for automatic insertion of secondary events.

Schedule file formats that can be imported as playlists are:

- Importing from Text (*.csv, *.txt, *.vicc)
- Importing from XML (*.xml)
- Importing from Excel (*.xls, *.xlsx)
- Harris ADC-100 (*.lst, *.rcv)
- Omnibus (*.osc)

If this is the **first time you are importing a schedule**, you must first complete the procedures in the following sections:

- General Settings
- Working with Rules

### To Import a Schedule

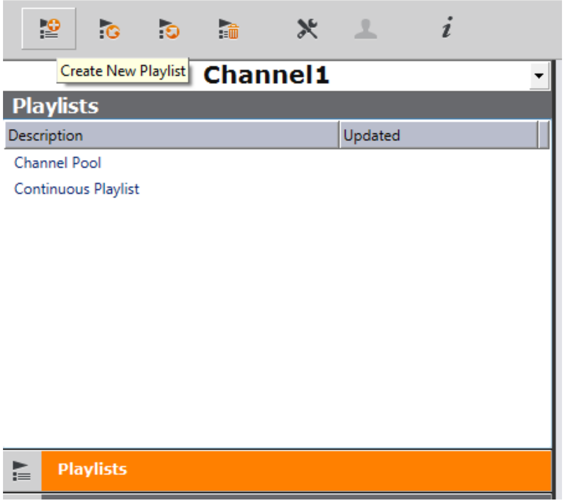
1. Select your channel
2. From the toolbar, click the **Import schedule** button
3. Select the file to import and click **Open**.

## 15.4.2 Creating a Playlist

Viz Multichannel can create a playlist based on an imported schedule, or create one manually. Note that the common and preferred way is to import schedules from scheduling systems.

### To Manually Create a Playlist

1. Select your channel.
2. Click the Playlists button.
   All existing playlists for the selected channel are displayed.



3. Click the **Create New Playlist** button
   A new row representing the new playlist is added.

## 15.4.3 Deleting a Playlist

### To Manually Delete a Playlist

1. Select your channel.
2. Click the Playlists button.
   All existing playlists for the selected channel are displayed.
3. Select the playlist and do one of the following:
   - Right-click the playlist item and select **Delete Playlist** from the context menu, or
   - Click the **Delete Playlist** button on the toolbar
4. Click **Yes** to confirm.

## 15.4.4 Adding Primary Events to a Playlist

### To Manually Add Primary Events to 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. Right-click the relevant playlist and from the appearing context menu, select **Add Primary Event**.

## 15.4.5 Adding Secondary Events to a Playlist

Secondary events can be pages and video clips, or a combination thereof.

### To Manually Add a Page to the Playlist

1. Select your channel.
2. Click the Playlists button.
   All existing playlists for the selected channel are displayed.
3. Open your playlist.
4. Click the **Templates and** Pages or **Video Clips** button.
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 by reference only to the original page.

   - The secondary event appears below the primary event. If there are several secondary events, they will be sorted according to their offsets.
   - 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.

**See Also**

- Working with Templates and Pages
- Working with Video Clips

## 15.4.6 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** from the appearing context menu.

## 15.4.7 Updating a Playlist

If there are changes to be made to a playlist, you have the following options:

- Full Update
- Full Update Without Event Deletion
- Update Times.

⚠ 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 of primary events' begin time, duration and offset from changes that have been made to the schedule. Any primary event additions or deletions are ignored. Only the times are updated.

## 15.4.8 Exporting a Playlist

### To Export a Playlist

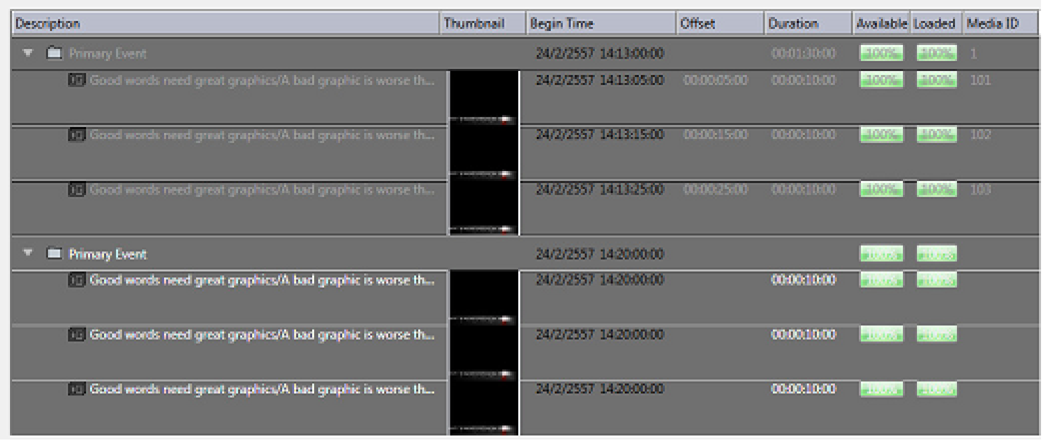
1. Select your channel.
2. Click the Playlists button.
   All existing playlists for the selected channel are displayed.
3. Open your playlist.
4. From the toolbar, click the **Export Playlist** button.
   If the original playlist was LST, then the exported playlist can be merged with the original file.

For all other formats, you are prompted to save the file and assign a name in the relevant format.

## 15.4.9 Playlist State Definitions

The playlist currently has 6 states for elements that reside in it. Each of these states use color on the playlist elements to give a quick visual indication of their state:

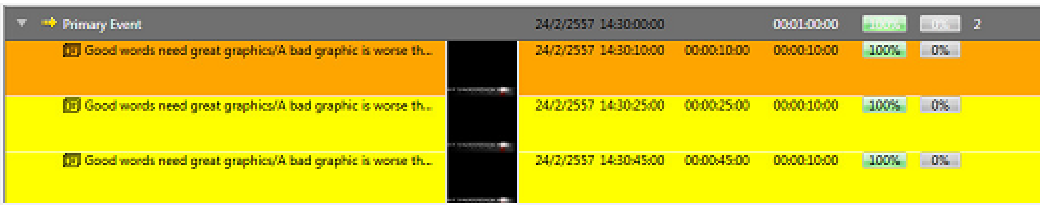
- **Normal state** – The state of an un-played element. Elements are shown in a *White* text.
- **Played state** – The state of an element where it has been played out at least once. Elements in are shown in a *Gray* text.



| Description | Thumbnail | Begin Time | Offset | Duration | Available | Loaded | Media ID |
|---|---|---|---|---|---|---|---|
| Primary Event | | 24/2/2557 14:13:00:00 | | 00:01:30:00 | 100% | 100% | 1 |
| Good words need great graphics/A bad graphic is worse th... | | 24/2/2557 14:13:05:00 | 00:00:05:00 | 00:00:10:00 | 100% | 100% | 101 |
| Good words need great graphics/A bad graphic is worse th... | | 24/2/2557 14:13:15:00 | 00:00:15:00 | 00:00:10:00 | 100% | 100% | 102 |
| Good words need great graphics/A bad graphic is worse th... | | 24/2/2557 14:13:25:00 | 00:00:25:00 | 00:00:10:00 | 100% | 100% | 103 |
| Primary Event | | 24/2/2557 14:20:00:00 | | | [illegible] | [illegible] | |
| Good words need great graphics/A bad graphic is worse th... | | 24/2/2557 14:20:00:00 | | 00:00:10:00 | [illegible] | [illegible] | |
| Good words need great graphics/A bad graphic is worse th... | | 24/2/2557 14:20:00:00 | | 00:00:10:00 | [illegible] | [illegible] | |
| Good words need great graphics/A bad graphic is worse th... | | 24/2/2557 14:20:00:00 | | 00:00:10:00 | [illegible] | [illegible] | |

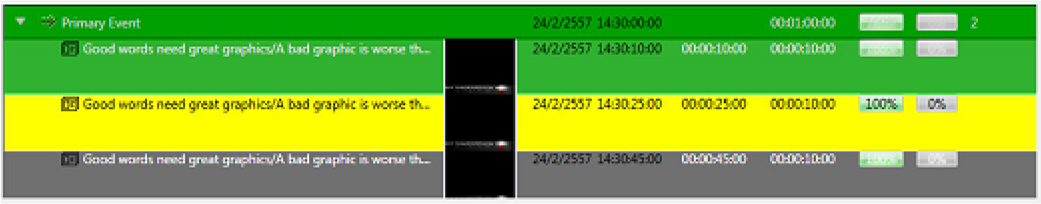
- **Cue state** – The state of an element in a group has been called by MSE, to prepare for playing. These elements are shown in *Yellow*.

- **Pre-queue state** – The state of a next element that will start playing within 5 seconds. Elements are shown in *Orange*.



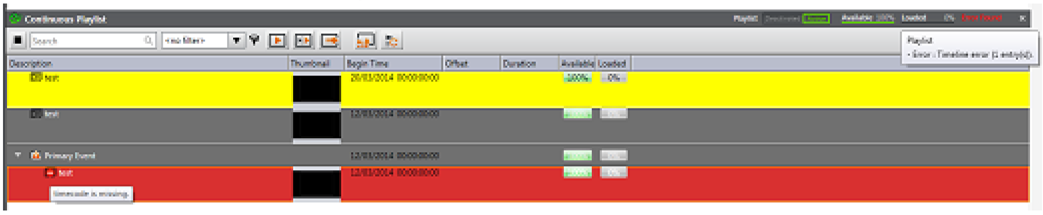
Primary Event 24/2/2557 14:30:00:00 00:01:00:00 100% 0% 2
Good words need great graphics/A bad graphic is worse th... 24/2/2557 14:30:10:00 00:00:10:00 00:00:10:00 100% 0%
Good words need great graphics/A bad graphic is worse th... 24/2/2557 14:30:25:00 00:00:25:00 00:00:10:00 100% 0%
Good words need great graphics/A bad graphic is worse th... 24/2/2557 14:30:45:00 00:00:45:00 00:00:10:00 100% 0%

- **Playing state** – The state of an element which are currently playing. These elements are shown in *Green* color. Cue state of the playlist that are activated by an *external trigger* have a *yellow* arrow in front of its element, otherwise the playing state has a *green* arrow in front. External triggering can for example be a VDCP-signal from the automation system



Primary Event 24/2/2557 14:30:00:00 00:01:00:00 2
Good words need great graphics/A bad graphic is worse th... 24/2/2557 14:30:10:00 00:00:10:00 00:00:10:00
Good words need great graphics/A bad graphic is worse th... 24/2/2557 14:30:25:00 00:00:25:00 00:00:10:00 100% 0%
Good words need great graphics/A bad graphic is worse th... 24/2/2557 14:30:45:00 00:00:45:00 00:00:10:00

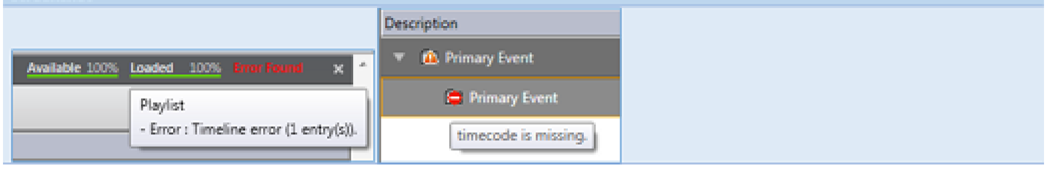
- **Error state** – The state of an element where it has been set incorrectly and cannot be played. Elements in error are shown in *Red* and will also have a tool-tip indicating what caused the error, if the error condition provides this information. Most errors do.



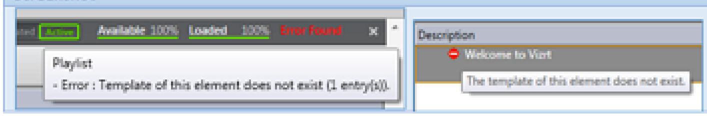
## 15.4.10 Playlist Errors

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**. It's cause must be be investigated and corrective action to prevent it from happening again must be taken. Errors can occur because the playlist was set incorrectly, or the elements reside in the playlist aren't created correctly or due to some other circumstance. Errors are separated into 6 groups as shown in the following list and one additional error type if the MAM system Viz One is enabled:

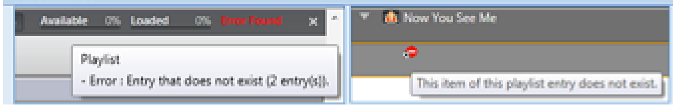
- Timeline error:

| **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** | |
|  | |

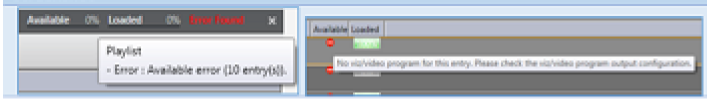
- Template does not exist:

| | |
|---|---|
| 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 | |
| <br>Playlist<br>- Error : Template of this element does not exist (1 entry(s)).<br>Description<br>Welcome to Vizrt<br>The template of this element does not exist. | |

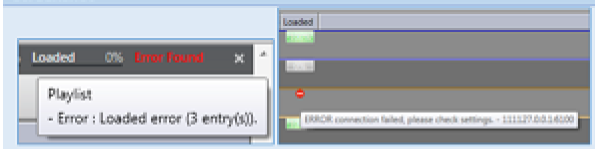
- Entry does not exist:

| | |
|---|---|
| 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 | |
| <br>Playlist<br>- Error : Entry that does not exist (2 entry(s)).<br>Now You See Me<br>This item of this playlist entry does not exist. | |

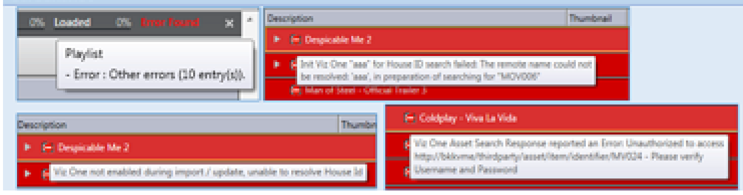
- Available error for playout devices:

| | |
|---|---|
| Error Type | **Available error** |
| Description | This error occurred when the **Playout Devices** isn't set correctly in a setting form. |
| Screenshot | |
| <br>Playlist<br>- Error : Available error (10 entry(s)).<br>No viz/video program for this entry. Please check the viz/video program output configuration. | |

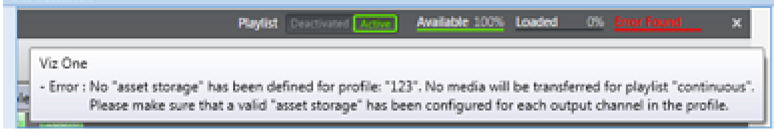
- Loading error:

| | |
|---|---|
| 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 | |
| <br>Playlist<br>- Error : Loaded error (3 entry(s)).<br>ERROR connection failed, please check settings. - 111127.0.0.16100 | |

- 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 | |
| <br>Playlist<br>- Error : Other errors (10 entry(s)).<br>Despicable Me 2<br>Init Viz One "aaa" for House ID search failed: The remote name could not be resolved: 'aaa', in preparation of searching for "MOV006"<br>Man of Steel - Official Trailer 3<br>Despicable Me 2<br>Viz One not enabled during import / update, unable to resolve House Id<br>Coldplay - Viva La Vida<br>Viz One Asset Search Response reported an Error: Unauthorized to access http://bklovme/thirdparty/asset/item/identifier/MV024 - Please verify Username and Password | |

- Viz One specific errors:
  Viz One Error is an error related to an asset storage which has a specific detail of 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 | |
| <br>Viz One<br>- Error : No "asset storage" has been defined for profile: "123". No media will be transferred for playlist "continuous". Please make sure that a valid "asset storage" has been configured for each output channel in the profile. | |

## 15.5 Working With Video Clips

Viz Multichannel integrates with Viz One which allows you to add video clips to your graphics, and conversely. Graphics with video clips can be added to the playlist as secondary events.

In order to preview video clips locally you need to install and configure the relevant codecs (see Video Preview Configuration). In order to work with graphics and video clips together you also need to have a Preview Server configured and running.

This section contains information on the following topics:

- Preview of Video Clips
- Video Clips in Playlists
- Video Clips and Graphics

### 15.5.1 Preview of Video Clips

Preview of a video clip inside Viz Multichannel is based on a low-resolution 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

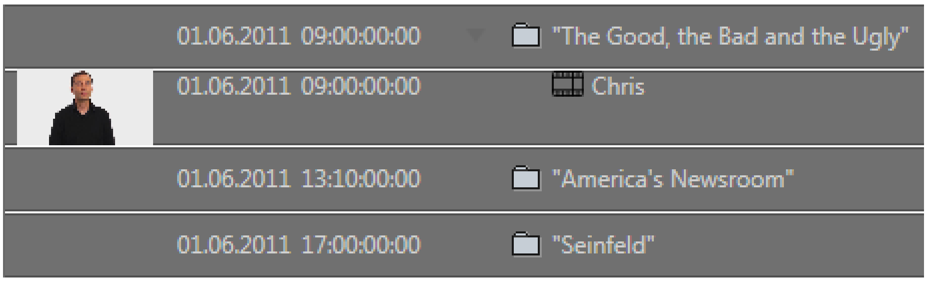
1. Click the Clips button.
2. Search for a video.
3. Double-click the video to preview it.
   Alternatively, you can

1. Click the Playlists button and open your playlist.
2. Search for a video.
3. Double-click the video to preview it.

### 15.5.2 Video Clips in Playlists

Note that video clips as **secondary events** may be manually added to the playlist; however, it is more common to import them from the schedule.

#### To Add Video Clips to the Playlist



01.06.2011 09:00:00:00 "The Good, the Bad and the Ugly"

01.06.2011 09:00:00:00 Chris

01.06.2011 13:10:00:00 "America's Newsroom"

01.06.2011 17:00:00:00 "Seinfeld"

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.

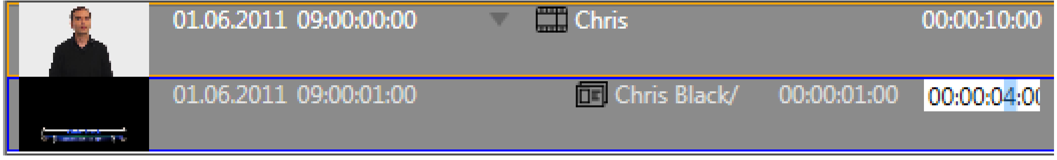
## 15.5.3 Video Clips and Graphics

Viz Multichannel supports two ways of combining graphics and video. You can either 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 Graphics to Video Clips

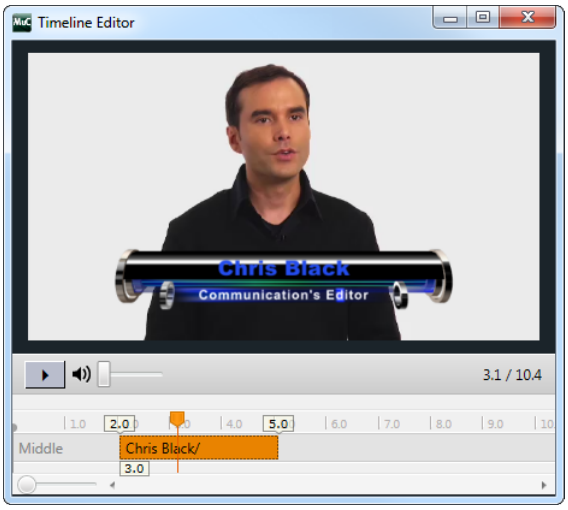
1. Select your **channel**
2. Click the Playlists button and open your playlist.



01.06.2011 09:00:00:00 Chris 00:00:10:00

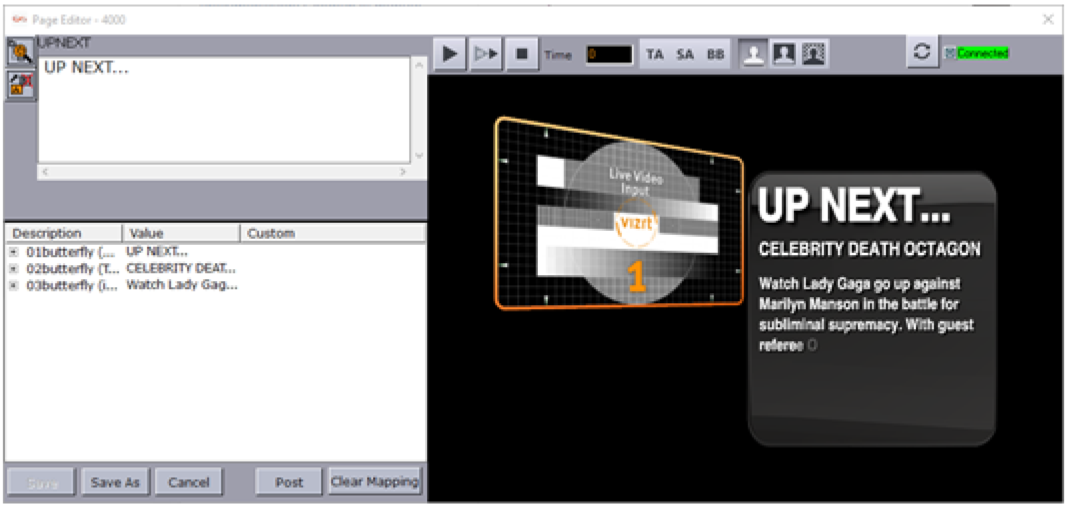
01.06.2011 09:00:01:00 Chris Black/ 00:00:01:00 00:00:04:0

3. Add a *video clip as a secondary event* (see To Add Video Clips to the Playlist).
4. Add a page to the video clip (see To Manually Add a Page to the Playlist).
   This will add the page as sub element of the video clip, grouping them together.
   If the page does not have a default duration set, you can set one by editing the page’s **Duration** column.



5. Double-click the video clip to open the Timeline Editor.
   This will allow you to adjust the offset and duration of the graphics.

### To Add Video Clips to Graphics



UPNEXT

UP NEXT...

| Description | Value | Custom |
|---|---|---|
| 01butterfly (... | UP NEXT... | |
| 02butterfly (T... | CELEBRITY DEAT... | |
| 03butterfly (i... | Watch Lady Gag... | |

UP NEXT...

CELEBRITY DEATH OCTAGON

Watch Lady Gaga go up against Marilyn Manson in the battle for subliminal supremacy. With guest referee 0

1. Select your **channel**.
2. Click the Templates and Pages button.
3. Open your template or page to open the Page Editor.
4. Select the tab field that will hold the video.
   This will open the video search frame.
5. Search for and select (e.g. double-click) a video.
   If the clip channel is set to automatically play the video you should be able to see a low-resolution version of the video.
6. Click **Save As** and enter a new page name.

**See Also**

- Page Editor
- Timeline Editor

## 15.6 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:

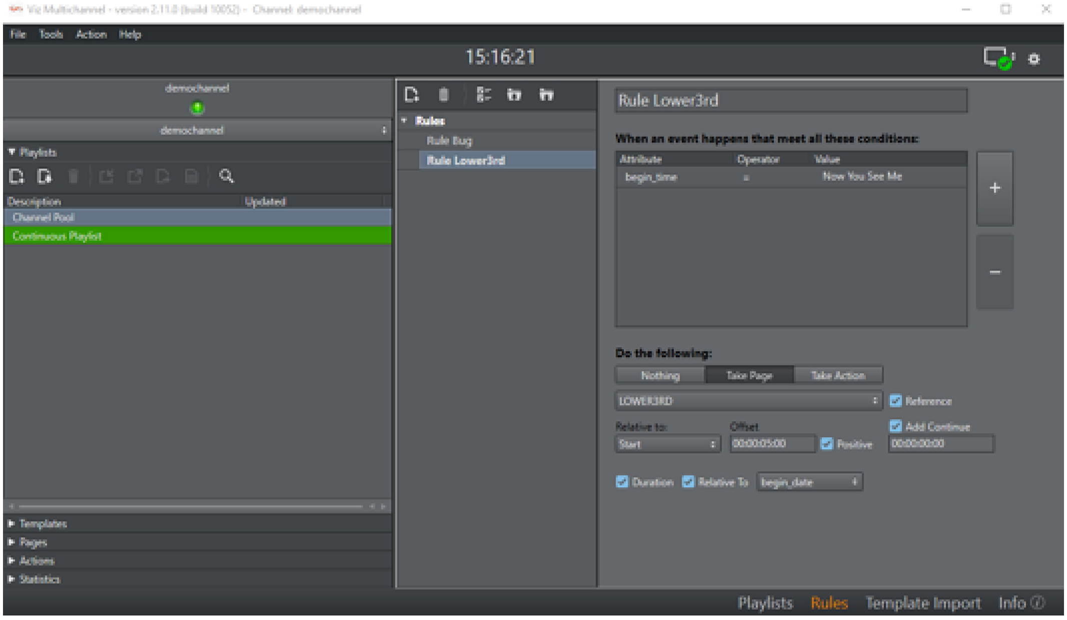
- When schedules are received (or changed):
  - For each event in the schedule the rules engine will try to *match* each rule against the current event read.
  - *If* the rule matches *then* apply the action/page specified in the rule, *otherwise* do nothing and insert the event to the playlist unchanged.
- Repeat this until all events are read and processed.

- Rules UI and using Rules Overview

## 15.6.1 Rules UI and using Rules Overview

To create, delete or change the definitions of rules:

1. From the main menu, make sure that the channel you want to work with is selected.
2. Select **Rules** from the menu actions, this will bring up the user interface for working with rules:



Rule Lower3rd

When an event happens that meet all these conditions:

| Attribute | Operator | Value |
|---|---|---|
| begin_time | = | Now You See Me |

Do the following:

Nothing | Take Page | Take Action

LOWER3RD ☑ Reference

Relative to: Start

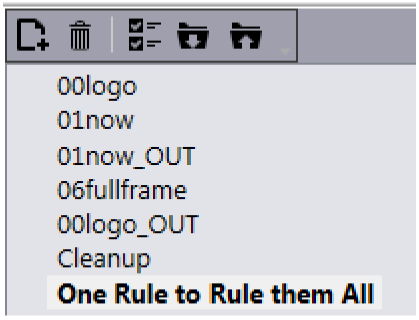
Offset: 00:00:05:00 ☑ Positive

☑ Add Continue 00:00:00:00

☑ Duration ☑ Relative To begin_date

In the screenshot above, on the right-hand side we now have two window panes:
In the left pane we can create, delete, activate/deactivate, select and import/export rules.
In the right pane we can work with a selected rule: name it and give it the appropriate conditions and actions.

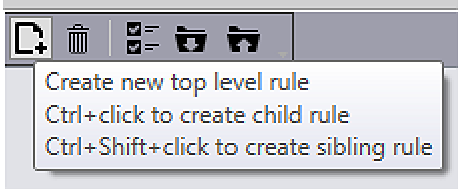
### 15.6.2 Working with Rules



00logo
01now
01now_OUT
06fullframe
00logo_OUT
Cleanup
**One Rule to Rule them All**

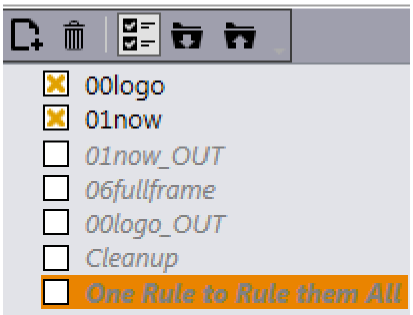
Select by clicking the rule you want to work with. Otherwise select the appropriate icon to create, delete, activate/deactivate or import/export rules.

Rules can be grouped into a hierarchy with children and siblings. Hover the mouse 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:



Create new top level rule
Ctrl+click to create child rule
Ctrl+Shift+click to create sibling rule

Existing rules can be activated/deactivated as appropriate. In the example below only the rules named *00logo* and *01now* are activated. The other rules are present, but will not be acted upon.

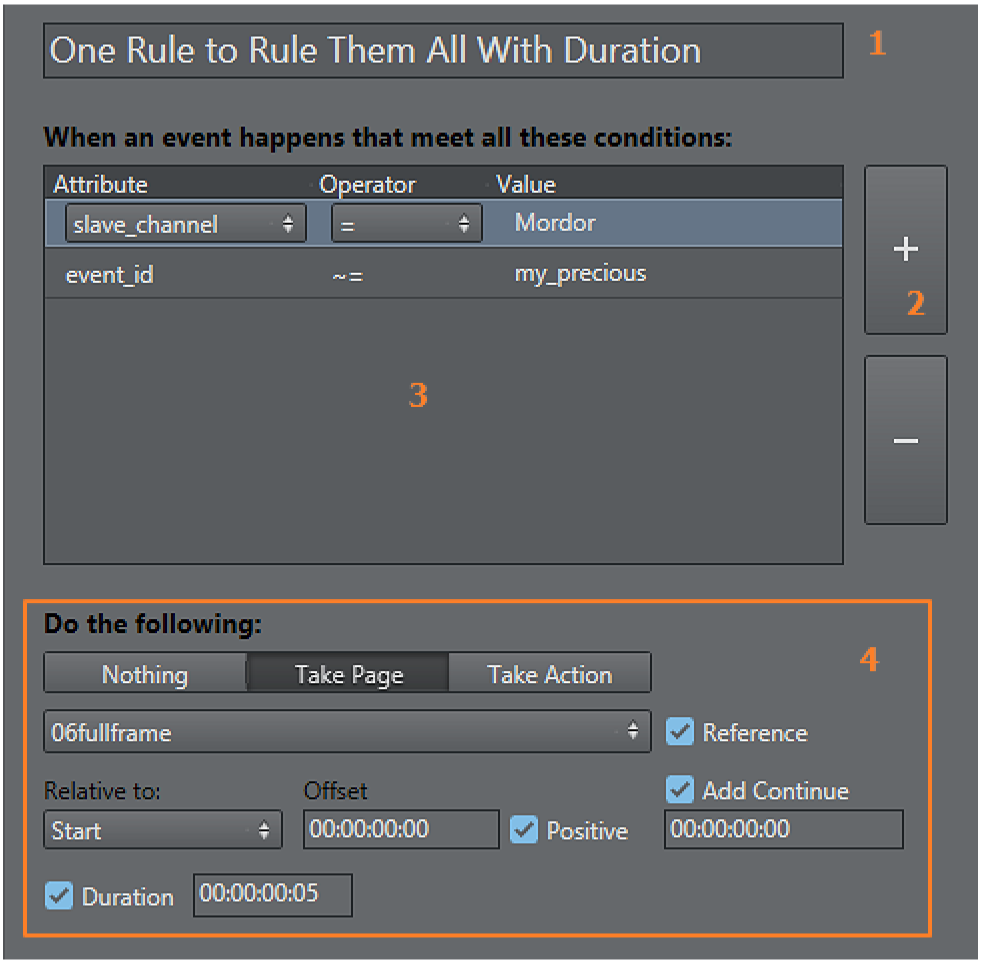


☒ 00logo
☒ 01now
□ *01now_OUT*
□ *06fullframe*
□ *00logo_OUT*
□ *Cleanup*
□ ***One Rule to Rule them All***

Import/Export of rules are from/to XML-files. When importing rules, you will overwrite existing rules. So make sure you want to do this (or create a copy safely stored to disk as backup) before importing.

### 15.6.3 Defining Rules to have Conditions and Actions

Having selected a rule, you can use the rules panel to create conditions and actions for the rule:



One Rule to Rule Them All With Duration

When an event happens that meet all these conditions:

| Attribute | Operator | Value |
|---|---|---|
| slave_channel | = | Mordor |
| event_id | ~= | my_precious |

Do the following:

06fullframe

☑ Reference

Relative to: Start

Offset 00:00:00:00

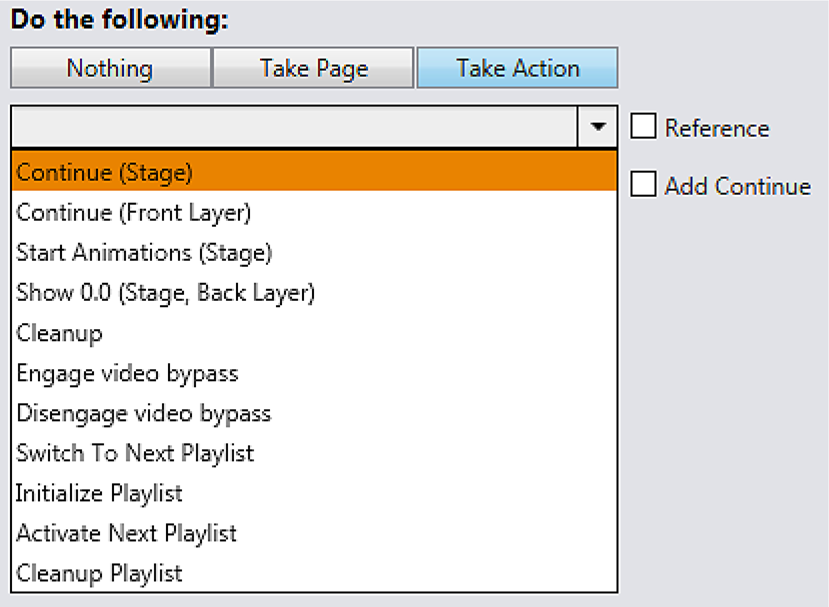
☑ Positive

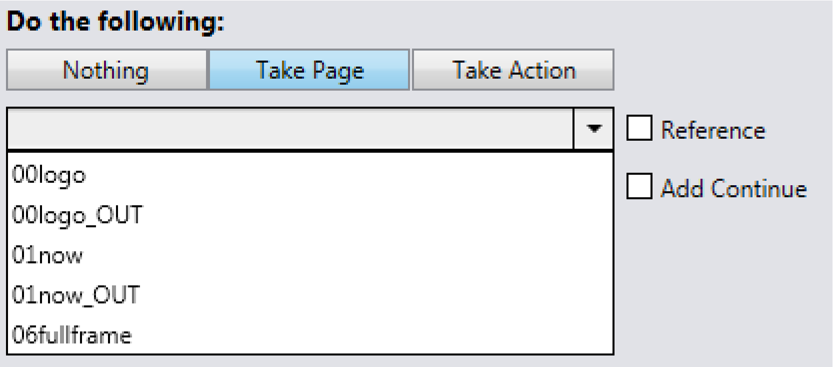
☑ Add Continue 00:00:00:00

☑ Duration 00:00:00:05

Referring to the screen-shot above:

1. Name the rule (if newly created)
2. Create or delete *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. The various condition-types are explained in a little bit more detail in the next section Rules criteria and conditions.
4. *If* all conditions for the rule are met *then* the rule will be acted upon and you can specify what should happen. This could be to do *Nothing*, take a specific *Page* or perform a predefined *Action*. Clicking on the **Page** or **Action** button and the dropdown list below will allow you to select either the appropriate page or action as illustrated in the screenshot below:





Modifiers / checkbox selectors in the Rules UI panel:

- Select the **Reference** checkbox to allow page/actions to be added as a referenced page/ action.
- Select the **Add Continue** checkbox to add Continue action after the event added by rules. When this option is checked, it allows to set offset time for the continue action.
- **Relative to**: the starting reference point for time calculations, example: Start or End.
- **Offset**: The offset from the selected reference point.
- **Positive**: if selected, the offset is positive from the reference point.
- **Duration**: Duration for the action. If no duration is set, the call-back will be the origin page.

## 15.6.4 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. *Without* 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 that 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 (e.g. 19:00:00) Viz Multichannel can look up elements that start after a given time (>) or at a given time and later (>=). In addition Viz Multichannel can do partial matches on strings which allows you to find primary elements that is “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 have no impact on the final result as there is no adjustment for any pages already added. Said in another way: when setting up the rules you should make sure that the end-result you are seeking is not dependent on the order the rules run.

While defining rules, there might be a possibility that many of them share some common conditions. So having a parent rule defined with such conditions, we can define the remaining in child rules. Hence, the child rules will only run if *all* the conditions in *parent rule* is met.

## 15.6.5 Partial Match

In order to find partial matches such as “NEWS*” to match against “NEWSROOM” you 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 would be a proper check of “time” (e.g. at 19:00:00) and start-time of a primary element (e.g. at 18:45:00) and its duration (e.g. 00:30:00) and detect that the primary element is currently on air.

## 15.6.6 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 |

| Operator | Description |
|---|---|
| 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 out of scope to explain regexp in detail in this manual, so please use one of the many books or internet resources that explains RegExp.

Some simple examples for the RegExp operator are shown below.

| 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 + |

## 15.6.7 Back up and Restore 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**.

## 15.7 Resolving Conflicts

Viz Multichannel ensures that if all necessary data is available to the event, there will never be conflicts or clashes. The playlist must include the following data for every instance:

- Event ID
- Primary event begin time
- Primary event duration
- Secondary event offset
- Secondary event duration

## 15.8 Sending Playlists To The Master Control

Once all aspects of the Playlist have been completely and properly configured, you can send it to the master control room (MCR), for broadcast.

### 15.8.1 To Send a Playlist to the Master Control

1. Select your channel.
2. Click the Playlists button.
   All existing playlists for the selected channel are displayed.
3. Right-click the relevant Playlist, and select **Send Playlist** from the appearing context menu.
   The Playlist is sent to the MCR machine indicated when defining the remote host(s) in the Remote Hosts field.

## 15.9 Generating The Channels File

Once you have finished all scheduling tasks, before proceeding to monitor your playlists in MCR mode, you should manually regenerate the channels file. It is essential that the IP addresses for the **Remote Hosts** have been properly configured per-channel in the General Settings.

## 15.9.1 To Regenerate 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 Multichannel to overwrite the existing file.

3. Click **Yes**.

The channels file is normally located in *C:\ProgramData\Vizrt\Multichannel* and is named `channels.ini`. It is a normal text file and can be manually edited.

# 16 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
- Monitoring
- Live Schedule Updates
- Failover Procedure

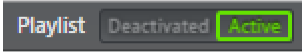
## 16.1 Activate The Playlist

A playlist that has been activated for the currently selected channel enables Media Sequencer to start transfers of full screen video clips from your media asset management system (e.g. Viz One) to Viz Engine for playout.

When activated the Media Sequencer is made aware of the playlist’s schedule. Without it, the 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 may be automatically activated based on Rules and/or Day Switch Options (the latter is common for broadcast day playlists) or manually by clicking the Activate button (when the playlist is open).

### 16.1.1 To Manually Activate the 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.

## 16.2 Initialize The Playlist

An initialized playlist, initialized for the currently selected channel, has all its pages (i.e. the graphics which includes embedded video clips and imagery) loaded on the program Viz Engine for playout.

When activated and initialized the Media Sequencer is aware of the playlist's schedule and is able to Run the Playlist.

A playlist may be initialized based on Rules (common for continuous playlists), Day Switch Options (common for broadcast day playlists) or manually by clicking the Initialize button (when the playlist is open and active).

### 16.2.1 To Manually Initialize the 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. From the Playlist Toolbar, click the **Initialize Playlist** button.
6. Run the Playlist.

---

## 16.3 Run The Playlist

A playlist in run mode has been activated and initialized for the currently selected channel. This means that all full screen video clips and all pages (i.e. the graphics which includes embedded video clips and imagery) are available and loaded on the program Viz Engine for playout.

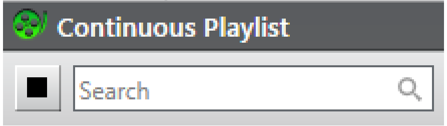
When in run mode the Media Sequencer has a playlist that is activated and initialized and the system can take elements on air.

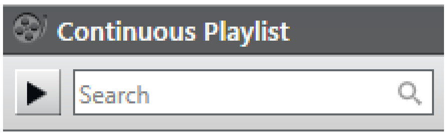
A playlist is in almost all cases automatically run by the Media Sequencer. If there are multiple broadcast day playlists, they run according to **Day Switch Options**. A playlist can be manually run by an operator, but this is not recommended nor common.

**Note**: Only one playlist per channel can be in Run mode at a time.

### 16.3.1 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. Initialize the Playlist.
6. From the Playlist Toolbar, click the **Run Playlist** button.





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.

Whenever **manual intervention** is required, you may 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.

## 16.4 Triggers

Triggers can be set on Primary Events an Secondary Events (i.e. pages and video clips). Note that triggers are configured in the Basic Settings. Triggers are only accepted when a Playlist is in Run mode.

### 16.4.1 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 Viz Multichannel. Playout of the secondary events is relative to the time of the trigger of the primary event and based on the secondary events' defined offset.

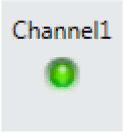
### 16.4.2 Secondary events

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 Multichannel will simply act as a slave system listening for triggers to take events on air.

## 16.5 Monitoring

In most cases Viz Multichannel requires little or no monitoring as it in most cases is the automation system that controls the playout. Also, in the Control Room you typically monitor playout live on TV displays set up for that purpose. In addition you can also see the same on the dedicated Viz Engine machine (if one is setup).



When monitoring is required you can see the status of the Media Sequencer in the left pane. If the channel configuration is OK, lights will be green. If there is any problem, lights will be red. Mouse over the red light for an indication of the problem. This will tell you if the Media Sequencer and your configured playout devices are working properly.

### 16.5.1 To Disable the System Status LED Panel

1. In Windows Explorer, browse to Viz Multichannel’s installation folder and open the `VizRundown.ini` file in the text editor of your choice.
2. Under `GUI OPTIONS`, ensure that the value for `IsIndicatorsPanelEnabled` is set to **0**.
3. Save and close the file.
4. Restart Viz Multichannel.

## 16.6 Live Schedule Updates

If playlists are expected to by highly changeable, it is recommended that updates be done in the scheduling system and updates to Viz Multichannel be done automatically and frequently, using the automatic import.

Once a playlist is in Run mode, Viz Multichannel can be 100% synchronized with the schedule from the automation system.

## 16.7 Failover Procedure

In the case of failover, you need to be prepared beforehand to accommodate such a situation. The following procedure will help you be prepared and to subsequently take additional actions as required.

The prerequisite to backing up is to have an extra machine for the Viz Engine and Media Sequencer, as well as for the Viz Multichannel client.

### 16.7.1 To Back up your Viz Multichannel Client

1. Prepare the client machine as described in the Playout section.
2. From the main client, copy the `channels.ini` files to the backup client.
3. In the event of a client failover, launch Viz Multichannel from this backup client.

### 16.7.2 To Back up your Channels

1. Prepare a Viz Engine machine that can accommodate any existing channels, meaning with Viz Engine and Media Sequencer installed.
2. Add the remote host as a new remote host for all channels to this backup machine, in accordance with the General Settings.

### 16.7.3 Channel Failover Procedure

1. Modify the `channels.ini` file on the Viz Multichannel client so that failed channel points to the backup machine, as described in the Playout section.
2. Restart the Viz Multichannel client.
3. Select the failed channel, initialize today’s playlist, and run it.
4. Set the video router to point to the new machine.

**See Also**

- Traffic Department Workflow

# 17 Logging

## 17.1 Daily Log

For information and troubleshooting purposes there is a daily log file that provides details on every event that was played. The file is located on the **Media Sequencer** machine at: `%ProgramData%\Vizrt\Media Sequencer`. It is not set up as part of the installation process, hence, 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 that calendar day, which may not necessarily corresponds 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.

Below is an example of a log file's contents:

```
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
2007-12-18 12:18:12: Running: read on /VizRundown/channels/AD/rundowns/21_11_2005_AD/
data/102
2007-12-18 12:18:17: Running: read on /VizRundown/channels/AD/rundowns/21_11_2005_AD/
data/102
```

## 17.2 As Run Logs

Daily log files in and of themselves are difficult to understand. However, you can submit them to log analysis to mine them for useful information.

In order to analyze the As Run logs you need to manually transfer the logs to your machine. The log directory on your remote Media Sequencer is set under the Logging section.

Each log entry contains the event date and time, template name, with the content of each template field.

If you want to perform simple analysis of your logs using Viz Multichannel you can use its built-in analysis tool.

### 17.2.1 Analyzing a Log

1. Start Viz Multichannel
2. Select **Analyze Log** from the **Tools** menu
3. Select your **channel**.
4. Select the **from** and **to** dates and click **OK.**
5. Select the **log folder** where you transferred the files to.
6. Click **OK**.
    - A log file is replicated with the same file name except it starts with `Report-`.

### 17.2.2 Standard Log Sample

Below is an example of a standard log file's contents:

```
Date|Time|Action|Event Type|Description|Values
2013-05-31|14:03:57.970:|take|Element|1000|Fields: 1:
```

### 17.2.3 JSON Log Sample

Below is an example of a JSON log file's content:

```
2006-10-06 23:29:34:  Template: START_SQUEEZE Fields: 30: IMAGE*world_sports/swoosh
31: IMAGE*YES_NEW/YesStars/Empty_Pic2 32: IMAGE*YES_NEW/YesStars/Empty_Pic2 Header:
Tonight IMG1: IMAGE*YES_NEW/YesStars/Empty_Pic IMG2: IMAGE*YES_NEW/YesStars/Empty_Pic
IMG3: IMAGE*YES_NEW/YesStars/Empty_Pic Text1: Special Text2: aaaaaaaaaaaaa
2006-10-06 23:29:37:  Template: Top-Year Fields: 1: <fo:wrapper scale-
x="0"><fo:wrapper scale-y="0">1</fo:wrapper></fo:wrapper><fo:wrapper scale-
x="0"><fo:wrapper scale-y="0">9</fo:wrapper></fo:wrapper><fo:wrapper scale-
x="0"><fo:wrapper scale-y="0">9</fo:wrapper></fo:wrapper><fo:wrapper scale-
x="0"><fo:wrapper scale-y="0">2</fo:wrapper></fo:wrapper>
2006-10-06 23:29:38:  Template: PURPLE_TOP_COMM_CLIP Fields: Text1: bla bla  Text2:
soccer wife Title1: help me Title2: today 22:00 comm_clip: ./dataclip/Dog.avi
2006-10-06 23:31:54:  Template: START_SQUEEZE Fields: 30: IMAGE*world_sports/swoosh
31: IMAGE*YES_NEW/YesStars/Empty_Pic2 32: IMAGE*YES_NEW/YesStars/Empty_Pic2 Header:
Tonight IMG1: IMAGE*YES_NEW/YesStars/Empty_Pic IMG2: IMAGE*YES_NEW/YesStars/Empty_Pic
IMG3: IMAGE*YES_NEW/YesStars/Empty_Pic Text1: Special Text2: aaaaaaaaaaaaa
2006-10-06 23:32:26:  Template: START_SQUEEZE Fields: 30: IMAGE*world_sports/swoosh
31: IMAGE*YES_NEW/YesStars/Empty_Pic2 32: IMAGE*YES_NEW/YesStars/Empty_Pic2 Header:
Tonight IMG1: IMAGE*YES_NEW/YesStars/Empty_Pic IMG2: IMAGE*YES_NEW/YesStars/Empty_Pic
IMG3: IMAGE*YES_NEW/YesStars/Empty_Pic Text1: Special Text2: aaaaaaaaaaaaa
2006-10-06 23:33:11:  Template: START_SQUEEZE Fields: 30: IMAGE*world_sports/swoosh
31: IMAGE*YES_NEW/YesStars/Empty_Pic2 32: IMAGE*YES_NEW/YesStars/Empty_Pic2 Header:
Tonight IMG1: IMAGE*YES_NEW/YesStars/Empty_Pic IMG2: IMAGE*YES_NEW/YesStars/Empty_Pic
IMG3: IMAGE*YES_NEW/YesStars/Empty_Pic Text1: Special Text2: aaaaaaaaaaaaa
```

## 17.3 Channel And Playlist Reports

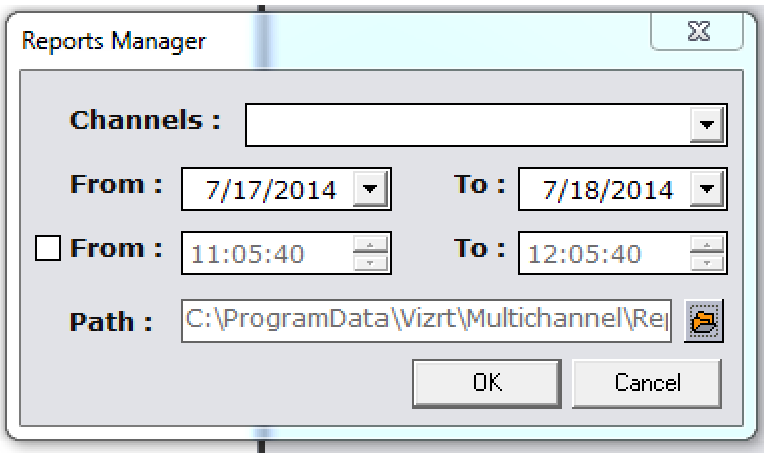
Channel and playlist reports can be created two ways; for a single channel and playlist or for multiple channels and playlists.

### 17.3.1 To Create a Report based on a Single Channel and Playlist

For information on how to set the *Playlist Reports Folder*, see the Miscellaneous section.

1. Select your channel.
2. Click the Playlists button.
3. Right-click your playlist and from the appearing context menu select Generate Report.
    - An HTML report will open in your default browser.

### 17.3.2 To Create a Report based on Multiple Channels and Playlists



1. Select **Tools** from the main menu, and then **Reports Manager**.
2. From the **Channels** combo-box, select all the channels to run the report on.
3. Select the **From** and **To** dates to run the report on.
4. If desired, enable the **From** and **To** times with the check box, and select the times to run the report on.
5. Select the **Path** where the text file report will be saved. The default path used is %ProgramData%/Vizrt/Multichannel/Reports (i.e. normally C:\ProgramData\Vizrt/Multichannel/Reports) but if required you can select another location to use.
6. Click **OK.**
   The resulting report looks as follows:

```
Channel|Date|Time|Event Type|Duration|Program|Template|Fields
Channel1|continuous|2013-05-29T12:55:00.000|Primary|01:34:40:18|Glory God is peeking
through the blinds|
Channel1|continuous|2013-05-29T12:55:00.000|Secondary|00:00:00:00||1000|Bergen in
snow
```

## 17.4 Multichannel Log Files

Multichannel and several multichannel components write log files when they run. The log files are ordinary textfiles and can be helpful for reporting or error detection and debugging.

- The log files are written to the directory *%PROGRAMDATA%\Vizrt\Multichannel*
- This will normally be: C:\ProgramData\Vizrt\Multichannel
- Note that the *ProgramData* directory normally is a Windows “hidden” directory so you will be required to explicit insert *the directory name %PROGRAMDATA%* in the address field in Windows Explorer to reach this directory.
- Each Multichannel component (for instance PlayListImporter, Schedule Collector and so forth) will write it’s log files in a separate directory as documented below.

⚠ **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.

## 17.5 Log File Locations

As explained above, all log files in the list below is relative to the log file base directory, so .*\Logs* means *C:\ProgramData\Vizrt\MultiChannel\Logs*

- Multichannel: .*\Logs*
- Harris Integration Module: .*\HarrisIntegration*
- Pebble Beach Marina Integration Module: .*\MarinaIntegration*
- %ProgramData%\Vizrt\Multichannel\PlaylistImporterV2
  For 32 bit (legacy v1 setup): PlayListImporter: .*\PlaylistImporter*
- SCC Schedule Collector: .*\ScheduleCollector*
- Playlist History Manager: .*\PlaylistHistoryManager*
- Promo Server: *No log-file on disk, use Windows Event Viewer.*
- Integrations Hub: *No log-file on disk, use Windows Event Viewer.*

## 17.6 ELF Log Dump Files

In case of program crash, ELF dump files can be created. ELF files are binary files, sometimes used by Vizrt support for debugging. If generated, the Elf file(s) will be created in:

```
%PROGRAMDATA%\Vizrt\Multichannel\Logs
%PROGRAMDATA%\Vizrt\Multichannel\PlaylistHistoryManager
%PROGRAMDATA%\Vizrt\Multichannel\PlaylistImporterV2
```

# 18 Scheduling System Integrations

A Viz Multichannel system has to integrate with external systems. This can be integrations to other Vizrt systems such as Viz One, Viz Ticker or third party systems. Third party systems are often categorized as being a scheduling or an automation system.

In general, Viz Multichannel 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 and/or other device actions).

Viz Multichannel's integration with scheduling systems is by reading (i.e. importing) the schedule from them. 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). Viz Multichannel imports the file and 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, Viz Multichannel can read the schedule directly from it, skipping the file part.
   Optional integration is reporting based on As Run Logs.

**Scheduling** is the mode used for building a schedule (to Viz Multichannel known as a playlist). This is where scenes are imported, rules are created, the templates' auto-content is configured, etc. In Scheduling mode usually one Viz Multichannel client connects to one Media Sequencer, both installed on the same machine, storing data from multiple (up to all) channels. Scheduling mode is required when manual action is expected in building the playlist in Viz Multichannel (manual secondary events scheduling/editing, preview for approval, etc.). Viz Multichannel running in scheduling mode is usually installed in the traffic department.

⚠ **Note:** Viz Multichannel integrates with a range of scheduling systems not mentioned in this document. Please contact your local Vizrt representative for more information.

**See Also**

- Integration with Automation Systems

# 19 Integration With Automation Systems

The Multichannel playout schedule (also called *playlist* or *running order*) is usually provided from an external source like an automation, traffic and/or scheduling system. An automation system can be, for example Viz Mosart, Miranda, Pebble Beach or Harris.

This chapter describes some common automation system integrations to Viz Multichannel.

## 19.1 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, Character Generators).

A Viz Multichannel integration to an automation system typically performs:

- **Triggering**: Automation system triggers Viz Multichannel (via Media Sequencer) for starting of either primary or secondary events. Triggering protocols supported by Vizrt are TCP/IP, Intelligent Interface, VDCP, GPI, and REST. It is important to understand that though Multichannel is managing the playlist and events, the actual triggering of actions will be done by the Media Sequencer.
- **Schedules and last minute updates**: If an automation system provides an API or an option to automatically generate an update file, Viz Multichannel can update its playlist using these. This is usually done with the Schedule Collector and the PlayListImporter tools. The Schedule Collector’s biggest advantage is that it stores previous data in an internal cache and therefore makes updates more efficient by saving idle updates and passing only changed events.

## 19.2 Automation System Integration Types

Exactly how the external automation systems delivers the schedule and triggering varies greatly, but the flexible and open architecture of Multichannel ensures that most commonly used automation systems are directly supported.

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 format, possibly of type Text, XML, Excel, CSV, database or proprietary binary. In this case, the typical workflow will be to have a file or directory produced by the automation system watched (or have time-based tasks running) and use PlaylistImporter and Schedule Collector to fetch schedules and updates to the Multichannel playlist.

**Direct Integration** systems are automation and schedule systems that have an API (Application Programming Interface) that Multichannel can interface and read information from, sometimes bi-directional. The main advantage with direct integration systems is that the file-based workflow can be avoided. Normally this leads to significant performance gains. It 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 read the following sections:

- Direct Integration Architecture Overview
- 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
- Neptune by Pebble Beach Systems

> ⚠ **Note**: If you do not find your automation system listed we may still support it. Please contact your local Vizrt representative for more information.

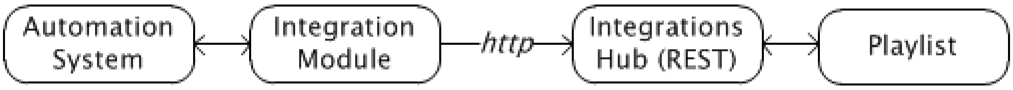
## 19.3 Direct Integration Architecture Overview

The architecture flow of an external system communicating with the Integrations Hub is called Direct Integration (DI). The Integrations Hub component is a REST webservice, allowing for an external system to fully query, add, update and remove contents of a playlist.

The preferred approach is that the external system have a native driver for communicating with the Integrations Hub, using the REST interface, to maintain the sync between the systems. As 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.

### 19.3.1 Integration Module

If the external system lacks the ability to implement a native driver or already have their own webservice, an alternative approach is to use a custom **Integration Module**. The integration module will handle all details about interfacing a specific automation system and forward the required data (schedules etc.) to the Integrations Hub. The specific integration module could be provided by Vizrt, but it is also possible for vendors and 3. parties to develop integration modules.



**See Also**

- Introduction to REST (external wikipedia infopage)
- Integrations Hub

## 19.4 Integration Module For Marina From Pebble Beach

The integration for Pebble Beach (PB) Marina system uses the PB Marina Integrations Module as explained in the Integrations Module section of Direct Integration Architecture Overview.

Before using the PB integration module, make sure that the Integrations Hub service is installed. In most cases, it is also required to have the Promo Server installed.

### 19.4.1 Installing the integration module for PB Marina

Locate the installer: `VizMultichannel_MarinaIntegration-VERSION.exe` where *VERSION* is a four-digit version/release/fix/build number.

1. Double-click `VizMultichannel_MarinaIntegration-VERSION.exe`
2. Press **Next** to start the installation
3. If required change the installation directory. This is normally not required nor recommended. Press **Next**
4. Press the **Install** button and wait for the installation to finish.

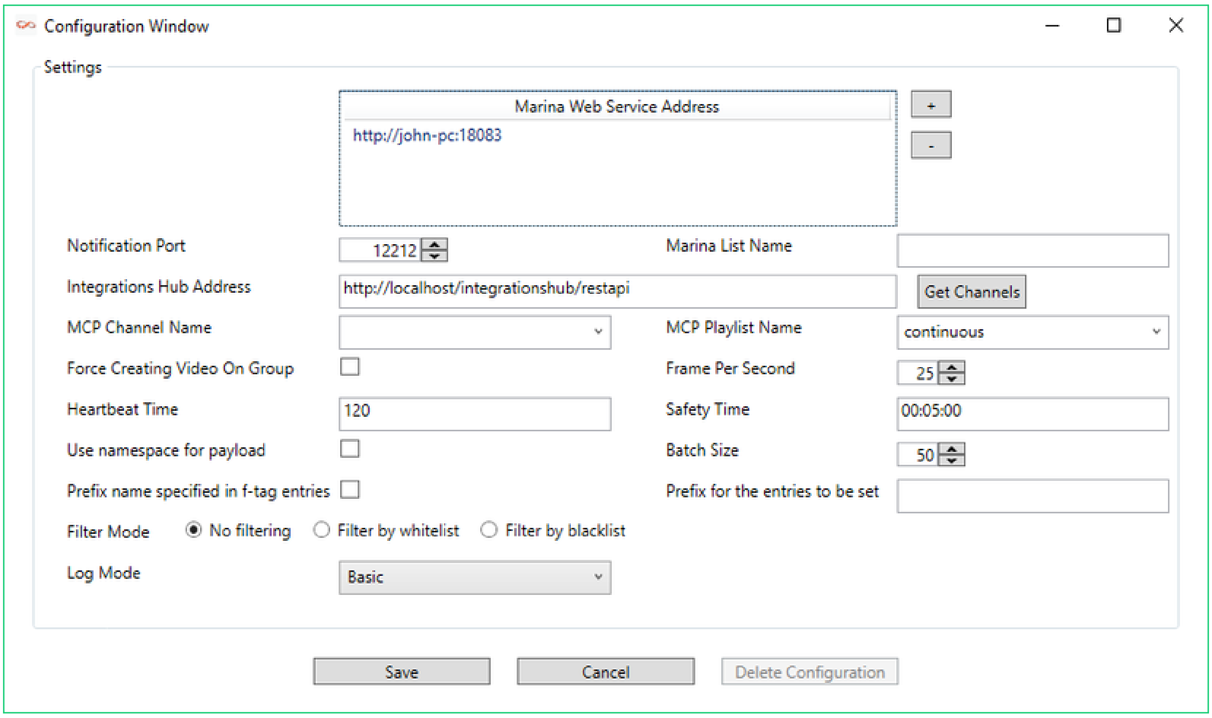
> ⚠ **IMPORTANT!** The PB Marina requires *Administrator privileges* to run. It must be started by a user with Windows Administrator privileges or the application must be elevated upon start-up. How to do this varies slightly between Windows version but the basic concepts are the same.

### 19.4.2 Configuring the Integration Module for PB Marina

Before you start the PB Marina integration module you will need to collect a few details from your local Viz Multichannel and PB Marina administrators. In particular you will need the network address (or IP-address) of Marina servers and hostname where the Integrations Hub is installed. The required information is documented below.

- To start the Marina Integration Module go to **Start > All Programs > Vizrt > Viz Multichannel > Viz Multichannel - Marina Integration**

- Click the New button to create a new configuration. This will display the configuration window:



Configuration Window

Settings

Marina Web Service Address: http://john-pc:18083

Notification Port: 12212

Marina List Name:

Integrations Hub Address: http://localhost/integrationshub/restapi

MCP Channel Name:

MCP Playlist Name: continuous

Force Creating Video On Group □

Frame Per Second: 25

Heartbeat Time: 120

Safety Time: 00:05:00

Use namespace for payload □

Batch Size: 50

Prefix name specified in f-tag entries □

Prefix for the entries to be set:

Filter Mode ● No filtering ○ Filter by whitelist ○ Filter by blacklist

Log Mode: Basic

| Detail | Explanation |
|---|---|
| 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 the possibility to have several housekeeper address ready in advance makes it easy to quickly switch between them when needed. In addition, if an Marina servers is not reachable, perhaps due to Marina server problem or network issues, 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 Multichannel Integrations Hub module. Example: http://MyServer/integrationshub/restapi |
| **Marina List Name** | The Marina list name for the integration |
| **MCP Channel Name** | The channel name you have defined in Multichannel that should be used with this direct integration module |

| Detail | Explanation |
|---|---|
| **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 needs to be enabled in Multichannel |
| **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 |
| **Use namespace for payload** | Switch, either *on* or *off*. This option will depend on the current version / configuration of your PB Marina installation<br>If unchecked: If the PB Marina has AuxData node then the Marina Integration module will generate the element with attributes from the payload node inside AuxData.<br>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, Multichannel will ask for 50 events per each set. If Marina has more than 50 events, Multichannel will ask multiple times until all events are received |
| **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 entries to be **used**: If prefixing f-tag entries, use the prefix specified |

| Detail | Explanation |
|---|---|
| **Filter Mode** | Select from **No Filtering** or filtering by **Whitelist** or by *Blacklist *- explained in the bullets below. Comma-separated values for whitelist or blacklist must be provided in the edit-box.<br>· Filter by **whitelist**: Each elements 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.<br>· Filter by **blacklist**: Each elements 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. |
| **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\Multichannel\MarinaIntegration*. |

- When all required information is entered, press **Save** to save the data and return to the main application window.
- Press **Start** to start the Marina Integration service

## 19.5 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 Direct Integration Architecture Overview.

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.

- Installing the Integration Module for PB Neptune
- Configure the Integration Module for PB Neptune
- Neptune Attributes

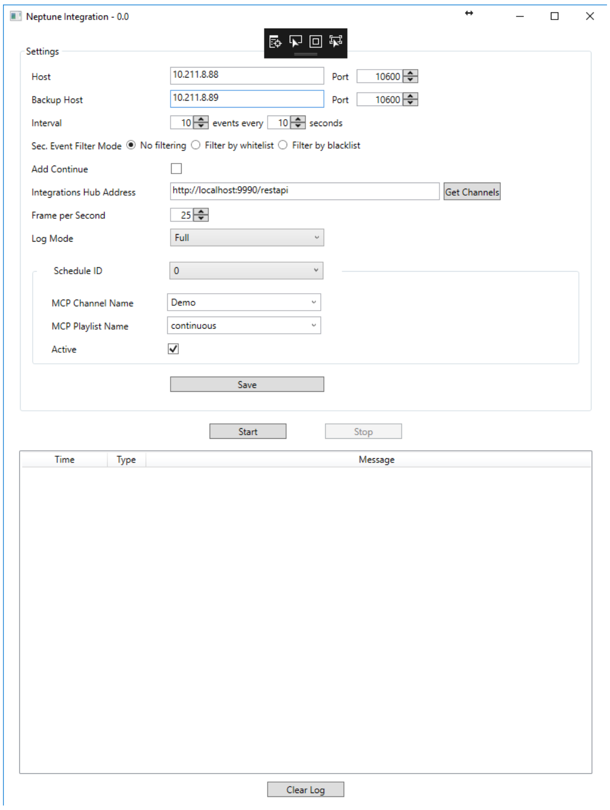
### 19.5.1 Installing the Integration Module for PB Neptune

Locate the installer: *VizMultichannel_NeptuneIntegration-VERSION.exe* where *VERSION* is a version number (format: version.release.fix.build).

1. Double-click *VizMultichannel_NeptuneIntegration-VERSION.exe*
2. Press **Next** to start the installation
3. If required change the installation directory. This is normally not required nor recommended. Press **Next**
4. Press the **Install** button and wait for the installation to finish.

### 19.5.2 Configure the Integration Module for PB Neptune

Before you start the PB Neptune integration module you will need to collect a few details from your local Viz Multichannel and PB Neptune administrators. In particular you will need the network address (or IP-address) of Neptune servers and hostname where the Integrations Hub is installed. The required information is shown in the screen-shot below.



Neptune Integration - 0.0

Settings

Host 10.211.8.88 Port 10600

Backup Host 10.211.8.89 Port 10600

Interval 10 events every 10 seconds

Sec. Event Filter Mode ● No filtering ○ Filter by whitelist ○ Filter by blacklist

Add Continue □

Integrations Hub Address http://localhost:9990/restapi

Frame per Second 25

Log Mode Full

Schedule ID 0

MCP Channel Name Demo

MCP Playlist Name continuous

Active ☑

| Time | Type | Message |
|---|---|---|

| 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.<br>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 |

| Form Input | Description |
|---|---|
| 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.<br>· *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.<br>· *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 Multichannel Integrations Hub module. Example: http://MyServer/integrationshub/restapi |
| MCP Channel Name | The channel name you have defined in Multichannel 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 |
| 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\Multichannel\NeptuneIntegration` |

- When all the required information is entered, press **Save** to save the data.
- Press **Start** to start the Neptune Integration service.

### 19.5.3 Neptune Attributes

All attributes found from Neptune will be added to the corresponding Viz Multichannel **primary** event.

# 19.6 Integration Module For Harris ADC

The Harris automation system can use the Harris Integrations Module as explained in the Integrations Module section of Direct Integration Architecture Overview. The integration is based on the Harris ADC-12 Web API.

Before using the integration module, in most cases, the Promo Server must be installed.

> ⚠ **Note:** Harris is now renamed Imagine Communications (http://www.imaginecommunications.com), this documentation currently uses the old Harris brand name.

- Installing the Harris Integration Module
- Configuration and Startup
- Pushing Optional Metadata from Harris to Multichannel

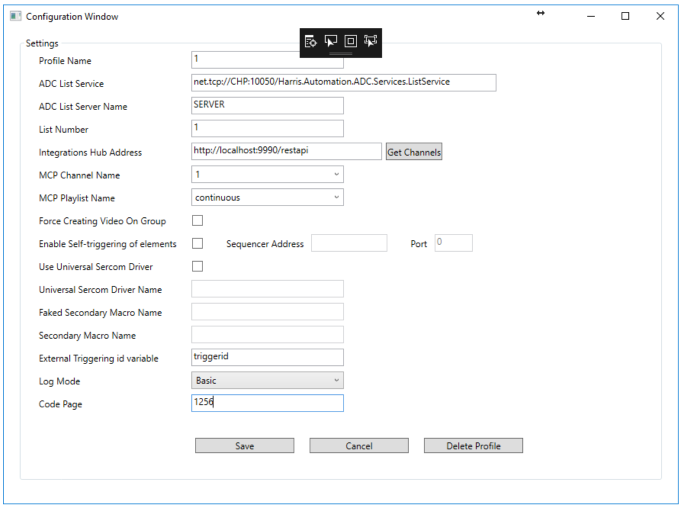
## 19.6.1 Installing the Harris Integration Module

1. To install the Harris Integration Module locate the *VizMultichannel_HarrisIntegration-VERSION.exe* installer where *VERSION* is the version/release/build number. Double click the installer to start the installation.
2. Press **Next** to start the installation
3. If required change the installation directory. This is normally not required nor recommended. Press **Next**
4. Press the **Install** button and wait for the installation to finish.

## 19.6.2 Configuration and Startup

- To start and configure the integration module for Harris, click the Windows **Start** menu and open the program by browsing to **All Programs > Vizrt > Viz Multichannel > Viz Multichannel - Harris Integration**

The Harris integration module will display the Configuration and start window:



| | | | | | |
|---|---|---|---|---|---|
| Profile Name | 1 | | | | |
| ADC List Service | net.tcp://CHP:10050/Harris.Automation.ADC.Services.ListService | | | | |
| ADC List Server Name | SERVER | | | | |
| List Number | 1 | | | | |
| Integrations Hub Address | http://localhost:9990/restapi | | | | |
| MCP Channel Name | 1 | | | | |
| MCP Playlist Name | continuous | | | | |
| Force Creating Video On Group | ☐ | | | | |
| Enable Self-triggering of elements | ☐ | Sequencer Address | | Port | 0 |
| Use Universal Sercom Driver | ☐ | | | | |
| Universal Sercom Driver Name | | | | | |
| Faked Secondary Macro Name | | | | | |
| Secondary Macro Name | | | | | |
| External Triggering id variable | triggerid | | | | |
| Log Mode | Basic | | | | |
| Code Page | 1256 | | | | |

| Detail | Description |
|---|---|
| **Profile Name** | You can have multiple settings profiles. If you run multiple instances of the Harris Integrations service is it suggested to do configuration changes from one and only one instance 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 we want to use "*Playlist 2*" for the integration, specify **2** as the list-number. |
| **Integrations Hub Address** | Specify the URL of the REST service. Note: You can browse to this URL with a browser to see that the Integrations Hub is installed and working. Expected result answer is: *Integrations Hub* |
| **MCP Channel Name** | The channel name you have defined in Multichannel 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** | 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 needs to be enabled in Multichannel |
| **Use Universal Sercom Driver Name** | If checked, Harris Integration will activate the Universal Sercom Driver Name and in doing this also Macro Name in the process |
| **Universal Sercom Driver Name** | This ID must match the ID column in Harris Automation. Ask your Harris administrator for this information |
| **Faked Secondary Macro Name** | For using a secondary event to re-label the Primary events trigger-id, define the macro used here |
| **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. |

| Detail | Description |
|---|---|
| **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\Multichannel\HarrisIntegration*. 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.<br>For example, code page with identifier *1256* will convert bytes to ANSI Arabic. |

- When all the appropriate setting are configured press **Save**
- Press **START**.

### 19.6.3 Pushing Optional Metadata from Harris to Multichannel

ADC systems using the LST12 format can optionally use the generic Harris "*Data*" field (max 4096 characters) to push meta-data information to Multichannel. If the Harris Data-field contains entries of type "*value=somedata*<CRLF>" (or several formatted as "value1=somedata<CRLF>value2=somemoredata<CRLF>" etc) Multichannel will parse these entries and if a valid combination *fieldname=value* is found push this payload to the Integrations Hub.

## 19.7 ADC-100 By Harris Broadcast

This section covers configuring the Automated Content Management and Distribution system (ADC-100) by Harris Broadcast for Viz Multichannel. It describes the setup on the ADC side and contains information on the following topics:

## 19.7.1 General Information and Setup

An ADC-100 system can trigger Viz Multichannel (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 is supported.

Advantages of VDCP are:

- Standard
- Proved to work for many customers
  Advantages of USD are:

- Can be used over TCP/IP thus saving serial ports and additional cabling
  This section contains information on the following topics:

### 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 Multichannel and is must be set in order to receive playlist updates. An empty or corrupted Reconcile value will omit that event from the update procedure.

If **primary events triggering** is selected then Viz Multichannel can import the same LST file that is provided by the ADC-100 system because the data required for the triggering is available in it.

If **secondary events triggering** is selected and all Vizrt secondary events are received from the scheduling system (i.e. in the LST file) then Viz Multichannel can **import** the same LST file that is provided by the ADC-100 system because the data required for the triggering is available in it (as long as secondary events contain assigned ID fields).

If **secondary events triggering** is selected and there are Vizrt secondary events that are generated by Viz Multichannel (manually or by import rules) then Viz Multichannel must **export** a new LST with the data required for the triggering in it.

### Updating LST by Export

While exporting Viz Multichannel will make sure all secondary events have a **Media ID** value assigned (if secondary events triggering is selected) and assign it to the LST ID field.

Either exporting LST from scratch or merging into the original LST is optional. Merging is preferable since it only appends the new secondary events to the LST file and will assure that original data is not touched.

### Updating LST by Import

Two mapping options are available, either with or without secondary events.

Refer to (folder path) folder in Viz Multichannel installation for sample mapping settings and playlists.

Channel name can be read from the filename. If filename does not contain a channel identifier, either do not map channel value or set Traffic Channel Name to 1 in Viz Multichannel's Basic Settings.

### Updating Live

**Recovery (RCV)**: The Harris ADC-100 can auto-generate a recovery file (*.RCV) on every playlist change. RCV files are in LST format. Viz Multichannel can monitor a folder and update its playlist according to the RCV file once it is generated.

**Broadcast Exchange Format (BXF)**: Viz Multichannel supports processing BXF schedules.

**LST files managed by Schedule Collector tool**: Schedule Collector can monitor a folder for LST update files. Using its cache memory, updates will be attempted only for changed events.

### Limitations

The LST format only allows for 32 character titles and misses other necessary fields such as local language titles.

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.

## 19.7.2 Trigger Viz Multichannel Using VDCP

The standard configuration of Viz Multichannel is triggering by a serial connection using Video Disk Control Protocol (VDCP).

Any Device Server version allows controlling Viz Multichannel, provided that it has a Video Disk driver.

### 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 the Media Sequencer to the ADC-100 system.

### ADC–100 to Media Sequencer: Connections

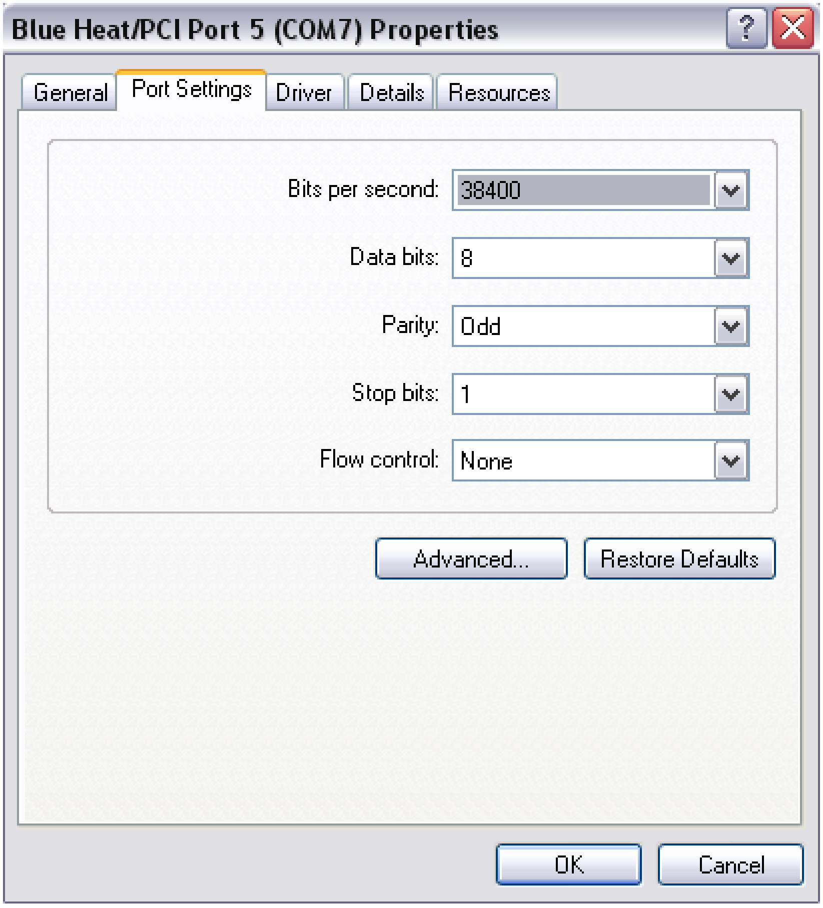
| ADC-100 | Media Sequencer Machine |
|---|---|
| 2 (Rx -) | 2 (Tx -) |
| 3 (Tx +) | 3 (Rx +) |
| 7 (Rx +) | 7 (Tx +) |

| ADC-100 | Media Sequencer Machine |
|---|---|
| 8 (Tx -) | 8 (Rx -) |

## Device Configuration

### To Configure the Software

- Make sure that under Windows Configuration on the Media Sequencer machine, the COM port used has the following suggested parameters.



Blue Heat/PCI Port 5 (COM7) Properties

Bits per second: 38400

Data bits: 8

Parity: Odd

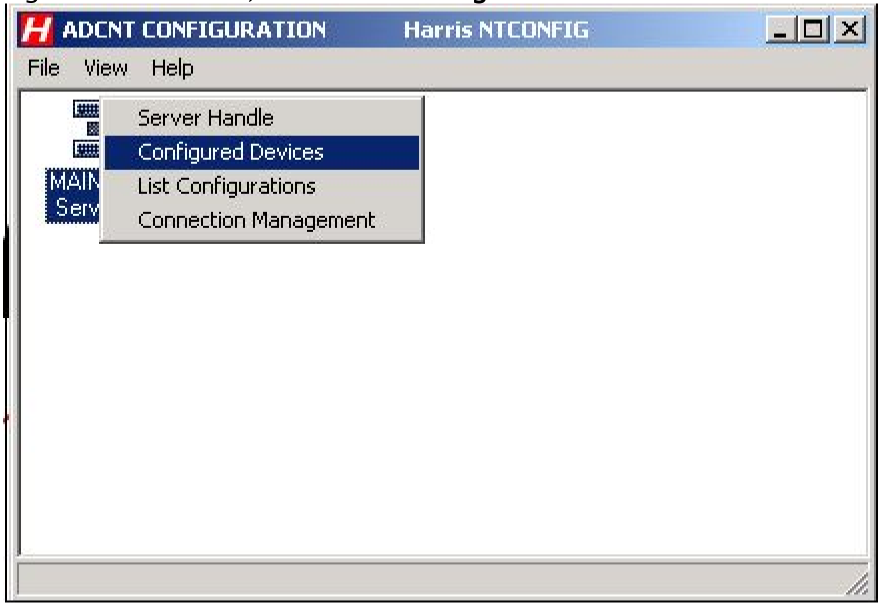
Stop bits: 1

Flow control: None

    - Bits per second: **38400**
    - Data bits: **8**
    - Parity: **Odd**
    - Stop bits: **1**
    - Flow control: **None**
- Enable the VDCP device on Viz Multichannel and ensure it is configured properly with that port and also turned on.

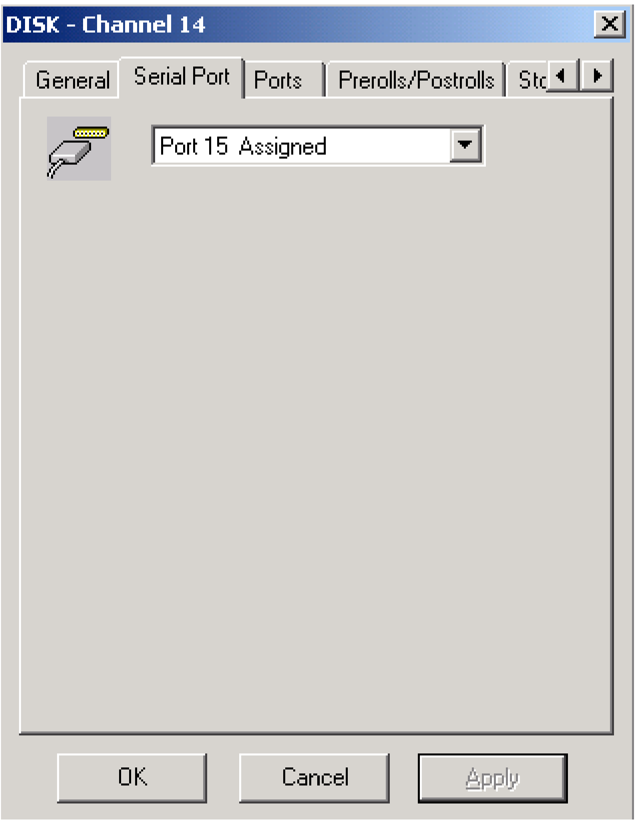
## To Configure the Viz driver

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 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.

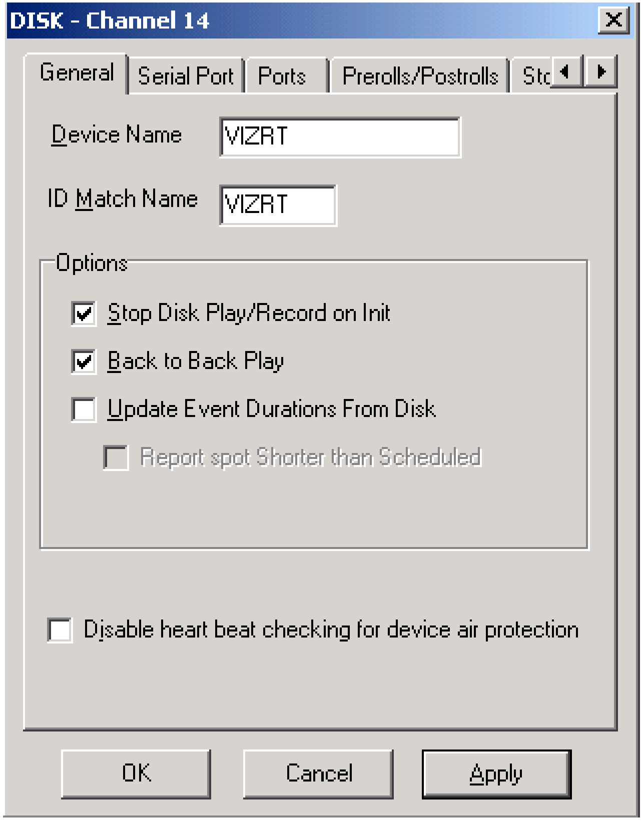
4. Double-click on the Disk device that was just created. From the resulting dialog select the* Serial Port* tab.



Port 15 Assigned

5. Select the physical port the Media Sequencer is connected to. The Harris Device Server is now configured properly to communicate with the Media Sequencer.
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 for easy identification later.
   - **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 set or not depending on the need.
   - **Back to Back Play** can be set or not depending on the need.

- **Update Event Duration from Disk** must be *unchecked*.



Device Name: VIZRT

ID Match Name: VIZRT

Options

☑ Stop Disk Play/Record on Init

☑ Back to Back Play

☐ Update Event Durations From Disk

☐ Report spot Shorter than Scheduled

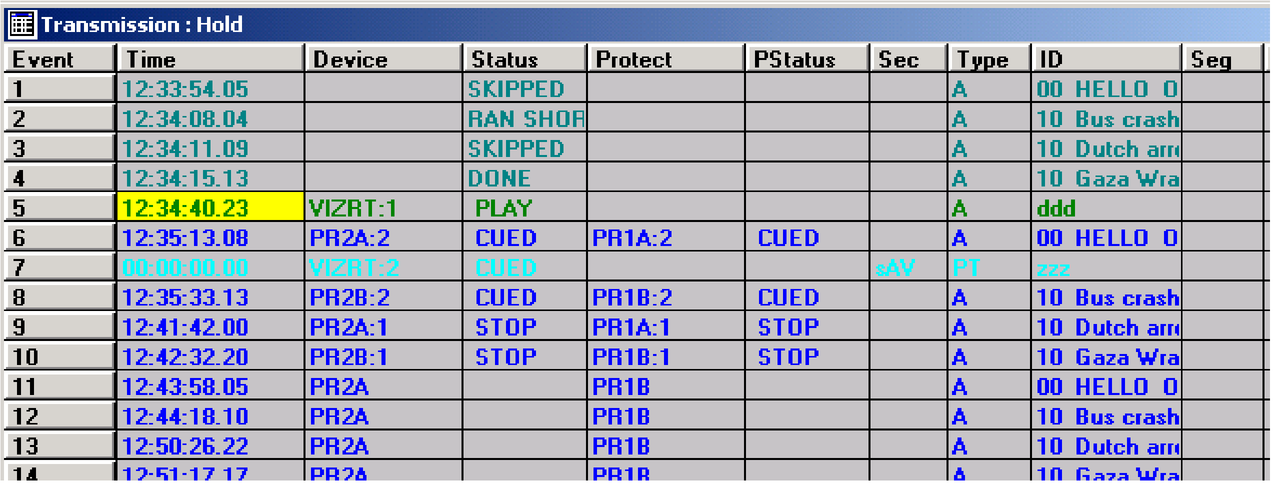
☐ Disable heart beat checking for device air protection

7. The Vizrt Standard Video Disk must also be assigned to a Transmission List to be schedule to air.

> **Note:** If you are using Extended IDs, you may not use the Standard Video Disk driver. Additionally, the Media Sequencer does not have the notion of duration, so the parameter **Update Duration from Disk** must be *unchecked* to avoid too many error messages when ADC driver asks for the IDs list.

## Operation

Viz Multichannel uses Secondary A/V or Secondary Backtimed events to execute the pre-loading and playout of media.



Transmission : Hold

| Event | Time | Device | Status | Protect | PStatus | Sec | Type | ID | Seg |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 12:33:54.05 | | SKIPPED | | | | A | 00 HELLO 0 | |
| 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 0 | |
| 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 0 | |
| 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 | |

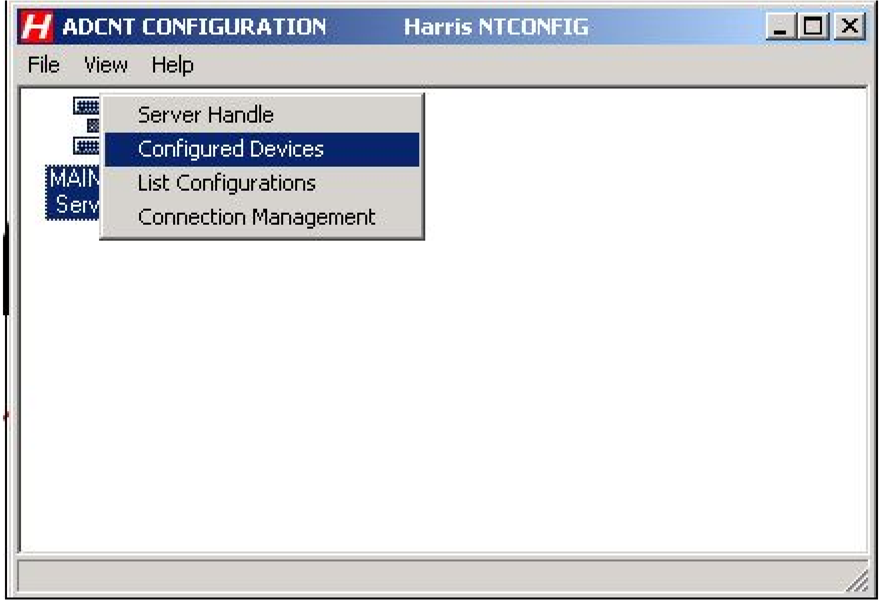
## 19.7.3 Trigger Viz Multichannel Using USD

When configuring Viz Multichannel to be triggered by a Universal Sercom Device (USD) driver you can either configure it to use TCP or Serial Communication.

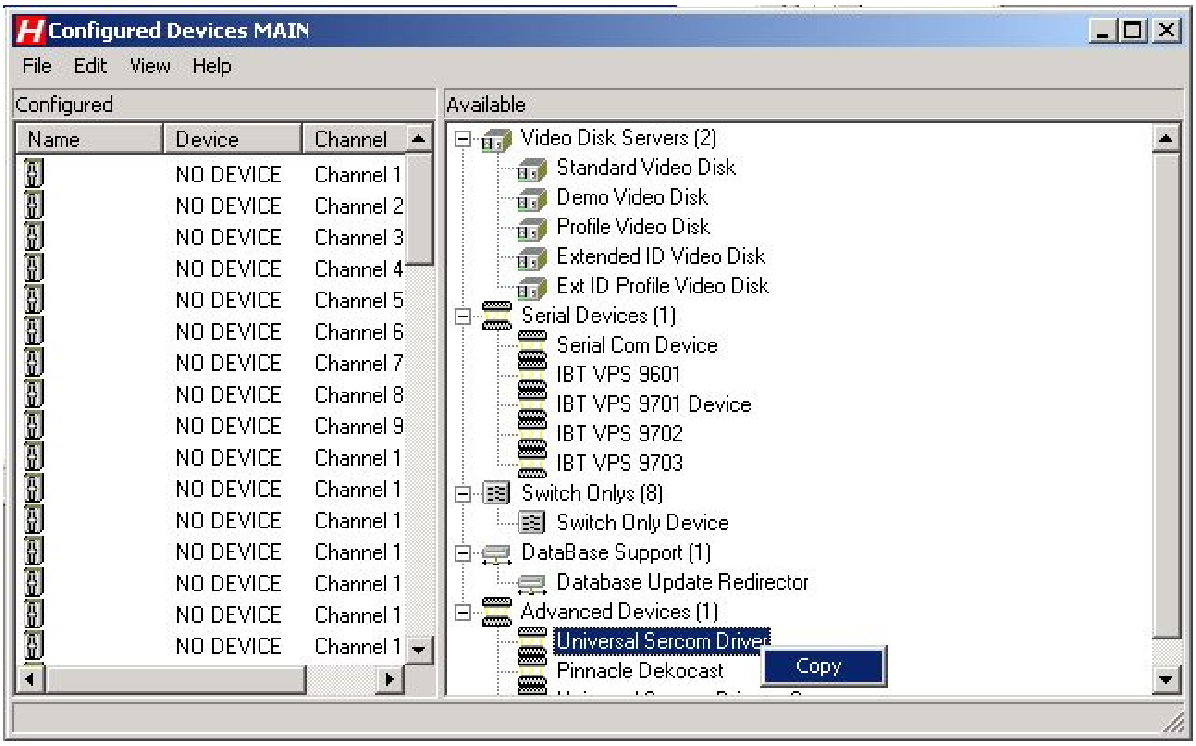
### USD Device Configuration

### 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*.



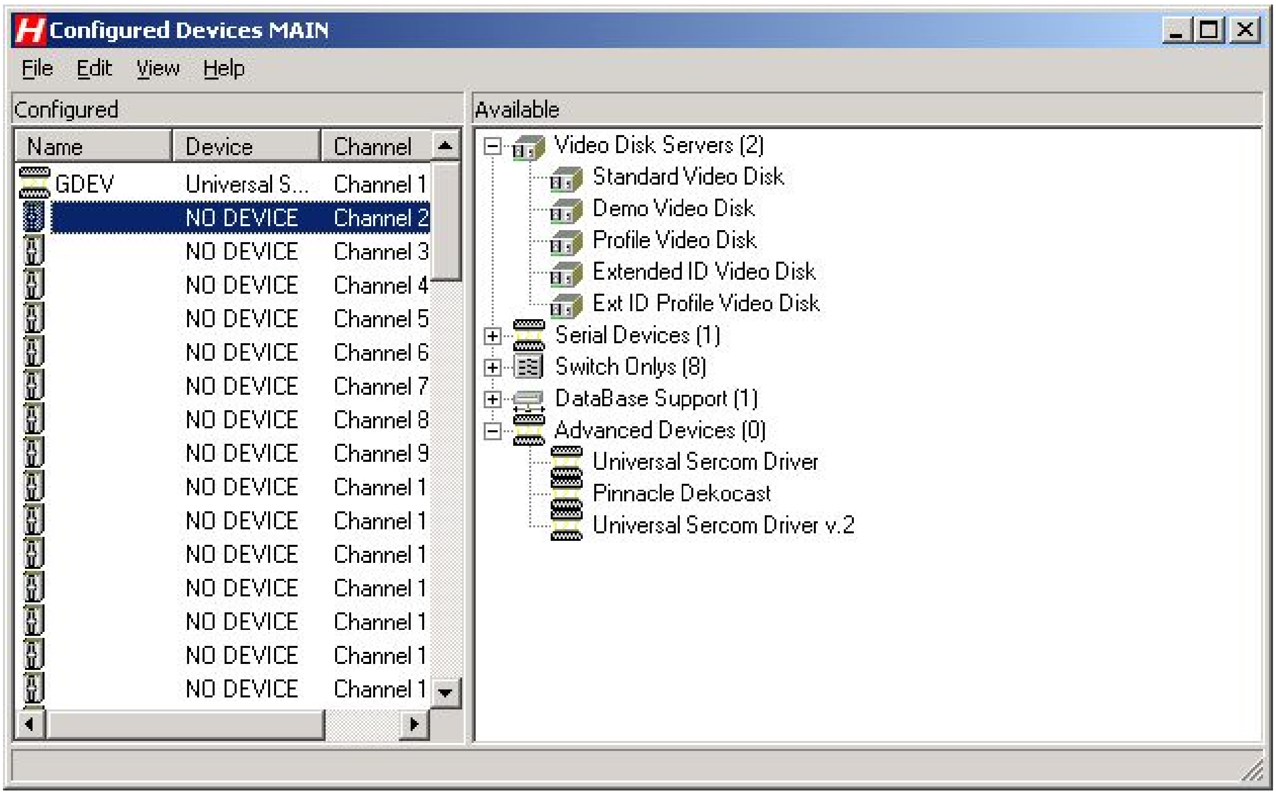
3. In the **Available** column on the left, under the **Advanced Devices** category, locate the **Universal Sercom Driver**. Click and drag the driver to one of the channels in the **Configured** column that reads **NO DEVICE**.



| Name | Device | Channel |
|---|---|---|
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 2 |
| | NO DEVICE | Channel 3 |
| | NO DEVICE | Channel 4 |
| | NO DEVICE | Channel 5 |
| | NO DEVICE | Channel 6 |
| | NO DEVICE | Channel 7 |
| | NO DEVICE | Channel 8 |
| | NO DEVICE | Channel 9 |
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 1 |

- Video Disk Servers (2)
  - Standard Video Disk
  - Demo Video Disk
  - Profile Video Disk
  - Extended ID Video Disk
  - Ext ID Profile Video Disk
- Serial Devices (1)
  - Serial Com Device
  - IBT VPS 9601
  - IBT VPS 9701 Device
  - IBT VPS 9702
  - IBT VPS 9703
- Switch Onlys (8)
  - Switch Only Device
- DataBase Support (1)
  - Database Update Redirector
- Advanced Devices (1)
  - Universal Sercom Driver
  - Pinnacle Dekocast

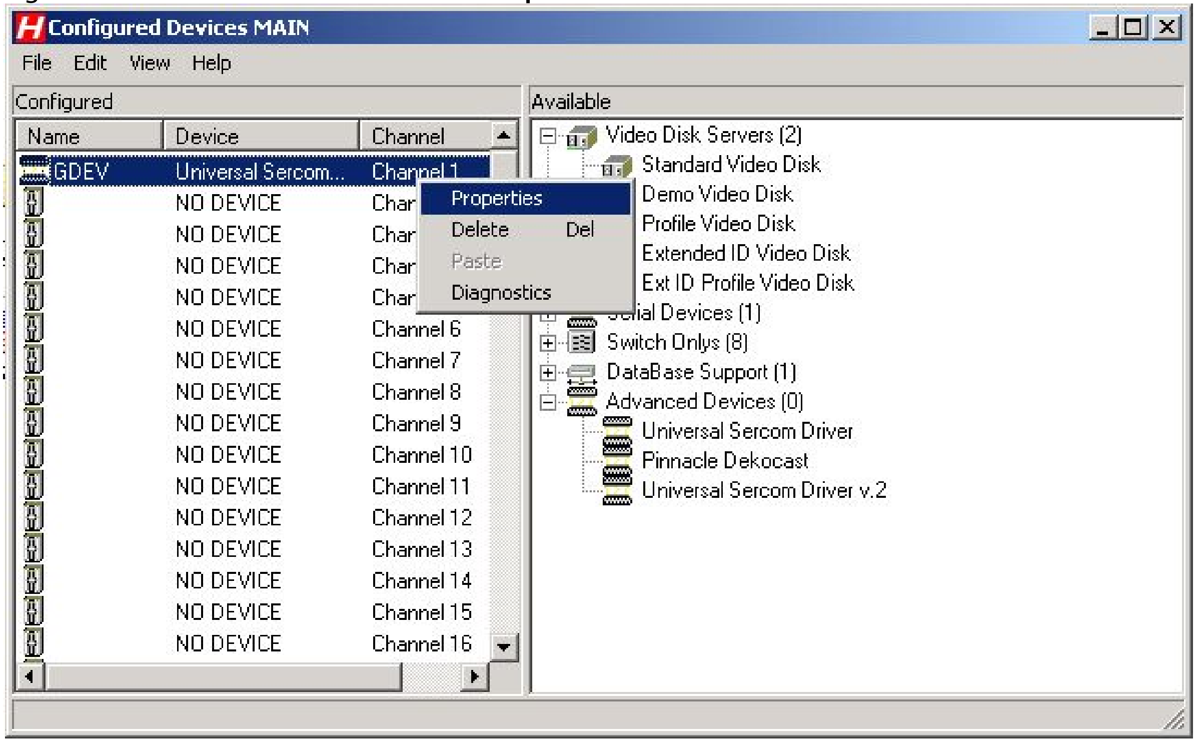
A new device will appear in the left pane of Configured Devices main window.



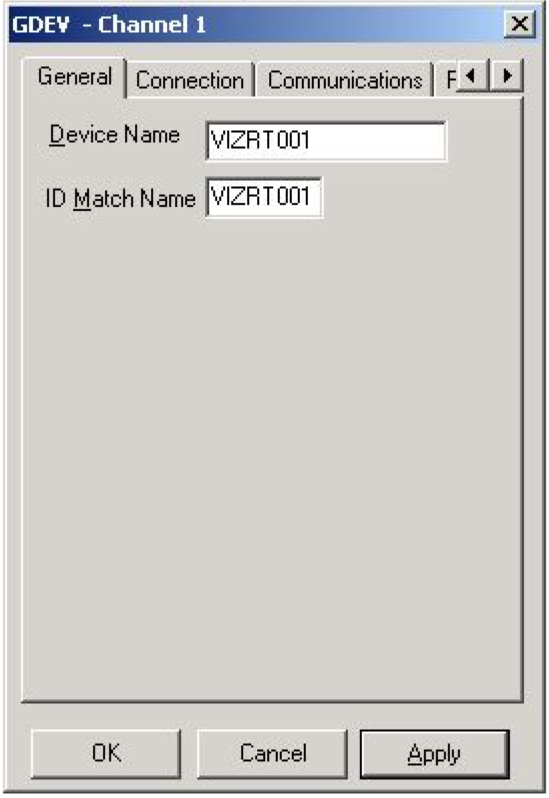
| Name | Device | Channel |
|---|---|---|
| GDEV | Universal S... | Channel 1 |
| | NO DEVICE | Channel 2 |
| | NO DEVICE | Channel 3 |
| | NO DEVICE | Channel 4 |
| | NO DEVICE | Channel 5 |
| | NO DEVICE | Channel 6 |
| | NO DEVICE | Channel 7 |
| | NO DEVICE | Channel 8 |
| | NO DEVICE | Channel 9 |
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 1 |
| | NO DEVICE | Channel 1 |

- Video Disk Servers (2)
  - Standard Video Disk
  - Demo Video Disk
  - Profile Video Disk
  - Extended ID Video Disk
  - Ext ID Profile Video Disk
- Serial Devices (1)
- Switch Onlys (8)
- DataBase Support (1)
- Advanced Devices (0)
  - Universal Sercom Driver
  - Pinnacle Dekocast
  - Universal Sercom Driver v.2

1. Right-click the new device and select **Properties.**



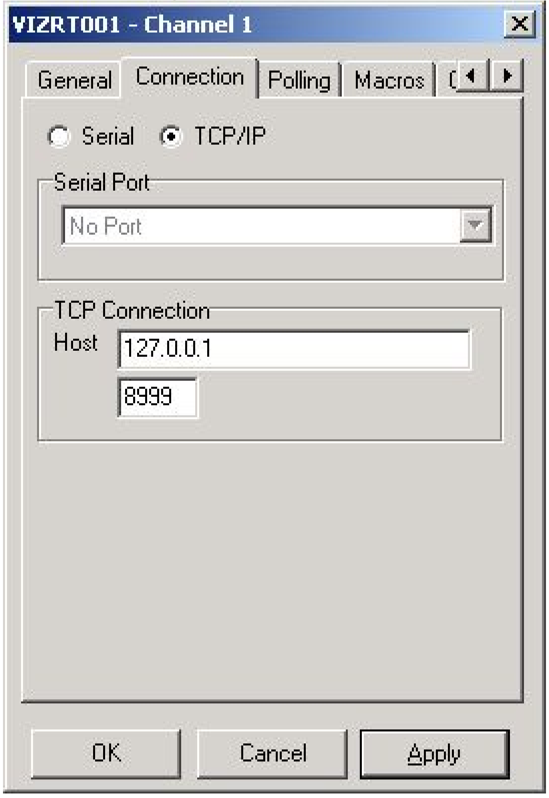
2. In* General* tab replace default name and ID with **VIZRT001** and click **Apply**.



Device Name: VIZRT001

ID Match Name: VIZRT001

3. In **Connection** tab, insert Media Sequencer machine's IP/Host or Serial value and the Port (for TCP) as configured in Viz Multichannel TCP or VDCP device settings.



VIZRT001 - Channel 1

○ Serial ● TCP/IP

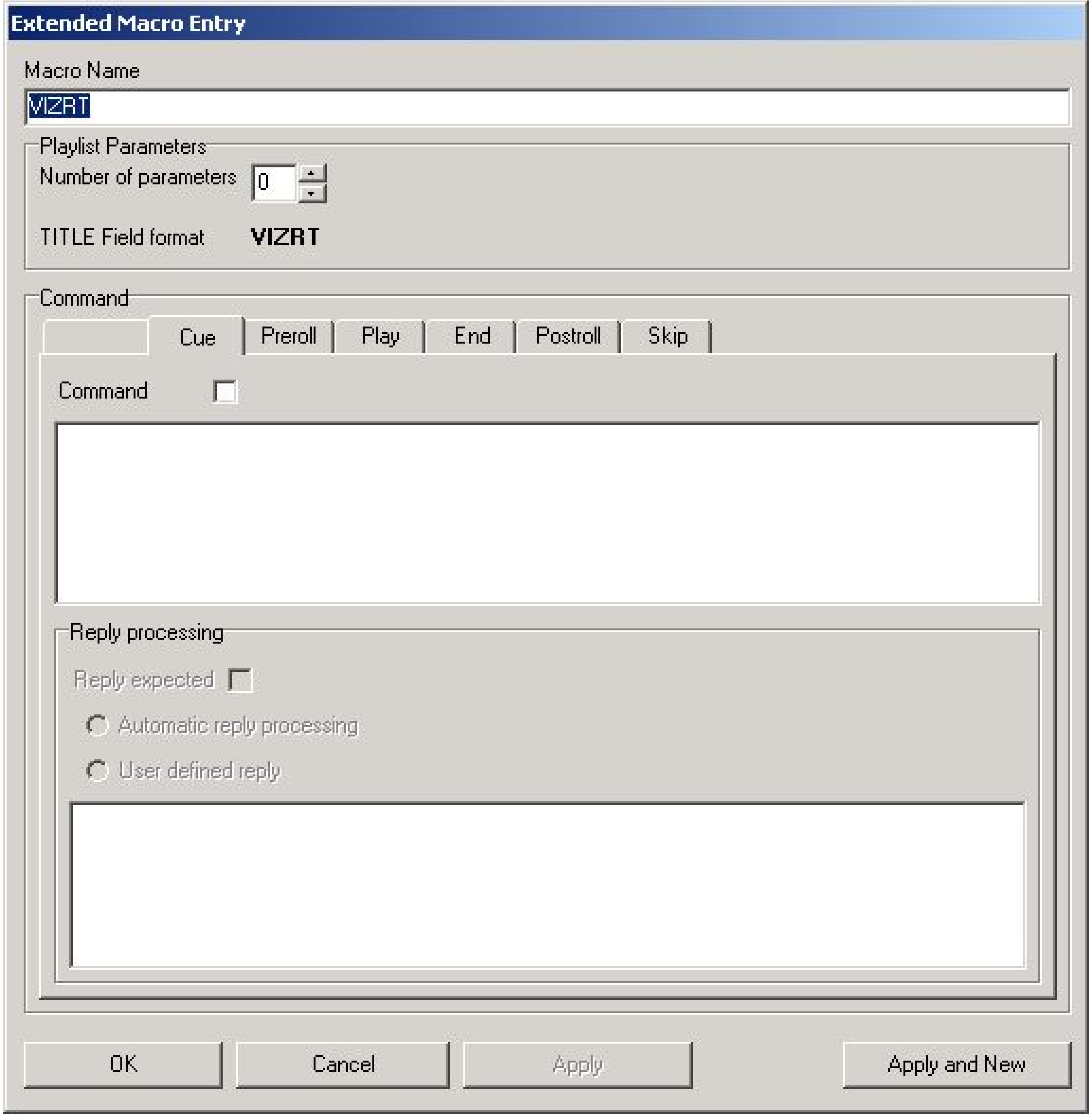
Serial Port: No Port

TCP Connection

Host: 127.0.0.1

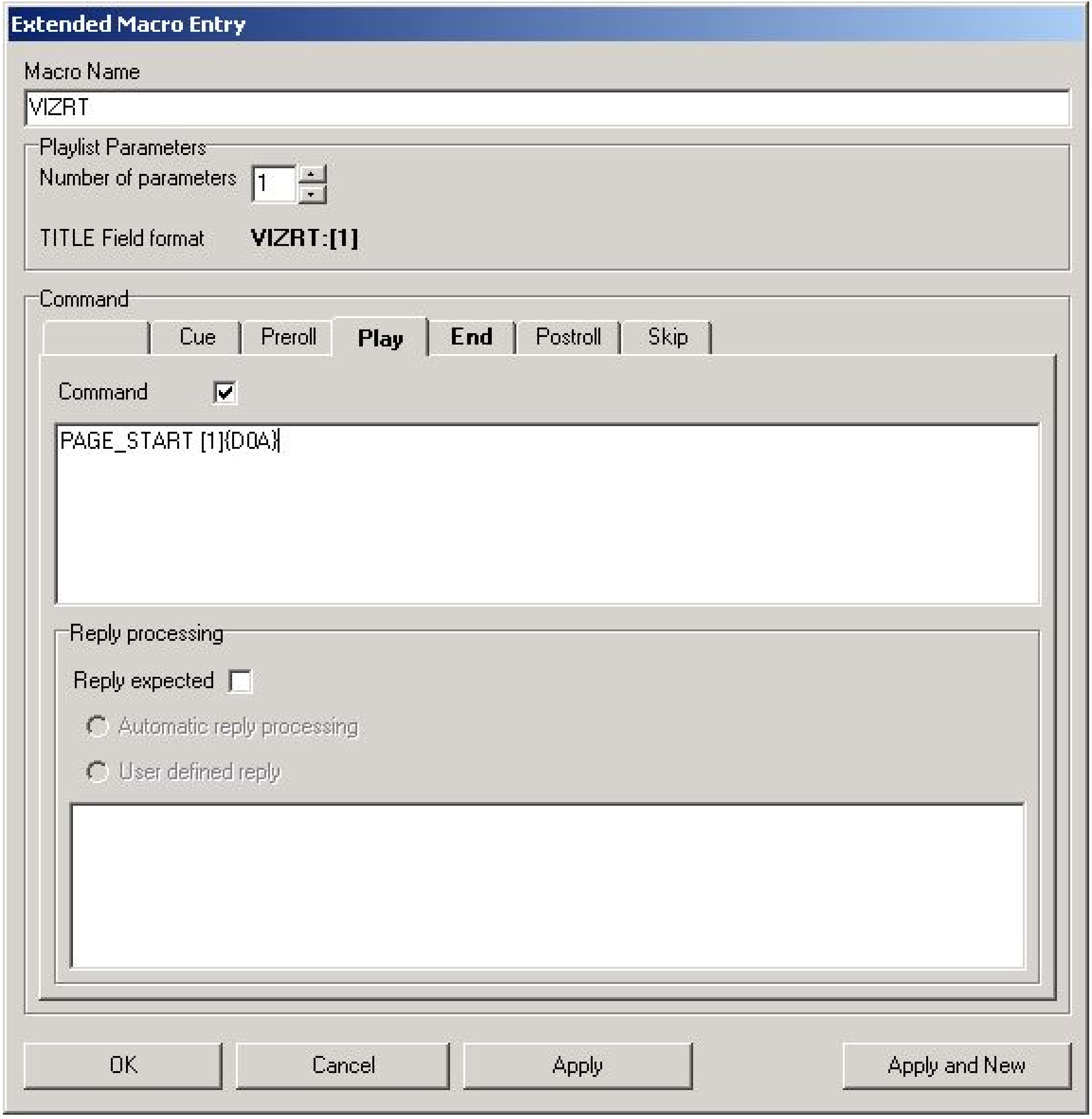
8999

4. In **Macros** tab click the **Add** button to create a new macro.
   The Extended Macro Entry dialog is displayed.



5. In the **Cue** tab set the* Macro Name* to **VIZRT**.
6. Set the **Number of Parameters** to **1**.
7. Check the **Command** check box.

8. In the **Command** field, enter the command to be used.



Extended Macro Entry

Macro Name: VIZRT

Playlist Parameters

Number of parameters: 1

TITLE Field format: **VIZRT:[1]**

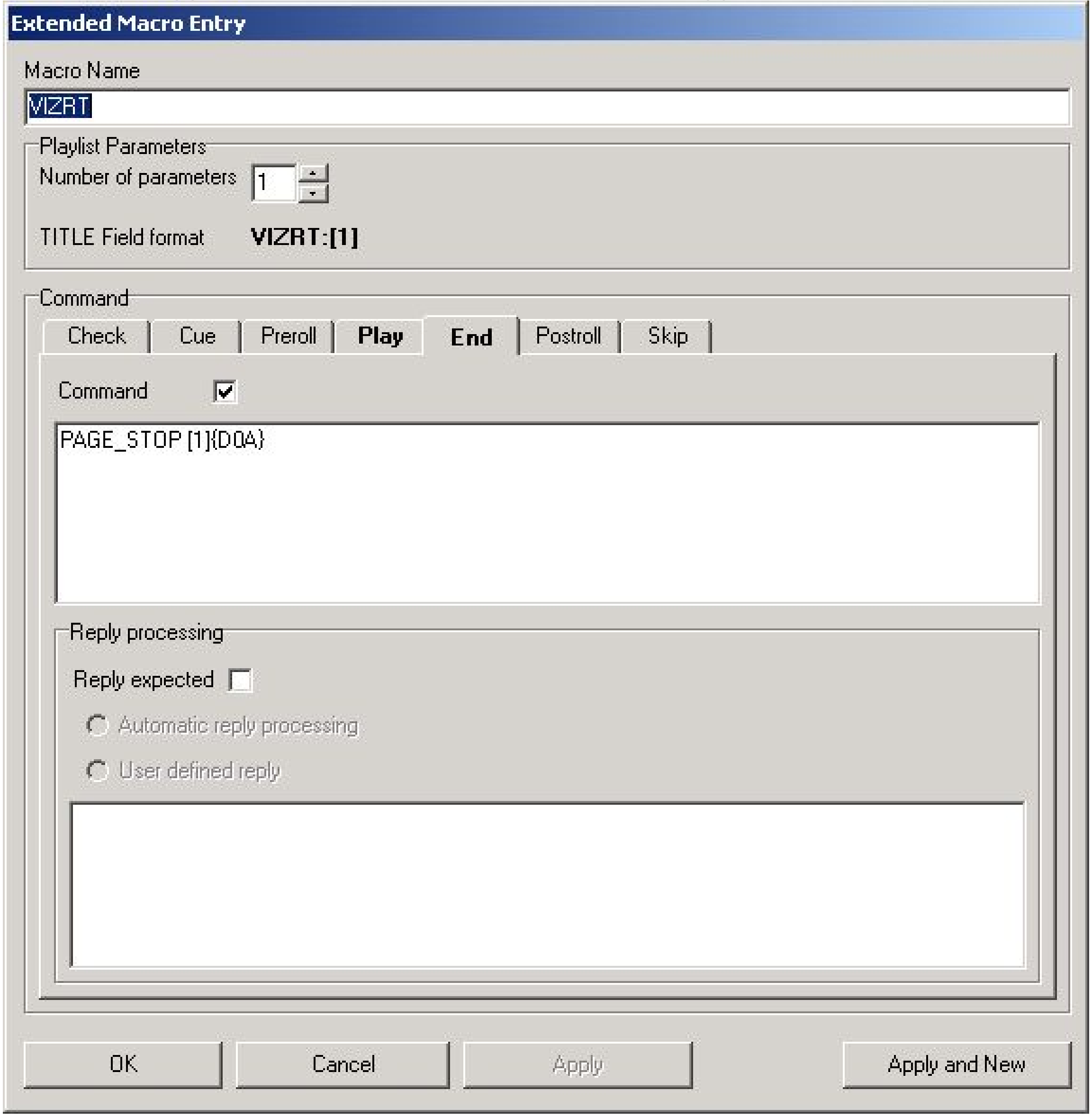
Command ☑

```
PAGE_START [1]{D0A}
```

Reply expected □

9. To edit the graphics stopping macro string, click the **End** tab.
10. Check the **Command** check box.

11. Set the **Number of Parameters** to **1**.
    Play will be triggered when an event starts. End will be triggered when an event ends.



Macro Name: VIZRT

Number of parameters: 1

TITLE Field format: VIZRT:[1]

Command ☑

```
PAGE_STOP [1]{D0A}
```

Reply expected □

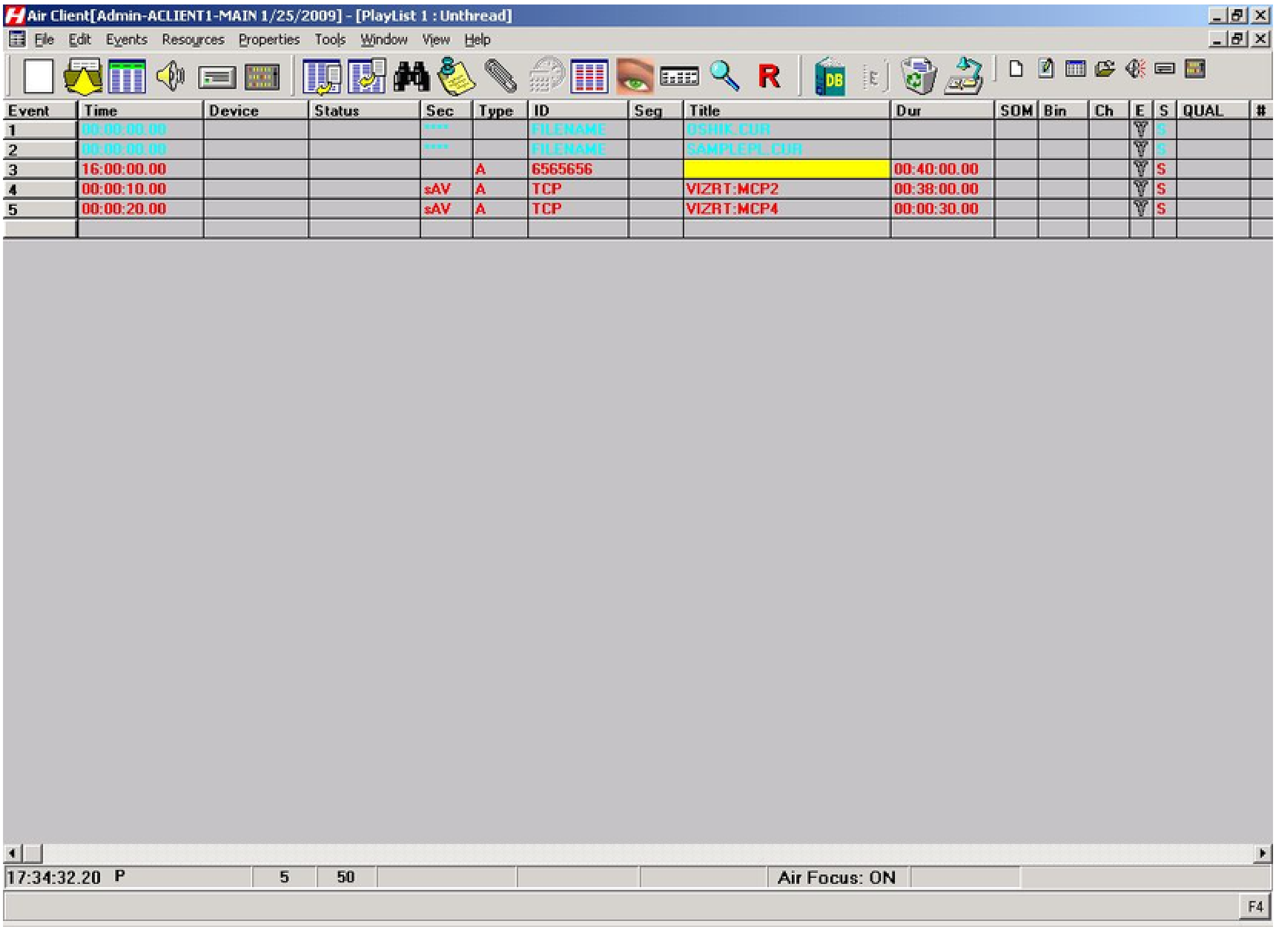
## Notes

When an event is triggered (primary or secondary), the command **take id** is sent where id is the `Media ID` value in Viz Multichannel 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.

## Operation

Viz uses A/V, Secondary A/V to execute the pre-loading and playout of media.


Air Client[Admin-ACLIENT1-MAIN 1/25/2009] - [PlayList 1 : Unthread]
File Edit Events Resources Properties Tools Window View Help
DB
R
F4

| Event | Time | Device | Status | Sec | Type | ID | Seg | Title | Dur | SOM | Bin | Ch | E | S | QUAL | # |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 00:00:00.00 | | | **** | | FILENAME | | OSHIK.CUR | | | | | Y | S | | |
| 2 | 00:00:00.00 | | | **** | | FILENAME | | SAMPLEPL.CUR | | | | | Y | S | | |
| 3 | 16:00:00.00 | | | | A | 6565656 | | | 00:40:00.00 | | | | Y | S | | |
| 4 | 00:00:10.00 | | | sAV | A | TCP | | VIZRT:MCP2 | 00:38:00.00 | | | | Y | S | | |
| 5 | 00:00:20.00 | | | sAV | A | TCP | | VIZRT:MCP4 | 00:00:30.00 | | | | Y | S | | |

17:34:32.20 P | 5 | 50 | Air Focus: ON

## 19.7.4 Automatic Playlist Updates Using RCV Files

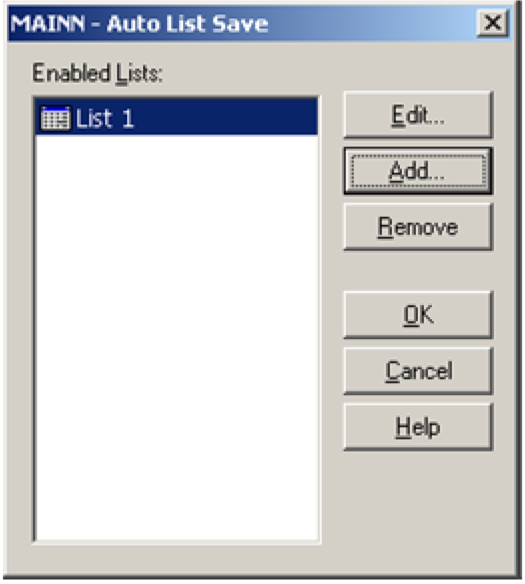
Auto List Save allows automatic list savings based on the number of list changes that occur on an active on-air transmission list.

Number of changes value indicates how often the list is automatically saved to an RCV file.

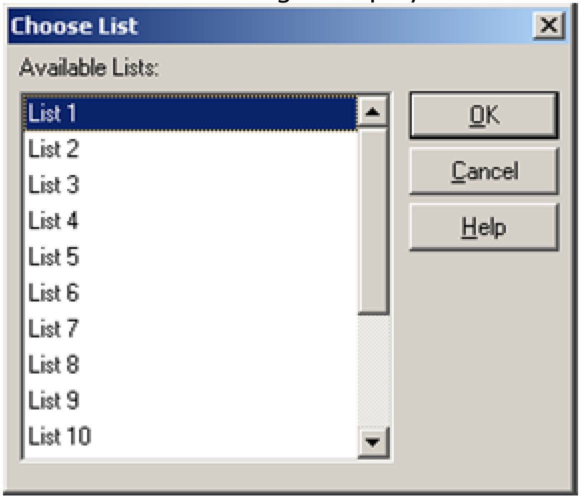
RCV file will be read by Multichannel and update its playlist accordingly.

## To Enable Auto List Save

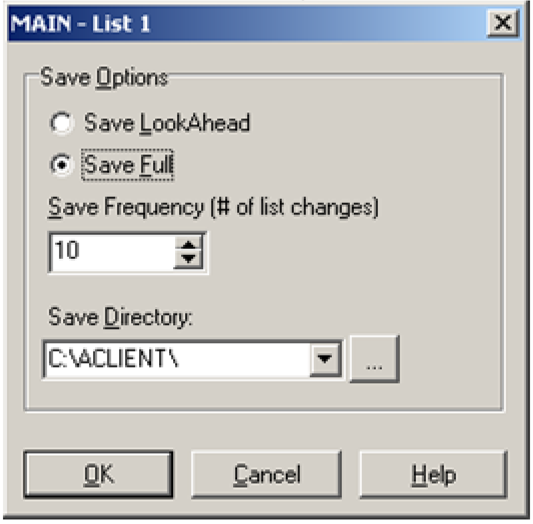
1. From the main menu select Properties > Auto List Save. The Auto List Save dialog is displayed.



2. Click the **Add** button.
   The Choose List dialog is displayed.



3. Select a list from the Available Lists pane and click **OK**.
   The Save Options dialog displayed.



MAIN - List 1

Save Options

○ Save LookAhead

● Save Full

Save Frequency (# of list changes)

10

Save Directory:

C:\ACLIENT\

4. **Save LookAhead**: (Default: enabled) Check to enable save of the list look ahead 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.) Viz Multichannel will scan the specified folder for new RCV file updates.
8. Click **OK**.
   The list is added to the Enabled Lists pane.

## 19.8 D–Series By Harris Broadcast

This section describes configuring the Playout Automation for Enterprise Environments system (D-Series) by Harris Broadcast to work with Viz Multichannel:

- General Information and Setup
- Updating Live
- Limitations

### 19.8.1 General Information and Setup

On the Viz Multichannel side it 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.

### 19.8.2 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). Viz Multichannel can scan that folder and update playlists accordingly.

### 19.8.3 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

## 19.9 Neptune By Pebble Beach Systems

The Neptune automation system by Pebble Beach Systems (PBS) can trigger Viz Multichannel (through Media Sequencer) as a TCP device (TCP/IP).

⚠ **Note:** Whenever possible, use the PB Neptune Integration Module. See Integration Module for Neptune from Pebble Beach.

The Neptune integrates with Viz Multichannel in the following ways:

1. Triggering primary events (via a TCP port to the Media Sequencer machine)
2. Triggering secondary events (via a TCP port or CII serial port to the Media Sequencer machine)
3. Automatically update playlist (via a TCP port to the Schedule Collector machine)
4. Hosting Viz Content Pilot’s Newsroom Component for scheduling actions in Neptune.

This section contains information on the following topics:

- Commands
- Importing and Updating
- Updating Live
- Newsroom Component
- Limitations

### 19.9.1 Commands

The following commands are supported in Media Sequencer:

- `PROGRAM_START id` will play primary event with `Media ID = id`
- `PROGRAM_STOP id` will stop primary event with `Media ID = id`
- `PROGRAM_LOAD id` will load primary event with `Media ID = id`
- `PAGE_START id` will play secondary event with `Media ID = id`
- `PAGE_STOP id` will stop secondary event with `Media ID = id`
- `PAGE_LOAD id` will load secondary event with `Media ID = id`

### 19.9.2 Importing and Updating

The Neptune system reads playlist data via files in XML format.

**Channel traffic name** (`channel`) and **playlist date** (`txdate`) are provided in the list node. The rest of the fields are read from the event nodes (eventlist/event). **Begin Time** should be read from `onairtime` node. Different time formats are supported.

An event’s on-air date can be read from `onairdate` node if available. Sometimes it is merged with `onairtime`. Secondary events (if available) can be read from inside their primary event node (`../secondaryeventlist/secondaryevent`).

### 19.9.3 Updating Live

Two major workflows are usually implemented:

**Reading playlists from a scheduling system**: Viz Multichannel (usually in scheduling mode) reads playlist (in any format) from the scheduling system, generates all secondary events either manually, automatically (by Rules), from the playlist file, or any combination of the three. Once a playlist is ready (usually sent to Viz Multichannel in Master Control Room), it is triggered by the Neptune automation system to play either primary or secondary events. This workflow may include live updates from Neptune via the Schedule Collector tool.

**Reading playlists directly from the Neptune automation system**: Viz Multichannel runs only in MCR mode and reads playlists directly from Neptune. That is done via the Schedule Collector tool. Schedule Collector uses the Pebble Beach Systems API, connects it every configured interval and gets X (configured) next events information from it. Schedule Collector then, whenever it recognizes any changes in the schedule, generates an XML update file and triggers the Playlist Importer. A Neptune schedule is always automatically mirrored in Viz Multichannel.

This workflow provides several different options:

- Reading primary and secondary events with content from Neptune.
- Reading only primary events from Neptune and generating secondary events by Rules.
- Reading primary and secondary events from Neptune but populating secondary events content by Viz Multichannel,
- Any combination of the three above.

> ⚠ **Note:** In order to enable that feature in Neptune, a Pebble Beach Systems’ XML API license is required.

### 19.9.4 Newsroom Component

Viz Content Pilot’s Newsroom Component provides the ability to view available templates for a channel, create new editable templates, edit, preview and schedule them in the playlist.

For more information on Viz Content Pilot’s Newsroom Component see the *Viz Content Pilot User’s Guide.*

> ⚠ **Note:** This integration connects to Media Sequencer for reading template information and not Viz Content Pilot’s database.

### 19.9.5 Limitations

The Pebble Beach Systems XML format is closed and strict. Viz Multichannel needs more information to add flexibility when scheduling (e.g. rules, conditions, auto-content); however, this is not possible.

# 20 Viz One Integration

This chapter gives an overview of the steps to take before you make use of your Viz One:

- Viz One and Viz Engine Configuration
- Viz Multichannel Configuration
- Video Preview Configuration
- Installation Options

## 20.1 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 meta data.
- 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.

## 20.2 Viz Multichannel Configuration

When integrating Viz Multichannel with Viz One you must:

- allow Viz Multichannel 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. Also see Import Options on how to create placeholders in Viz One for missing House Ids.
- allow Viz Multichannel to *search* for and *preview media assets* locally before adding it to a template/page and/or the playlist.
- allow Viz Multichannel to *configure one or multiple device channels* (i.e. Viz Engine) with a Viz One connection for asset transfers to the same device channel.
  For more information on how to configure Viz Multichannel's Viz One connection and playout devices, see the Viz One and Configuring Playout Devices sections.

## 20.3 Video Preview Configuration

If you are previewing proxy versions of video from Viz One using VCP’s Newsroom Component (i.e. Timeline Editor) or Viz Engine (i.e. Viz Engine installed on your control client machine) you will have to 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

## 20.4 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, so customers must avoid this situation by ensuring that the *LAV decoder* is selected using the Windows 7 filter tweaker tool. See how To 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 the marker is at.
Customers that need a high level of frame accuracy should consider using other solutions.

The video decoders from *MainConcept* and *LAV* operate in a different way, meaning that 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.

## 20.4.1 To Install Codecs for Local Preview

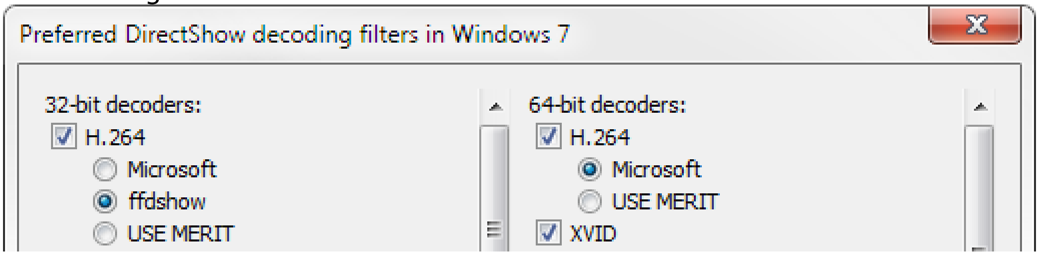
The example below sets up support for h.264 playback using the FFDShow MPEG-4 codec package and a Matroska Splitter from Haali.

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

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

## 20.4.2 To Set Preferred Decoder

1. **Run** the **Windows 7 DirectShow Filter Tweaker**
2. In the dialog box click **Preferred decoders**



Preferred DirectShow decoding filters in Windows 7

32-bit decoders:
☑ H.264
○ Microsoft
● ffdshow
○ USE MERIT

64-bit decoders:
☑ H.264
● Microsoft
○ USE MERIT
☑ XVID

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

**See Also**

- Working with Video Clips

# 21 Preview Server

Preview Server is used in situations where Viz Engine must provide frames for snapshot or thumbnail generation. Preview Server is used by several Vizrt products such as Viz Trio, Viz Multichannel and Viz Content Pilot Newsroom Component, to fetch previews of overlay graphics for the Timeline Editor.

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

This section contains information on the following topics:

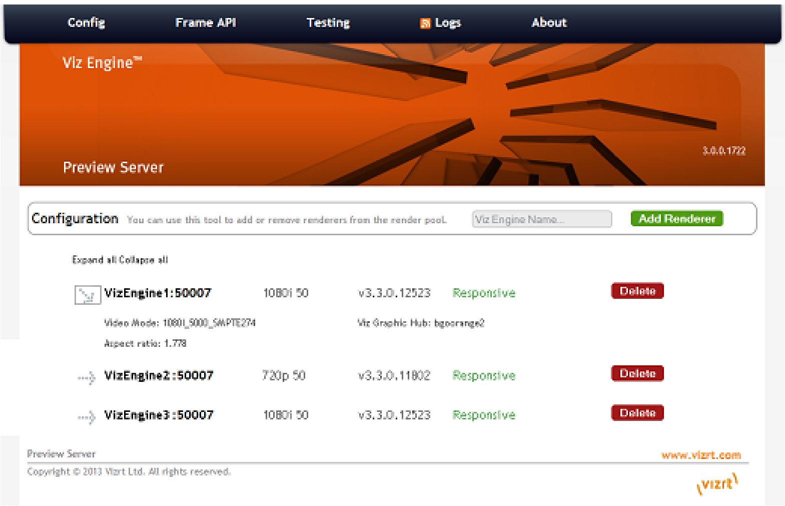
- Preview Server Web Interface
- VMC-Installation and Configuration

**See Also**

- *Viz Engine Administrator Guide* for information on installing Viz Engine.

## 21.1 Preview Server Web Interface

The Preview Server has a web interface for accessing the Preview Server's features.



Configuration You can use this tool to add or remove renderers from the render pool.

VizEngine1:50007 1080i 50 v3.3.0.12523 Responsive
Video Mode: 1080I_5000_SMPTE274 Viz Graphic Hub: bgoorange2
Aspect ratio: 1.778
VizEngine2:50007 720p 50 v3.3.0.11802 Responsive
VizEngine3:50007 1080i 50 v3.3.0.12523 Responsive

Copyright © 2013 Vizrt Ltd. All rights reserved.

### 21.1.1 To Access the Preview Server Web Interface

- Open the Preview Server Web Interface by navigating 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 the 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

## 21.1.2 Config

The 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/
```

### To Add 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, the Viz Engine is added with 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*).

### To Remove a Viz Engine

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

## 21.1.3 Frame API

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

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

## 21.1.4 Testing

This page provides:

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

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
```

### 21.1.5 Logs

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

```
http://<hostname>:54000/logs
```

### 21.1.6 About

Opens the About window with details of installed Preview Server version. A link to the list of Third Party Component Credits is also provided.

## 21.2 VMC-Installation And Configuration

This section describes

- Requirements
- Installation
- Configuration

of the Preview Server.

### 21.2.1 Requirements

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

### 21.2.2 Installation

The Preview Server is a separate installer and can be downloaded from ftp.vizrt.com in the _products/VizContentPilot/Latest Version _folder. Please carefully review the Release Notes before installing the product.

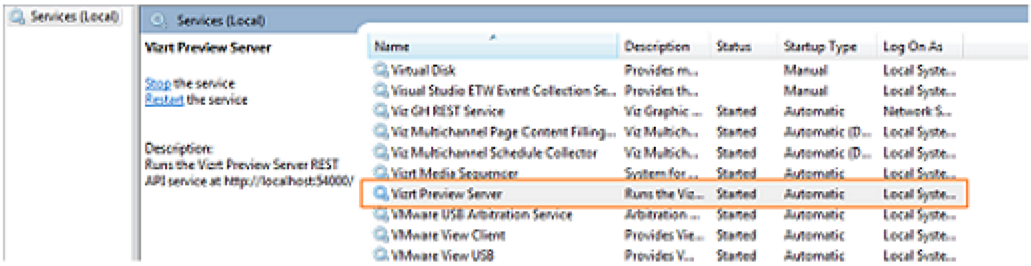
#### To Install or Upgrade the Preview Server

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.

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:



Vizrt Preview Server

Stop the service
Restart the service

Description:
Runs the Vizrt Preview Server REST API service at http://localhost:54000/

| Name | Description | Status | Startup Type | Log On As |
|---|---|---|---|---|
| Virtual Disk | Provides m... | | Manual | Local Syste... |
| Visual Studio ETW Event Collection Se... | Provides th... | | Manual | Local Syste... |
| Viz GH REST Service | Viz Graphic ... | Started | Automatic | Network S... |
| Viz Multichannel Page Content Filling... | Viz Multich... | Started | Automatic (D... | Local Syste... |
| Viz Multichannel Schedule Collector | Viz Multich... | Started | Automatic (D... | Local Syste... |
| Vizrt Media Sequencer | System for ... | Started | Automatic | Local Syste... |
| Vizrt Preview Server | Runs the Viz... | Started | Automatic | Local Syste... |
| VMware USB Arbitration Service | Arbitration ... | Started | Automatic | Local Syste... |
| VMware View Client | Provides Vie... | Started | Automatic | Local Syste... |
| VMware View USB | Provides V... | Started | Automatic | Local Syste... |

## 21.2.3 Configuration

### Preview Server Service

The 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

The Preview Server can run on the same machine as a Viz Engine (also used for the Newsroom Component data element preview), but 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.

# 22 Quick Scheduler Server

Quick Scheduler Server is a back-end component that hosts all schedules that is worked on via any Quick Scheduler client. This section presents:

## 22.1 Installation

1. Locate the current Quick Scheduler Server installer from your Vizrt representative or from the ftp server download.vizrt.com.
   The Quick Scheduler Server is a separate installer named *VizMultichannel_QuickSchedulerServer-VERSION.exe* where *VERSION* will be numbers indicating the release/version of the software. Double-click on the installer to start the installation.
2. Press **Next** and on the next page **Next** to install Quick Scheduler Server to the default installation directory.
3. Press the **Install** button and wait until the software is installed.
   You can now exit the installer by pressing **Finish**.

> ⚠ **Important!**
> The Quick Scheduler Server run as a Windows Service process. It is recommended that the Quick Scheduler Server service process is enabled to start automatically and that the Quick Scheduler Server is run as an Administrator service level process

## 22.2 Configuration

### 22.2.1 Quick Scheduler Server Configuration

To configure Quick Scheduler Server follow these steps:

1. Use Windows Explorer to locate the directory `%ProgramData%\Vizrt\Multichannel\QuickSchedulerServer` and locate the config file.
   Normally: `c:\ProgramData\Vizrt\Multichannel\QuickSchedulerServer\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
    c. Press **Enter** to start the program.

## 22.2.2 Channel Settings

After finishing the steps in Quick Scheduler Server Configuration and restarting the service process, there will be folders created for each channels in the directory `%ProgramData%\Vizrt\Multichannel\QuickSchedulerServer`.

To configure the channel settings please follow these steps:

1. Use Windows Explorer to locate the directory `%ProgramData%\Vizrt\Multichannel\QuickSchedulerServer\[channel_name]` and locate the settings file.
   Example: `c:\ProgramData\Vizrt\Multichannel\QuickSchedulerServer\MUSIC\settings.xml`
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"
    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 have configured (or changed) the `settings.xml` file you need to 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.

# 23 Appendix

This appendix provides information for the following topics:

- Upgrading Multichannel
- Vizrt Data Format (VDF)
- Page Content Filling Aliases
- Hosted Page Editor via ActiveX
- Installing and Upgrading the Microsoft .NET library
- Enabling Windows Crash Dumps
- REST Interface for Integrations Hub Version 2.0

## 23.1 Upgrading Multichannel

Vizrt recommends that customers contact their local Vizrt representative before upgrading. Tools are available to assist upgrade from previous versions of Viz Multichannel.

⚠ **IMPORTANT!** The Upgrade tool shipped with Multichannel will upgrade an existing installation from 2.6.2 or higher 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 Multichannel for each version, follow the instructions below.

After upgrading please check that the Timezone settings in Multichannel are correct. In particular, after upgrading from 2.6 to 2.7, users *must* set a Timezone in the configuration: **General Settings > Timezone Settings** for Channel timezone and Schedule timezone, see Timezone Settings.

- Upgrading Multichannel from 2.6.2+ to 4.x
- Working with UpgradeTool
- Upgrading Multichannel from 2.5 or 2.6 to 2.6.2
- Upgrade Tools from Multichannel Versions Lower than 2.5
- Troubleshooting the Upgrade Tool

### 23.1.1 Upgrading Multichannel from 2.6.2+ to 4.x

1. Uninstall previous version.

2. Install Viz Multichannel 4.x (which includes **UpgradeTool**).

3. Run **UpgradeTool** (see Working with UpgradeTool and upgrade appropriate channels.

4. Media Sequencer should be upgraded to 4.1 or newer.

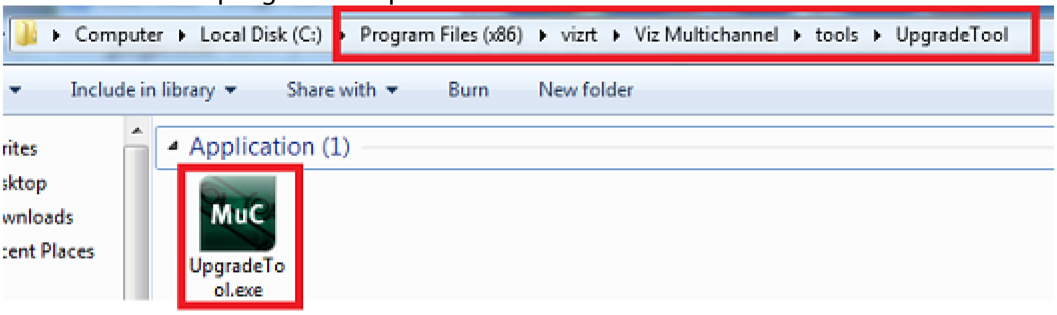
⚠ **Note:** Before upgrading Multichannel, turn *off* Media Sequencer handlers. After the upgrade is performed, (including upgrading Media Sequencer version where applicable), verify trigger settings and then enable the handlers again in **External Triggering** panel in Multichannel settings.

### 23.1.2 Working with UpgradeTool

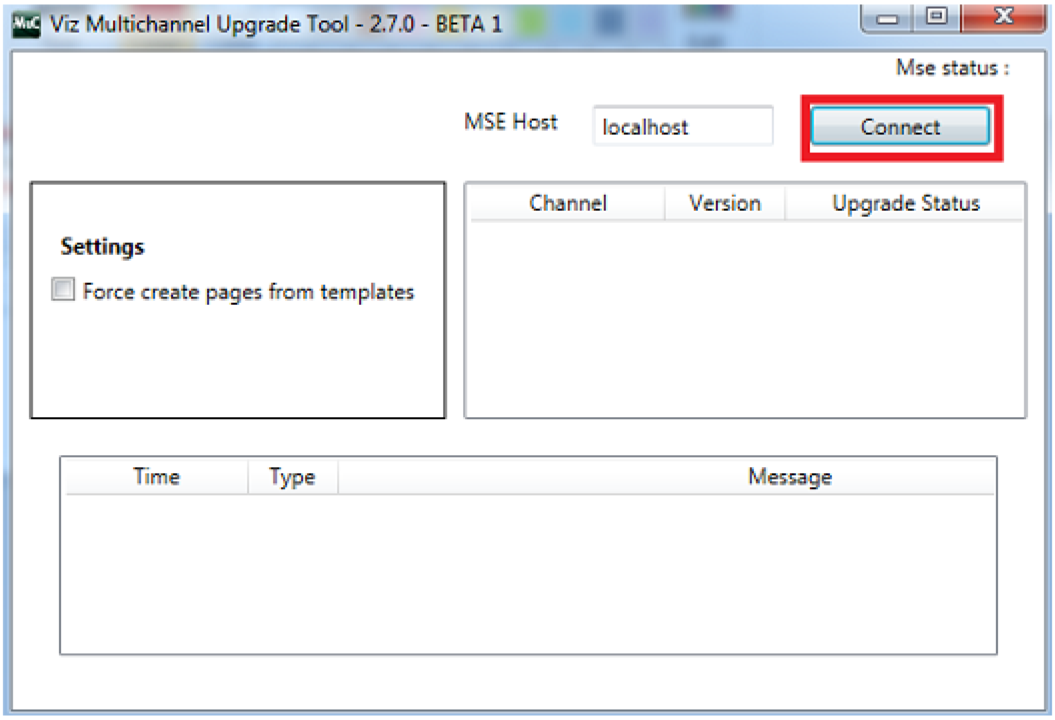
(included in 2.7 and later)

Multichannel 2.7+ provides an update tool which can be found at `[Viz Multichannel installation folder]\tools\UpgradeTool]`.

1. Double-click the program to open it.

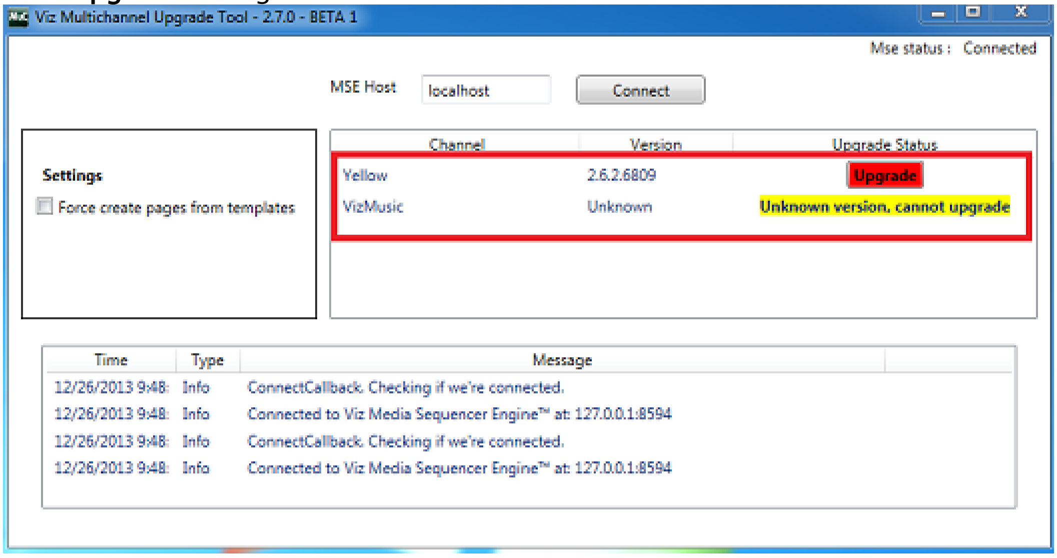


2. Click **Connect** to show a channel list



The channel lists which are upgradable will be shown with an **Upgrade** button.

3. Click **Upgrade** to begin



Mse status : Connected

MSE Host: localhost

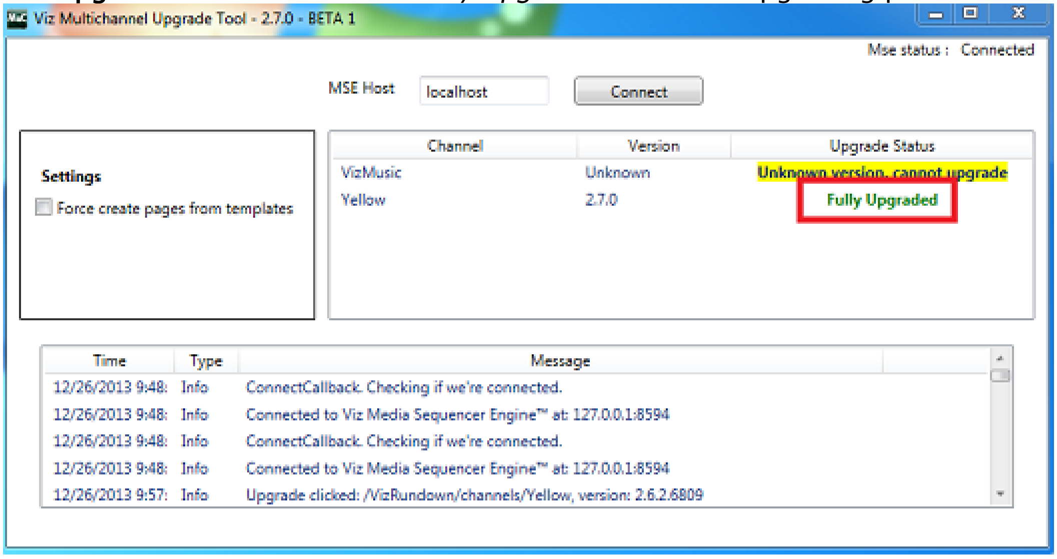
**Settings**

□ Force create pages from templates

| Channel | Version | Upgrade Status |
|---|---|---|
| Yellow | 2.6.2.6809 | Upgrade |
| VizMusic | Unknown | Unknown version, cannot upgrade |

| Time | Type | Message |
|---|---|---|
| 12/26/2013 9:48: | Info | ConnectCallback. Checking if we're connected. |
| 12/26/2013 9:48: | Info | Connected to Viz Media Sequencer Engine™ at: 127.0.0.1:8594 |
| 12/26/2013 9:48: | Info | ConnectCallback. Checking if we're connected. |
| 12/26/2013 9:48: | Info | Connected to Viz Media Sequencer Engine™ at: 127.0.0.1:8594 |

4. The **Upgrade Status** will show a *Fully Upgraded* when the upgrading process is completed.



Mse status : Connected

MSE Host: localhost

**Settings**

□ Force create pages from templates

| Channel | Version | Upgrade Status |
|---|---|---|
| VizMusic | Unknown | Unknown version, cannot upgrade |
| Yellow | 2.7.0 | Fully Upgraded |

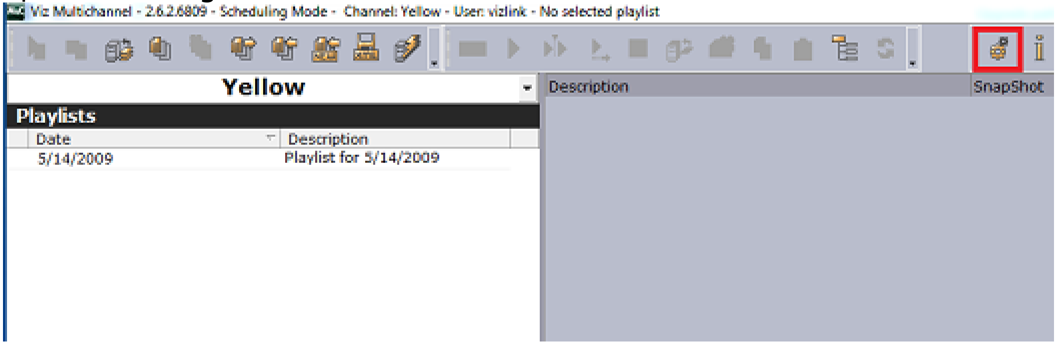
| Time | Type | Message |
|---|---|---|
| 12/26/2013 9:48: | Info | ConnectCallback. Checking if we're connected. |
| 12/26/2013 9:48: | Info | Connected to Viz Media Sequencer Engine™ at: 127.0.0.1:8594 |
| 12/26/2013 9:48: | Info | ConnectCallback. Checking if we're connected. |
| 12/26/2013 9:48: | Info | Connected to Viz Media Sequencer Engine™ at: 127.0.0.1:8594 |
| 12/26/2013 9:57: | Info | Upgrade clicked: /VizRundown/channels/Yellow, version: 2.6.2.6809 |

## 23.1.3 Upgrading Multichannel from 2.5 or 2.6 to 2.6.2

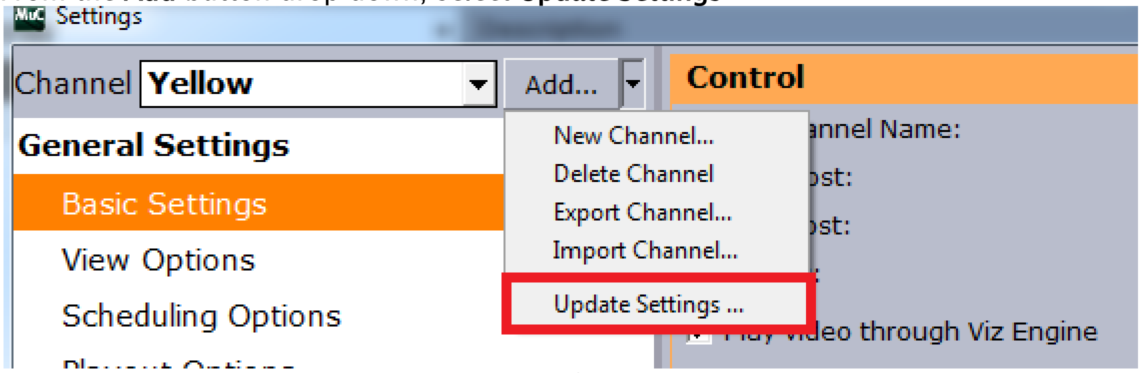
⚠ **Caution:** Always make sure you have a fully backed-up all relevant configuration settings before attempting an upgrade.

1. Double-click the application icon to open MuC 2.6.2

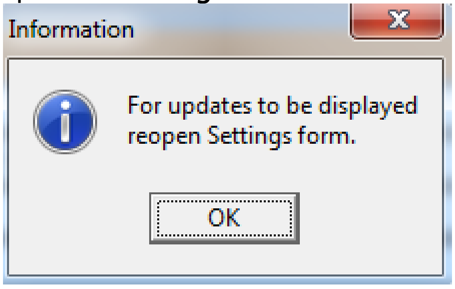
2. Open the **Settings** window from the Viz Multichannel toolbar


Viz Multichannel - 2.6.2.6809 - Scheduling Mode - Channel: Yellow - User: vizlink - No selected playlist
Yellow
Playlists
Date
Description
5/14/2009
Playlist for 5/14/2009
Description
SnapShot

3. From the **Add** button drop-down, select **Update Settings**



4. After the upgrade has been performed an **Information** dialogue box will be displayed. Re-open the **Settings** form to review the upgraded configuration.



For updates to be displayed reopen Settings form.

## 23.1.4 Upgrade Tools from Multichannel Versions Lower than 2.5

There are upgrade tools in the Multichannel 2.6 installation folder with names according to the Multichannel versions being upgraded.
The upgrade tools are designed to perform a complete upgrade after double-clicking on the tool.
The upgrade tools are:

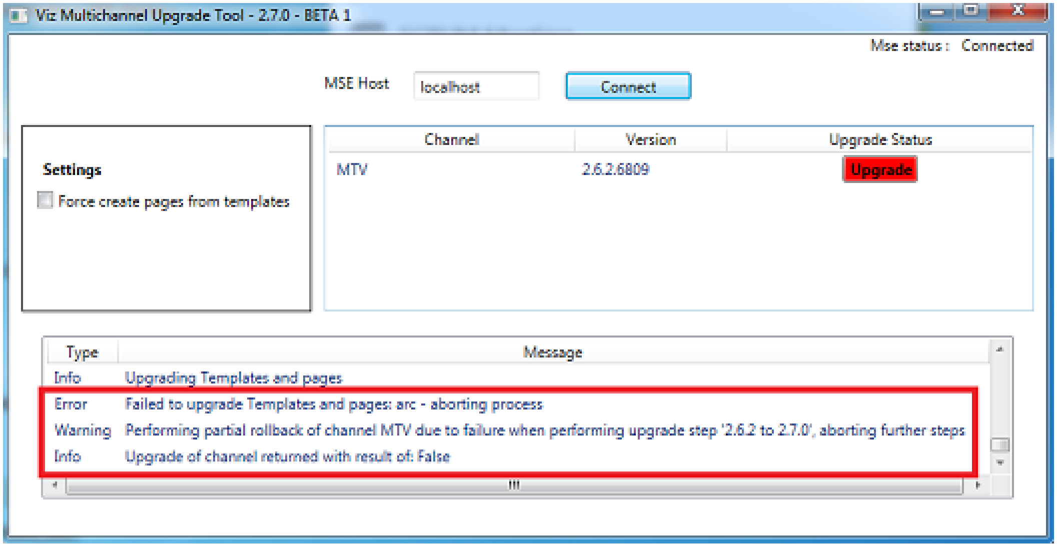
```
UpgradeTo20.exe is used for upgrading a default configuration of MuC to version 2.0.
UpgradeTo23.exe is used for upgrading a default configuration of MuC to version 2.3.
UpgradeTo25.exe is used for upgrading a default configuration of MuC to version 2.5.
```

## 23.1.5 Troubleshooting the Upgrade Tool

Normally the upgrade tool performs a complete upgrade automatically. In some cases a configuration or scene settings can prevent the upgrade tool from performing a complete upgrade. In case of failure, the upgrade tool either rolls-back the complete process (in case of an unrecoverable error) or provides warning messages (in case of non-critical errors).

### Critical Failures when Running the Upgrade tool

- The template and page information are missing


Viz Multichannel Upgrade Tool - 2.7.0 - BETA 1
Mse status : Connected
MSE Host
localhost
Connect
Settings
□ Force create pages from templates
Upgrade

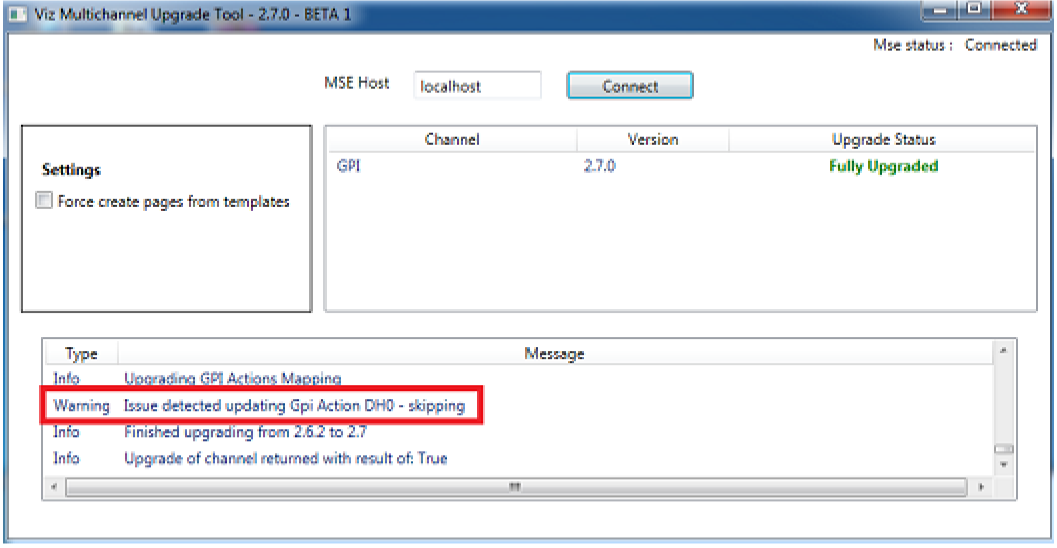
| Channel | Version | Upgrade Status |
|---|---|---|
| MTV | 2.6.2.6809 | Upgrade |

| Type | Message |
|---|---|
| Info | Upgrading Templates and pages |
| Error | Failed to upgrade Templates and pages: arc - aborting process |
| Warning | Performing partial rollback of channel MTV due to failure when performing upgrade step '2.6.2 to 2.7.0', aborting further steps |
| Info | Upgrade of channel returned with result of: False |

To fix this, re-import or delete the templates and pages indicating by the error message. Then run Upgrade Tool again by clicking the **Upgrade** button

### Non-Critical Warnings when Running the Upgrade tool

- The node value is null or empty


Viz Multichannel Upgrade Tool - 2.7.0 - BETA 1
Mse status : Connected
MSE Host
localhost
Connect
Settings
□ Force create pages from templates

| Channel | Version | Upgrade Status |
|---|---|---|
| GPI | 2.7.0 | Fully Upgraded |

| Type | Message |
|---|---|
| Info | Upgrading GPI Actions Mapping |
| Warning | Issue detected updating Gpi Action DH0 - skipping |
| Info | Finished upgrading from 2.6.2 to 2.7 |
| Info | Upgrade of channel returned with result of: True |

The “Node value is null or empty” is not a serious problem. The Upgrade tool is able to upgrade successfully while skipping all of this failure type. However, to prevent a further error, users should fill all the missing node values *after* the completion of an upgrade. Often this relates to GPI-settings. Contact Vizrt support if unsure.

## 23.2 Vizrt Data Format (VDF)

To implement an external update service, you need to parse and return a Vizrt Data Format (VDF) document to the Media Sequencer. The document is posted to the update service by the Media Sequencer.

Working with the data formats requires knowledge on how the data is structured when sent from the 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.

### 23.2.1 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.

## 23.3 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

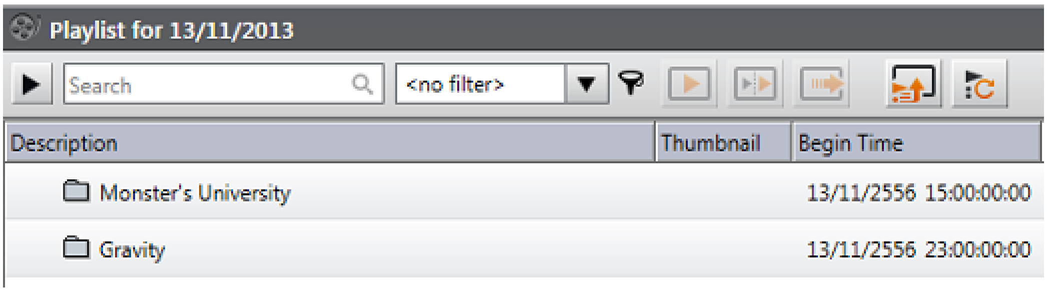
### 23.3.1 Page Content Filling Aliases

- **begin_date**: is the date property extracted from **begin_at** attribute, which refers to the *date* that the primary event is scheduled for play out.
- **begin_time**: is the time property extracted from **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 following states:

- **Next** It represents the next program from currently selected program.
- **Today** It represents the program within the same playlist, whose begin_time is less than 9 pm but doesn’t indicate the program which is next from currently selected program.
- **Tonight** It represents the program within the same playlist, whose begin_time is greater than 9 pm but doesn’t indicate the program which is next from currently selected program.
- **Tomorrow** It represents the program from next day's playlist, which is scheduled to be played tomorrow.

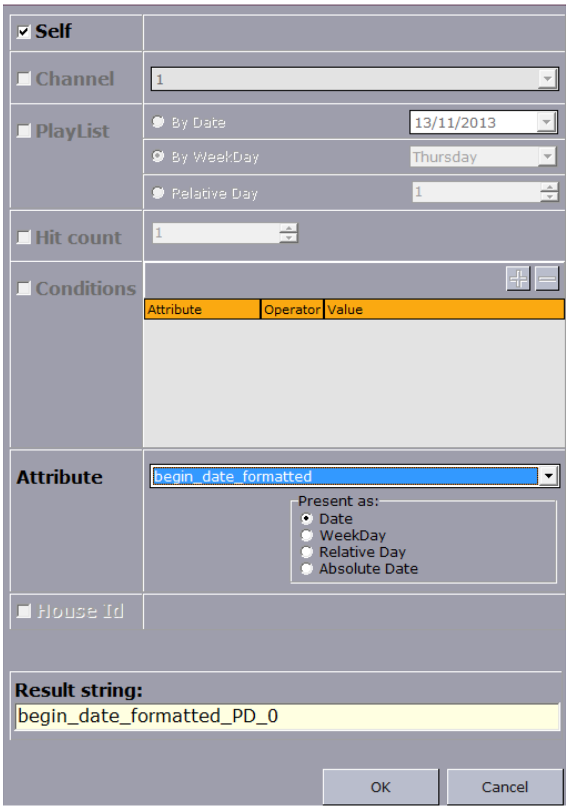
### 23.3.2 Page Content Filling Example

Consider an example playlist, in the image below Playlist 13/11/2013:



| Description | Thumbnail | Begin Time |
|---|---|---|
| Monster's University | | 13/11/2556 15:00:00:00 |
| Gravity | | 13/11/2556 23:00:00:00 |

For any page, check Self and set Attribute to begin_date_formatted:



☑ Self

□ Channel: 1

□ PlayList
- ○ By Date: 13/11/2013
- ● By WeekDay: Thursday
- ○ Relative Day: 1

□ Hit count: 1

□ Conditions

| Attribute | Operator | Value |
|---|---|---|

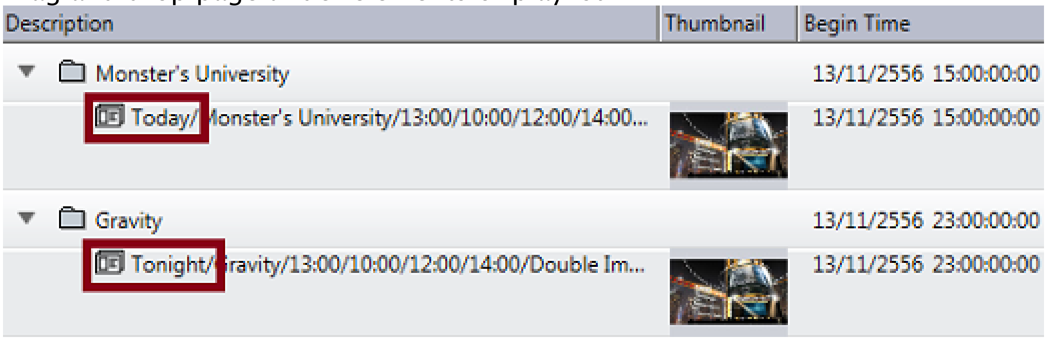
Attribute: begin_date_formatted

Present as:
- ● Date
- ○ WeekDay
- ○ Relative Day
- ○ Absolute Date

□ House Id

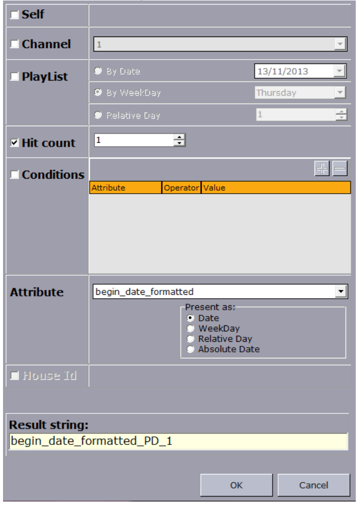
Result string: begin_date_formatted_PD_0

Drag and drop page under elements of playlist:



| Description | Thumbnail | Begin Time |
|---|---|---|
| Monster's University | | 13/11/2556 15:00:00:00 |
| Today/Monster's University/13:00/10:00/12:00/14:00... | | 13/11/2556 15:00:00:00 |
| Gravity | | 13/11/2556 23:00:00:00 |
| Tonight/Gravity/13:00/10:00/12:00/14:00/Double Im... | | 13/11/2556 23:00:00:00 |

For any page, set **Item** -> **Relative Item** -> **After (1)** and *Attribute* to **begin_date_formatted**:



□ Self

□ Channel: 1

□ PlayList
- ○ By Date: 13/11/2013
- ● By WeekDay: Thursday
- ○ Relative Day: 1

☑ Hit count: 1

□ Conditions

| Attribute | Operator | Value |
|---|---|---|

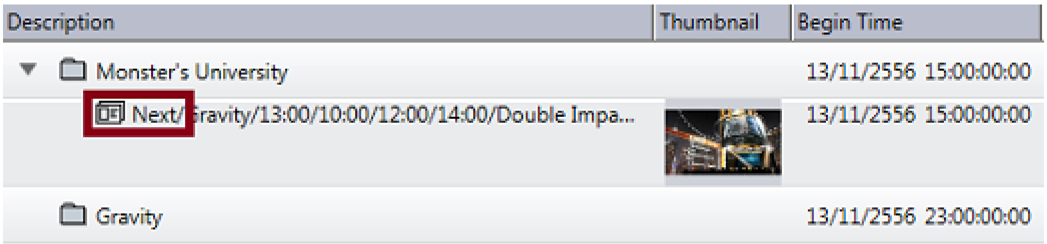
Attribute: begin_date_formatted

Present as:
- ● Date
- ○ WeekDay
- ○ Relative Day
- ○ Absolute Date

□ House Id

Result string:
begin_date_formatted_PD_1

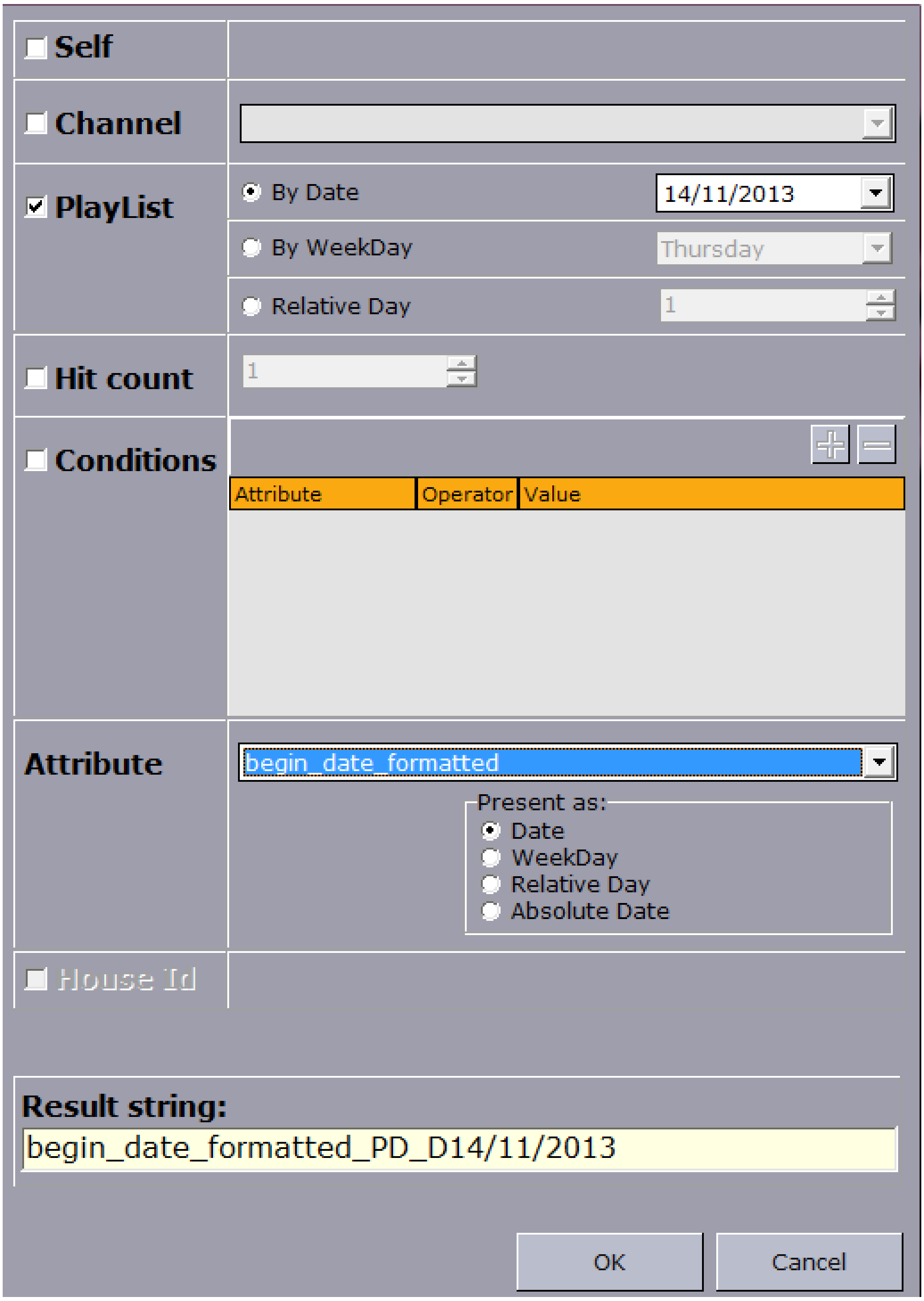
Drag and drop page under elements of playlist:



| Description | Thumbnail | Begin Time |
|---|---|---|
| Monster's University | | 13/11/2556 15:00:00:00 |
| Next/Gravity/13:00/10:00/12:00/14:00/Double Impa... | | 13/11/2556 15:00:00:00 |
| Gravity | | 13/11/2556 23:00:00:00 |

Next, create another playlist, say *Playlist 14/11/2013* with some elements.

For any page in previous playlist *13/11/2013*, set Playlist to the newly created playlist and Attribute to **begin_date_formatted**:



□ Self

□ Channel

☑ PlayList

● By Date 14/11/2013

○ By WeekDay Thursday

○ Relative Day 1

□ Hit count 1

□ Conditions

| Attribute | Operator | Value |
|---|---|---|

Attribute: begin_date_formatted

Present as:

● Date

○ WeekDay

○ Relative Day

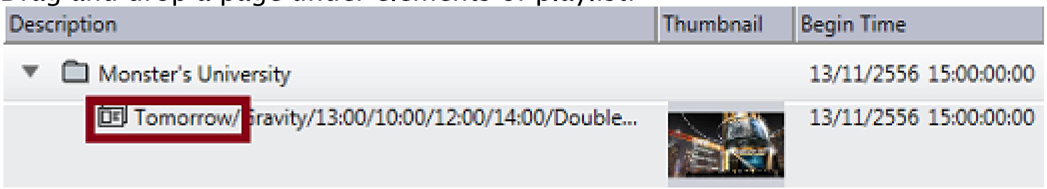
○ Absolute Date

□ House Id

Result string:

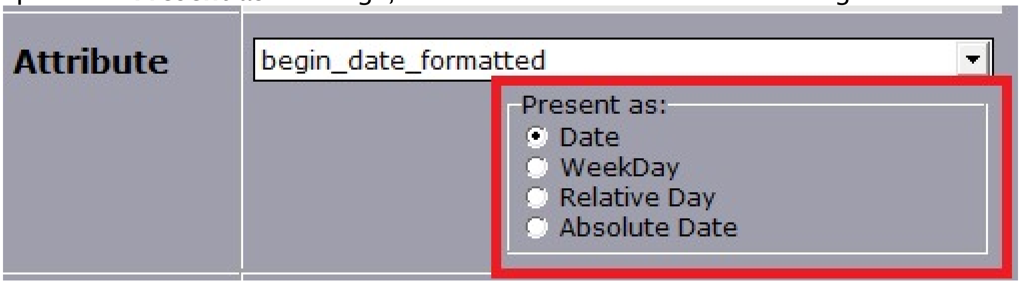
begin_date_formatted_PD_D14/11/2013

Drag and drop a page under elements of playlist:



| Description | Thumbnail | Begin Time |
|---|---|---|
| Monster's University | | 13/11/2556 15:00:00:00 |
| Tomorrow/Gravity/13:00/10:00/12:00/14:00/Double... | | 13/11/2556 15:00:00:00 |

For the states of the program which is scheduled for the days *later* than *tomorrow*, it depends upon the "**Present as**" settings, which is discussed below the image:



**Attribute** begin_date_formatted

Present as:

● Date
○ WeekDay
○ Relative Day
○ Absolute Date

- **Date**: Date is the default setting for attribute *begin_date_formatted*. It uses *Next*, *Today*, *Tonight* and *Tomorrow* states to define date. To represent program which is scheduled for the days later, it symbolizes the date of corresponding playlist as: *At dd/mm/yyyy*
- **WeekDay**: It uses *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: *At Sunday*
- **Relative Day**: It uses *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 as: *In **n*** days
- **Absolute Date**: It uses *dd/mm/yyyy* to represent date. And to represent a program which is scheduled for the days later it uses: At *dd/mm/yyyy*

## 23.4 Hosted Page Editor Via ActiveX

Included with the Viz Multichannel package is the Hosted Page Editor, delivered as an ActiveX component. This component can be hosted in a custom web-page or 3rd party application.

**Caution:** As of version 4.0 the **Hosted Page Editor** via ActiveX is *DEPRECATED*. This means that *no* more features will be added to it and it may be removed from future releases.

- Register the ActiveX
- Open the ActiveX
- Product and Channel Settings
- Templates and Pages
- The Page Editor
- ActiveX API Error Messages

**See Also**

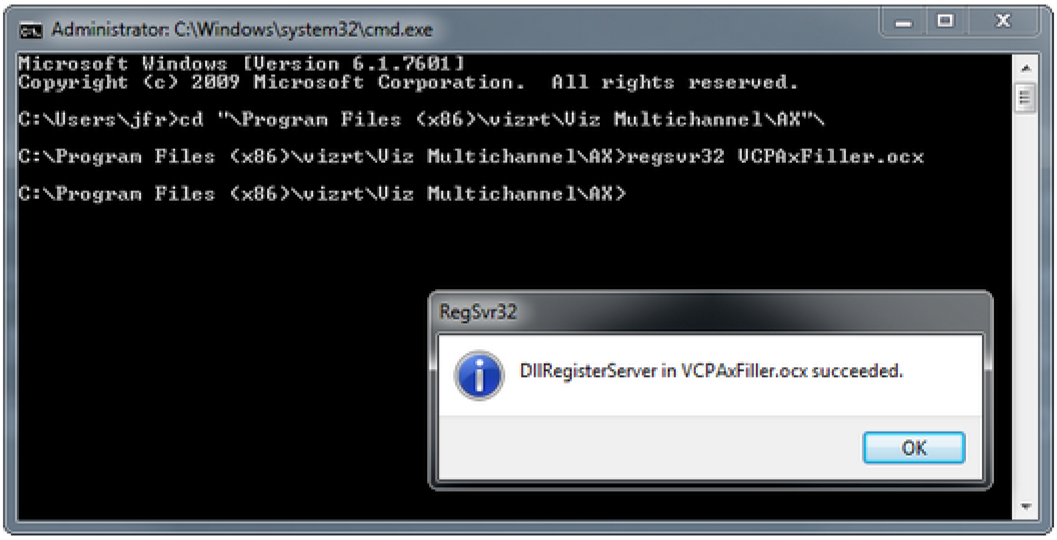
- Page Editor

### 23.4.1 Register the ActiveX

In the Viz Multichannel installation folder you will find a folder named AX containing the actual ActiveX component, its settings file and an HTML page for testing.

As the ActiveX component is not installed and registered during regular installation you need to register and configure it manually in order to create and test your own Page Editor.

### To Register the ActiveX Component



Administrator: C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\jfr>cd "\Program Files (x86)\vizrt\Viz Multichannel\AX"\

C:\Program Files (x86)\vizrt\Viz Multichannel\AX>regsvr32 VCPAxFiller.ocx

C:\Program Files (x86)\vizrt\Viz Multichannel\AX>
```

RegSvr32

DllRegisterServer in VCPAxFiller.ocx succeeded.

1. Run Command
2. Change directory to Viz Multichannel’s AX folder (e.g. C:\Program Files (x86)\Vizrt\Viz Multichannel\Viz Multichannel ActiveX)
3. Enter `regsvr32 VCPAxFiller.ocx` and press Enter

- This will register the ActiveX component.

### To Register ActiveX Component Settings

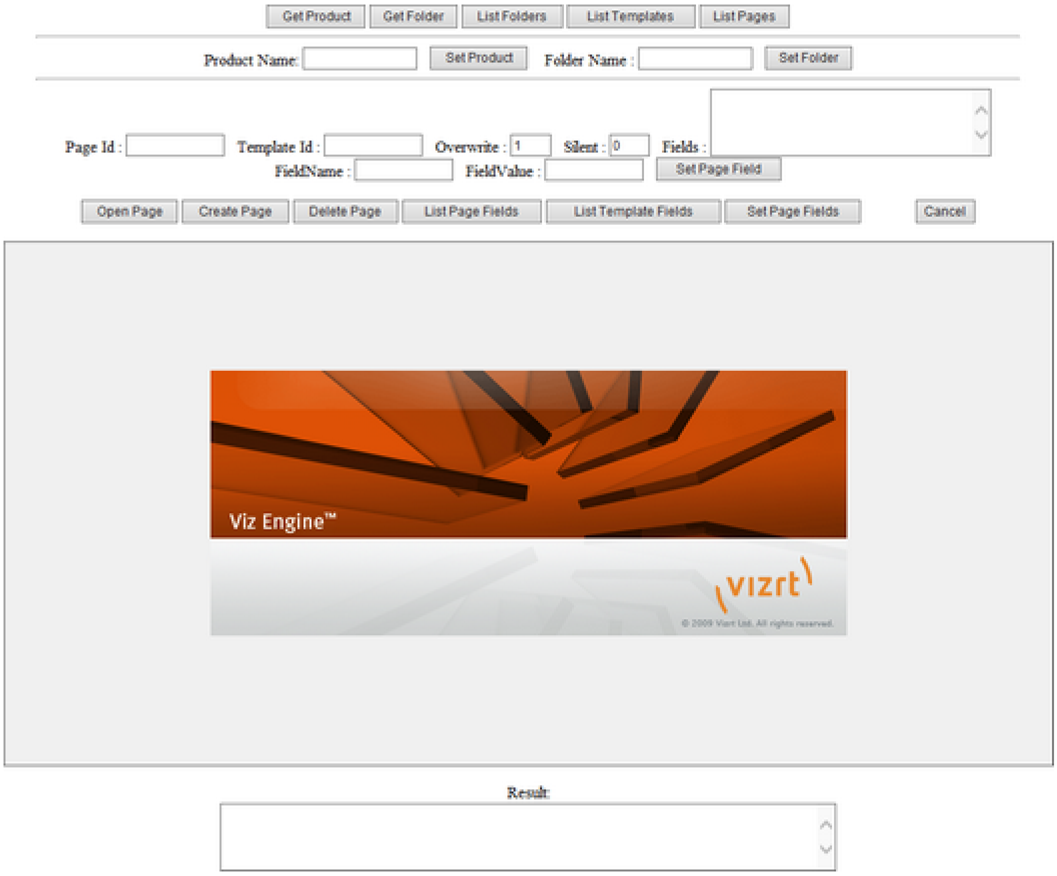
1. Open the **AX** folder in the Viz Multichannel installation directory.
   - If Viz Engine and Media Sequencer are not running on your local host you must edit the registry file and set the correct host names.
2. Double-click the **AX-HKEY_LOCAL_MACHINE.reg** file.
   - This will set **PreviewPort** to `50008`, **PreviewHost** to `localhost`, and **MseHost** to `localhost`, and in registry.

## 23.4.2 Open the ActiveX

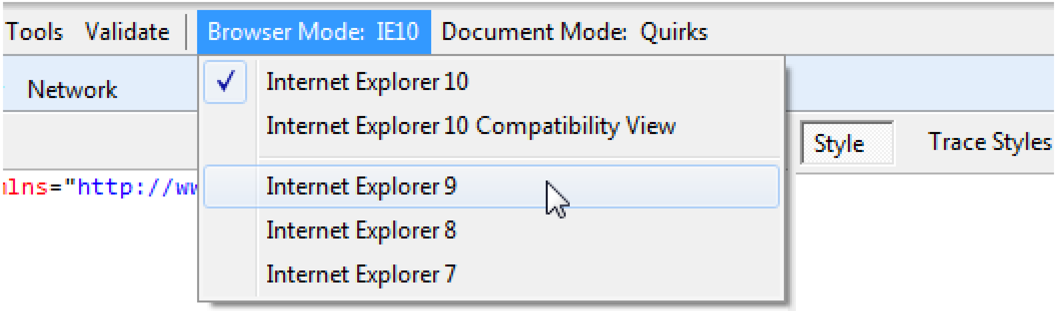
Before you can open the ActiveX component and its features you need To Register the ActiveX Component. Once that is done you can continue testing the ActiveX.

⚠ **Note:** The ActiveX component must run in **browser mode**: Internet Explorer 9.

To open the ActiveX component



1. Open the **AX** folder in the Viz Multichannel installation directory.
2. Open the **VizrtActiveXTestPage.html** file in **Internet Explorer**.



3. If Internet Explorer 10 is used, press **F12** and set browser mode to **Internet Explorer 9**.
4. **Allow** Internet Explorer to run *blocked content* (i.e. ActiveX).

## 23.4.3 Product and Channel Settings

Before you can use the ActiveX to create, edit and delete pages based on already available templates and pages you need to define the product and channel the you will work with.

### To Set the Product

- Click the **Get Product** button.
- This will display the configured product in the **Result** area (at the bottom of the page).
- If `trio` is shown, enter `multichannel` in the **Product Name** field and click the **Set Product** button.

### To Set a Channel

1. Click the **Get Folder** button, and the **Result** area will show the **channel name currently in use**.
2. If nothing is shown in the **Result** area then press the **List Folders** button, and the Result area will show a list of **channel names** stored on the Media Sequencer.
3. Enter the **channel name** in the **Folder Name** field and click the **Set Folder** button.

   > ⚠ **Note:** Channel name is case sensitive.

4. Click the **Get Folder** button again.

## 23.4.4 Templates and Pages

### To List Templates and Template Fields

1. Click the **List Templates** button, and the **Result** area will show templates currently available.

```
<result>
<item>0400</item>
<item>0030</item>
<item>0200</item>
...
</result>
```

2. Enter the item value (e.g. 0400) in the **Template ID** field and click the **List Template Fields** button, and the **Result** area will show the templates' exposed fields.

```
<result>
<item name="001" description="text" type="richtext">Arial-Regular</item>
</result>
```

### To List Available Pages and Page Fields

1. Click the **List Pages** button, and the **Result** area will show pages currently available.

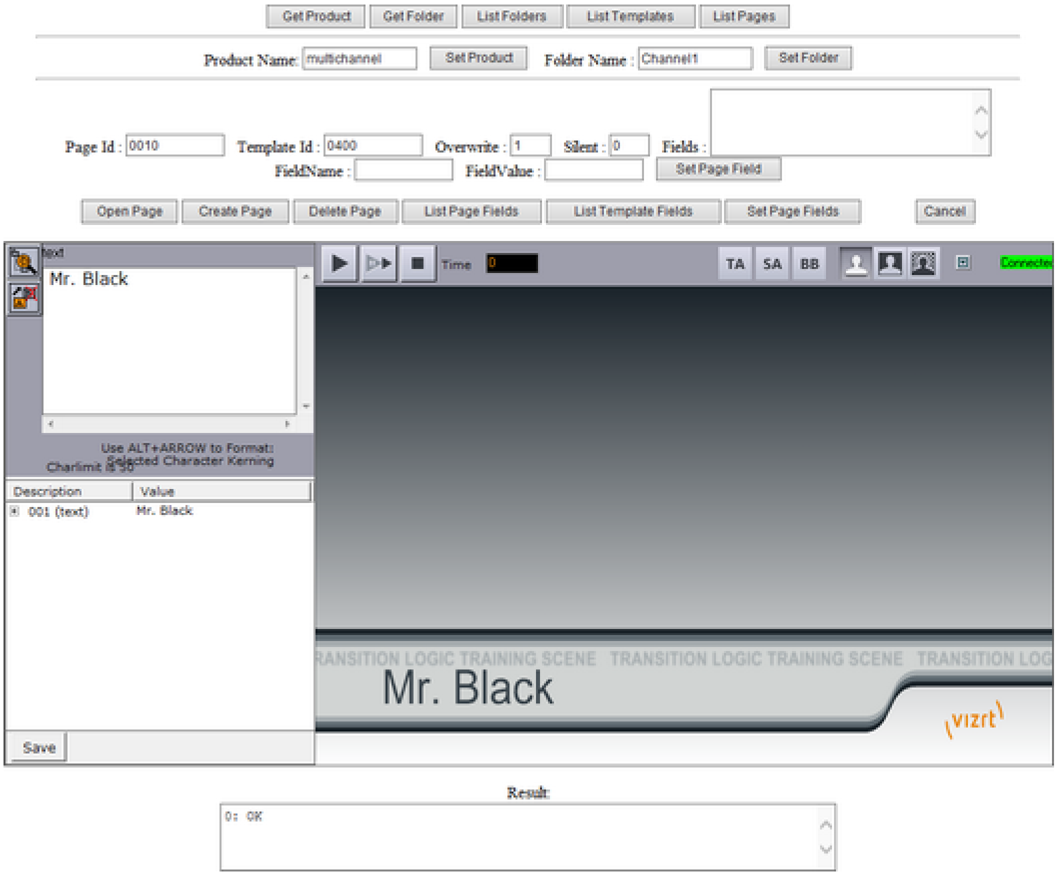
```
<result>
<item>0011</item>
<item>0012</item>
...
</result>
```

2. Enter the item value (e.g. 0011) in the **Page ID** field and click the **List Page Fields** button, and the **Result** area will show the pages' filled content.

```
<result>
  <item name="001" description="text" type="richtext">Mr. Black</item>
</result>
```

## 23.4.5 The Page Editor

### To Create a Page



Product Name: multichannel

Folder Name : Channel1

Page Id : 0010 Template Id : 0400 Overwrite : 1 Silent : 0

text

Mr. Black

Use ALT+ARROW to Format: Selected Character Kerning

Charlimit is 50

| Description | Value |
|---|---|
| 001 (text) | Mr. Black |

Result:

0: OK

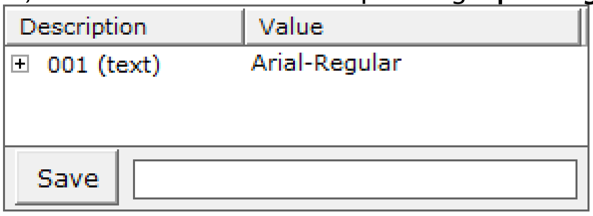
1. Enter a new **item** (e.g. 0010) in the **Page Id** field (see how To List Available Pages and Page Fields).

> ⚠ **Note**: Alternatively, enter a new Page Id when saving the new page.

2. Enter an existing **item value** in the **Template Id** field (see how To List Templates and Template Fields).

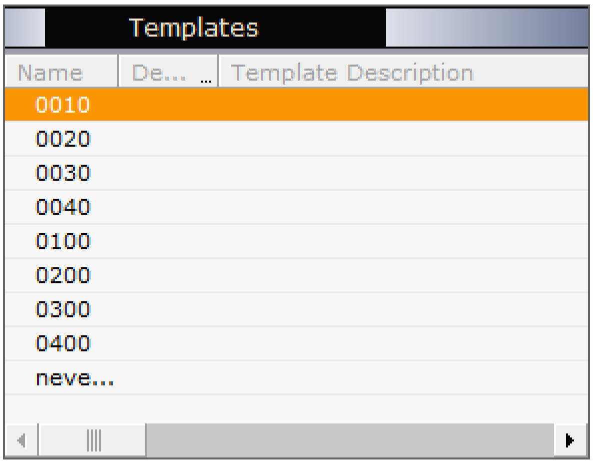
> ⚠ **Note**: Alternatively, select the Template Id from the Page Editor’s template list.

3. Click the **Create Page** button.
4. Edit the page and click the embedded page editor’s **Save** button (see Page Editor).
   Similar to creating a page you may also **edit** or **delete** a page by providing an existing Page Id, and then click the corresponding **Open Page** or **Delete Page** button.



| Description | Value |
|---|---|
| ⊞ 001 (text) | Arial-Regular |

If you do not enter a new Page Id before creating a new page you may enter it before you save the page.



Templates

| Name | De... | Template Description |
|---|---|---|
| 0010 | | |
| 0020 | | |
| 0030 | | |
| 0040 | | |
| 0100 | | |
| 0200 | | |
| 0300 | | |
| 0400 | | |
| neve... | | |

If you do not enter a Template Id before creating a new page the Page Editor will list all available templates. This also requires that no Page Id has been set before clicking the Create Page button.

## To Edit a Single Page Field Without a Page Editor

1. Enter an existing **item** (e.g. 0010) in the **Page Id** field (see how To list available pages and page fields).
2. Click the **List Page Fields** button, and the **Result** area will show the pages’ existing content.
3. Enter the **item name** in the **Field Name** field.
4. Enter the new item value in the **Field Value** field.
5. Click the **Set Page Field** button.
6. Click the **List Page Fields** button, and the **Result** area will show the pages’ new content.

## To Edit Multiple Page Fields Without a Page Editor

1. Enter XML to the **Fields** area in the following format:

```
<fields>
<item name="[field name]">[new value]</item>
<item name="[field name]">[new value]</item>
<item name="[field name]">[new value]</item>
...
</fields>
```

2. Click the **Set Page Fields** button.

### 23.4.6 ActiveX API Error Messages

- In case of errors in the ActiveX (AX) API, the following list shows the possible messages and error codes that a client can receive:

```
0: OK
1: No Change Has Occurred During The function Call
2: Unknown Product
3: Unknown Folder
4: Page Id Not Supplied
5: Template Id Not Supplied
6: Page Id Does Not Exists
7: Page Id Already Exists
8: Template Id Does Not Exists
9: Error Opening Page, Ex Info: No templates found for page pagename
9: Error Opening Page, Ex Info: Read canceled because scene does not exist: scenename
9: Error Opening Page, Ex Info: Page 222 refers to /VizRundown/channels/1/sceneinfos/
02_GFX\VizMultiChannel\MOVIE\06fullframe which was not found.
10: Error Opening Template, Ex Info: No templates found for page pagename
10: Error Opening Template, Ex Info: Page 06fullframe refers to /VizRundown/channels/
1/sceneinfos/02_GFX\VizMultiChannel\MOVIE\06fullframe which was not found.
11: Error Saving Page, Ex Info: A page id cannot contain the character '\q.
12: Invalid Fields XML, Ex Info: invalid fields XML
12: Invalid Fields XML, Ex Info: Invalid at the top level of the document.
13: Field Does Not Exists, Ex Info: field 'fieldname' does not exists
14: List Templates Error
15: List Folders Error
16: Context not Fully Set. Check Folder
17: List Pages Error
900: Last API Call Was Not Completed
1000: Media Sequencer Engine Is Disconnected
1100: Not Connected to Viz Engine
Page already exists, can't overwrite
Unable to save. please enter a page id
Unable to open page
```

- Following error messages are present but not displayed in client:

```
11: Error Saving Page
Unable to save. Page not set
Error Opening Page, Ex Info: can't open a template for editing
Error Opening Template, Ex Info: can't open a template for editing
```

## 23.5 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?.*
```

which will list out all the versions of .NET installed, latest first.

## 23.6 Enabling Windows Crash Dumps

### 23.6.1 MS Windows Logging

It is recommended to enable Microsoft Windows to generate user-mode crash dumps.
This makes debugging easier, particular if there are hardware or general windows issues affecting program behavior.

#### Enabling User-mode Dumps requires Windows 7 or higher.

- 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.

⚠ **Logging Features in Viz Multichannel**

There are valuable log files for understanding and analyzing unexpected Viz Multichannel program behavior. Refer to section Logging for more details about the logging options in Viz Multichannel.

## 23.7 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.

The following procedures describe how to configure Viz Multichannel 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

### 23.7.1 To Set Playlist Imports and Update Destination Path for Schedule Files

1. Start Viz Multichannel 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 (e.g. \network\...) are supported; however, mapped drives are not.

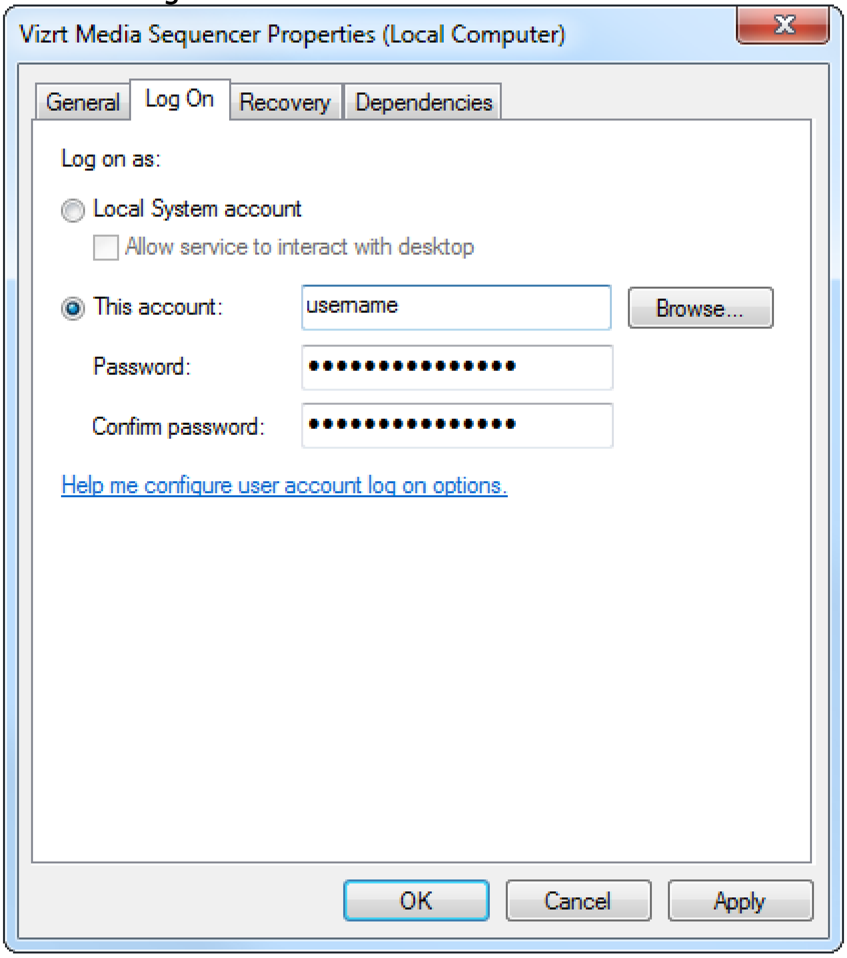
### 23.7.2 To Enable Automatic Import

1. Click **Global Settings > Import/Export settings > Automatic Import**
2. Set
   - **Automatic import interval**
   **- Playlist importer folder path**, and
   **- Playlist importer log level**

### 23.7.3 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**

6. Select the **Log On** tab



Vizrt Media Sequencer Properties (Local Computer)

Log on as:

○ Local System account

□ Allow service to interact with desktop

● This account: username

Password: •••••••••••••••

Confirm password: •••••••••••••••

Help me configure user account log on options.

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 the Vizrt Media Sequencer service is already running, please *restart* the service for the new settings to take effect.

## 23.7.4 Troubleshooting

### Microsoft Excel schedule files cannot be imported

> 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`

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*.

## 23.8 REST Interface For Integrations Hub Version 2.0

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

> This (version 2.0) API package is included for your convenience and offers more functionality than version 1.0. Please note that as this product is not an official Vizrt delivery, use of these (version 2.0) APIs is *entirely at your own risk*, and is *not* covered by any Vizrt Support agreement.

Currently, only the GET statement is listed in the documentation. Other REST verbs such as POST, PUT and DELETE is supported but not documented in this version. 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 Channel (Version 2.0)
- GET Channel State (Version 2.0)
- GET Channel Settings (Version 2.0)
- GET Channel Page Pool (Version 2.0)
- GET Channel Playlists (Version 2.0)
- GET Playlist in Channel (Version 2.0)
- GET Playlist State (Version 2.0)
- GET Primary Events in Playlist (Version 2.0)
- GET Primary Event (Version 2.0)
- GET Secondary Events under Primary Event (Version 2.0)
- GET Secondary Event (Version 2.0)

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

### 23.8.1 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.

### Syntax

```
GET /
```

### Request

```
GET / HTTP/1.1
```

### Response

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml
<atom:service xmlns:vaext="http://www.vizrt.com/atom-ext" xmlns:vizid="http://
www.vizrt.com/opensearch/vizid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub" xmlns="http://www.w3.org/2007/
app" xmlns:atom="http://www.w3.org/2005/Atom">
   <atom:workspace>
      <atom:title>Data Api</atom:title>
      <atom:collection href="http://127.0.0.1:9991/api/channels">
         <atom:title>Channels</atom:title>
         <atom:categories>
            <atom:category scheme="http://www.vizrt.com/types" term="channels"/>
         </atom:categories>
         <vaext:resolve type="application/atom+xml;type=entry" id="vizid:channel"
 template="http://127.0.0.1:9991/api/channels/~{vizid:channel}"/>
      </atom:collection>
   </atom:workspace>
</atom:service>
```

## 23.8.2 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/atomsvc+xml
<atom:feed xmlns:vaext="http://www.vizrt.com/atom-ext" xmlns:vizid="http://
www.vizrt.com/opensearch/vizid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub" xmlns="http://www.w3.org/2007/
app" xmlns:atom="http://www.w3.org/2005/Atom">
   <atom:entry>
      <inthub:name>test</inthub:name>
      <atom:id>test</atom:id>
      <atom:link rel="self" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test"/>
      <atom:link rel="playlists" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/test/playlists"/>
      <atom:link rel="settings" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/test/settings"/>
      <atom:link rel="page_pool" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/test/page_pool"/>
      <atom:link rel="state" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/test/state"/>
   </atom:entry>
   <atom:entry>
      <inthub:name>otherchannel</inthub:name>
      <atom:id>otherchannel</atom:id>
      <atom:link rel="self" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/otherchannel"/>
      <atom:link rel="playlists" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/otherchannel/playlists"/>
      <atom:link rel="settings" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/otherchannel/settings"/>
      <atom:link rel="page_pool" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/otherchannel/page_pool"/>
      <atom:link rel="state" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/otherchannel/state"/>
   </atom:entry>
</atom:feed>
```

### 23.8.3 GET Channel (Version 2.0)

Get the top endpoint for a single channel

### Syntax

```
GET /api/channels/test
```

### Request

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

### Response

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml<atom:entry xmlns:vaext="http://www.vizrt.com/
atom-ext" xmlns:vizid="http://www.vizrt.com/opensearch/vizid" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/
integrationshub" xmlns="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/
2005/Atom">
   <inthub:name>test</inthub:name>
   <atom:id>test</atom:id>
   <atom:link rel="self" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test"/>
   <atom:link rel="playlists" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/test/playlists"/>
   <atom:link rel="settings" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/test/settings"/>
   <atom:link rel="page_pool" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/test/page_pool"/>
   <atom:link rel="state" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/test/state"/>
</atom:entry>
```

## 23.8.4 GET Channel State (Version 2.0)

Retrieve channel state info. It is a collection of all playlists states. Each state link is also found via each playlists state-downlink.

### Syntax

```
GET /api/channels/test/state
```

### Request

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

### Response

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml
<atom:entry xmlns:vaext="http://www.vizrt.com/atom-ext" xmlns:vizid="http://
www.vizrt.com/opensearch/vizid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub" xmlns="http://www.w3.org/2007/
app" xmlns:atom="http://www.w3.org/2005/Atom">
   <inthub:name>test</inthub:name>
   <atom:id>test</atom:id>
   <atom:link rel="self" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test/state"/>
   <inthub:running_playlists>
      <inthub:running_playlist>
         <inthub:name>continuous</inthub:name>

<atom:id>_slash_VizRundown_slash_channels_slash_test_slash_playlists_slash_continuous
</atom:id>
         <inthub:description>Continuous Playlist</inthub:description>
         <inthub:playlist>/VizRundown/channels/test/playlists/continuous</
inthub:playlist>
         <inthub:user>jom</inthub:user>
         <inthub:state>running</inthub:state>
      </inthub:running_playlist>
   </inthub:running_playlists>
</atom:entry>
```

## 23.8.5 GET Channel Settings (Version 2.0)

Get the complete settings for a specific channel

### Syntax

```
GET /api/channels/test/settings
```

## Request

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

## Response

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml<atom:entry xmlns:vaext="http://www.vizrt.com/
atom-ext" xmlns:vizid="http://www.vizrt.com/opensearch/vizid" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:inthub="http://www.vizrt.com/
integrationshub" xmlns="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/
2005/Atom">
   <inthub:name>settings</inthub:name>
   <atom:id>settings</atom:id>
   <atom:link rel="self" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test/settings"/>
   <inthub:general>
      <inthub:entry
name="GeneralSettings.BasicSettings.ChannelBasicInfo.ScreenDimension">1920*1080</
inthub:entry>
      <inthub:entry
name="GeneralSettings.BasicSettings.ChannelBasicInfo.AspectRatio">16/9</inthub:entry>
      <inthub:entry
name="GeneralSettings.BasicSettings.ChannelBasicInfo.FramesPerSec">25</inthub:entry>
      <inthub:entry
name="GeneralSettings.BasicSettings.ChannelBasicInfo.BroadcastDayStartTime">04:00:
00</inthub:entry>
      <inthub:entry
name="GeneralSettings.BasicSettings.TriggeringControl.TimedPlayout">false</
inthub:entry>
      <inthub:entry
name="GeneralSettings.BasicSettings.TriggeringControl.TriggerType">Primary Events</
inthub:entry>
      <inthub:entry
name="GeneralSettings.BasicSettings.PlaylistDateManagement.AllEventsInOnePlaylist">tr
ue</inthub:entry>
      <inthub:entry
name="GeneralSettings.TimezoneSettings.Timezone.ChannelTimeZone">+07:00</
inthub:entry>
      <inthub:entry
name="GeneralSettings.TimezoneSettings.Timezone.ScheduleTimeZone">+07:00</
inthub:entry>
      <inthub:entry
name="GeneralSettings.TimezoneSettings.DSTSetting.ChannelInDSTTimeZone">false</
inthub:entry>
      <inthub:entry
name="GeneralSettings.TimezoneSettings.DSTSetting.DynamicDSTSetting">true</
inthub:entry>
      <inthub:entry
name="GeneralSettings.TimezoneSettings.DynamicDSTSetting.DynamicStartDate">6:7:3</
inthub:entry>
```

```
      <inthub:entry
name="GeneralSettings.TimezoneSettings.DynamicDSTSetting.DynamicStartTime">01:00:00</
inthub:entry>
      <inthub:entry
name="GeneralSettings.TimezoneSettings.DynamicDSTSetting.DynamicEndDate">6:7:10</
inthub:entry>
      <inthub:entry
name="GeneralSettings.TimezoneSettings.DynamicDSTSetting.DynamicEndTime">02:00:00</
inthub:entry>
      <inthub:entry
name="GeneralSettings.TimezoneSettings.ExplicitDSTSetting.ExplicitStartDate">01-08
-2013</inthub:entry>
      <inthub:entry
name="GeneralSettings.TimezoneSettings.ExplicitDSTSetting.ExplicitStartTime">01:00:
00</inthub:entry>
      <inthub:entry
name="GeneralSettings.TimezoneSettings.ExplicitDSTSetting.ExplicitEndDate">31-08
-2013</inthub:entry>
      <inthub:entry
name="GeneralSettings.TimezoneSettings.ExplicitDSTSetting.ExplicitEndTime">02:00:00</
inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.GeneralPlayoutOptions.AllowItemReplay">true</
inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.PlaylistOptions.IsAutoExpandRunningElement">true<
/inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.PlaylistOptions.LockViewToRuningPrimaryEvent">fal
se</inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.PlaylistOptions.DefaultPrimaryEventDuration">00:
10:00:00</inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.PlaylistOptions.DefaultLiveEventDuration">01:00:
00:00</inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.EditOptions.ClearPageIdAfterEdit">false</
inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.SceneImport.ReportDVEErrors">false</inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.VizOneIntegration.AutoExtractVizOneMetaData">fals
e</inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.PlaylistArchiving.EnableArchiving">false</
inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.PlaylistArchiving.ArchivingNumOfDays">0</
inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.PlaylistArchiving.ArchivingFolder"/>
```

```
      <inthub:entry
name="GeneralSettings.Miscellaneous.PlaylistArchiving.ArchivingTime">00:00:00</
inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.PlaylistArchiving.DeletePages">0</inthub:entry>
      <inthub:entry
name="GeneralSettings.Miscellaneous.DefaultScenes.AutomaticallyTakeLogoPage">false</
inthub:entry>
      <inthub:entry name="GeneralSettings.Miscellaneous.DefaultScenes.LogoScene"/>
      <inthub:entry name="GeneralSettings.Miscellaneous.DefaultScenes.ErrorScene"/>
      <inthub:entry name="GeneralSettings.Miscellaneous.DefaultScenes.VideoOnlyScene"
/>
      <inthub:entry name="GeneralSettings.Miscellaneous.QuickTakePages"/>
      <inthub:entry
name="GeneralSettings.Miscellaneous.Reporting.PlaylistReportsFolder">C:
\ProgramData\Vizrt\Multichannel\Reports\Playlist</inthub:entry>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.ChannelName.ChannelName"/>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.RemoteHosts.RemoteHosts"/>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.AutomaticImportSettings.EnableAutomaticImpo
rt">false</inthub:entry>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.AutomaticImportSettings.ImportFolder"/>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.AutomaticImportSettings.UpdateFolder"/>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.ImportOptions.CreatePlaceholderForHouseId">t
rue</inthub:entry>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.ImportOptions.ImportSecondaryAsRef">false</
inthub:entry>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.ImportOptions.RemoveSecondaryEventsExceedin
gPrimaryDuration">false</inthub:entry>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.ImportOptions.AllowPagesByRulesWithOffsetEx
ceedingPrimaryDuration">false</inthub:entry>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.ImportOptions.AllowPageToForceMappingForPag
eContentFilling">false</inthub:entry>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.PageContentFilling.RefreshPageContentofPlay
list">true</inthub:entry>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.PageContentFilling.RefreshPageContentofRunn
ingPlaylist">false</inthub:entry>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.PageContentFilling.SkipRefreshAlreadyPlayed
Page">true</inthub:entry>
```

```
      <inthub:entry
name="Import_slash_Export.GeneralSettings.PageContentFilling.PromoServerAddress"/>
      <inthub:entry
name="Import_slash_Export.GeneralSettings.PageContentFilling.TranslationTables"/>
      <inthub:entry
name="Import_slash_Export.DaySwitchOptions.DaySwitchOptions.ActivateDaySwitch">false<
/inthub:entry>
      <inthub:entry
name="Import_slash_Export.DaySwitchOptions.DaySwitchOptions.ActivatePlaylist">00:00:
00</inthub:entry>
      <inthub:entry
name="Import_slash_Export.DaySwitchOptions.DaySwitchOptions.InsertCleanUp">false</
inthub:entry>
      <inthub:entry
name="Import_slash_Export.DaySwitchOptions.DaySwitchOptions.TimeBeforeCleanUp">00:00:
00</inthub:entry>
      <inthub:entry
name="Import_slash_Export.DaySwitchOptions.DaySwitchOptions.InsertInitialize">false</
inthub:entry>
      <inthub:entry
name="Import_slash_Export.DaySwitchOptions.DaySwitchOptions.TimeBeforeInitialize">00:
00:00</inthub:entry>
      <inthub:entry
name="Import_slash_Export.FilteringOptions.ImportWindow.ImportStartOffset"/>
      <inthub:entry
name="Import_slash_Export.FilteringOptions.ImportWindow.ImportEndOffset"/>
      <inthub:entry
name="Import_slash_Export.FilteringOptions.ImportWindow.ImportPrimaryEventsOutsideImp
ortWindow">false</inthub:entry>
      <inthub:entry
name="Import_slash_Export.FilteringOptions.ImportWindow.MarkPartiallyImportedSchedule
AsDone">true</inthub:entry>
      <inthub:entry name="Import_slash_Export.FilteringOptions.PrimaryEventFiltering"
/>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.AutomaticUpdateMode">Full</
inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.DeleteProcessedFiles">2</
inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.MSELockEvents">00:01:00</
inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.ClearPlayedEvents">00:00:00</
inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.ClearUnplayedEvents">00:00:
00</inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.AllowUpdatesOfRunningPlaylist">f
alse</inthub:entry>
```

```
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.LockEditedPageWhenUpdate">true<
/inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.AllowUpdatesOfAlreadyPlayedSec
ondaryEvents">false</inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.RetrieveDateFromRunningPlaylis
t">false</inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.CreatePlaylistIfNotExists">fal
se</inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.SendPlaylist">false</
inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.SendPages">false</
inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.SendActions">false</
inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateOptions.SendPagesWhenNoop">false</
inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.FullUpdateOptionsForSecEvents.GetFromExterna
lPlaylist">true</inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.FullUpdateOptionsForSecEvents.RescheduleByRu
les">true</inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.FullUpdateOptionsForSecEvents.LeaveManuallyA
ddedItems">true</inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateTimeOptionsForSecEvents.GetFromExterna
lPlaylist">true</inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateTimeOptionsForSecEvents.RescheduleByRu
les">true</inthub:entry>
      <inthub:entry
name="Import_slash_Export.UpdateSettings.UpdateTimeOptionsForSecEvents.LeaveManuallyA
ddedItems">true</inthub:entry>
      <inthub:entry
name="Import_slash_Export.ExportSettings.ExportOptions.AutoExportFolder"/>
      <inthub:entry
name="Import_slash_Export.ExportSettings.ExportOptions.ChannelNameInAutomation"/>
      <inthub:entry
name="Import_slash_Export.ExportSettings.ExportOptions.ExportSecEventsWithNoOffset">f
alse</inthub:entry>
      <inthub:entry
name="Import_slash_Export.ExportSettings.SecEventIDAssignment.AssignmentMode">Regener
ate</inthub:entry>
```

```
      <inthub:entry
name="Import_slash_Export.ExportSettings.SecEventIDAssignment.GenerateRandomId">true<
/inthub:entry>
      <inthub:entry
name="Import_slash_Export.ExportSettings.SecEventIDAssignment.CapitalLettersForRandom
Id">true</inthub:entry>
      <inthub:entry
name="Import_slash_Export.ExportSettings.SecEventIDAssignment.PrefixSecondaryId"/>
      <inthub:entry
name="Import_slash_Export.ExportSettings.SecEventIDAssignment.InitialSecondaryId"/>
      <inthub:entry
name="Import_slash_Export.ExportSettings.Harris.MergeIntoExistingFile">false</
inthub:entry>
      <inthub:entry
name="Import_slash_Export.GraphicSettings.GraphicSettings.IncludeGraphicsWhenSendingP
laylist">false</inthub:entry>
      <inthub:entry
name="Import_slash_Export.GraphicSettings.GraphicSettings.IncludeGraphicsWhenReceivin
gPlaylist">false</inthub:entry>
      <inthub:entry
name="Import_slash_Export.GraphicSettings.GraphicSettings.SharedVizArchiveFolder"/>
      <inthub:entry name="ExternalTriggering.GPI.IsEnabled">false</inthub:entry>
      <inthub:entry
name="ExternalTriggering.GPI.GPIDeviceDefinition.BitsPerSecond">19200</inthub:entry>
      <inthub:entry name="ExternalTriggering.GPI.GPIDeviceDefinition.Port">----</
inthub:entry>
      <inthub:entry name="ExternalTriggering.GPI.GPIDeviceDefinition.StopBits">1</
inthub:entry>
      <inthub:entry name="ExternalTriggering.GPI.GPIDeviceDefinition.DataBits">8</
inthub:entry>
      <inthub:entry name="ExternalTriggering.GPI.GPIDeviceDefinition.Parity">None</
inthub:entry>
      <inthub:entry
name="ExternalTriggering.GPI.GPIDeviceDefinition.FlowControl">None</inthub:entry>
      <inthub:entry name="ExternalTriggering.GPI.GPIDeviceDefinition.BoxType">SRC-8</
inthub:entry>
      <inthub:entry name="ExternalTriggering.GPI.GPIDeviceDefinition.IP"/>
      <inthub:entry name="ExternalTriggering.GPI.GPIDeviceDefinition.Unit">0</
inthub:entry>
      <inthub:entry name="ExternalTriggering.GPI.GPIDeviceDefinition.SlaveId">0</
inthub:entry>
      <inthub:entry name="ExternalTriggering.GPI.GPIActionsMapping"/>
      <inthub:entry name="ExternalTriggering.VDCP.IsEnabled">false</inthub:entry>
      <inthub:entry name="ExternalTriggering.VDCP.VDCP.IsForceExtendedMode">false</
inthub:entry>
      <inthub:entry name="ExternalTriggering.VDCP.VDCP.BitsPerSecond">38400</
inthub:entry>
      <inthub:entry name="ExternalTriggering.VDCP.VDCP.Port">----</inthub:entry>
      <inthub:entry name="ExternalTriggering.VDCP.VDCP.ByteSize">8</inthub:entry>
      <inthub:entry name="ExternalTriggering.VDCP.VDCP.ExtraStopBits">1</
inthub:entry>
      <inthub:entry name="ExternalTriggering.VDCP.VDCP.Parity">1</inthub:entry>
      <inthub:entry name="ExternalTriggering.IntelligentInterface.IsEnabled">false</
inthub:entry>
      <inthub:entry
name="ExternalTriggering.IntelligentInterface.IntelligentInterface.BitsPerSecond">192
00</inthub:entry>
      <inthub:entry
name="ExternalTriggering.IntelligentInterface.IntelligentInterface.Port">----</
inthub:entry>
      <inthub:entry
name="ExternalTriggering.IntelligentInterface.IntelligentInterface.Encoding">UTF-8</
inthub:entry>
      <inthub:entry
name="ExternalTriggering.IntelligentInterface.IntelligentInterface.Space">false</
inthub:entry>
      <inthub:entry name="ExternalTriggering.TCP.IsEnabled">false</inthub:entry>
      <inthub:entry name="ExternalTriggering.TCP.TCP.Port"/>
      <inthub:entry
name="ExternalDataSources.DatabaseConnections.DatabaseConnections.Links"/>
      <inthub:entry
name="ExternalDataSources.UpdateService.UpdateService.HostName">http://localhost:
56999</inthub:entry>
      <inthub:entry
name="ExternalDataSources.UpdateService.UpdateService.Take">true</inthub:entry>
      <inthub:entry
name="ExternalDataSources.UpdateService.UpdateService.TakeTimeout">5</inthub:entry>
      <inthub:entry
name="ExternalDataSources.UpdateService.UpdateService.Preview">true</inthub:entry>
      <inthub:entry
name="ExternalDataSources.UpdateService.UpdateService.PreviewTimeout">5</
inthub:entry>
      <inthub:entry
name="ExternalDataSources.UpdateService.UpdateService.Update">true</inthub:entry>
      <inthub:entry
name="ExternalDataSources.UpdateService.UpdateService.UpdateTimeout">5</inthub:entry>
   </inthub:general>
   <inthub:mappings>
      <inthub:mapping xsi:type="inthub:xml" type="Any XML">
         <inthub:filenamepattern>*.*</inthub:filenamepattern>
         <inthub:schedulefilenameinfo/>
         <inthub:columnmappings>
            <inthub:columnmapping column="channel" field_name="channel"/>
            <inthub:columnmapping field_name="ignore"/>
            <inthub:columnmapping field_name="date"/>
            <inthub:columnmapping field_name="time"/>
            <inthub:columnmapping column="@dateT@time" field_name="begin_at"
 caption="Begin Time"/>
            <inthub:columnmapping field_name="duration" caption="Duration"
 edit_events="true"/>
            <inthub:columnmapping column="offset" field_name="timecode"
 caption="Offset" edit_events="true"/>
            <inthub:columnmapping column="title" field_name="description"
 caption="Description" edit_events="true"/>
```

```
            <inthub:columnmapping field_name="event_id" caption="Event ID"/>
            <inthub:columnmapping field_name="trigger_id" caption="Media ID"/>
            <inthub:columnmapping field_name="event_type"/>
         </inthub:columnmappings>
         <inthub:primarydateformat>DD/MM/YYYY</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>/</inthub:primaryeventpath>
         <inthub:secondaryinsideprimaryevents>false</
inthub:secondaryinsideprimaryevents>
         <inthub:secondaryeventpath>/</inthub:secondaryeventpath>
         <inthub:channelnodelocationtakenfromroot>true</
inthub:channelnodelocationtakenfromroot>
         <inthub:datenodelocationyakenfromroot>true</
inthub:datenodelocationyakenfromroot>
      </inthub:mapping>
   </inthub:mappings>
   <inthub:primaryeventfilters/>
</atom:entry>
```

## 23.8.6 GET Channel Page Pool (Version 2.0)

Get the content, the listed pages, of the channels Page Pool.

### Syntax

```
GET /api/channels/test/page_pool
```

### Request

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

### Response

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml
<atom:feed xmlns:vaext="http://www.vizrt.com/atom-ext" xmlns:vizid="http://
www.vizrt.com/opensearch/vizid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub" xmlns="http://www.w3.org/2007/
app" xmlns:atom="http://www.w3.org/2005/Atom">
   <atom:entry>
      <inthub:name>logo</inthub:name>
      <atom:id>logo</atom:id>
      <atom:link rel="self" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test/page_pool/logo"/>
      <inthub:description>GEOM*products/MCP/Packages/Music_demo/CONCEPT=HD/
DEFAULT_LOGO</inthub:description>
   </atom:entry>
   <atom:entry>
      <inthub:name>lowerthird</inthub:name>
      <atom:id>lowerthird</atom:id>
      <atom:link rel="self" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test/page_pool/lowerthird"/>
      <inthub:description>The Smashing Pumpkins/Bullet With Butterfly Wings</
inthub:description>
   </atom:entry>
</atom:feed>
```

## 23.8.7 GET Channel Playlists (Version 2.0)

Get the full collection of playlists associated with this channel.

### Syntax

```
GET /api/channels/test/playlists
```

### Request

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

## Response

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml
<atom:feed xmlns:vaext="http://www.vizrt.com/atom-ext" xmlns:vizid="http://
www.vizrt.com/opensearch/vizid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub" xmlns="http://www.w3.org/2007/
app" xmlns:atom="http://www.w3.org/2005/Atom">
   <atom:entry>
      <inthub:name>channel_pool</inthub:name>
      <atom:id>channel_pool</atom:id>
      <atom:link rel="self" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/channel_pool"/>
      <atom:link rel="edit" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/channel_pool"/>
      <atom:link rel="refresh" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/channel_pool/refresh"/>
      <atom:link rel="primaryevents" type="application/atom+xml;type=feed"
 href="http://127.0.0.1:9991/api/channels/test/playlists/channel_pool/primaryevents"/
>
      <atom:link rel="state" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/test/playlists/channel_pool/state"/>
      <atom:meta_data/>
      <inthub:description>Channel Pool</inthub:description>
   </atom:entry>
   <atom:entry>
      <inthub:name>continuous</inthub:name>
      <atom:id>continuous</atom:id>
      <atom:link rel="self" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous"/>
      <atom:link rel="edit" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous"/>
      <atom:link rel="refresh" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/refresh"/>
      <atom:link rel="primaryevents" type="application/atom+xml;type=feed"
 href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents"/>
      <atom:link rel="state" type="application/atom+xml;type=feed" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/state"/>
      <atom:meta_data/>
      <inthub:description>Continuous Playlist</inthub:description>
      <inthub:last_taken_element>primaryevents/21479131448478/secondaryevents/
070bf70c-0dbd-4fa6-a609-71bd34d6361c</inthub:last_taken_element>
      <inthub:broadcastDate>continuous</inthub:broadcastDate>
   </atom:entry>
</atom:feed>
```

## 23.8.8 GET 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/test/playlists/continuous
```

### Request

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

### Response

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml
<atom:entry xmlns:vaext="http://www.vizrt.com/atom-ext" xmlns:vizid="http://
www.vizrt.com/opensearch/vizid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub" xmlns="http://www.w3.org/2007/
app" xmlns:atom="http://www.w3.org/2005/Atom">
        <inthub:name>continuous</inthub:name>
        <atom:id>continuous</atom:id>
        <atom:link rel="self" type="application/atom+xml; type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous" />
        <atom:link rel="edit" type="application/atom+xml; type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous" />
        <atom:link rel="refresh" type="application/atom+xml; type=entry"
 href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/refresh" />
        <atom:link rel="primaryevents" type="application/atom+xml; type=feed"
 href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents" />
        <atom:link rel="state" type="application/atom+xml; type=feed" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/state" />
        <atom:meta_data />
        <inthub:description>Continuous Playlist</inthub:description>
        <inthub:broadcastDate>continuous</inthub:broadcastDate>
        <inthub:import_file>D:\Schedules\20100923_demo.txt</inthub:import_file>
</atom:entry>
```

## 23.8.9 GET Playlist State (Version 2.0)

Retrieve the state-information of a playlist.

## Syntax

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

## Request

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

## Response if playlist is RUNNING

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml
<atom:entry xmlns:vaext="http://www.vizrt.com/atom-ext" xmlns:vizid="http://
www.vizrt.com/opensearch/vizid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub" xmlns="http://www.w3.org/2007/
app" xmlns:atom="http://www.w3.org/2005/Atom">
   <inthub:name>continuous</inthub:name>

<atom:id>_slash_VizRundown_slash_channels_slash_test_slash_playlists_slash_continuous
</atom:id>
   <atom:link rel="self" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/state"/>
   <inthub:description>Continuous Playlist</inthub:description>
   <inthub:playlist>/VizRundown/channels/test/playlists/continuous</inthub:playlist>
   <inthub:user>NN</inthub:user>
   <inthub:state>running</inthub:state>
</atom:entry>
```

### Response if playlist is STOPPED

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml
<atom:entry xmlns:vaext="http://www.vizrt.com/atom-ext" xmlns:vizid="http://
www.vizrt.com/opensearch/vizid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub" xmlns="http://www.w3.org/2007/
app" xmlns:atom="http://www.w3.org/2005/Atom">
   <inthub:name>continuous</inthub:name>

<atom:id>_slash_VizRundown_slash_channels_slash_test_slash_playlists_slash_continuous
</atom:id>
   <atom:link rel="self" type="application/atom+xml;type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/state"/>
   <inthub:description>Continuous Playlist</inthub:description>
   <inthub:playlist>/VizRundown/channels/test/playlists/continuous</inthub:playlist>
   <inthub:user>NN</inthub:user>
   <inthub:state>stopped</inthub:state>
</atom:entry>
```

## 23.8.10 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 entries under meta_data sub-section is not set in size or content, and depends on the actual meta data attributes on the primary events in the playlist.

### Syntax

```
GET /api/channels/test/playlists/continuous/primaryevents
```

### Request

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

## Response

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml
<atom:feed xmlns:vaext="http://www.vizrt.com/atom-ext" xmlns:vizid="http://
www.vizrt.com/opensearch/vizid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub" xmlns="http://www.w3.org/2007/
app" xmlns:atom="http://www.w3.org/2005/Atom">
        <atom:entry>
                <inthub:name>1051111_1_VM_16_01_1</inthub:name>
                <atom:id>1051111_1_VM_16_01_1</atom:id>
                <atom:link rel="self" type="application/atom+xml; type=entry"
 href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1051111_1_VM_16_01_1" />
                <atom:link rel="edit" type="application/atom+xml; type=entry"
 href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1051111_1_VM_16_01_1" />
                <atom:link rel="secondaryevents" type="application/atom+xml;
type=feed" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/
primaryevents/1051111_1_VM_16_01_1/secondaryevents" />
                <atom:meta_data>
                        <atom:attribute name="continue_count">0</atom:attribute>
                        <atom:attribute name="created">6/11/2018 17:33:51</
atom:attribute>
                        <atom:attribute name="date">2010-09-23</atom:attribute>
                        <atom:attribute name="fixed_duration">yes</atom:attribute>
                        <atom:attribute name="seqment">1</atom:attribute>
                        <atom:attribute name="time">08:50:15:24</atom:attribute>
                </atom:meta_data>
                <inthub:description>In-channel trailer</inthub:description>
                <inthub:begin_at>2018-11-06T07:13:40.000Z</inthub:begin_at>
                <inthub:duration>00:00:35:00</inthub:duration>
                <inthub:ignore>False</inthub:ignore>
                <inthub:event_id>1051111_1_VM_16_01_1</inthub:event_id>
                <inthub:trigger_id>1051111_1_VM_16_01_1</inthub:trigger_id>
                <inthub:instance>composite_test</inthub:instance>
                <inthub:adjust_begin_at_after_resume>yes</
inthub:adjust_begin_at_after_resume>
                <inthub:isManuallyEdited>true</inthub:isManuallyEdited>
                <inthub:available>1.00</inthub:available>
                <inthub:loaded>1.00</inthub:loaded>
                <inthub:take_count>0</inthub:take_count>
                <inthub:infoIntegrationsHub>[A:13:27:24]</inthub:infoIntegrationsHub>
        </atom:entry>
        <atom:entry>
                <inthub:name>1053845_2_VM_16_01_1</inthub:name>
                <atom:id>1053845_2_VM_16_01_1</atom:id>
                <atom:link rel="self" type="application/atom+xml; type=entry"
 href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1053845_2_VM_16_01_1" />
                <atom:link rel="edit" type="application/atom+xml; type=entry"
 href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1053845_2_VM_16_01_1" />
                <atom:link rel="secondaryevents" type="application/atom+xml;
type=feed" href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/
primaryevents/1053845_2_VM_16_01_1/secondaryevents" />
                <atom:meta_data>
                        <atom:attribute name="continue_count">0</atom:attribute>
                        <atom:attribute name="created">6/11/2018 17:33:51</
atom:attribute>
                        <atom:attribute name="date">2010-09-23</atom:attribute>
                        <atom:attribute name="fixed_duration">yes</atom:attribute>
                        <atom:attribute name="seqment">1</atom:attribute>
                        <atom:attribute name="time">09:30:04:24</atom:attribute>
                </atom:meta_data>
                <inthub:description>News Today</inthub:description>
                <inthub:begin_at>2018-11-06T11:30:00.000Z</inthub:begin_at>
                <inthub:duration>00:25:00:00</inthub:duration>
                <inthub:ignore>False</inthub:ignore>
                <inthub:event_id>1053845_2_VM_16_01_1</inthub:event_id>
                <inthub:trigger_id>1053845_2_VM_16_01_1</inthub:trigger_id>
                <inthub:instance>composite_test</inthub:instance>
                <inthub:adjust_begin_at_after_resume>no</
inthub:adjust_begin_at_after_resume>
                <inthub:pause_after_scheduling>yes</inthub:pause_after_scheduling>
                <inthub:isManuallyEdited>true</inthub:isManuallyEdited>
                <inthub:available>1.00</inthub:available>
                <inthub:loaded>1.00</inthub:loaded>
                <inthub:take_count>0</inthub:take_count>
                <inthub:infoIntegrationsHub>[U:13:49:12]</inthub:infoIntegrationsHub>
        </atom:entry>
</atom:feed>
```

## 23.8.11 GET Primary Event (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 events in the playlist.

### Syntax

```
GET /api/channels/test/playlists/continuous/primaryevents/[primary_event_id]
```

## Request

```
GET /api/channels/test/playlists/continuous/primaryevents/[primary_event_id] HTTP/1.1
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml
<atom:entry xmlns:vaext="http://www.vizrt.com/atom-ext" xmlns:vizid="http://
www.vizrt.com/opensearch/vizid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub" xmlns="http://www.w3.org/2007/
app" xmlns:atom="http://www.w3.org/2005/Atom">
        <inthub:name>1051111_1_VM_16_01_1</inthub:name>
        <atom:id>1051111_1_VM_16_01_1</atom:id>
        <atom:link rel="self" type="application/atom+xml; type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1051111_1_VM_16_01_1" />
        <atom:link rel="edit" type="application/atom+xml; type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1051111_1_VM_16_01_1" />
        <atom:link rel="secondaryevents" type="application/atom+xml; type=feed"
 href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1051111_1_VM_16_01_1/secondaryevents" />
        <atom:meta_data>
                <atom:attribute name="continue_count">0</atom:attribute>
                <atom:attribute name="created">6/11/2018 17:33:51</atom:attribute>
                <atom:attribute name="date">2010-09-23</atom:attribute>
                <atom:attribute name="fixed_duration">yes</atom:attribute>
                <atom:attribute name="seqment">1</atom:attribute>
                <atom:attribute name="time">08:50:15:24</atom:attribute>
        </atom:meta_data>
        <inthub:description>In-channel trailer</inthub:description>
        <inthub:begin_at>2018-11-06T07:13:40.000Z</inthub:begin_at>
        <inthub:duration>00:00:35:00</inthub:duration>
        <inthub:ignore>False</inthub:ignore>
        <inthub:event_id>1051111_1_VM_16_01_1</inthub:event_id>
        <inthub:trigger_id>1051111_1_VM_16_01_1</inthub:trigger_id>
        <inthub:instance>composite_test</inthub:instance>
        <inthub:adjust_begin_at_after_resume>yes</
inthub:adjust_begin_at_after_resume>
        <inthub:isManuallyEdited>true</inthub:isManuallyEdited>
        <inthub:available>1.00</inthub:available>
        <inthub:loaded>1.00</inthub:loaded>
        <inthub:take_count>0</inthub:take_count>
        <inthub:infoIntegrationsHub>[A:13:27:24]</inthub:infoIntegrationsHub>
</atom:entry>
```

### 23.8.12 GET Secondary Events under Primary Event (Version 2.0)

Follow the downlink under a primary event to list secondary events we get a collection of secondary events under this 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 secondary events in the playlist.

#### Syntax

```
GET /api/channels/test/playlists/continuous/primaryevents/[primary_event_id]/
secondaryevents
```

#### Request

```
GET /api/channels/test/playlists/continuous/primaryevents/[primary_event_id]/
secondaryevents HTTP/1.1
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml
<atom:feed xmlns:vaext="http://www.vizrt.com/atom-ext" xmlns:vizid="http://
www.vizrt.com/opensearch/vizid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub" xmlns="http://www.w3.org/2007/
app" xmlns:atom="http://www.w3.org/2005/Atom">
        <atom:entry>
                <inthub:name>b6fdce17-2ac8-4ae0-9f46-e11770b547da</inthub:name>
                <atom:id>b6fdce17-2ac8-4ae0-9f46-e11770b547da</atom:id>
                <atom:link rel="self" type="application/atom+xml; type=entry"
 href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1051867_1_VM_16_01_1/secondaryevents/b6fdce17-2ac8-4ae0-9f46-e11770b547da" />
                <atom:link rel="edit" type="application/atom+xml; type=entry"
 href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1051867_1_VM_16_01_1/secondaryevents/b6fdce17-2ac8-4ae0-9f46-e11770b547da" />
                <atom:meta_data>
                        <atom:attribute name="continue_count">0</atom:attribute>
                        <atom:attribute name="guid">2018-11-06T10:53:02Z_Johan-
PC2.vizrt.internal_12020_0</atom:attribute>
                        <atom:attribute name="infoPromoServer">[20:28:58]</
atom:attribute>
                        <atom:attribute name="modified_date">2018-11-06T10:53:02.
224Z</atom:attribute>
                        <atom:attribute name="showautodescription">true</
atom:attribute>
                        <atom:attribute name="templatedescription">Lower third</
atom:attribute>
                </atom:meta_data>
                <inthub:description>Preview: Man of Steel/You are watching</
inthub:description>
                <inthub:obj_id>6000</inthub:obj_id>
                <inthub:timecode>00:00:10:00</inthub:timecode>
                <inthub:take_count>0</inthub:take_count>
                <inthub:duration>00:00:12:00</inthub:duration>
                <inthub:created_by>schedule</inthub:created_by>
                <inthub:infoIntegrationsHub>[U:14:02:56]</inthub:infoIntegrationsHub>
        </atom:entry>
        <atom:entry>
                <inthub:name>3f8b34e1-2551-4d2c-83e2-19b81915e39b</inthub:name>
                <atom:id>3f8b34e1-2551-4d2c-83e2-19b81915e39b</atom:id>
                <atom:link rel="self" type="application/atom+xml; type=entry"
 href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1051867_1_VM_16_01_1/secondaryevents/3f8b34e1-2551-4d2c-83e2-19b81915e39b" />
                <atom:link rel="edit" type="application/atom+xml; type=entry"
 href="http://127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1051867_1_VM_16_01_1/secondaryevents/3f8b34e1-2551-4d2c-83e2-19b81915e39b" />
                <atom:meta_data>
```

```
                        <atom:attribute name="modified_date">2018-11-06T10:53:06.
258Z</atom:attribute>
                        <atom:attribute name="showautodescription">true</
atom:attribute>
                        <atom:attribute name="created">12/10/2018 03:44:38</
atom:attribute>
                </atom:meta_data>
                <inthub:description>1000</inthub:description>
                <inthub:obj_id>1000</inthub:obj_id>
                <inthub:timecode>00:00:00:00</inthub:timecode>
                <inthub:duration>00:13:39:00</inthub:duration>
                <inthub:created_by>rule</inthub:created_by>
        </atom:entry>
</atom:feed>
```

## 23.8.13 GET Secondary Event (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 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/test/playlists/continuous/primaryevents/[primary_event_id]/
secondaryevents/[page_name]
```

### Request

```
GET /api/channels/test/playlists/continuous/primaryevents/[primary_event_id]/
secondaryevents/[page_name] HTTP/1.1
```

## Response

```
HTTP/1.1 200 OK
Content-Type: application/atomsvc+xml
<atom:entry xmlns:vaext="http://www.vizrt.com/atom-ext" xmlns:vizid="http://
www.vizrt.com/opensearch/vizid" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:inthub="http://www.vizrt.com/integrationshub" xmlns="http://www.w3.org/2007/
app" xmlns:atom="http://www.w3.org/2005/Atom">
        <inthub:name>b6fdce17-2ac8-4ae0-9f46-e11770b547da</inthub:name>
        <atom:id>b6fdce17-2ac8-4ae0-9f46-e11770b547da</atom:id>
        <atom:link rel="self" type="application/atom+xml; type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1051867_1_VM_16_01_1/secondaryevents/b6fdce17-2ac8-4ae0-9f46-e11770b547da" />
        <atom:link rel="edit" type="application/atom+xml; type=entry" href="http://
127.0.0.1:9991/api/channels/test/playlists/continuous/primaryevents/
1051867_1_VM_16_01_1/secondaryevents/b6fdce17-2ac8-4ae0-9f46-e11770b547da" />
        <atom:meta_data>
                <atom:attribute name="continue_count">0</atom:attribute>
                <atom:attribute name="guid">2018-11-06T10:53:02Z_Johan-
PC2.vizrt.internal_12020_0</atom:attribute>
                <atom:attribute name="infoPromoServer">[20:28:58]</atom:attribute>
                <atom:attribute name="modified_date">2018-11-06T10:53:02.224Z</
atom:attribute>
                <atom:attribute name="showautodescription">true</atom:attribute>
                <atom:attribute name="templatedescription">Lower third</
atom:attribute>
        </atom:meta_data>
        <inthub:description>Preview: Man of Steel/You are watching</
inthub:description>
        <inthub:obj_id>6000</inthub:obj_id>
        <inthub:timecode>00:00:10:00</inthub:timecode>
        <inthub:take_count>0</inthub:take_count>
        <inthub:duration>00:00:12:00</inthub:duration>
        <inthub:created_by>schedule</inthub:created_by>
        <inthub:infoIntegrationsHub>[U:14:02:56]</inthub:infoIntegrationsHub>
</atom:entry>
```