Viz Ticker User Guide

Version 4.0

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

2021/06/29

Contents

1 Introduction

This is the user guide for Viz Ticker, an advanced ticker system that makes it possible to build creative and compelling graphic tickers with total flexibility.

1.1 Related Documents

For additional information, see the following documents:

- Viz Artist User Guide: Contains information about how to create scenes and details about ticker specific plug-ins and scene design.
- Viz Pilot User Guide: Contains information on how to operate Template Wizard, the application that creates presentation files.
- Feed Streamer User Guide: Describes how to capture data from web feeds that use output formatting that is not supported by Viz Ticker, and how to moderate and push this content to the Ticker System.

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   About Viz Ticker

Viz Ticker is an advanced ticker system that makes it possible to build creative and compelling graphic tickers with total flexibility. The system supports multiple clients for content insertion and control. Input can be manually entered, or automatically generated from various feeds; for example news feeds, databases, or the Internet.

With Viz Ticker, multiple graphical elements with various speed can be displayed simultaneously. The elements can include animated text, images, and so on. The real-time aspect of Viz Ticker takes the elements directly, with no delay, from content editing to air.

Viz Ticker has features that are not usually included in ticker products, for example:

- Ticker elements can trigger events when entering a scrolling ticker. An event may be the start of an animation, or the display of pre-defined text or graphic objects positioned anywhere on the screen. Stock price movements can generate up or down markers, voting results can render 3D charts, and so on.
- The system has multi-render support and can be set up to feed several multiple output channels from one message carousel. Multiple clients can control a shared ticker message pool, as well as performing on-air control.

This section contains information on the following topics:

- System Applications
- Ticker Creation Workflow
- Quick Start Guide
- Ticker Plug-ins
- Ticker OneBox

2.1   System Applications

Various applications make up the Ticker System:

Application	Description
Viz Ticker Client	The Viz Ticker client application, for operating and providing data to defined tickers. In this system, messages are created, added to a carousel, and sent On Air. The typical user of Viz Ticker Client is a journalist or operator.
Viz Ticker Web Client	A companion application to the desktop client enabling remote connection to the Ticker carousels. A typical user is a journalist or operator modifying the content of a ticker from a remote location.

Applica tion	Description
Ticker Wizard	The Viz Ticker configuration application for importing scenes created in Viz Artist and creating tickers. In this system, carousels are selected and equipped with templates. Adjustments of the GUI settings in Viz Ticker Client can also be defined from this system. The typical user of Ticker Wizard is a graphical artist.
Ticker Feed	A service for connecting external feeds to a Ticker System. It makes it possible to populate a scrolling or flipping carousel with data elements from a file that is updated from an external source. It is optional to include this component in the Viz Ticker setup.
Ticker Service	A component responsible for communicating ticker messages to the Viz Engine renderer. This component is necessary in order to run a scrolling carousel in Viz Ticker.
Ticker System Tool	A console application that performs various important system operations, such as archiving (export/import), upgrading and system deletion.

2.2   Ticker Creation Workflow
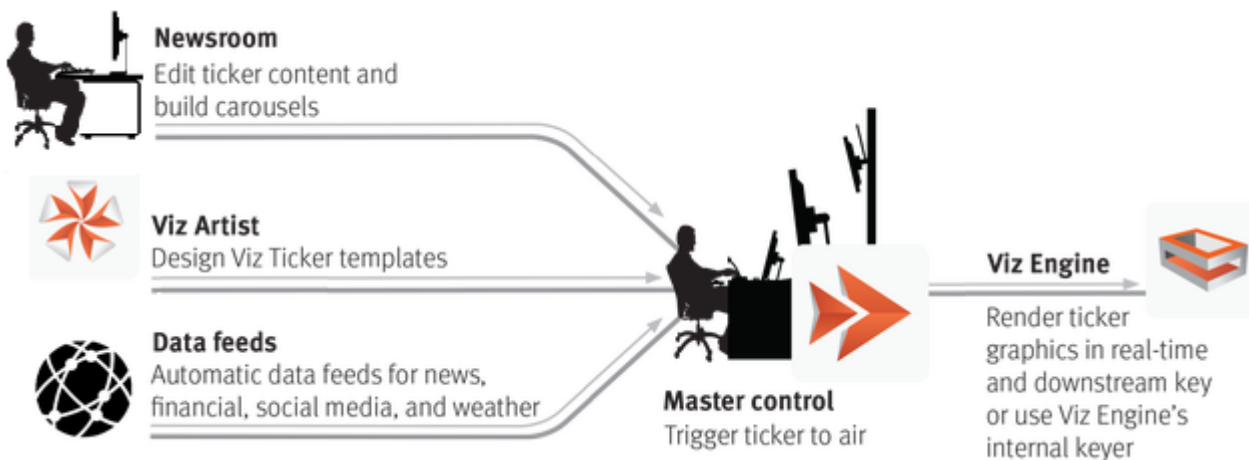
The Ticker creation workflow typically starts with the design of a scene in Viz Artist. The ticker can have almost any given design, except for a few Design Conventions that must be followed to make the ticker work properly.

The next step in the ticker creation workflow is to scan the scene in Ticker Wizard. The scanning process verifies the scene structure. After the scanning, the logic for the ticker can be built. This is done by performing a few quick configuration steps in Ticker Wizard.

The final step in the ticker creation workflow is to launch Viz Ticker Client. In this client application, messages can be created and added to the carousel. When the desired messages have been added, the ticker is ready to go On Air.

2.3 Quick Start Guide

This section lists the steps required to set up and use a basic Viz Ticker carousel. Each of the steps is described in more detail in the following sections of this User Guide.

2.3.1 Prepare All Systems

- Start *Viz Artist* and set **On Air.**
- Make sure that the *Media Sequencer* is running.
- Start *Ticker Wizard* and delete any existing carousels or output channels. These must be deleted before setting up a new Ticker System, see Ticker Carousels Panel and Output Channels Panel.
- Make sure that *Ticker Service* is running (only required for scrolling carousels).

2.3.2 Set Up a Carousel in Ticker Wizard

- Connect *Ticker Wizard* to Viz Engine (Viz Engine Connection Frame).
- Click **Select Scene** to select the carousel scene (Main Panel).
- Click **Create Scrolling/Flipping Carousel** and select the carousel and templates you require (Wizards).

2.3.3 Use a Carousel in Viz Ticker Client

- Start *Viz Ticker Client*.
- Select the **<CarouselName>** tab (Carousel Specific Panel).
- Create new messages by clicking **New Message** or **New Template Element.**
- Edit the contents of each message in the Elements Editor.
- Move the new messages to the Carousel List.

2.3.4 Activate the System

- In *Viz Ticker Client*, select the **CONTROL** tab (Control Panel). This is only visible if the Viz Ticker Client Startup Parameters include *-control on.*
- Set *Ticker System Main* to **On.**
- Set *Program Main* to **On.**
- Take the carousel on air by setting *<CarouselName> Main* to **On.** This button is available in the <CarouselName> tab and the CONTROL tab. Your ticker carousel should now be displayed in the rendering window.

2.4   Ticker Plug-Ins

In order for a ticker carousel to work properly, the Viz Artist scene that the carousel is based upon must be created according to specific Design Conventions. There are also various Viz Artist plug-ins that must be available in order for a ticker carousel to work properly:

- **Scroller:** The Scroller plug-in is a geometry plug-in, mostly used for creating scrolling tickers, which generates a dynamic line of scrolling elements. Information about the Scroller plug-in can be found in the *Viz Artist User Guide's Geometry Plugins/Ticker* section.
- **Scroller Action:** The Scroller Action plug-in makes it possible to design scrolling tickers that trigger actions when scroller elements are played out. Information about the Scroller Action plug-in can be found in the *Viz Artist User Guide's Container Plugins/Ticker* section.
- **Toggle:** The Toggle plug-in is used to toggle between objects in flipping carousel scenes. Information about the Toggle plug-in can be found in the *Viz Artist User Guide's Container Plugins/Container* section.

There are also numerous *Control* plug-ins that must be available in order to modify the ticker elements.

> ⚠ **Note:** The plug-ins mentioned in this section are required for Viz Ticker to operate correctly. These plug-ins are shipped with the Viz Artist installer/distribution, not with Viz Ticker.

See Also

- Viz Artist User Guide
- Design Conventions

2.5   Ticker OneBox

The Ticker OneBox system is a cost efficient alternative when running more than one output channel from the same Ticker system. Traditionally, there had to be one Ticker Service per physical output channel (Viz Engine), but with the OneBox system enabled, one Ticker service can now feed multiple Viz Engines on the same host. In this way, a full Ticker system (including Ticker Wizard, Ticker Client, Ticker Service and dual/quad Viz Engines) can run on one computer.

The output can come from a traditional SDI out from a Matrox video board, or from NDI out from each Engine. In the latter case, the host computer does not need a video board, making the solution even more cost effective.

2.5.1   Enabling Ticker OneBox

To enable a OneBox system, install and configure the Ticker system. For each Output channel, check the **Enable OneBox** checkbox. See Output Channels.

3   Installing Viz Ticker

Viz Ticker comes with several installers to setup the system.

This section contains information on the following topics:

- Important Before Installation
- Requirements
- Program Starting Sequence
- Installing Ticker Wizard
- Installing Viz Ticker Client
- Installing Ticker Web Client
- Installing Ticker Service
- Installing Ticker Feed
- Upgrading the Existing Viz Ticker Configuration
- Installation Directories
- System Log Files

Depending on your setup, you may also need to install the following:

- Ticker Service: Required for scrolling Ticker Systems.
- Ticker Feed: Used for populating carousels with data from an external source.

3.1   Important Before Installation

- It is recommended to uninstall previous versions of Viz Ticker or Ticker components before installing a new system or upgrading.
- Make sure that the Requirements are met. Also refer to the Release Notes.

3.2   Requirements

The following applications or services must be installed with the version indicated.

Program or Service	Required version
Media Sequencer	Version 2.0, or later.
Viz Artist/Viz Engine	Version 3.6.3, or later. Viz Engine version 3.7.1, or later, is required for the Field Linking with Apply Shared Memory workflow. Viz Engine version 3.7.0 should not be used.
Graphic Hub	Version 2.3.1, or later.

Program or Service	Required version
Graphic Hub REST	Version 1.1.0, or later Version 2.0.0 is required for VTW image search, see note below.
Microsoft .NET Framework	Version 4.6.1, or later.

For the .NET Framework, visit www.microsoft.com and search for .Net.

> ⚠ **Note:** Graphic Hub REST is only required if message templates are using tab-fields pointing to Graphic Hub resources, or if using the VTW image search functionality with Graphic Hub as search provider storage. See Search Providers on Graphic Hub.

3.2.1 System Requirements for Ticker OneBox

The Viz Ticker OneBox system requires the following versions of Viz Engine:

- 3.14.5 or above on the 3.x release line, or
- 4.3.0 or above on the 4.x release line

3.2.2 Operating System Support

Viz Ticker Client is a 32-bit application. It can run on both 32-bit and 64-bit versions of Windows. It has been tested with the software mentioned above on Windows 7 and Windows 10. It probably also runs on Windows 8 and Windows Server 2012.

Operating systems that have reached their end-of-life cycle and are no longer officially supported by their manufacturers, are not supported for use with Viz Ticker.

Viz Ticker Client supports *UAC* (User Access Control) and does not require administrator privileges to run, but in most cases requires admin rights to install.

> 🔴 **IMPORTANT!** Please review the Release Notes for Viz Ticker for any updates to the recommended version of Viz Ticker and supporting applications or services.

3.3 Program Starting Sequence

After installation, the applications must be started in a specific sequence:

1. The Media Sequencer must be started before Ticker Wizard and Viz Ticker Client.

   > ⚠ **Note:** A restart/shutdown of the Media Sequencer requires a restart/shutdown of Ticker Wizard and Viz Ticker Client.

2. Viz Engine must be started (and put On Air) before Ticker Wizard.

3. Ticker Service must be running before *Ticker System* is enabled in Viz Ticker Client.

> ⚠ **Note:** Ticker Wizard and Viz Ticker Client can run while starting Ticker Service.

3.4   Installing Ticker Wizard

3.4.1   Installing Ticker Wizard

> ⚠ **Note:** If Ticker Wizard is already installed on the machine, remove the old version before installing the new version. Also, the Ticker System Tool is installed together with Ticker Wizard, if this option is selected during installation.

1. Run `TickerWizard-<Product Version>.msi`.

3.4.2   Uninstalling Ticker Wizard

1. Select **Start > Control Panel > Programs > Uninstall a program**.

> ⚠ **Note:** The software settings are not deleted when Ticker Wizard is uninstalled.

3.4.3   Ticker Wizard Startup Parameters

After installing Ticker Wizard, the Media Sequencer startup parameter can be modified in the Ticker Wizard Properties.
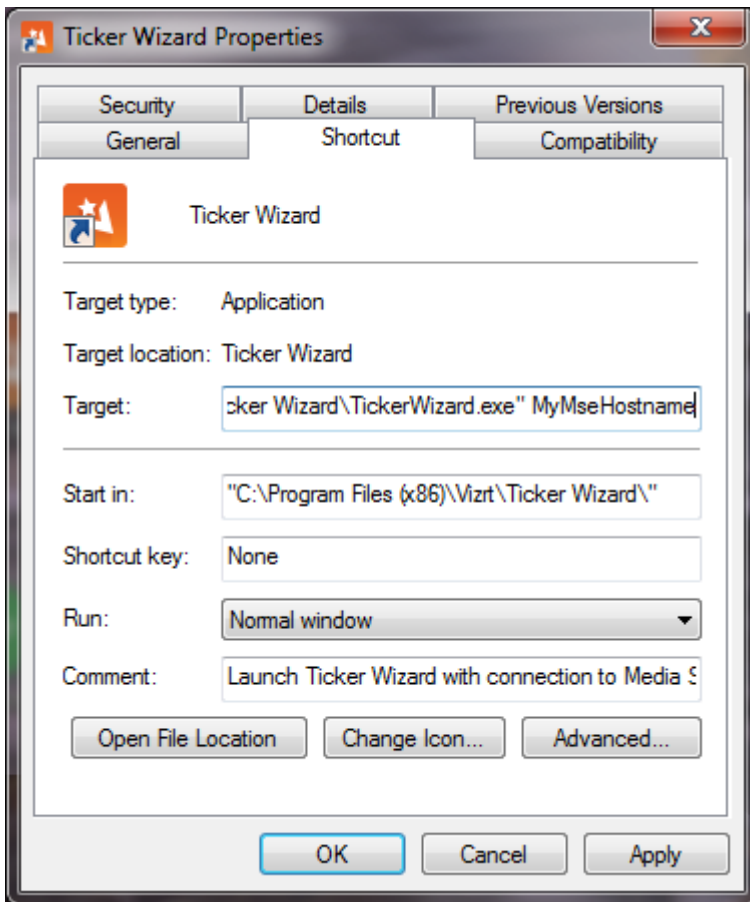
To modify the parameter, make sure that the **Shortcut** tab is selected, and modify the **Target** text box.

If Ticker Wizard should connect to a Media Sequencer on a specific machine, type the machine name as the first parameter after the program path.

See Also

- Upgrading the Existing Viz Ticker Configuration

3.5 Installing Viz Ticker Client

3.5.1 Installing Viz Ticker Client

> ⚠ **Note:** If Viz Ticker Client is already installed on the machine, remove it before installing the new version.

1. Run `VizTickerClient-<Product Version>.msi`.

3.5.2   Uninstalling Viz Ticker Client

1. Select *Start > Control Panel > Programs > Uninstall a program.*

> ⚠ **Note:** The software settings are not deleted when Viz Ticker Client is uninstalled.

3.5.3   Viz Ticker Client Startup Parameters

After installing Viz Ticker Client, various startup parameters can be modified in the Viz Ticker Client Properties window.
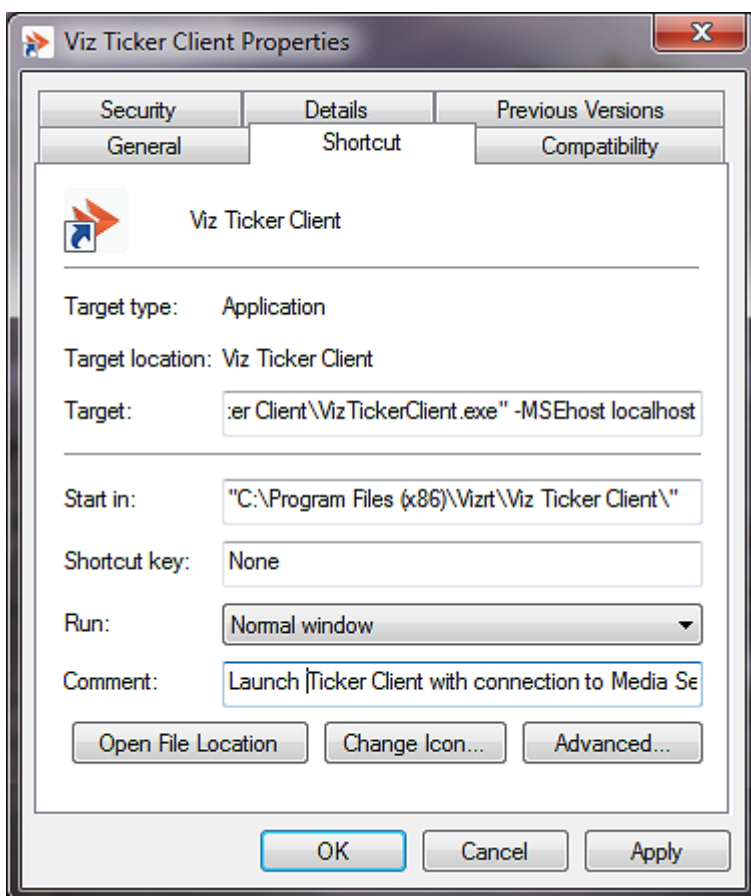
To modify the parameters, make sure that the **Shortcut** tab is selected, and modify the **Target** text box.

3.5.4   Viz Ticker Client Startup Parameters

Setting	Values	Description
-msehost	hostname	Hostname of the Media Sequencer for the Ticker System.
-vizdata	hostname	The Viz Engine host to use for the object pool browser.
-accessgroup	group name	Defines the group this client belongs to. Carousels can be given restricted access so that only clients belonging to one or more specific groups can access it. For more information, see Access Groups Frame.
-config	on|off	Defines whether or not the Configuration Panel should be visible in Viz Ticker Client. Usually this is set to off on journalist machines, and on on operator machines.
-control	on|off	Defines whether or not the Control Panel should be visible in Viz Ticker Client. Usually this is set to off on journalist machines, and on on operator machines.
-commandqueueport	number	Socket port for external access to client commands.

3.6   Installing Ticker Web Client

The Viz Ticker Web companion must be installed on the PC running the Media Sequencer. The web application is hosted from the Media Sequencer.

1. Copy the installer to the Media Sequencer PC.
2. Double click the installer and run it.

After the installation, the Ticker Web client can be found at http://localhost:8580/app/ticker/.

3.7   Installing Ticker Service

Ticker Service is a component responsible for storing and controlling the sequence of ticker messages in Viz Ticker. This component is necessary to run a scrolling carousel in Viz Ticker.

> ⬥ **IMPORTANT!** Viz Ticker *must* be installed and run on the Viz Engine used for output for scrolling carousels.

This section contains information on the following topics:

- To Install Ticker Service
- To Start Ticker Service as a Service
- To Start Ticker Service as a Console Application
- Ticker Service Options

3.7.1 To Install Ticker Service

> ⚠ **Note:** If Ticker Service is already installed on the machine, you must first remove the old program version before installing the new one.

- Run the `TickerService-<Product Version>.msi` installer.
- Follow the on-screen installation procedure.

3.7.2 To Start Ticker Service as a Service

1. At a command prompt (as Administrator) go to the installation directory of Ticker Service, see Installation Directories.

   > ⚠ **Note:** On **Windows 10**, Ticker Service must be run with administrator privileges. To do so, make sure the command line has administrator privileges (**Run as administrator**) and start the service as usual.

2. Install `TickerService` as a Windows service by running: `TickerService.exe --Service`
3. Start the service from the command line: `net start VizTicker`
   Expected output:
   ```
   The VizTicker service is starting.
   The VizTicker service was started successfully.
   ```

   > ✓ **Tip:** It is also possible to start Ticker Service as a service by using the *Microsoft Management Console.*

3.7.3 To Start Ticker Service as a Console Application

To actually see the actions of this service, it is possible to launch Ticker Service as a console application.

- Click **Start > All Programs > Vizrt > Viz Ticker > Ticker Service**.

> ⚠ **Note:** On **Windows 10**, Ticker Service must be run with administrator privileges. To do so, right-click *TickerService.exe* and Select **Run as administrator**. Or, to make Ticker Service run with administrator privileges every time, right-click *TickerService.exe* and select **Properties > Compatibility tab > Run this program as an administrator**.

> ✅ **Tip:** It is also possible to start Ticker Service as a console application by using the command line.

3.7.4   Ticker Service Options

By running *TickerService.exe* with `--help` as an argument, all Ticker Service Options are listed:

Options	Behavior
`--help`	Shows the Ticker Service help message that lists all these options.
`--log-level arg (=50)`	Defines the log level to use, a value between `0` and `100`.
`--port arg (=6301)`	Defines the port to bind the socket interface to.
`--ignore-inactive-uh-groups`	Ignores inactive UpdateHub groups, groups that have a field called *active* with value `0` are ignored.
`--RegServer`	Registers Ticker Service to run as a server in a console.
`--UnregServer`	Removes the Ticker Service from the registry, only valid when Ticker Service is registered as a server.
`--Service`	Registers Ticker Service to run as a service.
`--elevated`	Used internally by Ticker Service on Windows Vista to indicate whether Ticker Service has elevated privileges or not. This parameter should not be set by the user.

3.8   Installing Ticker Feed

> ⚠ **Note:** If Ticker Feed is already installed on the machine, you must first remove the old program version before installing the new one, see To uninstall Ticker Feed, see below.

1. Run `TickerFeed-<Product Version>.msi`.
2. Follow the on-screen instructions. Ticker Feed is now installed on the machine at the defined location.

3.8.1   To uninstall Ticker Feed

1. Select **Start > Control Panel > Programs > Uninstall a program.** Ticker Feed is no longer available on the machine.

> ⚠ **Note:** The software settings are not deleted with the software.

3.9   Upgrading The Existing Viz Ticker Configuration

> ✅ **Tip:** The Ticker System Tool can be used to upgrade Viz Ticker configuration 2.4, or later.

Whenever Viz Ticker is used, the configuration of the whole Ticker System gets stored in the Media Sequencer (Vdom configuration).

Each time Ticker Wizard is started, it checks the Media Sequencer configuration. If the Ticker Wizard version stored in the Media Sequencer configuration is older than the version now running, then Ticker Wizard is able to upgrade the Media Sequencer configuration to the latest version.

> ⚠ **Note:** In the Media Sequencer configuration, the currently installed version of Ticker Wizard is stored under `/config/versions/TickerSystemStandard`.

3.9.1   To Upgrade the Viz Ticker Configuration

1. Launch Ticker Wizard. If the Ticker Wizard version stored in the Media Sequencer configuration is older than the version running, then the upgrade starts. The Upgrade window appears asking whether you want to proceed.
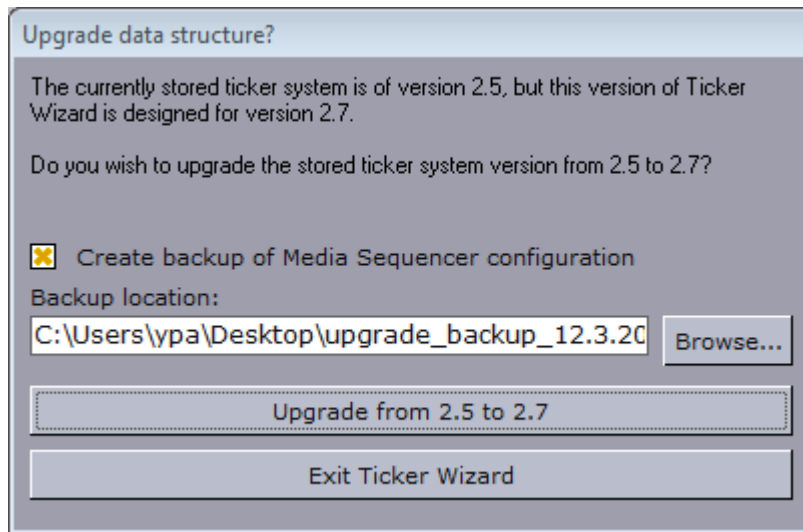
In this case, the Media Sequencer configuration is upgraded from Viz Ticker 2.5 to Viz Ticker 2.7.

2. Specify the desired path in the **Backup location** textbox. A copy of the current Media Sequencer configuration is saved at the given path in case the upgrade fails.
3. Click the **Upgrade from <old version> to <new version>** button.
4. If the upgrade is successful, Ticker Wizard starts.
5. If the upgrade is canceled or is not successful, Ticker Wizard cannot initialize since the current Media Sequencer configuration is not compatible with the new version of Ticker Wizard.
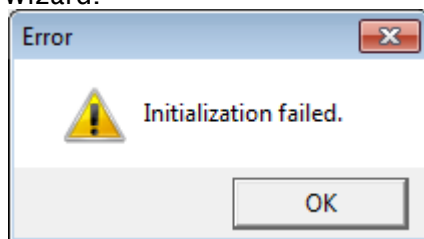
3.9.2   To Resolve a Failed Upgrade

The upgrade may fail, especially if the Media Sequencer configuration has been modified in a non-standard way. In case of failure, submit a support request to Vizrt.

Typically, upgrade issues are resolved by:

· Fixing the Media Sequencer configuration, if possible, and executing the upgrade again.
· Recreating the system with the new version of Ticker Wizard.
· Continuing to use the older version of the Ticker System until the issue has been fixed in a maintenance version of Ticker Wizard.

3.10   Installation Directories

Most commonly the applications are installed to `%ProgramFiles%` or `%ProgramFiles(x86)%` dependin g on 32- or 64-bit Windows. The actual installation directory is referred to as `<Installation-Directory>` later in this document. The standard installation directories are:

Application	Platform	Installation Directory
Ticker Wizard	32-bit Windows	`%ProgramFiles%\Vizrt\Ticker Wizard`
	64-bit Windows	`%ProgramFiles(x86)%\Vizrt\Ticker Wizard`
Viz Ticker Client	32-bit Windows	`%ProgramFiles%\Vizrt\Viz Ticker Client`
	64-bit Windows	`%ProgramFiles(x86)%\Vizrt\Viz Ticker Client`
Ticker Feed	32-bit Windows	`%ProgramFiles%\Vizrt\Ticker Feed`
	64-bit Windows	`%ProgramFiles(x86)%\Vizrt\Ticker Feed`
Ticker Service	32-bit Windows	`%ProgramFiles%\Vizrt\Ticker Service`
	64-bit Windows	`%ProgramFiles(x86)%\Vizrt\Ticker Service`

3.11   System Log Files

· Logfiles for **Viz Ticker Client**, **Ticker Feed**, **Ticker Wizard** and **Ticker System Tool** are stored at `%AppData%\Vizrt` in the directories *Viz Ticker Client*, *Ticker Feed*, *Ticker Wizard* and *TickerSystemTool*. Assuming your user running the programs is called *VizUser*, then you normally find config and logfiles in:

```
C:\Users\VizUser\AppData\Roaming\Vizrt
```

· Logfiles for the service component **Ticker Service** are stored at `%ProgramData%\Vizrt` in the directory *Ticker Service*, normally this would be `C:\ProgramData\vizrt\Ticker Service`.

> ⚠ **Note:** `%AppData%` and `%ProgramData%` are Windows environment variables. Windows Explorer resolves them into a folder path depending on operation system, system drive, current user and so on. For example, on Windows 7, `%ProgramData%\Vizrt` resolves to `C:\ProgramData\Vizrt` if Windows is installed on the C: drive.

See Also

- Ticker System Tool Log Files

4   Ticker Wizard

Ticker Wizard is the Vizrt Ticker configuration application.

> 🔴 **IMPORTANT!** Before making changes to the Ticker system configuration in Ticker Wizard, the Viz Ticker Client should be closed, if open. Ticker Wizard and Viz Ticker Client should not be used at the same time.

This section contains information on the following topics:

- Getting Started
- Main Window
- Main Menu
- Main Panel
- Ticker Carousels Panel
- Output Channels Panel
- GPI Panel
- Settings Panel
- Buttons Panel
- Wizards

4.1   Getting Started

> ⚠️ **Note:** If the Ticker System is currently *on*, it is highly recommended *not* to start Ticker Wizard, as the playout is likely to be disrupted.
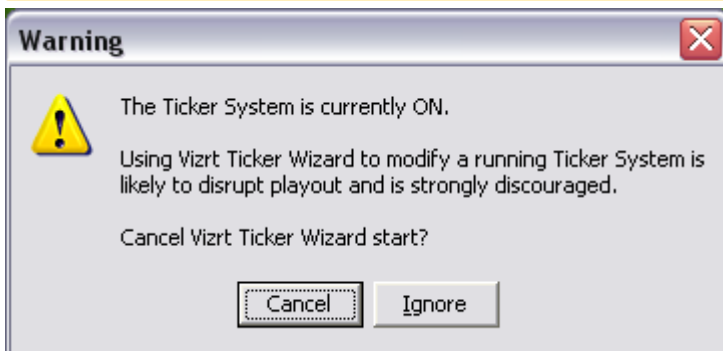
4.1.1   To Start Ticker Wizard

- Start Ticker Wizard from **Start > All Programs > Vizrt > Viz Ticker > Ticker Wizard**.

4.2   Main Window

At system startup, the Main Window appears on the screen.
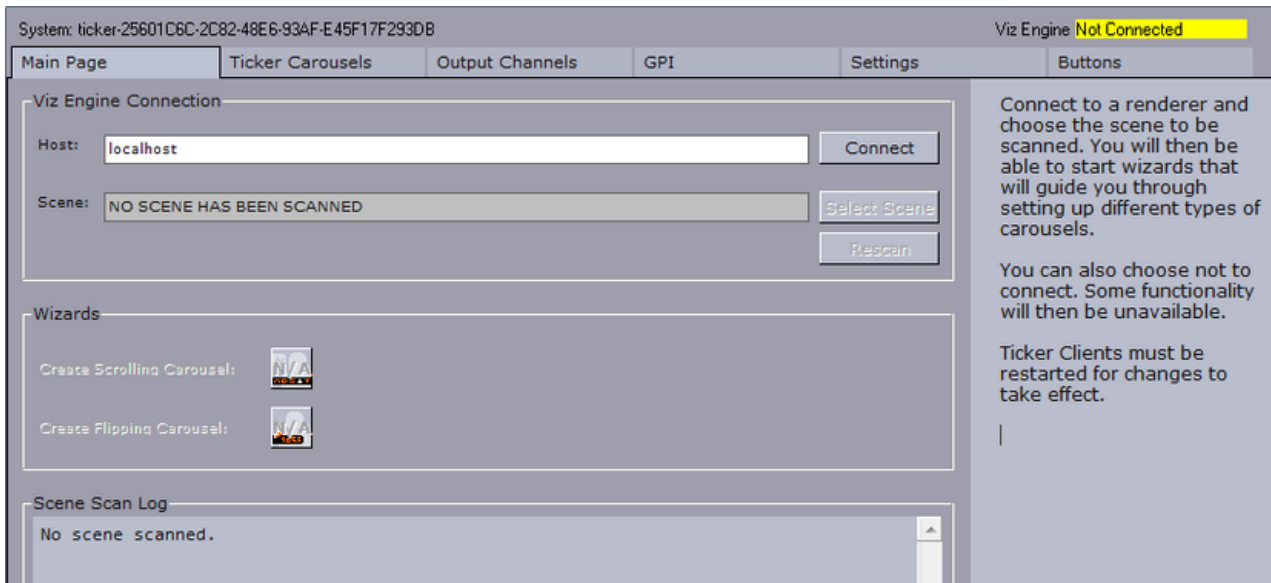
4.3   Main Menu

At the top of the Main Window is the Main Menu.

The Main Menu contains various tabs that open respective panels.

- **Main:** Opens the Main Panel.
- **Ticker Carousels:** Opens the Ticker Carousels Panel.
- **Output Channels:** Opens the Output Channels Panel.
- **GPI:** Opens the GPI Panel.
- **Settings:** Opens the Settings Panel.
- **Buttons:** Opens the Buttons Panel.

4.4   Main Panel

The Main Panel opens when clicking the **Main Page** tab in the Main Menu.
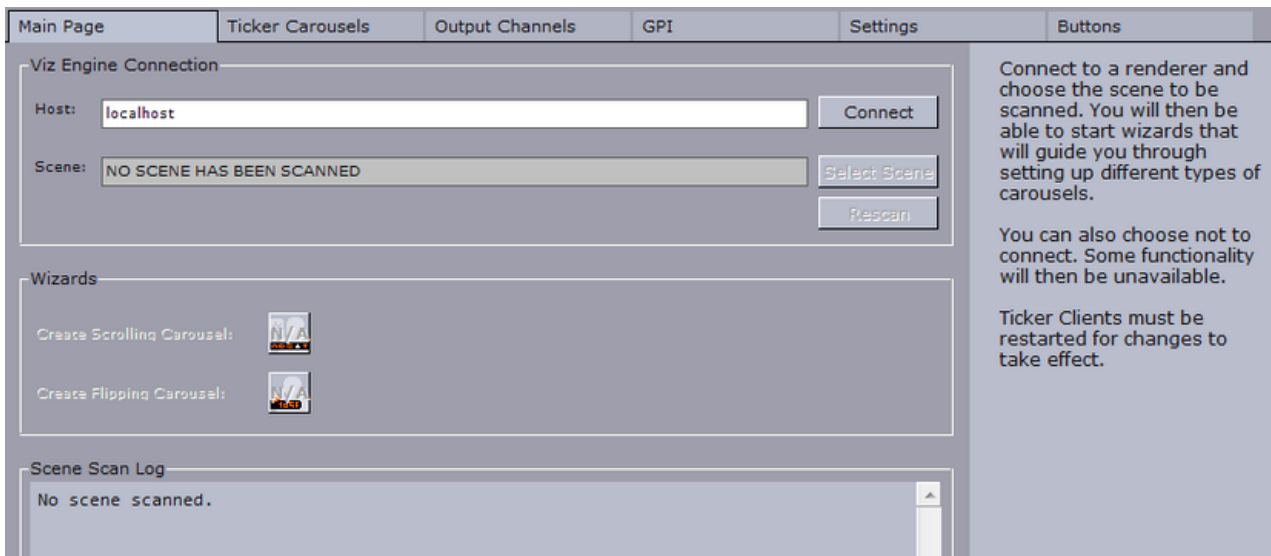
From the Main Panel it is possible to connect the system to a render engine, scan ticker scenes, and so on.

The Main Panel contains three frames:

- Viz Engine Connection Frame
- Wizards Frame
- Scene Scan Log Frame

4.4.1 Viz Engine Connection Frame

At the top of the Main Panel is the Viz Engine Connection frame, which contains various host and scene settings.

Viz Engine Connection Parameters

- **Host:** Defines the name of the Viz Engine host Ticker Wizard connects to. The hostname can be either a text string or an IP address.
- **Connect:** Connects Ticker Wizard to the defined Viz Engine host.

> ❗ **IMPORTANT!** Most Ticker Wizard options are not available until the system is successfully connected to a Viz Engine host.

a. Make sure that Viz Engine is running in *On Air* mode.

      b. Type the desired hostname in the **Host** text box.
      c. Click the **Connect** button.
      d. Check the Status of the Viz Engine Connection, see below.

- **Scene:** Defines the name of the currently scanned scene. When a scene has been successfully scanned, the scene name and path are presented in the text box.
- **Select Scene:** Allows you to select a scene from the Viz Engine.
- **Rescan:** Rescans the currently scanned scene and informs Ticker Wizard of changes that have occurred in the scene design since the last scanning.

Status of the Viz Engine Connection

Check the Viz Engine connection status in the upper right corner of the Main Window. The status can be one of the following:

Status	Icon	Description
Connected	Viz Engine Connected	Indicates that the Ticker System is successfully connected to a running Viz Engine in *On Air* mode.
Lost Connection	Viz Engine Lost Connection	Indicates that Viz Engine has been shut down correctly, and that the Ticker System is no longer connected to the Viz Engine.
Not Connected	Viz Engine Not Connected	Indicates that no attempt has been made to connect the Ticker System to a Viz Engine.
Not On Air	Viz Engine Not On Air	Indicates that the connected Viz Engine is not in *on-air* mode.
Timeout	Viz Engine Timeout	Indicates that the Ticker System is no longer connected to the specified Viz Engine. The connection has been broken for some unknown reason, without the Viz Engine being able to communicate the exact problem.
Unable To Connect	Viz Engine Unable To Connect	Indicates that the specified Viz Engine is not running.
Unknown Host: <hostname>	Viz Engine Unknown Host: MyHost	Indicates that the specified Viz Engine host cannot be found on the network.

To Scan a Scene

1. Make sure that Ticker Wizard is connected to a running Viz Engine host (See the Status of the Viz Engine Connection).
2. Make sure that any existing carousels or output channels are deleted (See Troubleshooting).

3. Click the **Select Scene** button.
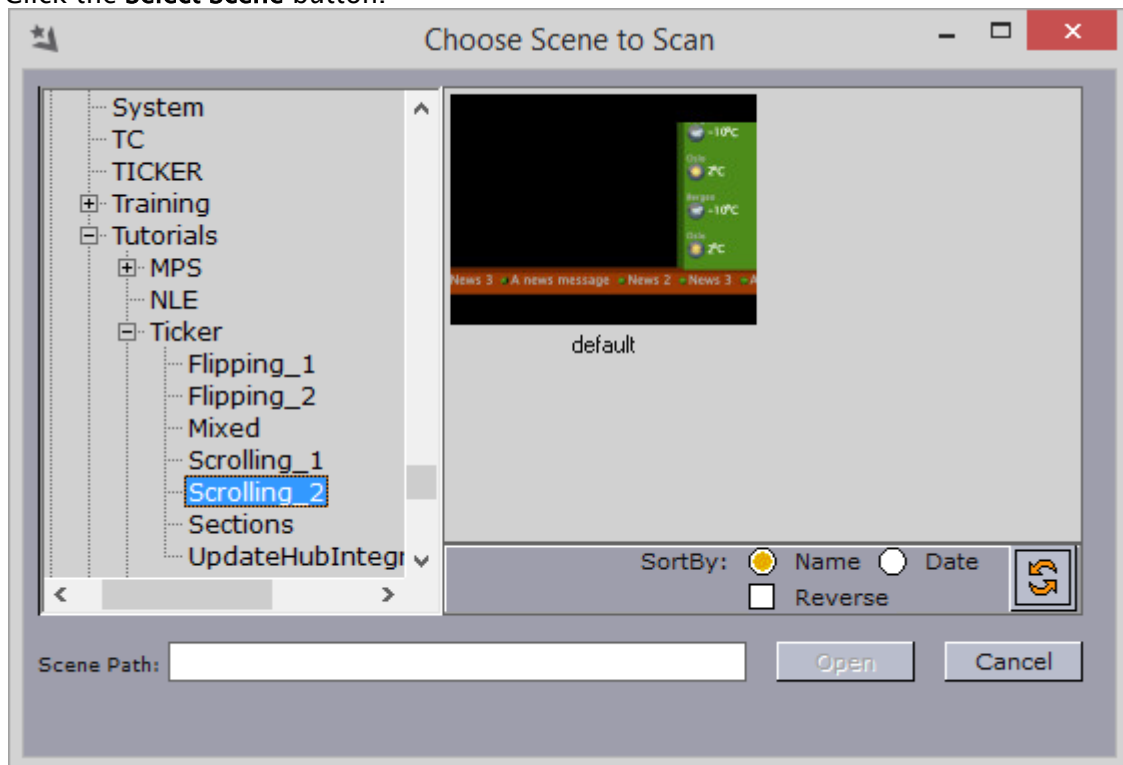
4. In the window that opens, double-click the thumbnail of the desired scene.

> ✅ **Tip:** In this window, scenes can be sorted by name or date, and presented in regular or reversed order.

4.4.2   Wizards Frame

In the center of the .Main Panel v4.0 is the Wizards Frame.

The Wizards Frame contains two buttons that open wizards for creating scrolling and flipping carousels.

- **Create Scrolling Carousel:** Opens the Scrolling Carousel Wizard.
- **Create Flipping Carousel:** Opens the Flipping Carousel Wizard.

> ⚠ **Note:** The buttons become available once respective carousel scenes have been successfully scanned.

4.4.3 Scene Scan Log Frame

At the bottom of the .Main Panel v4.0 is the Scene Scan Log Frame.
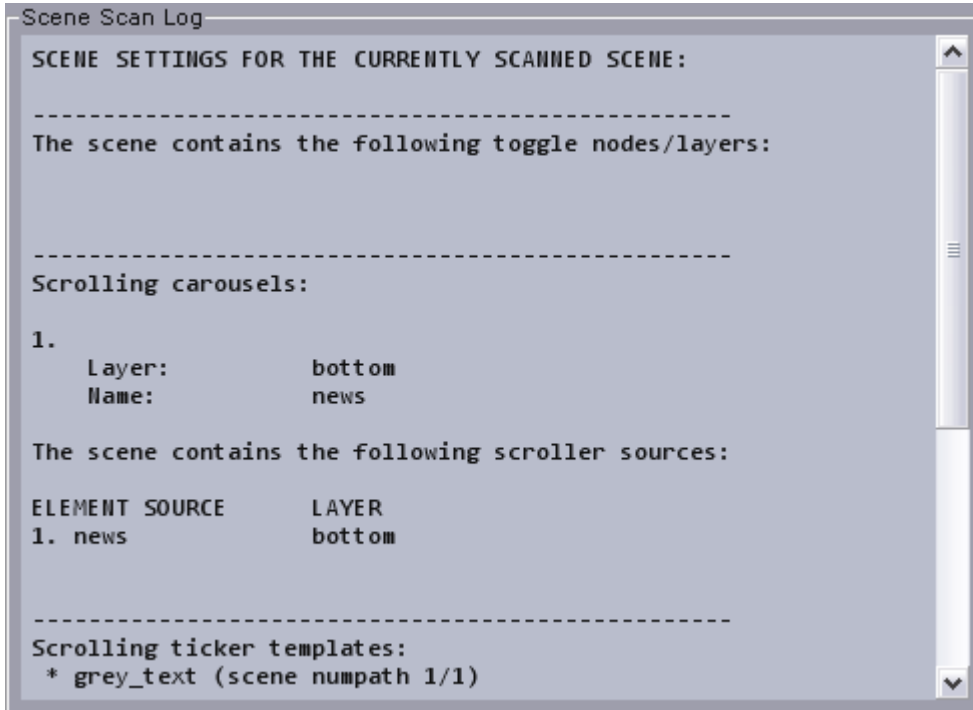
This shows an overview of the s, carousels, and templates Ticker Wizard identifies when scanning the currently selected scene.

4.5 Ticker Carousels Panel

The Ticker Carousels Panel opens when clicking the **Ticker Carousels** tab in the Main Menu.
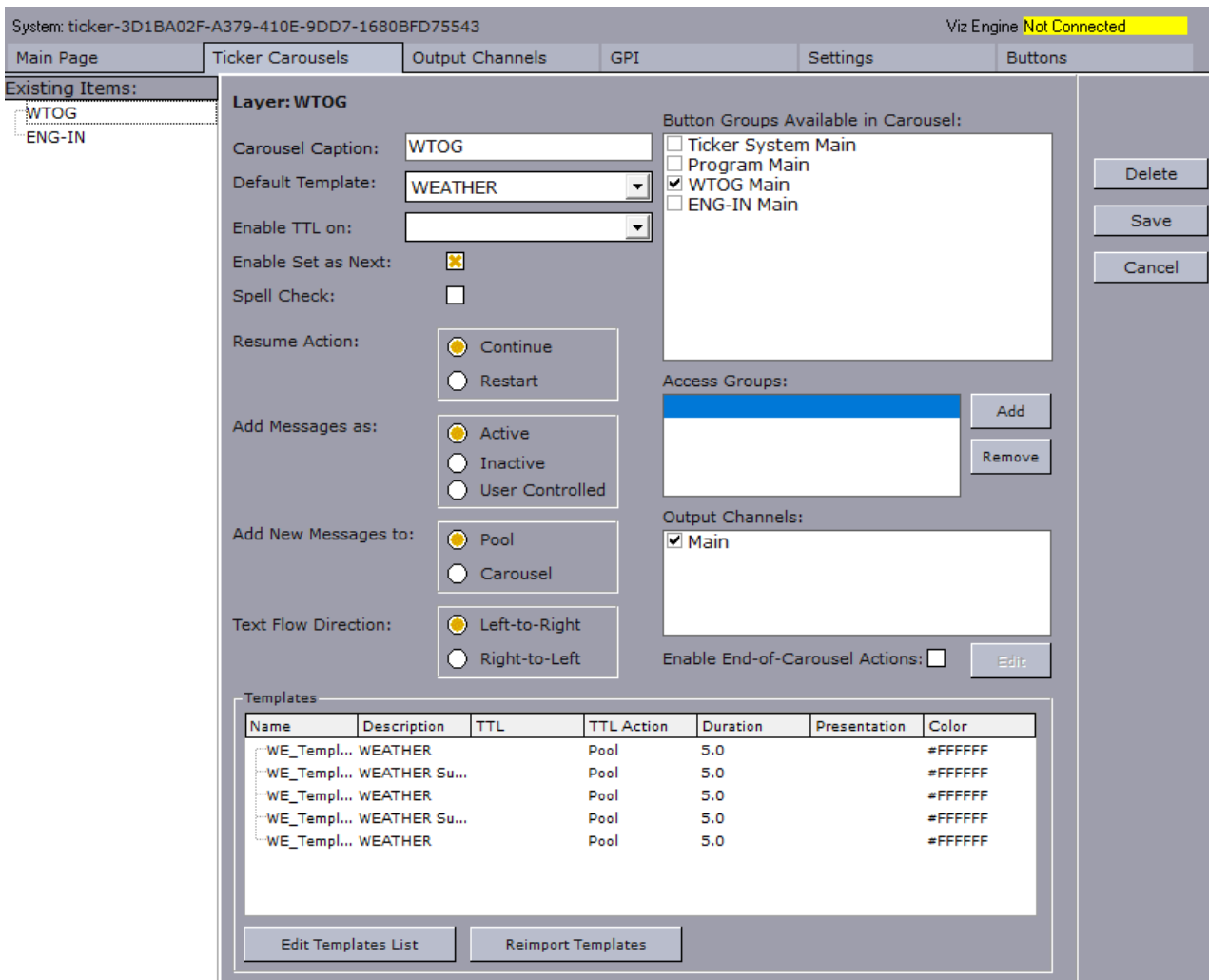
This section contains information on the following topics:

- Existing Carousels List
- Carousel Details Frame

4.5.1 Existing Carousels List

On the left side of the .Ticker Carousels Panel v4.0 is the Existing Carousels List.
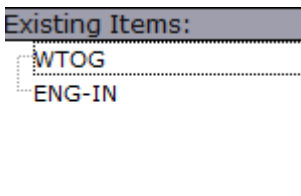
The Existing Carousels List contains the names of all available carousels. Carousels become available once they have been created in one of the Wizards.

To view the details for a specific carousel in the Carousel Details Frame, double-click an item in the list.

To Delete a Carousel

1. Double click a carousel in the Existing Carousels List.
2. Click the **Delete** button.

4.5.2   Carousel Details Frame

The Carousel Details Frame is in the center of the Ticker Carousels Panel.

The Carousel Details Frame shows various information about the carousel that is currently selected in the Existing Carousels List. The settings in this frame affect how the carousel behaves in Viz Ticker Client.

- **Layer:** Reflects the name of the that holds the stop points in the Viz Artist scene.
- **State:** Reflects the name of the active *in* state. This label is only available for scrolling carousels.
- **Carousel Caption:** Defines the carousel specific tab name in Viz Ticker Client. The caption is originally defined when setting up the carousel in one of the carousel wizards, see Carousel Panel.

> ⚠ **Note:** The carousel caption should contain alphanumeric characters only. For more details, see Global Conventions.

- **Default Template:** Defines the template that is used as default when clicking the **New Message** button in the Carousel Specific Panel in Viz Ticker Client.
- **Enable TTL on:** Defines an output channel that displays the two *Time To Live* frames in Viz Ticker Client's Elements Editor and makes the *TTL* column available in the Carousel Frame. For more information, see Settings Frame and Carousel List.

> ⚠ **Note:** It is only possible to enable TTL on one output channel at a time. In order to enable an output channel, the channel must first be activated in the Output Channels Frame.

- **Enable Set as Next:** Makes the *Set as Next* feature available in Viz Ticker Client when right-clicking a message in the Carousel List, if enabled.
- **Spell Check:** Makes the Spell Check checkbox selected by default when creating a new message in Viz Ticker Client, if enabled. This feature implies that the message text is spell checked.
- **Resume Action:** Defines what should happen to the carousel when resuming from a break.
    - **Continue:** Continues where the system left off when resuming, if enabled.
    - **Restart:** Restarts the system with the first message in the carousel when resuming from the break, if enabled.
- **Add Messages as:** Refers to the message status when messages are added to the Carousel List in Viz Ticker Client.
    - **Active:** Messages are added as active on all output channels.
    - **Inactive:** Messages are added as inactive on all output channels.
    - **User Controlled:** Adds the Default Output Channel Activation Frame in Viz Ticker Client, where the default activation status on each available output channel can be defined. For more information, see Activating Messages.
- **Add New Messages to:** Refers to where messages should be placed after creation in Viz Ticker Client.
    - **Pool:** Adds new messages to the Pool List.
    - **Carousel:** Adds new messages to the Carousel List.
- **Text Flow Direction:** Configures the flow direction of text in a carousel. It applies to all templates in the carousel and all tab-fields under those templates.
    - **Left-to-Right** (default): Text flows from left to right.
    - **Right-to-Left:** Text flows from right to left.

Button Groups Available in Carousel Frame

This frame lists all available button groups.
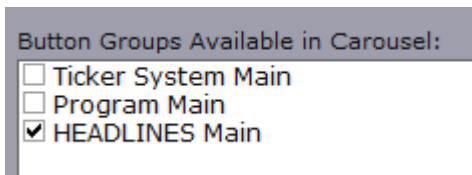
The items in this list reflect the items in the Button Groups and Button List in the Buttons Panel.

Button groups that are selected in this list are shown in the Buttons Frame in Viz Ticker Client's Carousel Specific Panel.

Access Groups Frame

The Access Groups Frame lists all groups that should have access to this carousel. Only clients that have defined the right group name in its shortcut properties have access to such a restricted carousel.
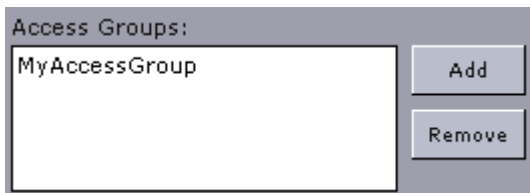
To set access restrictions to a carousel

1. In the Access Groups Frame's text field, type a descriptive group name, for example *MyAccessGroup*, and then press **ENTER**.
2. To add another group, click the **Add** button.
3. Type the name of the second group in the text field, for example *AnotherAccessGroup*.
4. Change the Viz Ticker Client Startup Parameters, so that the target access group matches one of the groups defined here.

> ⓘ **Example:** *-accessgroup MyAccessGroup*

Output Channels Frame

The Output Channels Frame lists all available output channels.
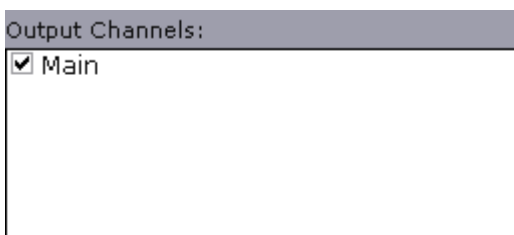
Output channels that are selected in this list are available in Viz Ticker Client. For example, belonging *On/Off* buttons appear in the Actions Frame, On/Off buttons can appear in the Buttons Frame (if the belonging button group is selected in the Button Groups Available in Carousel Frame), and columns are created in the Carousel Frame.

End of Carousel Actions
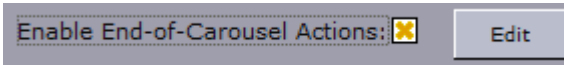
Check this option to enable one or more Ticker actions to be executed when this carousel reaches its end. Clicking Edit will open the End of Carousel actions list:
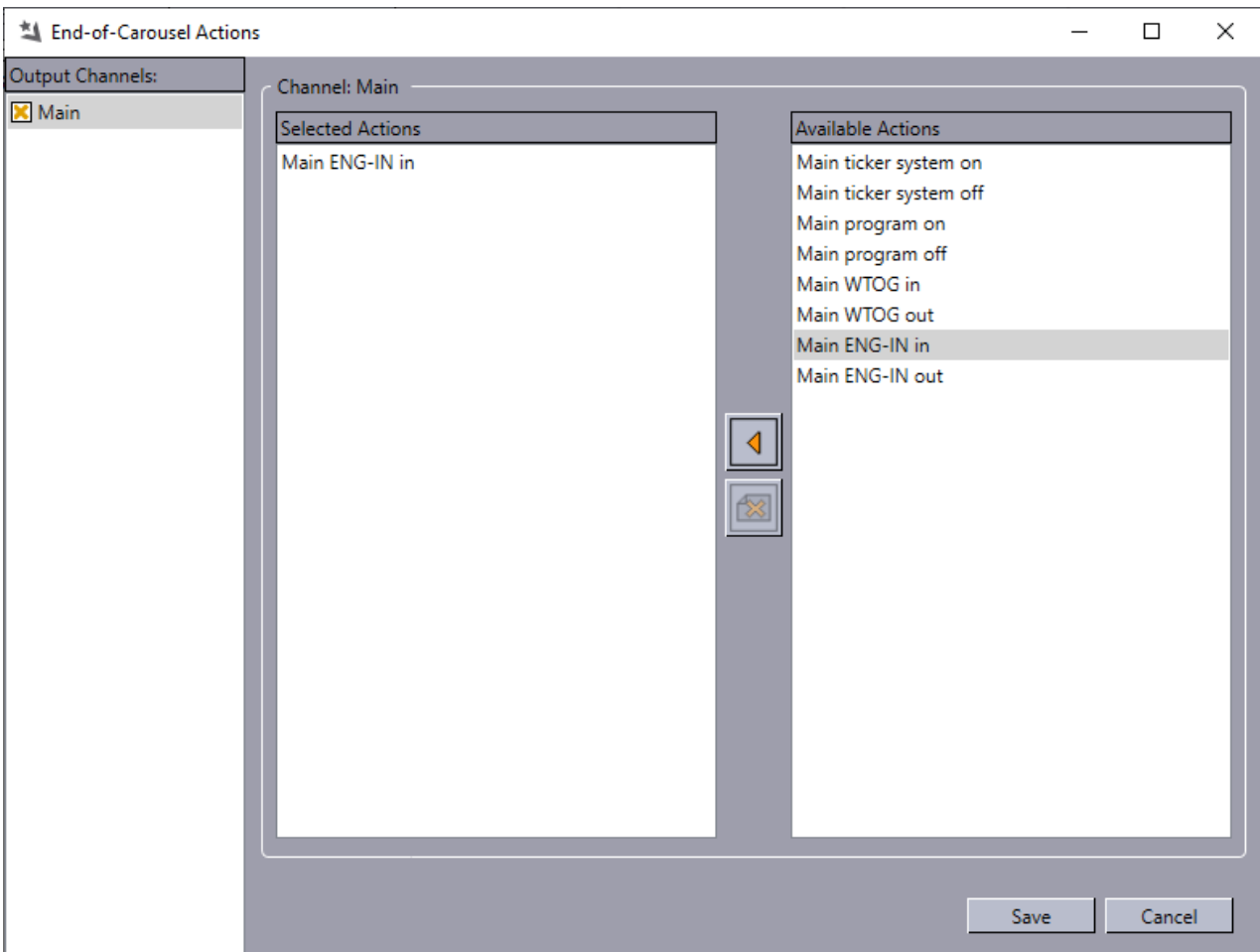
For the current carousel, each Output Channel can have a set of defined End of Carousel actions. The available actions are listed on the right side, while the active actions are listed on the left. Whenever the current carousel on the selected output channel reaches it end, the list of Selected Actions will be executed. This can for instance be used to chain carousels together or trigger GPI actions.

Templates Frame

At the bottom of the Carousel Details Frame is the Templates frame.

The Templates frame contains a list of all templates that are available for this carousel. Templates are added when setting up a carousel in one of the carousel Wizards. Each template is presented with the following information:

- **Name:** Shows the name of the template.
- **Description:** Shows the text that is displayed in the *Template* column in Viz Ticker Client's Carousel Frame.
- **TTL:** Shows the TTL (Time To Live) value. TTL refers to how many times a message should run before it is taken out of the carousel.
- **TTL Action:** Shows what should happen to the message when the TTL value is reached.
- **Presentation:** Shows the name and path of the presentation file that can be used to create and manage messages in Viz Ticker Client. To use the default editor, leave the text field empty. For more information, see Presentation Files.
- **Color:** Displays all messages based on this template in Viz Ticker Client's Carousel List with the selected color. For a more detailed explanation of these settings, see Scrolling Templates Panel. Right-clicking a template in the list, opens a menu, which makes it is possible to add Field Linking and Feed Browsing functionality to the template, so that content can be fetched from external sources.

The Templates frame also contains two buttons:

- **Edit Templates List:** Opens the Scrolling Carousel Wizard or Flipping Carousel Wizard, where it is possible to modify the template settings.
- **Reimport Templates:** Opens the Reimport Templates Wizard where it is possible to reimport templates that have changed scene design. For more information, see To reimport templates.

4.6   Output Channels Panel

The Output Channels Panel opens when clicking the **Output Channels** tab in the Main Menu.

> ✅ **Tip:** For information about how to create a new output channel, see Output Channel Wizard and To create an output channel.

4.6.1 Existing Output Channels List

On the left side of the Output Channels Panel is the Existing Output Channels List.
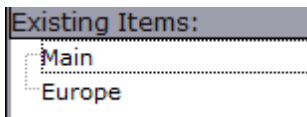
The Existing Output Channels List contains the names of all available output channels. Output channels become available once they have been created in the Output Channel Wizard. To view the details for a specific output channel in the Output Channel Details Frame, double-click an item in the list.

> ⚠ **Note:** The *Main* output channel is automatically generated when setting up a carousel in one of the carousel Wizards. The *Main* output channel is by default based on the Viz Engine *localhost*.

To Delete an Output Channel

1. Double click an output channel in the Existing Output Channels List.
2. Click the **Delete** button.

4.6.2 Output Channel Details Frame

In the center of the Output Channels Panel is the Output Channel Details Frame.

The Output Channel Details Frame shows information about the output channel that is currently selected in the Existing Output Channels List.

· **Channel Identifier:** Shows the name of the output channel.

> ⚠ **Note:** It is not possible to modify the channel identifier (name) of an output channel.

· **Renderer Hostname:** Shows the hostname of the Viz Engine that the Ticker System connects to when using this particular output channel.
· **Renderer Port:** Shows the port number of the Viz Engine that the Ticker System connects to when using this particular output channel.

- **Enable OneBox:** Enables multiple output channels from the same Ticker system. Should be enabled if this output channel is a part of a dual/quad Viz Engine running multiple output channels on the same host.

4.7   GPI Panel

The GPI Panel opens when clicking the **GPI** tab in the Main Menu.

GPI units register electrical tension (voltage) in one end as *high* or *low*, and output a digital signal (*on* or *off*) in the other end. This is used to connect control desks and similar devices to arbitrary software. Various Vizrt software, including Viz Ticker, can be controlled by GPI through the Media Sequencer.

This section contains information on the following topics:

- Existing GPI Handlers List
- GPI Settings Frame
- GPI List

4.7.1   Existing GPI Handlers List

On the left side of the GPI Panel is the Existing GPI Handlers List.
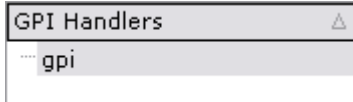
The Existing GPI Handlers List contains the names of all available GPI handlers. To view the details for a specific GPI handler in the GPI Settings Frame and GPI List, double-click an item in the list.

To Create a GPI Handler

1. In Ticker Wizard, click the **GPI** tab in the Main Menu.
2. Right-click the handler that should be deleted in the Existing GPI Handlers List. A new GPI handler is then created and added to the Existing GPI Handlers List.

To Delete a GPI Handler

1. In Ticker Wizard, click the **GPI** tab in the Main Menu.
2. Select the handler that should be deleted in the Existing GPI Handlers List.
3. Right-click the Existing GPI Handlers List.
4. In the menu that appears, click **Delete Selected GPI Handler**. The selected GPI handler is then deleted and removed from the Existing GPI Handlers List.

4.7.2   GPI Settings Frame

At the right side of the GPI Panel is the GPI Settings Frame.
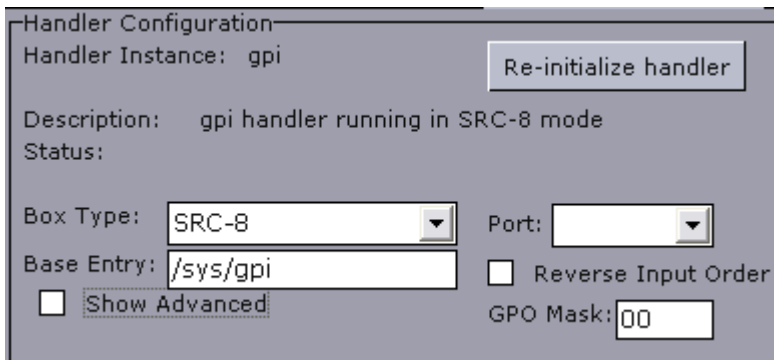
The GPI Settings Frame shows various information about the GPI handler that is currently selected in the Existing GPI Handlers List.

- **Handler Instance:** Shows the name of the GPI handler (equal to the name in the Existing GPI Handlers List).
- **Re-initialize Handler:** Re-initializes the GPI handler.
- **Description:** Shows a description of the GPI handler.
- **Status:** Shows the status of the GPI handler.
- **Box Type:** Defines the type of GPI box that is being configured. The box type can be set to *SRC-8*, *SRC-8 III*, *SeaLevel*, or *SeaLevelMAX*.

- **Port:** Defines the port that the GPI box is connected to. The port can be set to *COM1-COM8*, or none.
- **Base Entry:** This is the node in the Media Sequencer's data structure where the systems look for the GPI actions. The base entry is by default set to */sys/gpi*.
- **Reverse Input Order:** If enabled, the signal line that originally triggered GPI action *DL0/DH0* now triggers GPI action *DL7/DH7*, the signal line that originally triggered GPI action *DL1/DH1* now triggers GPI action *DL6/DH6*, and so on. This option is only available if the box type is set to *SRC-8*.
- **Use Card:** Defines the GPI card that should be used. This option is only available if the box type is set to *SeaLevel*.
- **IP:** Defines the IP address of the Ethernet box. This option is only available if the box type is set to *SeaLevelMAX*.
- **Unit:** Defines the available Ethernet boxes. This option is only available if the box type is set to *SeaLevelMAX*.
  - **Unit:** Defines the unit number of the Ethernet box.
  - **Slave Id:** Defines the Slave ID of the Ethernet box.

Advanced GPI Settings

The Advanced GPI Settings become available when **Show Advanced** is selected.

- **Verbose:** Outputs log information from the Media Sequencer's GPI handler, if enabled. This information can be useful for debugging.
- **Data Bits:** Defines the number of data bits that should be used for each transmitted and received character. The communicating computer or device must have the same setting. The number of databits can be set between 4 and 8.
- **Stop Bits:** Sets the interval (bps) for when characters should be transmitted. Stop bits can be set to 1, 1.5, or 2.

- **Parity:** Changes the type of error checking that is used for the selected port. The communicating computer or device must have the same setting. The parity can be set to:
    - **Even:** May add a parity bit to make the number of ones in the data bits even. This enables error checking.
    - **Odd:** May add a parity bit to make the number of ones in the data bits odd. This enables error checking.
    - **None:** Does not add a parity bit to the data bits sent from this port. This disables error checking.
    - **Mark:** Adds a parity bit set to `0`.
    - **Space:** Adds a parity bit set to `1`.
- **Baud Rate:** Defines the maximum rate of bits per second (bps) that should be transmitted through this port. The baud rate can be set between `75` and `128.000`. It is recommended to use the highest rate that is supported by the computer or device that is being used.
- **Flow Control:** Changes how the flow of data is controlled. The flow control can be set to:
    - **XonXoff:** Uses the standard method of controlling the flow of data between two modems. This type of flow control is often referred to as *software handshaking*.
    - **Hardware:** Uses the standard method of controlling the flow of data between a computer and a serial device. This type of flow control is often referred to as *hardware handshaking*.
    - **None:** Does not use a method of controlling dataflow.
- **Sample Interval (microseconds):** Defines how often the PCI card should be checked for updates. The interval is defined in microseconds. This option is only available if the box type is set to *SeaLevel*.
- **Poll Delay:** Defines how often the Ethernet box should be checked for updates. The interval is defined in microseconds. This option is only available if the box type is set to *SeaLevelMAX*.

4.7.3   GPI List

At the bottom of the GPI Panel is the GPI List.

GPI Handler	GPI Line	Server Command	Description
gpi	DH0		
gpi	DL0		
gpi	DH1		
gpi	DL1		
gpi	DH2		
gpi	DL2		
gpi	DH3		
gpi	DL3		
gpi	DH4		
gpi	DL4		
gpi	DH5		
gpi	DL5		
gpi	DH6		
gpi	DL6		
gpi	DH7		
gpi	DL7		

The GPI List contains a list of all available GPI handlers and signal lines.

- **GPI Handler:** Shows the GPI handler that the currently selected GPI line belongs to.
- **GPI Line:** Shows the name of the GPI line.
- **Server Command:** Shows a text box in every row, where the action to be performed on this GPI line can be defined.

> ⓘ **Example:** If the server command *Program Main* is set to be the GPI action of *DL0/DH0*, *DL0* (low line 0) sets Program to `off` for the main feed, and *DH0* (high line 0) sets Program to `on`.

- **Description:** Shows a description of the GPI line action.

4.8   Settings Panel

The Settings Panel opens when clicking the **Settings** tab in the Main Menu.

The features in this panel are related to settings in Viz Ticker Client.

> ⚠ **Note:** Viz Ticker Client must be restarted before the changes in this panel take effect.

This section contains information on the following topics:

4.8.1   Ticker System Information Frame

At the top of the Settings Panel is the **Ticker System Information** frame, which associates the Ticker system with an identifier, name and description.

- · **System ID:** This identifier is automatically generated by Ticker Wizard. It cannot be modified.
- · **Name:** The name for the ticker system is automatically generated by Ticker Wizard. If preferred, it can be modified.

> ⚠ **Note:** Ticker Wizard must be restarted for the name changes to take effect.

- · **Description:** Optional descriptive details for the ticker system can be added, but is not required.

4.8.2   Graphic Hub Configuration Frame

In the middle of the Settings Panel is the **Graphic Hub Configuration** frame, which allows you to set the service endpoint, and eliminates the need for Viz Engine to be on air when using the Graphic Hub browser in the Viz Ticker Client.

> ⚠ **Note:** The Service Document URL is optional since only some Viz Ticker functionality requires it. It is required for storing Image Search Providers on Graphic Hub, or for using the standard Elements Editor with image fields.

> 🛑 **IMPORTANT!** It is recommended that the Graphic Hub Rest configured here is the same as the one configured in your Viz Engine. If different Graphic Hubs are configured (one here and another in the Viz Engine) then they must have the same data structure. If the structure is not the same. then Viz Ticker Client may fail to fetch the image or may fetch an incorrect image.

- · **Service Document URL:** Enter Service Document URL for the Graphic Hub REST (e.g. http://vgh1:19398).
The **authentication** is on a per-user basis, where the credentials are securely stored for the logged-in user in roaming user-data. Before prompting the user for credentials, Ticker attempts to authenticate with previously stored credentials and/or default account credentials.

> ⚠ **Note:** If Viz Multiplay is in use, then any changes made to the service document setting in Ticker Wizard are reflected in the settings of Viz Multiplay, and vice versa.

4.8.3   Show in Main Control Frame

In the middle of the Settings Panel is the **Show in Main Control** frame.
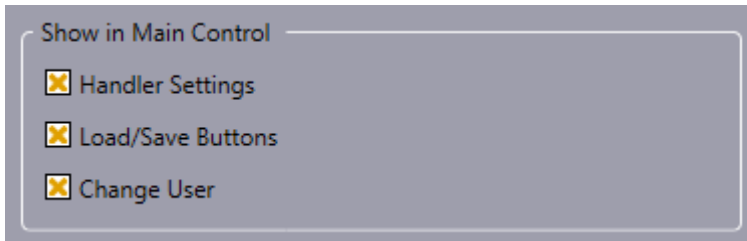
- **Handler Settings:** Makes the Handler Settings Frame visible in Viz Ticker Client's Control Panel.
- **Load/Save Buttons:** Makes the **Load** and **Save** buttons visible in Viz Ticker Client's Data Frame.
- **Change User:** Makes the **Change User** button visible in Viz Ticker Client's Data Frame.

> ⚠ **Note:** If both the **Load/Save Buttons** and **Change User** check boxes are cleared, the Data Frame in Viz Ticker Client's Control Panel is unavailable.

4.8.4   End of Carousel Actions

Enable or disable the a context menu item in the carousel playlist, deciding whether the user of the Ticker client should be able to turn on and off the End-Of-Carousel actions in the Ticker Client.

4.8.5   Enable/Disable Features Frame

In the middle of the Settings Panel is the **Enable/Disable Features** frame.
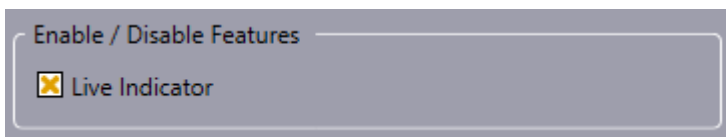
- **Live Indicator:** Enables the Live Indicator functionality, which makes it possible to see in Viz Ticker Client's status bar whether the carousel is live On Air.

To Enable the Live Indicator

1. Select the **Live Indicator** checkbox and click **Save**. The Live Indicator in Viz Ticker Client then becomes available from the status bar (lower left corner of Viz Ticker Client window).

Warning: This Ticker System is not live.

4.8.6 Image Search Provider Frame

Graphic Hub and Pilot Data Server can be used to store the Vizrt Asset Search providers. These search providers are used when doing a search for VTW Image component images (see Presentation Files for Image Components). This allows you to search and select images from configured search providers such as VOS, Viz One, or any other compatible search provider.

> ⚠ **Note:** This requires the Graphic Hub REST API (Graphic Hub 2.0 or later).

> ✅ **Tip:** Once Image Search is configured, the Media Search window opens in Viz Ticker Client whenever a relevant message is being edited. See Adding an Image using Media Search.

The search providers can be specified in either of two ways:

- Search Providers on Graphic Hub
- Search Providers on Pilot Data Server

Search Providers on Graphic Hub

Graphic Hub Configuration
Service Document URL: http://vgh1:19398

Show in Main Control
☐ Handler Settings
☐ Load/Save Buttons
☐ Change User

Enable / Disable Features
☐ Live Indicator

Image Search Provider
Select the location of image search providers:
⦿ Viz Graphic Hub (default) ◯ Pilot Data Server
[Edit Search Providers]

Search providers and pre-authentication information can be specified explicitly in Ticker Wizard. The search providers persist in Graphic Hub. This works independently of Pilot Data Server.

- To use **Viz Graphic Hub** for search provider persistence:

- In the Graphic Hub Configuration Frame, enter the **Service Document URL** for the Graphic Hub REST API.
- In the Image Search Provider frame, select **Viz Graphic Hub.**
- To **configure the Service Providers**, click the **Edit Search Providers** button. The Search Providers window opens:

- Add a new item by clicking under the Asset Search Providers or Pre-Authenticated Hosts tables. *Service Document URL* is mandatory for Asset Search Provider items, and both *Host URL* and *User Name* are mandatory for Pre-Authenticated Hosts items.
- Edit an existing item by double clicking a cell in the table. Note that the *Description* is provided by the server and cannot be edited.
- Delete an existing item by pressing **Delete**, or by right clicking on a row handle (leftmost part of the row) and selecting Delete Row.

Search Providers on Pilot Data Server

In this case, search providers are retrieved from a Pilot Data Server (PDS) instance. This is convenient for users that have already configured a Viz Pilot setup, as they can access the same search providers but only have to maintain it in one place (in the PDS). The search providers must be configured on **Pilot Data Server's Settings > Search Providers** page.

- To use **Pilot Data Server** for search provider persistence:
  - In the Image Search Provider frame, select **Pilot Data Server.**

- Specify the Service Document URL in the **Pilot Data Server URL** field (for example, http://vcp1:8177/service):
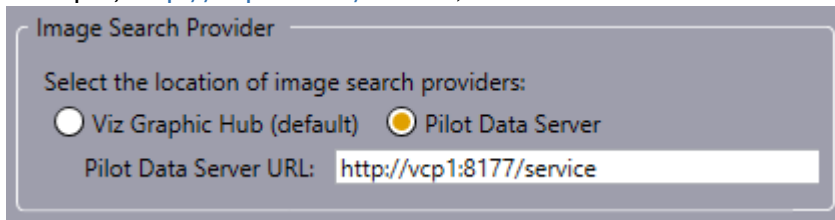
- To **edit the search providers** in Pilot Data Server:
    - Open a web browser that supports Silverlight.
    - Go to the Settings page (for example, http://vcp1:8177/settings#Search).

> ⚠ **Note:** Changes made to the search provider information in PDS via a web browser do not take effect until Viz Ticker Client is restarted.

For more information see the Search Providers section in the Viz Pilot User Guide.

4.9   Buttons Panel

The Buttons Panel opens when clicking the **Buttons** tab in the Main Menu.
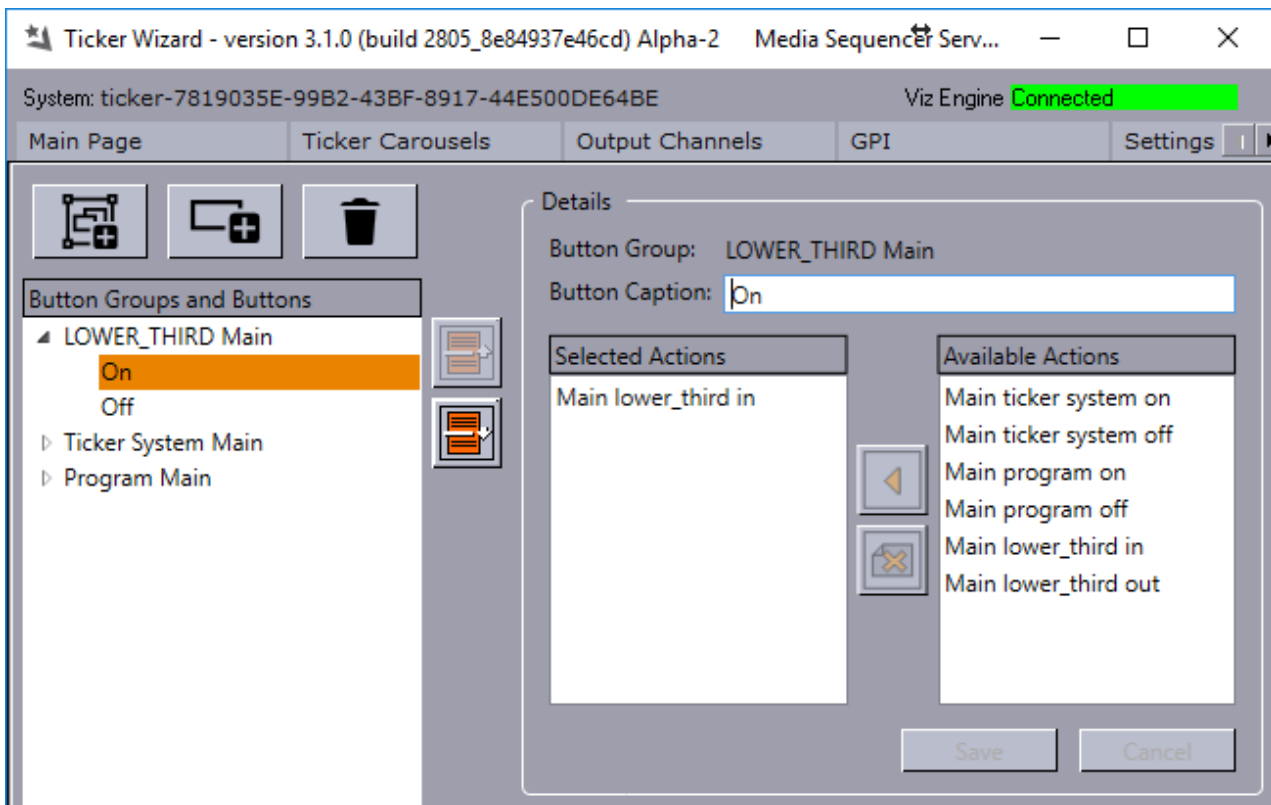
In the Buttons Panel, it is possible to manage button groups. Button groups are available in the Actions Frame in Viz Ticker Client's Control Panel. It is also possible to add button groups to the carousel specific panels, see Button Groups Available in Carousel Frame.

This section contains information on the following topics:

- To Add or Remove Buttons and Button Groups
- Button Groups and Buttons List
- Button Group/Button Details Frame

4.9.1   To Add or Remove Buttons and Button Groups
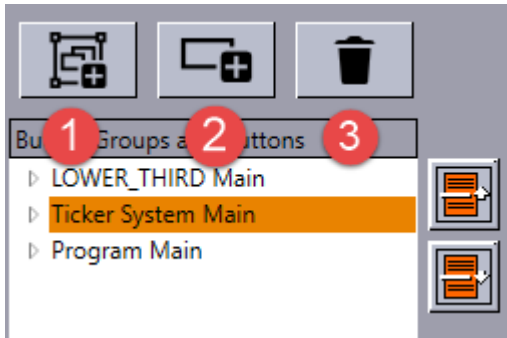
1. Add a new button group.
2. Add a new button.
3. Delete selected button or button group.

4.9.2   Button Groups and Buttons List

At the left side of the .Buttons Panel v3.1 is the Button Groups and Buttons List.
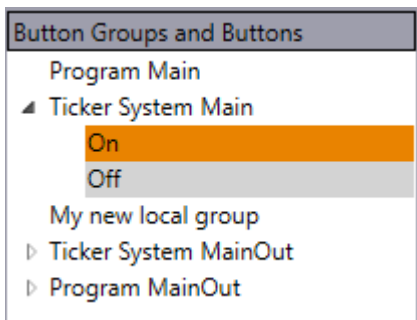
- Orange background for the current item in navigation.
- Gray background for the selected (or opened) item whose details are displayed in Button Group/Button Details Frame.

The Button Groups and Buttons List contain the names of all available button groups with belonging buttons. For each output channel, there are button groups called *Ticker System* and *Program*. For each carousel, there is also one button group per output channel.

When working with the Button and Button Groups list the following keyboard shortcuts and mouse options are available:

Key	Action
ENTER (on selected item)	Selects the item in the list and shows the details in Button Group/Button Details Frame.
UP ARROW	Moves to item above.
DOWN ARROW	Moves to item below.
RIGHT ARROW or **Numpad plus (+) or (*)**	Expands a button group.
LEFT ARROW or Numpad minus (-)	Collapses a button group.
HOME / END	Focuses (Moves to) First /Last item.
F2	Enables renaming of item. Press Enter after making changes to save or press **ESC** to cancel changes.

If you prefer to use the mouse, the button group list supports drag-and-drop, which provides an easy way to rearrange the items:

- Move a button group to the location of another button group.
- Move a button to the location of another button.
- Move a button to another button group. In this case, the button's parent group changes.
- Right-click on any item in the list for a context menu providing these options:
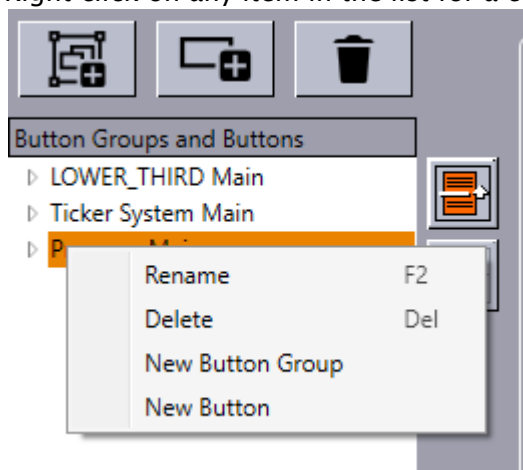
To view the details of a button/button group, double-click an item in the list or press **Enter** on the selected item. Detailed information appears in the Button Group/Button Details Frame.

To move a button/button group up or down in the list, click the arrow buttons positioned at the right side of the list.

4.9.3 Button Group/Button Details Frame

In the center of the Buttons panel is the Button Group/Button Details Frame.
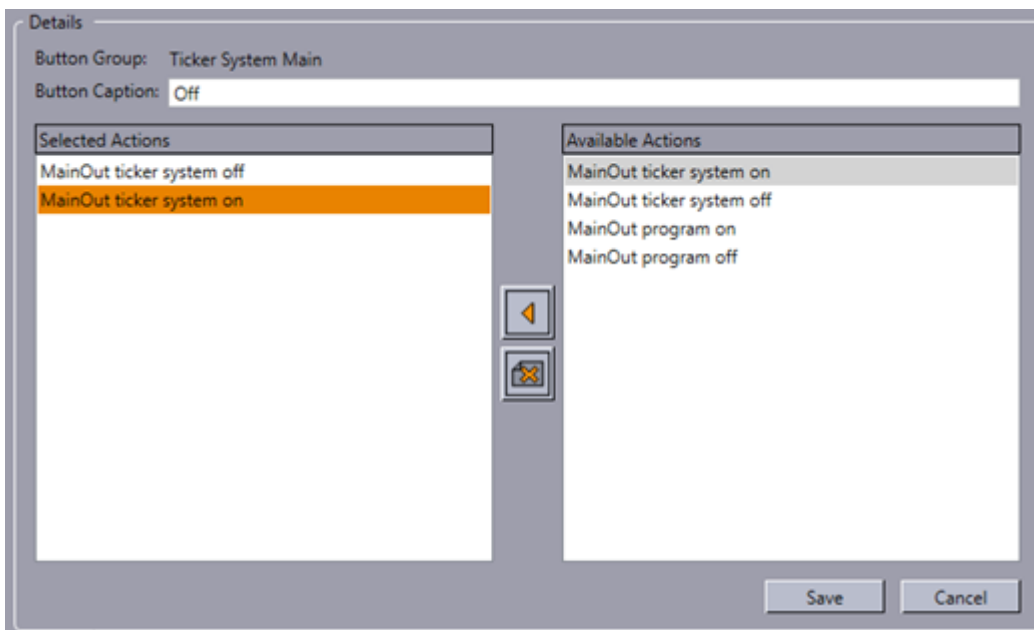
- · **Button Group:** Defines the name of the button group. This option can only be modified if a button group is selected in the Button Groups and Buttons List.
- · **Button Caption:** Defines the caption that appears on the button in Viz Ticker Client. If a button group is selected in the Button Groups and Buttons List, the **Button Group** text box is the only option that is displayed in this frame. If a button is selected in the Button Groups and Buttons List, the following options are available:

    - · **Add Selected Action Button:** Adds the action that is selected in the **Available Actions** list to the selected button.

    - · **Remove Action Button:** Removes the action that is selected in the **Selected Actions** list from the selected button.

    Note that multiple identical actions connected to one button may cause errors and is therefore not recommended. If you add an identical action, this warning dialog is presented:
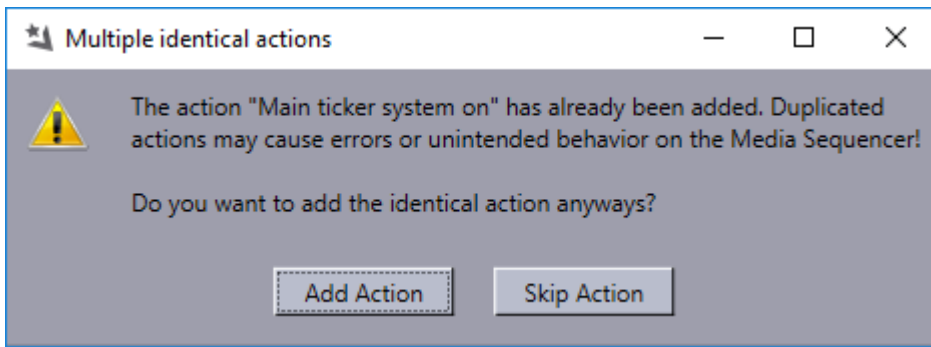
4.10   Wizards

Various wizards are available in Ticker Wizard.

This section contains information on the following topics:

- Scrolling Carousel Wizard
- Flipping Carousel Wizard
- Reimport Templates Wizard
- Output Channel Wizard

> ⚠ **Note:** These sections describe how to use the Carousel Wizard to create a new carousel. Existing carousels can be edited in the Ticker Carousels Panel.

4.10.1   Scrolling Carousel Wizard

Scrolling carousel scenes are characterized by containing one or more Scroller plug-ins with the relevant directors and stop points. The elements in a scrolling ticker enter the screen from one side, scroll through the screen, and exit on the other side.

> ⚠ **Note:** This section describes using the Carousel Wizard to create a new carousel. Existing carousels can be edited in the Ticker Carousels Panel.

Using the Scrolling Carousel Wizard

1. To open the Scrolling Carousel Wizard, click the **Create Scrolling Carousel** button in the Main Panel.
2. The wizard opens and you see the Carousel Panel.
   a. Give the Carousel a name in the **Carousel Caption** box.

   > ⚠ **Note:** The carousel caption should contain alphanumeric characters only. For more details, see Global Conventions.

   b. Double click a carousel in the **Available Carousels** list.

3. The wizard then proceeds to the Scrolling Templates Panel.
    a. Select one or more templates from the **Available Templates** list.
    b. *Optional:* Add Presentation Files. (See To select the presentation file for a template).
    c. *Optional:* Select a color for the messages. (See To select the background color for a template).
    d. Click **Next**.
4. The wizard proceeds to the Confirm Panel.
    a. Make sure that the setup information is correct.
    b. Click **Finish**. This exits the wizard and create the new carousel.

Carousel Panel
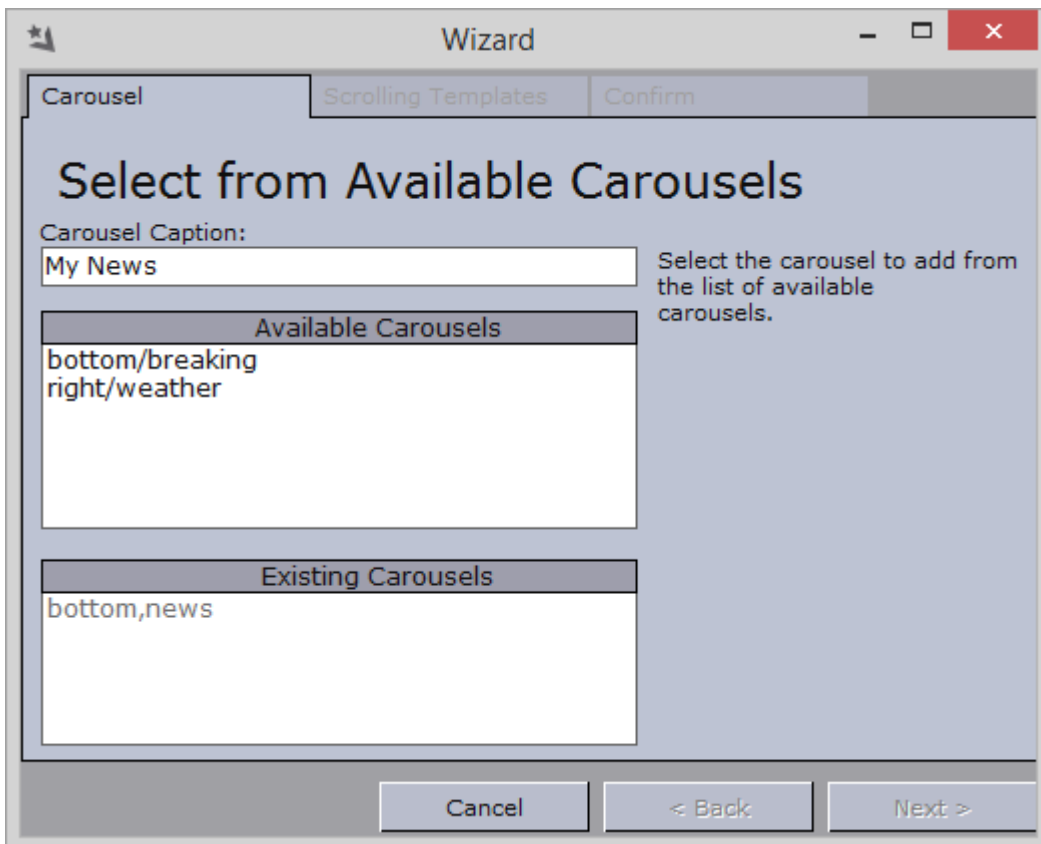
The Carousel Panel opens at wizard startup and contains the following options:

- **Carousel Caption:** Defines the carousel specific tab name in Viz Ticker Client. By default, the carousel caption is equal to the carousel name.
- **Available Carousels:** Shows a list of all available carousels. Carousels become available once a scene has been scanned. For more information, see To scan a scene.
- **Existing Carousels:** Shows a list of all carousels that have already been created.

Scrolling Templates Panel

The Scrolling Templates Panel contains a list of all available templates.
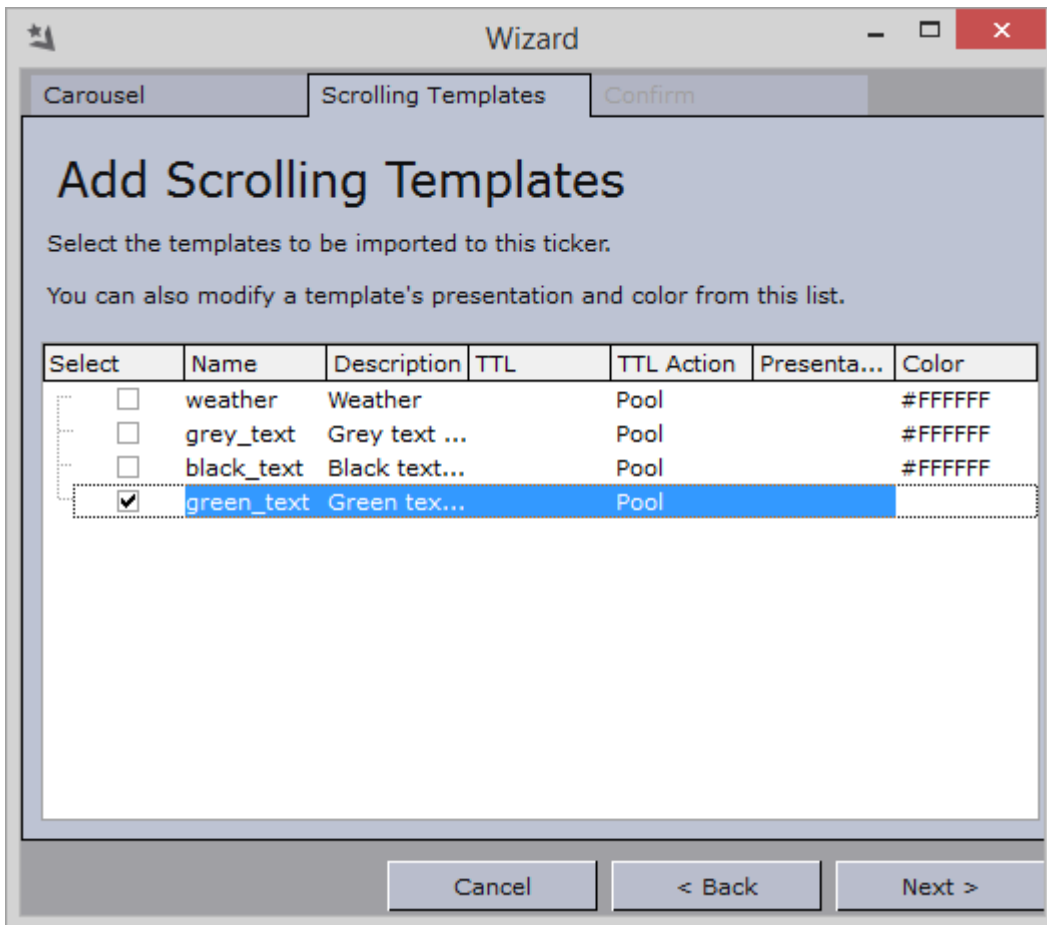
The settings that are defined in this panel affect messages that are created in Viz Ticker Client. To edit the settings of a specific template, click the various columns. Each template is presented with the following information:

- **Select:** Selects one or more templates. The templates that are selected here are available when creating messages in Viz Ticker Client.

  > ⚠ **Note:** The same template can be available for various carousels.

- **Name:** Shows the name of the template. The name is used for example when changing the default template for a carousel. The template name is defined in the Viz Artist scene settings, and cannot be changed in Ticker Wizard. The name reflects the template root container (direct sub-container of *ticker_templates*).
- **Description:** Shows the text that is displayed in the *Template* column in Viz Ticker Client's Carousel Frame. By default, the description is taken from the ControlObject description field specified during scene design.

  > ⚠ **Note:** Non-ASCII characters must not be used in the description. Also, the system resets descriptions of templates that are reimported.

- **TTL:** Defines the TTL (Time To Live) value. TTL refers to how many times the message should run before it is taken out of the carousel. By default, the TTL is set to *infinite* (column is cleared).
- **TTL Action:** Defines what should happen to the message when the TTL value is reached.
    - **Pool:** Moves the message from the Carousel List to the Pool List.
    - **Remove:** Deletes the message.
    - **Inactive:** The message status is set to inactive, and the message remains in the Carousel List.

    > ⚠ **Note:** If the **Show Time to Live** checkbox has been selected in Ticker Wizard's Ticker Carousels Panel, the *TTL* and *TTL Action* options can be modified in Viz Ticker Client's Settings Frame when creating or modifying a message.

- **Presentation:** Presentation Files can be used instead of the default editor when messages are created or edited in Viz Ticker Client. For more information, see To select the presentation file for a template.
- **Color:** All messages based on this template are displayed in Viz Ticker Client's Carousel List with the selected color. For more information, see To select the background color for a template.

To Select the Presentation File for a Template

1. In the Scrolling Templates Panel, click to enable the **Presentation** file column.
2. Click the **...** button that appears.
3. In the **Select Presentation File** dialog box that opens, browse to locate a presentation file (*.vtw* or *.vtwc*).

    > ✅ **Tip:** Presentation files are placed at the following location in the installation directory of Viz Ticker Client normally *<Installation-Directory>\presentations\<MyPresentationFile>.vtwc*. See Installation Directories.

4. Click the **Open** button. Presentation Files can be used instead of the default editor when messages are created or edited in Viz Ticker Client. Presentation Files are created in Template Wizard.

For more information, see To prepare a scrolling carousel for presentation file creation and To create a presentation file in Template Wizard.

To Select the Background Color for a Template

1. In the Scrolling Templates Panel, click to enable the **Color** column.
2. Enter the HTML color hex code in the text field that appears (for example *#00FFFF)*, or

3. Click the **...** button that appears.
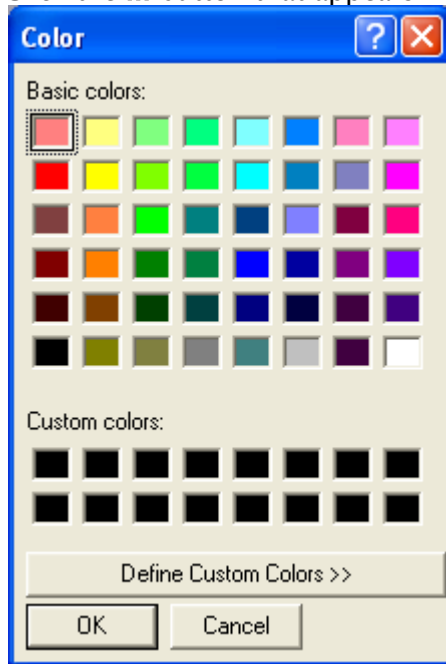
4. In the Color Dialog Box that opens, select a color.
5. Click the **OK** button.

> ✅ **Tip:** The text color in Viz Ticker Client is automatically adjusted to have a good contrast to the selected background color.

Confirm Panel

The Confirm Panel contains a summary of the carousel setup information.
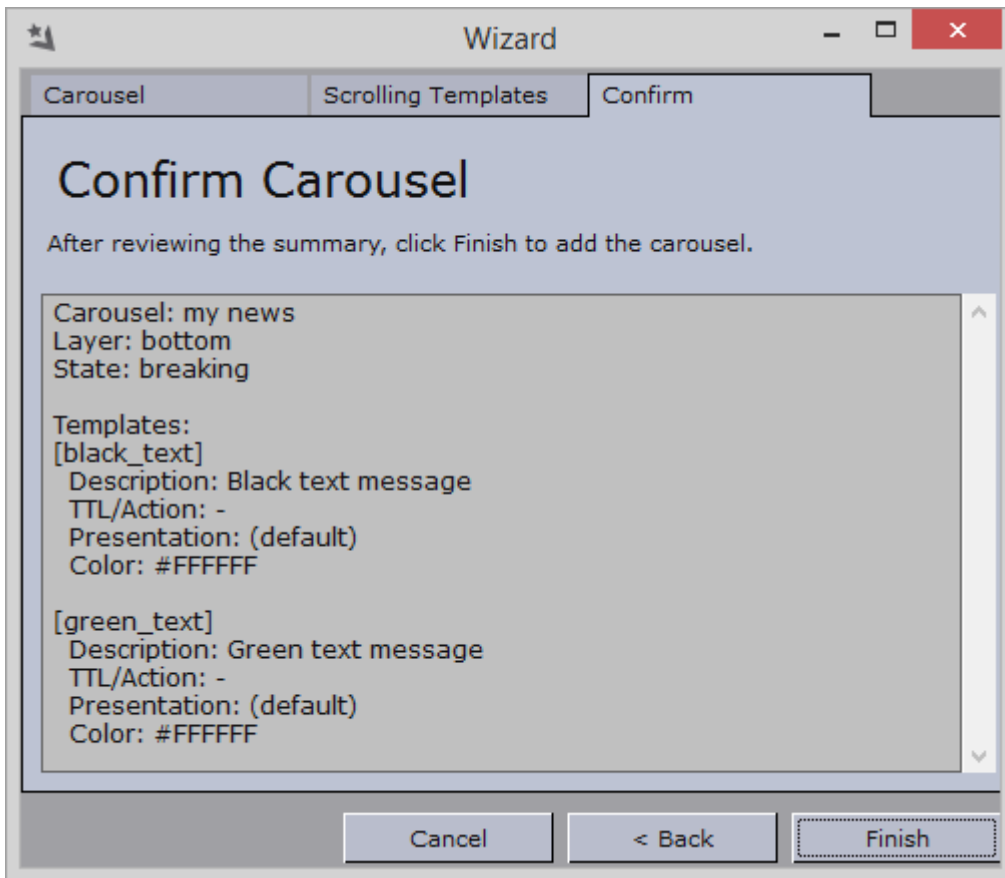
4.10.2   Flipping Carousel Wizard

Flipping carousel scenes are characterized by containing one or more Toggle plug-ins with relevant s and stop points. The elements in a flipping ticker are flipped in a static position.

The Flipping Carousel Wizard is similar to the Scrolling Carousel Wizard. The Carousel Panel and Confirm Panel are identical in the two wizards, while the settings in the Scrolling Templates Panel and Flipping Templates Panel are slightly different.

> ⚠ **Note:** This section describes using the Carousel Wizard to create a new carousel. Existing carousels can be edited in the Ticker Carousels Panel.

Using the Flipping Carousel Wizard

1. To open the Flipping Carousel Wizard, click the **Create Flipping Carousel** button in the Main Panel.
2. The wizard opens and you see the Carousel Panel.
    a. Give the Carousel a name in the **Carousel Caption** box.
    b. Double click a carousel in the **Available Carousels** list.
3. The wizard then proceeds to the Flipping Templates Panel.
    a. Select one or more templates from the **Template Scenes** list.

> ✅ **Tip:** The scenes can be sorted by name or date, and presented in forward or reversed order.

      b. *Optional:* Add Presentation Files. (See To select the presentation file for a template).
      c. *Optional:* Select a color for the messages. (See To select the background color for a template).
      d. Click **Next**.
4. The wizard proceeds to the Confirm Panel.
      a. Make sure that the setup information is correct.
      b. Click **Finish**. This exits the wizard and create the new carousel.

Flipping Templates Panel

The Flipping Templates Panel shows the paths and names of all available carousel and template scenes.
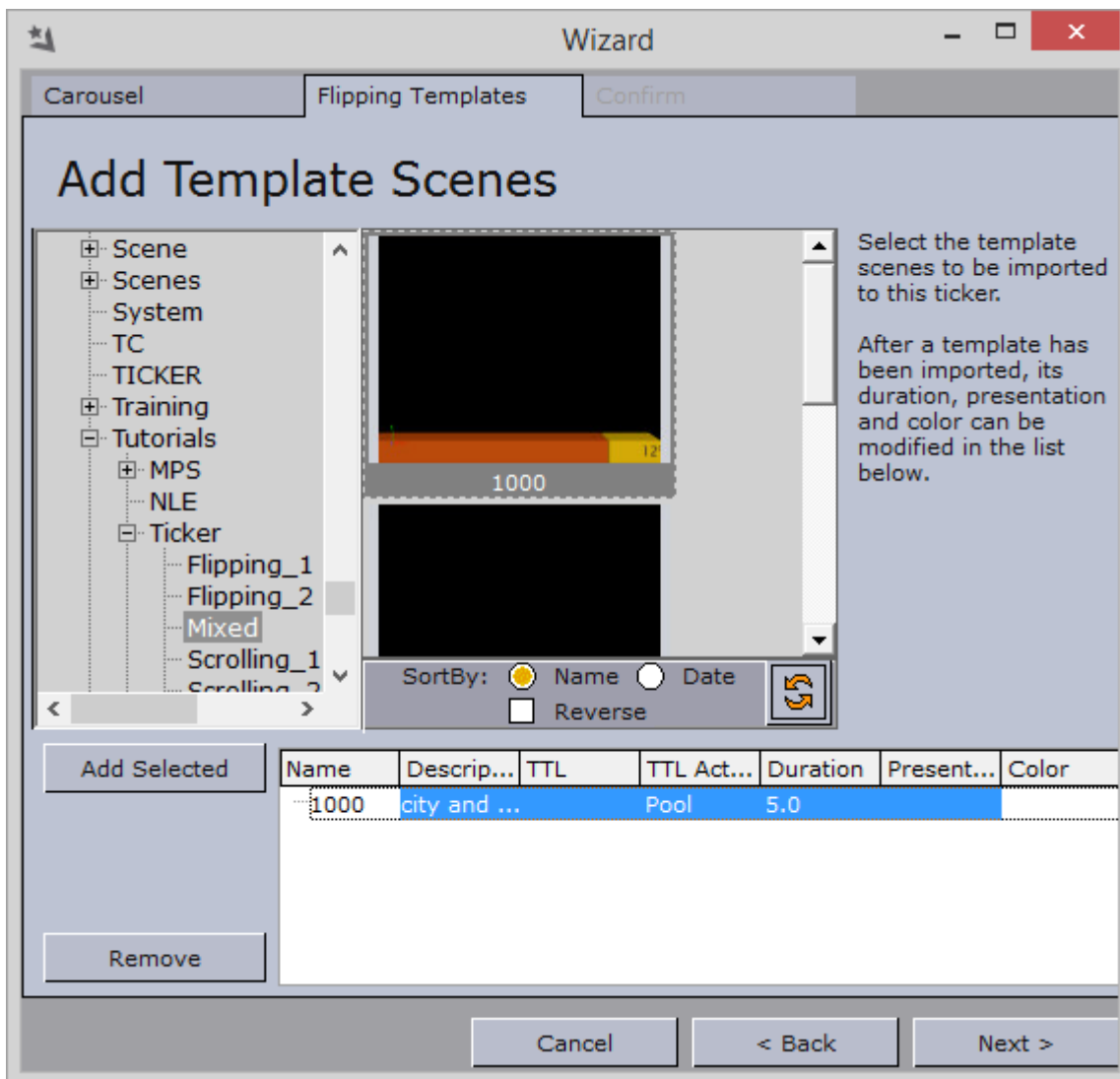
The settings that are defined in this panel affect the messages that are created in Viz Ticker Client. Each template is presented with information such as name, description, TTL, TTL action, duration,

presentation file, and color. For detailed information about these settings, see Scrolling Templates Panel.

> ⚠ **Note:** The *Duration* column is only available for flipping templates. Duration refers to how many seconds the message should be on air, before flipping to the next message in the carousel.

To edit the settings of a specific template, click the various columns in the list.

4.10.3   Reimport Templates Wizard

The Reimport Templates Wizard allows reimporting existing templates that have changed scene design since the original import. Templates that have changed control plug-ins, animations, container structure, or other settings must be reimported before the changes take effect.
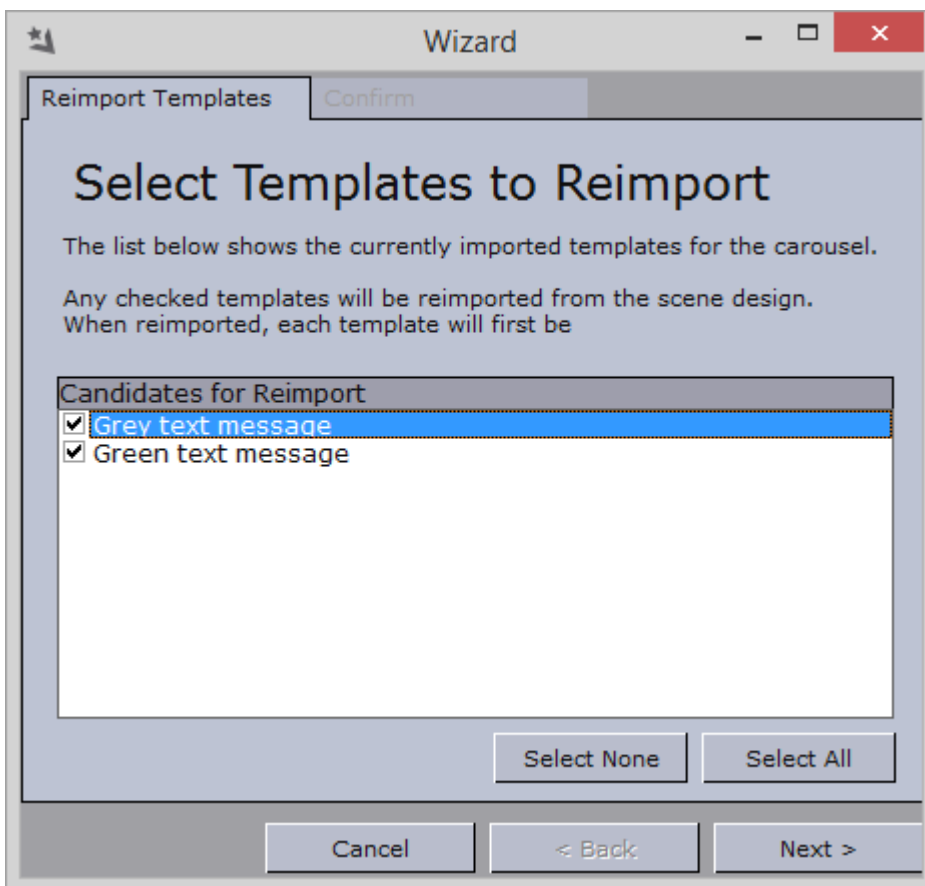
To Reimport Templates

1. In the Ticker Carousels Panel, click the **Reimport Templates** button.
2. Select the templates that should be reimported.
3. Click **Next**.
4. Check that the details are correct and click **Finish**. This exits the wizard and reimports the selected templates.

> ⚠ **Note:** The selected templates are deleted first from the Media Sequencer, and then reimported as new templates.

4.10.4   Output Channel Wizard

The Output Channel Wizard allows creating new output channels.

The *Main* output channel is created by default when a carousel is set up in one of the carousel wizards Other output channels must be created from this Output Channel Wizard.

To Create an Output Channel

1. In the Output Channels Panel, click the **New** button. The Output Channel Wizard opens.
2. Type a descriptive name in the **Output Channel Name** text box.

   > ⚠ **Note:** The output channel name must contain alphanumeric characters only. For more details, see Global Conventions.

3. Enter the **Renderer Hostname** and **Renderer Port**, and check the **Enable OneBox** checkbox if this output channel is part of a dual/quad Viz Engine setup where output channels are driven by several Viz Engine instances on the same host machine.
4. Click **Next**.
5. In the **Confirm Carousel** panel that appears, make sure that the output channel settings are correct, and then click the **Finish** button. This exits the Output Channel Wizard and create a new global output channel.

5   Viz Ticker Client

Viz Ticker Client is the Vizrt Ticker Client application. This section provides information on how to operate Viz Ticker Client.

> ⚠ **Note:** If settings that affect Viz Ticker Client are modified in Ticker Wizard, Viz Ticker Client must be restarted in order for the changes to take effect. It is recommended to quit Viz Ticker Client before starting Ticker Wizard.

This section contains information on the following topics:

- Getting Started
- Main Menu
- Configuration Panel
- Control Panel
- Carousel Specific Panel

5.1   Getting Started

5.1.1   To Start Viz Ticker Client

Start Viz Ticker Client from **Start > All Programs > Vizrt > Viz Ticker > Viz Ticker Client**.

5.1.2   Logging In

At system startup, the Login Window appears.

The login is used for example to define the owner of messages, see Settings Frame.

- **Username:** Defines the name of the current user. The username can be any name defined by the user.
- **Auto Login:** Logs into the system automatically with the current user details, if enabled.
  If **Auto Login** is enabled, Viz Ticker Client automatically logs in to the system with the current user details. This setting should typically be used on single-user computers.
  If **Auto Login** is disabled, the Login Window appears at system startup, and a username must

be entered before being able to log in. This setting should typically be used on multi-user computers.

> ⚠ **Note:** This button is only available if the **Change User** checkbox has been enabled in Ticker Wizard's Settings Panel.

5.1.3 To Change the User

1. In Viz Ticker Client's Control Panel, click the **Change User** button.

> ⚠ **Note:** This button is only available if the **Change User** checkbox has been enabled in Ticker Wizard's Settings Panel.

2. In the Login window that opens, type a username.
3. Click the **OK** button.

5.2 Main Menu

The Main Menu, at the top of the main window, contains various tabs, which open the respective panels.

- **Config:** Opens the Configuration Panel.

> ⚠ **Note:** The Configuration Panel is only available if the Viz Ticker Client Startup Parameters have been set to *-config on*.

- **Control:** Opens the Control Panel. The Main Menu also contains one or more carousel specific tabs. For each available carousel, there is one tab and one carousel specific panel.
- **<Carousel Specific>:** Opens the Carousel Specific Panel.

5.3 Configuration Panel

The Configuration Panel opens when clicking the **Config** tab in the Main Menu.
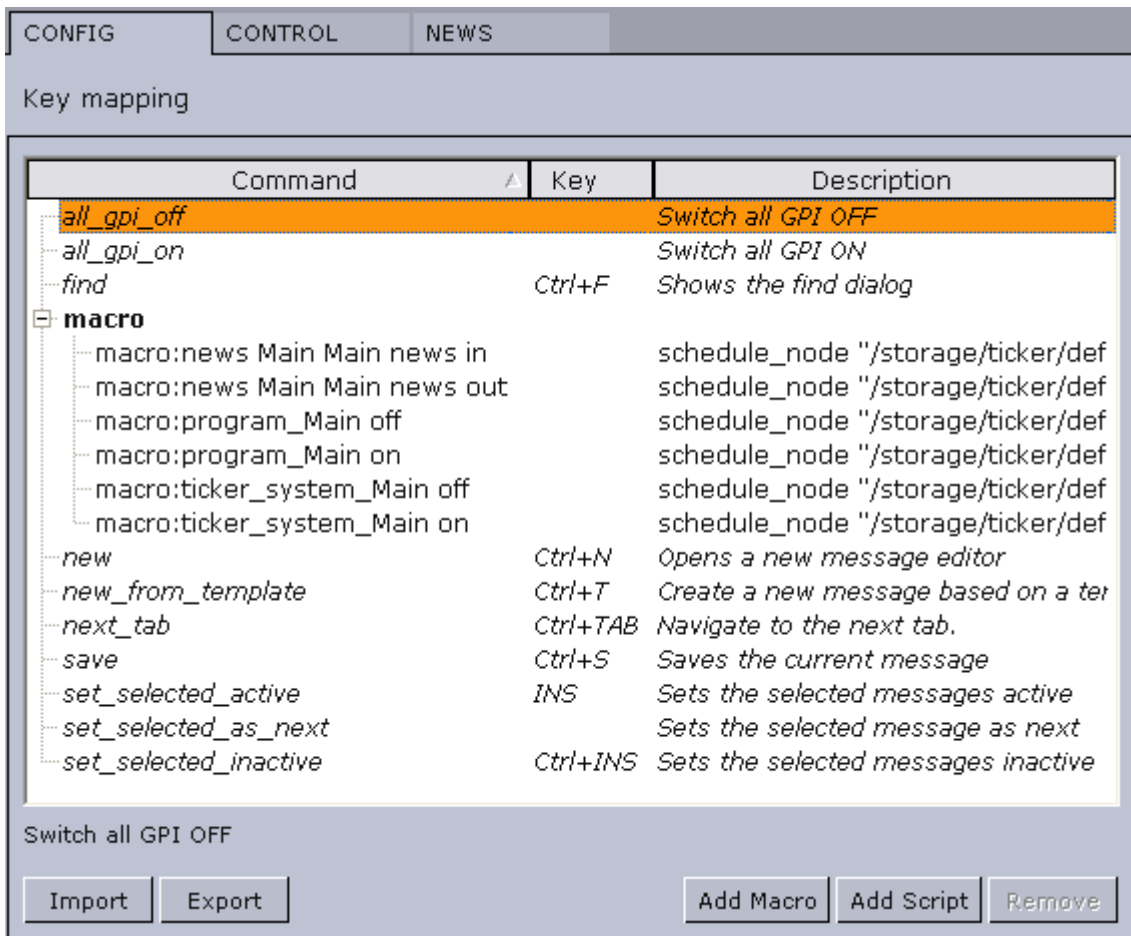
> ⚠ **Note:** The Configuration Panel is only available if the Viz Ticker Client Startup Parameters have been set to *-config on*.

The .Configuration Panel v3.1 shows a list of all existing keyboard shortcuts. From this panel it is possible to add user-defined keyboard shortcuts. It is also possible to import a keyboard shortcut file, or to export such a file.

> ⚠ **Note:** Changing the keyboard shortcuts affects all Ticker Systems connected to the same Media Sequencer.

- **Import and Export:** Imports and exports shortcut settings to and from an XML-file for backup and reuse of typical configurations.

  > ⚠ **Note:** The keyboard file has a .kbd file extension.

- **Add Macro:** Writes a macro command and links it to a shortcut key.
- **Add Script:** Writes a Visual Basic script and links it to a shortcut key. If a key combination is already in use, you are given the option of overriding the existing command. This leaves the other command without an assigned shortcut.

· **Remove:** Removes macros and scripts that are custom made. Note that no warning is issued when performing this operation.

> 🛑 **IMPORTANT!** Avoid using **SHIFT** as part of a command, as this disables the possibility to use the assigned key when writing characters (for example in upper case).

5.4   Control Panel

The Control Panel opens when clicking the **Control** tab in Viz Ticker Client's Main Menu.
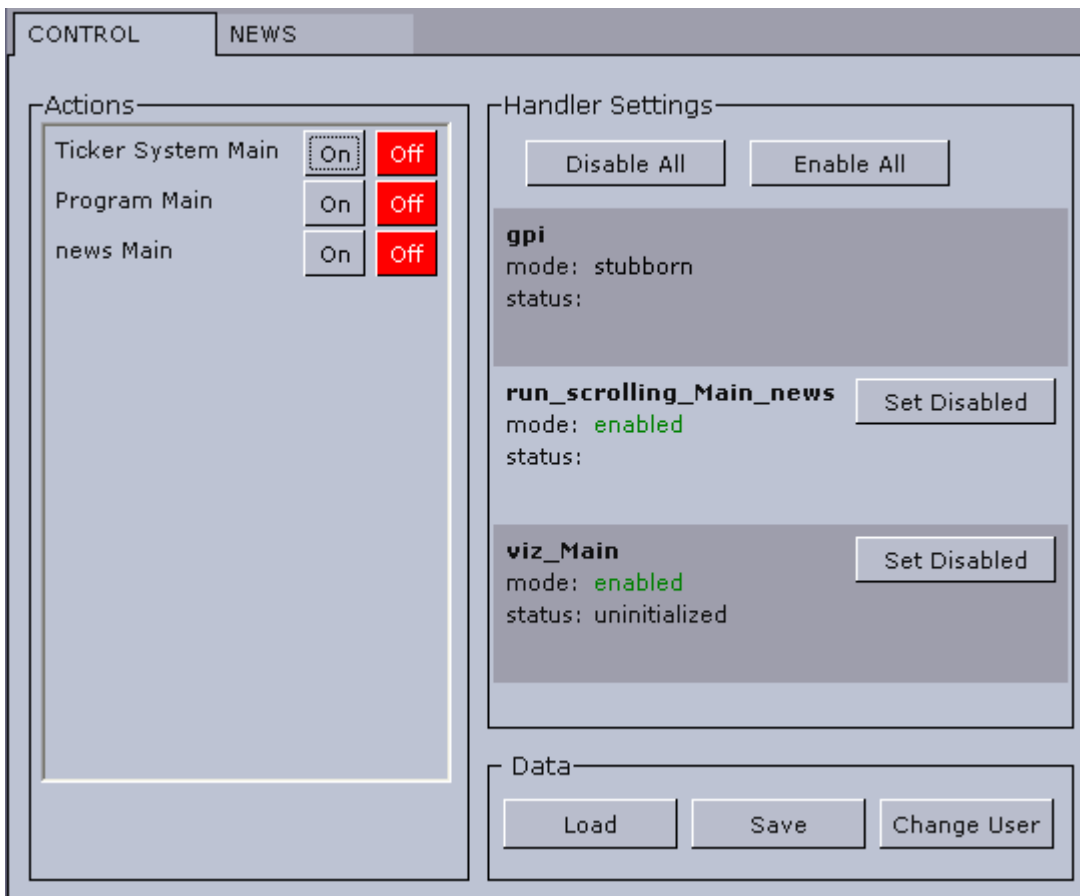
> ⚠️ **Note:** The Control Panel is visible per default, but can be hidden if the Viz Ticker Client Startup Parameters has been set to *-control off*.

5.4.1　Actions Frame

The button groups and buttons in the Actions Frame reflect the list in Ticker Wizard's Buttons Panel. The button groups and buttons are created and modified in the Buttons Panel.

> ⚠ **Note:** If the Control Panel is hidden, the on and off states of carousels must be controlled from the Carousel Specific Panel. For information on how to show/hide the .Control Panel v3.1, see Viz Ticker Client Startup Parameters. For information on how to add buttons to the Carousel Specific Panel, see Button Groups Available in Carousel Frame.

For every existing output channel, *Ticker System On/Off* and *Program On/Off* buttons appear in the Actions Frame.

- **Ticker System On:** Starts the Ticker System so that commands can be processed and carousels sent On Air.

  > ⚠ **Note:** The commands are not actually processed and carousels are not actually sent On Air until *Program On* is activated.

- **Ticker System Off:** Takes the Ticker System off, making sure that commands can no longer be processed and carousels sent On Air. When the Ticker System is in its `off` state, nothing happens if the *Program On/Off* or the On/Off buttons belonging to a carousel are selected. When the *Ticker System Off* button is selected, the belonging *Program Off* button is also selected.
  The *Program On/Off* functionality is meant for short breaks, for example commercial breaks.

- **Program On:** Sends the carousels that are in its *in* state On Air. The *Program On* button can only be selected if the belonging *Ticker System On* button is selected.
- **Program Off:** Takes all carousels off air. The *Program Off* button pauses the carousel, so that the next message being played when the carousel resumes, is the next active message in the carousel. During the break (*Program Off*) it is possible to apply new states to the carousels. The changes are not executed until the system resumes from the break (*Program On*).
  On/Off buttons belonging to each available carousel on each available output channel are also available in the Actions Frame. All carousels can then be controlled directly from the Actions Frame, and more importantly, the carousels can be controlled externally with GPI.

- **<Carousel Specific> On:** Takes the specific carousel On Air.

> ⚠ **Note:** The carousel only goes On Air if the *Ticker System On* and *Program On* buttons are also selected.

· **<Carousel Specific> Off:** Takes the specific carousel off air.

> ✅ **Tip:**
> Button groups can also be added to the Buttons Frame in the Carousel Specific Panel . To do this, enable the belonging button group in Ticker Wizard's Button Groups Available in Carousel Frame.

5.4.2 Handler Settings Frame
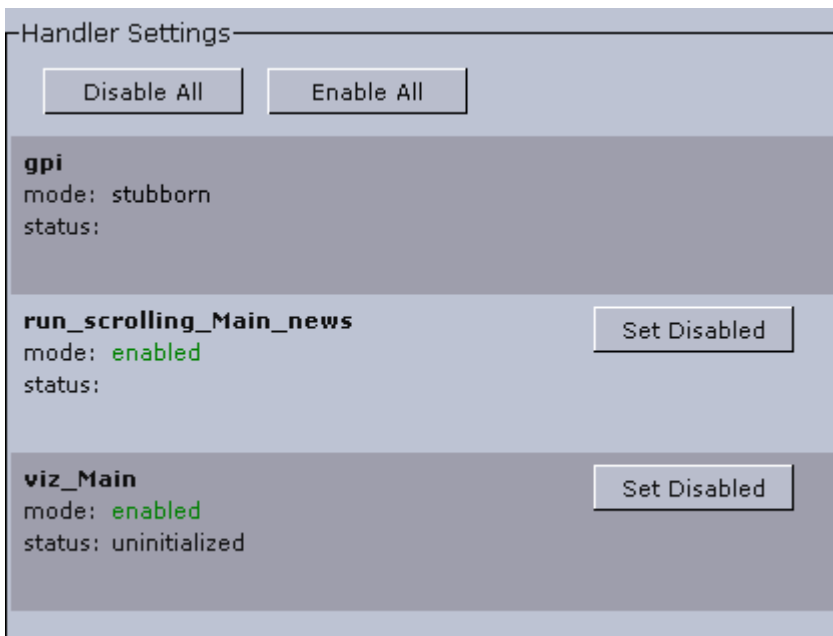
> ⚠ **Note:** The Handler Settings frame is only available if the **Handler Settings** check box is selected in Ticker Wizard's Settings Panel.

A handler is a component that the Ticker System uses to communicate with a certain Viz Engine. A handler has the ability to disable itself if an error occurs. Separate handlers are created for each output channel and scrolling carousel.

The Handler List provides information about the handlers.

· **Name:** Shows the name of the handler.

> ⓘ **Example:** viz_<output channel>

· **Mode:** Shows the mode of the handler. The mode can be either *enabled* (green) or *disabled* (red).

- **Status:** Shows the status of the handler. An example of a status error message can be seen in the Troubleshooting section.

5.4.3 Data Frame
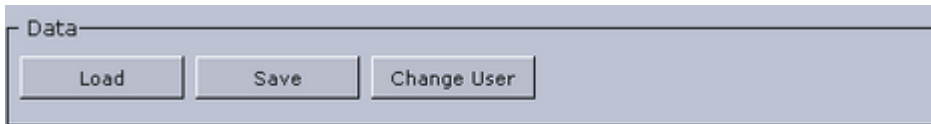
> ⚠ **Note:** The Data frame is only available if the **Load/Save Buttons** or **Change User** checkboxes are selected in Ticker Wizard's Settings Panel.

- **Load:** Opens a dialog box where previously saved carousels can be browsed and loaded. For more information, see To load previously saved carousels.
- **Save:** Opens a dialog box where the current carousels can be saved as an XML file.
- **Change User:** Opens the Login Window. See Logging In.

To Load Previously Saved Carousels

1. In Viz Ticker Client's Data Frame, click the **Load** button.

> ❗ **Caution:** Loading a previously saved carousel replaces all current messages.

2. Browse to find a previously saved XML file and click **Open**.
3. Click one or more *Ticker System* **Off** buttons in the Actions Frame.
4. Restart Viz Ticker Client. The messages in the imported carousel then replace the messages in the current carousel.

5.5 Carousel Specific Panel

The Carousel Specific Panel opens when clicking the <Carousel Specific> tab in Viz Ticker Client's Main Menu.

The tab name is defined in Ticker Wizard, either when setting up the carousel in one of the carousel Wizards, see Carousel Panel, or when specifying the carousel details in the Ticker Carousels Panel.
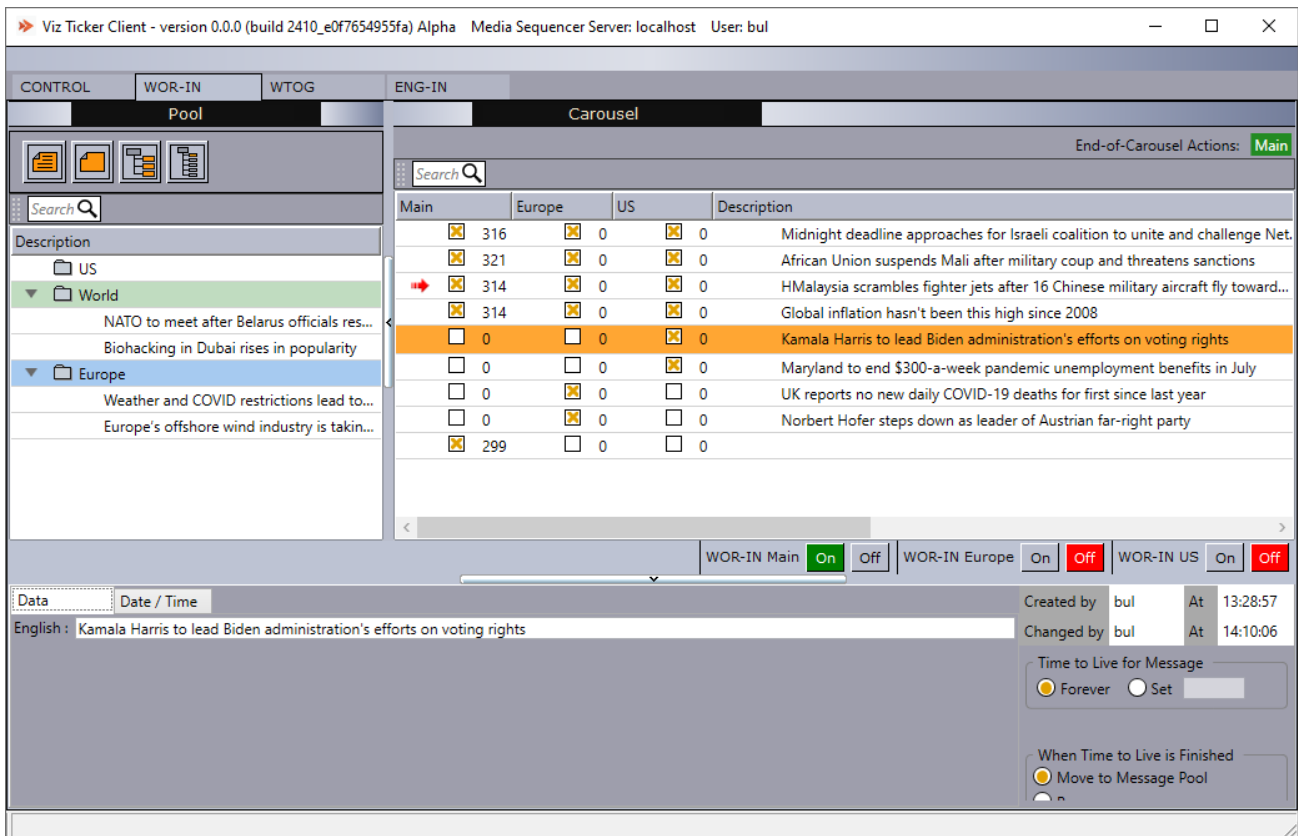
One or more carousel specific tabs in the Main Menu open corresponding carousel specific panels.

This Carousel Specific Panel contains the following areas:

- Pool Frame
  - Pool Frame Buttons
  - Pool Search
  - Showing/Hiding the Pool Frame
  - Pool List
  - Pool List Context Menu
  - Pool List Message Context Menu
- Carousel Frame
  - Carousel List
  - Carousel Search
  - Carousel Columns Menu
  - Carousel List Message Menu
  - End-of-Carousel Action Status
  - Default Output Channel Activation Frame
  - Activating Messages
- Buttons Frame
- Elements Editor
  - Showing/Hiding the Elements Editor
  - Creating a Message
  - Adding an Image using Media Search
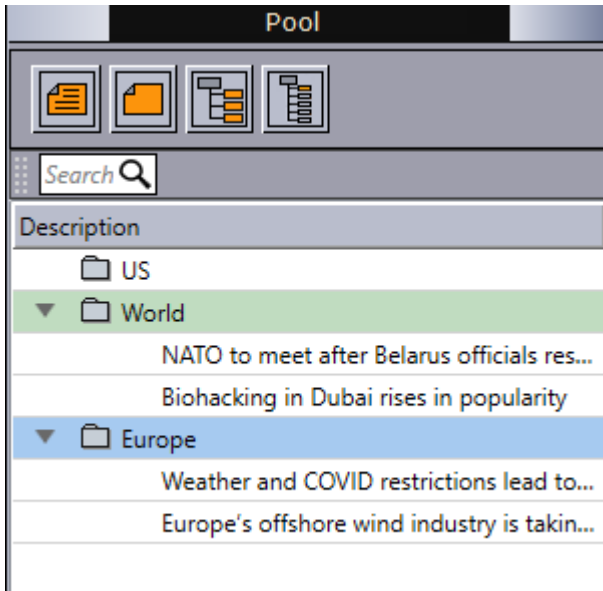
5.5.1   Pool Frame

In the upper left part of the Carousel Specific Panel is the Pool Frame:

Pool Frame Buttons

The Pool Frame contains the following:

· **New Message:** Makes it possible to create a new message in the Elements Editor, based on the default template.

> ✅ **Tip:** The default template is defined in Ticker Wizard's Ticker Carousels Panel.

· **New Template Element:** Opens the **Select Template** dialog box, where any available template can be selected. The message that opens in the Elements Editor is based on this template.

> ✅ **Tip:** Templates are added in Ticker Wizard when setting up the carousel in one of the carousel Wizards, see Scrolling Templates Panel or Flipping Templates Panel.

· **New Group:** Create a new group in the Elements Editor. Groups help to organize the messages in a carousel. See Creating a Group or Section.
· **New Section:** Create a new section in the Elements Editor. Sections are advanced groups, that help organize the messages in a carousel. See Creating a Group or Section.

Pool Search

Search for all item types in the Pool List, including messages, groups, and sections.

Showing/Hiding the Pool Frame

Resize or hide the Pool Frame by dragging or clicking the handle on the frame border.

Pool List

When new messages are created (saved), they are added either to the Pool List or Carousel List, depending on which of the *Add New Messages* option buttons has been selected in Ticker Wizard's Ticker Carousels Panel. Messages can be dragged from one list to the other.

Pool List Context Menu

Right-click the column headers to access the following options:

- **Description:** Shows the Description column, which displays the actual message text.
- **Template:** Shows the Template column, which displays the name of the template the message is based on.
- **Store as Default:** Defines the currently selected columns as default.
- **Use Default:** Activates the columns in the Pool List based on the default settings.
- **Auto Fit Columns:** Adjusts the width of the columns automatically.

Pool List Message Context Menu
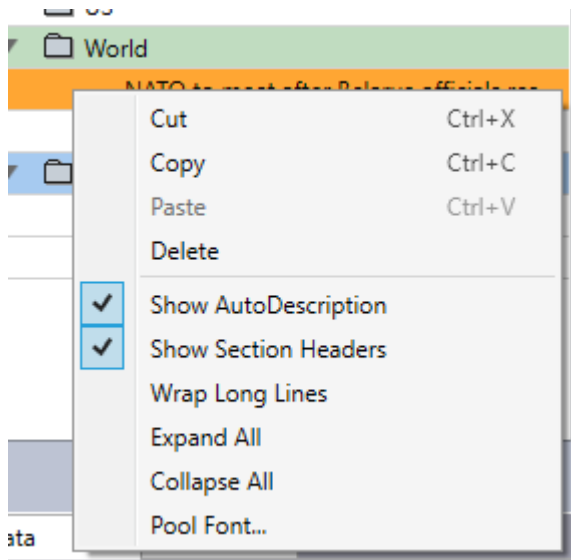
Right-click a message in the Pool List to access the following options:

- **Cut / Copy / Delete:** Performs this action on the currently selected messages.
- **Paste:** Pastes the currently cut messages.

- **Wrap Long Lines:** Wraps text in the Pool list.
- **Show AutoDescription:** Allows manually changing the description of the element when disabled.
- **Show Section Headers:** Enables headers for the sections in the Pool List.
- **Expand All / Collapse All:** Expands/collapses all groups and sections in the Pool List.
- **Pool Font:** Selects a new font for all messages in the Pool List.

> ⚠ **Note:** The font settings only apply to the layout in Viz Ticker Client, not the actual messages that are sent On Air.

> ✅ **Tip:** See also Keyboard Shortcuts.

5.5.2 Carousel Frame

In the upper right part of the Carousel Specific Panel is the Carousel Frame:

Carousel List

The Carousel List contains groups, sections, and messages that should be taken On Air. The default behaviors for adding and activating messages are defined in the Ticker Wizard Carousel Details Frame.

Carousel Search

Search for all item types in the Carousel List; messages, groups, and sections.

Carousel Columns Menu

Right-click the column headers to access the following options:

- **Description:** Show the Description column, which displays the actual message text.
- **End:** Show the End column, containing the message end date and time.

  > ⚠ **Note:** The *End* information is only available if the *Date/Time* settings have been defined in the Elements Editor.

- **<Output Channel>:** One column is available for each output channel. Selecting the checkbox in one of these columns activates the message. The current message in the carousel is emphasized by a red arrow positioned at the left side of the active/inactive checkbox. The number of times the message has run in the carousel, is displayed at the right side of the checkbox.
- **Start:** Show the Start column, containing the message start date and time.

  > ⚠ **Note:** The *Start* information is only available if the *Date/Time* settings have been defined in the Elements Editor.

- **Template:** Show the Template column, containing the description of the template the message is based on. This description is defined when setting up a carousel in one of the carousel Wizards in Ticker Wizard, see Scrolling Templates Panel.
- **Duration:** Show the Duration column, containing how many seconds the message should be On Air, before flipping to the next message in the carousel.

  > ⚠ **Note:** The *Duration* column is only available for flipping carousels.

- **TTL:** Show the TTL column, containing how many times the message should run before it is taken out of the carousel. When the number of runs reaches the Time To Live (TTL)-limit, the *TTL Action* is performed.

  > ⚠ **Note:** The *TTL* option is only available if the *Show Time to Live* checkbox has been selected in Ticker Wizard's Ticker Carousels Panel.

- **Store as Default:** Defines the currently selected columns as default.
- **Use Default:** Shows the columns in the Carousel List based on the default settings.
- **Auto Fit Columns:** Adjusts the width of the columns automatically.

Carousel List Message Menu
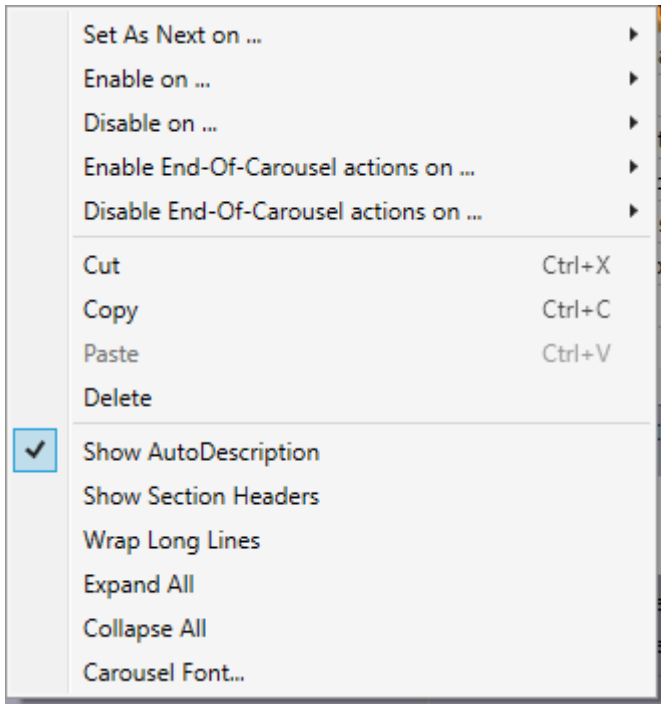
Right-click a message in the Carousel List to access the following options:

- **Set as Next on:** Carousels can have multiple output channels:
    - **<output channel>:** For this particular output channel, changes the playout order so that the selected message is next to play.
    - **All Enabled Channels:** On all enabled output channels, sets the currently selected message to the next item in the carousel when resuming from a pause.

        > ⚠ **Note:** The *All Enabled Channels* option is only available for carousels with multiple output channels.

        > ⚠ **Note:** Before using Set as Next, the *Ticker System* should be set to **ON** for the chosen Output Channel (see Control Panel). Otherwise, Set as Next is ignored when the carousel is later switched on.

- **Enable on:** Carousels can have multiple output channels:
    - **All Channels:** Activates the currently selected message on all output channels.
    - **<output channel>:** Activates the currently selected message on this particular output channel.
- **Disable on:** Carousels can have multiple output channels:
    - **All Channels:** Deactivates the currently selected message on all output channels.
    - **<output channel>:** Deactivates the currently selected message on this particular output channel.

- **Enable/Disable End-Of-Carousel actions on:** Enables and disables End-of-Carousel actions per channel if configured to be visible in the Ticker Wizard.
- **Cut / Copy:** Performs this action on the currently selected messages.
- **Paste:** Pastes the currently cut messages.
- **Delete:** Moves the currently selected messages to the Pool List.
- **Show Auto Description:** Shows the automatically generated description for messages when enabled. When disabled, it is possible to change the description manually.
- **Show Section Headers:** Enables headers for the sections in the Carousel List.
- **Wrap Long Lines:** Presents the whole message descriptions (on multiple lines if necessary) if enabled.
- **Expand All:** Expands all groups and sections in the Carousel List.
- **Collapse All:** Collapses all groups and sections in the Carousel List.
- **Carousel Font:** Selects a new font for all messages in the Carousel List.

> ⚠ **Note:** The font settings only apply to the layout in Viz Ticker Client, not the actual messages that are sent On Air.

> ✅ **Tip:** See also Keyboard Shortcuts.

End-of-Carousel Action Status

In the upper right part of the window, there is a visual indication on whether the End-of-Carousel actions are activated on this carousel, per channel:

Default Output Channel Activation Frame

The Default Output Channel Activation Frame allows you to define the default activation settings for each available output channel. Messages get their activation status when they are added to the Carousel List.

You can enable this feature for each carousel by going to **Ticker Wizard >** Ticker Carousels Panel **> <MyCarousel> > Add Messages > User Controlled**.

For example, in the image above, messages that are added to the Carousel List are active on the *Main* channel and inactive on *MyOutputChannel*.

Activating Messages

- To activate all new messages in a carousel by default, going to **Ticker Wizard >** Ticker Carousels Panel **> <MyCarousel> > Add Messages > Active**.
- To activate existing messages in the Carousel List, right-click one or more messages, and select **Enable On** <Output Channel>.

- If the <Output Channel> column is shown in the Carousel List, selecting the checkbox in front of the message.

5.5.3   Buttons Frame

In the lower right part (center right if Elements Editor is open) of the Carousel Specific Panel is the Buttons Frame.

HEADLINES Main On Off

Buttons are available in this frame according to the button groups that have been added to the carousel in Ticker Wizard's Ticker Carousels Panel, see Button Groups Available in Carousel Frame.

> ✅ **Tip:** The button groups in Viz Ticker Client's Control Panel perform the same actions as the buttons in this frame.

5.5.4   Elements Editor

At the bottom of the Carousel Specific Panel is the Elements Editor, where messages, groups, and sections are created and modified.

The appearance of the Elements Editor changes according to the item that is being created, whether it is a message, group, or section.

If the template is configured to fetch content from external sources, one or more Feed Browsing icons are available in this frame. See Feed Browsing in Viz Ticker Client.

> ⚠ **Note:** When the Ticker system is running, it buffers the upcoming message. This means that changes made to that message are not reflected on the renderer until next playout loop.

Showing/Hiding the Elements Editor

Resize or hide the Pool Frame by dragging or clicking the handle on the frame border.

Creating a Message

Create a new message, in either of the following ways:

- Click the **New Message** button in the Pool Frame. This opens a new message based on the default template.
- Or, click the **New Template Element** button in the Pool Frame then select a template from the list.

> ⚠ **Note:** To include text in a message, the template must be text based (i.e. the scene contains the ControlText plug-in). If the element contains other Control plug-ins (for example materials, objects, or images), various other datapools are displayed instead of the text box.

Right-click the Data Frame to access the following options:

- **Editor Font:** Select a new font for the content in the Data Frame. This setting only applies to the layout in Viz Ticker Client, not the actual messages that are sent On Air.

> ⚠ **Note:** The **flow direction** of text (left-to-right or right-to-left) in a carousel can be configured in the Carousel Details Frame.

For additional message settings, see Pool Frame Buttons and Activating Messages.

Adding an Image using Media Search

Image Search requires that a search provider has been configured in the Image Search Provider Frame in Ticker Wizard, and that the message contains a TTWImage component with *isImageMediaSearch* enabled (see Image Search using Presentation Files).

The Elements Editor shows the items defined in the Presentation Files, including an image icon.
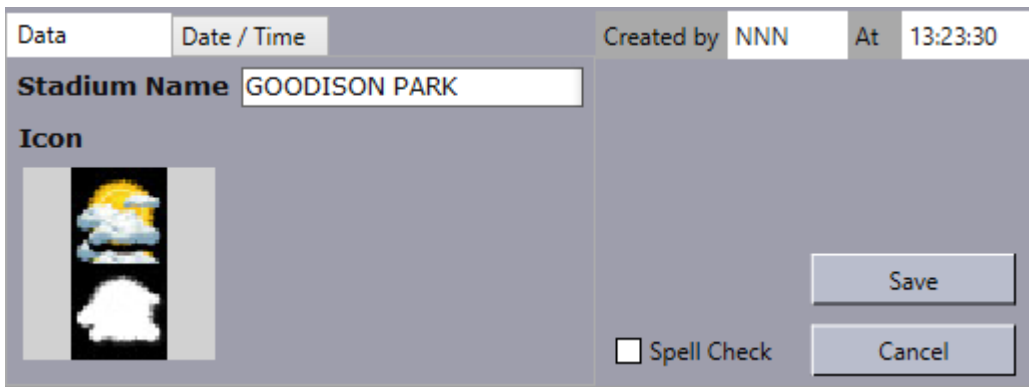
Clicking on the image icon opens up the Media Search dialog (below), allowing you to select an image from one of the search providers.
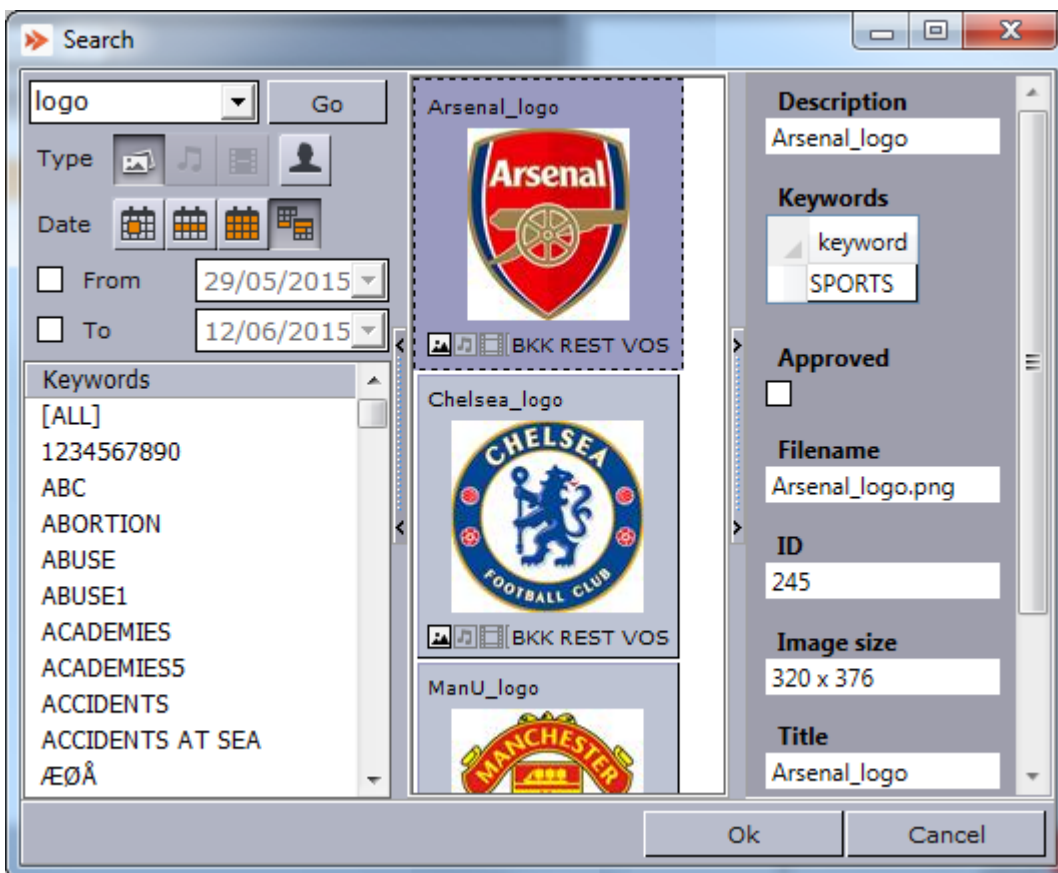
- In the **Search** pane on the left, you can combine the use of filters, text search, keywords, and from/to dates to narrow down your results.
- The default **Type** is Stills, which searches for still images only.
- If selected, the **Person Filter** filters the results by only displaying pictures that are defined as person (i.e. profile) images in Object Store. This is only available if a RestVOS search provider is configured.
- **Date:** Filters results based on date range. Select from **Last 24 hours, Last week, Last month,** and **Custom date range**.

- **Keywords:** Templates that use the RequiredKeyWords property have the relevant keywords pre-selected in the list. Keywords are configured for each Service Provider.
- Display an item's metadata by selecting **Show Details View** from the context menu in the search results. The metadata is received from the asset search provider, and the type of data available can vary. This can be configured in each asset search provider.
  For full details see the section on *Media Search* in the Viz Pilot User Guide.

Creating a Group or Section
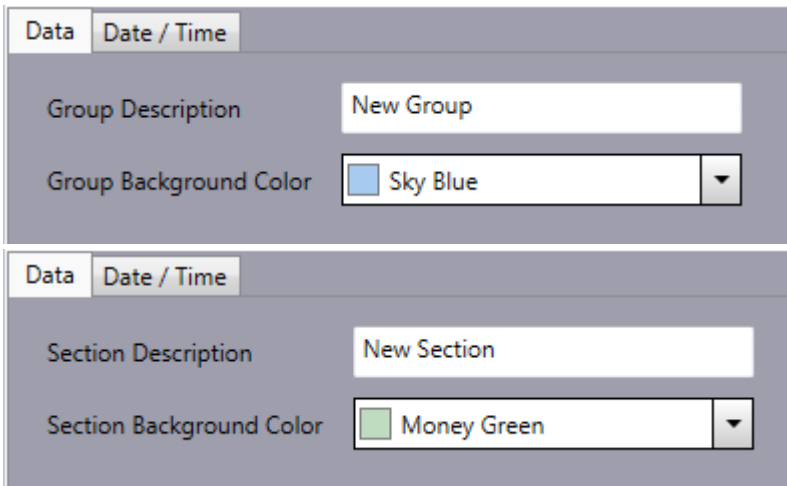
Click the **New Group** or **New Section** button in the Pool Frame. You can enter a name and select a background color for the group/section. Groups and Sections can be nested.

The *settings* for a group or section apply for all items within the group/section. For example, enabling a group/section on a certain output channel also enables all messages in this group/section on this output channel.

A section contains two parts:

- **Section Heads:** Contains one or more element, which are played out first of all the elements. For example, a notification that the following elements are news, weather, or sports messages. After a break, the section heads elements are played out first, before continuing with the element that was next before going to the break. The section heads elements are

typically static, so it is possible to hide the section headers to prevent users from accidentally updating these elements, as in the following image:

> ✅ **Tip:** The section headers can be shown/hidden from the Pool List Message Context Menu or Carousel List Message Menu.

· **Section Elements:** Contains one or more elements for the ticker.

> ⚠️ **Note:** If a section contains one or more section heads elements, but no section elements, the elements in this section are not shown at all, as it does not make sense to show a header with no content.

Date and Time Frame

The Date/Time Frame becomes available when selecting the **Date/Time** tab in the Elements Editor.

The time format is automatically adjusted according to the system settings on the machine. The format is either 12 hour (01:00:00-12:59:59 AM/PM) or 24 hour (00:00:00-23:59:59). AM/PM dropdown lists become available next to the Start/End boxes on machines with 12 hour time format.

> ⚠ **Note:** If a message should be valid only on one certain day, then time information must be added, 23h 59m 59s if necessary, otherwise the information is read as all year round, and the information is discarded.

Settings Frame

At the right side of the Elements Editor is the Settings Frame.

- · **Created By:** Displays the username of the original message creator, and also the time of creation.
- · **Changed By:** If a message has been modified, the name of the user and the time of the last change are displayed.

  > ✅ **Tip:** The user name is defined when Logging In.

- · **Time To Live For Message:** The Time To Live (TTL) value refers to how many times the message should run before it is taken out of the carousel.
  - · **Forever:** Sets the TTL value to infinite, so that the message runs forever.
  - · **Set:** Defines the number of times the message should run before it is taken out of the carousel.
- · **When Time To Live is Finished:** This option refers to what should happen to the message when the TTL value is reached.
  - · **Move to Message Pool:** Moves the message from the Carousel List to the Pool List.
  - · **Remove:** Deletes the message.
  - · **Set Inactive:** The message status is set to inactive, and the message remains in the Carousel List.

    > ⚠ **Note:** The two *Time To Live* frames are only available if the **Show Time to Live** checkbox has been selected in Ticker Wizard's Ticker Carousels Panel.

> ⚠ **Note:** The default **TTL** settings for each carousel are defined in Ticker Wizard's Carousel Details Frame.

· **Spell Check:** If enabled, the text in the **text** text box (left side of the Elements Editor when **Data** tab is selected) is spell checked.

> ⚠ **Note**: Spell-check is not supported if a template uses a presentation file. The *Spell Check* checkbox will not appear in the message editor.

> ⚠ **Note:** The spell check function is based on the *Microsoft Office* dictionary, and uses the default language in *Microsoft Word*.

> ⚠ **Note:** The default **spell check** settings for each carousel are defined in Ticker Wizard's Carousel Details Frame.

6   Viz Ticker Web Client

The **Viz Ticker Web Client** allows journalists and operators to access tickers remotely. It runs in any modern browser, on any device. The Ticker Web Client is hosted at the URL http://mse-host:8580/app/ticker/ where *mse-host* is the hostname or IP address of the Media Sequencer hosting the web application. The Ticker Web Client is a **companion app** to the desktop client. This means that the web client is not feature complete compared to the desktop client, but aims to provide the most used functionality to operate a ticker system on a daily basis. These are the main functions:

- Carousels (Pool and Active)
    - The Carousels
    - The Pool
    - The Active Carousel
- Adding and Modifying Elements
- Layout Editor
- Control, Handlers, Users
    - Control
    - Handlers
    - Users

6.1   Carousels (Pool And Active)

6.1.1   The Carousels

The tab list below the logo is populated with the carousels on this Ticker system. Click each tab to display a carousel.

6.1.2   The Pool

The left side of the window contains the pool of inactive elements for the carousel. New elements are added to this list by default, unless the Active carousel is set to default in Ticker Wizard. To activate an element, drag it over to the Active Carousel. The Pool can contain elements, sections and groups.

6.1.3   The Active Carousel

The right side of the window contains the Active Carousel. These are the elements of an active, running carousel. Each output channel has its own column in the Active Carousel. The default output channel is called Main. In this column, it is possible to track the cursor of elements:

- **Gray cursor:** The section having the active, playing element.
- **Red cursor:** The active, playing element.
- **Yellow cursor:** The next element, if manually set.

The checkbox in this column indicates whether the element is active. If not, the element is not displayed. The Number to the right of the checkbox indicates the number of times this element has run.

6.2   Adding And Modifying Elements

To add new content into the carousel, click one of the following buttons:

From left:

- Add a new element based on the default template for this carousel.
- Add a new element, and select from the available templates for this carousel.
- Add a group.
- Add a section.

A section is a group with two sub groups: Section heads and section elements. This is handy when a section should display a mandatory section element before the actual section elements. The section headers can be shown or hidden in the active playlist by right clicking a section and enabling or disabling **Show section headers**.

Set as next on... ▶

Enable on... ▶

Disable on... ▶

Copy

Cut

Paste

Delete

Expand all

Collapse all

⚪ Wrap long lines

⚪ Show section headers
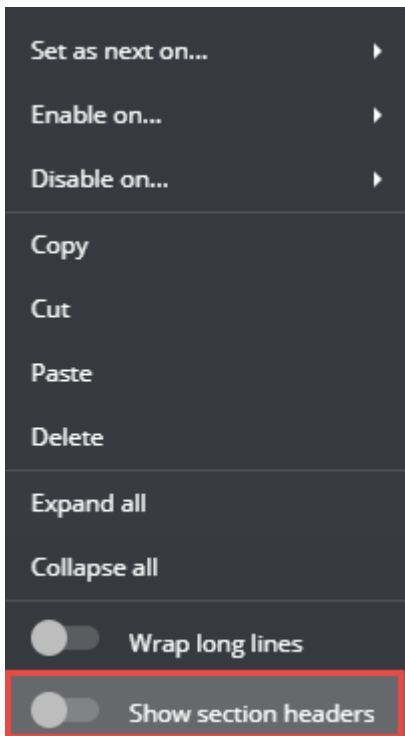
6.3 Layout Editor

When adding or modifying elements, the ticker element editor appears with a default layout. This layout is auto generated based on the fields in the template. It is possible to change this layout.

- Add a new element based on the template you want to change.
- Click the **Edit Layout** button in the lower right corner of the editor: [Edit layout]

Now you enter layout mode. In this mode it is possible to:

- Resize and move input widgets. The layout is column based to be responsive when the panel aspect and size change, so it is not possible to freely adjust size and position. The adjustments snap to an invisible grid.
- Edit label of input widgets.
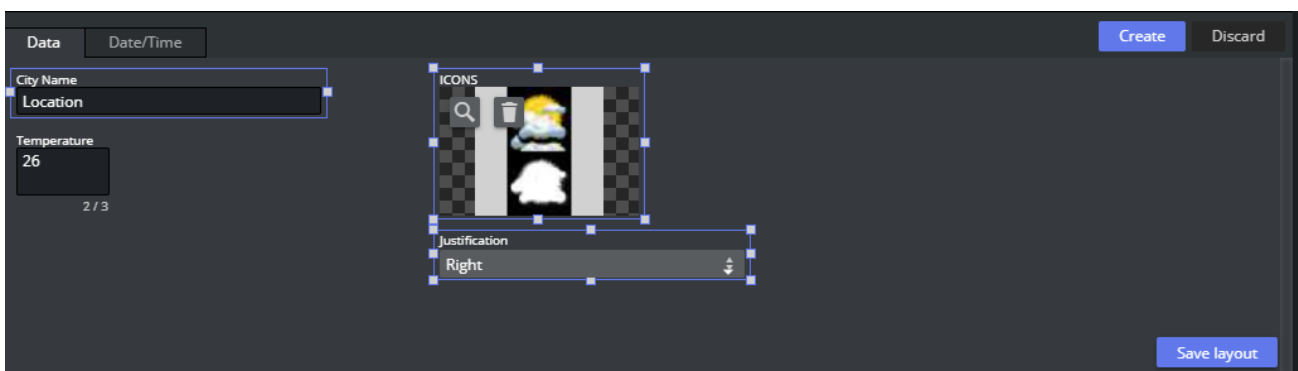
When done changing the layout, click **Save Layout**. To discard the layout changes, click **Discard**, The layout is now changed for all elements based on this template.

6.4   Control, Handlers, Users

These functions are for more advanced users.

6.4.1   Control

By clicking the **Control** button on the toolbar, a panel appears with all the Ticker buttons. The system buttons are as follows:

- For each output channel (default channel is called Main) there are on/off buttons for:
    - **Ticker System:** Enabling or disabling the Ticker system for this channel.
    - **Program:** Sending the background scene to the renderer, preparing the visual layout of the Ticker
- **For each carousel:** Starting and stopping each carousel with the animation designed by the scene designer
- **Custom:** In the Ticker Wizard it is possible to create new Button Groups with custom buttons. These buttons do not have a state (on or off), and they are displayed default in blue color.
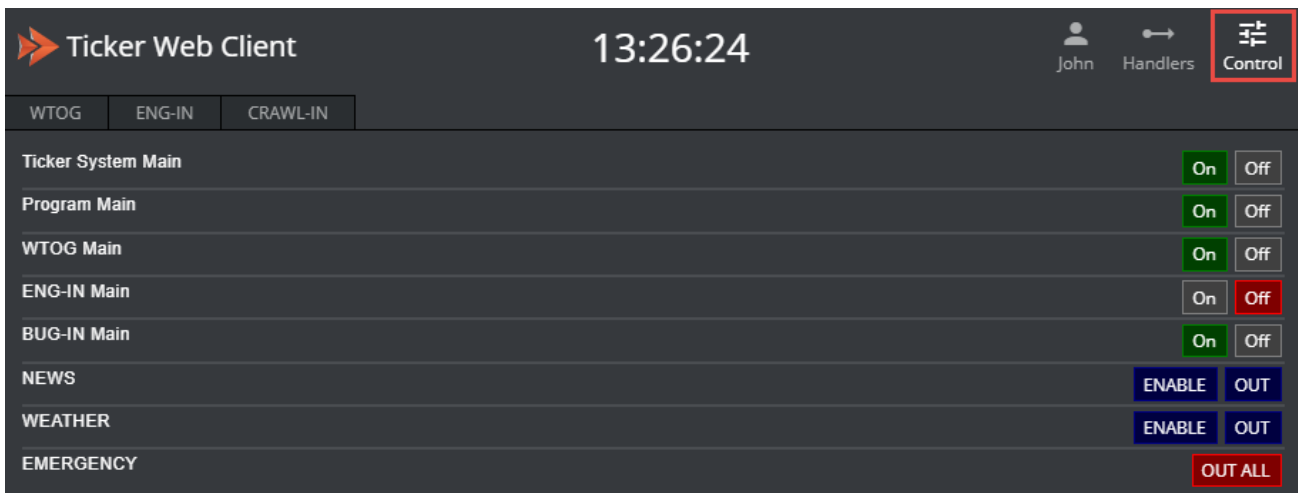
6.4.2   Handlers

This function must be enabled in **Ticker Wizard > Settings > Show in Main Control**:
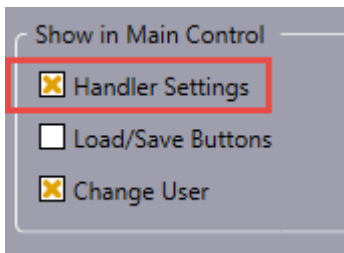
If enabled, a button on the toolbar appears. When clicking it, a dialog appears with the list of Media Sequencer handlers used by this Ticker system:
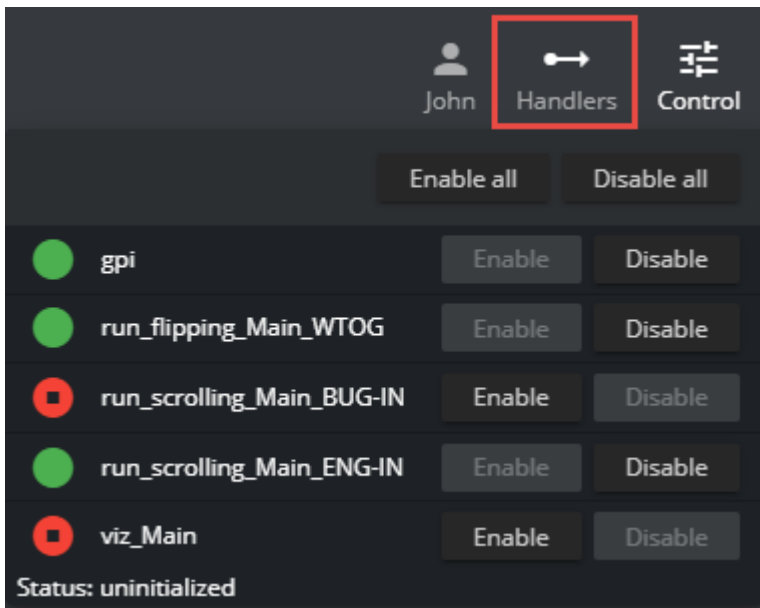
The Media Sequencer handlers are the mechanisms in use when the Ticker system is running. There are handlers for GPI trigging, carousel handlers and Viz Engine handlers. Normally these handlers should be enabled (green), but in some cases it may be necessary to restart a handler (disable and re-enable it). The handlers also provide a low level message of their state. This information and this functionality is usually needed only by system administrators.

6.4.3   Users

This function must be enabled in **Ticker Wizard > Settings > Show in Main Control**:
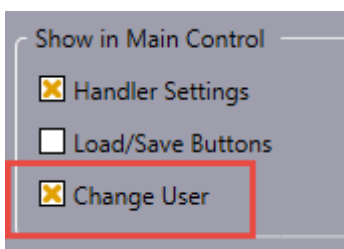
If enabled, the user name is editable from by clicking the user icon in the header menu. A dialog pops up prompting for a user name:
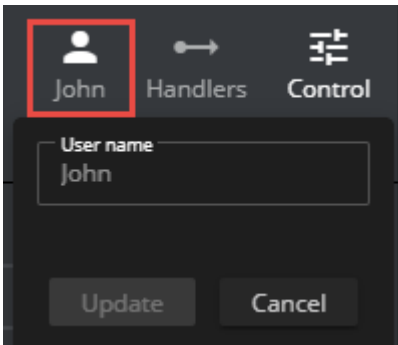
This user name is used to add a personal tag to each element, on who created it:
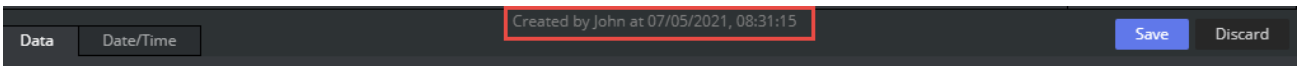
7   Design Conventions

For a ticker carousel to work properly, the Viz Artist scene that the carousel is based upon must be designed according to specific design conventions. Some design conventions apply for both scrolling and flipping carousels, such as the requirement to use control plug-ins and in/out-states in scenes. Other conventions are limited to affect only scrolling carousel scenes, containing the Scroller plug-in, or flipping carousel scenes, containing the Toggle plug-in.

Tutorial examples on how to apply the design conventions can be found in this manual, see the Creating a Scrolling Carousel and Creating a Flipping Carousel sections.

This chapter describes various design conventions that apply to ticker projects:

- Global Conventions
- Scrolling Carousel Conventions
- Flipping Carousel Conventions
- Logo Animations as Flipping Carousels

7.1   Global Conventions

> ✅ **Tip:** When naming scene identifiers (element source, layer identifier, state identifiers, and so on), strive to use alphanumeric characters only. Special characters should not be used, such as space, \ (backslash), / (forward-slash), < (less-than), > (greater-than), & (ampersand), ' (single-quote) and " (double-quote).

Ticker carousels are created in Viz Artist as scenes. The following design conventions apply for all carousels (both scrolling and flipping):

- **Control plug-ins:** Elements that should be modified in Viz Ticker Client must have control plug-ins attached. The Control Object plug-in must be added to the template root container, while additional Control plug-ins should be added to sub-containers.

  > ✅ **Tip:** If text should be edited, add the Control Text plug-in to the text container.

- **States:** Carousels must have one *out* state, and at least one *in* state. The in state means that the graphics are visible on-screen, while the out state means that the graphics are positioned outside the visual bounds. The scene can have multiple stop points that define the out state, but they must all be named *O*, as there is only one out state.

- **Horizontal Alignment:** The horizontal alignment of carousel templates can be set to either left, right, top, or bottom. Leftward scrolling should be left centered, rightward scrolling should be right centered, upward scrolling should be top centered, and downward scrolling should be bottom centered.

  > ✅ **Tip:** A ticker with Arabic text should typically be right centered, as Arabic is a right-to-left language.

See Also

- Scrolling Carousel Conventions
- Flipping Carousel Conventions

7.2   Scrolling Carousel Conventions

The following design conventions apply only for scrolling carousel scenes.

Scrolling carousel scenes are characterized by containing one or more *Scroller* plug-ins with belonging directors and stop points.

A scrolling carousel scene must be designed according to the following conventions:

- The *Layer* setting in the Scroller plug-in editor must be identical to the director holding the in and out stop points.
- The *Element Source* in the Scroller plug-in editor must be identical to the in state stop point of the layer director.
- The out state stop point of the layer director must be named the letter *O*. All other states are in state stop points.
- All template containers must be organized under a root container called *ticker_templates*. It is important to position the *ticker_templates* container at the top of the root level of the scene tree, as Viz Engine performs a depth-first search when identifying containers. This means that the system starts with the first root container, and searches through all of its sub-containers, before moving on to the second root container, and so on. Keeping template containers high up in the tree structure improves performance.
- Do not add any containers under the container holding the Scroller plug-in. The scroller generates its own tree structure with copies of the template containers.

See Also

- Global Conventions
- Flipping Carousel Conventions
- *Geometry plug-ins/Ticker* section in Viz Artist User Guide

7.3   Flipping Carousel Conventions

The following design conventions apply only for flipping carousel scenes. Flipping carousel scenes are characterized by containing one or more *Toggle* plug-ins with belonging directors and stop points.

A flipping carousel scene must be designed according to the following conventions:

- The master scene and the scene templates (object scenes) must be built as separate scenes.
- All flipping carousel template scenes must have a root container named *object*.
- The container holding the Toggle plug-in must have the same name as the director holding the in and out stop points.
- In the Control Object editor for each object scene, the Layer Identifier must be identical to the director holding the in and out stop points in the master scene.

- In the Control Object editor in the object scene, the *State Identifier* must be identical to an in-state stop point of the layer identifier director.

See Also

- Global Conventions
- Scrolling Carousel Conventions
- *Container plug-ins/Container* section in Viz Artist User Guide

7.4 Logo Animations As Flipping Carousels

To create a logo animation, create a flipping carousel. While a regular flipping carousel is equipped with multiple template elements, a logo animation is only equipped with one. The `On` and `Off` buttons in Viz Ticker Client animate the logo in and out.

8   Importing Ticker Example Scenes

This section shows you how to import Viz Ticker scenes into Viz Artist. A set of example scenes are bundled together with the Ticker Wizard installer. These scenes can be studied and used as a starting point for your own designs.

Also see the sections Creating a Scrolling Carousel and Creating a Flipping Carousel which describe how to create your own ticker carousel scenes in Viz Artist.

8.1   To Import Viz Artist 3 Example Scenes

1. Start Viz Artist 3.
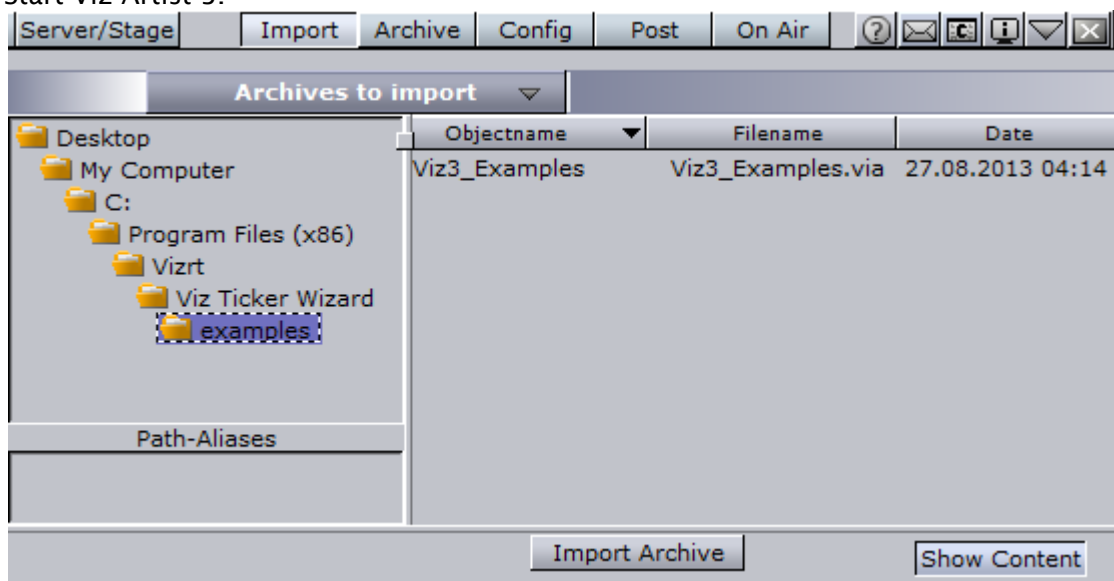
2. From the **Main** menu, click **Import**.
3. In the Import panel that appears, browse to the example files in the installation directory of Ticker Wizard. The files can be found at *<Installation-Directory>\examples*. See Installation Directories.
4. Make sure that *Archives to Import* is selected in the drop-down list at the top of the Import panel. The Import panel then shows all available archives that can be imported from the selected folder.
5. Select the *Viz3_Examples.via* archive.
6. Click the **Import Archive** button. The archive is imported and available from the list of scenes in Viz Artist. The archive contains various scenes; one flipping carousel example, and one scrolling carousel example.

8.2   To View The Scenes

1. From the **Main** menu, click **Server.**
2. Make sure that **Server - Scenes** is selected in the drop-down list at the top of the server area.
3. In the list of scenes, browse to view the project `Vizrt\Tutorials\Ticker`.
4. Double click one of the thumbnails of the scenes.

9 Creating A Scrolling Carousel

This section describes how to create a scrolling carousel scene in Viz Artist.

To create these carousel scenes, you need to know how to perform basic operations in Viz Artist, such as creating scenes, adding items, creating animations, and so on. For more information refer to the Viz Artist User Guide.

To help get started, a set of example scenes can be imported, see Importing Ticker Example Scenes.

9.1 Scrolling Carousels

What characterizes a scrolling carousel is that objects, most often text messages, enter the screen from one side, scroll through the screen, and then exit on the other side. A scrolling carousel scene contains both the design for the background and the templates that are used to create content for the ticker.

Scrolling carousel scenes must contain one or more Scroller plug-ins with belonging directors and stop points.

> ✓ **Tip:** If you do not want to perform these steps, scrolling carousel example scenes can be imported by following the instructions in To Import Viz Artist 3 Example Scenes. Note that the design in the example scenes are slightly different from the scene created here, so the name of the scenes, carousels, templates, and so on may vary.

This section contains information on the following topics:

- Creating a Scrolling Carousel Scene in Viz Artist
  - Getting Started
- Adding a Background
  - To Add a Rectangle
  - To Add Color
  - To Add the Scroller Plug-in
  - Scroller Properties
  - To Add a Director
  - To Add a Stop Point
  - To Add Another Stop Point
  - To Add an Animation
- Stopping, Starting, and Clearing the Carousel
  - To Add an Action
  - To Add a Keyframe
  - To Add Another Keyframe
- Adding a Template
  - To Add a Group
  - To Add Another Group
  - To Add the ControlObject Plug-in

9.1.1 Creating a Scrolling Carousel Scene in Viz Artist

When the tasks in this section have been carried through, this scrolling carousel scene includes a background, a scroller with a text message template, and state transitions for animating the carousel in and out.

Getting Started

> ✅ **Tip:** Remember to save scenes in Viz Artist regularly.

- · In Viz Artist, create a new scene named *tutorial-scrolling*.
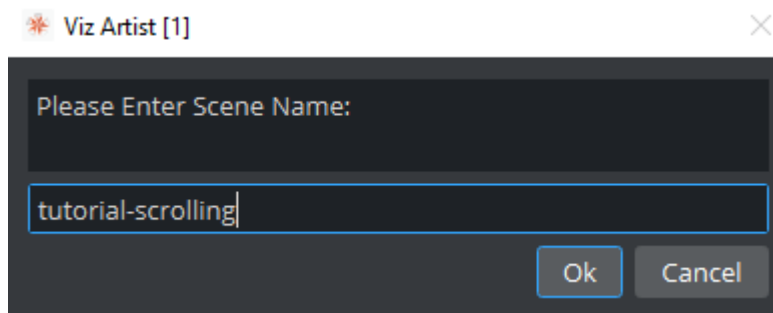
- · Open the new scene.
- · Add a group and rename the group container to *LayerContainer*.
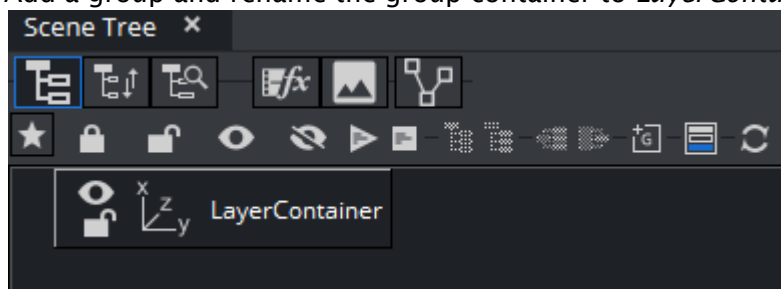
- · In the **Transformation Editor**, *Position* the new group to Y= `-170` (note the minus sign).
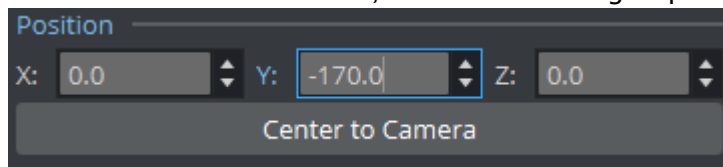
9.1.2   Adding a Background

To Add a Rectangle

The rectangle functions as the background for the scrolling ticker. The background can be any object, for example a circle, cube, sphere, and so on.

· Go to **Plugins** (/Built Ins) > **Geom Plugins** > **Default.**
· Add a **Rectangle** as a subcontainer of the *LayerContainer*.
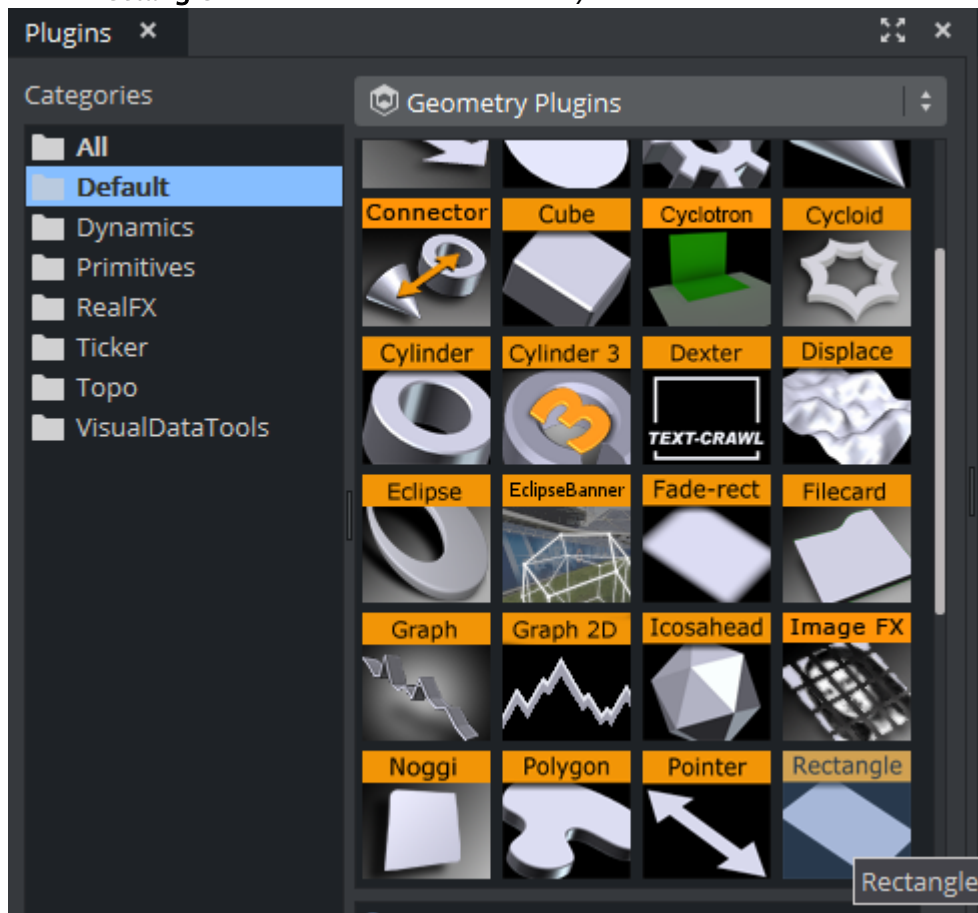
· Rename the rectangle container to *Background*.
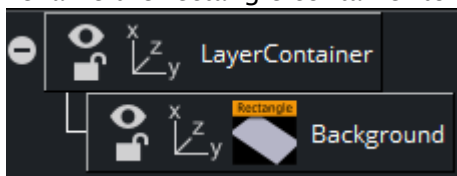
· Open the *Background* container's **Transformation Editor** and set the **Position** to Y= –10.
· Set the **Scaling** type to (*S)ingle* (so that each axis can be scaled by itself).
· Set the **Scaling** to X= 8, Y= 0.5, and Z= 1.

To Add Color
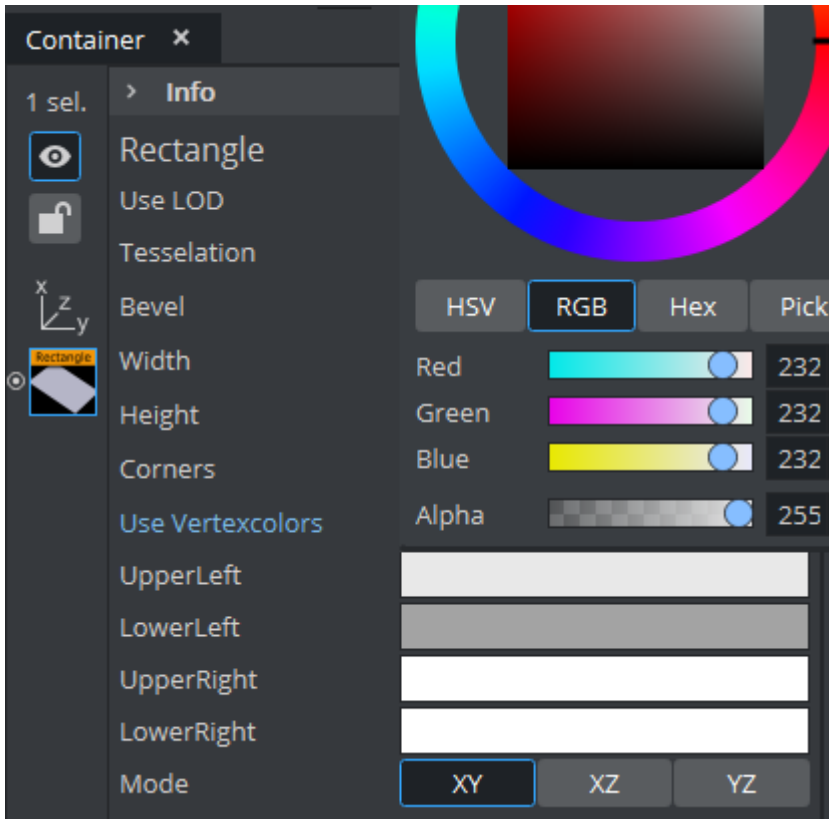
- Open the **Rectangle Editor** and enable **Use Vertexcolors**.
- Add the following colors:
    - **UpperLeft:** RGB= `232, 232, 232, 255.`
    - **LowerLeft:** RGB= `163, 163, 163, 255.`
    - **UpperRight:** RGB= `255, 255, 255, 255.` *(default value).*
    - **LowerRight:** RGB= `255, 255, 255, 255.` *(default value).*

The background then has the following appearance:

To Add the Scroller Plug-in

1. Go to **Plugins** (/Built Ins) > **Geom Plugins** > **Ticker.**
2. Add a **Scroller** plug-in as a subcontainer of the *LayerContainer*.

> ⚠ **Note:** The Scroller plug-in is a Viz Artist geometry that generates a dynamic line of scrolling elements, where each element is an instance of a template.

> ⚠ **Note:** The scroller container name must be different from the **Element source** in the **Scroller Editor**.
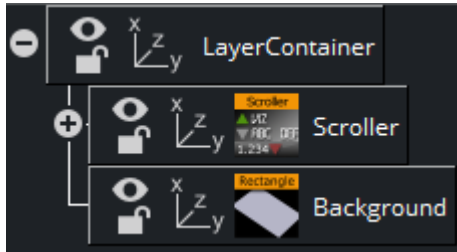
3. Open the *Scroller* container's **Transformation Editor** and set the **Position** to Z= 1. This positions the scroller slightly in front of the background to make sure that the scroller

content becomes visible on screen.
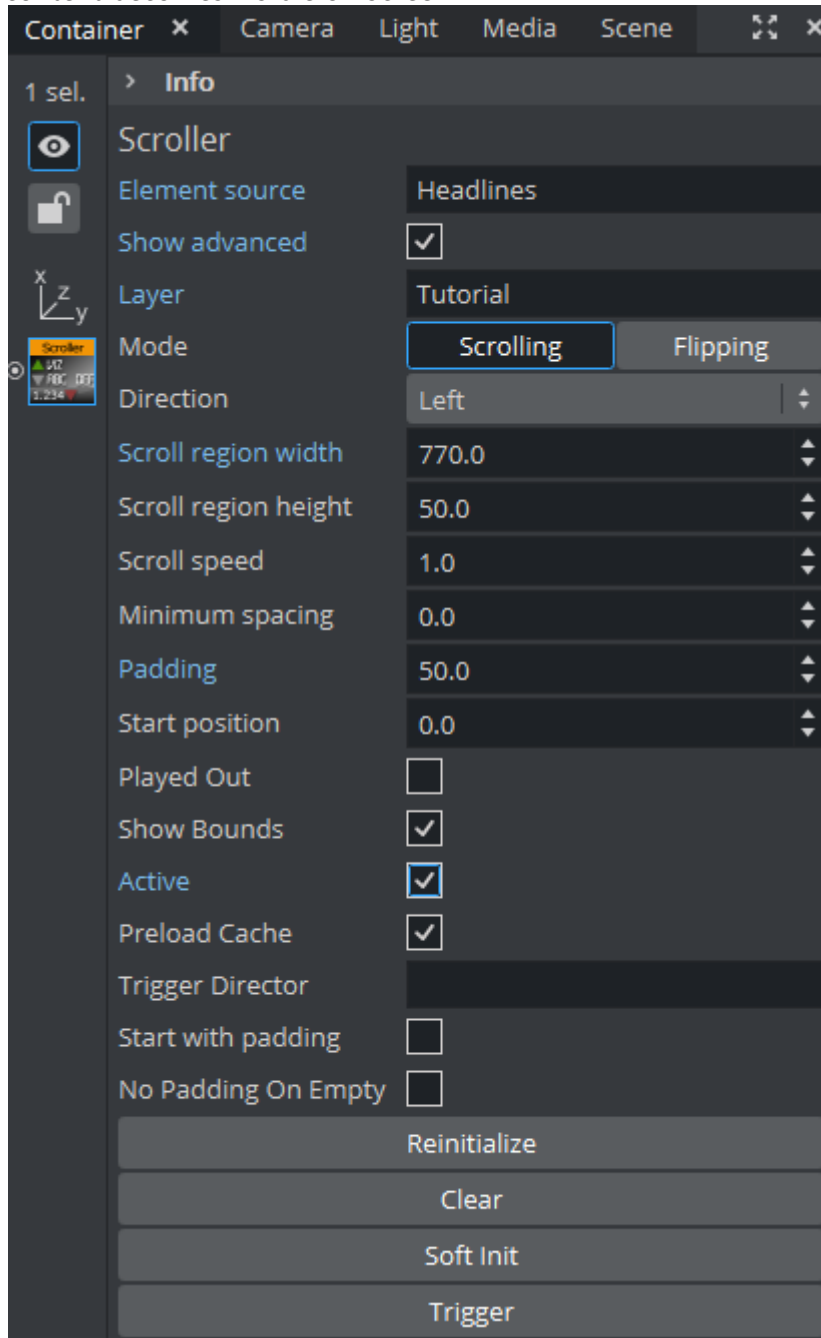
4. Open the **Scroller Editor**, and set the following:
   - **Element source:** `Headlines`. (Note that the *Element* source must be identical to the *in* state stop point of the layer director (not created yet)).

   > ✅ **Tip:** The element source should consist of alphanumeric characters only. For more details, see Global Conventions.

   - **Show advanced:** `On` (enabled).

- **Layer:** `Tutorial`. (Note that the *Layer* name must be identical to the name of the layer director (not created yet)).

  > ✅ **Tip:** The layer name should consist of alphanumeric characters only. For more details, see Global Conventions.

- **Scroll region width:** `770`. (For output format 16:9, a scroll region width of *770* puts the scroller bounds outside the screen area. This maintains the illusion of a constantly running scroller. For output format 4:3, the scroll region width should be at least *600*).
- **Padding:** `50` (adds space between messages).
- **Active:** `On` (enabled - activates the scroller).

Scroller Properties

- **Element Source:** Element descriptions are added here. Descriptions must be entered in the element source syntax, or in the name of a Shared Memory segment that uses the Shared Memory protocol. When designing and testing a scene, the element source syntax is used to make the system scroll instances of the design containers.
- **Show Advanced:** Shows additional parameters. Defaults to `Off`.
- **Layer:** Defines the layer name for the scroller. The layer name must be identical to the name of the director that holds the in and out points for the carousel.
- **Mode:** Defines if elements should be shown as scrolling or flipping elements for scrolling and flipping carousels, respectively.
- **Direction:** Defines the direction of the scroller.
- **Scroll Region Width:** Defines the width of the scroller. This option is only relevant for left-to-right/right-to-left scrollers.
- **Scroll Region Height:** Defines the height of the scroller. This option is only relevant for up-down/down-up scrollers.
- **Scroll Speed:** Defines the speed of the scroller. The *x* position of the scrolling items are decremented with this value for each frame that passes.
- **Minimum spacing:** Sets the minimum spacing between scrolling elements.
- **Padding:** Defines an extra padding value that is added to the width of the bounding box of an item when calculating the spacing between the items. Negative padding can be used to make items overlap.
- **Start Position:** Sets the initial position as a percentage value in the range `0-100`. A value of `100` means the scroller starts fully populated with content. The default value is `0`.
- **Played Out:** Tracks movement of scrolling element and notify ticker service on specified position.
  - **Element:** Position of the scrolling element to be checked with the marker. When set to *Beginning hits*, notify when beginning of element hits the marker. When set to *End leaves*, notify when end of element leaves the marker.
  - **Marker:** Position of the marker related to side of Scroll region.
- **Show Bounds:** Sets the area in which the scroller tries to maintain the illusion of a continuously scrolling flow of items is visible on-screen when enabled.

- **Active:** Makes scrolling or flipping available when enabled.
- **Preload Cache:** Moves all spike-load related to item creation to the initialization phase. Use this on configurations with no ringbuffer. Designs named *DESIGN_designname_Xnnn* have *n nn* instances created in advance, where *nnn* is an integer.
- **Trigger Director:** Triggers director when the **Trigger** button is pressed.
- **Start with padding:** Sets padding for the first scrolling item, for example when **Start Position** is set to `100%` or the scroller container is hidden at the beginning.
- **No Padding On Empty:** Does not create any extra spacing between elements for elements with zero bounding box.
- **Reinitialize:** Reinitializes the plug-in by deleting all cached objects and reconstructing the initial node structure. This parameter can be used to update the scroller after changing for example the item source.
- **Clear:** Clears all current ticker items in the rendering window.
- **Trigger:** Triggers the director named in the Trigger Director field provided that there are items in queue to be scrolled in. If no director is specified, the scroller is activated.

To Add a Director

- In the **Stage Editor**, add a **Director** and name it *Tutorial*.
  The name of the director is identical to the *Layer* name in the **Scroller Editor**.
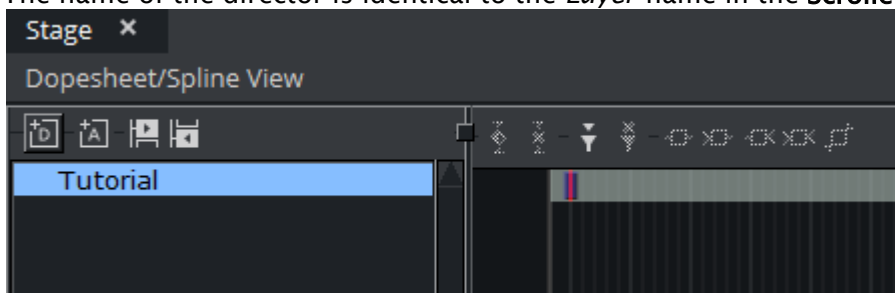
> ✅ **Tip:** The director name should consist of alphanumeric characters only. For more details, see Global Conventions.

To Add a Stop Point

The director must have at least two stop points, one for the *in* state (shown on screen) and one for the *out* state (hidden from screen). Viz Ticker Client creates control buttons based on these stop points. The `On` button, based on the in state stop point, takes the carousel on air. The `Off` button, based on the out state stop point, takes the carousel off air.

To equip the director with two stop points, perform the following steps:

1. Make sure that the *Tutorial* director is selected, and the **Timeline** is set to 0.
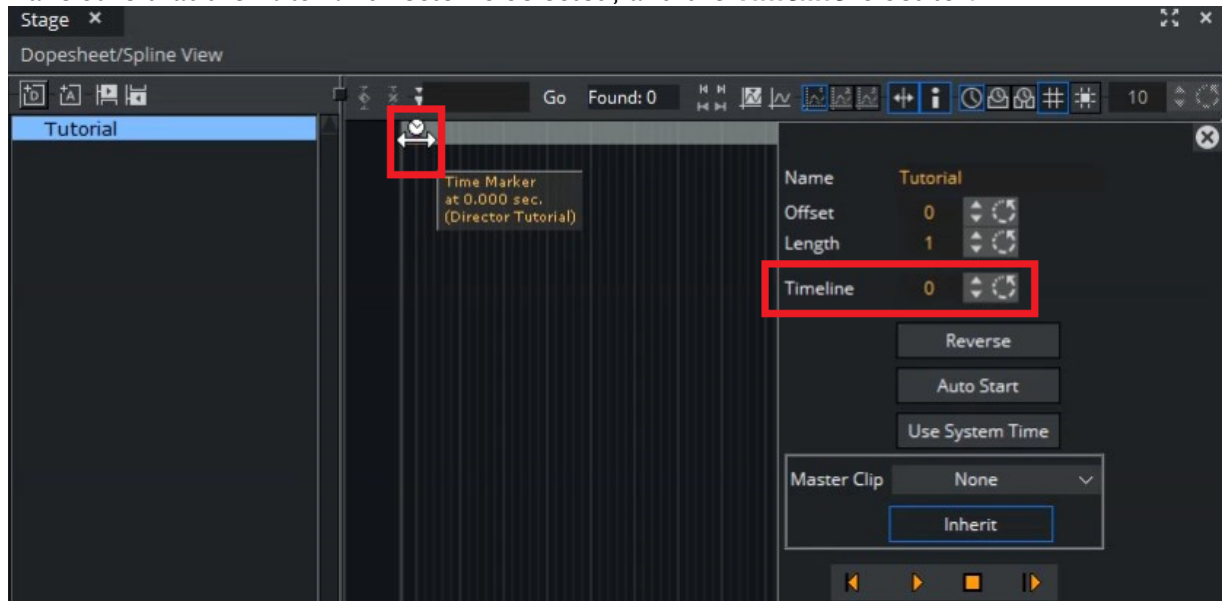
2. Click the **Add Stop** button.
3. Select this stop point in the **Stage Editor**,
4. In the **Name** field in the **Stop Point** editor enter the letter *o* (capitalized - this is a standard design convention for the out state).
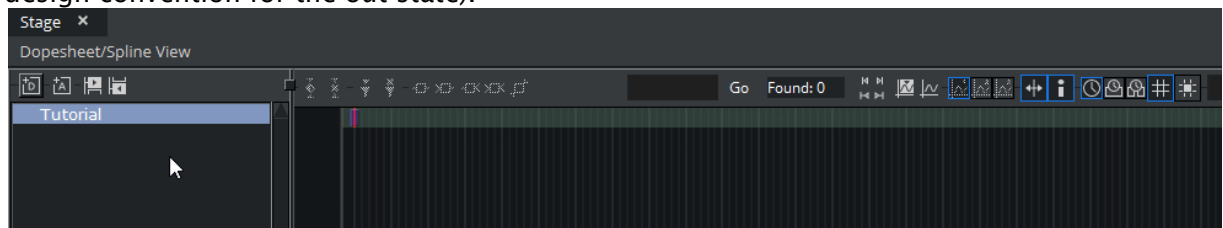
To Add Another Stop Point

- Select the *Tutorial* director, and set the **Timeline** to 50.

- Click the **Add Stop** button again, and name the new stop point *Headlines*. As the *O* stop point is automatically the out state of the scene, this new stop point is the in state.
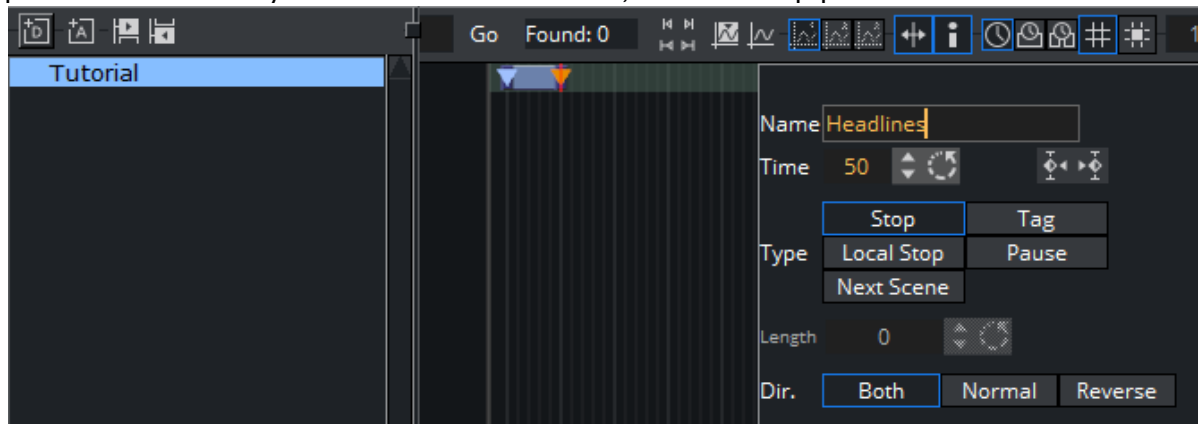
> ⚠ **Note:** The *in* state stop point name is identical to the **Element source** in the Scroller editor seen above.

To Add an Animation

To create a simple in/out animation, perform the following steps:

1. Select the *LayerContainer* in the Scene tree.
2. Make sure the Timeline is at position 50 in the Tutorial director, and that the Tutorial director is selected.
3. Click the **Set key frame** button in the Animation bar in the Scene Editor.
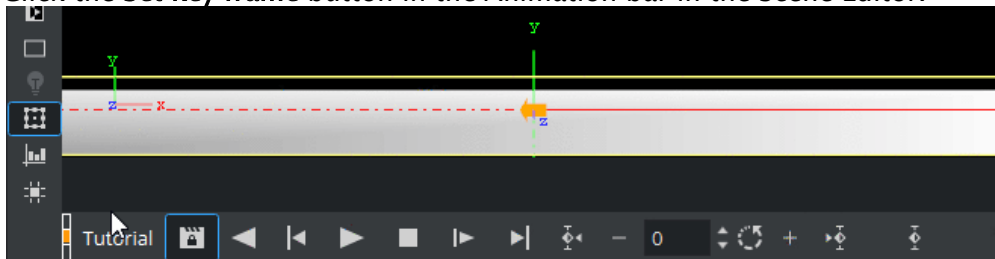
4. Open the *LayerContainer*'s **Transformation Editor** and set the **Position** to Y= -240 (note the minus sign).
5. Set the **Timeline** to *0*, and click the **Set key frame** button again.
   In the **Stage Editor**, a new animation has appeared, and the animation can be played.
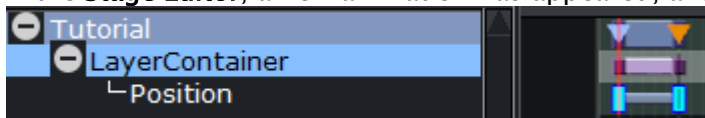
6. Click the **Start** and then **Continue animation** buttons to reset the scene to its *out* state, and to run the scene to its *in* state. If the previous tasks have been carried out correctly, a horizontally oblong rectangle object slides upwards into the screen's lower area.

At this point, the scene tree should have the following appearance:
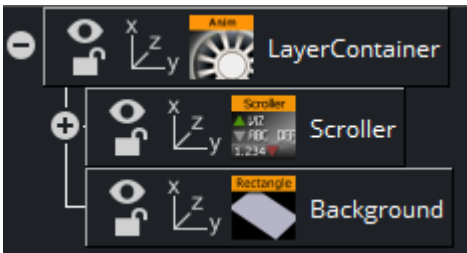
9.1.3   Stopping, Starting, and Clearing the Carousel

When taking a scrolling carousel off air, for example at a commercial break, the messages in the carousel should stop scrolling.

To Add an Action

- In the **Stage Editor** select the *Tutorial* director.
- Click the **Add Action** button and name the new action *start_stop_clear*.
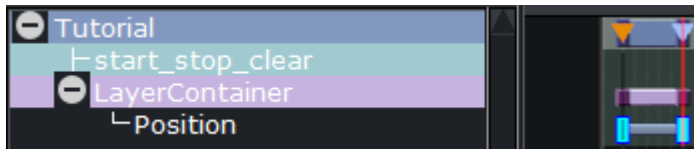
To Add a Keyframe

- Select the new action in the director tree.
- Set the timeline to 1, and click the **Add a key frame** button in the **Stage Editor**.
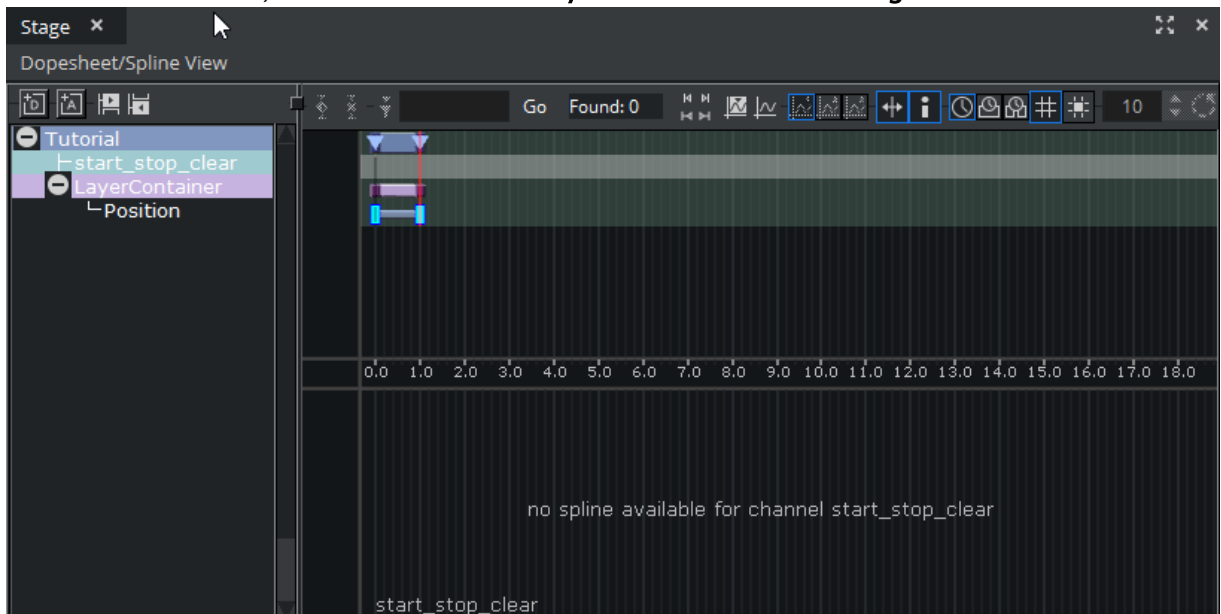
A new keyframe is then created close to the time of the scroller's out state. The action that is performed on this keyframe is not visible On Air as it is only one field away from the out state. This keyframe makes sure that the carousel is stopped when it is taken off air, so that

the next message in the carousel after the break is the correct one. If the carousel is not stopped, the carousel continues to run during the break, and then the first message after the break is a random one, according to how long the break has lasted.

· In the **Keyframe Editor**, change the **Name** of the new keyframe to *stop*.
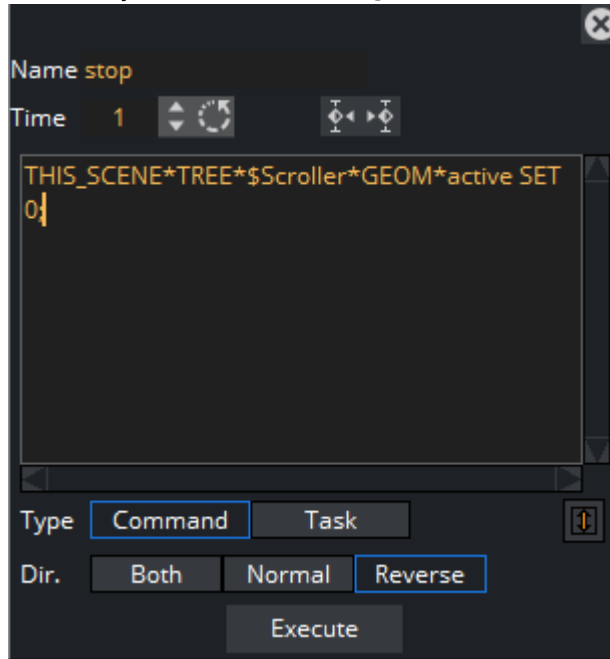
· Make sure the **Time** is set to `1`.
· In the big text field, type: `THIS_SCENE*TREE*$Scroller*GEOM*active SET 0`; **This command** disables the **Active** option in the **Scroller Editor** and stops the scroller.
· Click the **Reverse** button. This option disables the **Active** option in the **Scroller Editor** only when the timer slides leftward.

To Add Another Keyframe

· Set the timeline to `2`.
· Select the *start_stop_clear* action in the director tree.
· Click the **Add a key frame** button. Another keyframe is created close to the time of the scroller out state. This keyframe makes sure that the carousel is started when it is taken On Air.

- In the **Keyframe Editor** that appears, type a **Name**, for example *start_clear*.
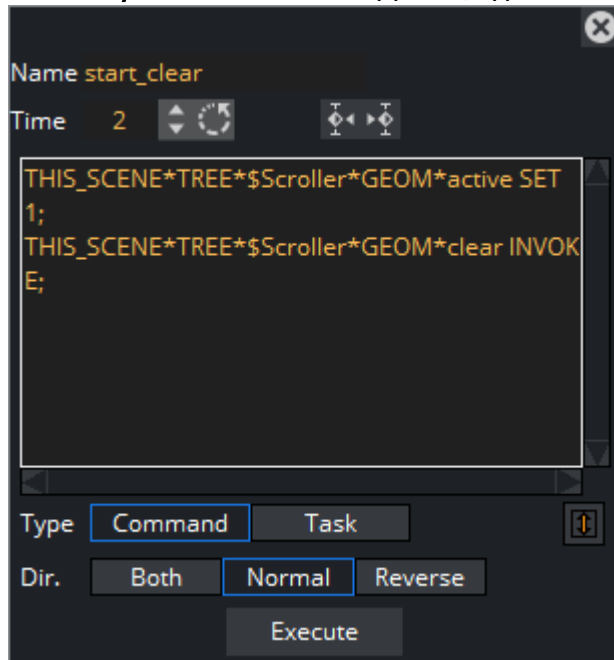
- In the big text field, add: `THIS_SCENE*TREE*$Scroller*GEOM*active SET 1;` This command enables the **Active** option in the **Scroller Editor** and starts the scroller.
- In the big text field, also add: `THIS_SCENE*TREE*$Scroller*GEOM*clear INVOKE;`. This command invokes the **Clear** option in the **Scroller Editor**, and clears all current messages in the scroller.
- Click the **Normal** button. This option enables the **Active** option in the **Scroller** editor only when the timer slides rightward. The background is now finished.

9.1.4 Adding a Template

The tasks in this paragraph create a template for messages that should be put on the scroller.

To Add a Group

- Add a new **Group** to the scene tree root, at the *top* of the scene tree.
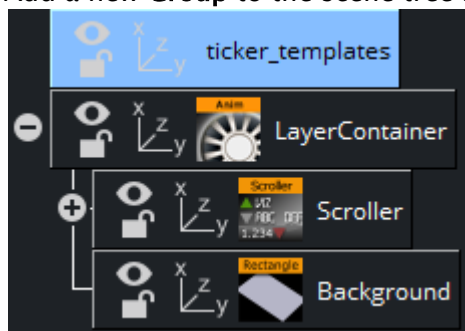
- Rename the new group to *ticker_templates*.

> ⚠ **Note:** All template containers must be subcontainers of this template root container. It is important to position the *ticker_templates* container at the top of the scene tree, above the *LayerContainer*, as the Scroller plug-in caches design. When the scroller needs to cache new design, it searches through the scene tree from the top and downward, and selects the first match it finds. By making sure that all the templates are located above the cache, it is assured that the copied design is originally from a template, and not from a design that is already in the cache.

To Add Another Group

· Add a new **Group** as a subcontainer of the *ticker_templates* container.
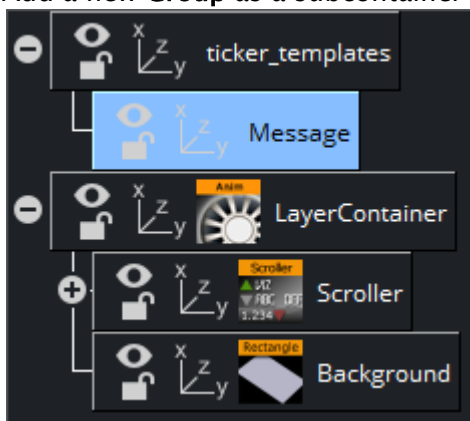
· Rename the new group to *Message.*

To Add the ControlObject Plug-in

· Go to **Plugins** (/Built Ins)> **Container Plugins** > **Control.**
· Drag the **ControlObject** plug-in (sorted simply as "Object") to the *Message* container.
  Any template that should be edited in Viz Ticker must be equipped with a *ControlObject* function on the template root container. As the scroller in this scene example rolls text messages across the screen, the template must be equipped with a control function, or else the text would be identical each time a new instance of the template was put on the scroller.
· Open the *Message* container's ***Transformation Editor** and set the **Position** to Y=* -170 (note the minus sign), Z= 1. When this scene is played in Viz Engine, the template is positioned on the scroller independently of the template's actual position. The template position is only set here to provide an indication of what the finished scene is going to look like. As the template's origin is positioned at the right end of the scroller's *x*-axis, it is only necessary to set the *Z* and *Y* coordinates identically to the scroller coordinates to see how it appears.

The scene tree should now have the following appearance:
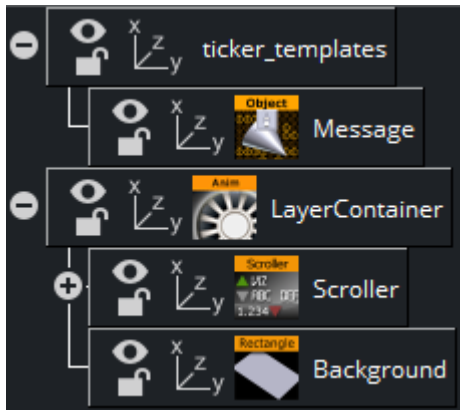
To Add a Font

- From the **Server** dock, select the **F** tab (fonts).
- Find a font and add it as a subcontainer of *Message*.
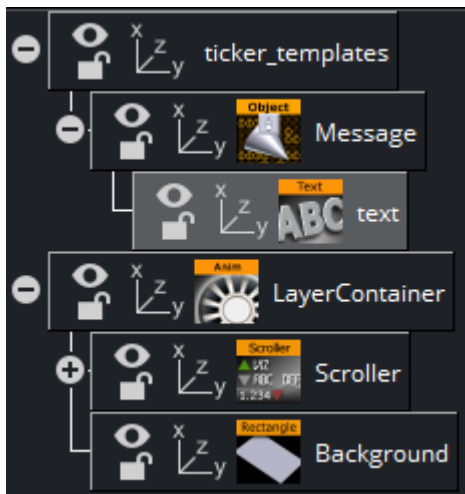
- Rename the new font container to *Text*.
- Open the *Text* container's ***Transformation Editor** and set the **Scaling** to X,Y,Z=* `0.33`.

To Add a Material

- From the **Server** dock select the **M** tab (materials).
- Add a material, for example orange R= `255`, G= `102`, B= `0` , to the *Text* container.

To Add the ControlText Plug-in

- Go to **Plugins** (/built Ins)> **Container Plugins** > **Control.**
- Drag the **ControlText** plug-in (sorted simply as "Text") to the *Text* container.

> ⚠ **Note:** A font template container must hold the *ControlText* function.

- In the **ControlText Editor:**
    - Disable **Expose kerning**.
    - Disable **Use formatted text**.
    - Type *Hello there!* in the **Input value** text box.
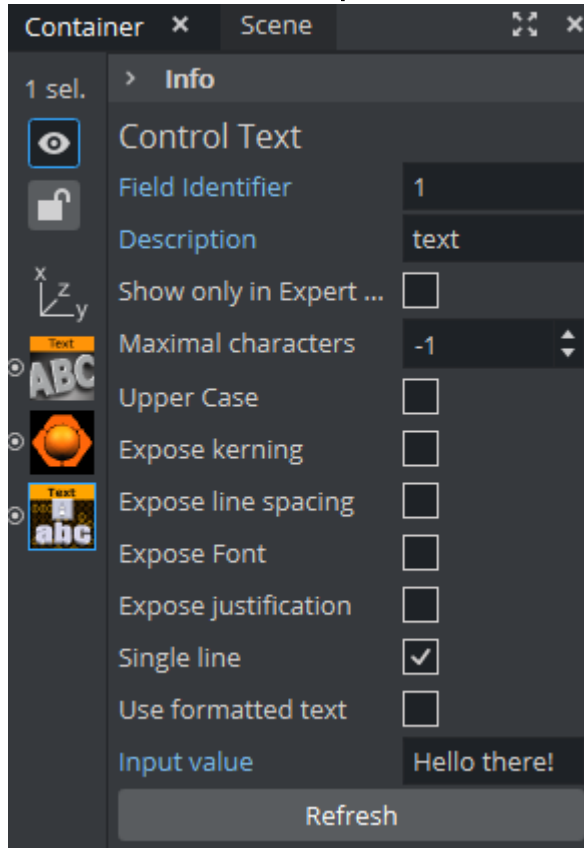
To Reposition the Text

The text message is not placed correctly on the scroller. Reposition it by performing the following steps:

- Open the *Message* container's **Transformation Editor** and set
    - **Axis Center** Y= C(enter).
    - **Position** Y= -170.

· **Axis Center** X= `L(eft)`.
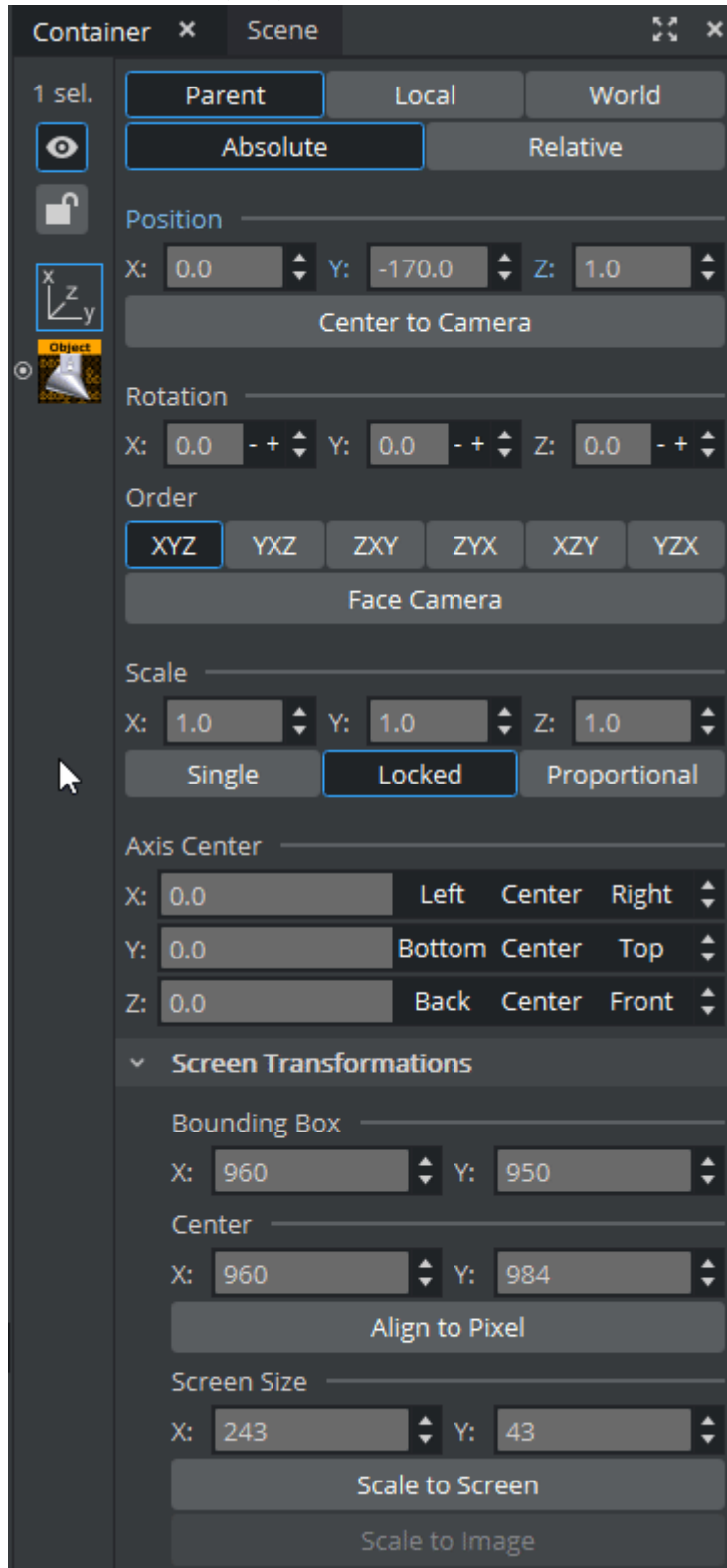
As this template enters the scroller from the right and exits on the left side, it is now accurately left centered.

> ⚠ **Note:** Leftward scrolling should be left centered (X=L button), rightward scrolling should be right centered (X=R button), upward scrolling should be top centered (Y=T button), and downward scrolling should be bottom centered (Y=B button).

To Hide the Template

- Click the eye symbol ▣ on the *ticker_templates* container. The symbol shows a closed eye ▨, which represents a hidden template. If the template had not been hidden, it would appear on the screen in its default shape (with the "Hello there!" text) during playout.

9.1.5 Saving the Scene

The scrolling carousel scene is now finished.

- Set the scene **Timeline** to `0`.
- Click the **Save** button.

10 Creating A Flipping Carousel

This section describes how to create flipping carousel background and template scenes in Viz Artist, and how to set up these scenes in Ticker Wizard and Viz Ticker Client. To create these carousel scenes, you need to know how to perform basic operations in Viz Artist, such as creating scenes, adding items, creating animations, and so on. For more information refer to the *Viz Artist User Guide.*

To help get started, a set of example scenes can be imported, see Importing Ticker Example Scenes.

10.1 Flipping Carousels

Flipping carousel scenes are characterized by containing one or more Toggle plug-ins with belonging directors and stop points. The elements in a flipping ticker are flipped in a static position. A flipping carousel is made up from multiple scenes; one that contains the design for the background, and another one that contains the templates that are used to create content for the ticker.

> ✅ **Tip:** If you do not want to perform these steps, flipping carousel example scenes can be imported by following the instructions in To Import Viz Artist 3 Example Scenes. Note that the design in the example scenes are slightly different from the scenes created here, so the name of the scenes, carousels, templates, and so on may vary.

10.1.1 Creating a Flipping Carousel Background Scene in Viz Artist

When finished with the tasks in this section, the flipping carousel background scene includes an animated cube object and state transitions for animating the background in and out of the scene. The Toggle plug-in makes it possible to add text message template elements to the scene.

10.1.2 To Get Started

Start Viz Artist 4.

10.1.3 To Create a New Scene
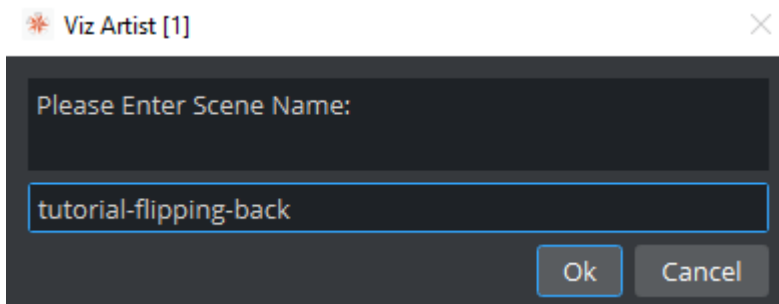
- Create a new scene named *tutorial-flipping-back*.

- Open the new scene.

10.1.4 To Add a Cube

- Drag a **Cube** to the scene tree (**Plugins > Geometry Plugins > Cube**).
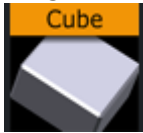
- Rename it to *background*.
- Open the *background* container's **Transformation Editor** and set the **Position** to X,Y= 180.
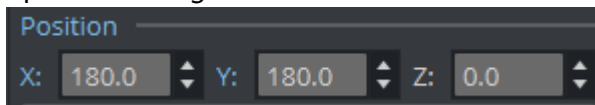
- Open the **Cube Editor**, and set the **Size X** to 160, **Size Y** to 30, and **Size Z** to 9.
- Set the **Bevel** to 2.
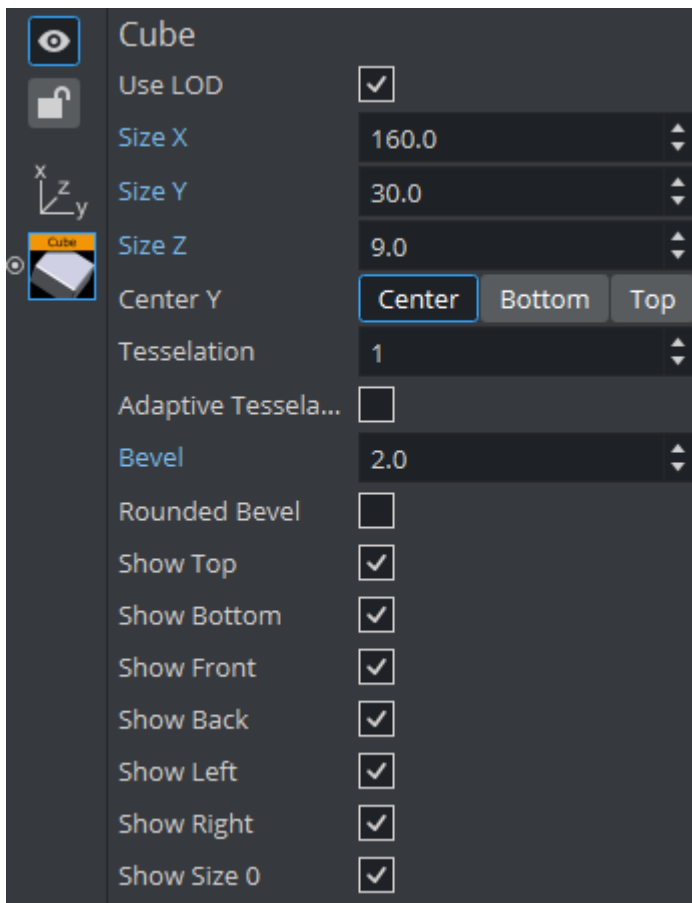
10.1.5   To Add a Material

· Add a material, for example grey (R= 235, G= 237, B= 255), to the *background* container.

At this point, the output of the scene has the following appearance:
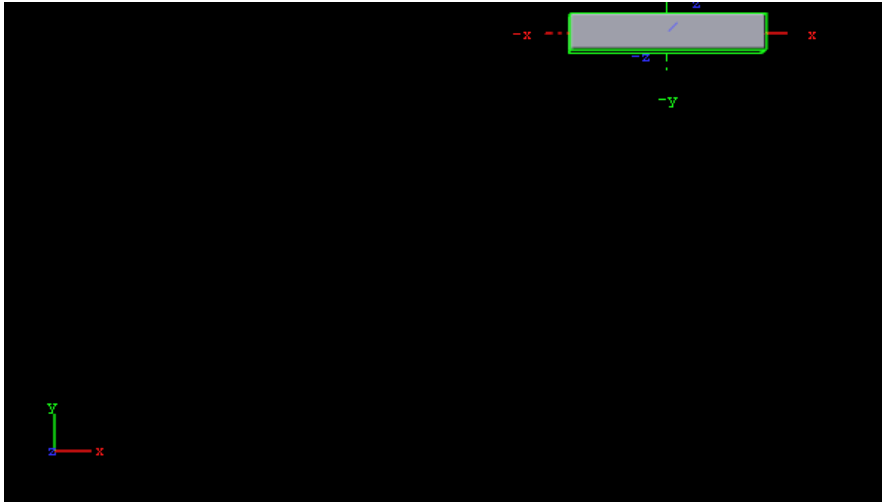
10.1.6   To Add a Director

- In the **Stage Editor**, click the **Add Director** button:
- Rename the director to *Tutorial*.

> ✅ **Tip:** The director name should consist of alphanumeric characters only. For more details, see Global Conventions.

10.1.7   To Add a Stop Point

- Make sure that the *Tutorial* director is selected, and the **Timeline** is set to 0.

- Click the **Add Stop** button:

- Select the stop point that appears in the **Stage Editor**.
- In the appearing **Stop Point** editor enter the letter O (capitalized) to the **Name** field.
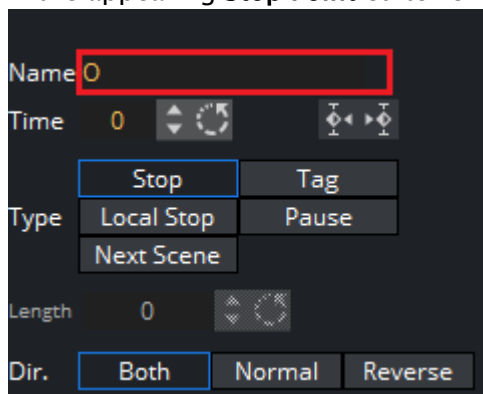
  - *O* is a standard design convention for the out state.

10.1.8   To Add Another Stop Point

- Select the *Tutorial* director, and set the **Timeline** to `50`.
- Click the **Add Stop** button again, and name the new stop point *Headlines*. As the *O* stop point is automatically the *out* state of the scene, this new stop point is the *in* state.
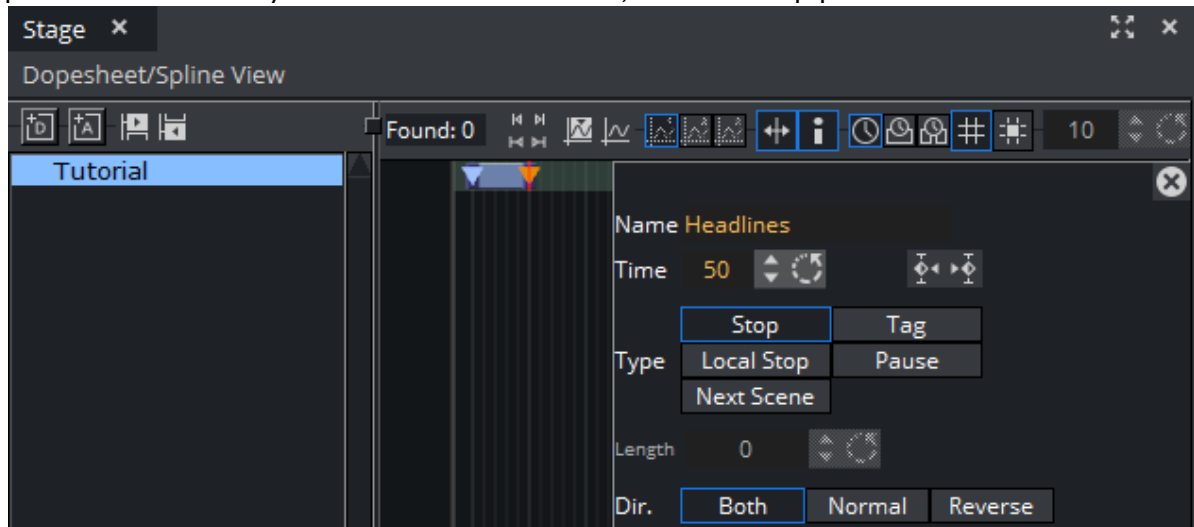
10.1.9   To Add Keyframes

To create a simple in/out animation, perform the following steps:

- Select the *background* in the Scene tree, and *Tutorial* in the Stage.
- Make sure the Timeline is at position `50`.

- Click the **Set key frame** button in the **Animation** bar in the **Scene Editor**.
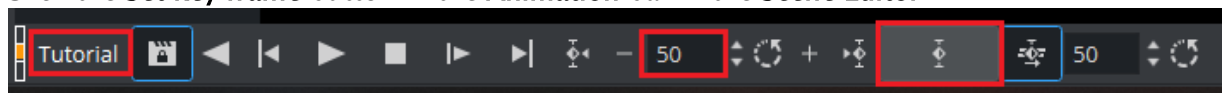
The second part of the animation, is to add a second key frame

- Set the **Timeline** to `0`.
- Open the *background* container's **Transformation Editor** and set the **Rotation** to X= `180`.

- Set the **Position** to Y= 240.
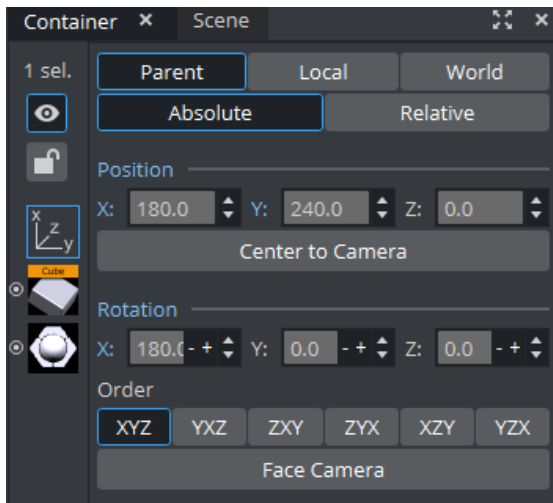
- Click the **Set key frame** button again. In the **Stage**, a new animation appears.
- In the Director tree, make sure that the new *background* animation is placed inside the *Tutorial* director. The Director tree now looks like this:
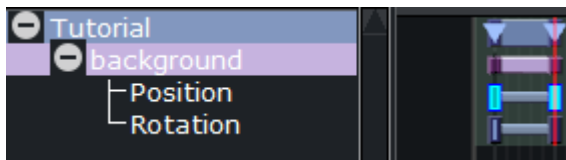
The animation can now be played.

- Click the **Start** and then **Continue** buttons to reset the scene to its *out* state, and to run the scene to its *in* state.

If the previous steps have been carried out correctly, a horizontally oblong cube object rotates downwards into the screen's upper right corner.
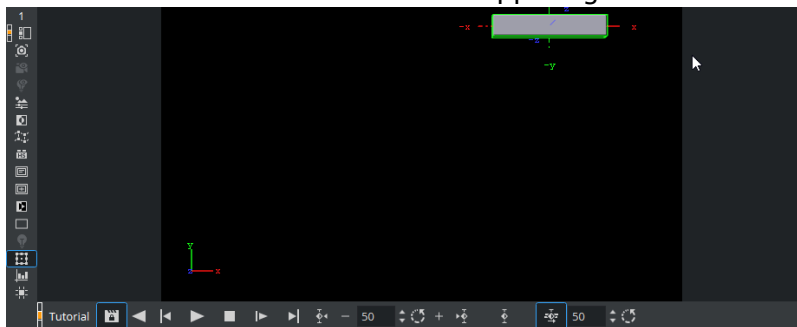
At this point, the scene tree still only contains one node:

10.1.10   To Add a Group

- Add a **Group** to the scene tree, and place it as a subcontainer of *background*.
- Rename the new group to `Tutorial`.
- Open the *Tutorial* container's ***Transformation Editor*** *and set the* ***Position*** *to X=* `-75` (note the minus sign), Z= `17`.
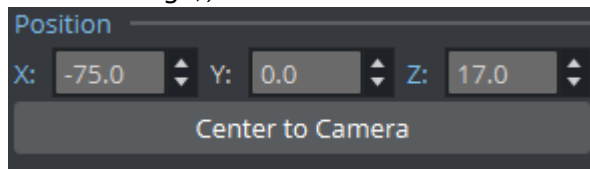
10.1.11   To Add the Toggle Plug-in

- In the **Plugins** (/Built Ins) dock, go to **Container Plugins > Container.**
- Drag the Toggle plug-in onto the *Tutorial* container.
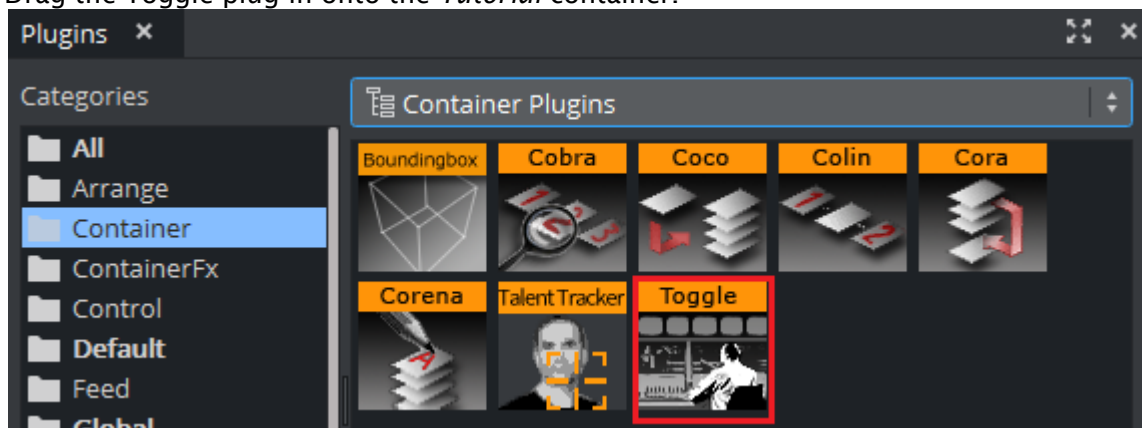
Four child containers are automatically added to the *Tutorial* container. The scene tree now looks like this:
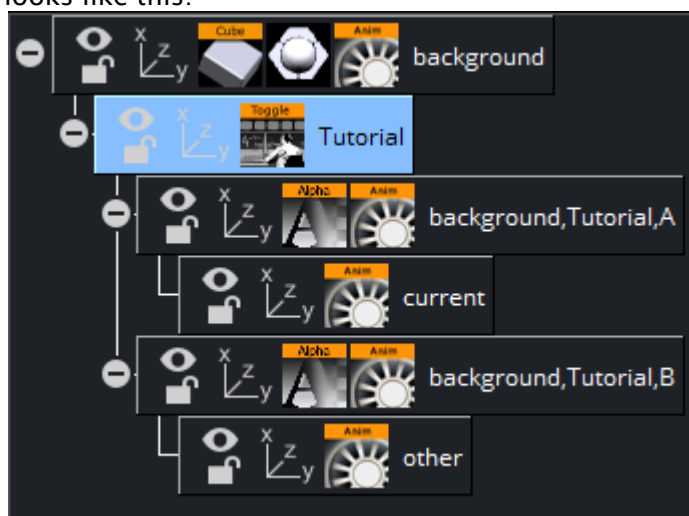
> ⚠️ **Note:** The container holding the Toggle plug-in must have the same name as the director holding the stop points.
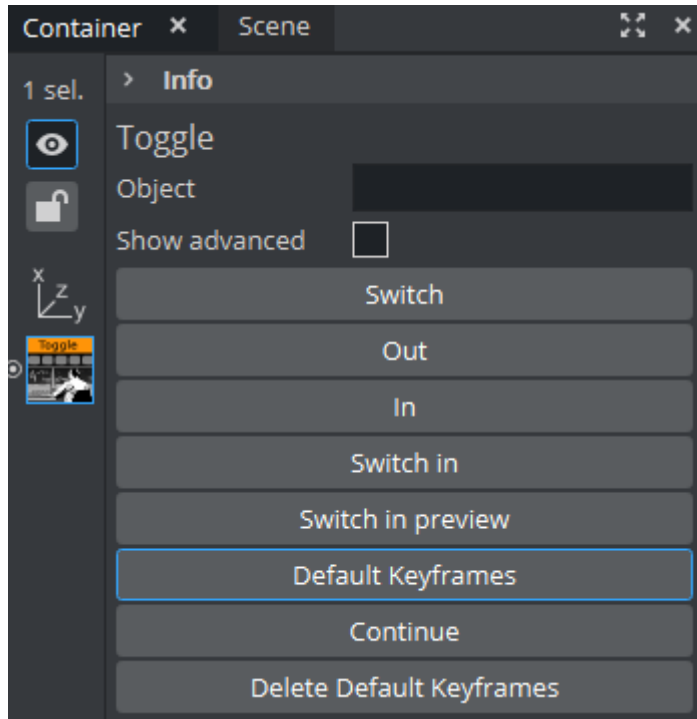
• In the **Toggle Editor**, click the **Default Keyframes** button. This adds additional keyframes in the **Stage Editor**. The **Stage Editor** now has the following appearance:
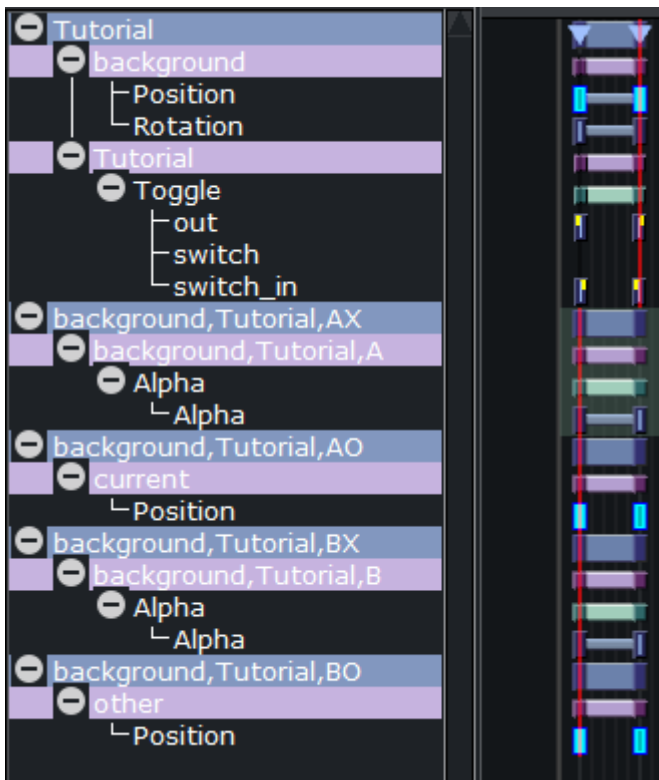
10.1.12 To Save the Scene

The flipping carousel background scene is now finished.

- Make sure that the **Timeline** is set to 0 on all directors.
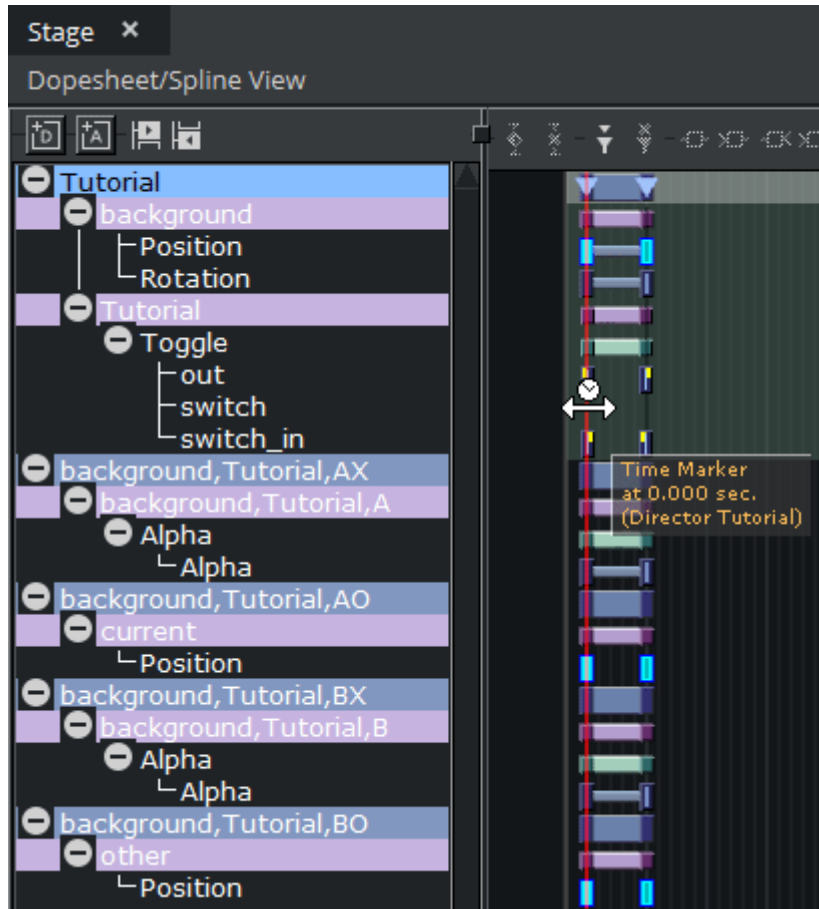
- Click the **Save** button.

See Also

- Ticker Plug-ins
- Flipping Carousel Conventions

10.2 Creating A Flipping Carousel Template Scene In Viz Artist

This section describes how to create a flipping carousel template scene in Viz Artist.

> ✅ **Tip:** A copy of the flipping carousel template scene has already been created and can be imported by following the steps in To Import Viz Artist 3 Example Scenes.

When finished with the steps in this section, this template scene is used by the flipping carousel background scene. The Toggle plug-in in the background scene makes it possible to add text message templates to the carousel.

10.2.1   To Get Started
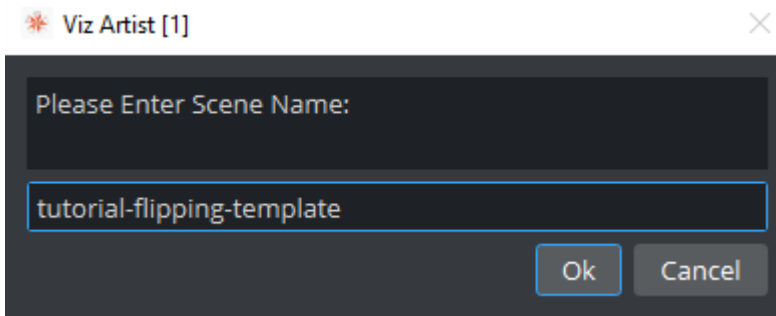
- Create a new scene named *tutorial-flipping-template*.
- Open the new scene.

10.2.2   To Add a Group

- Add a new **Group** to the scene tree.
- Rename it to *object*.
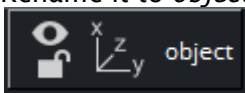
> ⚠ **Note:** All flipping carousel template scenes <u>must</u> have a root container named *object*.

10.2.3   To Add the ControlObject Plug-in

- Drag the **ControlObject** plug-in to the *object* container (found in **Plugins** (/Built Ins) →
  Container Plugins → **Control**, sorted simply as "Object"). Placing the control plug-in on the
  root container optimizes the process, as the whole template scene tree is copied when a new
  template is added, and the root container is the first one to be scanned.
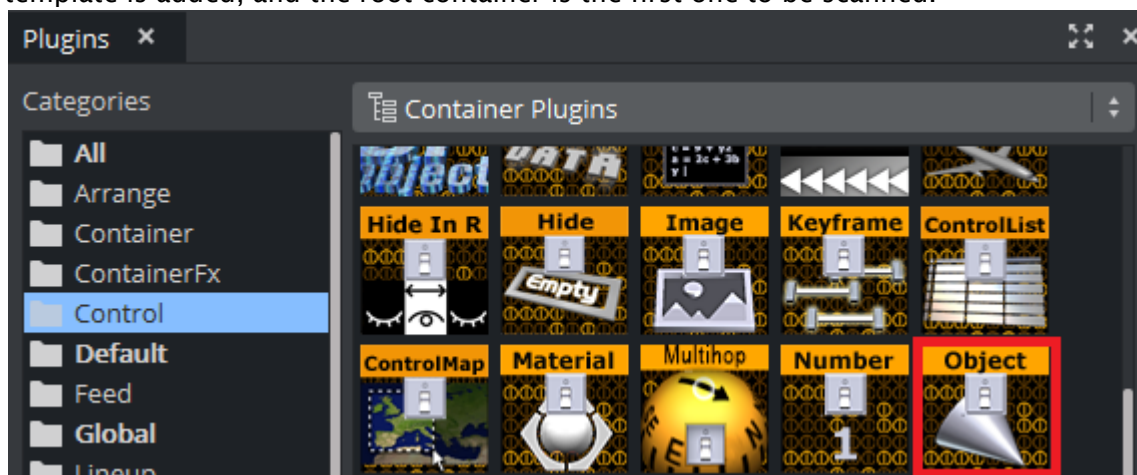
- In the **ControlObject Editor:**
    - Enable **Transition Logic**.
    - Set the **Layer identifier** to `Tutorial`. The *layer identifier* refers to the layer director in the flipping carousel background scene.

        > ✅ **Tip:** The layer identifier should consist of alphanumeric characters only. For more details, see Global Conventions .

    - Set the **State identifier** to `Headlines`. The *state identifier* refers to the *in* state stop point.

        > ✅ **Tip:** The state identifier should consist of alphanumeric characters only. For more details, see Global Conventions .

    - Set the **Background scene** to *tutorial-flipping-back*.
    - Set the **Description** to *simple text message*. The description is used when selecting templates in Ticker Wizard and Viz Ticker Client.
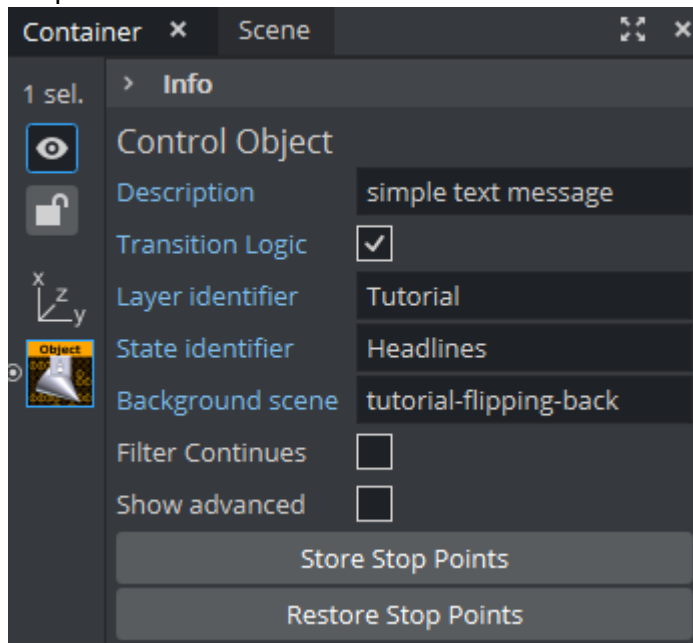
10.2.4   To Add a Font

- Drag a font to the scene tree, for example *Arial-Black*, and place it as a subcontainer of *object* in the scene tree.
- Rename the new container to *text*.

—

- Open the *text* container's ***Transformation Editor*** *and set the* ***Scaling*** *to* X,Y,Z= 0.2.

With this size, the text fits the background.

10.2.5   To Add a Material

Add a material, for example orange (R= 255, G= 102, B= 0), to the *text* container. If no color is added, the font inherits the color of the container holding the Toggle plug-in in the background scene.

10.2.6   To Add the ControlText Plug-in

- Drag the **ControlText** plug-in to the *text* container (found in **Plugins** (/Built Ins) → **Container Plugins** → **Control**, sorted simply as "Text").
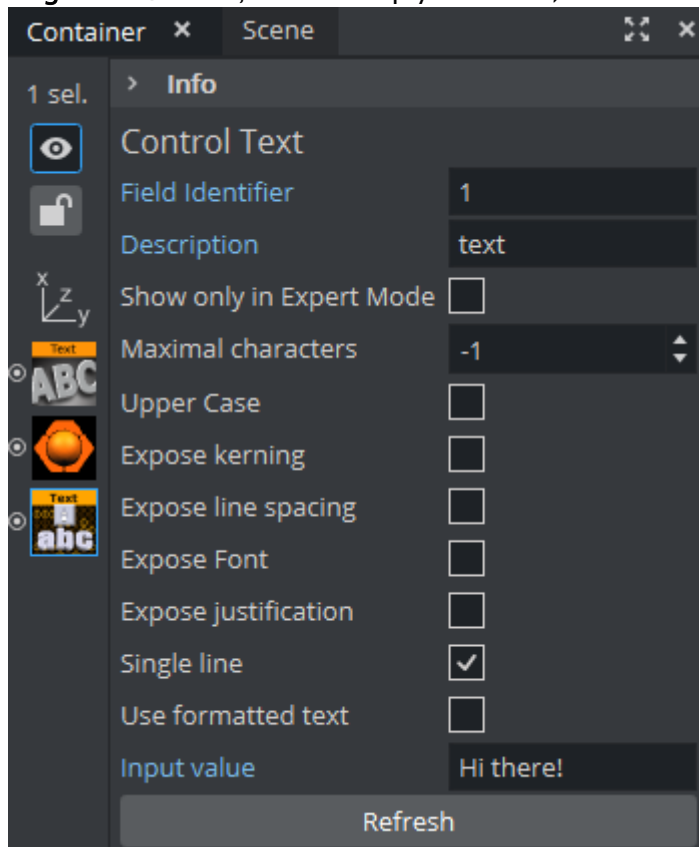
- In the **ControlText Editor**, make sure that the **Expose kerning** option is disabled.
- Disable **Use formatted text**.
- Type *Hi there!* in the **Input value** text box. The scene tree now has the following appearance:
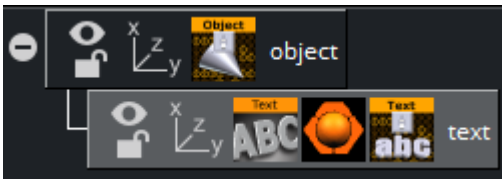
10.2.7   To Fine Tune the Position of the Template

· Open the *object* container's ***Transformation Editor*** *and set the* ***Axis Center*** *to Y=* **C**enter.

When changing the axis center, the position changes too.

· Reset the **Position** to Y= 0. These changes put the center of the template scene in the center of the Toggle plug-in placeholder container.
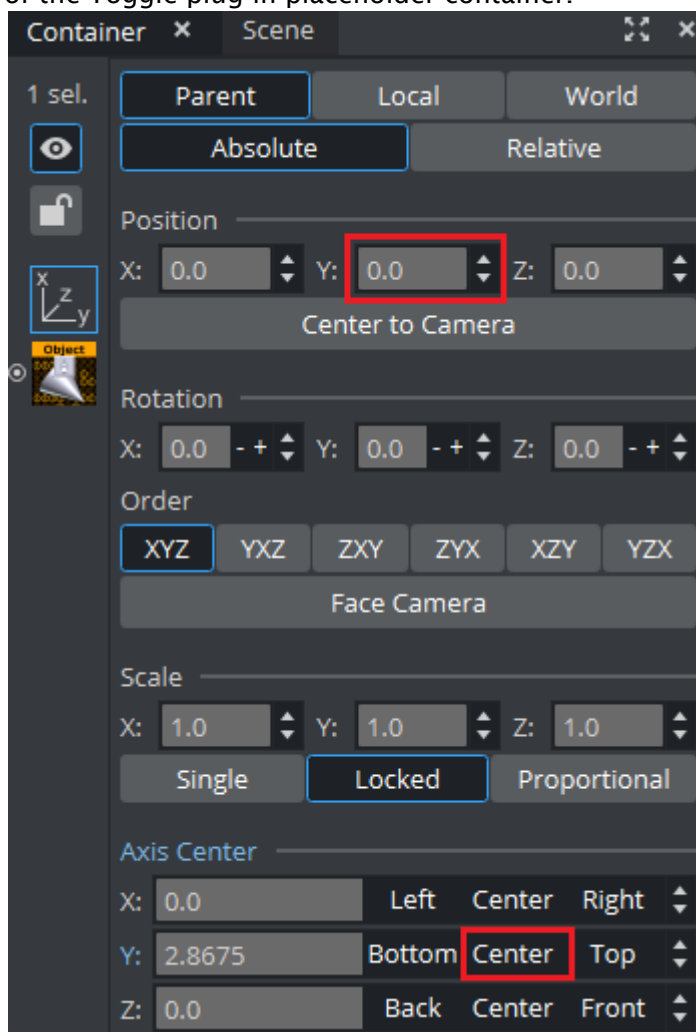
10.2.8   To Save the Scene

The template scene is now finished. Click the **Save** button.

11　Ticker Feed

Ticker Feed is a Viz Ticker add-on program that makes it possible to populate a scrolling or flipping carousel with data elements from an external source. The program can access the file through a URL, which is checked for updates at a defined interval. Ticker Feed communicates with the Ticker System through the Media Sequencer, which is the server process for a Ticker System.
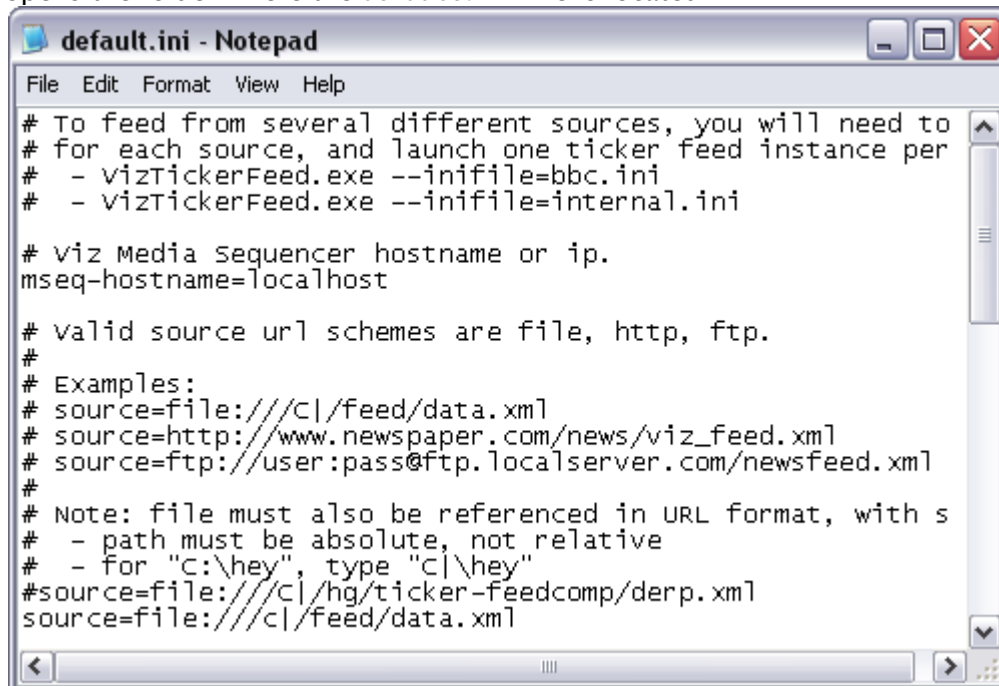
This section contains information on the following topics:

- Getting Started
- System Configuration
- XML Data Structure
- Ticker Feed Troubleshooting
- External Components License

11.1　Getting Started

11.1.1　To Configure a Single Source Ticker Feed

1. Click **Start > All Programs > Vizrt > Viz Ticker > Ticker Feed > Open Settings Folder**. This opens the folder where the `default.ini` file is located.

The `default.ini` file must be created under `%APPDATA%\Vizrt\Ticker Feed\Settings`. The file can be manually copied from the Ticker Feed installation folder, where it is called `default.ini.example`, and then renamed to `default.ini`. Alternatively start Ticker Feed once, and the system automatically performs the copying/renaming operation.

2. Open the `default.ini` file.
3. Define the Media Sequencer Hostname.

> ⓘ **Example:** mseq-hostname=localhost

4. Define the Source XML file that should feed data elements to the Ticker System.

> ⓘ **Example:** source=ftp://user:pass@ftp.localserver.com/newsfeed.xml

5. Define the Transformation.

> ⓘ **Example:**
> transformation=feedcomp:rss20,xsl:filters\rss20_to_TickerFeedExample.xsl

11.1.2   To Configure a Multi-source Ticker Feed

> ⚠ **Note:** To set up a multi-source Ticker Feed configuration, it is necessary to launch one Ticker Feed instance for each source. Each source must also have a separate INI file.

1. Click **Start > All Programs > Vizrt > Viz Ticker > Ticker Feed > Open Settings Folder**. This opens the folder where the `default.ini` file is located. The default.ini file must be created under: `%APPDATA%\Vizrt\Ticker Feed\Settings`. The file can be manually copied from the Ticker Feed installation folder, where it is called `default.ini.example`, and then renamed to `default.ini`. Alternatively start Ticker Feed once, and the system automatically performs the copying/renaming operation.
2. Save the `default.ini` file with a new descriptive name, for example `internal.ini` and place it in the same folder.
3. Create a shortcut of the `TickerFeed.exe` file in the installation directory of Ticker Feed. See Installation Directories.

4. In the Ticker Feed Properties text box, type `--inifile=internal.ini` (if the example file name above has been used).
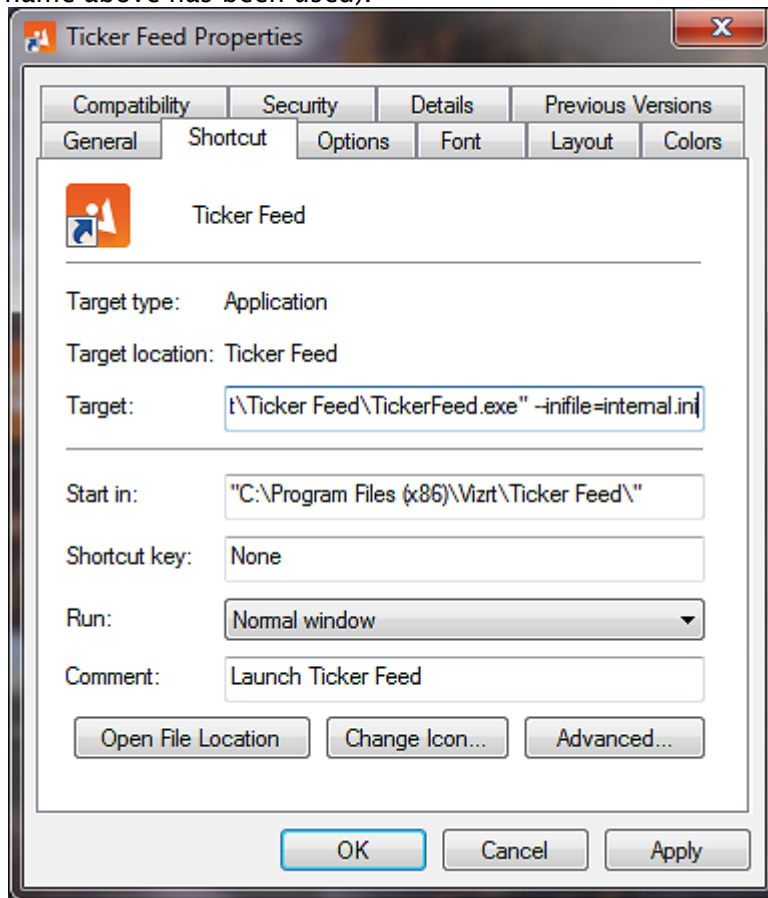
5. Define the Media Sequencer Hostname.

> ⓘ **Example:** mseq-hostname=localhost

6. Define the Source XML file that should feed data elements to the Ticker system.

> ⓘ **Example:** source=ftp://user:pass@ftp.localserver.com/newsfeed.xml

7. Define the Transformation.

> ⓘ **Example:**
> transformation=feedcomp:rss20,xsl:filters\rss20_to_TickerFeedExample.xsl

For all additional sources, repeat the steps in this procedure so that one Ticker Feed instance with a separate INI file can be launched for each source. Make sure to name the INI file and instance parameter differently for each source.

11.1.3   To Start Ticker Feed

> ⚠ **Note:** The correct INI file settings must be entered before the program is started. For more information, see To set up a single source Ticker Feed.

- Click **Start > All Programs > Vizrt > Ticker Feed > Ticker Feed**. When launched, the program starts to monitor a URL and send data to the Media Sequencer.

11.2   System Configuration

The Ticker Feed configuration file is named `default.ini`. The following settings can be defined in the INI file:

- Media Sequencer Hostname
- Source
- Basic HTTP authentication
- Transformation
- Log Level

11.2.1   Media Sequencer Hostname

Ticker Feed communicates with the Ticker System through the Media Sequencer, which is the server process for the system. The Media Sequencer can be targeted either by hostname or IP address.

> ⓘ **Example:** mseq-hostname=localhost

11.2.2   Source

Ticker Feed makes it possible to populate a ticker carousel with data elements from an external source. The program access the file through a URL, which is checked for updates at a defined interval. Valid source URL schemes are **file**, **HTTP**, and **FTP**. For more information about source XML files, see XML Data Structure.

> ⓘ **Example:** File on disk: `source=file:///C|/feed/data.xml`

> ⓘ **Example:** Web Resource (HTTP): `source=http://www.newspaper.com/news/viz\_feed.xml`

> ⓘ **Example:** FTP Server: `source=ftp://user:pass@ftp.localserver.com/newsfeed.xml`

Important Notes about File on Disk

Special rules apply to the URL reference for files on disk:

> ⚠️ **Note:** The path must be absolute, not relative.

> ⓘ **Example:** Instead of `C:\file.xml`, type `C|/file.xml`.

When using a disk file as source, it is advisable to write to a temporary file, and then copy content to the file that is being monitored. This reduces the chance of reading corrupt/incomplete files.

11.2.3 Basic HTTP authentication

For basic HTTP authentication from feeds that have user/password restrictions, you can optionally instruct the URI to be constructed as (example):

```
http://[username]:[password]@hostname/data.xml
```

Some, but not all, browsers accept this basic authentication. Be aware that the username and password are exposed in clear text in the configuration file. This is a security issue if access to the server is exposed. The uid / password must be set in the configuration file. Example:

```
# HTTP basic authentication username and password. # Warning: Using HTTP basic
authentication is potentially insecure since your username and password are #  -
exposed in clear text in this settings file (not an issue if access to the file is
restricted). #  - potentially exposed on an insecure connection (not an issue if the
network itself is trusted or if using a secure connection). #url-username= #url-
password=
```

11.2.4 Transformation

If the source data provides acceptable output in XML format, it is not necessary to define the transformation settings. If the source data must be converted to acceptable XML format, specify one or more transformations of the form:

```
<type-1>:<transformation-1>,<type-2>:<transformation-2>,...
```

Currently recognized transformation types are:

- **xsl**: Specify an XSL file as transformation. If the feed source outputs XML, but of a different format, write an XSL transformation to be applied to the feed data, for example:

  ```
  transformation=xsl:c:\feedtools\yourxml_to_tickerxml.xsl
  ```

- **prog:** Specify an external executable as transformation. The feed data is piped through the program via standard input/output. If for example the feed source outputs data as lines of text, write a VB script to translate this to XML and specify like this:

```
transformation=prog:cscript //NoLogo c:\feedtools\yourdata_to_tickerxml.vbs
```

> ✅ **Tip:** For more information, see the example script in *textfile_to_xml.vbs* . The file can be found at <Installation-Directory>\filters. See Installation Directories .

- **feedcomp:** Apply a built-in transformation to the feed data. Currently, available transformations are:
- **rss20:** Transforms any RSS XML to the RSS 2.0 format. To read any RSS feed and insert it into the ticker, write for example:

```
transformation=feedcomp:rss20,xsl:filters\rss20_to_TickerFeedExample.xsl
```

> ✅ **Tip:** For more information, go to the installation directory of Ticker Feed and see the example script in `rss20_to_TickerFeedExample.xsl`. The file can be found at <Installation-Directory>\filters. See Installation Directories.

If the feed is not trusted, it is possible to apply a censoring filter before inserting the messages to the ticker, for example:

```
transformation=feedcomp:rss20,prog:c:
\feedtools\rss_censor.exe,xsl:rss20_to_TickerFeedExample.xsl
```

Using External Programs to Transform Feed Data

The value specified under the *transformation* key is a transformation pipeline, meaning that it enumerates a sequence of transformation steps that data obtained from the data source passes through before being applied to the Ticker System.

One of the transformation steps that can be configured is a transformation of type *prog*, which specifies an external executable that the data is piped through. This means that the data is piped into the standard-in stream of the executable process, and data is read back out from the program through the standard-out stream of the executable process.

This places the following requirements on executables that can be used for transforming data:

- The executable must read all data provided to it on standard-in fully. The executable cannot choose to not read some of the data provided, since doing so causes a *Pipe Broken* error to be generated when the application exits without having read all available data, and this prevents the transformation pipeline from working.
- The executable must write all outgoing data to standard-out before ending the process.
- The executable must end the process with exit code *0* to indicate that the transformation was successful. Any other exit code is interpreted as an error, and the entire output is

discarded.

If the Transformation Error (see image below) appears in the console or log file, the system has most likely tried to use an external executable that did not read all the data it was given on standard-in before it exited. To fix this problem, modify the external program so that it runs in a loop, reading all available data from standard-in input stream until it reaches the end of file (EOF) marker in the input stream. Only then let the program exit.

```
2007-04-22 16:11:46 ERROR    ScriptFilter   [Errno 32] Broken pipe
Traceback (most recent call last):
  File "feedlib\script_filter.pyc", line 44, in input_anything
IOError: [Errno 32] Broken pipe
2007-04-22 16:11:46 ERROR    ScriptFilter   [Errno 32] Broken pipe
```

11.2.5   Log Level

It is possible to define various logging levels for Ticker Feed. The log level can be set to:

- **debug:** Logs all necessary information, useful for debugging purposes as it maximizes the logging detail.
- **info:** Logs information necessary for normal use. This log level is selected by default.
- **error:** Logs only critical errors.

> (i) **Example:** log_level=debug

11.3   XML Data Structure

> ⚠ **Note:** <tickerfeed version="2.4"> is still used in this version of Ticker Feed. This means that Ticker Feed sources that worked in Ticker Feed 2.4 or later versions also work in this version.

This section of the document contains information on the following topics:

- XML Example
- XML DTD
    - XML DTD Explanation
    - Carousel Overview

11.3.1   XML Example

This section contains an example of the standard XML format for feeding data into a ticker carousel. It corresponds to the Document Type Definition specified in XML DTD.

> ⬥ **IMPORTANT!** If the input source has an XML format that varies from the Ticker Feed XML data structure, it is necessary to create an XSL transformation file to convert it to this format. If the input source is not in XML format, create a program/script that converts the input to XML.

Ticker Feed understands the following data structure:

```
<tickerfeed version="2.4">
  <playlist type="scrolling_carousel" name="news" target="carousel">
    <element>
      <template>grey_text</template>
      <field name="text">Data message element 1</field>
    </element>
    <element>
      <template>grey_text</template>
      <field name="text">Data message element 2</field>
    </element>
  </playlist>
</tickerfeed>
```

11.3.2   XML DTD

The XML format that is used for feeding data into a ticker carousel can also be described as an XML Document Type Definition (DTD):

```
<!ELEMENT tickerfeed (playlist*)>
<!ATTLIST tickerfeed version CDATA #FIXED "2.4">
<!ELEMENT playlist (defaults?,(element|group|section)*)>
<!ATTLIST playlist type (flipping_carousel|scrolling_carousel) #REQUIRED name CDATA
#REQUIRED target (carousel|pool) #REQUIRED>
<!ELEMENT defaults (field|duration|outputchannels|timematch|ttl|template|gui-color|
attributes)*>
<!ELEMENT element (field|duration|outputchannels|timematch|ttl|template|gui-color|
attributes)*>
<!ELEMENT group (description,(outputchannels|timematch|gui-color)*,elements?)>
<!ATTLIST group use_existing CDATA #IMPLIED>
<!ELEMENT description (#PCDATA)>
<!ELEMENT elements (group|section|element)*>
<!ELEMENT section (description,(outputchannels|timematch|gui-color)*,head?,body?)>
<!ATTLIST section use_existing CDATA #IMPLIED>
<!ELEMENT head (element|group|section)*>
<!ELEMENT body (element|group|section)*>
<!ELEMENT field (#PCDATA)>
<!ATTLIST field name CDATA #REQUIRED>
<!ELEMENT duration (#PCDATA)>
<!ELEMENT outputchannels (active*)>
<!ELEMENT active (#PCDATA)>
<!ELEMENT timematch (begin,end)>
<!ELEMENT begin (#PCDATA)>
<!ELEMENT end (#PCDATA)>
<!ELEMENT ttl (#PCDATA)>
<!ATTLIST ttl action CDATA #REQUIRED>
<!ELEMENT template (#PCDATA)>
<!ELEMENT gui-color (#PCDATA)>
<!ELEMENT attributes (attribute*)>
<!ELEMENT attribute (#PCDATA)>
<!ATTLIST attribute name CDATA #REQUIRED>
```

XML DTD Explanation

```
<!ELEMENT tickerfeed (playlist*)>
<!ATTLIST tickerfeed version CDATA #FIXED "2.4">
```

<tickerfeed> is the root container for the entire feed.

- <tickerfeed> can contain any number of <playlist> nodes.

  ```
  <!ELEMENT playlist (defaults?,(element|group|section)*)>
  <!ATTLIST playlist type (flipping_carousel|scrolling_carousel) #REQUIRED name
  CDATA #REQUIRED target (carousel|pool) #REQUIRED>
  ```

<playlist> is a collection of elements, groups, and sections.

- Attribute *type* is either *flipping_carousel* or *scrolling_carousel*.

- Attribute *name* must correspond to the ticker name, as listed in Ticker Wizard. The name does not necessarily equal the tab name in Viz Ticker Client.
- Attribute *target* is either *carousel* or *pool*, and specifies where the elements should be inserted.
- Can contain zero or one <defaults>.
- Can contain any number of <element>, <group>, and <section> nodes.

```
<!ELEMENT element (field|duration|outputchannels|timematch|ttl|template|gui-
color|attributes)*>
```

<element> refers to an actual ticker message. This container has no attributes.

- Can contain any number of <field> nodes.
- Can contain zero or one <duration> node.
- Can contain zero or one <outputchannels> node.
- Can contain zero or one <timematch> node.
- Can contain zero or one <ttl> node.
- Can contain zero or one <template> node. If a <template> node is not present, it must be defined in a <defaults> node.
- Can contain zero or one <gui-color> node.
- Can contain zero or one <attributes> node.

```
<!ELEMENT defaults (field|duration|outputchannels|timematch|ttl|template|gui-
color|attributes)*>
```

A **<defaults>** node is similar to an <element> node, but contains default values that are set on all <element>, <section>, and <group> nodes in the same <playlist> node, unless they explicitly set the values otherwise.

```
<!ELEMENT group (description,(outputchannels|timematch|gui-color)*,elements?)>
<!ATTLIST group use_existing CDATA #IMPLIED>
```

Corresponds to a **group** in the Viz Ticker Client rundown.

- The optional attribute *use_existing* can be set to `yes`. If set otherwise, or missing, it is treated as `no`. When set to `yes`, Ticker Feed does not create a new group, but instead locates and uses an existing group with the same description as this, and uses that instead. If no such group exists, an error is reported and no messages are added.
- Can contain zero or one <outputchannels> node.
- Can contain zero or one <timematch> node.
- Can contain zero or one <gui-color> node.
- Must contain one <description> node.
- Can contain zero or one <elements> nodes.

```
<!ELEMENT description (#PCDATA)>
```

The group's **description**. Visible in the rundown, and the same as can otherwise be manually set in Viz Ticker Client for non-feeded groups.

```
<!ELEMENT elements (group|section|element)*>
```

Constitutes the list of messages under a group.

- Can contain any number of <element>, <group>, and <section> nodes.

```
<!ELEMENT section (description,(outputchannels|timematch|gui-
color)*,head?,body?)>
<!ATTLIST section use_existing CDATA #IMPLIED>
```

Corresponds to a **<section>** in the Viz Ticker Client rundown.

- The optional attribute *use_existing* can be set to yes. If set otherwise, or missing, it is treated as no. When set to yes, Ticker Feed does not create a new group, but instead locates and uses an existing group with the same description as this, and uses that instead. If no such group exists, an error is reported and no messages are added.
- Can contain zero or one <outputchannels> node.
- Can contain zero or one <timematch> node.
- Can contain zero or one <gui-color> node.
- Must contain one <description> node.
- Can contain zero or one <head> nodes.
- Can contain zero or one <body> nodes.

```
<!ELEMENT head (element|group|section)*>
```

<head> constitutes the list of header messages under a section.

- Can contain any number of <element>, <group>, and <section> nodes.

```
<!ELEMENT body (element|group|section)*>
```

<body> constitutes the list of messages under a section.

- Can contain any number of <element>, <group>, and <section> nodes.

```
<!ELEMENT field (#PCDATA)>
<!ATTLIST field name CDATA #REQUIRED>
```

Refers to the data **field**, as designed in the scene template, and also visible as a line in the standard Viz Ticker Client editor.

- Attribute "name" must be the same as the attribute name defined in the template.
- Contains the field value as a text string.

```
<!ELEMENT duration (#PCDATA)>
```

Defines the **duration** of a flipping message.

· Contains duration as a text string. Must be in plain numerical format, with "." as the decimal separator if applicable.

```
<!ELEMENT outputchannels (active*)>
```

Defines which output channels an element/group/section should be enabled for. If **<outputchannels>** is not specified at all, the element/group/section is enabled for all available channels. If specified, the element/group/section is only enabled for the specified channels.

· Can contain any number of <active> nodes.

```
<!ELEMENT active (#PCDATA)>
```

<active> means an element should be enabled for this channel. The channels checkbox is selected in the rundown.

· Contains the output channel name as a text string.

```
<!ELEMENT timematch (begin,end)>
```

<timematch> means an element/group/section should only be considered enabled during a specified time.

· Must contain one <begin> node.
· Must contain one <end> node.

```
<!ELEMENT begin (#PCDATA)>
```

Contains **begin** timecode as text string. If the element should be valid only at a specific time interval, this attribute describes the start time. It can contain either a crontab timecode or a 24-hour hh:mm:ss string, for example *14:30:00*. The crontab timecode is almost in standard Unix crontab format, except the "/" operator is not supported and it has an extra seconds field. For example *00 00 22 01 12 ** , means that the message begins at 22:00 on December 1 playing all days of the week.

```
<!ELEMENT end (#PCDATA)>
```

Contains **end** timecode as text string. The format is equal to the *begin* node.

```
<!ELEMENT ttl (#PCDATA)>
```

If present, means the element should have **<ttl>** enabled.

> ⚠ **Note:** TTL must also have been enabled for the carousel in Ticker Wizard.

```
<!ATTLIST ttl action CDATA #REQUIRED>
```

- · Attribute **action** defines the the ttl action. Can be either "remove", "pool", or "inactive".
- · Contains ttl value as text string. Must be a whole number (integer).

```
<!ELEMENT template (#PCDATA)>
```

The name of the ticker **template** to use. See the carousel list in Ticker Wizard for a list of valid template names.

```
<!ELEMENT gui-color (#PCDATA)>
```

HTML style **color** format. Defines if an element should appear in Viz Ticker Client with a specific row background color, for example *#00FF00* (green).

```
<!ELEMENT attributes (attribute*)>
```

If non standard attributes must be specified for special integration purposes, they can be defined in the **<attributes>** node.

- · Contains any number of <attribute> nodes.

```
<!ELEMENT attribute (#PCDATA)> <!ATTLIST attribute name CDATA #REQUIRED>
```

Specifies an element **attribute**, for example <attribute name="this">that</attribute>.
- · Node attribute "name" is the name of the element attribute to define.
- · Contains element attribute value as a text string.

> ⚠ **Note:** Although it is possible, it is not recommended to use the Time To Live (TTL) attributes when feeding a ticker carousel from Ticker Feed. When feeding a carousel with elements that include the TTL actions *remove* or *pool*, the changes made by these actions, are overwritten by Ticker Feed during the next update that occurs after a feed XML change or Ticker Feed restart. Manually setting an element as inactive or moving it to the pool in Viz Ticker Client likewise overwrites during the next update.

Carousel Overview

The Carousel Overview illustrates the connection between the elements in Viz Ticker Client, the data in the XML file, and the plug-in setup in Viz Artist.

User Interface

Carousel Name

MAIN CONTROL | CAROUSEL1

Pool - U messages

main

Elements in Viz Ticker Client

Data	Date / Time			First text line
Id	Name	Description	Active	
1	First text li...	text1		
2	Second text line	text2		
3	Third text line	text3		
4	Fourth text line	text4		

XML Data

```xml
<?xml version="1.0" encoding="utf-8"?>
<tickerfeed version="2.4">
  <playlist type="flipping_carousel" name="carousel1" target="carousel">
    <element>
      <template>1000</template>
      <field name="1">First text line</field>
      <field name="2">Second text line</field>
      <field name="3">Third text line</field>
      <field name="4">Fourth text line</field>
    </element>
  </playlist>
</tickerfeed>
```

Viz Artist Control Plug-ins in template scene 1000

ControlText Editor

Field Identifier | 1

Description | text1

Ma

ControlText Editor

Field Identifier | 2

Desc

Max

ControlText Editor

Field Identifier | 3

Des

ControlText Editor

Ma Field Identifier | 4

Description | text4

Maximal characters | -1 | R

> ⚠ **Note:** Control Material acts a bit different than other control plug-ins. While other plug-ins use field identifier for setting the value, Control Material uses *fieldid.material:*

```
<field name="2.material">MATERIAL*ABC</field>
```

> ⚠ **Note:** When using Ticker Feed to feed data from a file on disk to a Control Image plug-in, make sure to use forward slashes and not backslashes when defining the path, or the file is not found. Define the path like this:

```
<field name="6">d:/data/tmp/thumb125.jpg</field>.
```

> ⚠ **Note:** Fields containing a "." (dot) in the middle of the text are not part of the auto generated description. This has been done to prevent fields like *25.kerning* or *35.rotation*, which generally is not very descriptive, to pollute the auto description.

11.4 Ticker Feed Troubleshooting

This section describes the most common problem areas that arise when working with Ticker Feed.

- **Q -** Is the XML data visible from Viz Ticker Client when it is coming in? For example, can they open the client and look at the XML fed data?
- **A -** Yes, the XML data is visible like all other elements. It is recommended to make the feed set a specific GUI color on the messages from the feed source, so that they are distinguished from the rest of the (manually created) messages.

- **Q -** Is it possible to change data coming from the XML in Viz Ticker Client? Is it overwritten if reloaded from the XML?
- **A -** The messages can be opened, viewed, and also modified in Viz Ticker Client, but there is no point changing the data manually as the messages are overwritten on the next Ticker Feed update.

- **Q -** Can data be directed to the pool in Viz Ticker Client instead of going straight On Air?
- **A -** Yes, but be aware that if elements are directed to the pool from Ticker Feed, and then manually moved to the carousel, a new instance of each element is recreated in the pool on the next Ticker Feed update.

11.5   External Components License

Ticker Feed uses components from the opensource Universal Feed Parser library, for code see https://code.google.com/p/feedparser/.

This Python library is Copyright:

12 Ticker Service

Ticker Service is a component responsible for storing and controlling the sequence of ticker messages in Viz Ticker. It normally runs as a Windows service but can also be used as a console application. This component is **required** to run a scrolling carousel in Viz Ticker.

> ⚠ **IMPORTANT!** Viz Ticker must be installed and kept running **on the Viz Engine** used for output for scrolling carousels.

To install and start Ticker Service see Installing Ticker Service.

13   Ticker System Tool

From version 3.0, the *Ticker System Tool* is one of the System Applications in the Viz Ticker package. The Ticker System Tool is a non-GUI console application that performs various important system operations, such as archiving (export/import), upgrading and system deletion.

In order to run, this console application requires that Microsoft .NET Framework is installed on the machine. For more details on the required .NET version number and relevant download details, see Requirements.

> ⚠ **Note**: Ticker Wizard and Viz Ticker Client should be closed before making any modifications using the Ticker System Tool. After upgrading the existing system or importing another system, Ticker Wizard and Viz Ticker Client must be restarted for the changes to take effect.

13.1   To Install The Ticker System Tool

The Ticker System Tool is installed together with Ticker Wizard, if this option is selected during installation. After installation, the Ticker System Tool program files are available in the Ticker Wizard installation folder, in the sub-folder called *Ticker System Tool*.

> ✅ **Tip**: For more details, see Installing Ticker Wizard.

13.2   Generic Ticker System Tool Commands

Options relevant for all Ticker System Tool commands:

Option	Description
`--silent`	Disables printing trace to stdout (see also About Logging).
`--verbose`	Prints trace info to stdout (see also About Logging).
`-e`	Prints how long the command ran for.
`-h, --help`	Shows the help message.

> ✅ **Tip**: Use the `help` command to see all supported commands.

13.2.1   Archiving

Archiving can be useful for transferring ticker systems from test to production environments, or interchanging between multiple ticker systems in the same environment.

With the archiving commands, various details for the ticker system can be exported/imported:

- System information such as identifier, name and version.
- Carousel settings.
- Channel settings.
- Playlists (pool and rundown).
- Buttons.
- GPI settings.
- Other settings, such as Graphic Hub configuration and image search provider settings.

Relevant archiving commands:

- Export
- Import

13.2.2   Important Notes

Note that when working with the archiving commands:

- The Ticker System Tool supports the *.ticker* archive type only.
- Scene archiving is not possible.
- Only complete ticker systems can be archived.
- Custom configuration settings, including macros and keyboard shortcut keys, are not archived. For more details, see Configuration Panel.
- A subset of GPIs can be archived, which may be useful when working with Viz Multichannel, Viz Trio or another control application on the same Media Sequencer.

> ◆ **Caution:** GPI archiving is disabled by default. Specified GPIs are overridden during import! Any modifications made to the GPIs since the archive was created (exported) are lost.

Export

The `export` command exports the Ticker System from the given Media Sequencer and creates a *.ticker* archive at the specified path.

- Required:
    - **system_archive:** Path to the archive-file (.ticker).
    - **mse_host:** The Media Sequencer host to connect to.
- Options:
    - **gpi=VALUE:** Includes given GPI.
    - **gpi_all:** Includes all GPIs.

> ⓘ **Example:** `D:\TickerSystemTool.exe` `export` `D:\TickerArchives\MorningNews.ticker` localhost

> ⓘ **Example:** `D:\TickerSystemTool.exe` export `D:\TickerArchives\MorningNews.ticker` localhost --gpi=gpi1 --gpi=gpi2

> ⓘ **Example:** `D:\TickerSystemTool.exe` export `D:\TickerArchives\MorningNews.ticker` localhost --gpi_all

> ⚠ **Note:** If the export operation fails, an output archive is created at the specified path, with a *.fail* extension and time stamp as identifier (for example MorningNews-2016-06-09-180900738.fail).

13.2.3 Import

The `import` command loads the archived Ticker System from the specified *.ticker* file and imports it into the given Media Sequencer.

> 🛑 **Caution:** This command only works if no other ticker system exist on the given Media Sequencer. It is recommended to back up the existing configuration on the Media Sequencer before performing the import operation.

- Required:
  - **system_archive:** Path to the archive-file (.ticker).
  - **mse_host:** The Media Sequencer host to connect to.
- Options:
  - **gpi=VALUE:** Includes given GPI.
  - **gpi_all:** Includes all GPIs.

  > ⓘ **Example:** `D:\TickerSystemTool.exe` import `D:\TickerArchives\MorningNews.ticker` localhost

  > ⓘ **Example:** `D:\TickerSystemTool.exe` import `D:\TickerArchives\MorningNews.ticker` localhost --gpi=gpi1 --gpi=gpi2

  > ⓘ **Example:** `D:\TickerSystemTool.exe` import `D:\TickerArchives\MorningNews.ticker` localhost --gpi_all

13.3 Deletion

> 🛑 **Caution:** It is not possible to recover a deleted system, so use this command with caution.

13.3.1   Delete

The `delete` command deletes the current Ticker System on the given Media Sequencer.

- Required:
    - **mse_host:** The Media Sequencer host from which the system is deleted.

        ⓘ **Example:** `D:\TickerSystemTool.exe` delete localhost

13.4   Summary

The `summary` command provides a quick overview of the current Ticker System on the given Media Sequencer. It can be useful for example when setting up a Ticker system, before exporting, or after importing a system.

⚠ **Note:** This command does not modify the state of the Media Sequencer (it is a read-only operation).

- Required:
    - **mse_host:** The Media Sequencer host on which to print the summary of the Ticker System

        ⓘ **Example:** D:\TickerSystemTool.exe summary localhost

13.5   Upgrading

The Ticker System Tool `upgrade` command supports Ticker System 2.4, or later. For older versions, Ticker Wizard must be used for upgrading, see Upgrading the Existing Viz Ticker Configuration.

13.5.1   Upgrade

The `upgrade` command upgrades the current Ticker System on the given Media Sequencer to the latest (or specified target) standard version.

⚠ **Note:** This command upgrades the Ticker System to the latest version if no *target* is specified.

- Required:
    - **mse_host:** The Media Sequencer host on which to perform the upgrade.
- Options:
    - **-t, --target=VALUE:** The target Ticker System version to upgrade to.

- **-a, --maturity=VALUE:** The maturity of the Ticker System version to upgrade to. Default is empty (release maturity).

> ⓘ **Example:** D:\TickerSystemTool.exe upgrade localhost -t v3.0

13.6 About Logging

The Ticker System Tool supports logging. By default, the system logs to a file and to the console. This logging can be configured, if needed.

13.6.1 Logging Configuration File

The default configuration for the logging can be found in `TickerSystemTool.exe.config`, located in the same folder as the Ticker System Tool program files (for more details, see To Install the Ticker System Tool). The logging configuration can be modified by editing this .config file.

13.6.2 Console Logging

Log information can be sent to the console. A *convenience flag* (silent/verbose) adjusts the console logging. For example, if invoking the Ticker System Tool from a script, it might be appropriate to use the *silent* option. For more details, see Generic Ticker System Tool Commands.

Colors are used to distinguish between various log levels. For example, red is used for errors, yellow for warnings, gray for information, and so on.

13.6.3 Log Files

Ticker System Tool log files are created and stored in the same location as all the other Viz Ticker system applications' log files (for more details, see System Log Files).

> ⓘ **Example:** %AppData%\Vizrt\TickerSystemTool\log\TickerSystemTool-yyyy-MM-dd.log

By default, a 14 day log rotation is applied.

External Components License

The Ticker System Tool uses the *NLog* logging platform to produce log details. The following NLog copyright details apply:

14 Presentation Files

When creating messages in Viz Ticker Client it is possible to use presentation files instead of the default message editor.

This section contains the following topics:

- Using Presentation Files
- Image Search using Presentation Files

14.1 Using Presentation Files

Presentation (*.vtw*) files are created in Template Wizard. From Ticker Wizard, it is possible to add a presentation file to each template when setting up the carousel in one of the Wizards. The presentation file appears in the Elements Editor.

Presentation files must be placed at the following location under the installation directory of Viz Ticker Client (*<Installation-Directory>\presentations\<MyPresentationFile>.vtwc.*). See Installation Directories.

14.1.1 To Prepare a Scrolling Carousel for Presentation File Creation

A *.vtw* template must be based on a scene containing only the template. For flipping carousels, this is not a problem as the background and template scenes are created separately. For scrolling carousels, the background and templates are designed within the same scene. This makes it necessary to create new scenes holding only the container structure of each template. If creating a presentation file for a scrolling carousel, perform the following steps:

1. Start Viz Artist.
2. Open the scrolling carousel scene.
3. Create a copy of the scene, and call it *MyTemplate*.
4. In the new scene, remove all containers except the ones belonging to the template (or the first template if the scene contains more than one template).

> ⚠ **Note:** Remove the *ticker_templates* root container, all other root containers (with subcontainers), and also any subcontainers of *ticker_templates* that does not belong to the first template.

5. Repeat steps 2-4 for all templates, if the scene contains more than one template.

14.1.2 To Create a Presentation File in Template Wizard

> ⚠ **Note:** The presentation files can be added to a carousel when it is first created (as described below), or can be added to existing carousels by going to the Ticker Carousels Panel > Templates Frame > **Edit Templates List**.

1. Start Template Wizard.

> ⚠ **Note:** Template Wizard is installed as part of the Viz Pilot package.

2. From the **Main** menu, click **File > Wizard**.
3. From the **Wizard** window that opens, navigate to the template scene.

> ⚠ **Note:** A *.vtw* template must be based on a scene containing only the template. For a scrolling carousel, a separate template scene must be created, see To prepare a scrolling carousel for presentation file creation.

4. Double click the template scene.
5. Click **Next**.
6. Select the elements that should be included in the *.vtw* template.
7. Click **Finish**.
8. Back in the **Main** window, rename and drag the elements so that they suit the desired layout.
9. Resize the template so that it fits the content. If the template is too big to fit the Elements Editor in Viz Ticker Client, scroll bars appear.
10. From the **Main** menu, click **File** > **Save to file**.
11. In the **Save Templates** window that opens, give the template a descriptive name, for example *MyPresentationFile*.
12. Click **Save**.

For detailed instructions on how to operate Template Wizard, see the *Viz Pilot User Guide*.

For instructions on how to add the *.vtw* file to a template, see To select the presentation file for a template.

14.2   Image Search Using Presentation Files

Image Search is a feature that allows you to search and select images using Vizrt Asset Search providers (Viz Graphic Hub, Pilot Data Server, Viz One, etc.). The Ticker System can be configured to support Image Search using template presentations (VTW). Search providers can be stored on either Viz Graphic Hub or Pilot Data Server. See Image Search Provider Frame.

14.2.1   Presentation Files for Image Components

Template Wizard is a component of Viz Pilot and is used to create VTW presentation files (.vtw/.vtwc). The screenshot below shows an example VTW presentation file with an image component. To enable Image Search, the "IsImageMediaSearch" property of the image component should be checked. Now the VTW presentation file is ready to be used in Viz Ticker Client.

For more information on the *Image Component* see the Viz Pilot User Guide.
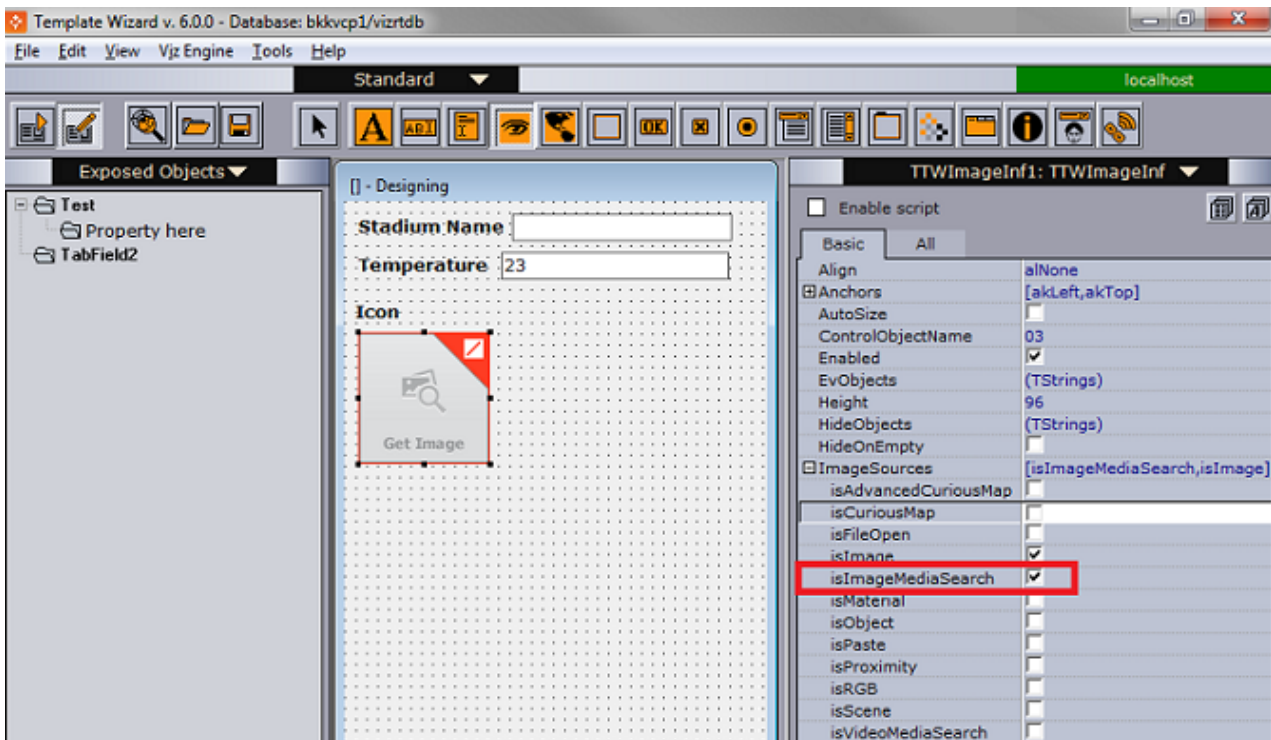
For more details see To Create a Presentation File in Template Wizard.

14.2.2 Image Search Configuration

Some configuration is required in Ticker Wizard to have Viz Ticker Client launch Image Search from VTW presentation files. For details, see Image Search Provider Frame.

14.2.3 Using Image Search

See Adding an Image using Media Search.

15 Field Linking And Feed Browsing

Viz Ticker supports browsing tab field values from external sources. Instead of editing tab field values manually, values from an external feed can be selected.

Field Linking assists the user in filling in data by selecting items from a feed rather than editing the value(s) manually. The feedsource can for instance be a network URI or a local XML file. Each Ticker message then uses the feed browser in Viz Ticker Client to edit the tab field value. The specified feed URI in Viz Ticker overwrites any existing feed URI from the control object.

When defining and configuring the field linking in Ticker Wizard you can select to make the item values returned from a feed read-only. By default the values returned from a feed linked item fetched with Viz Ticker Client feed browser can be edited. For more information see Field Linking with edit restrictions.

> ⚠ **Note:** Viz Ticker supports *Atom 1.0* formatted feeds. See the Technical Reference for Field Linking section for more information.

A field in general has several properties. Each property relates to a control plug-in in the scene. Properties are grouped under a tab field by a naming convention. To be able to set the values of multiple tabfield properties from the same feed entry, the properties must be grouped together under the tabfield. This is done by giving them the same prefix followed by the period character.

For example, the following tabfield names generates two groups of tabfields (`candidate1` and `candidate2`):

```
candidate1.name candidate1.image candidate2.name candidate2.image
```

A further example to illustrate tab field grouping:

The ControlText plug-ins specifying the field identifiers 1.*name*, 1.*score* and 1.*image* forms a tab field *1* with the properties *name*, *score* and *image*. Selecting a feed item for the tab field *1* should apply the parts of the selected item to the properties *name*, *score* and *image* at once. The feed linking is usable across all tab field types.

> ⚠ **Note:** Make sure to define matching object types; a thumbnail link for control image tab fields, and so on.

This section contains information on the following topics:

- Workflow
- Field Linking
- Feed Browsing
- Shared Memory Support
- Technical Reference for Field Linking

15.1   Workflow

Field linking in Viz Ticker is a two step procedure:

- **Field Linking:** Linking the fields in a template to various parts of data that comes from entries in a feed. This is a setup step that is performed in Ticker Wizard, and is usually done once for a template.
- **Feed Browsing:** After field linking for a template has been successfully set up, a property editor for selecting an entry from a feed is available in Viz Ticker Client's Elements Editor. This feed browser lets you select an item in a feed that contains the values that should be applied to the field values.

See Also

- Field Linking
- To Configure Field Linking in Ticker Wizard
- Feed Browsing
- To Browse Feed Data in Viz Ticker Client

15.2   Field Linking

Field linking can be configured in Ticker Wizard.

15.2.1   To Configure Field Linking in Ticker Wizard

1. Scan and set up a carousel in one of the Wizards.
2. Open the carousel in the Ticker Carousels Panel.
3. Right-click a template from the Templates Frame.
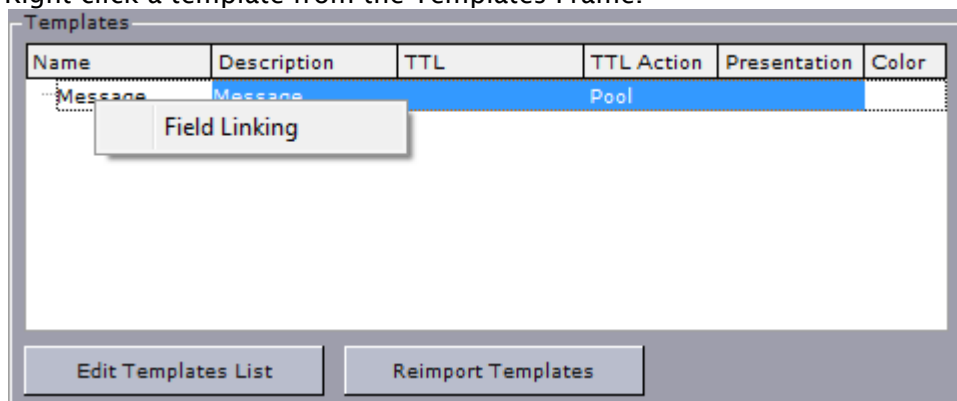
4. In the popup that appears, click **Field Linking.**
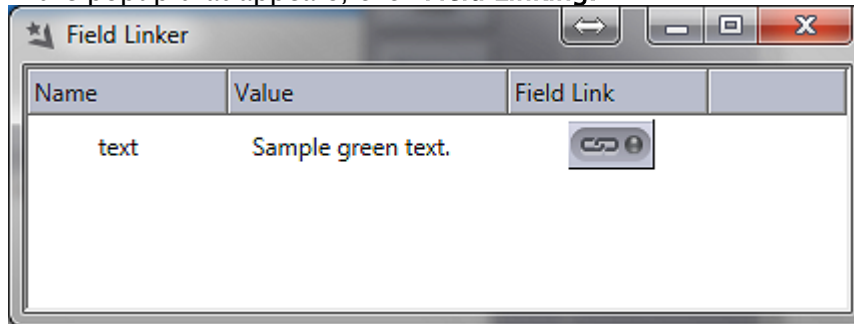
5. In the window that opens, click the **Field Link** button next to the field that you want to update.
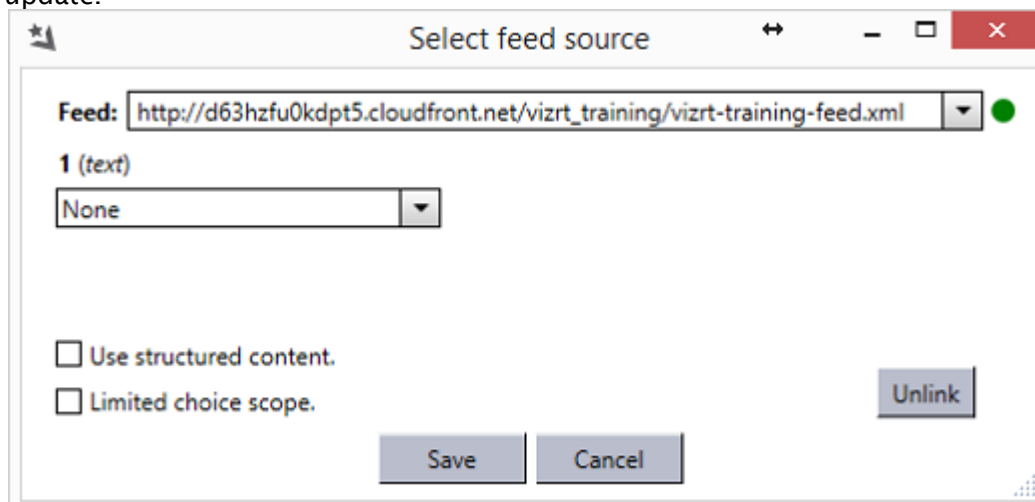
6. In the window that opens, specify the feed URI in the **Feed** text box.

> ⚠ **Note:** The feed source can also be located on the file system. Assuming a file is located at `C:\feeds\feed.xml` then it can be referenced by file:///C:/feeds/feed.xml. Notice it has to be a well-formed URI.

7. Specify which part of the selected feed item should be used as value for this field; content, thumbnail, title, and so on. If the field is grouped the user can select multiple parts/ properties to be linked.

> ⚠ **Note:** Make sure to define matching object types; a thumbnail link for control image tab fields, and so on.

8. *Optional:* Enable the **Use structured content** check box to automatically map tabfield values from the VDF payload content in the selected Atom entry. In order to use structured content, the feed source must serve out data in the VDF (Viz Data Format) defined format.
9. *Optional:* Enable the **Limited choice scope** check box to restrict the field linking choices. Enabling this option means that the user browsing the feed *cannot change the values* of items returned from the feed. Effectively this option makes browsing the feed a read-only operation. For more information see Feed Browsing in Viz Ticker Client and Field Linking with Edit Restrictions.

> ⚠ **Note:** Linking to an ATOM feed is the property of a template in Viz Ticker and is therefore shared by all messages using that template.

10. Click the **Save** button.
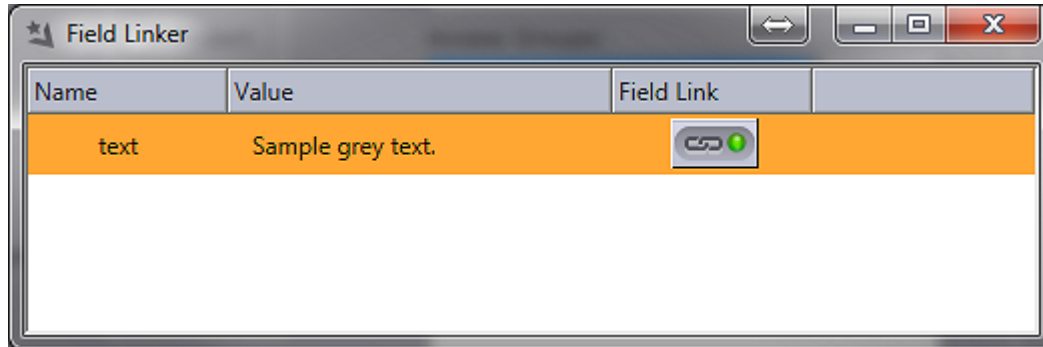
A green icon in the **Field Linker** window reflects a configured Field Linking connection.

> ⚠ **Note:** The green icon only indicates that there has been set up a field linking configuration for this particular field. It does not verify that the URL is correct, available, and so on. By holding the pointer over the icon, the configured URL is shown as a tooltip.

If the template contains more than one field that should be linked, select additional **Field Link** buttons, and repeat steps 6-9.

To continue the workflow, and actually use this feed linking setup as the source for carousel creation in Viz Ticker, follow the steps in To Browse Feed Data in Viz Ticker Client.

15.3   Feed Browsing

Feed browsing is the process of fetching carousel content from a working Field Linking instance.

15.3.1   Feed Browsing in Viz Ticker Client

After setting up a working Field Linking instance in Ticker Wizard, the feed can be used to generate output for ticker carousels In Viz Ticker Client.

To Browse Feed Data in Viz Ticker Client

1. Make sure that a valid field linking feed has been configured, see To Configure Field Linking in Ticker Wizard.
2. In Viz Ticker Client, create a message or open the message that should be edited. Instead of the regular look of the editor, the Elements Editor contains a Feed Browsing Icon **if** a feed has been configured to one of the fields in the template.
3. Click the Feed Browsing Icon.

While browsing the feed you can right click on the property editor to bring up the context menu that enables various options for displaying and sorting the feed.

- Grid View
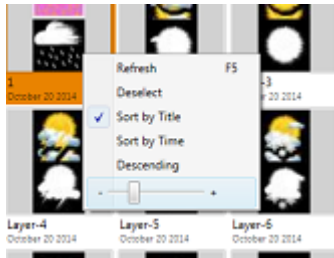
- List View
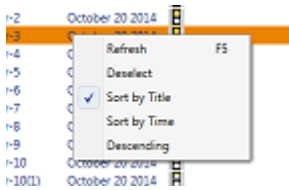
4. In the Feed Browsing Property Editor that opens, double-click the feed item that serves as content for the field. Use the control icons on the right-hand side to choose style (tabular etc.) for the feed data. You can drill down results shown using the Feed Browsing search field to filter the results.
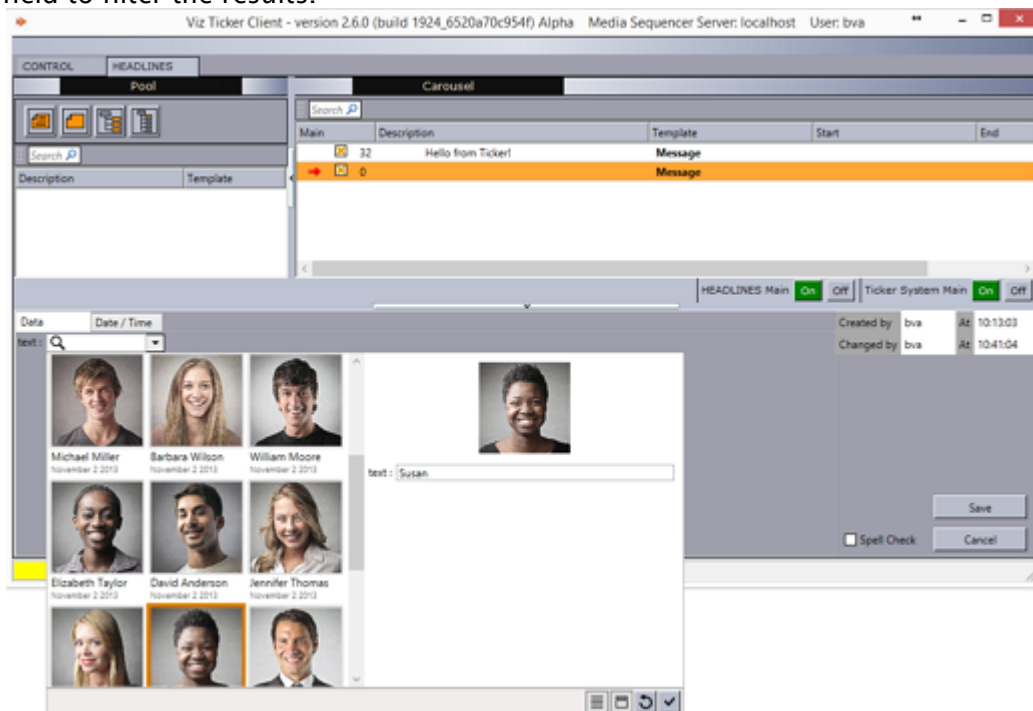
The content being fetched depends on which field has been designed to be linked, whether it is a title, thumbnail and so on.

5. Back in the Elements Editor, click the **Save** button. The message now contains content from the configured feed. While browsing a feed, the icons below are displayed in the feed browser window. The two icons on the left are switches and change depending on current

selection as explained in the table below:

Icon	Action when clicked
☰	Displays the feed as list view (tabular format).
▤	Hides details view.
▥	Shows details view.
▦	Shows as grid.
↺	Resets the field value to the value from the feed.
✓	Selects current item and saves value.

15.3.2   Field Linking with Edit Restrictions

As explained in the previous section Feed Browsing in Viz Ticker Client, the user by default can *edit* the values of items returned from a feed. This changes the payload and the changed item value used in tickers. Usually, this is appropriate because it provides flexibility. However, if the user doing the feed browsing should be restricted from changing values received from the feed this can be achieved by setting the field linking option *Limited choice scope* when setting up and configuring the field linking. If this option is active for the feed, items received from the feed cannot be changed, effectively reducing the feed to read-only.

15.3.3   Searching and Filtering

If the feed supports *OpenSearch*, then the search box is enabled and all searches are done on the server. If not, then the search box is a local text filter box.

See Also

· OpenSearch link

15.4   Shared Memory Support

Viz Ticker supports the Viz Engine Shared Memory (**SHM**) functionality. Combining Field Linking with the Viz Engine *ApplySharedMemory* plug-in provides an easy way to update data instantly. A typical scenario would be updating stock prices where the actual stock values are fetched from external data sources and the ticker displaying the stock values are updated in near realtime.

The combination of Field Linking and SHM allows the user of Viz Ticker Client to simply select an item using Field Linking, which behind the scenes extracts information from the feed item, including the **locator** element (the key used to set the data in the *ApplySharedMemory* plug-in). Realtime data can be sent to the *ApplySharedMemory* plug-in using any application that supports it, for instance Viz DataHub. It could also be any custom application sending data via UDP, TCP or by sending commands directly to the Viz Engine.

> ❗ **IMPORTANT!** This functionality requires Viz Engine 3.7 (REV 43635) or above.

15.4.1   Configure Field Linking and SHM support

The Viz Engine used must be configured for Shared Memory Support and the port used for communication must be agreed upon. In Viz Artist go to **Configuration > Communication > Shared Memory-tab**. In the example below, we use TCP on port `11000` for communication.
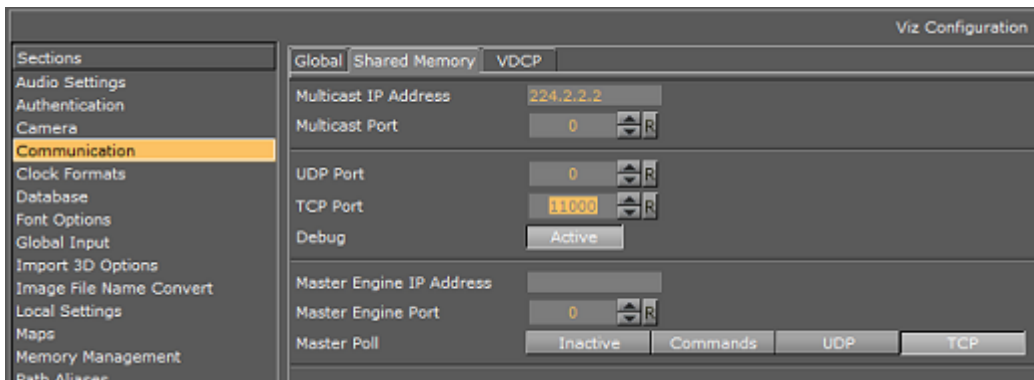
Design and configure the scene to be used as a ticker for SHM. A typical scene tree is shown below:

Make sure the *ApplySharedMemory* plug-in is configured:
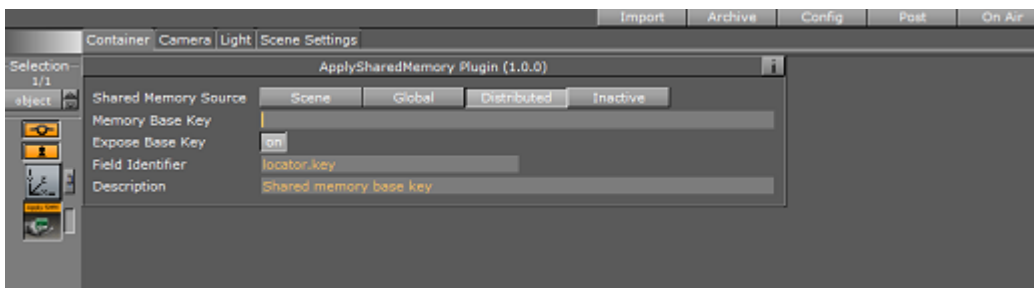
You can now use Ticker Wizard and Field Linking with Shared Memory support using the SHM key/ value combination you have chosen.

15.4.2　Realtime data updates

Use any custom application to send updates to the port you have configured for SHM use. The message must include the SHM key and the value. The key should be in */ YOUR_SHARED_MEMORY_BASE_KEY/PERHAPS_WITH_PATH/value* format.

The Viz Engine is shipped with an example implementation (using C#) showing you how to send data to the engine. The source is typically installed in the directory *<Viz Artist-Engine Installation Directory>\SendToSMM*. Using this requires compiler and programming experience, so end users should have other tools prepared for sending data to SHM enabled tickers.

15.4.3　Technical Specification

After setting up SHM use in the Viz Artist scene and Field Linking in Ticker Wizard the vdf (Viz Data Model) look like this example:

```
<model
    xmlns="http://www.vizrt.com/types">
    <schema>
        <fielddef name="locator">
            <choice scope="suggest">
                <collection src="source.xml" />
            </choice>
        <fielddef name="key" label="Shared memory base key" mediatype="application/
vnd.vizrt.smmkey">
            <fromatom select="locator" />
            <value></value>
        </fielddef>        (...)
    </schema>
</model>
```

The locator element can then be extracted from atom elements like this:

```
<entry>
    <id>MSFT</id>
    <published>2012-04-08T16:42:58+07:00</published>
    <updated>2012-09-25T10:34:29Z</updated>
    <title type="text">Microsoft</title>
    <locator type='application/vnd.vizrt.smmkey'
        xmlns='http://www.vizrt.com/atom-ext'>/stocks/random/MSFT
    </locator>
</entry>
```

> ⚠ **Note:** Notice that namespace of the <locator /> element is the vizrt *atom-ext* and the type is *application/vnd.vizrt.smmkey*.

See Also

- See the sections about *Apply Shared memory plugin* and *Viz Artist Shared Memory* in the Viz Artist Guide
- Viz Engine Guide

15.5   Technical Reference For Field Linking

This section contains information on the following topics:

- Feed Linking Formats
- XML Namespaces
- Elements in Atom Feed
- Elements in Atom Entry
- Fields that can be Linked

· Fields that cannot be Linked

15.5.1   Feed Linking Formats

Viz Ticker supports *Atom 1.0* formatted feeds.

The feed browser supports two types of feeds:

· Flat feeds.
· Hierarchical feeds (folder structure).

Linking to Social Media Sources

Some of the most popular web feeds, such as Twitter and Facebook, output content in non-supported formats or they use proprietary APIs. In order to get these and other Social feeds working together with Viz Ticker, the Ticker System must be combined with Vizrt's Social TV solution (*Feed Streamer*). The Social TV solution then captures content from the social feeds and pushes moderated content to Viz Ticker wrapped in an open standard based network container format (example: Atom/XML) that can be combined with Ticker.

15.5.2   XML Namespaces

The following XML Namespace Prefixes are used when referring to XML elements:

XML Namespace Prefixes

Prefix	URL
atom	http://www.w3.org/2005/Atom
viz	http://www.vizrt.com/types
vaext	http://www.vizrt.com/atom-ext
media	http://search.yahoo.com/mrss/
thr	http://purl.org/syndication/thread/1.0
opensearch	http://a9.com/-/spec/opensearch/1.1/

> ⓘ **Example:** The notation `<atom:entry>` is to be interpreted as referring to the same element as `<entry xmlns="`http://www.w3.org/2005/Atom`">`.

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

15.5.3   Elements in Atom Feed

Elements that should be present in the `<atom:feed>` for the full user experience:

OpenSearch Link

```
<atom:link rel="search" type="application/opensearchdescription+xml" href="http://example.com/opensearchdescription.xml"/>
```

- · Optional, needed for support for server side searching.
- · The opensearchdescription XML file must contain a template node that returns search results as an atom feed:

```
<opensearch:Url type="application/atom+xml" template="http://..." />
```

- · For more details see http://www.opensearch.org/Specifications/OpenSearch/1.1

Up Link

```
<atom:link rel="up" type="application/atom+xml;type=feed" href="http://..." />
```

- · Optional, needed for nested collections.
- · The up link should link to the parent folder feed for hierarchical feeds.
- · The up link must have a type string equal to `"application/atom+xml;type=feed"`.

15.5.4   Elements in Atom Entry

Elements that should be present in the `<atom:entry>` for the full user experience:

Self Link

```
<atom:link rel="self" type="application/atom+xml;type=entry" href="http://..." />
```

- · The self link must link to the URL that returns the `<atom:entry>` XML.
  - · The self link must have a type value equal to `"application/atom+xml;type=entry"`.
  - · Needed for refresh/update of values from the item to work.

· Needed in combination with the up link for remembering and showing the selected feed entry in a hierarchy of feeds (folder structure). Not needed for remembering selection in flat feeds since then the `atom:id` is used.

Up Link

```
<atom:link rel="up" type="application/atom+xml;type=feed" href="http://..." />
```

· The up link must have a HREF value that is the URL of the feed the entry is in.
    · The up link must have a type value equal to `"application/atom+xml;type=feed"`.
    · Needed in combination with the self link for remembering and showing the selected feed entry in a folder structure.

Down Link

```
<atom:link rel="down" type="application/atom+xml;type=feed" href="http://..." />
```

· Needed if this `<atom:entry>` is to be considered a subfolder instead of a normal `<atom:entry>`.
    · The down link must have a type value equal to `"application/atom+xml;type=feed"`.
    · The down link must have a HREF value that is the URL of the feed (folder) the entry represents.
    · Link may contain `thr:count` attribute (RFC4685) indicating how many children there are. If the value of `thr:count` is `0` (zero) then the folder is not loaded since it is empty. This is an optimization.

Thumbnail Link

```
<media:thumbnail url="http://..."/>
```

· Needed for thumbnail icons to display on the entries in the feed browser.
    · The URL value must reference a JPEG or PNG image resource.
    · If many thumbnails are defined, the first one is selected as the default. This is according to Media RSS Specification Version 1.5.0.

Other Links

```
<link rel="..." href="http://..." type="..."/>
```

- Needed in order to select the URL to external resources. This can be images, video, text, and so on.
  - Limitation 1: The Atom 1.0 specs allow multiple links with the same relation, but when linking a tabfield to a link in Viz Ticker, only the first link is selected.
  - Limitation 2: Viz Ticker can only use the `href` URL as the value. It cannot download the resource at the URL and use its contents as the value. The Viz Engine control plug-in for the property has to be able to understand the URL and download the resource.

15.5.5   Fields that can be Linked

Field	Details
`<atom:author>`	The personal details of the author of the selected `<atom:entry>`, which might be an individual person or an organization. The data is provided in child elements as follows: • `<atom:name>`: The name of the author. The value of the `<atom:name>` element inside the first `<atom:author>` element that applies for the entry is recorded as the field value. This child element is required if you are specifying the `<atom:author>` element. • `<atom:uri>`: A URL associated with the author, such as a blog site or a company web site. The value of the `<atom:uri>` element inside the first `<atom:author>` element that applies for the entry is recorded as the field value. This child element is optional. • `<atom:email>`: The email address of the author. The value of the `<atom:email>` element inside the first `<atom:author>` element that applies for the entry is recorded as the field value. This child element is optional.
`<atom:content>`	• Both inline and URL content is supported. • There is no type match, so it is up to the Viz Ticker operator to figure out what content can be used. • For URL content the resource is downloaded first and then applied as the value. If you just want the URL you should use `<atom:link>`.
`<atom:entry>`	The whole entry XML.
`<atom:link>`	This element defines a reference from an entry to a Web resource; in other words this is the value of the `href` attribute.

Field	Details
<vaext:locator>	The text of the Vizrt Atom Extension `<vaext:locator>` element which has a `vaext:type` attribute equal to the "mediatype" attribute of `<viz:fielddef>` element in the model.
<atom:published>	This element is a Date construct indicating the initial creation or first availability of the entry. The value of the `<atom:published>` element in the entry is recorded as the field value.
<atom:summary>	This element is a Text construct that conveys a short summary, abstract, or excerpt of an entry. The content of the `<atom:summary>` element in the entry is recorded as the field value.
`<atom:thumbnail>`	The value of the URL attribute of the first thumbnail element in the entry is recorded as the field value.
<atom:title>	This element is a Text construct that conveys a human readable title for an entry or feed; the title of the selected `<atom:entry>`.
<atom:updated>	This element is a Date construct indicating the most recent instant in time when the selected `<atom:entry>` was modified. The value of the `<atom:updated>` element in the entry is recorded as the field value.

15.5.6   Fields that cannot be Linked

· `<atom:category>`
· `<atom:id>`
· `<atom:rights>`
· Any other elements.

See Also

· Social TV User Guide

16   REST Interface And Services

The Media Sequencer contains two APIs for interacting with the Ticker system via HTTP REST. Both are discoverable from the Media Sequencer Service Document. Please consult the Media Sequencer documentation on http://localhost:8580/doc on how to use the the **Actions** API and the **Ticker Button** API (new in Media Sequencer 5.4).

17   Keyboard Shortcuts

The following keyboard shortcuts can be used to operate Viz Ticker:

- Viz Ticker Client and Ticker Wizard
- Viz Ticker Client's Pool Frame
- Viz Ticker Client's Carousel Frame
- Viz Ticker Client's Elements Editor

17.1   Viz Ticker Client And Ticker Wizard

Function	Key + Mouse Combination
Select the next tab in the Main Menu	CTRL + TAB

17.2   Viz Ticker Client's Pool Frame

Function	Key + Mouse Combination
Jump to the Pool Search text box	CTRL + F
Copy currently selected messages	CTRL + C
Cut currently selected messages	CTRL + X
Delete currently selected messages	DELETE
Open a new message in Elements Editor based on the default template	CTRL + N
Open the **Select Template** dialog box, where any available message template can be selected	CTRL + T
Paste messages	CTRL + V
Select all messages	CTRL + A

17.3   Viz Ticker Client's Carousel Frame

Function	Key + Mouse Combination
Activate the **Find** text box in the Carousel Search	CTRL + F
Activate currently selected messages	INSERT
Deactivate currently selected messages	CTRL + INSERT
Delete currently selected messages	DELETE
Copy currently selected messages	CTRL + C
Cut currently selected messages	CTRL + X
Paste messages	CTRL + V
Select all messages	CTRL + A

## 17.4   Viz Ticker Client's Elements Editor

Function	Key + Mouse Combination
Save the current element	CTRL + S

18   Troubleshooting

This section contains troubleshooting information on the following topics:

- Not Possible to Scan Scene in Ticker Wizard
- Scrolling Carousel Messages Not On Air
- The System Seems to Freeze when Creating a Message
- Unable to Connect to Viz Engine
- Begin and End Times from Ticker Feed

## 18.1   Not Possible To Scan Scene In Ticker Wizard
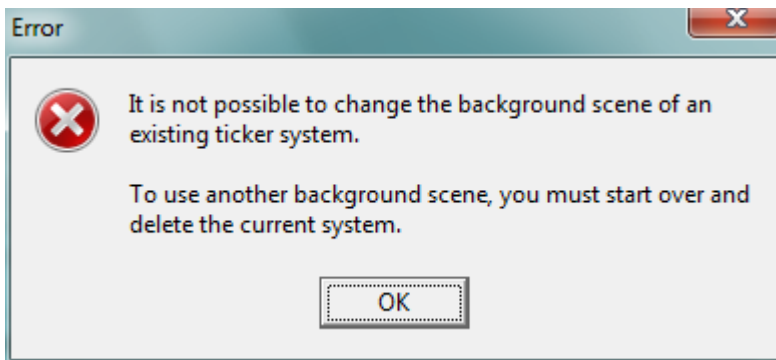
18.1.1   Problem

I want to scan a new scene in Ticker Wizard, but the system prevents me from doing this and the error message above appears: *"It is not possible to change the background scene of an existing ticker system. To use another background scene, you must start over and delete the current system."*

18.1.2   Solution

To scan a new scene, any existing setup must be deleted. To do so, perform the following steps:

1. Delete any existing carousels, see To Delete a Carousel.
2. Delete any existing output channels, see To Delete an Output Channel.
3. Try to scan the scene again, see To Scan a Scene. The scene should now be successfully scanned.

18.2   Scrolling Carousel Messages Not On Air
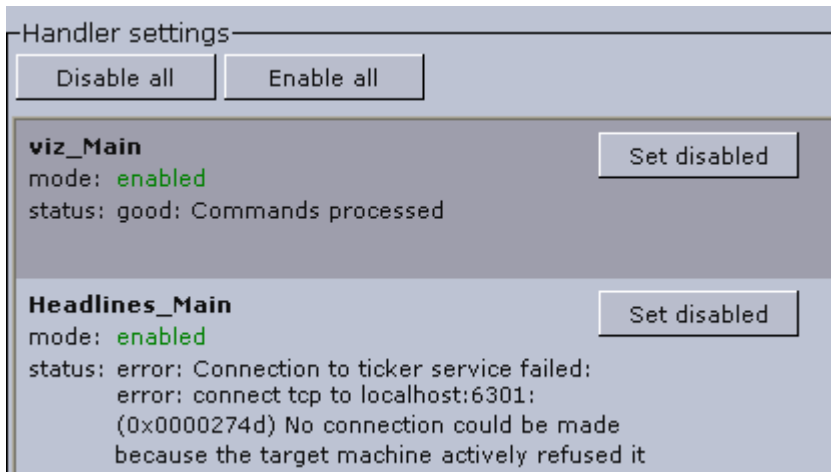
18.2.1   Problem

I want to take scrolling carousel messages On Air, but the messages do not appear as expected.
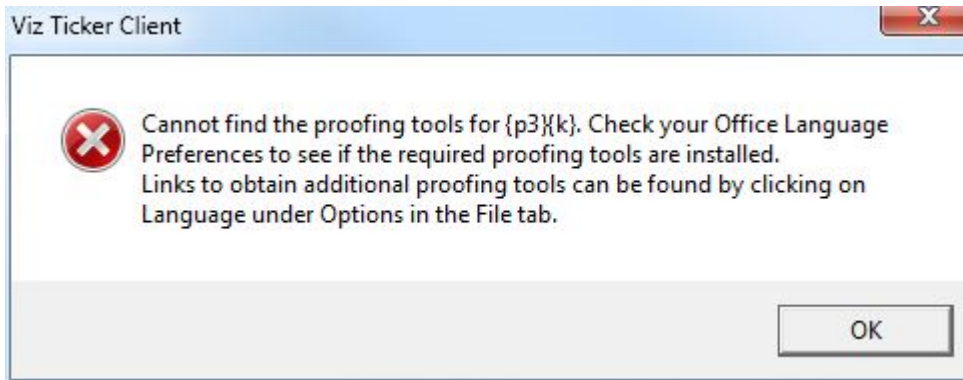
18.2.2   Solution

This problem can be caused by various errors.

· First, make sure that the Ticker Service is running. You can check if this Viz Ticker component is running, from the Handler Settings Frame in Viz Ticker Client. If the error message above appears, *error: Connection to ticker service failed*, Ticker Service is not running. The service can be launched from **Start > All Programs > Vizrt > Viz Ticker > Ticker Service**, or can be launched as a service.

> ⚠ **Note:** On **Windows 10**, Ticker Service must be run with administrator privileges. For more details, see To Start Ticker Service as a Service.

· If you are sure Ticker Service is running successfully, continue the problem-solving process by making sure that the scene is designed according to the scrolling carousel Design Conventions.

18.3   The System Seems To Freeze When Creating A Message

18.3.1   Problem

I want to create a message in Viz Ticker Client, while the spell check feature is activated, but the system seems to freeze when I try to do this.

18.3.2   Solution

An error message similar to the above may appear in the background when trying to create a message while the spell check feature is activated: *Cannot find the proofing tools for <your default language>. Check your Office Language Preferences to see if the required proofing tools are installed.* This dialog box must be closed for the system to work.

The spell check function is based on the Microsoft Office dictionary, and uses the default language in Microsoft Word. For the Viz Ticker spell check function to work, Microsoft Word must be installed on the machine, and the proofing tools must be updated accordingly.

18.4   Unable To Connect To Viz Engine

18.4.1   Problem

For some reason, the Ticker System is not successfully connected to the specified Viz Engine.

18.4.2   Solution

The Viz Engine may have been shut down, taken off air, or similar. For details about the Viz Engine connection status, see Status of the Viz Engine Connection.

18.5   Begin And End Times From Ticker Feed

18.5.1   Problem

Can I set the begin/end times for messages in the Ticker Feed source XML and then view/edit it in Viz Ticker Client?

18.5.2   Solution

It is not recommend to use Viz Ticker Client to edit messages controlled by Ticker Feed since manual changes are overridden on the next Ticker Feed update. Furthermore, Start/End time may show incorrectly in the UI when setting the begin/end elements in the Ticker Feed source XML since the UI can only be used to visualize certain begin/end combinations.